Proteome Informatics

New Developments in Mass Spectrometry

Editor-in-Chief:
Professor Simon J. Gaskell, *Queen Mary University of London, UK*

Series Editors:
Professor Ron M. A. Heeren, *Maastricht University, The Netherlands*
Professor Robert C. Murphy, *University of Colorado*, *Denver*, *USA*
Professor Mitsutoshi Setou, *Hamamatsu University School of Medicine, Japan*

Titles in the Series:
1: Quantitative Proteomics
2: Ambient Ionization Mass Spectrometry
3: Sector Field Mass Spectrometry for Elemental and Isotopic Analysis
4: Tandem Mass Spectrometry of Lipids: Molecular Analysis of Complex Lipids
5: Proteome Informatics

How to obtain future titles on publication:
A standing order plan is available for this series. A standing order will bring delivery of each new volume immediately on publication.

For further information please contact:
Book Sales Department, Royal Society of Chemistry, Thomas Graham House, Science Park, Milton Road, Cambridge, CB4 0WF, UK
Telephone: +44 (0)1223 420066, Fax: +44 (0)1223 420247
Email: booksales@rsc.org
Visit our website at www.rsc.org/books

Proteome Informatics

Edited by

Conrad Bessant
Queen Mary University of London , UK
Email: c.bessant@qmul.ac.uk

ROYAL SOCIETY OF CHEMISTRY

THE QUEEN'S AWARDS FOR ENTERPRISE: INTERNATIONAL TRADE 2013

New Developments in Mass Spectrometry No. 5

Print ISBN: 978-1-78262-428-8
PDF eISBN: 978-1-78262-673-2
EPUB eISBN: 978-1-78262-957-3
ISSN: 2044-253X

A catalogue record for this book is available from the British Library

Published by The Royal Society of Chemistry,
Thomas Graham House, Science Park, Milton Road,
Cambridge CB4 0WF, UK

Registered Charity Number 207890

For further information see our web site at www.rsc.org

Printed in the United Kingdom by CPI Group (UK) Ltd, Croydon, CR0 4YY, UK

Acknowledgements

I am indebted to all the academics who have taken time out from their busy schedules to contribute to this book – many thanks to you all. Thanks also to Simon Gaskell for inviting me to put this book together, and to the team at the Royal Society of Chemistry for being so supportive and professional throughout the commissioning and publication process. I am also grateful to Ryan Smith, who provided a valuable student's eye view of many of the chapters prior to final editing.

I would also like to take this opportunity to thank Dan Crowther and Ian Shadforth for getting me started in the fascinating field of proteome informatics, all those years ago.

Last but not least, thanks to Nieves for her tireless patience and support.

Conrad Bessant
London

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

Contents

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

Section II: Protein Quantitation

Section III: Open Source Software Environments for Proteome Informatics

CHAPTER 1

Introduction to Proteome Informatics

CONRAD BESSANT[a]

[a]School of Biological and Chemical Sciences, Queen Mary University of London, E1 4NS, UK
*E-mail: c.bessant@qmul.ac.uk

1.1 Introduction

In an era of biology dominated by genomics, and next generation sequencing (NGS) in particular, it is easy to forget that proteins are the real workhorses of biology. Among other tasks, proteins give organisms their structure, they transport molecules, and they take care of cell signalling. Proteins are even responsible for creating proteins when and where they are needed and disassembling them when they are no longer required. Monitoring proteins is therefore essential to understanding any biological system, and proteomics is the discipline tasked with achieving this.

Since the ground-breaking development of soft ionisation technologies by Masamichi Yamashita and John Fenn in 1984,[1] liquid chromatography coupled with tandem mass spectrometry (LC-MS/MS, introduced in the next section) has emerged as the most effective method for high throughput identification and quantification of proteins in complex biological mixtures.[2] Recent years have seen a succession of new and improved instruments bringing higher throughput, accuracy and sensitivity. Alongside these instrumental improvements, researchers have developed an extensive range

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

of protocols which optimally utilise the available instrumentation to answer a wide range of biological questions. Some protocols are concerned only with protein identification, whereas others seek to quantify the proteins as well. Depending on the particular biological study, a protocol may be selected because it provides the widest possible coverage of proteins present in a sample, whereas another protocol may be selected to target individual proteins of interest. Protocols have also been developed for specific applications, for example to study post-translational modification of proteins, *e.g.*,[3] to localise proteins to their particular subcellular location, *e.g.*,[4] and to study particular classes of protein, *e.g.*[5]

A common feature of all LC-MS/MS-based proteomics protocols is that they generate a large quantity of data. At the time of writing, a raw data file from a single LC-MS/MS run on a modern instrument is over a gigabyte (GB) in size, containing thousands of individual high resolution mass spectra. Because of their complexity, biological samples are often fractionated prior to analysis and ten individual LC-MS/MS runs per sample is not unusual, so a single sample can yield 10–20 GB of data. Given that most proteomics studies are intended to answer questions about protein dynamics, *e.g.* differences in protein expression between populations or at different time points, an experiment is likely to include many individual samples. Technical and biological replicates are always recommended, at least doubling the number of runs and volume of data collected. Hundreds of gigabytes of data per experiment is therefore not unusual.

Such data volumes are impossible to interpret without computational assistance. The volume of data per experiment is actually relatively modest compared to other fields, such as next generation sequencing or particle physics, but proteomics poses some very specific challenges due to the complexity of the samples involved, the many different proteins that exist, and the particularities of mass spectrometry. The path from spectral peaks to confident protein identification and quantitation is complex, and must be optimised according to the particular laboratory protocol used and the specific biological question being asked. As laboratory proteomics continues to evolve, so do the computational methods that go with it. It is a fast moving field, which has grown into a discipline in its own right. Proteome informatics is the term that we have given this discipline for this book, but many alternative terms are in use. The aim of the book is to provide a snapshot of current thinking in the field, and to impart the fundamental knowledge needed to use, assess and develop the proteomics algorithms and software that are now essential in biological research.

Proteomics is a truly interdisciplinary endeavour. Biological knowledge is required to appreciate the motivations of proteomics, understand the research questions being asked, and interpret results. Analytical science expertise is essential – despite instrument vendors' best efforts at making instruments reliable and easy to use, highly skilled analysts are needed to operate such instruments and develop the protocols needed for a given study. At least a basic knowledge of chemistry, biochemistry and physics is

required to understand the series of processes that happen between a sample being delivered to a proteomics lab and data being produced. Finally, specialised computational expertise is needed to handle the acquired data, and it is this expertise that this book seeks to impart. The computational skills cover a wide range of specialities, ranging from algorithm design to identify peptides (Chapters 2 and 3), statistics to score and validate identifications (Chapter 4), infer the presence of proteins (Chapter 5) and perform downstream analysis (Chapter 14), through signal processing to quantify proteins from acquired mass spectrometry peaks (Chapters 7 and 8) and software skills needed to devise and utilise data standards (Chapter 11) and analysis frameworks (Chapters 12–14), and integrate proteomics data with NGS data (Chapters 15 and 16).

1.2 Principles of LC-MS/MS Proteomics

The wide range of disciplines that overlap with proteome informatics draws in a great diversity of people including biologists, biochemists, computer scientists, physicists, statisticians, mathematicians and analytical chemists. This poses a challenge when writing a book on the subject as a core set of prior knowledge cannot be assumed. To mitigate this, this section provides a brief overview of the main concepts underlying proteomics, from a data-centric perspective, together with citations to sources of further detail.

1.2.1 Protein Fundamentals

A protein is a relatively large (median molecular weight around 40 000 Daltons) molecule that has evolved to perform a specific role within a biological organism. The role of a protein is determined by its chemical composition and 3D structure. In 1949 Frederick Sanger provided conclusive proof[6] that proteins consist of a polymer chain of amino acids (The 20 amino acids that occur naturally in proteins are listed in Table 1.1). Proteins are synthesised within cells by assembling amino acids in a sequence dictated by a gene – a specific region of DNA within the organism's genome. As it is produced, physical interactions between the amino acids causes the string of amino acids to fold up into the 3D structure of the finished protein. Because the folding process is deterministic (albeit difficult to model) it is convenient to assume a one-to-one relationship between amino acid sequence and structure so a protein is often represented by the sequence of letters corresponding to its amino acid sequence. These letters are said to represent residues, rather than amino acids, as two hydrogens and an oxygen are lost from an amino acid when it is incorporated into a protein so the letters cannot strictly be said to represent amino acid molecules.

Organisms typically have thousands of genes, *e.g.* around 20 000 in humans. The human body is therefore capable of producing over 20 000 distinct proteins, which illustrates one of the major challenges for proteomics – the large number of distinct proteins that may be present in a given sample

(referred to as the so-called search space when seeking to identify proteins). The situation is further complicated by alternative splicing,[7] where different combinations of segments of a gene are used to create different versions of the protein sequence, called protein isoforms. Because of alternative splicing each human gene can produce on average around five distinct protein isoforms per gene. So, our search space expands to ~100 000 distinct proteins. If we are working with samples from a population of different individuals, the search space expands still further as some individual genome variations will translate into variations in protein sequence, some of which have transformative effects on protein structure and function.

However, the situation is yet more complex because, after synthesis, a protein may be modified by covalent addition (and possibly later removal) of a chemical entity at one or more amino acids within the protein sequence. Phosphorylation is a very common example, known to be important in regulating the activity of many proteins. Phosphorylation involves the addition of a phosphoryl group, typically (but not exclusively) to an S, T or Y. Such post-translational modifications (PTMs) change the mass of proteins, and often their function. Because each protein contains many sites at

Table 1.1 The 20 amino acids that are the building blocks of peptides and proteins. Throughout this book we generally refer to amino acids by their single letter code. Isotopes and the concept of monoisotopic mass are explained in Chapter 7. Residue masses are ~18.01 Da lower than the equivalent amino acid mass because one oxygen and two hydrogens are lost from an amino acid when it is incorporated into a protein. Post-translational modifications add further chemical diversity to the amino acids listed here, and increase their mass, as explained in Chapter 6.

Amino acid	Abbreviation	Single letter code	Monoisotopic residue mass
Alanine	Ala	A	71.037114
Cysteine	Cys	C	103.009185
Aspartic acid	Asp	D	115.026943
Glutamic acid	Glu	E	129.042593
Phenylalanine	Phe	F	147.068414
Glycine	Gly	G	57.021464
Histidine	His	H	137.058912
Isoleucine	Ile	I	113.084064
Lysine	Lys	K	128.094963
Leucine	Leu	L	113.084064
Methionine	Met	M	131.040485
Asparagine	Asn	N	114.042927
Proline	Pro	P	97.052764
Glutamine	Gln	Q	128.058578
Arginine	Arg	R	156.101111
Serine	Ser	S	87.032028
Threonine	Thr	T	101.047679
Valine	Val	V	99.068414
Tryptophan	Trp	W	186.079313
Tyrosine	Tyr	Y	163.06333

which PTMs may occur, there is a large number of distinct combinations of PTMs that may be seen on a given protein. This increases the search space massively, and it is not an exaggeration to state that the number of distinct proteins that could be produced by a human cell exceeds one million. We will never find a million proteins in a single cell – a few thousand is more typical – but the fact that these few thousand must be identified from a potential list of over a million represents one of the biggest challenges in proteomics.

1.2.2 Shotgun Proteomics

The obvious way to identify proteins from a complex sample would be to separate them from each other, then analyse each protein one by one to determine what it is. Although conceptually simple, practical challenges of this so-called top-down method[8] have led the majority of labs to adopt the alternative bottom-up methodology, often called shotgun proteomics. This

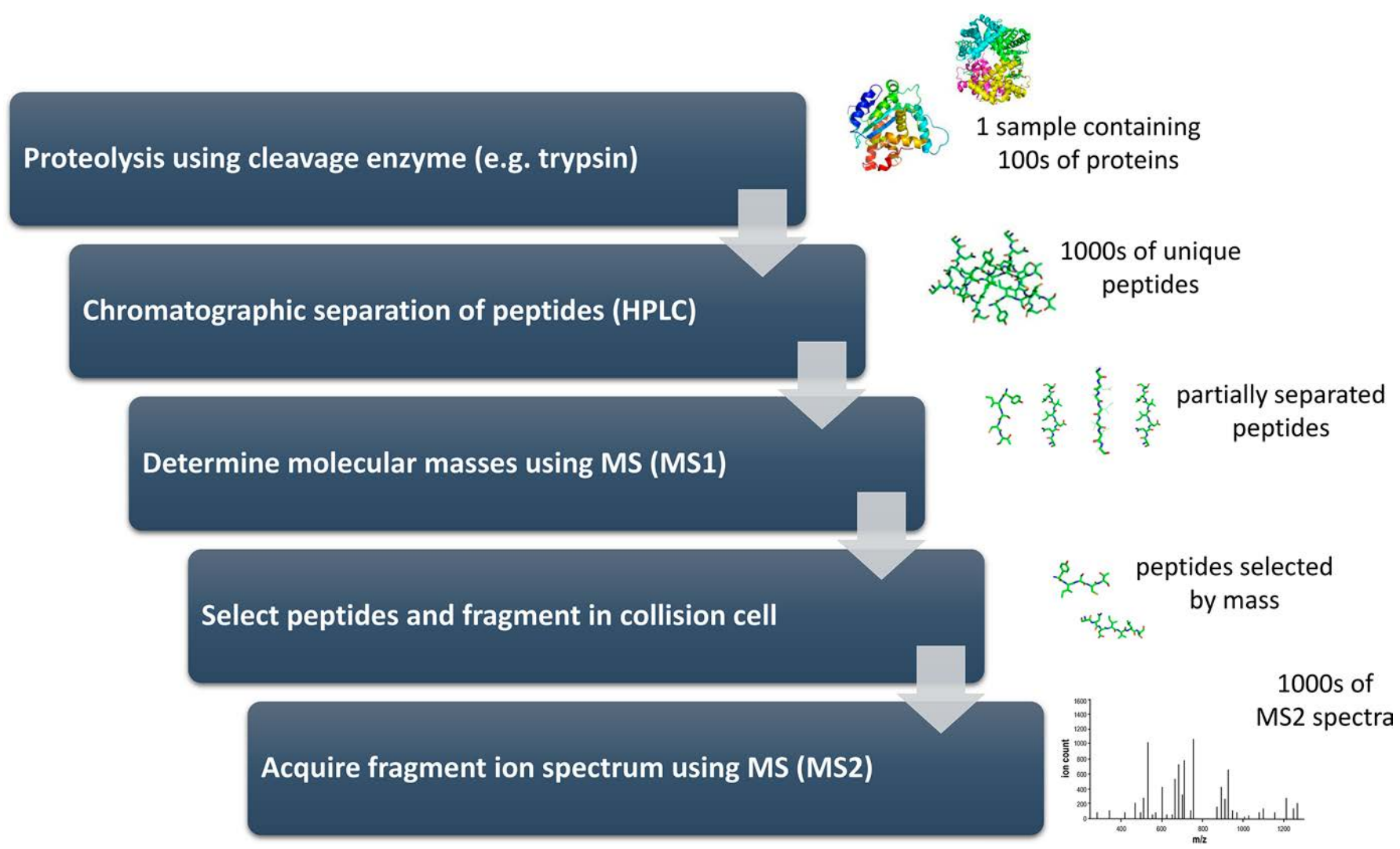

Figure 1.1 Schematic overview of a typical shotgun proteomics workflow. Analysis starts with a biological sample containing many hundreds or thousands of proteins. These proteins are digested into peptides by adding a proteolytic enzyme to the sample. Peptides are then partially separated using HPLC, prior to a first stage of MS (MS1). Peptides from this first stage of MS are selected for fragmentation, leading to the generation of fragmentation spectra in a second stage of MS. This is the starting point for computational analysis – fragmentation spectra can be used to infer which peptides are in the sample, and peak areas (typically from MS1, depending on the protocol) can be used to infer their abundance. Often a sample will be separated into several (*e.g.* 10) fractions prior to analysis to reduce complexity – each fraction is then analysed separately and results combined at the end.

book therefore deals almost exclusively with the analysis of data acquired using this methodology, which is shown schematically in Figure 1.1.

In shotgun proteomics, proteins are broken down into peptides – amino acid chains that are much shorter than the average protein. These peptides are then separated, identified and used to infer which proteins were in the sample. The cleavage of proteins to peptides is achieved using a proteolytic enzyme which is known to cleave the protein into peptides at specific points. Trypsin, a popular choice for this task, generally cuts proteins after K and R, unless these residues are followed by P. The majority of the peptides produced by trypsin have a length of between 4–26 amino acids, equivalent to a mass range of approximately 450–3000 Da, which is well suited to analysis by mass spectrometry. Given the sequence of a protein, it is computationally trivial to determine the set of peptides that will be produced by tryptic digestion. However, digestion is not always 100% efficient so any data analysis must also consider longer peptides that result from one or more missed cleavage sites.

1.2.3 Separation of Peptides by Chromatography

Adding an enzyme such as trypsin to a complex mixture of proteins results in an even more complex mixture of peptides. The next step in shotgun proteomics is therefore to separate these peptides. To achieve high throughput this is typically performed using high performance liquid chromatography (HPLC). Explanations of HPLC can be found in analytical chemistry textbooks, *e.g.*,[9] but in simple terms it works by dissolving the sample in a liquid, known as the mobile phase, and passing this under pressure through a column packed with a solid material called the solid phase. The solid phase is specifically selected such that it interacts with, and therefore retards, some compounds more than others based on their physical properties. This phenomenon is used to separate different compounds as they are retained in the column for different amounts of time (their individual retention time, RT) and therefore emerge from the column (elute) separately. In shotgun proteomics, the solid phase is usually chosen to separate peptides based on their hydrophobicity. Protocols vary, but a typical proteomics chromatography run takes 30–240 minutes depending on expected sample complexity and, after sample preparation, is the primary pace factor in most proteomic analyses.

While HPLC provides some form of peptide separation, the complexity of biological samples is such that many peptides co-elute, so further separation is needed. This is done in the subsequent mass spectrometry step, which also leads to peptide identification.

1.2.4 Mass Spectrometry

In the very simplest terms, mass spectrometry (MS) is a method for sorting molecules according to their mass. In shotgun proteomics, MS is used to separate co-eluting peptides after HPLC and to determine their mass. A detailed

explanation of mass spectrometry is beyond the scope of this chapter. The basic principles can be found in analytical chemistry textbooks, *e.g.*,[10] and an in-depth introduction to peptide MS can be found in ref. 11, but a key detail is that a molecule must be carrying a charge if it is to be detected. Peptides in the liquid phase must be ionised and transferred to the gas phase prior to entering the mass spectrometer. The so-called soft ionisation methods of electrospray ionisation (ESI)[1,12] and matrix assisted laser desorption–ionisation (MALDI)[13,14] are popular for this because they bestow charge on peptides without fragmenting them. In these methods a positive charge is endowed by transferring one or more protons to the peptide, a process called protonation. If a single proton is added, the peptides become a singly charged (1^+) ion but higher charge states are also possible (typically 2^+ or 3^+) as more than one proton may be added. The mass of a peptide correspondingly increases by one proton (~1.007 Da) for each charge state. Not every copy of every peptide gets ionised (this depends on the ionisation efficiency of the instrument) and it is worth noting that many peptides are very difficult to ionise, making them essentially undetectable in MS – this has a significant impact on how proteomics data are analysed as we will see in later chapters.

The charge state is denoted by z (*e.g.* $z = 2$ for a doubly charged ion) and the mass of a peptide by m. Mass spectrometers measure the mass to charge ratio of ions, so always report m/z, from which mass can be calculated if z can be determined. In a typical shotgun proteomics analysis, the mass spectrometer is programmed to perform a survey scan – a sweep across its whole m/z range – at regular intervals as peptides elute from the chromatography column. This results in a mass spectrum consisting of a series of peaks representing peptides whose horizontal position is indicative of their m/z (There are invariably additional peaks due to contaminants or other noise.). This set of peaks is often referred to as an MS1 spectrum, and thousands are usually acquired during one HPLC run, each at a specific retention time.

The current generation of mass spectrometers, such as those based on orbitrap technology[15] can provide a mass accuracy exceeding 1 ppm so, for example, the mass of a singly charged peptide with m/z of 400 can be determined to an accuracy of 0.0004 Da. Determining the mass of a peptide with this accuracy provides a useful indication of the composition of a peptide, but does not reveal its amino acid sequence because many different sequences can share the exact same mass.

To discover the sequence of a peptide we must break it apart and analyse the fragments generated. Typically, a data dependent acquisition (DDA) approach is used, where ions are selected in real time at each retention time by considering the MS1 spectrum, with the most abundant peptides (inferred from peak height) being passed to a collision chamber for fragmentation. Peptides are passed one at a time, providing a final step of separation, based on mass. A second stage of mass spectrometry is performed to produce a spectrum of the fragment ions (also called product ions) emerging from the peptide fragmentation – this is often called an MS2 spectrum (or MS/MS spectrum). Numerous methods have been developed to fragment peptides,

including electron transfer dissociation (ETD,[16]) and collision induced dissociation (CID,[17]). The crucial feature of these methods is that they predominantly break the peptide along its backbone, rather than at random bonds. This phenomenon, shown graphically in Figure 1.2, produces fragment ions whose masses can be used to determine the peptide's sequence.

The DDA approach has two notable limitations: it is biased towards peptides of high abundance, and there is no guarantee that a given peptide will be selected in different runs, making it difficult to combine data from multiple samples into a single dataset. Despite this, DDA remains popular at the time of writing, but two alternative methods are gaining ground. Selected reaction monitoring (SRM) aims to overcome DDA's limitations by *a priori* selection of peptides to monitor (see Chapter 9) at the expense of breadth of coverage, whereas data independent acquisition (DIA) simply aims to fragment every peptide (see Chapter 10).

1.3 Identification of Peptides and Proteins

Determining the peptide sequence represented by an acquired MS2 spectrum is the first major computational challenge dealt with in this book. The purest and least biased method is arguably *de novo* sequencing (Chapter 2) in which the sequence is determined purely from the mass difference

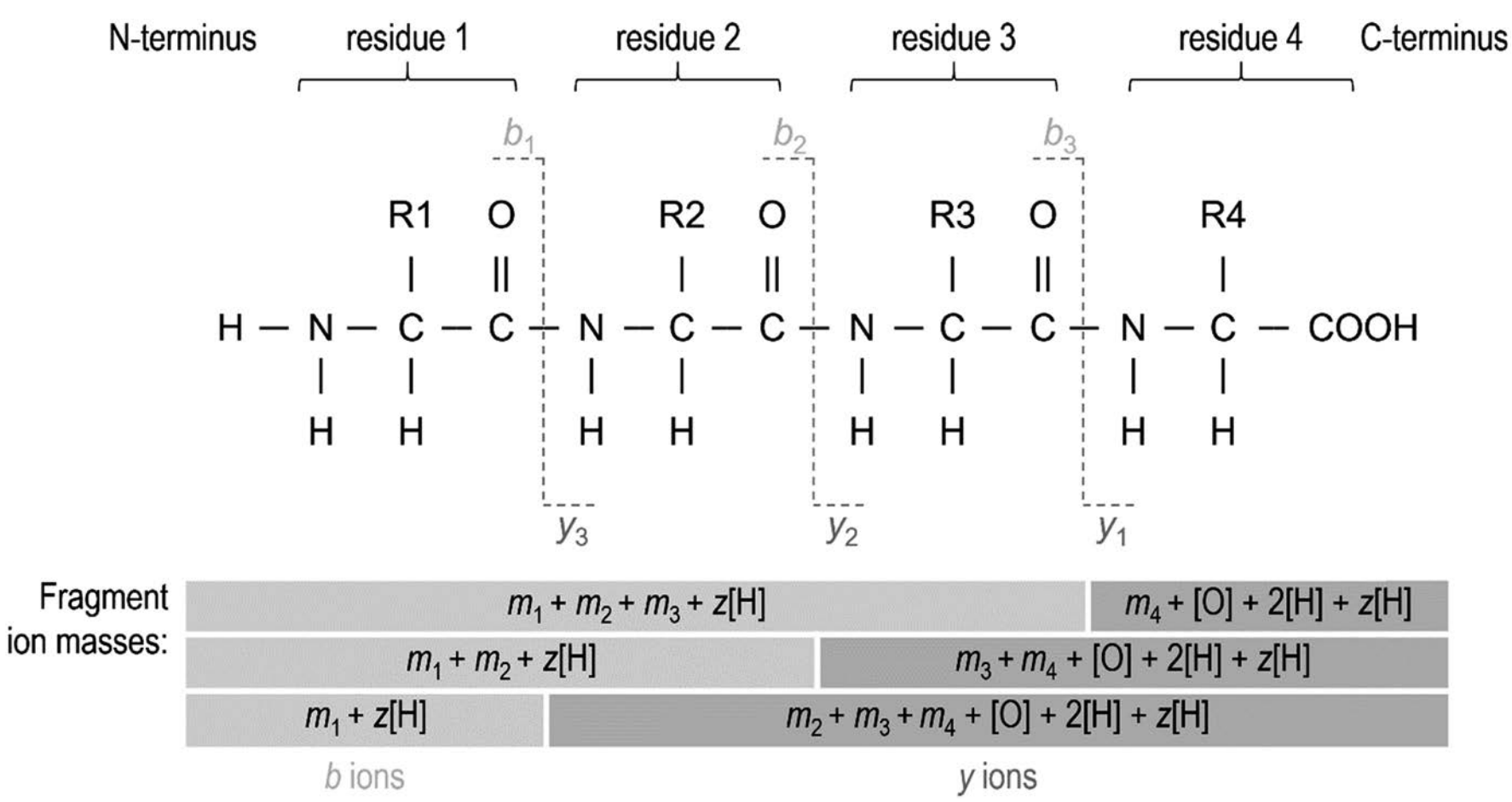

Figure 1.2 Generic four AA peptide, showing its chemical structure with vertical dotted lines indicating typical CID fragmentation points and, below, corresponding calculation of b- and y-ion masses. Peptides used to infer protein information are typically longer than this (~8–26 AAs), but the concept is the same. In the mass calculations, m_n represents the mass of residue n, [H] and [O] the mass of hydrogen and oxygen respectively, and z is the fragment's charge state. Differences between the number of hydrogen atoms shown in the figure and the number included in the calculation are due to the fragmentation process.[11]

between adjacent fragment ions. In practice, identifying peptides with the help of information from protein sequence databases such as UniProt[18] is generally considered more reliable and an array of competing algorithms have emerged for performing this task (Chapter 3). These algorithms require access to a representative proteome, which may not be available for non-model organisms and some other complex samples. In these cases, a sample specific database may be created from RNA-seq transcriptomics collected from the same sample (Chapter 16). Spectral library searching (also covered in Chapter 3) offers a further alternative, if a suitable library of peptide MS2 spectra exists for the sample under study.

None of the available algorithms gives a totally definitive peptide match for a given spectrum, but provide scores indicating the likelihood that the match is correct. Historically, each algorithm provided its own proprietary score but great strides have been made in recent years in developing statistical methods for objectively scoring and validating peptide spectrum matches independently of the identification algorithm used (see Chapter 4). Confidently identified peptides can then be used to infer which proteins are present in the sample. There are a number of challenges here, including the aforementioned problem of undetectable peptides, and the fact that many peptides map to multiple proteins. These issues, and current solutions to them, are covered in Chapter 5.

As mentioned earlier, the phenomenon of post-translational modification complicates protein identification considerably by massively increasing the search space. Chapter 6 discusses this issue and summarises current thinking on how best to deal with PTM identification and localisation.

1.4 Protein Quantitation

In most biological studies it is important to augment protein identifications with information about the abundance of those proteins. Laboratory protocols for quantitative proteomics are numerous and diverse, indeed there is a whole book in this series dedicated to the topic.[19] Each protocol requires different data processing, leading to a vast range of quantitative proteomics algorithms and workflows. For the purposes of this book we have made a distinction between methods that extract the quantitative information from MS1 spectra (covered in Chapter 7) and those that use MS2 spectra (Chapter 8). Despite the diversity of quantitation methods, the vast majority infer protein abundance from peptide-level features so there is much in common between the algorithms used.

1.5 Applications and Downstream Analysis

As we have seen, identifying and quantifying proteins is a complex process but is one that has matured enough to be widely applied in biological research. Most researchers now expect that a list of proteins and their abundances can

be extracted for a given biological sample. Of course, any serious research project is unlikely to conclude with a simple list of identified proteins and their abundance. Further analysis will be needed to interpret the results obtained to answer the biological question posed, from biomarker discovery through to systems biology studies.

Downstream analysis is not generally covered in this book, partly because there are too many potential workflows to cover, but mainly because many of the methods used are not specific to proteomics. For example, statistical approaches used for determining which proteins are differentially expressed between two populations are often similar to those used for finding differentially expressed genes – typically a significance test followed by some multiple testing correction.[20] Similarly, the pathway analysis performed with proteomics data is not dissimilar to that carried out with gene expression data.[21]

However, caution is needed when applying transcriptomics methods to proteomics data, as there are many subtle differences. Incomplete sequence coverage due to undetectable peptides is one important difference between proteomics and RNA-seq, and confidence of protein identification and quantification is also something that should be considered. For example, proteins identified based on a single peptide observation (so called "one hit wonders") should be avoided in any quantitative analysis as abundance accuracy is likely to be poor (see Chapter 5). PTMs are another important consideration, as they have the potential to affect a protein's role in pathway analysis. One area of downstream analysis that we have chosen to cover is genome annotation using proteomics data (proteogenomics, Chapter 15), as this is an excellent and very specific example of proteomics being combined with genomics, and sometimes also transcriptomics, to better understand an organism.

1.6 Proteomics Software

As the proteomics community has grown, so has the available software for handling proteomics data. It is not possible to cover all available software within a book of this size, and nor is it sensible as the situation is in constant flux, with new software being released, existing software updated and old software having support withdrawn (but rarely disappearing completely). For this reason, most of the chapters in this book avoid focussing on specific software packages, instead discussing more generic concepts and algorithms that are implemented across multiple packages. However, for the benefit of readers new to the field, it is worth briefly surveying the current proteomics software landscape.

At the time of writing, proteomics is dominated by a relatively small number of generally monolithic Windows-based desktop software packages. These include commercial offerings such as Proteome Discoverer from Thermo and Progenesis QI from Waters, and freely available software

such a MaxQuant[22] and Skyline.[22,23] Some of these packages support the whole data analysis workflow, from raw data through protein identification and quantitation and on to statistical analysis of the results. Reliance on Windows is unusual in the scientific research community, but perhaps explained by the fact that most mass spectrometer control software is Windows-based and some raw MS data formats can only be accessed using Windows-based software libraries.[24] From a bioinformatics perspective there are clear disadvantages of the *status quo*, including a lack of flexibility, lack of transparency due to closed source code in some cases, and doubts about whether desktop-based Windows software can scale to cope with growing datasets. However, bench scientists appreciate the quality and usability of these packages and they are likely to remain popular for the foreseeable future.

The aforementioned packages are complemented by a vast array of other software tools, most of which have been developed by academic groups and are freely available. Typically, these packages are reference implementations of a published algorithm designed to perform a specific task (*e.g.* peptide identification), or support a particular protocol (*e.g.* quantitation with specific labels). Assembling such tools into a pipeline can be challenging, but can be the best way of implementing a specialised workflow. To ease the process of integrating disparate tools, developers are increasingly making their software available within common open frameworks such as OpenMS (Chapter 12), Galaxy (Chapter 13), BioConductor (Chapter 14) and as a set of PSI-centric libraries (see Chapter 11). These frameworks are mainly differentiated by their user interfaces and the programming languages that underpin them (C++ for OpenMS, R for BioConductor and Java for the PSI libraries). Galaxy is largely language agnostic, although much of its internals are written in Python.

1.6.1 Proteomics Data Standards and Databases

As in other data rich fields of biological research, the proteomics community has established databases to share data from proteomics experiments, and to enable interoperability between different pieces of software. This has proven difficult due to the wide range of proteomics protocols in use and different opinions about the most appropriate way to represent the results of a proteomics experiment, *e.g.* should raw data be stored or is a list of identified proteins sufficient? Questions like these have been tackled by the Human Proteome Organisation Proteomics Standards Initiative (HUPO–PSI), who have drawn up guidelines for reporting minimum information about a proteomics experiment (MIAPE) and data formats that capture the necessary information in a consistent way (see Chapter 11).

Progress in community standards for reporting results has paved the way for public repositories of proteomics databases. Arguably PRIDE[25] is foremost among these as it is long established and at the time of writing is the

only proteomics database backed by a dedicated bioinformatics institution (the European Bioinformatics Institute). Several leading journals request, or require, deposition of data to PRIDE to support any paper that involves proteomics. Other well established databases include PeptideAtlas,[26] GPMDB[27] and PASSEL[28] (specifically for SRM data) but there are many more. A recent review article[29] provides an extensive overview of the current state of proteomic repositories.

1.7 Conclusions

At the time of writing, much crucial groundwork in proteome informatics is already in place, but many interesting challenges remain and new challenges continue to appear as new laboratory protocols and biological applications emerge and evolve. Proteome informatics is therefore an active area of research, and it is now easier to get into thanks to an abundance of excellent freely available software tools and large collections of high quality data in public repositories.

Acknowledgements

The author would like to acknowledge funding from BBSRC grants BB/M020118/1 and BB/M006174/1. Many thanks to Pedro Cutillas for reviewing this chapter.

References

1. M. Yamashita and J. B. Fenn, Electrospray ion source. Another variation on the free-jet theme, *J. Phys. Chem.*, 1984, **88**(20), 4451–4459.
2. B. Domon and R. Aebersold, Mass spectrometry and protein analysis, *Science*, 2006, **312**(5771), 212–217.
3. L. Beltran and P. R. Cutillas, Advances in phosphopeptide enrichment techniques for phosphoproteomics, *Amino Acids*, 2012, **43**(3), 1009–1024.
4. P. Sadowski, *et al.*, Quantitative proteomic approach to study subcellular localization of membrane proteins, *Nat. Protoc.*, 2006, **1**(4), 1778–1789.
5. S. M. Moore, S. M. Hess and J. W. Jorgenson, Extraction, Enrichment, Solubilization, and Digestion Techniques for Membrane Proteomics, *J. Proteome Res.*, 2016, **15**(4), 1243–1252.
6. F. Sanger, The terminal peptides of insulin, *Biochem. J.*, 1949, **45**(5), 563–574.
7. D. L. Black, Mechanisms of alternative pre-messenger RNA splicing, *Annu. Rev. Biochem.*, 2003, **72**, 291–336.
8. Z. R. Gregorich and Y. Ge, Top-down proteomics in health and disease: challenges and opportunities, *Proteomics*, 2014, **14**(10), 1195–1210.

9. G. D. Christian, P. K. Dasgupta and K. A. Schug, Liquid Chromatography and Electrophoresis, *Analytical Chemistry*, 7th edn, 2013, pp. 649–701.
10. G. D. Christian, P. K. Dasgupta and K. A. Schug, Mass Spectrometry, *Analytical Chemistry*, 7th edn, 2013, pp. 735–768.
11. S. D. Maleknia, *et al.*, Mass Spectrometry of Amino Acids and Proteins, in *Amino Acids, Peptides and Proteins in Organic Chemistry: Analysis and Function of Amino Acids and Peptides*, ed. A. B. Hughes, Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 2011, pp. 1–50.
12. S. J. Gaskell, Electrospray: Principles and Practice, *J. Mass Spectrom.*, 2016, **32**(7), 677–688.
13. M. Karas, D. Bachmann, U. Bahr and F. Hillenkamp, Matrix-assisted ultraviolet laser desorption of non-volatile compounds, *Int. J. Mass Spectrom. Ion Processes*, 1987, **78**, 53–68.
14. M. Karas and U. Bahr, Laser desorption ionization mass spectrometry of large biomolecules, *TrAC, Trends in Analytical Chemistry*, 2002, **9**(10), 321–325.
15. Q. Hu, *et al.*, The Orbitrap: a new mass spectrometer, *J. Mass Spectrom.*, 2005, **40**(4), 430–443.
16. J. E. P. Syka, *et al.*, Peptide and protein sequence analysis by electron transfer dissociation mass spectrometry, *Proc. Natl. Acad. Sci. U. S. A.*, 2004, 9528–9533.
17. J. M. Wells and S. A. McLuckey, Collision-induced dissociation (CID) of peptides and proteins, *Methods Enzymol.*, 2005, **402**, 148–185.
18. The UniProt Consortium, UniProt: a hub for protein information, *Nucleic Acids Research*, 2015, **43**, D204–D212.
19. C. E. Eyers and S. Gaskell, in *Quantitative Proteomics (New Developments in Mass Spectrometry)*, ed. S. Gaskell, Royal Society of Chemistry, 2014, p. 390.
20. W. S. Noble, How does multiple testing correction work?, *Nat. Biotechnol.*, 2009, **27**(12), 1135–1137.
21. M. A. Garcia-Campos, J. Espinal-Enriquez and E. Hernandez-Lemus, Pathway Analysis: State of the Art, *Front. Physiol.*, 2015, **6**, 383.
22. J. Cox and M. Mann, MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification, *Nat. Biotechnol.*, 2008, **26**(12), 1367–1372.
23. B. MacLean, *et al.*, Skyline: an open source document editor for creating and analyzing targeted proteomics experiments, *Bioinformatics,* 2010, **26**(7), 966–968.
24. J. D. Holman, D. L. Tabb and P. Mallick, Employing ProteoWizard to Convert Raw Mass Spectrometry Data, *Curr. Protoc. Bioinf.*, 2014, **46**, 13.24.1–9.
25. J. A. Vizcaino, *et al.*, The PRoteomics IDEntifications (PRIDE) database and associated tools: status in 2013, *Nucleic Acids Res.*, 2013, **41**(Database issue), D1063–D1069.
26. T. Farrah, *et al.*, State of the human proteome in 2013 as viewed through PeptideAtlas: comparing the kidney, urine, and plasma proteomes for

the biology- and disease-driven Human Proteome Project, *J. Proteome Res.*, 2014, **13**(1), 60–75.
27. R. Craig, J. P. Cortens and R. C. Beavis, Open source system for analyzing, validating, and storing protein identification data, *J. Proteome Res.*, 2004, **3**(6), 1234–1242.
28. T. Farrah, *et al.*, PASSEL: the PeptideAtlas SRMexperiment library, *Proteomics*, 2012, **12**(8), 1170–1175.
29. Y. Perez-Riverol, *et al.*, Making proteomics data accessible and reusable: current state of proteomics databases and repositories, *Proteomics*, 2015, **15**(5–6), 930–949.

Section I

Protein Identification

CHAPTER 2

De novo *Peptide Sequencing*

BIN MA[a]

[a]School of Computer Science, University of Waterloo, Canada
*E-mail: binma@uwaterloo.ca

2.1 Introduction

De novo peptide sequencing is one of the few computational methods used to sequence a peptide purely from its MS/MS spectrum. The method does not require a given sequence database or spectrum library. Instead, it derives the peptide sequence solely based on the spectrum. This fact makes *de novo* sequencing more useful than other methods in certain applications, but meanwhile makes it a much harder computational problem.

In earlier days of mass spectrometry based proteomics, a mass spectrometry experiment usually produced only a small number of MS/MS spectra. These spectra were often manually *de novo* sequenced to identify the peptides. Automated *de novo* sequencing software tools were also developed, but these earlier tools were not widely adopted because of a number of reasons. First, the accuracy of those tools were not nearly as good as manual *de novo* sequencing. Secondly, after the publication of the SEQUEST program in 1995,[1] the popularization of the database search method provides a viable alternative for peptide and protein identification with MS/MS datasets.

However, *de novo* sequencing has continuously attracted attention in the algorithm research community. Dozens of software tools have been developed and gained different popularities at different periods. Some more notable tools include Lutefisk,[2] PEAKS,[3] and PepNovo.[4] A more complete review

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

of these tools can be found in the computer software section of this chapter. Due to the research and development efforts from both the public and private sectors, the accuracy of automated *de novo* sequencing has continuously improved. Some of the latest developments (*e.g.* the Novor software[5]) started to employ large scale machine learning to learn sophisticated peptide fragmentation rules, and use these rules to assist the *de novo* sequencing algorithm. This has allowed the algorithm to make use of many rules that a human would use in manual *de novo* sequencing, resulting in significantly improved accuracy.

At the same time, both the mass spectrometer's throughput and mass accuracy have improved dramatically in the past two decades. While the high throughput prohibits manual *de novo* sequencing practice, the improved mass accuracy contributes to the accuracy of automated *de novo* sequencing. Consequently, automated *de novo* sequencing becomes both practical and indispensable for scientists to identify novel peptides from today's large mass spectrometry data.

The applications of *de novo* sequencing are no longer limited to the sequencing of novel peptides. First, some recent studies (*e.g.* ref. 6) showed that in a typical proteomics mass spectrometry dataset, a large number of high quality spectra are not assigned by the database search method. These spectra are potentially from mutated, modified, or novel peptides. *De novo* sequencing would be a perfect method to identify confident sequence tags from these spectra. Secondly, even if a spectrum can be assigned by a database peptide, it has been shown that one can use *de novo* sequencing to confirm the database peptide.[7] This leads to a higher identification rate at the same or reduced false discovery rate (FDR). Thirdly, in the identification of modified or mutated peptides, the confident *de novo* sequencing tags can be used to filter the sequence database for potential matches. The filtration significantly speeds up the database search in several software tools.[7–9]

Historically *de novo* sequencing and database search are regarded as two independent methods used in different applications. These recent developments suggest that both methods should be used in parallel to analyze proteomics mass spectrometry data, even if a protein sequence database is available.[5,10,11]

2.2 Manual *De novo* Sequencing

Although today's *de novo* sequencing is mostly carried out with automated software, it is possible, and sometimes necessary, for an experienced person to manually *de novo* sequence a peptide from its MS/MS spectrum. The basic principle of manual *de novo* sequencing is to use the highly abundant y-ion ladders to determine the amino acid sequence. In this section, the manual *de novo* sequencing process is illustrated with an example. Examining the manual process will help understand the computer algorithms used for automated *de novo* sequencing, as well as the challenges of developing such algorithms.

(a)

b_1	L	EDFLER	y_6
b_2	LE	DFLER	y_5
b_3	LED	FLER	y_4
b_4	LEDF	LER	y_3
b_5	LEDFL	ER	y_2
b_6	LEDFLE	R	y_1

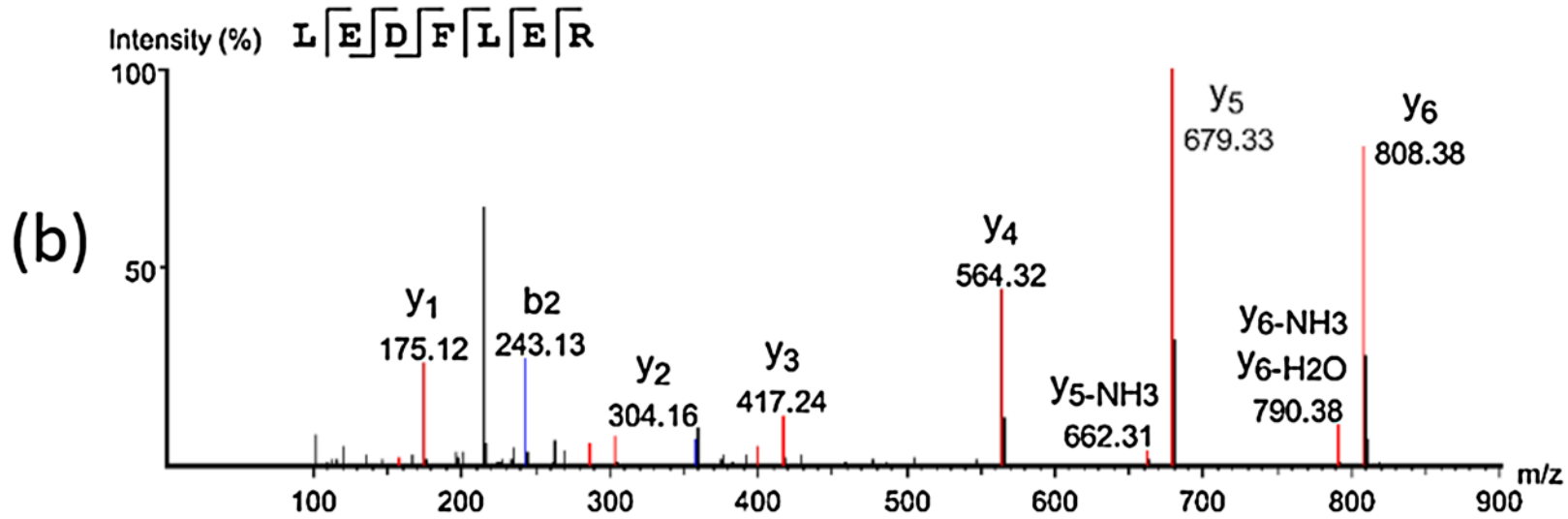

Figure 2.1 (a) The fragmentation at the peptide backbone produces b and y-ions in a CID spectrum. (b) The annotated MS/MS spectrum for the peptide.

Figure 2.1(a) illustrates the fragmentation at different sites of the peptide LEDFLER. Each possible fragmentation corresponds to a pair of complementary b and y-ions. The subscript of each ion type indicates the number of residues in the fragment. The mass of a singly charged y-ion is calculated as the total residue mass plus a constant mass shift (≈19.02 Da). For b-ions, this mass shift is approximately 1 Da. Therefore, it is fairly straightforward to compute the b and y-ion *m*/*z* values and annotate the peaks in the spectrum by matching the mass. Figure 2.1(b) shows the annotated CID (Collision Induced Dissociation) MS/MS spectrum for the peptide LEDFLER. Other ion types (such as the y-H_2O ion) are also possible. But the b and y-ions (and particularly the y-ions) are the most abundant ions in a CID spectrum.

If the spectrum is annotated as in Figure 2.1(b), the sequencing is fairly straightforward. This is because the mass difference between two adjacent singly charged y-ions is equal to the mass of a single residue. Thus, the sequence can be determined by examining the mass difference of every adjacent pair of y-ions in the spectrum. However, the annotation is unavailable before the peptide is sequenced. For an unannotated spectrum, the following procedure can be used for manual *de novo* sequencing:

1. Observe a clear ladder of highly abundant peaks.
2. Choose a highly abundant peak in the ladder as an "anchor", and assume it is a y-ion. For example, the peak at 679.33 in Figure 2.1(b) is chosen as the anchor.
3. Choose the next peak to the left (or right) in the ladder. Check if the mass difference is equal to an amino acid residue's mass. For example,

if the peak at 564.32 is chosen, the mass difference is equal to 679.33 − 564.32 = 115.01. This is approximately equal to the residue mass of aspartic acid (D). So we have figured out one residue on the sequence. The peak at 564.32 is then chosen as the new anchor.

4. Repeat step 3 in both directions to figure out the remaining amino acids of the peptide sequence.
5. The N-terminal residue's mass is derived by the y_1-ion mass minus 19.02 Da.
6. The C-terminal residue's mass is derived by the precursor mass minus the mass of the largest y-ion.

Keep in mind that the spectrum in Figure 2.1 is among the highest quality in a high-throughput proteomics experiment. When the quality is not as good, this procedure is usually carried out in a trial-and-error fashion, due to the following complications.

First, there may be more than one choice for the next y-ion peak in step 3. Sometimes the most abundant peak may not be the right choice. One may need to try each of the choices to check which one provides the longest extension. In certain situations, a confident disambiguation may not be possible.

Secondly, the y-ion ladder may be incomplete and the extension in steps 3 and 4 may stop in the middle of the spectrum. This leads to only a partial sequence ladder. Sometimes the problem caused by a y-ion missing can be rescued by checking the complementing b-ion, as well as other neutral loss ions (*e.g.* y-H_2O and y-NH_3). But these are not always possible. Additionally, considering those weaker ion types will meanwhile increase the chance of a false-positive peak assignment.

Another way to deal with the missing ion problem is to faithfully record the mass gap in the output. For example, a sequence tag LDV[168.09]ER indicates that there is not enough evidence to confidently tell the sequence for the mass gap 168.09. It is not hard to check that 168.09 is equal to the total mass of residues A and P. But since the fragment ion is missing, one cannot confidently tell whether it is AP or PA for that mass gap.

An experienced human may be able to use the domain knowledge about peptide fragmentation to deal with the missing ion problem. For example, it has been reported that the amino acid P causes an enhanced fragmentation at its N-terminal side and a reduced fragmentation at its C-terminal side.[12] Therefore, not seeing a fragment ion within the mass gap 168.09 suggests that the dipeptide is more likely to be PA instead of AP.

2.3 Computer Algorithms

2.3.1 Search Tree Pruning

Algorithms in earlier days (*e.g.* ref. 2, 13–15) for *de novo* sequencing tried to emulate the manual *de novo* sequencing procedure. When there are ambiguities in a peak assignment, the algorithm has to search for all possibilities.

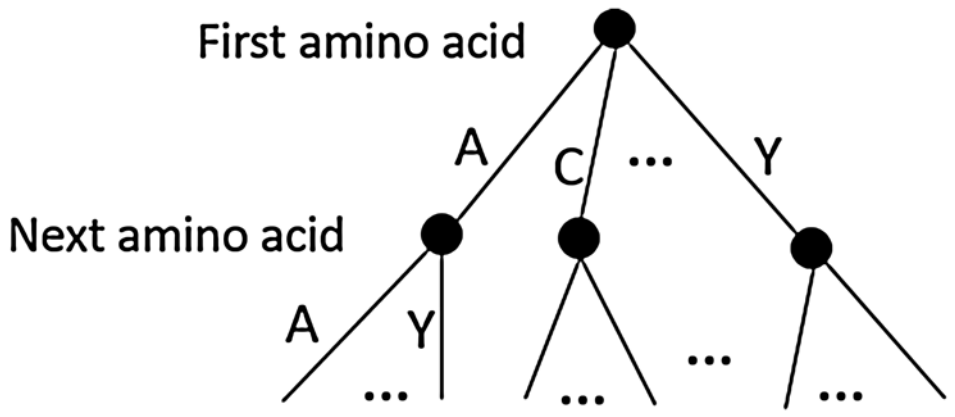

Figure 2.2 An illustration of the search tree to exhaustively search for the *de novo* sequence candidates.

This creates a branching node in the search tree (Figure 2.2). Following each possibility, the search tree may need to branch repeatedly to accommodate new ambiguities. The search tree size grows exponentially with respect to the number of branches on each search path.

To speed up, such search algorithms can use heuristics to prune the search tree. For example, if a partial search path already results in more than two missing y-ions, one may stop the search immediately and move to a different path. While the pruning generally speeds up the algorithm, there is no guarantee that the speed-up is enough to make the search algorithm efficient. Additionally, there is a chance that the path for the correct solution is pruned prematurely. The dilemma between search efficiency (that desires more aggressive pruning) and efficacy (that desires less aggressive pruning) is hard to solve.

In 1999, Dancik *et al.*[16] first used dynamic programming to solve the *de novo* sequencing problem. Dynamic programming is a standard algorithm design technique. Instead of exhaustively searching every feasible solution in the solution space, a dynamic programming algorithm exploits the structure of the search space, and constructs the optimal solution in polynomial time. This both ensures the algorithm's efficiency and the solution's optimality. However, dynamic programming only works when the solution space is well structured and satisfies some special properties. For this reason, choosing the right combinatorial model is crucial for the use of dynamic programming. In the next two sections, two different models commonly used in the literature are examined.

2.3.2 Spectrum Graph

Bartels described a "sequence spectrum" approach in 1990.[13] A cluster of fragment ions with different types are converted to a mass site on the peptide sequence. Then the algorithm walks on the sequence spectrum to connect sites with mass difference equal to a residue mass. This model was later used by authors of ref. 2 and ref. 14, and fully developed to the spectrum graph model by Dancik *et al.* in ref. 16. Let us first examine a simplified version of the spectrum graph model by only considering the y-ions.

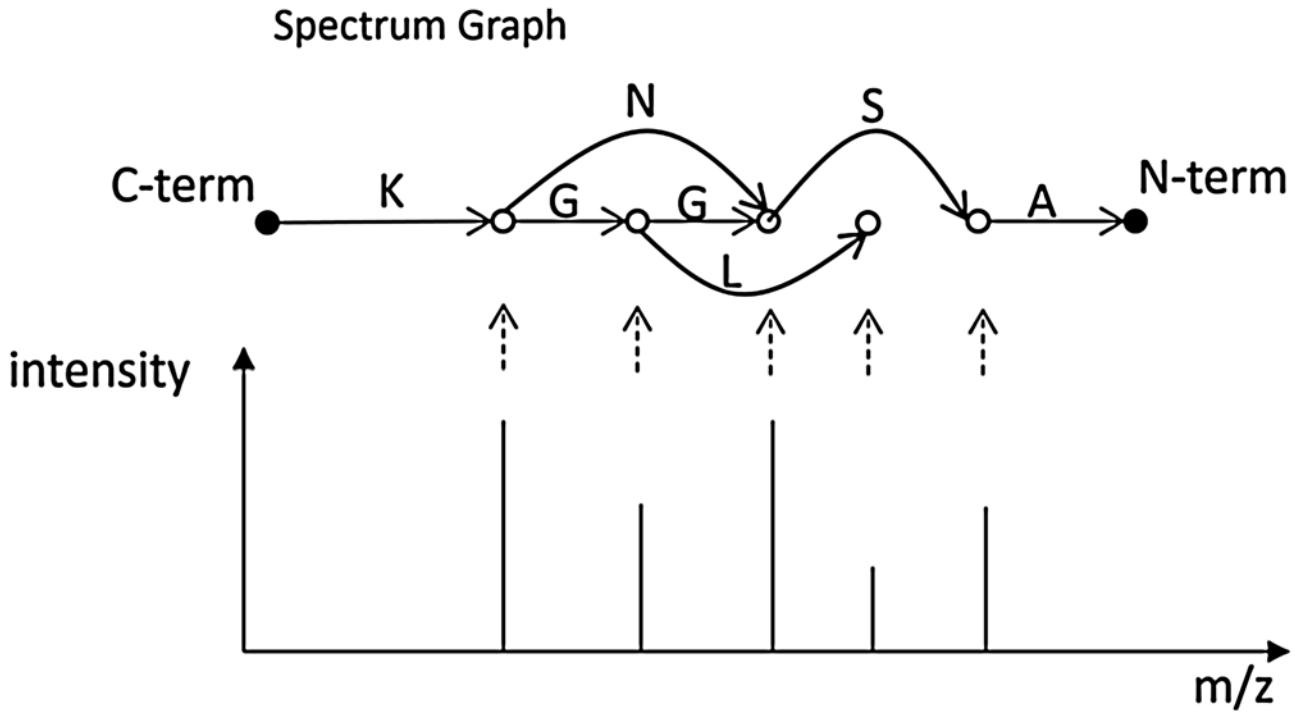

Figure 2.3 An example spectrum and its corresponding spectrum graph.

Figure 2.3 illustrates the construction of a spectrum graph from a spectrum. The procedure is outlined in the following:

1. Each peak in the spectrum corresponds to a vertex (the hollow circles) in the graph.
2. Two special vertices (the black dots) are added to the graph, representing the C-terminus and N-terminus of the peptide, respectively. The C-terminal and N-terminal vertices correspond to two imaginary peaks at 19.02 Da and the precursor mass, respectively. Note that because y-ions are concerned, the C-terminus vertex is at the left (low mass) end of the graph.
3. An edge is added to connect two vertices whenever their corresponding peaks' *m*/*z* values differ by the mass of a single amino acid residue. The edge is labeled with the residue name.
4. Assign a proper score for each edge.

With such a construction, finding the y-ion ladders in the spectrum is equivalent to finding a path in the graph that connects the C-terminal and N-terminal vertices. The edge labels on the path provide a candidate *de novo* sequence. Notice that there may be multiple paths that connect the two termini. The edge score comes into play here. With properly defined edge scores, the correct candidate should correspond to a path with the highest total edge score.

If the C-terminal and N-terminal vertices are connected in the spectrum graph, the optimal path can be computed with a simple dynamic programming algorithm. Suppose u_0, u_1, ..., u_n are the n vertices in the graph from left to right. Here u_0 and u_n are the C-terminal and N-terminal vertices, respectively. Let $P[i]$ denote the optimal partial path that connects u_0 with u_i. Let $S[i]$ be the total edge score of this optimal partial path. Moreover, let (u_j,u_i) be the last edge on $P[i]$. Then $P[i]$ must consist of the optimal partial path $P[j]$ from u_0 to u_j, plus the edge (u_j,u_i). Figure 2.4 illustrates the situation.

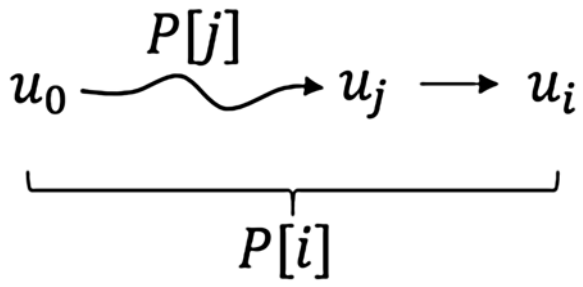

Figure 2.4 The optimal path from u_0 to u_i consists of an optimal path from u_0 to u_j, followed by the edge (u_j,u_i).

Therefore, $S[i] = S[j] + \text{score}(u_j,u_i)$. To find out which u_j precedes u_i, one only needs to try all possible j and select the one that maximizes $S[j] + \text{score}(u_j,u_i)$. Therefore, the following algorithm will compute the maximum weighted path.

Algorithm Spectrum Graph

1. Initialize $S[i] = -\infty$.
2. Let $P[0]$ be an empty path and $S[0] = 0$.
3. For i from 1 to n.
   a. Among all u_j that has an edge pointing to u_i, find the j that maximizes $S[j] + \text{score}(u_j,u_i)$
   b. Let $S[i] = S[j] + \text{score}(u_j,u_i)$ and $P[i] = P[j] + (u_j,u_i)$.
4. Output $P[n]$ as the reversed peptide sequence.

It is noteworthy that the algorithm is presented in a way that is easier understood by readers who are less familiar with dynamic programming. In a more canonical use of dynamic programming, one only needs to compute $S[i]$ but not the $P[i]$. The optimal path can be constructed by a standard back-tracking procedure after all $S[i]$ has been computed.

The previously-shown simplified model only uses the y-ions of a peptide, which is often insufficient to provide the complete sequence information. It would be beneficial if b-ions could also be used. In the model proposed by Dancik *et al.*,[16] each peak produces two vertices in the graph, corresponding to the two different interpretations of the peak (either a b-ion or a y-ion). Edges are added to connect two vertices if they have the same type and their mass values differ by a single residue. With this construction, a correct peptide corresponds to a pair of paths in the graph. One of the two paths uses only the b-ion vertices, and the other uses only the y-ion vertices. Since it is unlikely that one peak is both a b-ion and a y-ion, the model further requires that at most one of the two vertices for the same peak can be used in the two paths. This is called the antisymmetric path problem.[16] The paper claimed that there is a polynomial time algorithm for the problem. Later on, Chen *et al.*[17] published a polynomial time algorithm for the same problem independently.

Clearly, if there is a fragmentation site of which both b- and y-ions are missing, the path will be broken in the model, and the algorithm will fail. To address this problem, one can add edges between vertices with mass

difference equivalent to two amino acids.[2] With these additional edges, the paths are reconnected. Similarly, one can add edges to connect a gap of three or more residues. But these practices start to complicate the spectrum graph model, and introduce other problems including increased false positives.

Another important consideration of the spectrum graph model is how to score the edges (or the vertices). Even when the correct path is connected in the graph, an inferior scoring function may assign the highest score to a wrong path. The optimization of the scoring function will be examined in the "Scoring Function" section later in this chapter.

2.3.3 PEAKS Algorithm

Although the spectrum graph model is conceptually intuitive, the way it deals with missing ions greatly complicates the model. Ma *et al.*[18] dealt with the missing ions problem with a different approach and developed the PEAKS software.[3] In the PEAKS model, a spectrum is thought to have a peak at every mass. If no peak is present at a given mass, it is equivalent to having a 0-intensity peak. This small change completely avoids the missing ion problem. Since a proper scoring function usually favors the high abundant peaks, the algorithm will still try to use the real peaks with positive intensities first. But if such a solution is not possible, the algorithm has the freedom to use the 0-intensity peaks.

The original algorithm (called Sandwich algorithm) for the PEAKS model[18] is rather complicated. Here we examine a much simplified algorithm. The simplified algorithm has been previously presented at multiple conferences and workshops (such as the Symposium of Combinatorial Pattern Matching 2003) as a simpler case of the Sandwich algorithm.

To implement the PEAKS model, mass values are first discretized by multiplying a factor (*e.g.* 1000) and rounded to the nearest integer. To make the algorithm easier to understand, one can simply think that the nominal mass is used in all computation.

Suppose the peptide has a total residue mass M. The precursor ion of positive charge z would have m/z equal to $\frac{M + \text{mass}(H_2O)}{z}$ + mass(proton). Conversely, one can compute M from the precursor m/z and charge. Therefore, hereafter we assume M is known.

For a peptide sequence $a_1a_2...a_ia_{i+1}...a_n$, the fragmentation between a_i and a_{i+1} produces a prefix $a_1a_2...a_i$ and a suffix $a_{i+1}a_{i+2}...a_n$. Let $m(a)$ denote the mass of an amino acid residue a. Then the prefix mass $m_i = \sum_{j=1}^{i} m(a_j)$. Figure 2.5(a) illustrates an example peptide.

Each prefix mass m can be used to compute both the b-ion and y-ion mass values as illustrated in Figure 2.5(b). If peaks are found at the corresponding locations in the spectrum, a positive reward should be added to the score. Otherwise, a negative penalty should be added. Let $f(m)$ denote such a reward/penalty scheme. Notice that the definition of $f(m)$ only requires M and the spectrum, but not the actual peptide sequence.

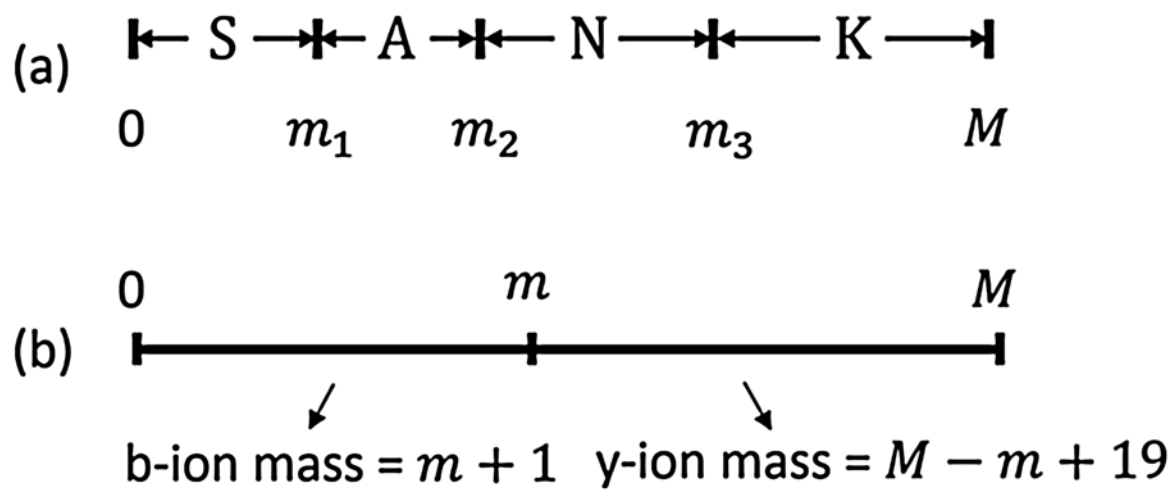

Figure 2.5 (a) An example peptide. Each prefix mass m_i defines a fragmentation site. (b) In general, a fragmentation at prefix mass m produces a b-ion with mass $m + 1$ and a y-ion with mass $M - m + 19$.

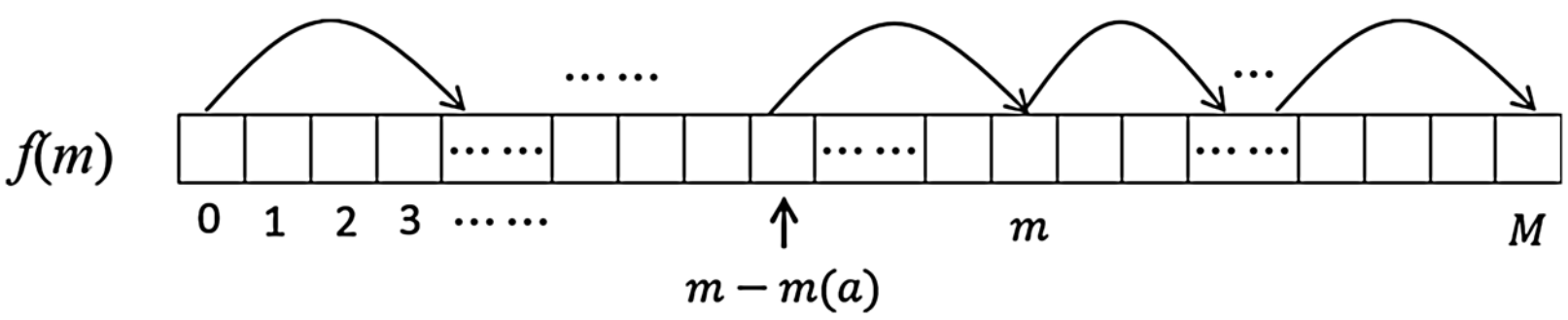

Figure 2.6 A peptide defines a path that connects mass 0 and M on the fragment score array.

Once the fragmentation score $f(m)$ is defined, the score of a peptide $P = a_1a_2...a_n$ is defined as $\text{score}(P) = \sum_{i=1}^{n-1} f(m_i)$. Here each $m_i = \sum_{j=1}^{i} m(a_j)$ is a prefix mass of the peptide. Thus, the task of *de novo* peptide sequencing becomes the finding of such a peptide P that maximizes score(P).

Intuitively, the optimal peptide defines a path that connects the mass 0 and M on the $f(m)$ score array (Figure 2.6). Each step of the path connects two cells with mass difference equivalent to a residue. The score of the path is the sum of the scores of the cells that the path visits. The score of a partial path from 0 to m can also be defined the same way. Let $P[m]$ be the partial path from 0 to m with the maximum score. Let $S[m]$ be the score of this optimal partial path. Suppose the last residue on $P[m]$ is a. It is easy to see that removing the last residue a from $P[m]$ gives an optimal partial path from 0 to $m - m(a)$. Therefore, $P[m] = (P[m - m(a)], a)$, and $S[m] = S[m - m(a)] + f(m)$. To find out the identity of a, one only needs to enumerate all possible residues and select the one that maximizes $S[m - m(a)]$. Thus, the optimal path can be computed with the following dynamic programming algorithm.

Algorithm
MassArray

1. Initialize $S[i] = -\infty$.
2. Let $P[0]$ be an empty path and $S[0] = 0$.
3. For m from 1 to M.
   a. Find the residue a that maximizes $S[m - m(a)]$.
   b. Let $S[m] = S[m - a] + f(m)$ and $P[m] = (P[m - m(a)], a)$.
4. Output $P[M]$ as the reversed peptide sequence.

This algorithm shares much similarity with the simplified spectrum graph model. However, unlike the spectrum graph model, the missing of a fragment ion is now only penalized through $f(m)$, but does not forbid the algorithm from finding a path that connects 0 and M.

The algorithm does not put any constraint to the actual definition of the fragmentation score $f(m)$. This leaves great flexibilities for a software implementation to optimize the scoring function. Such optimization will be discussed in the "Scoring Function" section later in this chapter.

There is a hidden problem if the MassArray algorithm is implemented in a straightforward way. The algorithm has a tendency to report a peptide that matches a highly abundant peak twice: once with a b-ion and the other time with a y-ion. By doing so, the peak contributes to the total fragmentation score twice at two different prefix mass values. In real-life peptides, the overlap of a pair of b- and y-ions is infrequent. However, since the algorithm is searching in all amino acid sequences (not just a database of real proteins), there is a great chance that the highest scoring peptide indeed double counts the highest abundant peaks.

For this reason, the earliest version of the PEAKS software used the more sophisticated "Sandwich algorithm".[18] Instead of dynamic programming with a single prefix mass m, the Sandwich algorithm used a pair of prefix mass m and suffix mass m' simultaneously. During the dynamic programming, the fragment ions at the two mass values are examined for possible overlap. A peak that is matched by more than one fragment ion is only counted once. The Sandwich algorithm solved the double-count problem. However, the complexity of that algorithm is significantly higher than with the algorithm MassArray described previously.

For computing efficiency, the algorithm used in later versions of the PEAKS software is based on the MassArray algorithm with many unpublished improvements. Other software tools that make use of the MassArray algorithm or a variation of the algorithm include the MSNovo,[19] DeNovoPTM,[20] and Novor.[5] Most of these tools use heuristic strategies to solve the double-count problem. First, the MassArray algorithm is called. If the resulting peptide annotates one or more major peaks with both b- and y-ions, the tool will then try different combinations of the annotation, and call the MassArray algorithm again with each of the combinations. With the extra rounds of computation, this practice is very effective in reducing the double-count effect.

2.4 Scoring Function

In a *de novo* sequencing program, the scoring function is the optimization goal of the algorithm. If the scoring function cannot score the correct sequence with the highest score, then it does not matter how fast the algorithm is. In this section we examine some techniques used by existing software to construct a good *de novo* sequencing scoring function.

For developing a scoring function for automated *de novo* sequencing, it is useful to examine the criteria that a human uses in manual *de novo*

sequencing. In general, a human often judges the correctness of a peptide by the following two facts:

1. Most of the major fragment ions of the peptide are observed in the spectrum.
2. Most of the highly abundant peaks in the spectrum are explained by some fragment ions.

In certain cases, the correlation between the fragmentation patterns and some special amino acid combinations can also be used to disambiguate the multiple explanations of the same spectrum. This has been illustrated in the manual *de novo* sequencing section, where the enhanced fragmentation at the N-terminal side of a proline is used to tell whether a dipeptide mass gap is [XP] or [PX].

However, there are a few technical difficulties to convert the human knowledge into a scoring function used by a computer algorithm. First, the human knowledge is usually qualitative. It is nontrivial to convert the qualitative knowledge into precise numeric values. For example, although the scoring function generally prefers abundant peaks, it is unclear how to convert an intensity value to a numeric score. Consider two peptide candidates that match two sets of peaks A and B, respectively. If A has one peak of intensity 300 that is not in B, and B has three peaks of intensity 100 that are not in A, should the scoring function rank A or B higher?

Secondly, the human knowledge is often *ad hoc*. Depending on different situations, the same person may apply different rules to judge the quality of the peptide spectrum matching. It is hard to have a complete list of all rules that a human expert would use. Even if such a complete list exists, there may be multiple rules that can apply to the same situation. Some of these rules may enhance or conflict each other. How to weigh the importance of each rule quantitatively becomes another nontrivial problem.

2.4.1 Likelihood Ratio

Likelihood ratio is a common way used in bioinformatics to define scoring functions. Likelihood ratio is first introduced into *de novo* sequencing by Dancik *et al.*[16] Today it serves the basis of the scoring functions in many different *de novo* sequencing tools.

Let S be a spectrum and P be a peptide. Consider a y-ion of P. Whether a peak corresponding to the y-ion appears in the spectrum is a random event, and the probability depends on whether P is the true peptide for S. Let

$$p = \Pr(\text{y-ion peak appears} \mid P \text{ is a correct peptide}), \text{ and}$$
$$q = \Pr(\text{y-ion peak appears} \mid P \text{ is a random peptide}).$$

Then, in calculating the peptide spectrum matching score, the contribution made by each y-ion match is equal to $\log\frac{p}{q}$. Similarly, the contribution

made by each unmatched y-ion is equal to $\log\frac{1-p}{1-q}$. Since normally $p > q$, the contribution is positive for a y-ion match and negative for a missing y-ion.

The values p and q can be easily determined from statistics using a large set of annotated spectra. The peptide spectrum matching score is then defined as the total of the scores of all y-ions of the peptide.

To also account for the b-ions, one can apply the same statistics to the b-ions, and add the b-ion scores to the scoring function. When P is the real peptide, the probability that a b-ion is matched is lower than the probability that a y-ion is matched. But when P is a random peptide, the two probabilities are similar. Therefore, it is easy to see that the log likelihood ratio of a b-ion match is lower than a y-ion match. This example shows that the scoring definition previously mentioned can automatically adjust the weights of different ion types.

Instead of only distinguishing match and mismatch, one can also account for peak intensities using the likelihood ratio idea. In ref. 4 and 21, the intensity is divided into a few intervals such as high, low, and absent. Then the score contribution of a y-ion matching a peak with intensity interval i is defined as:

$$\log\frac{\Pr(\text{y-ion peak's intensity in interval } i | P \text{ is a correct peptide})}{\Pr(\text{y-ion peak's intensity in interval } i | P \text{ is a random peptide})}$$

2.4.2 Utilization of Many Ion Types

Although the algorithms in the previous section are presented using b- and y-ions, a practical software tool usually considers many more ion types in its scoring function. For example, the Novor software considers nine ion types: y, b, a, y(2+), b(2+), b-18, b-17, y-18, and y-17. The consideration of more ion types does not usually increase the algorithm's complexity very much. However, in terms of the accuracy, the use of many ion types has a major impact. On one hand, more signal peaks may be matched by the additional ion types in consideration. On the other hand, an increasing number of false positive peak assignments may happen. Thus, a practical system will have to balance between the two effects, and use testing datasets from different data sources to determine the best subset of ion types to use.

2.4.3 Combined Use of Different Fragmentations

Nowadays, the most commonly used fragmentation methods for MS/MS include CID (Collision Induced Dissociation), HCD (High-energy Collision Dissociation), and ETD (Electron Transfer Dissociation). These methods may produce different spectra for the same peptide. The difference between ETD and the other two methods is particularly significant. While CID produces mostly y- and b-ions, the ETD method produces mostly c, z, and z + 1 ions. Additionally, they may preferably fragment different fragmentation sites of

the peptide. As a result, the combined use of these spectra for the same peptide may increase the *de novo* sequencing accuracy. This approach has been implemented in different tools such as those presented in ref. 22–25.

2.4.4 Machine Learning

The collective efforts of the proteomics community have produced a huge amount of publicly available mass spectrometry data. There are also well-annotated spectrum libraries. The availability of well annotated public datasets opened the possibility of using machine learning to automatically learn a scoring function. With certain off-the-shelf algorithms, machine learning can learn very sophisticated rules from the data. This fits our purpose perfectly: we know that there are a lot of rules that determine the fragmentation of a peptide and the formation of the peaks in the spectrum; however, we do not have a complete list of these rules and do not know how to quantitatively combine all the rules together. Thus, we rely on the machine learning algorithm to learn those rules from the data.

In ref. 5, Ma demonstrated the power of this approach with the development of the Novor software. With over 300 000 annotated MS/MS spectra from the NIST (National Institute of Standards and Technology) peptide spectral library, the machine learning algorithm automatically learned a decision tree with over 14 000 branching nodes. The decision tree is used to compute the confidence (probability of being correct) of each amino acid residue in a *de novo* sequence candidate. A peptide's score is then defined as the weighted average of its amino acids' confidence. To balance the heavy and light amino acids, the mass of each amino acid is used as the weight in computing the weighted average.

Figure 2.7 uses a trivial example to illustrate how the decision tree is used to determine the confidence of a residue. The algorithm starts the computation at the root of the tree, and keeps moving upward. At each branching node, it answers the yes or no question and moves up to one of its child nodes. Once a leaf node is reached, the value stored there is retrieved and returned as the confidence score of the residue. A decision tree used by a computer algorithm can be exceedingly large. However, the length of the path from the root to a leaf is usually short. This makes the score computation very efficient.

Meanwhile, any kind of 'yes-or-no' questions can be asked at each branching node. This allows us to use many different scoring features, such as different ion types, the intensities, the rank of peaks, the mass error, the charge state of the peptide, and the distance of the amino acid toward the N-terminus and C-terminus. Novor algorithm uses a total of 9 fragment ion types and 169 scoring features. The combined use of many scoring features significantly boosted the accuracy of the software. With the help of machine learning, Novor achieved a *de novo* sequencing speed of 300 spectra per second on a laptop computer, at a better accuracy than the state of the art.[5]

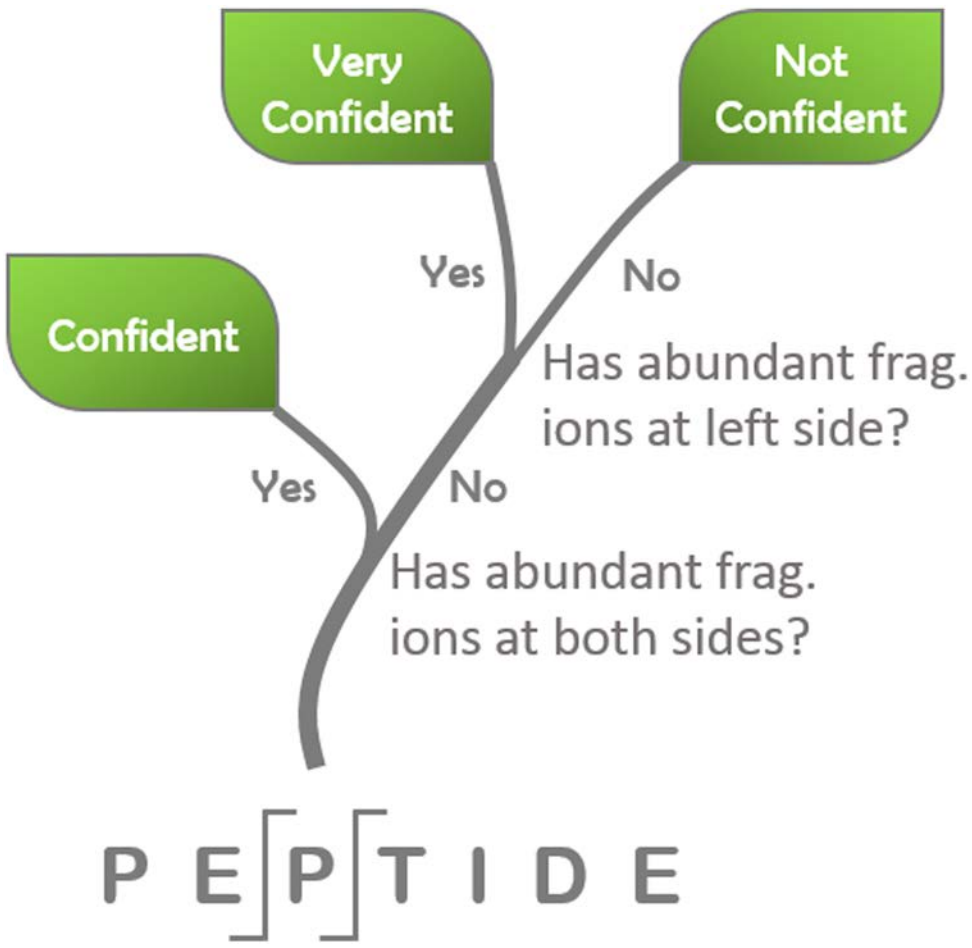

Figure 2.7 A trivial decision tree to determine the confidence of the amino acid P in the peptide sequence. (Figure with permission adapted from ref. 5.)

(a) [354.2]AFPAPLEK
(b) DAPAAFPAPLEK
1
0

Figure 2.8 Two ways to deal with low confident amino acids: (a) mass gap; (b) amino acid score.

Decision tree is not the only machine learning model that can be used. In fact, other models such as logistic regression and SVM are popularly used in defining scoring functions in both *de novo* sequencing and database search. In *de novo* sequencing, Novor is the first to apply machine learning on this scale of data size.

2.4.5 Amino Acid Score

An important fact about a *de novo* sequence is that not all amino acids have the same confidence. The highly confident amino acids are supported by strong b- and y-ions; whereas the least confident ones do not have any fragment ion support, and are computed merely as a filler to fill in a large mass gap. There are two ways in practical software to deal with the lower-confident amino acids. The first is to convert those amino acids to a mass gap (Figure 2.8(a)). Most software tools use this way. However, some software tools, including PEAKS[3] and Novor,[5] output an amino acid score for each amino acid in the *de novo* sequence (Figure 2.8(b)).

The second way is more flexible since a user can choose different score thresholds to do the mass gap conversion in different applications. However, the definition of an accurate amino acid score function often involves

additional work on top of defining the peptide scoring function. PEAKS' amino acid score function is unpublished. Novor's amino acid score is directly computed from the decision tree. A third software tool, PepNovo[4] does not have a built-in amino acid score. But in a separate work, Frank *et al.*[8] discussed an amino acid scoring function to filter the results of PepNovo.

2.5 Computer Software

In this section we review a list of better known *de novo* sequencing software tools in chronological order. Most of these tools are free, or free for academic use, with the exception of PEAKS and Sherenga that are commercial. The main purpose of this list is to review the novel techniques introduced by these tools, instead of the software itself. In fact, many of these software tools are no longer actively maintained.

2.5.1 Lutefisk

Taylor and Johnson published the Lutefisk software for *de novo* sequencing in 1997.[2] Although Lutefisk is not the first *de novo* sequencing software, it is likely the first to reach a fairly broad acceptance in the *de novo* sequencing community.[10] It employs an exhaustive search algorithm with many heuristic improvements, including the search tree pruning outlined earlier in this chapter. Further, if the number of partial solutions exceeds a predefined threshold during the search, Lutefisk will discard the lower scoring ones. Lutefisk has always been a free tool and is now released under GNU General Public License.

2.5.2 Sherenga

Dancik *et al.* published the Sherenga algorithm in 1999.[16] In their paper, the spectrum graph and antisymmetric path model were formally proposed. Without giving the details, their paper claimed that a dynamic programming algorithm would solve the antisymmetric path problem. The paper also proposed the use of the log likelihood ratio score in *de novo* sequencing. Sherenga was later incorporated in the commercial software Spectrum Mill as its *de novo* sequencing module.

2.5.3 PEAKS

Ma *et al.* released the PEAKS software as a commercial tool in 2002 at ASMS (American Society of Mass Spectrometry) annual conference. The software was published in ref. 3 and the algorithm was published in ref. 18. Unlike the spectrum graph model, PEAKS performs dynamic programming on the mass array. It also introduced a two round search scheme: the first round generates a large number of candidates with a simpler scoring function, and the second round re-scores the candidates with a more sophisticated scoring

function. PEAKS soon became the *de facto* standard software for *de novo* sequencing in this field. It is actively developed at Bioinformatics Solutions Inc. since its release.

2.5.4 PepNovo

Frank and Pevzner published the PepNovo software as a free tool in 2005.[4] PepNovo uses the spectrum graph model. Noticing that the intensities of different ion types are correlated, PepNovo uses a Bayesian network in its scoring function to capture the correlation. This is more accurate than treating the different ion types as independent factors (as done in many other software tools). PepNovo is still widely used as a free *de novo* sequencing tool.

2.5.5 DACSIM

Zhongqi Zhang developed a spectrum simulation method to predict the experimental spectrum for a given peptide in ref. 26. In a separate work, he employed the spectrum simulation for *de novo* peptide sequencing.[27] His DACSIM algorithm uses the similarity between a predicted spectrum and the real spectrum as the scoring function to measure a peptide candidate; and uses a divide-and-conquer algorithm to search for the best *de novo* sequence. This is the first tool that utilizes the predicted spectrum in the scoring function.

2.5.6 NovoHMM

Fischer *et al.* published NovoHMM in 2005.[28] The most notable feature of the NovoHMM program is that it uses a Hidden Markov Model (HMM) to formulate the *de novo* sequencing problem. This is different from both the spectrum graph and the mass array model. With HMM, the correlations between adjacent fragmentation peaks can be considered in the scoring function. This helped improve the accuracy of the *de novo* sequencing result.

2.5.7 MSNovo

Mo *et al.* published MSNovo in 2007.[19] It uses the mass array based dynamic programming. The paper includes many technical details on improving the scoring function and the algorithm's efficiency. Some advantages of using mass array over spectrum graph were reviewed in their paper.

2.5.8 PILOT

DiMaggo and Floudas published the PILOT software in 2007.[29] PILOT is unique for the integer linear programming (ILP) algorithm it uses. The *de novo* sequencing problem is formulated as an ILP problem. Once formulated,

there are standard solvers to solve the ILP. Multiple candidates were produced by solving the ILP, and then a post-processing is used to compare each candidate with the spectrum and pick the best.

2.5.9 pNovo

Chi *et al.* published the pNovo program in 2010.[30] It uses the spectrum graph approach and an empirical scoring function. The earlier version was designed to work on MS/MS spectrum produced by HCD (High-energy Collision Dissociation), but later versions also work on other fragmentation methods. The accuracy of the program was also improved periodically in these later versions.

2.5.10 Novor

Ma published the Novor program in 2015.[5] The main novelty of Novor is the use of a large scale machine learning approach to build its scoring function. The decision tree was used as the base model for machine learning. Novor's basic algorithm is the mass array based dynamic programming, with an additional refinement step to improve the result of dynamic programming. Novor is the most notable for its *de novo* sequencing speed of over 300 spectra per second on a laptop computer.

2.6 Conclusion: Applications and Limitations of *De novo* Sequencing

2.6.1 Sequencing Novel Peptides and Detecting Mutated Peptides

De novo peptide sequencing is the only viable choice if the organism in study does not have a protein sequence database. This makes it a useful method for studying the organisms whose genomes are not sequenced yet, although generation of protein databases from RNA sequencing data now provides a powerful alternative (see Chapter 16). *De novo* sequencing has also been used when the available protein database is incomplete and does not include the peptides of interest. For example, it has been used to study the neuropeptides[31] and peptides in venoms.[32,33]

Even for a well-characterized organism such as a human, the commonly used protein sequence databases do not include all the single amino acid polymorphisms (SAPs). *De novo* sequencing is useful for the identification of those mutated peptides that are different from the ones in the database. Special sequence search algorithms have been developed to utilize both the *de novo* sequencing results and the sequence database to detect those mutations. A more thorough review of this direction can be found in the review article.[10]

2.6.2 Assisting Database Search

De novo sequencing has also been used to assist the database search method to identify peptides in the database. *De novo* sequencing improves a database search method in both its speed and accuracy.

To improve the speed, the *de novo* sequence tags can be used to filter the sequence database, and only the approximately matching peptides are compared with the spectrum. This can significantly speed up the database search. With the availability of the super-fast *de novo* sequencing engine, Novor,[5] this advantage becomes very realistic.

To improve accuracy, the same spectrum is used for both *de novo* sequencing and database search. The agreement (or partial agreement) between the two results improve the confidence about the database search result. This has been used in the PEAKS DB algorithm to obtain a more accurate scoring function that better separates the true and false peptide identifications.[7] As a result, a higher number of true peptide-spectrum matches can be obtained at a lower false discovery rate.

2.6.3 *De novo* Protein Sequencing

De novo sequencing is mostly used for sequencing short peptides (usually shorter than 50 amino acids). However, in certain applications one may be interested in *de novo* sequencing a long protein. This can be achieved with a number of approaches.

The most common approach is to digest the protein with multiple enzymes to produce overlapping peptides. Then each peptide is *de novo* sequenced from its MS/MS spectrum. By utilizing the overlaps, one can assemble the peptides and reconstruct the original protein sequence. A number of successful applications and variants of this approach have been reported in the literature (*e.g.* ref. 36–38).

Another approach is to use top-down proteomics, where an intact protein is measured with MS/MS directly. There have been some pioneer works in this direction. However, the top-down mass spectrometry experiments are not fully developed yet, and the *de novo* sequencing software with top-down mass spectrometry data is very rudimentary. A review of the latest developments in top-down proteomics can be found in ref. 34. An algorithm for sequencing short sequence tags from top-down MS/MS spectrum was reported in ref. 35.

The third approach is to combine the top-down and bottom-up data together. The bottom-up data are used to *de novo* sequence the peptides, and the top-down data is used to guide the peptide assembly. Liu *et al.* presented an algorithm using this approach.[39]

2.6.4 Unspecified PTM Characterization

Normally a protein sequence database does not contain information about post-translational modifications (PTM). Thus, to identify peptides that have variable PTMs on them, a database search engine will have to try multiple

versions of the same sequence, with different PTMs turned on and off. This expands the search space exponentially with an increasing number of variable PTMs considered. As such, the database search algorithms can only allow the specification of very few variable PTMs for the search. The spectra for the peptides with unspecified PTMs are therefore left unassigned.

De novo sequencing may be used to obtain partial sequences for those unassigned spectra. Then the partial sequences are used to select peptides from the database, and the mass difference between the theoretical peptide mass and the observed peptide mass can be used to characterize the unspecified PTM. Such application of *de novo* sequencing in PTM characterization has been described in a number of publications including the PEAKS PTM algorithm[40] and InsPecT algorithm.[9]

2.6.5 Limitations

When the MS/MS spectrum does not include a complete ladder of fragment ion peaks (which is usually the case), *de novo* sequencing may not be able to sequence the whole peptide correctly. This is the single largest limitation of *de novo* peptide sequencing. This limitation has been gradually lessened in the past two decades, due to the continuous improvements in both computer algorithms and mass spectrometry instruments that are outlined in the following.

The availability of the amino acid score in certain *de novo* sequencing tools can help improve the result accuracy by filtering out the low confident amino acids. But this does not help with the sequence coverage and still leaves portions of the peptides un-sequenced. The correlation between certain sequence patterns and the peak missing events have been used to help identify the sequence when the peak ladder is incomplete.[5] The improvement of mass spectrometers' mass accuracy and signal to noise level has greatly improved the *de novo* sequencing accuracy and coverage. Additionally, the combined use of different fragmentation methods (such as CID and ETD) as described in Section 2.4 can help further improve both the accuracy and coverage. There have also been efforts to improve the quality of the ladder peaks by changing the mass spectrometry experiments, such as by labeling one terminus of the peptide,[41] and by using new fragmentation methods.[42,43] Finally, even if only a portion of each peptide can be confidently sequenced, one may still *de novo* sequence the whole protein by utilizing many overlapping peptides, each contributing a different segment of the protein.

With these continuous improvements in both the *de novo* sequencing software and the mass spectrometer hardware, the author speculates that *de novo* sequencing's use in proteomics will be greatly broadened in the coming years, to the extent that most applications in proteomics can benefit from the integration of *de novo* sequencing in their data analysis pipelines.

Acknowledgements

The work is supported by an NSERC discovery grant (RGPIN 238748).

References

1. D. F. Hunt, J. R. Yates, J. Shabanowitz, S. Winston and C. R. Hauer, Protein sequencing by tandem mass spectrometry, *Proc. Natl. Acad. Sci. U. S. A.*, 1986, **83**(17), 6233–6237.
2. J. A. Taylor and R. S. Johnson, Sequence database searches via de novo peptide sequencing by tandem mass spectrometry, *Rapid Commun. Mass Spectrom.*, 1997, **11**, 1067–1075.
3. B. Ma, K. Zhang, C. Hendrie, C. Liang, M. Li, A. Doherty-Kirby and G. Lajoie, PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry, *Rapid Commun. Mass Spectrom.*, 2003, **17**, 2337–2342.
4. A. Frank and P. A. Pevzner, PepNovo: de novo peptide sequencing via probabilistic network modeling, *Anal. Chem.*, 2005, **77**(4), 964–973.
5. B. Ma, Novor: Real-Time Peptide de Novo Sequencing Software, *J. Am. Soc. Mass Spectrom.*, 2015, **26**, 1885–1894.
6. J. M. Chick, D. Kolippakkam, D. P. Nusinow, B. Zhai, R. Rad, E. L. Huttlin and S. P. Gygi, A mass-tolerant database search identifies a large proportion of unassigned spectra in shotgun proteomics as modified peptides, *Nat. Biotechnol.*, 2015, **33**(7), 743–749.
7. J. Zhang, L. Xin, B. Shan, W. Chen, M. Xie, D. Yuen, W. Zhang, Z. Zhang, G. a. Lajoie and B. Ma, PEAKS DB: De Novo Sequencing Assisted Database Search for Sensitive and Accurate Peptide Identification, *Mol. Cell. Proteomics*, 2012, **11**, DOI: 10.1074/mcp.M111.010587.
8. A. Frank, S. Tanner, V. Bafna and P. Pevzner, Peptide Sequence Tags for Fast Database Search in Mass-Spectrometry, *J. Proteome Res.*, 2005, **4**(4), 1287–1295.
9. S. Tanner, H. Shu, A. Frank, L. C. L. C. Wang, E. Zandi, M. Mumby, P. a. Pevzner and V. Bafna, InsPecT: Identification of posttranslationally modified peptides from tandem mass spectra, *Anal. Chem.*, 2005, **77**(14), 4626–4639.
10. B. Ma and R. Johnson, De Novo Sequencing and Homology Searching, *Mol. Cell. Proteomics*, 2012, **11**, DOI: 10.1074/mcp.O111.014902.
11. M. W. Duncan, R. Aebersold and R. M. Caprioli, The pros and cons of peptide-centric proteomics, *Nat. Biotechnol.*, 2010, **28**(7), 659–664.
12. L. a. Breci, D. L. Tabb, J. R. Yates and V. H. Wysocki, Cleavage N-terminal to proline: Analysis of a database of peptide tandem mass spectra, *Anal. Chem.*, 2003, **75**(9), 1963–1971.
13. C. Bartels, Fast algorithm for peptide sequencing by mass spectroscopy, *Biomed. Environ. Mass Spectrom.*, 1990, **19**, 363–368.
14. W. M. Hines, A. M. Falick, A. L. Burlingame and B. W. Gibson, Pattern-Based Algorithm for Peptide Sequencing from Tandem High Energy Collision-Induced Dissociation Mass Spectra, *J. Am. Soc. Mass Spectrom.*, 1992, **3**(4), 326–336.
15. R. S. Johnson and J. A. Taylor, Searching sequence databases via de novo peptide sequencing by tandem mass spectrometry, *Mol. Biotechnol.*, 2002, **22**(3), 301–315.

16. D. Dancik, T. A. Addona, K. R. Clauser, J. E. Vath and P. A. Pevzner, De novo peptide sequencing via tandem mass spectrometry, *J. Comput. Biol.*, 1999, **6**, 327–342.
17. T. Chen, M. Y. Kao, M. Tepel, J. Rush and G. M. Church, A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry, *J. Comput. Biol.*, 2001, **8**(3), 325–337.
18. B. Ma, K. Zhang and C. Liang, An effective algorithm for peptide sequencing from MS/MS spectra, *J. Comput. Syst. Sci.*, 2005, **70**(3), 418–430.
19. L. Mo, D. Dutta, Y. Wan and T. Chen, MSNovo: a dynamic programming algorithm for de novo peptide sequencing via tandem mass spectrometry, *Anal. Chem.*, 2007, **79**(13), 4870–4878.
20. L. He, X. Han and B. Ma, De Novo Sequencing with Limited Number of Post-Translational Modifications per Peptide, *J. Bioinf. Comput. Biol.*, 2013, **11**(4), 1350007.
21. X. Liu, B. Shan, L. Xin and B. Ma, Better score function for peptide identification with ETD MS/MS spectra, *BMC Bioinf.*, 2010, **11**(suppl 1), S4.
22. M. M. M. Savitski, M. L. M. M. L. Nielsen, F. Kjeldsen and R. a. Zubarev, Proteomics-grade de novo sequencing approach, *J. Proteome Res.*, 2005, **4**(6), 2348–2354.
23. R. Datta and M. Bern, Spectrum fusion: using multiple mass spectra for de novo Peptide sequencing, *J. Comput. Biol.*, 2009, **16**(8), 1169–1182.
24. A. Bertsch, A. Leinenbach, A. Pervukhin, M. Lubeck, R. Hartmer, C. Baessmann, Y. A. Elnakady, R. Müller, S. Böcker, C. G. Huber and O. Kohlbacher, De novo peptide sequencing by tandem MS using complementary CID and electron transfer dissociation, *Electrophoresis*, 2009, **30**(21), 3736–3747.
25. L. He and B. Ma, ADEPTS: advanced peptide de novo sequencing with a pair of tandem mass spectra, *J. Bioinf. Comput. Biol.*, 2010, **8**(6), 981–994.
26. Z. Zhang, Prediction of Low-Energy Collision-Induced Dissociation Spectra of Peptides, *Anal. Chem.*, 2004, **76**, 3908–3922.
27. Z. Zhang, De novo peptide sequencing based on a divide-and-conquer algorithm and peptide tandem spectrum simulation, *Anal. Chem.*, 2004, **76**(21), 6374–6383.
28. B. Fischer, V. Roth, F. Roos, J. Grossmann, S. Baginsky, P. Widmayer, W. Gruissem and J. M. Buhmann, NovoHMM: a hidden Markov model for de novo peptide sequencing, *Anal. Chem.*, 2005, **77**(22), 7265–7273.
29. P. A. DiMaggio and C. A. Floudas, De novo peptide identification via tandem mass spectrometry and integer linear optimization, *Anal. Chem.*, 2007, **79**(4), 1433–1446.
30. H. Chi, R.-X. Sun, B. Yang, C.-Q. Song, L.-H. Wang, C. Liu, Y. Fu, Z.-F. Yuan, H.-P. Wang, S.-M. He and M.-Q. Dong, pNovo: de novo peptide sequencing and identification using HCD spectra, *J. Proteome Res.*, 2010, **9**(5), 2713–2724.
31. G. Menschaert, T. T. M. Vandekerckhove, G. Baggerman, B. Landuyt, J. V. Sweedler, L. Schoofs, W. Luyten and W. Van Criekinge, A hybrid, de novo based, genome-wide database search approach applied to the sea urchin neuropeptidome, *J. Proteome Res.*, 2010, **9**(2), 990–996.

32. K. W. Sanggaard, T. F. Dyrlund, L. R. Thomsen, T. a. Nielsen, L. Brøndum, T. Wang, I. B. Thøgersen and J. J. Enghild, Characterization of the gila monster (Heloderma suspectum suspectum) venom proteome, *J. Proteomics*, 2015, **117**, 1–11.
33. K. D. Zaqueo, A. M. Kayano, R. Simões-Silva, L. S. Moreira-Dill, C. F. C. Fernandes, A. L. Fuly, V. G. Maltarollo, K. M. Honório, S. L. da Silva, G. Acosta, M. A. O. Caballol, E. de Oliveira, F. Albericio, L. a Calderon, A. M. Soares and R. G. Stábeli, Isolation and biochemical characterization of a new thrombin-like serine protease from Bothrops pirajai snake venom, *BioMed Res. Int.*, 2014, **2014**, 595186.
34. A. D. Catherman, O. S. Skinner and N. L. Kelleher, Top Down proteomics: facts and perspectives, *Biochem. Biophys. Res. Commun.*, 2014, **445**(4), 683–693.
35. K. Vyatkina, S. Wu, L. J. M. Dekker, M. M. VanDuijn, X. Liu, N. Tolić, M. Dvorkin, S. Alexandrova, T. M. Luider, L. Paša-Tolić and P. a. Pevzner, De novo sequencing of peptides from top-down tandem mass spectra, *J. Proteome Res.*, 2015, 150928070803005.
36. X. Liu, Y. Han, D. Yuen and B. Ma, Automated protein (re)sequencing with MS/MS and a homologous database yields almost full coverage and accuracy, *Bioinformatics*, 2009, **25**(17), 2174–2180.
37. N. Bandeira, K. R. Clauser and P. a. Pevzner, Shotgun Protein Sequencing: Assembly of Peptide Tandem Mass Spectra from Mixtures of Modified Proteins, *Mol. Cell. Proteomics*, 2007, **6**(7), 1123–1134.
38. N. Bandeira, V. Pham, P. Pevzner, D. Arnott and J. R. Lill, Automated de novo protein sequencing of monoclonal antibodies, *Nat. Biotechnol.*, 2008, **26**(12), 1336–1338.
39. X. Liu, L. J. M. Dekker, S. Wu, M. M. Vanduijn, T. M. Luider, N. Tolic, Q. Kou, M. Dvorkin, S. Alexandrova, K. Vyatkina and L. Pas, De Novo Protein Sequencing by Combining Top-Down and Bottom- Up Tandem Mass Spectra, *J. Proteome Res.*, 2014, **13**, 3241–3248.
40. X. Han, L. He, L. Xin, B. Shan and B. Ma, PeaksPTM: Mass spectrometry-based identification of peptides with unspecified modifications, *J. Proteome Res.*, 2011, **10**, 2930–2936.
41. A. Chacon, D. S. Masterson, H. Yin, D. C. Liebler and N. A. Porter, N-Terminal amino acid side-chain cleavage of chemically modified peptides in the gas phase: A mass spectrometry technique for N-terminus identification, *Bioorg. Med. Chem.*, 2006, **14**(18), 6213–6222.
42. L. Zhang and J. P. Reilly, Peptide de novo sequencing using 157 nm photodissociation in a tandem time-of-flight mass spectrometer, *Anal. Chem.*, 2010, **82**(3), 898–908.
43. H. Bin Oh and M. Bongjin, Radical-Driven Peptide Backbone Dissociatoin Tandem Mass Spectrometry, *Mass Spectrom. Rev.*, 2015, **34**, 116–132.

CHAPTER 3

Peptide Spectrum Matching via *Database Search and Spectral Library Search*

BRIAN NETZEL[a] AND SURENDRA DASARI*[b]

[a]Department of Laboratory Medicine and Pathology, Mayo Clinic, Rochester, MN, USA; [b]Department of Health Sciences Research, Mayo Clinic, Rochester, MN, USA
*E-mail: Dasari.Surendra@mayo.edu

3.1 Introduction

Peptide and protein identification *via* mass spectrometry (MS) is the mainstay of the entire field of proteomics. This peptide-based mass spectrometry analysis is referred to as bottom-up proteomics or shotgun proteomics. Chronologically, the shotgun proteomics can be further subdivided into two major categories: peptide mass fingerprinting (PMF) and tandem mass spectrometry analysis (MS/MS). In 1993, various groups independently proposed the PMF method as a rapid method for peptide and protein identification.[1–5] This method starts by digesting the proteins present in the sample into peptides using a proteolytic enzyme (Trypsin is the most popular enzyme of choice.). A mass spectrometer is employed to measure the masses of the resulting peptides. The measured masses are compared to a protein sequence database using algorithms like MOWSE.[5] These algorithms start by

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

cutting the protein sequences in the database into peptide sequences using the experimental proteolytic enzyme. The theoretical masses of the resulting peptide sequences are computed and compared to the experimentally measured peptide masses. The results are subjected to statistical analysis to find the best set of peptide and protein matches for the set of experimental peptide masses. While this method is capable of rapidly identifying peptides and proteins from simple mixtures (of ≤2 proteins), the method is confounded by more complex protein mixtures. This is mainly because the PMF method assumes that peptide masses observed in a single experiment uniquely map to a set of peptides that come from one protein. Most PMF algorithms are confounded when this covenant is broken in a complex peptide digest where a peptide mass can match to multiple peptide sequences, which in turn can map to multiple protein sequences.

Tandem mass spectrometry (MS/MS) was developed to resolve ties observed in PMF experiments. When an experimental peptide mass matches to multiple different peptide sequences in a database, additional information is needed to break the tie. This is often accomplished by configuring the mass spectrometer to isolate the peptide ion of interest, fragment it *via* collision induced dissociation (CID), measure the masses of the resulting peptide fragment ions, and record their masses and intensities in a new mass spectrum. This mass spectrum of fragment ions is called a tandem mass spectrum (MS/MS) or MS2 spectrum. An MS2 spectrum contains all of the information necessary to identify the peptide that produced the spectrum.

There are two main methods for matching a peptide to an MS2 spectrum. In 1994, Eng *et al.*, reported in a seminal publication, a computer algorithm to match the peptide MS2 spectrum to protein amino acid sequences present in a database.[6] This method, known as database searching, is the most often used algorithmic method in the field of shotgun proteomics. Independently, once a peptide-spectrum match has been confidently made, it can be stored in a spectral library and used for rapid identification of MS2 spectra that are produced by the same peptide in many different samples.[7] This spectral library searching concept has recently emerged as an alternative to the traditional protein sequence database search method. The spectral library search concept significantly differs from database searching in two ways: (a) the method does not rely on the presence of a protein amino acid sequence database, and (b) the method is several orders of magnitude faster at identifying experimental MS/MS when compared to a traditional database search. In this chapter we will introduce the basic concept of peptide-spectrum matching (PSM) using the MyriMatch database search engine[8] as an example.

The importance of PSM algorithms to the field of shotgun proteomics is unarguable. The use of PMF method has largely been discontinued by the proteomics community as we collectively moved into the post-genome high-throughput sequencing era. This is mainly because the PMF method requires isolation of pure proteins from the biological matrix, which is a very time consuming task. Hence, proteomics researchers have adopted automated MS/MS-based shotgun proteomics methods for routine analysis. For

example, in 2011, Nagarjuna *et al.* reported the detection of ~160 K unique peptides belonging to ~10 K distinct human proteins in a cell lysate when using MS/MS-based shotgun proteomics.[9] This type of deep proteomic analysis has become the method of choice for characterizing both simple and complex biological matrices. This is often accomplished using modern, fast scanning, high resolution, mass spectrometers that have unprecedented sampling speed and they often produce thousands to millions of MS2 spectra per data set. Manually interrogating these spectra for peptide identification is impossible and proper peptide-spectrum matching is the first key step towards a successful proteomic analysis.

3.2 Protein Sequence Databases

All database search algorithms function by matching the MS2 spectra to protein amino acid sequences present in a database. One of the most basic requirements to obtain a correct peptide sequence match to an MS2 spectrum is, that the sequence of the peptide that generated the experimental spectrum must be in the protein sequence database. For example, consider the peptide fragmentation spectrum match shown in Figure 3.1. If the peptide sequence "GEMFILEKGEYPR" that produced the MS2 spectrum shown in Figure 3.1 is not present in the database, the database search algorithms will either fail to match the MS/MS or produce an inferior (and incorrect) match to another homologous peptide sequence. Hence, completeness of the protein sequence database is paramount to the success of peptide-spectrum matching and therefore to the production of accurate peptide identification search results.

There are a wide variety of protein sequence databases that are freely available over the Internet. These databases are generally divided into two distinct categories: repositories or curated. Table 3.1 presents some of the well-used sequence repositories for proteomics research. Repository-style databases are typically derived by *in silico* translation of an organism's reference genome (or transcriptome) into the corresponding proteome.[10] A major portion of the sequence predictions contained in these databases have not been verified to be present in a living system. GenPept is a good example of a pure sequence repository-style database as it is a conglomerate of genomic sequence translations obtained from multiple institutions.[11] The sequence redundancy present in GenPept type databases unnecessarily increases the computational time of the search algorithms and also produces redundant protein identifications. To remedy this, some of the repository-style databases like RefSeq provide a non-redundant collection of sequences for a limited number of species. However, these non-redundant, repository-style, sequence databases still contain vast numbers of unverified entries.

Curated protein sequence databases are derived from the repository-style databases by consolidating and compiling multiple reports for any given protein into a single entry, thereby vastly reducing the sequence redundancy in the database. Next, a team of curation experts comb through the entries

Peptide GEMFILEKGEYPR		Mass (m/z; 2+) 1567.77 (784.89)			
Mass (+1)	**B-ion series**	**#**	**Mass (+1)**	**Y-ion series**	**#**
58.02933	G	b1	1511.75687	E-M-F-I-L-E-K-G-E-Y-P-R	y12
187.07193	G-E	b2	1382.71428	M-F-I-L-E-K-G-E-Y-P-R	y11
318.11241	G-E-M	b3	1251.67379	F-I-L-E-K-G-E-Y-P-R	y10
465.18083	G-E-M-F	b4	1104.60538	I-L-E-K-G-E-Y-P-R	y9
578.26489	G-E-M-F-I	b5	991.52131	L-E-K-G-E-Y-P-R	y8
691.34895	G-E-M-F-I-L	b6	878.43725	E-K-G-E-Y-P-R	y7
820.39155	G-E-M-F-I-L-E	b7	749.39466	K-G-E-Y-P-R	y6
948.48651	G-E-M-F-I-L-E-K	b8	621.29969	G-E-Y-P-R	y5
1005.50797	G-E-M-F-I-L-E-K-G	b9	564.27823	E-Y-P-R	y4
1134.55056	G-E-M-F-I-L-E-K-G-E	b10	435.23564	Y-P-R	y3
1297.61389	G-E-M-F-I-L-E-K-G-E-Y	b11	272.17231	P-R	y2
1394.66666	G-E-M-F-I-L-E-K-G-E-Y-P	b12	175.1195	R	y1

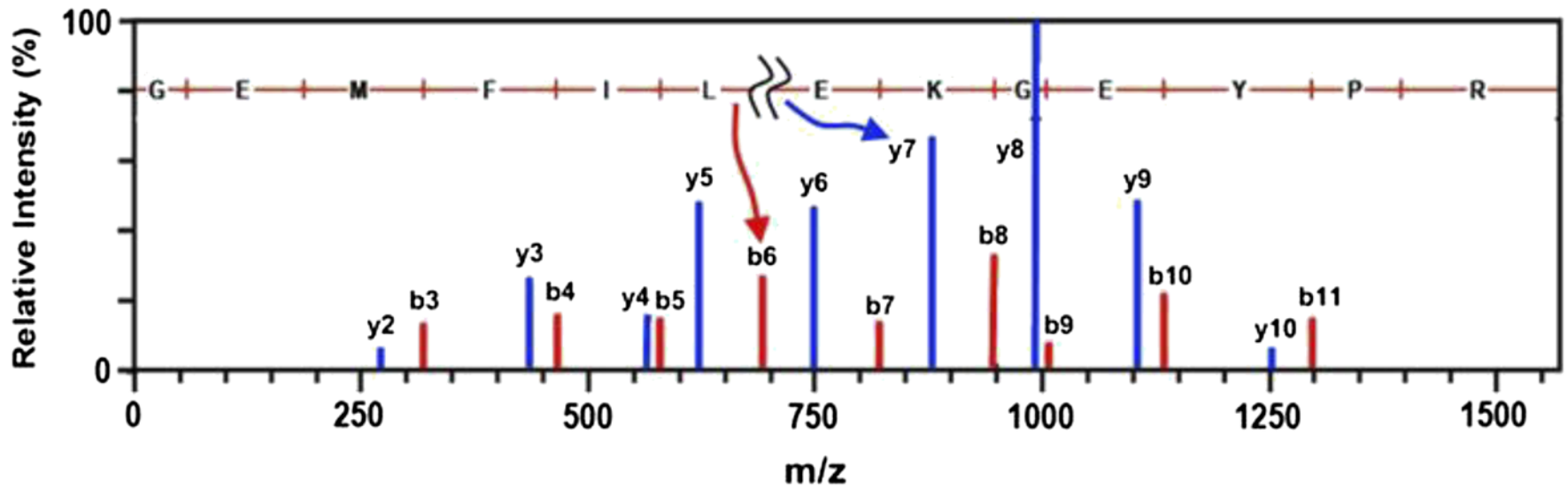

Figure 3.1 An idealized CID spectrum. Collision induced dissociation (CID) of peptide with sequence "GEMFILEKGEYPR" results in breaking of the peptide between amide bonds. Each cleavage produces a pair of fragment ions (called b-ion and y-ion) that are recorded in the mass spectrum. Complete dissociation of peptide will produce fragment ions from the entire backbone.

Table 3.1 List of Protein Sequence Databases. The listed databases are available, over the Internet, free-of-charge.

Database name	Website
Ensembl	http://useast.ensembl.org/index.html
GenPept	http://www.ncbi.nlm.nih.gov/protein
ProteinInformation Resource (PIR)	http://pir.georgetown.edu
Reference Sequence (RefSeq)	http://www.ncbi.nlm.nih.gov/refseq
SwissProt	http://web.expasy.org/docs/swiss-prot_guideline.html
UniprotKB/TrEMBL	http://web.expasy.org/docs/swiss-prot_guideline.html

and cull any protein sequences that do not have prior evidence in the literature. These experts also synthesize the biological and pathological significance of each protein sequence and annotate the corresponding entry, which increases the quality and reliability of the database.[12] The Protein Information Resource Database (PIR-PSD), created in 1984, is the oldest example of

a curated database, and focuses on classification *via* protein families with annotations including genetic, functional and structural data. SwissProt is widely regarded as an excellent curated protein sequence database containing hundreds of thousands of non-redundant entries, which are annotated with evidence of experimental confirmation of structure, function, and post-translational modifications.[13]

In certain situations, like initial exploration of a sample's proteome, both the completeness of the repository-style databases and the accuracy of curated databases are desired. As such, UniProtKB sequence database provides the best of both worlds for a large number of organisms. For any organism, this database contains the corresponding SwissProt entries (curated) and TrEMBL entries (translated). Further efforts were made to reduce the redundancy of UniProtKB by combining homologous sequences and sub-fragments into a separate UniRef database. The UniProtKB and UniRef databases are recommended as the best choices to maximize the protein identifications gleamed during the initial explorations of a sample set's proteome.

Establishing a reference protein sequence database is always the first, and deterministic, step of any proteomics experiment. Hence, the desired end result of the experiment must always be considered when establishing an experiment's reference database. For example, if a researcher wishes to analyze cancer proteomics data to identify oncogenic mutations, fusions, or alternate protein isoforms, none of the existing public reference databases would suffice. This is because an organism's protein sequence database is derived from a reference genome, which in turn is built using only a handful of representative subjects. Hence, the resulting reference protein sequence database will not represent the sequence diversity that exists in individuals or populations or various pathological conditions (like cancer). For these experiments where a standard proteome does not suffice, a database of sample-specific protein sequences from RNA-seq transcriptomic data is collected from the same sample. This approach is explained in Chapter 16.

3.3 Overview of Shotgun Proteomics Method

The MS2 spectra that are generated from peptides are the basic, low-level, data in a shotgun proteomics experiment. However, native proteins assume higher order structures (tertiary and quaternary) and these intact proteins are not readily amenable for mass spectrometric analysis. Hence, the first step of proteomic analysis *via* MS/MS starts by denaturing the proteins. A variety of physiochemical methods exist for protein denaturation and all of them leave the protein backbone intact while disrupting the higher order structures. Denatured proteins are highly cross-linked because of intact cysteine–cysteine disulfide bridges. These bridges are reduced and alkylated, resulting in unfurling of the protein to its primary structure. At this juncture, a protein can contains hundreds to thousands of amino acids that are linearly arranged like pearls on a string. These primary protein sequences contain anywhere between tens of amino acids (like proinsulin) to a few thousand amino acids (like thyroglobulin or titin). The proteins are enzymatically digested, with

proteolytic enzymes like trypsin, chymotrypsin, or Endoproteinase GluC, to produce smaller and predictable peptides. Trypsin is universally preferred because it cleaves very specifically after the arginine and lysine amino acids, leaving the cleaved peptide with at least two proton accepting sites, one at each terminus. This makes the tryptic peptides more amenable for fragmentation analysis with a mass spectrometer.

A peptide analysis *via* tandem mass spectrometry moves through three stages. First, peptides need to be ionized in order for the mass spectrometer to analyze them. There are two widely used methods for ionizing biological macromolecules like peptides. The matrix assisted laser desorption ionization (MALDI) method mixes the peptide with a proton donor solid matrix, which is excited using a laser, resulting in the charging of the peptides. The electrospray ionization (ESI) method encapsulates peptides in liquid droplets and ionizes them by applying ultra-high-voltage, which results in columbic explosion-mediated charge transfer to the peptides. Next, charged peptides are introduced in to the mass spectrometer and the mass to charge (m/z) ratio of each peptide ion and its intensity are measured in a mass spectrum (MS). A variety of mass spectrometers have been developed for this purpose and their discussion is out of scope for this chapter. Finally, each peptide ion is selected by the mass spectrometer and fragmented using a variety of dissociation methods. The resulting "product ions" are analyzed by the mass spectrometer and their m/z ratios and intensities are recorded as a tandem mass spectrum (MS2 spectrum). Each MS2 spectrum contains all of the information that is needed to successfully identify the peptide.

3.4 Collision Induced Dissociation Fragments Peptides in Predictable Ways

It is not possible to understand the concept of peptide identification without understanding the concept of peptide fragmentation. Collision induced dissociation (CID) is one of the most basic and popular methods employed by the mass spectrometers in order to obtain fragment level information on the peptides. This method accelerates a peptide ion with a constant energy and impacts it against a wall of inert gas. During this collision, the kinetic energy of the peptide is converted into internal energy. This internal energy quickly localizes to the amide bonds that bind the amino acids of the peptide together and breaks them. This results in the generation of a pair of fragment ions that are recorded in the MS2 spectrum. For example, consider the peptide "GEMFILEKGEYPR" shown in Figure 3.1. A single CID event can break the bond between the first two amino acids and generate two fragments (b1 and y12) (Figure 3.1). B-ions are formed when protons transfer to the N-terminal side of the cleavage; y-ions are formed when the proton migrates to the C-terminal side of the cleavage. Thus, the b1 fragment will have the amino acid glycine and the y12 fragment will have the rest of amino acids EMFILEKGEYPR. Both of these ions will be analyzed by the

mass spectrometer and their *m*/*z* ratios and intensities will be recorded in the MS2 spectrum. Likewise, another CID event can break the bond between 4th and 5th amino acids (from N-terminal). This will generate a b4 and y9 ion pair (Figure 3.1). Because a peptide often has multiple copies and each copy can break at any of the amide bonds, the CID process generates a series of b- and y-masses for each peptide (Figure 3.1). These mass ladders have all the information that is necessary for identifying the peptide that produced the MS2 spectrum. For example, if we subtract the masses of b2 and b1 ions, we obtain the mass of glutamic acid (E), which is the second amino acid in the peptide (from N-terminal). Likewise, subtracting the mass of b5 from b4 mass would result in the mass of isoleucine (I), which is the 5th amino acid in the peptide backbone (from N-terminal). Hence, it is conceivable to *de novo* sequence the peptide from the MS2 spectrum given an amino acid mass table and a calculator, as described in the previous chapter. However, real life MS2 spectra are far more complex than the idealized spectra shown in Figure 3.1. Real life spectra often has noise peaks and missing fragment ions. The b-ions and y-ions can lose ammonia and water, which introduces mass shifts. Amino acids in peptides may also have expected and unexpected post-translational modifications that alter the fragmentation patterns and mass ladders. Finally, several alternative dissociation strategies like electron transfer dissociation (ETD), electron capture dissociation (ECD), and infrared multiphoton dissociation (IRMD). Each of these dissociation modalities produces different types of fragment ions. Hence, any peptide identification method must be aware of the fragmentation method type used.

3.5 Overview of Database Searching

The primary goal of the database search is to generate a list of most likely peptide matches given a tandem mass spectrum. Before the inception of high-throughput shotgun proteomics, peptide identification was a cumbersome manual process.[14] Mass spectrometrists would use *de novo* sequencing methods to derive a list of potential sequences for the peptide that would have generated the MS2 spectrum. These sequences were searched against the known proteome using BLAST.[15] The results must be thoroughly scrutinized to pick the most likely candidate. This type of manual analysis was error prone, time consuming, and not scalable. With the advent of modern, fast scanning, mass spectrometers, a typical shotgun proteomics data set can contain anywhere between thousands of MS2 spectra to millions, which rules out manual peptide identification and necessitates computerized peptide identification.

There are several database search software tools that facilitate the automated peptide identification. Table 3.2 lists some commercial and open-source database search engines that were available at the time of writing. Figure 3.2 shows the general inputs and the main steps of any database search engine. All search engines start with two main inputs: MS2 spectrum and list of protein sequences (Figure 3.2). The algorithm starts by deriving peptides

from the protein sequence database by following the specificity rules of the experimental protease that was used to digest the proteins. The list of peptides is filtered using the measured precursor mass of the peptide that generated the query spectrum. The resulting candidates contain the sequence of the peptide that generated the spectrum. Theoretical fragmentation spectra

Table 3.2 List of database search software. The listed software were separated into commercial and open-source categories in alphabetical order.

Commercial software	
Search engine	Company
Byonic™	Protein metrics
Mascot®*	Matrix science
PEAKS DB®*	Bioinformatics solutions
Phenyx™	GeneBio
ProteinPilot™	Applied biosystems
SEQUEST®	Thermo-fisher scientific
SPECTRUM MILL™	Agilent

Open source software	
Search engine	Website
Andromeda	http://medusa.biochem.mpg.de/maxquant_doku/
Comet	http://comet-ms.sourceforge.net/
Compass	http://coon.chem.wisc.edu/content/software
Greylag	http://greylag.org/
MS Amanda	http://ms.imp.ac.at/?goto=msamanda
MS-GFDB	http://proteomics.ucsd.edu/Software/MSGFDB/
MyriMatch	http://proteowizard.sourceforge.net/
pFind	http://pfind.ict.ac.cn/
X!Tandem	http://www.thegpm.org/tandem/

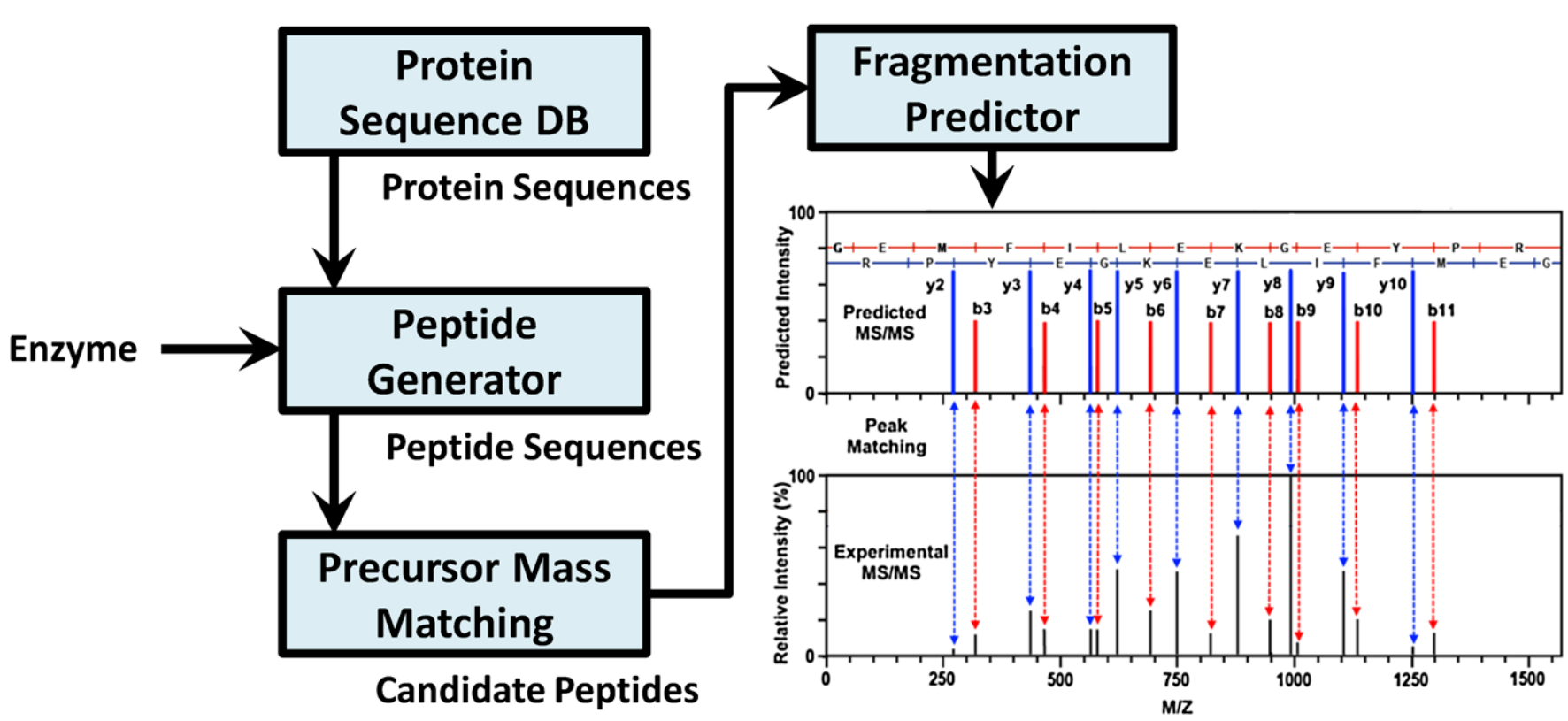

Figure 3.2 General steps of database search. The peptide that produces the best match between the predicted and experimental MS/MS is reported as the match.

are generated for each peptide by following the dissociation-specific fragmentation rules. For example, if CID was utilized to generate the MS spectrum, a list of b-ions and y-ions are generated for each candidate peptide by following the rules described in Section 3.4. The predicted peak list of each candidate is compared to the observed peak list of the experimental spectrum and a match score is computed. The match score captures the degree of similarity between the predicted and observed spectrum. Candidates are ranked and sorted based on their scores, and the best scoring candidate peptide is assigned to the spectrum.

The database search tools are highly automated and they require very minimal user intervention. A user need only spend a few minutes configuring the software with required inputs and search parameters. The software performs all computations with no additional user input and produces an output file containing the peptide matches for each input MS2 spectra and their respective match scores.

3.6 MyriMatch Database Search Engine

The database search workflow shown in Figure 3.2 is highly simplified and almost all traditional database search algorithms follow this workflow for peptide identification. However, there are several nuances in the peptide identification process and each database search software implements the workflow in a different fashion. We will describe this "peptide identification *via* database searching" process using the MyriMatch[8] database search engine as an example. The MyriMatch software requires three types of inputs: the raw MS/MS data, a protein sequence database, and a configuration file defining the parameter settings (Figure 3.3).

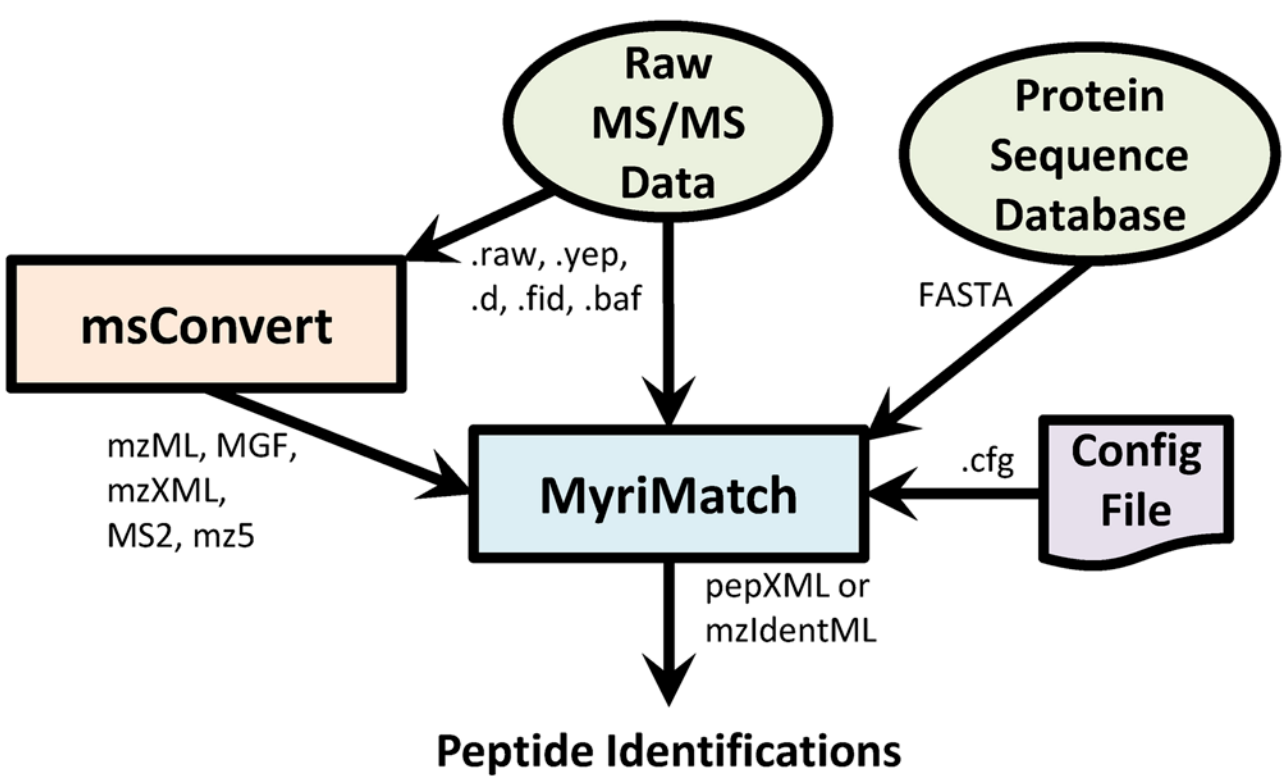

Figure 3.3 Inputs and output of MyriMatch software. The software uses text-based open source and binary-based native formats for reading raw MS/MS and writing protein identification results. The .cfg file is a text-based file to configure various parameters (like enzyme, precursor mass tolerance, *etc.*).

The software accepts the raw data in a variety of native file formats like RAW (produced by Thermo and Waters mass spectrometers), YEP, BAF, WIFF, FID, and .d (produced by Agilent and Bruker mass spectrometers). These instrument-specific file formats are binary in nature and the MS/MS data contained in these files are accessible only through special application program interfaces (APIs) supplied by the instrument manufacturers. This is problematic when sharing data across laboratories. To compensate for this, MyriMatch also accepts the MS/MS data in a variety of open source file formats like mzML[16] (see Chapter 11 for more information about this and other PSI standards), mzXML,[17] MGF (Matrix Science, UK), mz5,[18] and MS2.[19] These formats are text-based and they were developed to facilitate free exchange of data between laboratories. Users can utilize the msConvert[20,21] software of the ProteoWizard[20,21] library to convert the native raw data files into open source formats by following the protocol listed in ref. 22.

MyriMatch accepts the protein sequences in FASTA formatted files (Figure 3.3). For convenience, all configuration parameters are accepted in a single plain text formatted file. MyriMatch reads all the input files and starts by preprocessing the experimental MS2 spectra in order to prepare them for peptide-spectrum matching. Processed MS/MS are matched to peptide sequences derived from the protein sequence database. MyriMatch writes the resulting raw peptide identifications to either a pepXML[23] formatted file or an mzIdentML[24] formatted file.

3.6.1 Spectrum Preparation

Real life MS2 spectra always contain noise peaks mixed with fragment ion peaks that arise due to the peptide dissociation. These noise peaks are stochastic in nature and they interfere when the algorithm is matching the experimental spectrum with a predicted spectrum. This often results in producing spurious matches between fragments, which results in the production of false-positive peptide-spectrum matches (PSMs). Hence, the noise peaks must be removed from spectra prior to any peak matching. However, noise removal must be carried out with great caution because indiscriminate removal of peaks would get rid of true fragment ion peaks along with noise peaks, which decreases peptide identification sensitivity. At the same time, a relaxed approach to noise removal will retain noise peaks behind, which compromises the peptide spectrum match scoring. MyriMatch software uses a tunable total ion current (TIC) filter for noise filtering. The software takes an MS2 spectrum and computes its TIC. The user instructs the software to retain a proportion of the computed TIC (default is 98%). The software then sorts the peaks in decreasing order of their intensity and retains the minimum number of peaks that are required to meet the user specific TIC threshold. This tunable TIC filter is very different from that of X!Tandem (retains top 50 most intense peaks) or SEQUEST® (retains top 200 most intense peaks) and it was designed to scale with the number of peaks observed in a spectrum.

3.6.2 Peptide Harvesting from Database

MyriMatch software uses two key pieces of information when selecting peptides to compare against the experimental MS2 spectrum: digestion enzyme and peptide precursor mass. If trypsin was used for digestion, the software will only harvest tryptic peptides, which have an arginine or lysine residue at the C-terminus and an arginine or lysine residue immediately preceding its N-terminus. Knowledge has been built into the software to interpret the specificity of over 15 different enzymes used in proteomics studies and use that information to derive peptides from the protein sequence database. MyriMatch software can also be configured to ignore enzyme specificity and derive all possible peptides from the FASTA database. MyriMatch can also be instructed to clip the N-terminal methionine of the protein or ignore cleavage events on the N-terminal side of proline residues.

After the peptides are generated from the database, MyriMatch filters them to retain candidates whose calculated precursor masses match to that of the precursor mass that produced the MS2 spectrum, while accounting for the error in the measured precursor mass. The degree of the precursor mass tolerance (PMT) depends on the mass resolution of the mass spectrometer that was used to measure the peptide precursor masses and acquire the spectrum. Low resolution instruments like linear ion traps can accurately measure masses with an error of one part per thousand or higher. Hence, the PMT is often set to either 2 or 3 Daltons for these instruments. Medium resolution instruments like time-of-flight can measure masses with an error of one part per hundred thousand and the PMT for these types of instruments is often set to either 1 or 2 Daltons. Fourier transform-based mass spectrometers have the highest resolving power and these often can measure the precursor masses with an accuracy of one or sub-one part per million (ppm). The PMT for these instruments is typically set to 10 ppm. Setting the proper PMT is very important for generating accurate peptide spectrum matches. If the PMT is set too low for the experimental spectrum at hand, the correct peptide sequence derived from the database may not be compared to the experimental spectrum because its calculated mass does not match the measured precursor mass. On the other hand, if the PMT is set too wide, immaterial peptides get compared to the experimental spectrum, which increases the likelihood of matching the wrong peptide to the spectrum. Both of these scenarios will produce spurious peptide identifications, resulting in a higher false-positive rate and lower sensitivity and specificity of the peptide identification process.

3.6.3 Comparing Experimental MS/MS with Candidate Peptide Sequences

The peptide spectrum matching process has two main steps. First, the candidate peptide sequence derived from the FASTA database is converted into a theoretical spectrum, which is comprised of a list of m/z locations where

we would expect to see fragment ions in an experimental MS2 spectrum produced by the candidate peptide. Next, peaks in the theoretical and experimental spectrum are compared (allowing for a certain amount of mass error). Finally, a peptide spectrum match score is computed to quantify the degree of similarity between the predicted peaks and experimental peaks.

Over the years, many methods have been developed to generate predicted spectra from a peptide sequence. MyriMatch uses the most basic fragmentation prediction model. If the software was configured to assume that the experimental spectra were produced with CID fragmentation, MyriMatch software generates a list of b-ions and y-ions that would be produced by the given peptide sequence (as described in Section 3.4). If the experimental spectrum was generated by fragmenting either a singly charged or doubly charged peptide, the software will only predict singly charged fragments. If the experimental spectrum was generated from a higher charged peptide (≥3+), the software will predict multiple charge fragments. For instance, if the peptide was of 3+ charge state, the software will determine which side of the candidate sequence is more likely to take on the additional charge and leave the other side with single charge. One should note that the software does not predict the intensity of the theoretical fragment ions. This basic fragmentation model is used by almost all contemporary search engines, like X!Tandem, SEQUEST®, and Mascot®. It should be noted that more sophisticated fragmentation spectra predictors that can predict very accurate spectra, which mimic their experimental counterparts, do exist. For instance, methods encoded in the Mass Analyzer[25] software use a "mobile proton" kinetic model to generate predicted spectra that often mimic experimental MS2 spectra,[26] including the intensity of the fragments as well as not predicting fragments that are not likely to be observed. However, these sophisticated fragmentation models are very computationally intensive, which prevents them from being employed for routine use.

Once a predicted MS2 spectrum is generated for a candidate peptide sequence, it needs to be matched to the experimental spectrum and scored for its fit. Numerous scoring algorithms have been developed over the years for this purpose. The most rudimentary method is known as "shared peak count" (SPC), which enumerates the number of matches between the experimental and predicted spectra. This number can be normalized to the total number of predicted peaks to generate a "percent peaks matched" metric. However, this method fails to account for experimental peaks that do not match as well as the intensity of the matched and non-matched experimental peaks. MyriMatch uses a tiered intensity-based scoring system when matching experimental and predicted spectra. This method starts by segregating the filtered peaks in the experimental spectra into three intensity classes (high, medium, and low). These intensity classes differ in the number of peaks they contain such that the high intensity class holds the fewest and the medium and low intensity classes each hold double the number of peaks in the next most intense class. This tiered-class system allows rewarding peak matches based on their experimental intensity.

MyriMatch assumes that matching a peak from the high intense class is better than matching a peak from the low intense class and hence a high intense peak match should contribute more to the peptide score than a low intense peak match. Given a set of predicted peaks, the software marches through each predicted peak, computes whether the peak has been observed in the experimental spectrum, and if so, it records the intensity class of the matching experimental peak. A peptide spectrum match score is computed as the probability of observing the distribution of matched peaks' intensity classes by random chance. The software uses a multivariate hypergeometric (MVH) distribution to compute the score and reports the negative logarithm of the probability as match score.[8] This scoring method produces high scores for candidate peptides that predominantly match the most intense peaks when compared to candidate peptides that match low intensity peaks. MyriMatch stores, sorts (by decreasing order of the match score), and reports the top five scoring candidate sequences for each experimental spectrum.

The fragment ion mass tolerance (FMT) has a large effect on the accuracy of the peptide match scoring. The FMT parameter defines the maximum distance (in *m*/*z* units) a predicted peak may be from its *m*/*z* location in the experimental spectrum. Similar to PMT, if the FMT was set to a narrow window, many of the predicted peaks might not match the experimental peaks. If this parameter is set too wide, random peaks in the experimental spectrum will match to the predicted peaks. Both of these scenarios would generate spurious matches between the predicted and experimental spectra, resulting in false positive peptide spectrum matches. Similar to that of PMT, the range of the FMT depends on the resolving power of the mass spectrometer that was employed to record the experimental spectrum. FMT is typically set to 0.5 *m*/*z* units for low-resolution mass spectrometers, 50–75 ppm for medium resolution mass spectrometers, and 10 ppm for high-resolution mass spectrometers.

Unlike most database search engines, MyriMatch software computes an "mzFidelity" score that measures how well the predicted fragment ions match the experimental peaks in *m*/*z* space.[8] This scoring metric assumes that the fragment mass errors (*i.e. m*/*z* difference between matched predicted and experimental peaks) are normally distributed with $\mu = 0$ and $\sigma = (\text{FMT}/2)$. The *m*/*z* error space between 0 and $\pm 2 \times \sigma$ is divided into three "mzFidelity" classes: narrow, medium, and wide. The *m*/*z* width of the narrow mzFidelity class is half that of the medium mzFidelity class, which is half that of the wide mzFidelity class. The ratio of class size to the total size of all classes is computed and assigned as the probability of obtaining a random mass error of a particular mzFidelity class. This probability is 1/7 for the narrow class, 2/7 for the medium class, and 4/7 for the wide class. Next, MyriMatch attempts to find the closest matching experimental peak for each predicted peak. The mass errors between all matching peak pairs are computed and classified into the previously mentioned three mzFidelity classes. Predicted peaks that fail to match any experimental peaks (within the FMT) are classified into a

separate mzFidelity class X. The probability to obtain a mismatch (*i.e.* class X match) by random chance is computed by dividing the total *m*/*z* peak space of the spectrum with the FMT window. The software uses a multinomial distribution to compute the probability that the distribution of mass errors and peak mismatches observed during the peak matching step may have occurred by random chance.[8] The mzFidelity score rewards peptides whose predicted peaks match the experimental peaks with narrow mass error. The mzFidelity score of a peptide is also high if a majority of the predicted peaks for a candidate peptide sequence can be found in the experimental spectrum. MyriMatch software reports the negative logarithm of the resulting probability as the "mzFidelity" score of the peptide spectrum match.

Different search engines use different scoring methods as their primary scorers. The intensity-based MVH score is the primary scoring metric for MyriMatch. For SEQUEST®,[6] cross-correlation score (XCorr) is the primary scoring metric. For X!Tandem, the dot product-based hyperscore is the primary scoring metric.[27,28] If we set aside the idiosyncrasies of these different scoring systems, all of them attempt to better estimate the quality of a peptide spectrum match. Hence, the principal scoring systems employed by the search engines is their most distinguishing feature.

In summary, the MyriMatch database search algorithm contains the following principal steps: the software generates candidate peptide sequences from a protein sequence database using the enzyme employed for digestion and the precursor mass of the peptide that produced the experimental MS2 spectrum. Next, the software predicts the fragmentation spectrum for each peptide and compares it with the experimental spectrum. Peptide match scores are computed for each match, which evaluate the intensity of the matched peaks, fragment mass errors of peak matches, and the number of missed peak matches. Peptide matches of the experimental spectrum are ranked from best to worst using the decreasing order of the principal scoring metric.

3.7 Accounting for Post-Translational Modifications During Database Search

Protein post-translational modifications (PTMs) occur very commonly in biological samples. These PTMs are associated with several biological functions of the protein like folding, enzymatic activity, and protein degradation. PTMs introduce mass shifts in amino acids that host them. If a database search engine is not made aware of potential PTMs in the sample, the algorithm will use unmodified amino acid masses when calculating the predicted peaks of candidate peptides. This will produce predicted spectra whose peaks do not match that of experimental spectrum even though the correct candidate peptide sequence was used to generate the predicted peak list. Hence, database search engines must be instructed *a priori* about potential PTMs that may be present in the biological matrix.

Like most other database search tools, MyriMatch supports two types of PTMs: static and dynamic. A static modification is used to instruct the software that a particular amino acid is always modified with a certain PTM (*i.e.* the amino acid is not present in the sample in unmodified form). Carbamidomethylation of cysteine is an example static modification. This modification is a byproduct of the reduction and alkylation process of the cysteine–cysteine disulfide bridges. This PTM introduces a mass shift of +57 Daltons in all cysteine residues that are present in any protein, which increases the residue mass from 103 Daltons to 160 Daltons. When instructed properly, the MyriMatch software would use the modified mass of 160 Daltons for all cysteine residues that are encountered in the peptide spectrum matching process. If this PTM is present in the sample and MyriMatch was not made aware of this PTM, the software would generate incorrect predicted spectra for all cysteine containing peptides, which prevents their identification.

In contrast to static PTMs, dynamic PTMs are considered by the MyriMatch software as potential modifications to the database residues (*i.e.* the residue may be present in the biological matrix in either modified or unmodified form). When an amino acid with a dynamic modification is present in the sequence, MyriMatch software will match all possible combination of the modified and unmodified peptide forms against the experimental spectra. Identification of phosphorylation is a textbook example of a dynamic PTM search. Protein phosphorylation adds a HPO_3 moiety (mass shift of ~80 Daltons) to serine, threonine, and tyrosine residues. This PTM is only present on residues located in key catalytic sites of the protein. Hence, phosphorylation has to be specified as a dynamic modification of ~80 Daltons, occurring on the aforementioned amino acids. It should be noted that the dynamic PTMs exponentially increase the search space and search times. This is because, unlike static PTMs, MyriMatch needs to search multiple forms of a peptide that can harbor a dynamic PTM. Even though the dynamic PTMs can cause exponential increase in search times, MyriMatch algorithm doesn't impose upper limits on the number and variety of dynamic PTMs that are considered during the database search, which is in contrast to other search engines like SEQUEST®. In other search engines, static and dynamic PTMs are often referred to as fixed and variable modifications, respectively. More detailed information about dealing with PTMs can be found in Chapter 6.

3.8 Reporting of Database Search Peptide Identifications

Modern high-throughput shotgun proteomics methods can generate tens of thousands of MS2 spectra for each LC-MS/MS experiment. MyriMatch database search software searches all of these spectra in one run and stores, in memory, the top five best matching peptides for each spectrum. These peptide spectrum matches must be written to an output in machine readable format. This enables downstream peptide filtering (see Chapter 4) and

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<MzIdentML id="qe3_2016apr07_01_40t5_p16050-m.mzML /home/m105991/fasta/20130621-RefSeq59-Human-Cntms.fasta MyriMatch 2.1.138" creationDate=
"2016-04-14T11:54:42" version="1.1.0" xsi:schemaLocation="http://psidev.info/psi/pi/mzIdentML/1.1 ../schema/mzIdentML1.1.0.xsd" xmlns=
"http://psidev.info/psi/pi/mzIdentML/1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <cvList>
  <AnalysisSoftwareList>
  <SequenceCollection>
    <DBSequence id="DBSeq_gi|155030188|ref|NP_001094286.1|" accession="gi|155030188|ref|NP_001094286.1|" searchDatabase_ref="SDB"/>
    ...
    <Peptide id="PEP_1">
      <PeptideSequence>DAALCVLIDEMNER</PeptideSequence>
      <Modification location="5" residues="C" avgMassDelta="57.021464" monoisotopicMassDelta="57.021464">
        <cvParam cvRef="UNIMOD" accession="UNIMOD:4" name="Carbamidomethyl" value=""/>
      </Modification>
    </Peptide>
    ...
    <PeptideEvidence id="DBSeq_gi|155030188|ref|NP_001094286.1|_PEP_1" peptide_ref="PEP_1" dBSequence_ref="DBSeq_gi|155030188|ref|NP_001094286.1|"
     pre="K" post="P" isDecoy="false"/>
    ...
  </SequenceCollection>
  <AnalysisCollection>
    <SpectrumIdentification id="SI" spectrumIdentificationProtocol_ref="SIP" spectrumIdentificationList_ref="SIL" activityDate=
    "2016-04-14T11:54:42">
  </AnalysisCollection>
  <AnalysisProtocolCollection>
    <SpectrumIdentificationProtocol id="SIP" analysisSoftware_ref="AS">
  </AnalysisProtocolCollection>
  <DataCollection>
    <Inputs>
      <SearchDatabase id="SDB" name="20130621-RefSeq59-Human-Cntms.fasta" location="/home/m105991/fasta/20130621-RefSeq59-Human-Cntms.fasta">
      <SpectraData id="SD" name="qe3_2016apr07_01_40t5_p16050-m.mzML" location="qe3_2016apr07_01_40t5_p16050-m.mzML">
    </Inputs>
    <AnalysisData>
      <SpectrumIdentificationList id="SIL" numSequencesSearched="72580">
        <SpectrumIdentificationResult id="SIR_1" spectrumID="controllerType=0 controllerNumber=1 scan=1746" spectraData_ref="SD">
          <SpectrumIdentificationItem id="SIR_1_SII_1" rank="1" chargeState="4" peptide_ref="PEP_1" experimentalMassToCharge="412.950108334111"
          calculatedMassToCharge="412.94718695063" passThreshold="true" massTable_ref="MT">
            <PeptideEvidenceRef peptideEvidence_ref="DBSeq_gi|155030188|ref|NP_001094286.1|_PEP_1"/>
            <cvParam cvRef="MS" accession="MS:1001589" name="MyriMatch:MVH" value="28.103254830082"/>
            <cvParam cvRef="MS" accession="MS:1001590" name="MyriMatch:mzFidelity" value="32.201349376942"/>
            <userParam name="xcorr" value="1.0961943680341275"/>
          </SpectrumIdentificationItem>
          <SpectrumIdentificationItem id="SIR_1_SII_2" rank="2" chargeState="4" peptide_ref="PEP_2" experimentalMassToCharge="412.950108334111"
          calculatedMassToCharge="412.947304133055" passThreshold="true" massTable_ref="MT">
        </SpectrumIdentificationResult>
        ...
      </SpectrumIdentificationList>
    </AnalysisData>
  </DataCollection>
</MzIdentML>
```

Figure 3.4 Sample mzIdentML output of MyriMatch database search. Complete details of the mzIdentML format are listed in ref. 24. Excerpts of relevant elements from a sample database search are presented here.

protein inference (Chapter 5) based on the MyriMatch search results. MyriMatch is capable of producing the peptide identifications of each raw file in text-based pepXML or mzIdentML format. Figure 3.4 shows an excerpt of an mzIdentML file from a sample MyriMatch database search. A detailed description of mzIdentML format is out of scope here. Readers interested in understanding the mzIdentML XML schema are encouraged to consult Chapter 11 and the appropriate publication.[24] The file starts by describing the analysis software used for the peptide spectrum matching (XML element "AnalysisSoftwareList" in Figure 3.4). Next, the list of protein sequences and their corresponding candidate peptides are described using the "SequenceCollection" element (Figure 3.4). The "PeptideEvidence" element in the "SequenceCollection" links the generated candidate peptides to their corresponding parent protein sequences that are present in the FASTA file. After this, the complete configuration parameters of the search engine are described using the "AnalysisCollection" and "AnalysisProtocolCollection" elements. Next, each peptide spectrum match is described using the "SpectrumIdentificationItem" element (Figure 3.4). Each PSM description contains details of the precursor mass of the experimental MS2 spectrum, charge state of the precursor, rank of the peptide match, and associated scoring metrics (Figure 3.4). The top five ranking PSMs of each spectrum are

listed by the decreasing order of their MVH score (Figure 3.4) and grouped together using the "SpectrumIdentificationResult" element, which contains the mass spectrometer assigned native identifier of the MS/MS spectrum (Figure 3.4). This mzIdentML results file is used by the downstream processing and reporting of peptide and protein identifications.

3.9 Spectral Library Search Concept

Shotgun proteomics and database searching has been routinely employed by hundreds of laboratories across the world for proteome characterization studies. This wide-spread use of the technique has identified two critical weaknesses. First, repeated identification of the same peptides by protein sequence database searches is time consuming. Second, searching for many (>2) post-translational modifications by sequence database searching is impractical due to the exponential relationship between the search space and the number of PTMs in the search query. This has led to the development of peptide spectral library searching as a viable alternative to protein sequence database searching. This method starts by collating the existing peptide spectrum matches into a searchable peptide spectral library. Peptide MS2 spectra from a new experiment are identified by matching them against the MS2 spectra present in the library. This method has two key advantages. First, it efficiently identifies an experimental MS2 spectrum that has a representative MS2 spectrum in the library because it bypasses the time consuming process of harvesting candidate peptides from a protein sequence database and generating predicted MS2 spectra for correlating against the experimental MS2 spectrum. This improved efficiency has led to development of spectral libraries that are specialized for identifying PTMs.[29–31] Second, the library MS2 spectra are more representative of the experimental MS2 spectrum, when compared to the predicted MS2 spectra generated by the sequence database search for match scoring. For example, spectral library MS2 spectra reflect the actual fragmentation pattern observed for each peptide (including neutral losses and internal fragmentation) and their corresponding actual fragment ion intensities. In contrast, theoretical spectra predicted by the protein sequence database searches are often limited to b-ions and y-ions at fixed intensities. This allows the match scorers to use the fragment ion intensity information when evaluating the goodness of match between an experimental MS2 spectrum and a library MS2 spectrum. The additional intensity information increases the accuracy of the match scorers, decreases the false discovery rates, and increases the peptide identification sensitivity.[32,33] Despite these tremendous advantages of spectral library searching, the method suffers from a significant Achilles heel. For a spectral library search to successfully identify an MS2 spectrum, it needs a representative MS2 spectrum in the library. Hence, experimental peptides and their PTMs that are not represented by the spectral library will go unidentified. This shortcoming has held back the wide spread application of the spectral library search as a frontline tool in proteomics.

3.10 Peptide Spectral Libraries

The choice of spectral library is a more important step for a spectral library search when compared to the choice of protein sequence database to the database search. This is because all major protein sequence databases (shown in Table 3.1) are fairly complete and choosing any of them would still result in identification of a majority of peptides present in the sample. In contrast, peptide spectral libraries are relatively young, first suggested in the literature in 1998,[7] and not curated on a large scale until 2006.[32] Spectral library searching found its origins in analytical chemistry labs focused on small molecules,[34–36] with the earliest publication on computer searching of mass spectral data published in 1971.[37] However, not until the onset of more refined liquid chromatography systems and fast scanning mass spectrometers, could spectral library searching be considered possible for proteomics data. Another hurdle to the success of spectral libraries was the accrual and compilation of large numbers of high quality MS2 spectra to create a useful library. The recent explosion of proteomic data being generated and fed into public repositories coupled to the standardization of—and the capability to interconvert data—formats have now made useful and functional spectral libraries a reality.[38]

A proteomics experiment has a wide variety of parameters that determine the types of peptides and MS2 spectra observed. For instance, utilizing chymotrypsin as a proteolytic enzyme, instead of trypsin, would result in observing a different set of peptides. Blocking the reduced cysteine amino acid residues with iodoacetic acid, instead of iodoacetamide, would result in observing a different mass shift in cysteine-containing peptides. Employing ETD to fragment the peptides, instead of CID, would result in obtaining different types of fragment ions in the resulting MS2 spectra. Even when using CID fragmentation, employing different energies to fragment peptides will result in observing slightly different fragmentation patterns and ions in the resulting MS2 spectra. Because of this complex nature of proteomics experiments, it is very important that the spectral libraries contain MS2 spectra obtained from multiple sources, including sample preparation, instrumentation, and data acquisition. A thorough description of spectral library construction is out of scope for this chapter. However, an avid reader could consult the following references to understand the details of constructing a comprehensive peptide MS2 spectral library.[30,38–40]

In general, the process of spectral library building starts with a large collection of high quality peptide-spectrum matches that have been obtained by a protein sequence database search. These PSMs are imported into a raw spectral library, which is processed to denoise the MS2 spectra of the PSMs and merge redundant PSMs into a single entry. A "consensus MS2 spectrum" is generated for each non-redundant PSM entry by merging its replicate MS2 spectra. This MS2 spectral merging process amplifies the signal of the actual fragment ion peaks while suppressing the intensity of random noise peaks. This results in a high signal-to-noise ratio for the fragment ion peaks that

Table 3.3 Publicly available Peptide MS2 Spectral Libraries. Most comprehensive MS2 spectral libraries are listed.

Spectral library	Website	# of organisms	Human?	# of human peptides	# of human MS2 spectra
NIST peptide spectral library	http://chemdata.nist.gov/dokuwiki/doku.php?id=peptidew:start	10	Yes	718 720	2 144 310
Global proteome machine	ftp://ftp.thegpm.org/projects/xhunter/libs/	145	Yes	270 345	1 002 326

are present in the consensus MS2 spectrum for the PSM entry in the library. Spectral library search engines match these consensus MS2 spectra to the experimental MS2 spectra in order to identify the peptides present in the sample.

In contrast to protein sequence databases, there are very few comprehensive, publicly available, peptide MS2 spectral libraries (Table 3.3). The National Institute of Standards and Technology (NIST) issued a comprehensive peptide MS2 spectral library for a variety of organisms, including *Homo sapiens*. This library contains MS2 spectra derived from a variety of mass spectrometers (ion trap-based CID fragmentation and quadrupole-based HCD fragmentation). NIST provides these libraries in both ASCII text format (MSP file) and NIST MS Search binary format. It also includes peptides with a variety of sample handling and *in vivo* post-translational modifications (like carbamidomethylation of cysteine, oxidation of methionine, phosphorylation of serine, threonine, and tyrosine residues, *etc.*). The Global Proteome Machine (GPM) issues a number of spectral libraries covering a large number of organisms (Table 3.3). The GPM provides the annotated spectrum libraries (ASLs) in two different formats: text-based Mascot Generic Format (MGF) and binary-based X!Hunter[41] Library Format (HLF). The GPM libraries were created by using high confident PSMs (expectation values < 0.0001) derived from the X!Tandem[28] search engine configured to process a variety of public data sets obtained from the following sources: ProteomeXchange,[42] MASSIVE, PeptideAtlas,[43] ProteomicsDB,[44] and the Chorus Project. Datasets that are made available from large proteome characterization projects, like the Clinical Proteomic Tumor Analysis Consortium (CPTAC) and the Human Proteome Atlas, are also processed and included in the spectral library. One of the key differences between the GPM and NIST MS2 spectral libraries is that the GPM libraries store only the top 20 most intense fragment ion peaks for a particular MS2 spectrum. However, the NIST library attempts to capture all relevant fragment ions when representing the peptide with a consensus MS2 library spectrum. Both NIST and GPM libraries include MS2 spectra of both tryptic and non-tryptic peptides. However, it must be noted that trypsin

is the most frequently used enzyme in all proteomics experiments. Hence, a majority of the peptides that are present in both NIST and GPM spectral library are tryptic in nature. A spectral library search is not recommended when the proteomics experiment at hand is using a non-traditional enzyme for obtaining peptides.

3.11 Overview of Spectral Library Searching

The primary goal of spectral library searching is rapid identification of peptides given a set of tandem mass spectra. There are several open-source spectral library search software tools that facilitate rapid and automated peptide identification. Table 3.4 shows a list of open-source spectral library search engines that were available at the time of writing this chapter. Figure 3.5 shows the general inputs and the main steps of any spectral library search engine. All spectral library search engines start with two main inputs: MS2 spectrum and a peptide MS2 spectral library. The algorithm starts with an optional step of deriving potential peptide spectrum matches from the MS2 library by using the specificity rules of the experimental protease that was used to digest the proteins (Figure 3.5). The list of the library peptide spectrum matches are filtered using the measured precursor mass of the peptide that generated the experimental MS2 spectrum. The library MS2 spectra of the candidate peptides are correlated with the experimental MS2 spectrum. A match score is computed that captures the similarity between the fragment ions peaks present in the library MS2 spectrum and the experimental MS2 spectrum. Candidates are ranked and sorted based on their scores, and the best scoring candidate peptide (from the library) is assigned to the spectrum.

All spectral library search tools are highly automated and they require very minimal user intervention, as do database search engines. A user would configure the software with required inputs and search parameters. The software performs all computations and produces an output file containing the peptide spectrum matches for each input MS2 spectrum and their respective match scores.

Table 3.4 List of spectral library search software in alphabetical order.

Search engine	Website
BiblioSpec	https://skyline.gs.washington.edu/labkey/project/home/software/BiblioSpec/begin.view?
NIST[a] MSPepSearch	http://chemdata.nist.gov/dokuwiki/doku.php?id=peptidew:mspepsearch
Pepitome	http://proteowizard.sourceforge.net/
SpectraST	http://www.peptideatlas.org/spectrast/
Tremolo	http://proteomics.ucsd.edu/Software/Tremolo/
X!Hunter	http://thegpm.org/HUNTER/index.html

[a]NIST stands for the National Institutes of Standards and Technology.

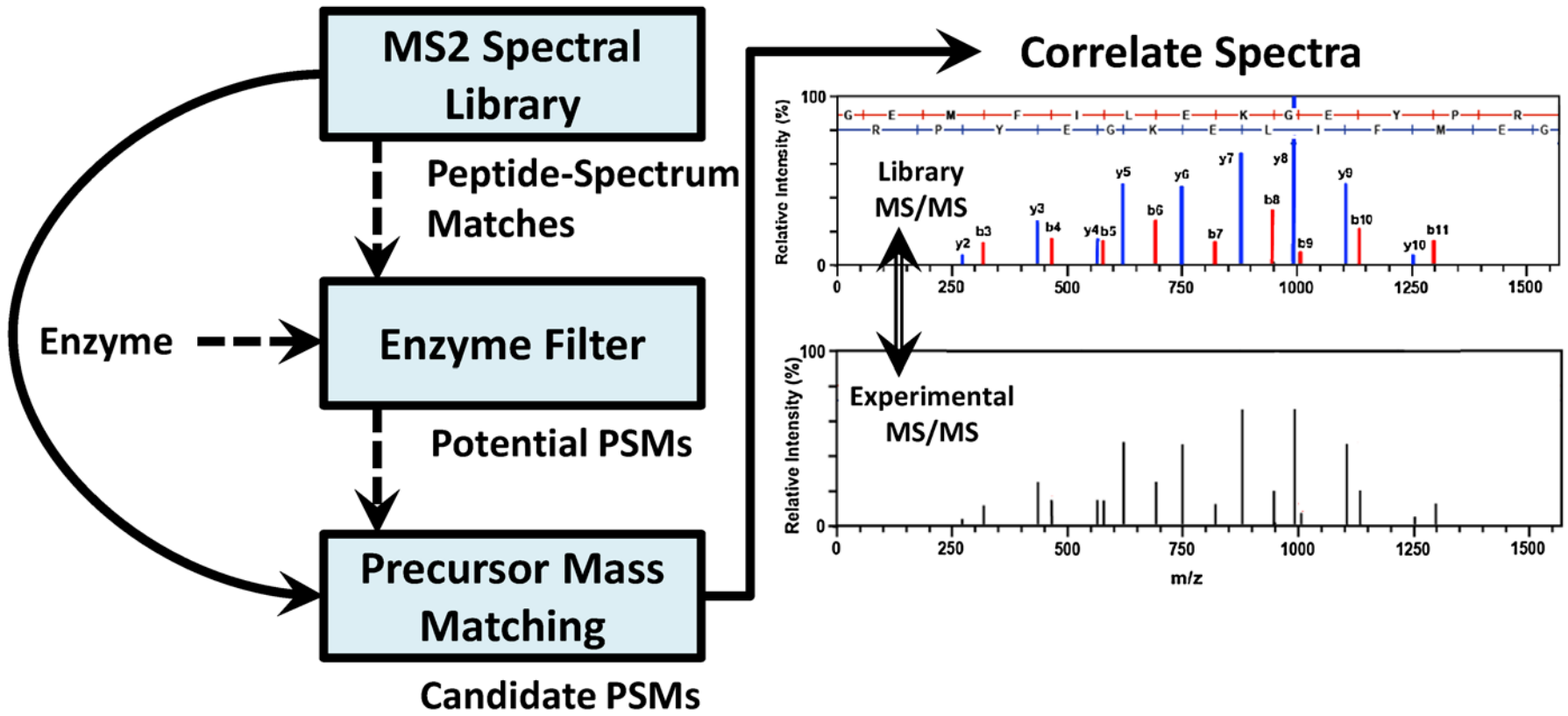

Figure 3.5 General steps of spectral library search. The peptide whose library MS2 spectrum best matches the experimental MS2 spectrum is reported as the match. Digestion enzyme filtering step is optional.

3.12 Pepitome Spectral Library Search Engine

The general spectral library search schema shown in Figure 3.5 has been oversimplified for didactic purposes. Each spectral library search engine shown in Table 3.4 uses different heuristics to filter the spectral library in order to obtain the candidate PSMs for comparison. Each search engine also implements spectrum–spectrum match scoring systems somewhat differently. We will describe the “peptide identification *via* spectral library” process using the Pepitome[45] spectral library search engine as an example. The Pepitome software requires four types of inputs: raw MS/MS data, a curated peptide MS2 spectral library, a protein sequence database that contains the peptide represented by the library, and a configuration file defining the parameter settings (Figure 3.6).

Like MyriMatch database search software, Pepitome spectral library search software accepts the raw MS2 data in a variety of binary-based native and text-based open source file formats (see second paragraph of Section 3.6 for details). The software accepts the spectral libraries in NIST’s MSP format or SpectraST’s .SPTXT format. Both of these formats are text-based and they contain all the information (like peptide sequence, potential PTMs, precursor mass, precursor charge, and fragment ion peak list) that is needed to successfully perform spectrum–spectrum matching. The software also needs a list of protein sequences, in FASTA format, whose peptides are represented by the spectral library. Pepitome software reads all the input files and starts preprocessing the experimental MS2 spectra in order to prepare them for spectrum–spectrum matching. Processed MS2 spectra are matched to the peptide spectra present in the library. Pepitome writes the resulting raw peptide identifications to a pepXML[23] formatted file.

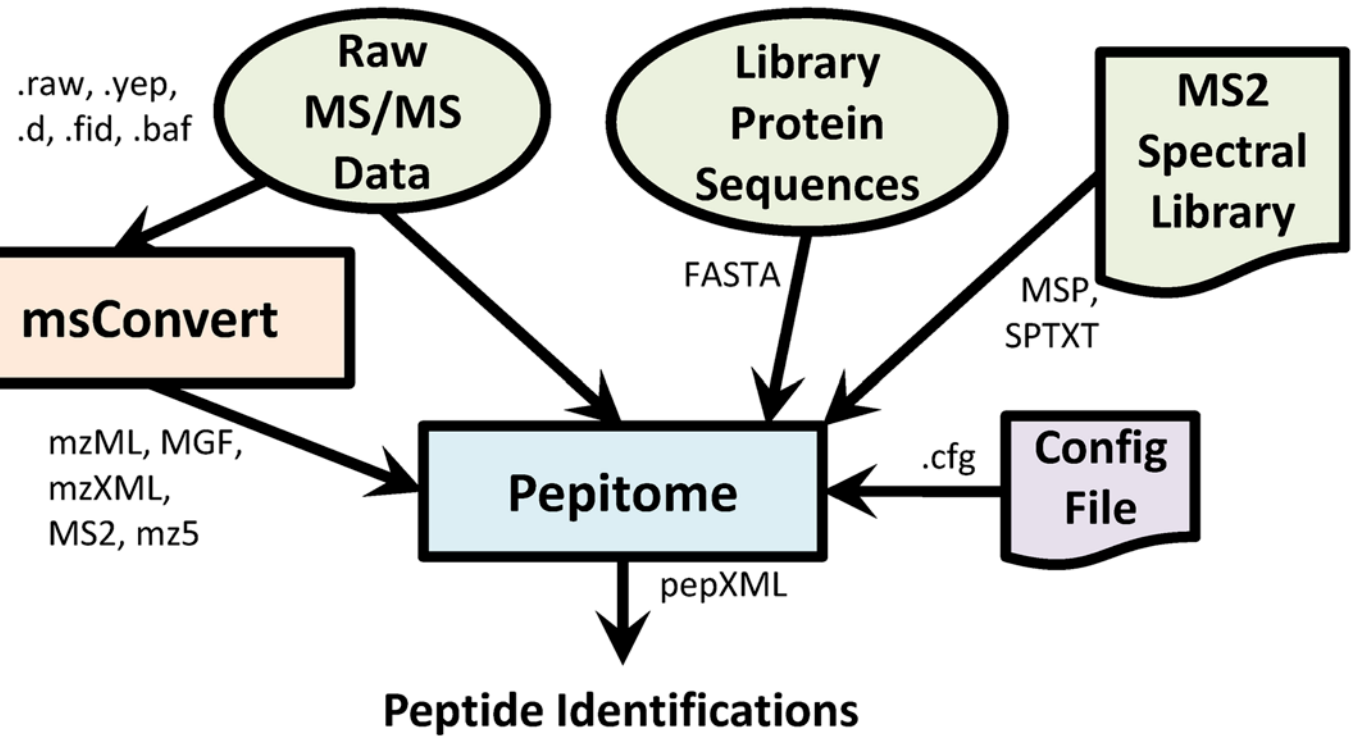

Figure 3.6 Input and output of Pepitome software. The software can read experimental MS/MS from either text-based open source formats or binary-based native formats. The FASTA file contains protein sequences whose peptides are in the spectral library. The .cfg is a text-based file to configure various search parameters (like enzyme, precursor mass tolerance, fragment mass tolerance, *etc.*).

3.12.1 Experimental MS2 Spectrum Preparation

Experimental MS2 spectra contain noise peaks mixed with true fragment ion peaks. These noise peaks are stochastic in nature and they are not represented in the library spectra (filtered during the "consensus MS2 spectrum" creation process). Hence, the noise peaks must be removed from the experimental spectrum in order to avoid spurious matches between experimental fragments and library fragments, which results in the production of false-positive spectrum–spectrum matches (SSMs). Pepitome uses either an adjustable TIC-based filter or a rigid peak count filter to remove noise peaks from experimental MS2 spectra. The TIC-based filter is described in the *spectrum preparation* subsection in Section 3.6. In contrast to the TIC-based filter, the peak count filter accepts only the N (user specified) most intense ions from the experimental MS2 spectrum. After preprocessing, the intensities of the remaining fragment ions in the spectrum are replaced by their ranks, where the most intense ion receives the lowest rank and the least intense ion receives the highest rank.

3.12.2 Library Spectrum Harvesting and Spectrum–Spectrum Matching

For each experimental MS2 spectrum, Pepitome loads into memory all library peptides that match the experimental precursor mass within a user-defined PMT window. The software applies identical MS2 spectrum preprocessing steps to the library spectra. Next, peak m/z positions in the library spectrum are matched to the peak m/z locations in the experimental MS/MS using a user-defined FMT window. If multiple library peaks match a single

experimental peak or *vice versa*, the peak pair with the lowest *m*/*z* error is considered as a match.

Pepitome computes three orthogonal scores to estimate the quality of a spectrum–spectrum match (SSM): a hypergeometric test (HGT), a Kendall τ statistic, and an evaluation of *m*/*z* fidelity. Given a pair of library and experimental MS2 spectra, the HGT score estimates the probability of obtaining more than the observed number of peak matches by random chance, which follows a hypergeometric distribution.[45] The Kendall τ score measures the correlation between the intensity ranks of matched peaks between the spectra. The raw Kendall τ score ranges from −1 (inverse correlation) to 1 (direct correlation) and it is converted into a probability of obtaining better than the observed intensity correlation by random chance, which is approximated[45] using a normal distribution. Like MyriMatch sequence database search engine, Pepitome uses the mzFidelity[46] score to estimate random probability of obtaining the observed mass errors between the *m*/*z* locations of the matched peaks. The software combines the HGT and Kendall τ scores, using Fisher's Method, into a single ranking score. All scores are transformed into negative logarithmic domain, and the combined score is used as a primary metric for sorting the library matches, with mzFidelity acting as a tie-breaker.

Different spectral library search engines use different scoring methods as their primary scorers. X!Hunter[41] produces expectation values from dot product scores, between experimental and library MS2 spectra, to infer statistical significance of the match. SpectraST derives a discriminant function that fuses the dot product, delta dot, and dot bias. Delta dot score attempts to quantify the significance of the top ranking match by computing the difference of the dot products of first and second best library match. A small delta dot score indicates that the experimental MS2 spectrum did not have enough information content to produce a distinguished library match. This could be due to either too few peaks, or too many noise peaks, or multiple peptides that are capable of producing similar MS2 spectra, like that of singly phosphorylated peptides with multiple sites next to each other. In all of these cases, the library search failed to detect a single, best, match. The dot bias score is a measure of how much of the dot product is "biased" toward a few dominating intense peaks.[32] SpectraST uses the discriminant score to rank the spectrum–spectrum matches for each experimental MS2 spectrum. Pepitome sidesteps the problems associated with dot product scoring by using rank-based correlation metrics.[45]

The precursor mass tolerance (PMT) and fragment ion mass tolerance (FMT) are the two most important tunable parameters for Pepitome spectral library search engine. As in database searching, the degree of PMT depends on the mass resolution of the mass spectrometer that was utilized to acquire the experimental MS2 spectra. Setting a narrow PMT would remove the true library PSMs from entering the spectrum–spectrum matching process. On the other hand, setting a wider tolerance will increase the number of candidate library PSMs that are being compared to the experimental MS2 spectra. In both of these scenarios this parameter would generate spurious

identifications and increase the false discovery rate of the peptide identification process. Pepitome uses the FMT when matching the peaks between experimental and library MS2 spectra. Similar to PMT, if the FMT was set to a narrow *m*/*z* window, many of the experimental fragment ion peaks will go unmatched to the fragment ion peaks in the library MS2 spectrum. If FMT was set to a wide *m*/*z*, random peaks in the experimental spectrum will match to the library peaks. Pepitome uses the same PMT and FMT windows as that of MyriMatch database search software. The guidelines for setting these parameters are described in greater detail in the subsections titled "*Peptide Harvesting from Database*" and "*Comparing Experimental MS/MS with Candidate Peptide Sequences*" in Section 3.6.

3.12.3 Results Reporting

Modern proteomics datasets contain thousands of MS2 spectra in each LC-MS/MS experiment. The pepitome spectral library search engine reads all experimental MS2 spectra in each raw data file, matches them against the library MS2 spectra, and stores the top five best matching library peptides for each spectrum. These (library) peptide spectrum matches are written to a machine readable, text-based, pepXML[23] formatted file. The PSMs are reported as if they are generated by a sequence database search (with peptide sequence of each match, protein sequence corresponding to each peptide match, search scores of the PSM). This enables the smooth processing of the library search results with downstream post-processing algorithms like IDPicker.[47,48]

In summary, the Pepitome spectral library search algorithm contains the following principal steps: the software generates candidate peptide-spectrum matches from a MS2 spectral library using the precursor mass of the peptide that produced the experimental MS2 spectrum. Next, the software compares the library MS2 spectra to the experimental MS2 spectrum. Spectrum–spectrum match scores are computed for each match, which evaluate the intensity of the matched peaks and fragment mass errors of peak matches. Library peptide matches to the experimental spectrum are ranked from best to worst using the decreasing order of the principal scoring metric.

3.13 Search Results Vary Between Various Database Search Engines and Different Peptide Identification Search Strategies

Table 3.2 lists a variety of database search engines that are available for use. All of these search engines have their own heuristics and approaches to peptide spectrum matching. However, a majority of the consequential differences between these search engines lay in the following five areas: spectrum preprocessing, peptide generation, candidate selection, predicted MS2 spectrum generation, and peptide spectrum match scoring. Because there is no single accepted "best" solution to generating peptide spectrum matches,

each software tool produces a slightly different sets of peptide spectrum matches even if they all start from the same set of spectra and FASTA protein sequence database. For example, in a recent study, only 73% of the reported peptide identifications were observed by both Mascot® and SEQUEST®, configured to search the same set of MS/MS spectra against the same protein sequence database.[49] Another comparative study between MyriMatch, X!Tandem, and SEQUEST® showed that 13% of peptides were identified with MyriMatch only, 10% were detected by X!Tandem only, 6% were only found by SEQUEST®, 21% were detected by at least two of the search engines, and 51% were detected by all three search engines.[8] This phenomenon of non-overlapping sets of peptide identifications between different search engines is routine. In fact, post-processing algorithms like Scaffold[50] often leverage this complementary nature of the database search engines to improve the coverage of peptide spectrum matches that can be obtained from a single dataset.

Spectral library and protein sequence database searches are two distinct paradigms for protein identification. As such, both of these methods can produce complementary peptide and protein identifications when working from the same input experimental MS2 spectra. For example, a spectral library search can identify peptides with unexpected PTMs, which are not detectable by a traditional database that requires upfront knowledge of all PTMs present in the sample. Also, a spectral library search attempts to match the experimental MS2 spectra to library MS2 spectra, which are derived using real life MS2 spectra. Hence, given a peptide sequence, a library MS2 spectrum represents the fragment intensities and types of observed fragment ions with greater fidelity than a predicted MS2 spectrum generated by a database search for the same peptide. This imparts higher peptide identification sensitivity to the library searches when compared to the database searches.[33,45] On the flip side, peptides and proteins that are not represented by the spectral library are only identifiable by a database search strategy. For example, tryptic digests often contain semitryptic peptides and peptides derived by *in vivo* protease activity. These non-traditional peptides are often not represented by canonical spectral libraries. A previous study showed that, when working from the same raw data file, only 61% of the peptides were identified by both database and spectral library search strategies, with 24% of the peptides identified only by the spectral library search, and 15% of the peptides detected only by the database search.[45] Hence, combining the search results of the complementary spectral library and protein sequence database searches by using post-processing algorithms like IDPicker[47,48] will yield a more comprehensive coverage of peptide identification than one can obtain from a single LC-MS/MS experiment.

3.14 Conclusion

Protein and peptide identification has been growing steadily from its inception in the early 1980s. A surge of new technological advances in mass spectrometry, electronics, and computing power has led to a stark increase in the amount of MS2 spectral data generated from each biological experiment. In parallel, proteomic researchers have been increasingly studying

a higher number of proteomes (obtained from large clinical trials or disease characterization cohorts) at a faster rate. This has resulted in an explosion of MS2 spectral data. Automated peptide identification pipelines that use sequence database searching and peptide spectral library searching have become the standard for any proteomics laboratory. For researchers, this affords rapid identification of peptides and proteins, while eliminating the complexity and error-proneness of manual methods of peptide identification.

Even though the database search method has been matured over more than two decades, the method still uses rudimentary predicted spectra while making peptide identifications. Spectral libraries have been developed to replace these predicted spectra with more realistic MS2 spectra during the peptide identification process. However, the spectral library search method fails to detect any peptide that is not represented in the library. The next step in the peptide identification process is to develop an efficient database search method that can predict a MS2 spectrum, from a peptide sequence, that looks more like an experimental MS2 spectrum. Likewise, the spectral library search method needs to keep expanding the peptides that are represented by a library. A hybrid search engine that would leverage the advantages of both database search engine and spectra library search engine might be an ideal goal to reach.

References

1. M. Mann, P. Hojrup and P. Roepstorff, Use of mass spectrometric molecular weight information to identify proteins in sequence databases, *Biol. Mass Spectrom.*, 1993, **22**(6), 338–345.
2. J. R. Yates 3rd, S. Speicher, P. R. Griffin and T. Hunkapiller, Peptide mass maps: a highly informative approach to protein identification, *Anal. Biochem.*, 1993, **214**(2), 397–408.
3. P. James, M. Quadroni, E. Carafoli and G. Gonnet, Protein identification by mass profile fingerprinting, *Biochem. Biophys. Res. Commun.*, 1993, **195**(1), 58–64.
4. W. J. Henzel, T. M. Billeci, J. T. Stults, S. C. Wong, C. Grimley and C. Watanabe, Identifying proteins from two-dimensional gels by molecular mass searching of peptide fragments in protein sequence databases, *Proc. Natl. Acad. Sci. U. S. A.*, 1993, **90**(11), 5011–5015.
5. D. J. Pappin, P. Hojrup and A. J. Bleasby, Rapid identification of proteins by peptide-mass fingerprinting, *Curr. Biol.*, 1993, **3**(6), 327–332.
6. J. K. Eng, A. L. McCormack and J. R. Yates, An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database, *J. Am. Soc. Mass Spectrom.*, 1994, **5**(11), 976–989.
7. J. R. Yates 3rd, S. F. Morgan, C. L. Gatlin, P. R. Griffin and J. K. Eng, Method to compare collision-induced dissociation spectra of peptides: potential for library searching and subtractive analysis, *Anal. Chem.*, 1998, **70**(17), 3557–3565.

8. D. L. Tabb, C. G. Fernando and M. C. Chambers, MyriMatch: highly accurate tandem mass spectral peptide identification by multivariate hypergeometric analysis, *J. Proteome Res.*, 2007, **6**(2), 654–661.
9. N. Nagaraj, J. R. Wisniewski, T. Geiger, J. Cox, M. Kircher, J. Kelso, S. Paabo and M. Mann, Deep proteome and transcriptome mapping of a human cancer cell line, *Mol. Syst. Biol.*, 2011, **7**, 548.
10. R. Apweiler, A. Bairoch and C. H. Wu, Protein sequence databases, *Curr. Opin. Chem. Biol.*, 2004, **8**(1), 76–80.
11. D. L. Wheeler, D. M. Church, S. Federhen, A. E. Lash, T. L. Madden, J. U. Pontius, G. D. Schuler, L. M. Schriml, E. Sequeira, T. A. Tatusova and L. Wagner, Database resources of the National Center for Biotechnology, *Nucleic Acids Res.*, 2003, **31**(1), 28–33.
12. K. D. Pruitt, T. Tatusova and D. R. Maglott, NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins, *Nucleic Acids Res.*, 2007, **35**(Database issue), D61–D65.
13. E. Gasteiger, E. Jung and A. Bairoch, SWISS-PROT: connecting biomolecular knowledge via a protein database, *Curr. Issues Mol. Biol.*, 2001, **3**(3), 47–55.
14. D. F. Hunt, J. R. Yates 3rd, J. Shabanowitz, S. Winston and C. R. Hauer, Protein sequencing by tandem mass spectrometry, *Proc. Natl. Acad. Sci. U. S. A.*, 1986, **83**(17), 6233–6237.
15. S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Res.*, 1997, **25**(17), 3389–3402.
16. L. Martens, M. Chambers, M. Sturm, D. Kessner, F. Levander, J. Shofstahl, W. H. Tang, A. Rompp, S. Neumann, A. D. Pizarro, L. Montecchi-Palazzi, N. Tasman, M. Coleman, F. Reisinger, P. Souda, H. Hermjakob, P. A. Binz and E. W. Deutsch, mzML–a community standard for mass spectrometry data, *Mol. Cell. Proteomics*, 2011, **10**(1), R110 000133.
17. P. G. Pedrioli, J. K. Eng, R. Hubley, M. Vogelzang, E. W. Deutsch, B. Raught, B. Pratt, E. Nilsson, R. H. Angeletti, R. Apweiler, K. Cheung, C. E. Costello, H. Hermjakob, S. Huang, R. K. Julian, E. Kapp, M. E. McComb, S. G. Oliver, G. Omenn, N. W. Paton, R. Simpson, R. Smith, C. F. Taylor, W. Zhu and R. Aebersold, A common open representation of mass spectrometry data and its application to proteomics research, *Nat. Biotechnol.*, 2004, **22**(11), 1459–1466.
18. M. Wilhelm, M. Kirchner, J. A. Steen and H. Steen, mz5: space- and time-efficient storage of mass spectrometry data sets, *Mol. Cell. Proteomics*, 2012, **11**(1), O111 011379.
19. W. H. McDonald, D. L. Tabb, R. G. Sadygov, M. J. MacCoss, J. Venable, J. Graumann, J. R. Johnson, D. Cociorva and J. R. Yates 3rd, MS1, MS2, and SQT-three unified, compact, and easily parsed file formats for the storage of shotgun proteomic spectra and identifications, *Rapid Commun. Mass Spectrom.*, 2004, **18**(18), 2162–2168.

20. D. Kessner, M. Chambers, R. Burke, D. Agus and P. Mallick, ProteoWizard: open source software for rapid proteomics tools development, *Bioinformatics*, 2008, **24**(21), 2534–2536.
21. M. C. Chambers, B. Maclean, R. Burke, D. Amodei, D. L. Ruderman, S. Neumann, L. Gatto, B. Fischer, B. Pratt, J. Egertson, K. Hoff, D. Kessner, N. Tasman, N. Shulman, B. Frewen, T. A. Baker, M. Y. Brusniak, C. Paulse, D. Creasy, L. Flashner, K. Kani, C. Moulding, S. L. Seymour, L. M. Nuwaysir, B. Lefebvre, F. Kuhlmann, J. Roark, P. Rainer, S. Detlev, T. Hemenway, A. Huhmer, J. Langridge, B. Connolly, T. Chadick, K. Holly, J. Eckels, E. W. Deutsch, R. L. Moritz, J. E. Katz, D. B. Agus, M. MacCoss, D. L. Tabb and P. Mallick, A cross-platform toolkit for mass spectrometry and proteomics, *Nat. Biotechnol.*, 2012, **30**(10), 918–920.
22. J. D. Holman, D. L. Tabb and P. Mallick, Employing ProteoWizard to Convert Raw Mass Spectrometry Data, *Curr. Protoc. in Bioinf.*, 2014, **46**, 13.24.1–13.24.9.
23. A. Keller, J. Eng, N. Zhang, X. J. Li and R. Aebersold, A uniform proteomics MS/MS analysis platform utilizing open XML file formats, *Mol. Syst. Biol.*, 2005, **1**, 2005 0017.
24. A. R. Jones, M. Eisenacher, G. Mayer, O. Kohlbacher, J. Siepen, S. J. Hubbard, J. N. Selley, B. C. Searle, J. Shofstahl, S. L. Seymour, R. Julian, P. A. Binz, E. W. Deutsch, H. Hermjakob, F. Reisinger, J. Griss, J. A. Vizcaino, M. Chambers, A. Pizarro and D. Creasy, The mzIdentML data standard for mass spectrometry-based proteomics results, *Mol. Cell. Proteomics*, 2012, **11**(7), M111 014381.
25. Z. Zhang, Prediction of low-energy collision-induced dissociation spectra of peptides, *Anal. Chem.*, 2004, **76**(14), 3908–3922.
26. Z. Zhang, Prediction of low-energy collision-induced dissociation spectra of peptides with three or more charges, *Anal. Chem.*, 2005, **77**(19), 6364–6373.
27. D. Fenyo and R. C. Beavis, A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes, *Anal. Chem.*, 2003, **75**(4), 768–774.
28. R. Craig and R. C. Beavis, TANDEM: matching proteins with tandem mass spectra, *Bioinformatics*, 2004, **20**(9), 1466–1467.
29. T. Srikumar, S. M. Jeram, H. Lam and B. Raught, A ubiquitin and ubiquitin-like protein spectral library, *Proteomics*, 2010, **10**(2), 337–342.
30. Y. Hu and H. Lam, Expanding tandem mass spectral libraries of phosphorylated peptides: advances and applications, *J. Proteome Res.*, 2013, **12**(12), 5971–5977.
31. D. K. Schweppe, J. D. Chavez, A. T. Navare, X. Wu, B. Ruiz, J. K. Eng, H. Lam and J. E. Bruce, Spectral Library Searching To Identify Cross-Linked Peptides, *J. Proteome Res.*, 2016, **15**(5), 1725–1731.
32. H. Lam, E. W. Deutsch, J. S. Eddes, J. K. Eng, N. King, S. E. Stein and R. Aebersold, Development and validation of a spectral library searching method for peptide identification from MS/MS, *Proteomics*, 2007, **7**(5), 655–667.

33. X. Zhang, Y. Li, W. Shao and H. Lam, Understanding the improved sensitivity of spectral library searching over sequence database searching in proteomics data analysis, *Proteomics*, 2011, **11**(6), 1075–1085.
34. S. E. Stein, Estimating probabilities of correct identification from results of mass spectral library searches, *J. Am. Soc. Mass Spectrom.*, 1994, **5**(4), 316–323.
35. S. E. Stein and D. R. Scott, Optimization and testing of mass spectral library search algorithms for compound identification, *J. Am. Soc. Mass Spectrom.*, 1994, **5**(9), 859–866.
36. S. E. Stein, Chemical substructure identification by mass spectral library searching, *J. Am. Soc. Mass Spectrom.*, 1995, **6**(8), 644–655.
37. L. W. McKeen and J. W. Taylor, Chemical information from computer-processed high resolution mass spectral data: determination of the fragmentation patterns of multifunctional compounds, *Anal. Chem.*, 1979, **51**(9), 1368–1374.
38. H. Lam, E. W. Deutsch, J. S. Eddes, J. K. Eng, S. E. Stein and R. Aebersold, Building consensus spectral libraries for peptide identification in proteomics, *Nat. Methods*, 2008, **5**(10), 873–875.
39. W. Shao and H. Lam, Denoising peptide tandem mass spectra for spectral libraries: a Bayesian approach, *J. Proteome Res.*, 2013, **12**(7), 3223–3232.
40. H. Lam, E. W. Deutsch and R. Aebersold, Artificial decoy spectral libraries for false discovery rate estimation in spectral library searching in proteomics, *J. Proteome Res.*, 2010, **9**(1), 605–610.
41. R. Craig, J. C. Cortens, D. Fenyo and R. C. Beavis, Using annotated peptide mass spectrum libraries for protein identification, *J. Proteome Res.*, 2006, **5**(8), 1843–1849.
42. H. Hermjakob and R. Apweiler, The Proteomics Identifications Database (PRIDE) and the ProteomExchange Consortium: making proteomics data accessible, *Expert Rev. Proteomics*, 2006, **3**(1), 1–3.
43. E. W. Deutsch, J. K. Eng, H. Zhang, N. L. King, A. I. Nesvizhskii, B. Lin, H. Lee, E. C. Yi, R. Ossola and R. Aebersold, Human Plasma PeptideAtlas, *Proteomics*, 2005, **5**(13), 3497–3500.
44. M. Wilhelm, J. Schlegl, H. Hahne, A. Moghaddas Gholami, M. Lieberenz, M. M. Savitski, E. Ziegler, L. Butzmann, S. Gessulat, H. Marx, T. Mathieson, S. Lemeer, K. Schnatbaum, U. Reimer, H. Wenschuh, M. Mollenhauer, J. Slotta-Huspenina, J. H. Boese, M. Bantscheff, A. Gerstmair, F. Faerber and B. Kuster, Mass-spectrometry-based draft of the human proteome, *Nature*, 2014, **509**(7502), 582–587.
45. S. Dasari, M. C. Chambers, M. A. Martinez, K. L. Carpenter, A. J. Ham, L. J. Vega-Montoto and D. L. Tabb, Pepitome: evaluating improved spectral library search for identification complementarity and quality assessment, *J. Proteome Res.*, 2012, **11**(3), 1686–1695.
46. S. Dasari, M. C. Chambers, R. J. Slebos, L. J. Zimmerman, A. J. Ham and D. L. Tabb, TagRecon: high-throughput mutation identification through sequence tagging, *J. Proteome Res.*, 2010, **9**(4), 1716–1726.

47. B. Zhang, M. C. Chambers and D. L. Tabb, Proteomic parsimony through bipartite graph analysis improves accuracy and transparency, *J. Proteome Res.*, 2007, **6**(9), 3549–3557.
48. Z. Q. Ma, S. Dasari, M. C. Chambers, M. D. Litton, S. M. Sobecki, L. J. Zimmerman, P. J. Halvey, B. Schilling, P. M. Drake, B. W. Gibson and D. L. Tabb, IDPicker 2.0: Improved protein assembly with high discrimination peptide identification filtering, *J. Proteome Res.*, 2009, **8**(8), 3872–3881.
49. J. A. Paulo, Practical and Efficient Searching in Proteomics: A Cross Engine Comparison, *Webmedcentral*, 2013, **4**(10), 1–15.
50. B. C. Searle, M. Turner and A. I. Nesvizhskii, Improving sensitivity by probabilistically combining results from multiple MS/MS search methodologies, *J. Proteome Res.*, 2008, **7**(1), 245–253.

CHAPTER 4

PSM Scoring and Validation

JAMES C. WRIGHT[a] AND JYOTI S. CHOUDHARY*[a]

[a]Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, UK
*E-mail: jc4@sanger.ac.uk

4.1 Introduction

Identification and quantification of peptides and proteins using mass spectrometry (MS) is a principal theme of proteomics research. Advances in mass spectrometry have enabled unprecedented depth and speed in the analysis of proteomes, providing qualitative and quantitative measurement of proteins and their modified forms. The current generation of instruments provide sensitive detection of thousands of peptides from proteolytic digested proteins in a single experiment. Accurate identification and quantification of peptides in a biological sample enable researchers to monitor changes in protein expression, interaction and modification as a reaction to disease and environmental changes. Proteomics is closely linked with other omics research areas, and is becoming a common tool for assisting in genome annotation and discovery of novel protein coding genes. The computational assignment of mass spectra to peptides and proteins is made possible by a growing selection of tools and methods. These can be distilled into three main approaches for assigning a peptide to spectrum match (PSM). The most commonly used method is sequence database searching which compares spectra to theoretical masses and ion series generated from a sequence database (see Chapter 3). *De novo* identification methods, covered in Chapter 2,

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

directly sequence amino acids by examining the delta masses between fragment peaks in the MS2 spectra.[1] Finally, spectral library searching identifies spectra by comparing an unknown experimental spectrum to a library of previously identified and curated known spectra.[2] The later method is gaining popularity especially for data independent acquisition (DIA) experiments like SWATH[3] and MSe (see Chapter 10). There are many different tools implementing each of these approaches, each using different scoring schemes, and having their own strengths and weaknesses. It is possible to combine multiple methods, such as using the faster spectral library and database search methods to quickly identify known peptides and spectra, and then applying a *de novo* approach to any remaining unidentified or low scoring spectra.[4]

The scoring and validation of peptide to spectrum matches (PSMs) is a persistent topic in proteomics.[5–9] In the early days of proteomic mass spectrometry it was common place to set a simple scoring threshold or implement an *ad hoc* filtering criteria as a measure of confidence in PSMs. Sequence database search algorithms implemented *e*-values and *p*-values along with PSM scores that extended filtering of PSMs using a fixed error probability representing the validity of individual identifications. The limitations of these approaches due to false positives arising from multiple testing became apparent with the high sampling rate of modern mass spectrometry and researchers moved towards assessing error in the whole dataset. The implementation of strategies based on false discovery rate (FDR) calculations usually derived from a target-decoy[10] approach has gained prominence for large-scale proteomics analysis. Building on this target-decoy approach there are now several approaches[10–15] to find the FDR in a set of results and a range of different statistical metrics, including *q*-values and posterior error probability (PEP)[16] that can be inferred from a sequence database search.

Another important consequence of these statistical approaches has been the ability to develop standardised scoring metrics that enable sophisticated data analysis. Most spectral identification algorithms report PSM scores correlating with accuracy of the identification, and these scores can be hard to interpret and compare, as they are generally not valid statistical measures of confidence. Development of elaborate spectral identification workflows to include multiple identification tools allows results from different programs to be merged, taking advantage of their different strengths and minimising weaknesses.[4,17–19] To merge these results a comparable statistical score is required and once merged the spectra will need to be assessed for overall consensus significance. There are, however, a variety of techniques and post-processing utilities that can convert raw PSM scores into meaningful statistical values and report identification false positive rates at a given score threshold. Several statistics commonly used in proteomics for assessing the significance of a PSM from its identification score include *p*-values, *e*-values, false discovery rate (FDR), *q*-values and posterior error probabilities (PEP). For sequence database searching, most of these statistics are commonly calculated using a target-decoy search methodology,[10,11] however, some recent

progress has been made using alternative methods to estimate PSM error such as the generating function.[20–22]

This chapter encompasses the scoring and assessment of database search results giving an overview of common methods and their advantages or disadvantages. Statistical concepts are presented in an accessible manner allowing researchers to adopt good statistical practice when assessing results of a proteomic mass spectrometry experiment. Examples of some freely available software are also presented to provide the tools needed to quickly obtain the required statistics.

4.2 Statistical Scores and What They Mean

The high throughput nature of modern proteomics and requirement for consistency places a reliance on identification scores that can be used to calculate an estimate of error. Such statistical significance measures make it possible to unify the interpretation of MS-based proteomic experiments. Statistical measures are used in all fields of scientific research, however, understanding the differences between these values and when to best apply them can be unclear. In proteomics, there are a range of approaches used, some measuring the confidence in a specific PSM identification whilst others report the rate or error across a dataset. PSM scores obtained from spectral identification software normally only represent the quality of the match between the theoretical spectra of candidate peptides against the query spectrum. This score can be associated with an expected rate of error and transformed into a more generic and comparable statistical measure. Table 4.1 summarises some of the statistical measures used when reporting proteomics results. This section will define and discuss the commonly used statistics that can be reported by a search algorithm or calculated from their results either manually or using post-processing software. The prominently used statistics in proteomics are defined here. Each of these statistics can be assessed in terms of their specificity and sensitivity (stringency).

Table 4.1 Commonly used statistical measures in proteomics.

Statistic	Abbreviation	Synonyms	Specificity	Stringency
p-Value	—	—	Score specific	Anticonservative
p-Value (Bonferroni corrected)	—	—	Score specific	Very conservative
Expect value	*e*-Value	Expectation score	Score specific	Very conservative
False Discovery Rate	FDR	Global FDR	Set specific	Optimal
q-Value	—	—	Set specific	Optimal
Posterior Error Probability	PEP	Local FDR	Score specific	Conservative

4.2.1 Statistical Probability *p*-Values and Multiple Testing

A widely used statistical significance measure is the *p*-value which represents the probability that an observation, or in the context of proteomics a PSM, could be incorrect. A low *p*-value therefore indicates that the probability of observing an incorrect PSM at this score is small. This *p*-value is directly linked to the PSM score and is implemented to assess individual identifications by modelling all possible candidate matching peptides. The *p*-value has a limitation when used for large datasets of PSMs, as are commonly generated in a shotgun proteomics experiment. The problem stems from the fact that in large datasets, with many PSMs each with their own *p*-values, a certain proportion of false positive low *p*-values will be observed simply by chance. For example, after searching a set of 50 000 spectra against a search database we might observe 10 000 PSMs each with a score associated with a *p*-value of 0.05 or less, representing the commonly used 95% confidence threshold applied to *p*-values. However, due to the high number of PSMs tested some of these *p*-values will be significant simply by chance. If the number of PSMs tested is multiplied by the *p*-value threshold we can estimate that 2500 incorrect false positive PSMs could be present in the results. To address this problem the concept of multiple testing correction can be applied. A simple but conservative form of multiple testing correction is the Bonferroni correction. The Bonferroni correction suggests that *p*-values should be adjusted by the number of tests performed. In the previous example the significance threshold for each PSM would need to be adjusted to a *p*-value of 0.000001 (0.05/50 000). Correcting for the number of query spectra and the number of candidate peptide sequences, leads to ultra conservative scoring thresholds and in practise would produce very few significant results at reasonable probability. Some search engines, such as Mascot™[23] and CRUX[24] report *p*-values by default and directly calculate PSM scores from the log of their *p*-values. These values are corrected for the number of candidate peptides compared to a spectrum but do not correct for the number of query spectra.

4.2.2 Expectation Scores

An alternative statistic is the *e*-value or expectation value, which describes the number of hits one could expect to see by chance at a particular score. The calculation used to derive an *e*-value is basically the reverse of the Bonferroni correction, the *p*-value is multiplied by the number of multiple tests performed. Continuing the previous example this would produce an *e*-value of 2500 (0.05 × 50 000). Hence if we are aiming for the same significance level the *e*-values will produce the same number of significant results as the Bonferroni corrected *p*-values. Several spectral identification programs, including Mascot,[23] X!tandem[25] and OMSSA[26] report *e*-values for their PSMs, however, these *e*-values again only take account of the number of candidate peptide sequences in the search database, and not the number of query spectra and

adjusting the *e*-value for a large dataset would be very conservative, greatly reducing the number of significant results.

4.2.3 False Discovery Rates

To avoid the multiple correction problems of *p*-values and *e*-values when assessing groups of identifications it has become standard practice in proteomics experiments to report a false discovery rate (FDR). The FDR can be defined as the expected proportion of incorrect observations in a dataset or alternatively the estimated fraction of a dataset that are false positive. In the case of proteomics datasets this represents the number of false positive PSMs above a given scoring threshold. Figure 4.1 displays the two different distributions of correct and incorrect PSMs and how they overlap. In a sample where the peptide make up is known such as a mix of purified proteins or synthetically generated peptides we can directly infer these distributions,

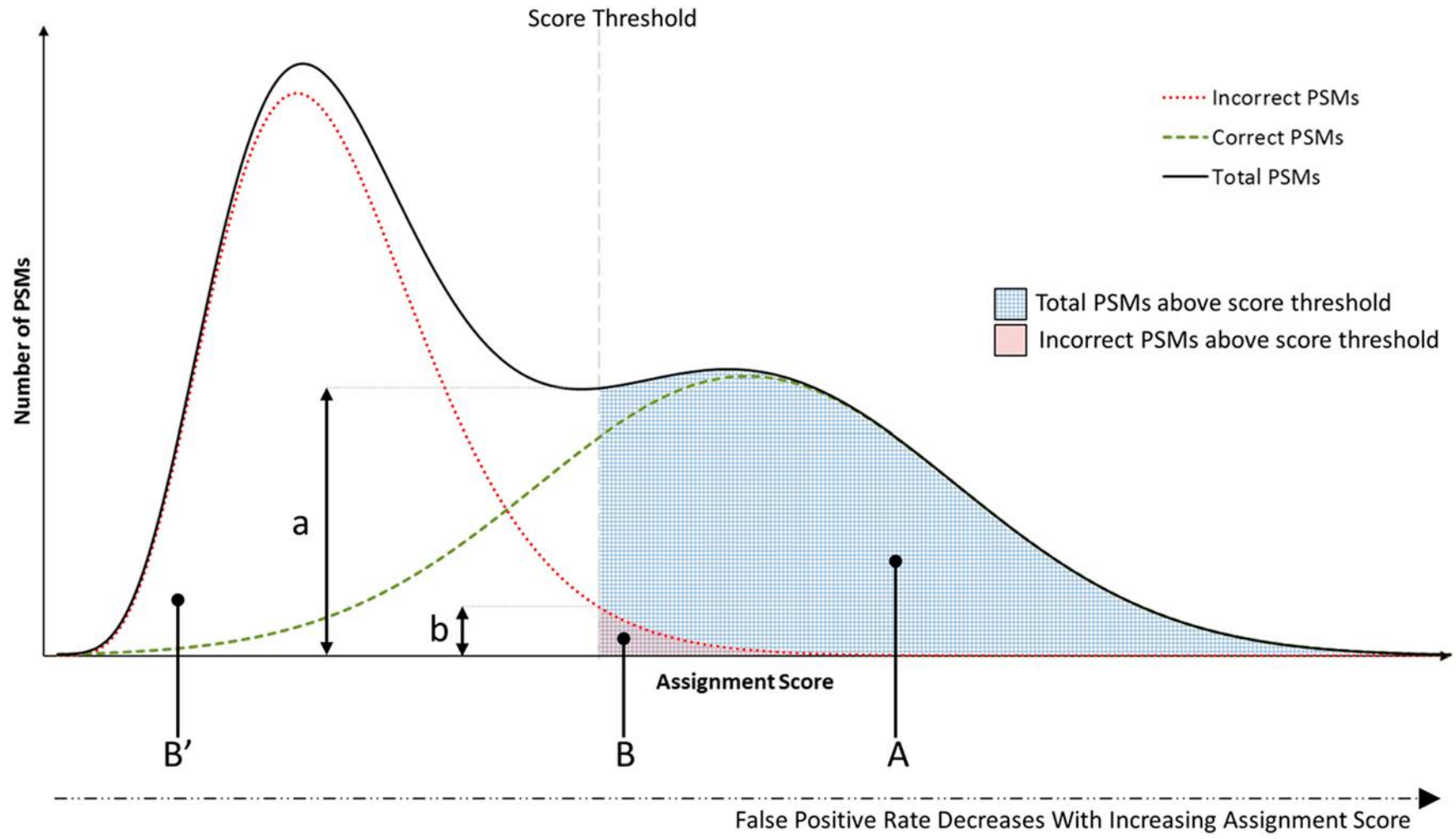

Figure 4.1 Model scoring distributions of correct and incorrect PSMs. The overall score distribution of PSMs in a proteomics experiment (solid curve) consists of a mixture of two underlying distributions, one usually modelled as a normal Gaussian distribution representing the correct PSMs (dashed curve) and one usually modelled as a Gamma distribution representing the incorrect PSMs (dotted curve). Above a chosen PSM score threshold (dashed vertical line), the crosshatched area, A, represents all PSMs that are accepted above that score, whilst the solid red filled area B represents the fraction of incorrectly identified PSMs above the cutoff. B together with B′ sums up all incorrect PSMs for the whole dataset. The false positive rate (FPR) and the false discovery rate (FDR) can be calculated from the numbers of PSMs in B, B′ and A. The posterior error probability (PEP) can be calculated from the height of the distributions at a given score threshold (b/a). The FDR can also be found from the average PEP of all PSMs above the scoring threshold.

however, the majority of the time this is not the case and the distributions have to be modelled, using methods such as a target-decoy search (see Section 4.2.7). FDR is calculated by dividing the number of incorrect PSMs above a defined score by the total PSMs above that same score. An example of this would be a set 10 000 PSMs above a selected scoring threshold where 500 of them are determined to be incorrect false positives; the resulting FDR would be 5% (500/10 000). Alternatively, we can use the FDR to find the scoring threshold we wish to use as trade-off between sensitivity and error. If every spectrum in a proteomic mass spectrometry experiment is matched to a peptide in a sequence database and the PSMs are then somehow separated into a list of correct and incorrect matches the scoring threshold at which a particular FDR is observed can be determined. If an FDR of 1% is set then this will give a list of PSMs where 99% of the matches are true and 1% false. If the FDR threshold is increased we can obtain a longer list of significant PSMs but with a larger proportion of them being incorrect. It should also be noted that the FDR is an estimate rather than an absolute measure. It could be assumed that an FDR of 1% for a set of 100 PSMs would mean that 99 PSMs are perfect identifications and 1 PSM is incorrect. The reality will be that the majority of the PSMs will be good, but not perfect, and a few will be weaker matches but not necessarily incorrect. The concept of the false discovery rate was originally proposed in 1995 by Benjamini and Hochberg,[27] and in 2002 Keller *et al.*[28] brought FDR estimation to proteomics using an empirical statistical model. The proposal of the target-decoy search approach by Elias and Gygi in 2007[10] improved FDR estimation and is now the standard method used in the majority of proteomics experiments. This approach uses a set of simulated decoy proteins in the spectral identification process to allow estimation of the scoring distribution of random false positive matches.

4.2.4 *q*-Values

When the FDR is calculated at each unique score throughout a dataset, it can fluctuate to have a situation where a less conservative scoring threshold produces a lower FDR. This demonstrates that the FDR is not a function of the underlying score. To resolve this for the field of genomics, in 2003 Storey *et al.* developed a *q*-value statistic,[29] which Kall *et al.*, later adapted for proteomics in 2008.[30] This value can be simply understood as the minimal FDR at which a PSM would be considered significant. The *q*-value statistic will transform the FDR so that increasing the scoring threshold will always lead to a lower FDR. FDR is a global value for a set of PSMs, whereas *q*-value is associated with an individual PSM. However, it is important to note that the *q*-value is still the same statistic as the FDR and is dependant of the dataset as a whole. Any changes to the set of identified PSMs due to changes in the initial search or filtering of the PSMs and spectra in post-processing will lead to a change in the *q*-values. For example after searching a large number of spectra against a sequence database and the results are ranked by their score, if it is determined that the top 100 PSMs contain 10 false positives,

then the PSM at position 100 will have a q-value of 0.1. If the same set of PSMs were searched again against a different sequence database, the original PSM might now be found in the top 50 PSMs with only 1 false positive in the 50, giving a q-value of 0.02. This PSM will have the same score in both searches and match the same peptide however; the q-value will have changed from 0.1 to 0.02. At the time of writing the q-value is now almost ubiquitous across proteomics as the main statistic reported for a set of results, having replaced standard FDR in most calculations, although it should be noted that q-values are still quite often reported as %FDR.

4.2.5 Posterior Error Probability

These FDR and q-value statistics reflect the error rate across a set of PSMs. However, experiments such as proteogenomic genome annotation, biomarker discovery, and targeted proteomics experiments, which focus on a particular set of proteins or peptides, require confidence at the level of individual PSM peptide identifications. To facilitate this, a posterior error probability (PEP) statistic is used to measure the significance of a single spectrum assignment with a specific score. The PEP represents the probability of an observed PSM being incorrect, thus a PSM with a PEP of 0.01 which can also be represented as 1%, means that there is a 99% chance that the peptide is present in the biological sample. In a set of results the PEPs will reflect the stronger and weaker confidence in PSM assignment. The PEP measures the probability of the error rate for a single PSM with a given score. Unlike the global FDR and q-value calculations, the PEP statistic requires knowledge of the underlying PSM scoring distributions of correct and incorrect identifications, and due to sparse scoring of the PSMs the PEP is usually estimated using a model. The height of the correct and incorrect PSM scoring distributions at any particular given score, are used to infer the PEP. This is visually demonstrated in Figure 4.1 showing how the PEP is specific to a PSM score and is obtained from the distributions. The figure also highlights the relationship to the FDR and how the sum of the PEPs above a selected score threshold divided by the number of PSMs above that threshold leads back to the overall global FDR for that set of PSMs. Posterior error probability is commonly estimated using machine learning approaches available in some of the post-search processing tools described in later sections.

4.2.6 Which Statistical Measure to Use and When

After assigning a set of mass spectra using one or more database search programs, criteria for assessing significance will need to be determined. The selection needs to be tailored to the aims of the experiment and any requirements of the downstream analysis. All these statistics are complementary and useful in different scenarios. Granholm *et al.*, provide a good discussion on when and where these statistics are best applied.[31]

In most experiments the goal is to identify as many PSMs, peptides and proteins as possible at a reasonable level of confidence using a valid statistical measure. The q-value is increasingly used as the metric of choice in proteomics; however, like FDRs q-values are a measure of error within a collection of results. They are suitable for examining groups of PSMs such as experiments that will go on to look at the enrichment of Gene Ontology[32] terms or overall changes in biological pathways and processes. However, when looking at individual spectra, such as might be done when assessing the expression of a specific biomarker protein under different conditions, analysing and localising biological post-translational modifications, or using PSMs for proteogenomic genome annotation, an appropriate spectral significance score should be chosen, such as the PEP. The PEP is a useful and complementary statistic measuring the probability of error in a single peptide to spectrum assignment. Additionally the PEP values can easily be converted into a score which can be used to merge and re-rank PSMs from multiple search programs into a consensus. Sometimes, experimental analysis can make use of both statistics using the FDR to initially filter a dataset and the PEP to assess specific peptide identifications of interest arising from the global analysis. Although the initial filtering may have set a q-value (FDR) threshold of 0.01 (1%) the PEP values for PSMs close to the threshold will likely be much larger than 0.01. There are cases where the distribution of PEP above a 1% q-value threshold is skewed, with a few PSMs close to the threshold demonstrating very high PEP indicating poor quality identifications and the remaining being good quality with very low PEPs. In this case these high PEP spectra are likely incorrect and should be filtered. To avoid this scenario a commonly used combination is to first apply a 0.01 q-value score threshold followed by a 0.05 PEP cut-off; this makes the final set of results slightly more conservative and can result in an FDR below 1%, however, it does remove the most likely incorrect PSMs from a dataset. By setting significance thresholds for both the FDR and the PEPs a balance can be found in the sensitivity and accuracy trade-off between false positives and false negatives.

The q-value and FDR describe the overall error in a dataset and can be used to assess the quality of the criteria used to filter and determine PSM significance, the PEP threshold can then be used to further refine this trade-off. The formulas for calculating FDR, PEP and FPR are as follows. See Figure 4.1 for a visual representation for the values used in these calculations.

$$\text{FDR} = \frac{B}{A} = \frac{\left(\sum_{i=1}^{A} \text{PEP}_i\right)}{A} \quad \text{PEP} = \frac{b}{a} \quad \text{FPR} = \frac{B}{\left(B' + B\right)}$$

where A represents the total number of PSMs above a chosen score threshold and B represents the number of incorrect PSMs above the same threshold. At any chosen score, a represents the height of the total PSM distribution and b represents the height of the incorrect PSM distribution. B' represents the remaining incorrect PSMs below a chosen scoring threshold.

The use of a PEP cut-off is more conservative than applying a q-value threshold, and when only concerned with examining the identified sample

proteome as a whole as is the case in most high-throughput shotgun mass spectrometry experiments then *q*-values are the best option.

An important benefit of *q*-values and FDR is that they are easily calculated for pretty much any dataset by using a non-parametric method based on the initial fitness scoring and *p*-values. However, PEP scores require additional analysis to verify the conditions required to calculate, such as the application of sophisticated Bayesian classification methodologies.[30,33]

4.2.7 Target Decoy Approaches for FDR Assessment

Spectral identification tools will always assign matching candidate peptides to a spectrum even if the best match is poor with nothing more than a matching precursor mass within the search tolerance. The search space used, which includes the initial sequences database searched and peptide modifications considered, will rarely be fully comprehensive of all the biological peptides present in a sample and random spectral assignments are inevitable. This can be further complicated by two peptides having similar mass and retention times leading to simultaneous fragmentation and mixed spectra. The initial score assigned to a PSM reflects the quality of the match assessing the level of signal to noise in the fragment spectra, the coverage of fragment ions present and the accuracy of fragment ion matches. Once we have a suitable score reported from a search we need to assess at what score threshold we consider a PSM to be confidently identified and non-random. This threshold is usually determined by the amount of error acceptable in the experiment which in itself is dependent on the size of the dataset and the size of the library or sequence database searched. The most common way of assessing the level of error and probability of random spectral identification at any given scoring threshold is to implement what is known as a target-decoy search.[10]

This target decoy methodology basically requires that the search space includes a proportional number of fabricated decoy peptides that would not be present in the protein sample. These decoy proteins and peptides can be generated in a variety of ways such as reversing the target protein sequences or shuffling the target peptide amino acid. The main principles of the decoy database are that the decoy proteins emulate the size and composition of the target proteins whilst not matching real proteins in the sample. There are two ways in which decoys are searched, either they are concatenated to the target database and searched in competition with the target peptides[10] or separate target and decoy searches are performed.[11] Both methods have their proponents and currently both are widely used valid methods as long as the correct filtering and formula is used to estimate the false positives. Figure 4.2 demonstrates the differences between concatenated and separated target-decoy approaches. When decoy peptides are matched to a spectrum the match is considered to be a random spurious result. Comparing the scores of these spurious matches to matches in the real or target search space allows the level of false positive identification to be determined and the FDR or

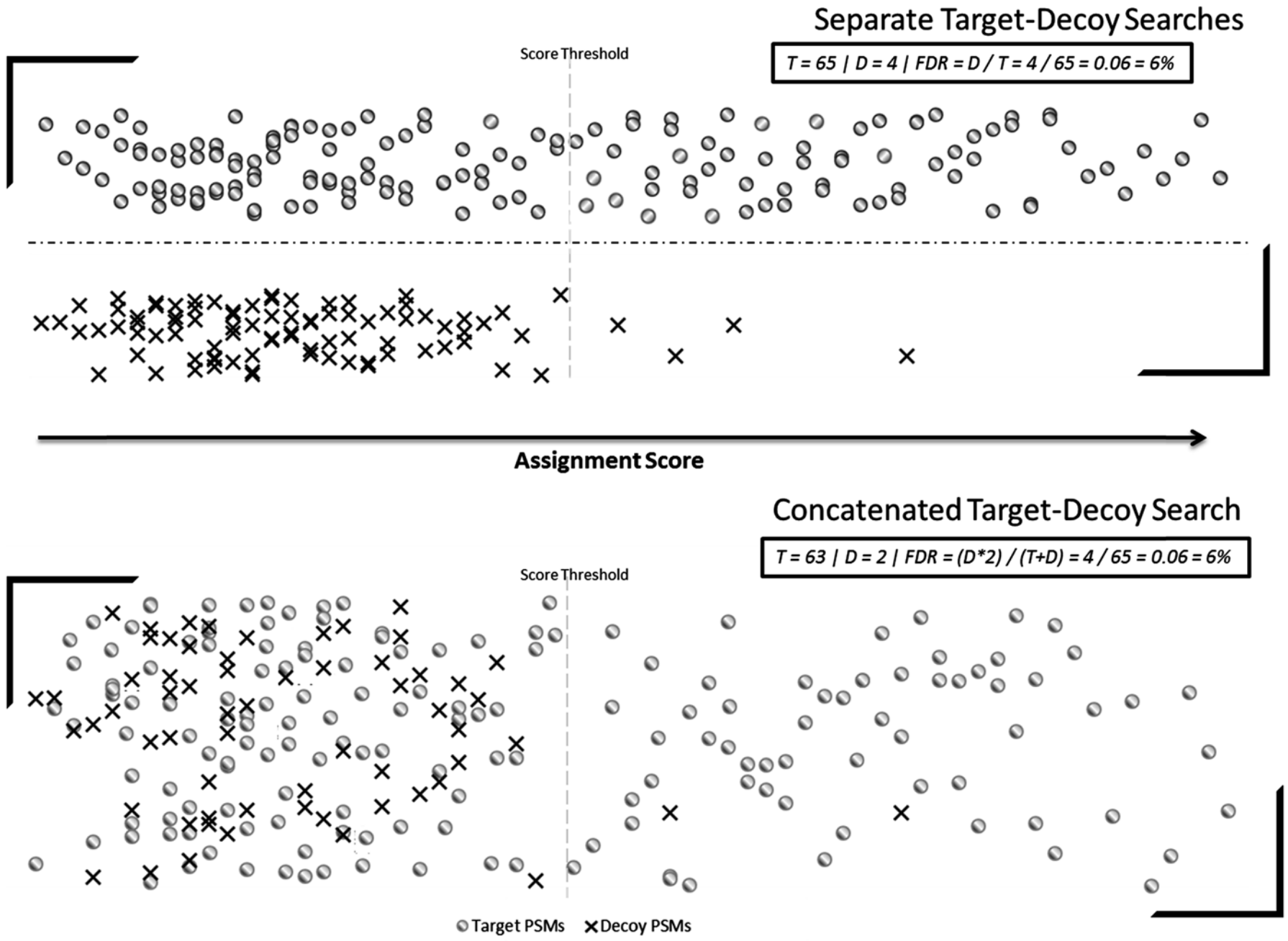

Figure 4.2 Target and decoy PSMs for separate and concatenated search strategies. The way in which the false discovery rate above a score threshold (grey dashed line) is calculated depends on the target-decoy search method used. The separate target and decoy search method will result in more decoy PSM identifications because there is no competition against the target database. This can cause a bias in the target-decoy model towards the decoys. However, this can be corrected using π_0 which represents the ratio of decoys to false positive targets. The concatenated search results in less decoy PSMs identified, however, for every decoy identified we must assume that a random match was also made in the target database, hence the number of decoys is doubled in the FDR calculation formula.

q-value to be calculated. Regardless of whether the target-decoy search was separate or concatenated, decoys must first be filtered to remove any redundancy to the target database – this should include isobaric peptides. Isobaric peptides have different amino acid sequences but have the same mass and fragmentation pattern in the mass spectrometer. An example of this would be peptides with leucine and isoleucine substitutions as these two amino acids have the same mass and elemental composition. More accurate PSM FDRs are possible when the true fraction of incorrect PSMs matching the target database is adjusted to reflect the true ratio of decoy PSMs to false positive target PSMs (π_0).[34] Depending on the approach taken the appropriate formula should be applied.

The standard FDR formula can be applied in the case of separate target and decoy searches as described by Kall *et al.*[11] This approach makes the assumption that the number of PSMs above a chosen scoring threshold in

the decoy database search (D) approximates the number of incorrect false positive PSMs above that same threshold in the target database search. This is divided by the total target PSMs above the score threshold (T).

$$\mathrm{FDR} = \frac{D}{T}$$

A correction to the standard FDR calculation improving the accuracy of the FDR calculation adjusts the formula by the fraction of true negatives, also known as π_0 or percentage incorrect targets (PIT).[11,34] When applied to a separate target-decoy search in proteomics this corrects for the fact that although all decoy PSMs are incorrect by design, not all target PSMs are correct and the ratio of decoy PSMs to false positive target PSMs is not balanced.

$$\mathrm{FDR} = \pi_0 \frac{D}{T}$$

An alternative method for estimating the FDR using separated target-decoy searches was proposed by Navarro *et al.*[13] This method suggests that decoys should not be used to blindly represent false positive target PSMs and that a competitive strategy should be applied to assess if a decoy PSM has a better match in the target search. This involves calculating the number of spectra with only a decoy PSM above the scoring threshold (Do), the number of spectra with both a decoy and target PSM above the scoring threshold with the decoy PSM having a higher score than the target PSM (Db), the number of spectra with both a decoy and target PSM above the scoring threshold with the target PSM having a higher score than the decoy PSM (Tb), and finally the number of spectra that have only a target PSM above the score threshold (To).

$$\mathrm{FDR} = \frac{(\mathrm{Do} + (2 \times \mathrm{Db}))}{(\mathrm{Tb} + \mathrm{To} + \mathrm{Db})}$$

For concatenated target-decoy searches as described by Elias and Gygi,[10] the competition between spectra either having a target or decoy best match changes the dynamic in the reference population. Hence, an adapted formula is used which assumes that random matches will equally distribute between the target and decoy sequences. The number of decoy PSMs above a chosen threshold (D) is considered to be equal to the number of incorrect false positive target PSMs, therefore the total number of incorrect PSMs in the dataset above a chosen score threshold is double the number of decoys. The total number of PSMs above the score threshold is the sum of target PSMs (T) and decoy PSMs (D).

$$\mathrm{FDR} = \frac{(2 \times D)}{(T + D)}$$

Cerqueira *et al.*[12,35] also proposed an alternative formula for the concatenated target-decoy search approach in which the decoys themselves are disregarded from the PSM population, so instead of doubling the number of decoys to reflect the target false positives, the decoys are removed from the total number of significant PSMs in the dataset.

$$\mathrm{FDR} = \frac{D}{(T - D)}$$

The best method to use when calculating FDR is still under discussion and all these methods incorporate some small bias which can affect sensitivity. An alternative more complex and involved method for estimating the FDR from a separate target-decoy search is the mixture-maximum procedure.[5]

The target-decoy approach is a simple non-parametric method for estimating the FDR in a set of PSMs. It has become commonplace in proteomics to adjust filtering criteria to achieve a fixed FDR. Additionally it has been suggested that improved discrimination of correct and incorrect PSMs can be achieved by binning the dataset prior to target-decoy analysis; these bins can be PSM properties such as the mass accuracy of the identification[14] or based on their source proteins as in the 2D FDR method.[15] Alternative methods for FDR estimation without using decoy peptides in the search have been proposed and implemented including the use of Bayesian nonparametric models,[36] generating functions,[20,37] retention time prediction[38] or leveraging PSMs that are not the best scoring match for a spectrum.[22] It should also be noted that the majority of these FDR estimation methods can also be applied to spectrum-to-spectrum matching and spectral library searching.[39–41]

4.3 Post-Search Validation Tools and Methods

It can be quite intimidating for non-informatician researchers to extract all the aforementioned statistics from their datasets, but many of the current generation of spectral database search programs and proteomics workflow suites provide these statistics directly in the output. There are also many free tools and utilities that can process proteomics data and provide statistical feedback.

Some tools simply estimate the error in a dataset using the previously mentioned formulae to allow filtering, other more complex post-processing tools make use of machine learning techniques to learn to discriminate correct and incorrect PSMs, rescoring and ranking PSMs based on a set of training features calculated for the target and decoy PSMs. This is of course dependant on the number of query spectra and the size of the search space being large enough to accurately model the correct and incorrect PSM distributions. Presented here are some useful tools for post-processing proteomics datasets to obtain statistical measures and in many cases improve the scoring discrimination between correct and incorrect identifications. We have focused mainly on standalone open-source and free to use software for calculating PSM level statistics, however there are always new tools and approaches on the horizon.

4.3.1 Qvality

One easy to use tool for transforming basic PSMs scores from identification software into *q*-values and PEPs is Qvality.[42] This tool uses non-parametric logistic regression from a separate target and decoy database searches to estimate the underlying scoring distributions of the correct and incorrect

PSMs. Qvality available from http://noble.gs.washington.edu/proj/qvality/ is a small, fully open source and stand-alone command-line application which can be readily applied to any set of PSM identifications regardless of the original program used to make the identifications. Running the utility simply requires two lists of PSM scores; one from a target sequence database search and another from a decoy sequence database search. Since no explicit assumptions of the type of the score distributions are made, the method was shown to be robust for many scoring systems and hence is not limited to one specific database search algorithm. This tool goes beyond the basic FDR target-decoy calculations and incorporates π_0 adjustments into the estimation.

4.3.2 PeptideProphet

A core part of the *Trans*-Proteomic Pipeline (TPP) workflow suite,[43] PeptideProphet[28,44,45] is an open source post-processing utility that uses statistical models to estimate the accuracy and to automatically validate PSMs assigned using a sequence database search program such as SEQUEST.[46] This was one of the first software tools in proteomic mass spectrometry to report PSM probabilities (P) akin to PEPs. This algorithm implements an expectation maximisation method to learn the distribution of correct and incorrect PSMs using various properties including the initial PSM score, accuracy of precursor ion mass, number of missed cleavages and enzymatic termini. PeptideProphet combines these multiple PSM properties into a fitness score which can be fitted against a Gaussian distribution for the true identifications and a gamma distribution for the false identifications. These fitted distributions allow the program to discriminate true and false identifications and compute robust probabilities for the likelihood that a PSM is correct. These calculated probabilities have been shown to provide a much higher sensitivity compared to using straightforward target-decoy FDR estimation. PeptideProphet can be downloaded as part of the larger TPP software package which incorporates a varied collection of useful tools for proteomic data analysis. These tools are available at http://tools.proteomecenter.org and http://peptideprophet.sourceforge.net/.

4.3.3 Percolator

Another widely used and popular post-processing utility is Percolator;[33,47] this program makes use of machine learning methods to improve the discrimination of true and false PSMs based on properties of the target and decoy matches and to infer *q*-values and PEPs. Percolator employs a large collection of customizable training features encompassing many PSM scoring properties. These features can include but are not limited to the initial PSM score or *p*-value, precursor mass accuracy, delta score (difference between the best and second best match for an individual spectrum), charge state, number of modifications, coverage of fragment ions, and accuracy of fragment ion assignments. The feature set used can be tailored to the search software used. However, features should be carefully chosen

to avoid introducing bias, and peptide or protein properties such as amino acid composition and sequence uniqueness should be avoided. Features such as peptide sequence composition or protein specificity, are not factors that directly determine the quality of a peptide to spectrum match. Using features such as these will influence the scoring of peptides in a manner which is not modelled by the target-decoy approach and hence will lead to bias in the training of the SVMs. Percolator takes these features from both the target and decoy PSMs in the search results and applies them to iteratively train a classifier. Initially, the most relevant single discriminating feature, usually the original PSM score, *p*-value or *e*-value, is selected and used to filter the results to a minimal FDR. This subset of target PSMs becomes the positive training set, all the identified decoy PSMs become the negative training set. These two sets of PSMs with their features are further divided into multiple cross validation sets and used to train multiple support vector machines (SVMs), a type of machine learning classifier. The SVM classifiers are then presented with all target PSMs and decoy PSMs in their entirety; the classifiers adjust the scores of the individual PSMs and the process is repeated, starting again by filtering the PSMs to minimise FDR. After several iterations the system converges resulting in a robust classifier with significantly better discrimination between correct and incorrect PSMs compared to the original PSM scores. This dynamic training and cross validation methodology allows the software to adapt to the unique properties of each dataset, optimising itself to the dataset quality, size, experimental setup and instrumentation used. Percolator is open source and is available from http://per-colator.com/, and various additional tools are bundled with the core utility to extract PSM feature tables from SEQUEST, X!Tandem[48,49] and MS-GFplus.[50,51] A java-based standalone tool MascotPercolator[52,53] is available for processing Mascot searches (http://www.sanger.ac.uk/science/tools/mascotpercolator). This has also been incorporated into Mascot-Server (Matrix Science) with some differences in the features calculated. An OMSSAPercolator[54] is also available from https://code.google.com/p/omssa-percolator/. Additionally Percolator has been incorporated into the popular ProteomeDiscoverer™ (ThermoScientific) and OpenMS[55] mass spectrometry data analysis and workflow platforms.

4.3.4 Mass Spectrometry Generating Function

An alternative method for estimating the PSM probabilities with or without the use of a target-decoy database search is MS-GF.[37] This tool makes use of the mathematical concept of the generating function and spectral energy/probability to discriminate correct and incorrect PSMs, and offers a powerful alternative to other post-processing tools. Originally implemented to support InsPect[56] and InsPect formatted results, the idea has been developed into a standalone search tool MS-GFplus which can be further combined with target-decoy searching and Percolator in a complementary fashion. MS-GF is available from https://bix-lab.ucsd.edu/display/CCMStools/MS-GF.

4.3.5 Nokoi

A recent method for decoy-free separation of correct and incorrect PSMs is Nokoi.[22] This post-processing tool developed for use with Mascot, uses a type of supervised machine learning called binary classification. In many search tools each spectrum can be assigned multiple candidate peptides reported in ranked order of score, usually, only the top rank peptide is used in the final results. Nokoi leverages these multiple hit ranks, instead of using spectra matching peptides in a decoy database to model false positives, it uses the low ranked hits to each spectrum. Although the pre-trained model does not have the adaptability of Percolator the tool is fast and the classification model is transferable to different types of dataset without retraining. This tool is available as a set of open source scripts from http://genesis.ugent.be/files/costore/Nokoi_utilities.zip.

4.3.6 PepDistiller

PepDistiller[57] validates Mascot Search results assessing their quality based on the number of tryptic termini and a refined FDR calculation using PIT. This tool works best with semi-tryptic search results. The freely available software and binaries are available from http://www.bprc.ac.cn/pepdistiller/.

4.3.7 Integrated Workflow and Pipeline Analysis Tools

There are several commonly used suites of software which allow researchers to build proteomic data analysis pipelines and workflows. These packages usually include a selection of tools for obtaining statistical metrics for reporting significant PSM identifications. The TPP as mentioned earlier makes use of the PeptideProphet tool for estimating PSM probabilities and also includes iProphet and ProteinProphet for statistical analysis at the peptide and protein level. OpenMS[55] (http://open-ms.sourceforge.net/) is an open source suite of tools and utilities for processing protein mass spectrometry data. These tools are easily accessed through The OpenMS Proteomics Pipeline (TOPP), which makes available several processing nodes for calculating FDR and estimating PEP scores from a mixture model, as well as a wrapper for Percolator. Additionally, the suite is able to build workflows that integrate and merge results from multiple spectral identification programs and calculate statistical measures such as q-values and PEP on the merged results.[19] A recent platform for analysis of proteomics data and targeted towards reanalysis of data using multiple search engines is PeptideShaker[58] available from http://compomics.github.io/projects/peptide-shaker.html. COMPASS[59] is a suite of analysis tools built around the OMSSA search software; this software provides various statistical metrics which can be used to filter and process results. This suite is available from https://github.com/dbaileychess/Compass. ProteinProspector[60] available

from http://prospector.ucsf.edu/prospector/mshome.htm also provides a variety of tools for mining proteomic mass spectrometry data *via* a web-based interface.

4.3.8 Developer Libraries

Developers wishing to build their own custom tools and software for proteomics analysis will find that there are quite a few libraries and developer resources which can be used to easily implement many of the most common proteomics statistical metrics. ProteoStats[61] is a software framework of open source tools and developer libraries. It offers a selection of statistical measures and methods for estimating errors in a dataset. It also incorporates some visualization utilities to compare and view results such as Venn plots and ROC curves. This software is available in Perl from https://sourceforge.net/projects/mssuite/files/ProteoStats/. MSstats[62] is an R-based package integrated into the Bioconductor project (http://www.bioconductor.org/). Although mainly targeted at protein quantification analysis it makes available various utilities for the statistical analysis of peptides and proteins.

4.4 Common Pitfalls and Problems in Statistical Analysis of Proteomics Data

Choosing a suitable statistical approach and significance filtering approach is always a trade-off between selectivity and sensitivity. The accuracy of the statistical metrics estimated for a dataset also depends on some assumptions and criteria being fulfilled. For all statistical approaches the size of the query dataset and the search identification space should be considered. For methods relying on a target-decoy approach the suitability of the decoys and any bias between the targets and decoys needs to be eliminated and may even be an unsuitable approach for certain types of experiment. This next section will discuss and highlight areas for further consideration when choosing a valid statistical approach to processing proteomics data.

4.4.1 Target-Decoy Peptide Assumptions

Many of the current tools used to establish statistical metrics in a proteomics experiment are dependent on the target-decoy methodology. The creation of the decoy database is therefore a key step in obtaining accurate statistics. It is difficult to completely rule out any particular peptide sequence not being present in the analyte sample; this means that decoys and hence false positives can quite often be overestimated. A comprehensive target database is important in minimising this, and should represent a complete proteome for the sample of interest plus any possible contaminants. Note that incomplete or heavily filtered target sequence databases increase the number of high quality spectra without a corresponding target sequence and also increase

the chances of false decoys with high scores. Although in certain circumstances filtering of a very large target database is acceptable and can improve sensitivity.[63] Another noteworthy factor to consider is the redundancy between the target and decoy databases, using the standard reversed target database method for decoy generation leads to an average peptide redundancy of around 5% between the two databases before considering isobaric peptides. Although it is easy to filter these out at the point of identification, and most tools will do this automatically, it does change the assumed equal proportions of target peptides to decoy peptides. Hence, it is best to address this when creating the decoy database. Most of these redundant peptides will be short sequences hence increasing the minimum peptide length required can also reduce this issue to some extent. Isobaric peptides between the target and decoy databases further increase the redundancy, and are a bit more insidious and difficult to avoid. These peptides will have different peptide sequences but have the same precursor mass as a target peptide and also produce the same or very similar MS/MS fragment ion spectrum. Another factor increasing error in the target-decoy model are unexpected peptides not represented in the target search space, and these can be from amino acid substitutions, incorrect sequence annotation in the search database or the existence of sequence variants such as those arising from single nucleotide polymorphisms in the sample donor(s). These peptides although reported as decoys will in fact be real true positive identifications.

4.4.2 Peptide Modifications

The chemical and biological modification of amino acids in peptides greatly increases the complexity and size of the search space to be interrogated. Techniques such as error tolerant or blind modification searches allow spectra to match non-enzyme specific peptides that can be shifted by any known and sometimes unknown protein modification or amino acid substitution mass. This is quite often done in a two-step approach that reduces the search space to proteins identified by spectra matching unmodified peptides in an initial search. This approach allows the discovery of unexpected modifications and sequence variations, however, this also increases the risk of high scoring false positive identifications. This is problematic for a standard target-decoy approach, firstly because of the restricted sequence database used (see next section), secondly because decoys will have higher scores due to the more tolerant peptide assignments, and thirdly the lack of enzyme specificity and allowed amino acid substitutions will increase the redundancy between target and decoy databases. This also causes problems for decoy free approaches as the increased error rate changes the distribution of correct and incorrect PSMs. Even in a standard database search the numbers and types of variable modifications allowed can impact FDR calculations. Searching a large number of variable modifications vastly increases the search space and comes with the same problems inherent in searching a very large sequence database as discussed in the next section. The other problem

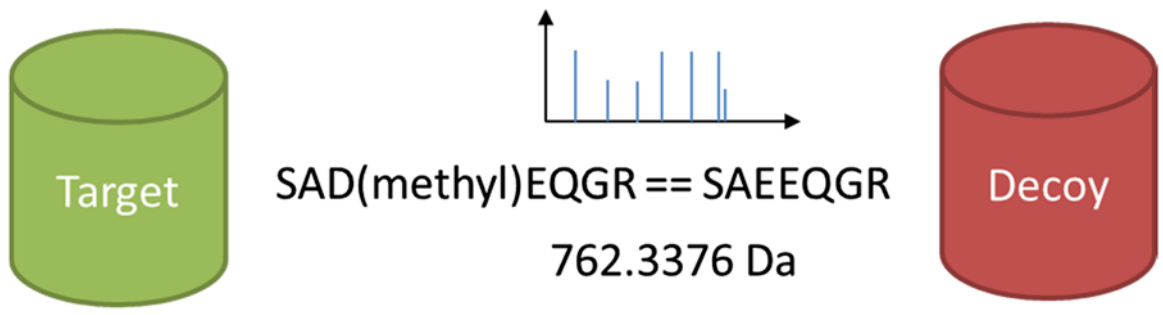

Figure 4.3 Modified peptides increases isobaric redundancy in target-decoy search. A single or a set of multiple amino acid modifications can alter the mass of a peptide by the equivalent of substituting one or more amino acids. This can increase redundancy between the target and decoy sequence databases and reduce the accuracy of FDR calculations. If the modifications occur at the same site as the equivalent substitution this leads to the theoretical fragment ions also having the same masses, this problem is further compounded by the fact that many spectra do not have complete ion series and the modification site is not always easily localised.

arising from variable modification is that certain modifications, such as methylation,[64] generate mass shifts in spectra that are similar to changes in the amino acid composition of a peptide (Figure 4.3). This can lead to an increased number of isobaric peptides in the target database and increasing the ambiguity in the multiple hit ranks of a spectrum. This also increases the redundancy between target and decoy search space which leads to less accurate FDR estimations using a target-decoy approach.

4.4.3 Search Space Size

Another important factor in the reliability of derived statistical measures in a proteomics experiment is both the size of the search space and the number of spectra searched. Most of the methods for finding FDR and PEP rely on fitting or learning the distributions of correct and incorrect identifications, which in turn rely on there being enough data points for this model to be accurate. A small number of spectra or a very restricted search database will lead to poor estimation of the true error due to low numbers of target and decoy data points. This effect is also encountered when using tight precursor match tolerances; in this case the tight tolerance reduces the number of candidate peptides available to match to a spectrum increasing the *p*-value of poor quality spectra.[65] Although the mass spectrometry instruments provide high resolution spectra the calibration of the detector can fluctuate during the course of an experiment. These problems can be somewhat mitigated by searching with slightly wider mass tolerances and by combining multiple datasets, for example boosting the number of spectra searched by merging multiple experiments or combining a small set of target protein sequences with a larger proteome sized sequence database. Problems also start to arise when the search database becomes very large or a large number of different experiments are combined; in this case the issues with an inaccurate target-decoy model become exaggerated with the possibility of high scoring false decoys increasing.

4.4.4 Distinct Peptides and Proteins

Everything discussed so far has focussed on the error rates associated with the identification of spectra as PSMs. This is usually the initial starting point for the analysis and significance filtering of a proteomics data set. The real aim in most experiments is to establish the presence of real peptides and proteins in a biological sample. However, it should be noted that after applying thresholds and filters to create a set of PSMs with a low error rate, such as a 1% FDR, when examining the error rate in terms of distinct peptides in the sample the FDR is amplified as shown in Figure 4.4. Due to the variable abundances of peptides and their level of detection in the mass spectrometer the number of PSMs is not linked to the number of distinct peptides in the sample, so to maintain the same level of error further statistical analysis is required. At the protein level the calculation of FDR becomes further complicated by the non-random mapping of peptides to proteins and the ambiguity of protein inference from shared peptides. Although a variety of software tools exist[66–69] to assist in protein level statistical analysis, several publications have highlighted how protein level FDR can be problematic[70–72] (see also Chapter 5).

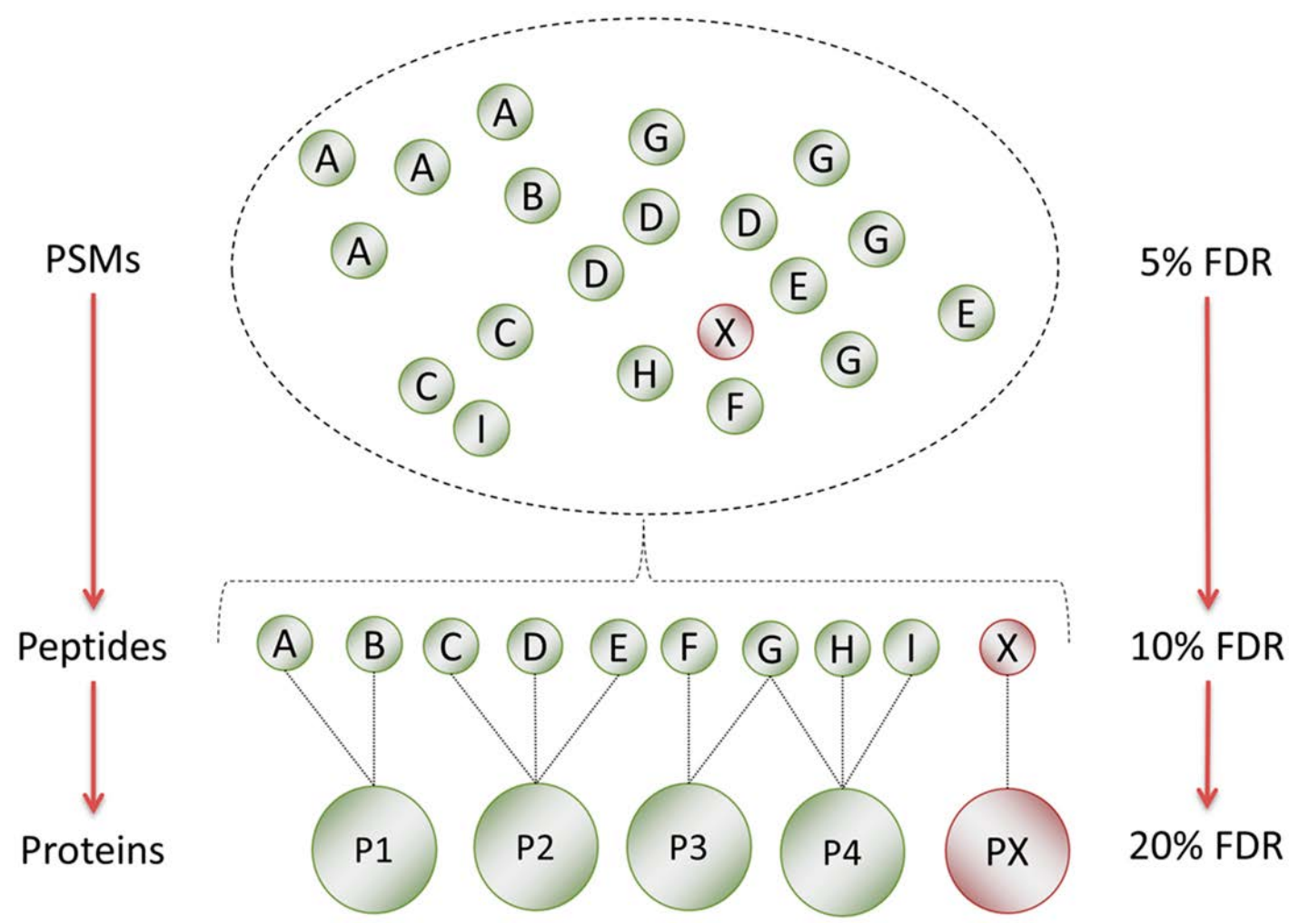

Figure 4.4 FDR Increase from PSM to peptide and protein level. An FDR calculated at the PSM level will significantly increase when the dataset is reduced to a set of distinct peptide sequences, and again as these peptides are inferred into a set of proteins. Hence, FDR must be recalculated at each level of analysis. In this example there is 1 incorrect PSM (labelled X) in a set of 20 – when this set is collapsed to a set of distinct peptides there is still 1 incorrect peptide but due to multiple identification of the same peptide sequences this is now in a set of 10 identifications, effectively doubling the FDR. These peptides can then be mapped back to a set of proteins, some uniquely and some ambiguously. In this example this results in 5 proteins, however, there is still one incorrect identification, doubling the FDR once again.

4.5 Conclusion and Future Trends

Robust statistical analysis of peptide spectral matches in protein mass spectrometry experiments is an essential step in data processing, with many scientific journals requiring that error rates be reported along with presented results. The community is moving towards FDR, *q*-values and PEP being reported as standard for any published experiment. To this effect there is now a plethora of search programs, post-processing utilities and pipeline software that support researchers in calculating and reporting statistical metrics for their results. The underlying assumption for any statistical analysis of proteomic mass spectrometry data is that the underlying scoring function that compares theoretical fragmentation of peptides from a sequence database to experimental spectra is good at discriminating good quality peptide to spectrum matches. Some of the post-processing tools discussed here build upon the initial PSM score improving the discriminatory power by learning, from a set of assignment properties, the differences between real correct identifications and random incorrect matches. As mass spectrometry technology continues to advance improving resolutions, ion separation and depth of analysis, additional parameters in these scoring functions will boost identifications and the confidence in correct identifications. Currently target-decoy searching remains the most widely used approach to measure error in PSM assignment, however, recent studies are pushing back against the limitations of this method and novel non-decoy algorithms for obtaining statistical metrics such as representative *p*-values without the need for anti-conservative multiple testing corrections applied to PSMs are on the rise.[73] The field of proteomic mass spectrometry continues to develop and new technologies are continually being made available to researchers. One such technology is the use of data independent acquisition (DIA) using SWATH[3] or MSe approaches and we can expect to see cultivation of tailored tools and statistical scoring methods leveraging these new approaches. These approaches and the increasing abundance of identified spectra are making spectral library searching a powerful contender to the dominance of sequence database searching. Many of the statistics and FDR methods used in sequence database searching are equally applicable to spectrum–spectrum matching with increased sensitivity and selectivity.[39,40,74]

References

1. A. Devabhaktuni and J. E. Elias, Application of *de novo* sequencing to large-scale complex proteomics datasets, *J. Proteome Res.*, 2016, **15**(3), 732–742.
2. X. Zhang, *et al.*, Understanding the improved sensitivity of spectral library searching over sequence database searching in proteomics data analysis, *Proteomics*, 2011, **11**(6), 1075–1085.
3. L. C. Gillet, *et al.*, Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: a new concept for consistent and accurate proteome analysis, *Mol. Cell. Proteomics*, 2012, **11**(6), O111.016717.

4. H. Thomas and A. Shevchenko, Simplified validation of borderline hits of database searches, *Proteomics*, 2008, **8**(20), 4173–4177.
5. U. Keich, A. Kertesz-Farkas and W. S. Noble, Improved False Discovery Rate Estimation Procedure for Shotgun Proteomics, *J. Proteome Res.*, 2015, **14**(8), 3148–3161.
6. K. Jeong, S. Kim and N. Bandeira, False discovery rates in spectral identification, *BMC Bioinf.*, 2012, **13**(suppl. 16), S2.
7. M. Vaudel, *et al.*, Peptide identification quality control, *Proteomics*, 2011, **11**(10), 2105–2114.
8. N. Gupta, *et al.*, Target-decoy approach and false discovery rate: when things may go wrong, *J. Am. Soc. Mass Spectrom.*, 2011, **22**(7), 1111–1120.
9. M. Fitzgibbon, Q. Li and M. McIntosh, Modes of inference for evaluating the confidence of peptide identifications, *J. Proteome Res.*, 2008, **7**(1), 35–39.
10. J. E. Elias and S. P. Gygi, Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry, *Nat. Methods*, 2007, **4**(3), 207–214.
11. L. Kall, *et al.*, Assigning significance to peptides identified by tandem mass spectrometry using decoy databases, *J. Proteome Res.*, 2008, **7**(1), 29–34.
12. F. R. Cerqueira, *et al.*, MUDE: a new approach for optimizing sensitivity in the target-decoy search strategy for large-scale peptide/protein identification, *J. Proteome Res.*, 2010, **9**(5), 2265–2277.
13. P. Navarro and J. Vazquez, A refined method to calculate false discovery rates for peptide identification using decoy databases, *J. Proteome Res.*, 2009, **8**(4), 1792–1796.
14. J. W. Joo, *et al.*, Target-Decoy with Mass Binning: a simple and effective validation method for shotgun proteomics using high resolution mass spectrometry, *J. Proteome Res.*, 2010, **9**(2), 1150–1156.
15. M. W. Bern and Y. J. Kil, Two-dimensional target decoy strategy for shotgun proteomics, *J. Proteome Res.*, 2011, **10**(12), 5296–5301.
16. L. Kall, *et al.*, Posterior error probabilities and false discovery rates: two sides of the same coin, *J. Proteome Res.*, 2008, **7**(1), 40–44.
17. W. Yu, *et al.*, Maximizing the sensitivity and reliability of peptide identification in large-scale proteomic experiments by harnessing multiple search engines, *Proteomics*, 2010, **10**(6), 1172–1189.
18. D. C. Wedge, *et al.*, FDRAnalysis: a tool for the integrated analysis of tandem mass spectrometry identification results from multiple search engines, *J. Proteome Res.*, 2011, **10**(4), 2088–2094.
19. S. Nahnsen, *et al.*, Probabilistic consensus scoring improves tandem mass spectrometry peptide identification, *J. Proteome Res.*, 2011, **10**(8), 3332–3343.
20. S. Kim, N. Gupta and P. A. Pevzner, Spectral probabilities and generating functions of tandem mass spectra: a strike against decoy databases, *J. Proteome Res.*, 2008, **7**(8), 3354–3363.
21. B. Y. Renard, *et al.*, Estimating the confidence of peptide identifications without decoy databases, *Anal. Chem.*, 2010, **82**(11), 4314–4318.

22. G. Gonnelli, *et al.*, A decoy-free approach to the identification of peptides, *J. Proteome Res.*, 2015, **14**(4), 1792–1798.
23. D. N. Perkins, *et al.*, Probability-based protein identification by searching sequence databases using mass spectrometry data, *Electrophoresis*, 1999, **20**(18), 3551–3567.
24. C. Y. Park, *et al.*, Rapid and accurate peptide identification from tandem mass spectra, *J. Proteome Res.*, 2008, **7**(7), 3022–3027.
25. R. Craig and R. C. Beavis, TANDEM: matching proteins with tandem mass spectra, *Bioinformatics*, 2004, **20**(9), 1466–1467.
26. L. Y. Geer, *et al.*, Open mass spectrometry search algorithm, *J. Proteome Res.*, 2004, **3**(5), 958–964.
27. Y. Benjamini and Y. Hochberg, Controlling the False Discovery Rate–a Practical and Powerful Approach to Multiple Testing, *J. R. Stat. Soc. Series B Stat. Methodol.*, 1995, **57**(1), 289–300.
28. A. Keller, *et al.*, Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search, *Anal. Chem.*, 2002, **74**(20), 5383–5392.
29. J. D. Storey and R. Tibshirani, Statistical significance for genomewide studies, *Proc. Natl. Acad. Sci. U. S. A.*, 2003, **100**(16), 9440–9445.
30. L. Kall, J. D. Storey and W. S. Noble, Non-parametric estimation of posterior error probabilities associated with peptides identified by tandem mass spectrometry, *Bioinformatics*, 2008, **24**(16), i42–8.
31. V. Granholm and L. Kall, Quality assessments of peptide-spectrum matches in shotgun proteomics, *Proteomics*, 2011, **11**(6), 1086–1093.
32. M. Ashburner, *et al.*, Gene ontology: tool for the unification of biology. The Gene Ontology Consortium, *Nat. Genet.*, 2000, **25**(1), 25–29.
33. L. Kall, *et al.*, Semi-supervised learning for peptide identification from shotgun proteomics datasets, *Nat. Methods*, 2007, **4**(11), 923–925.
34. J. D. Storey, A direct approach to false discovery rates, *J. R. Stat. Soc. Series B Stat. Methodol.*, 2002, **64**, 479–498.
35. F. R. Cerqueira, *et al.*, MUMAL: multivariate analysis in shotgun proteomics using machine learning techniques, *BMC Genomics*, 2012, **13**(suppl. 5), S4.
36. J. Zhang, *et al.*, Bayesian nonparametric model for the validation of peptide identification in shotgun proteomics, *Mol. Cell. Proteomics*, 2009, **8**(3), 547–557.
37. S. Kim, *et al.*, The generating function of CID, ETD, and CID/ETD pairs of tandem mass spectra: applications to database search, *Mol. Cell. Proteomics*, 2010, **9**(12), 2840–2852.
38. A. A. Goloborodko, *et al.*, Empirical approach to false discovery rate estimation in shotgun proteomics, *Rapid Commun. Mass Spectrom.*, 2010, **24**(4), 454–462.
39. W. Shao, K. Zhu and H. Lam, Refining similarity scoring to enable decoy-free validation in spectral library searching, *Proteomics*, 2013, **13**(22), 3273–3283.

40. M. Wang and N. Bandeira, Spectral library generating function for assessing spectrum-spectrum match significance, *J. Proteome Res.*, 2013, **12**(9), 3944–3951.
41. E. Ahrne, *et al.*, An improved method for the construction of decoy peptide MS/MS spectra suitable for the accurate estimation of false discovery rates, *Proteomics*, 2011, **11**(20), 4085–4095.
42. L. Kall, J. D. Storey and W. S. Noble, QVALITY: non-parametric estimation of q-values and posterior error probabilities, *Bioinformatics*, 2009, **25**(7), 964–966.
43. E. W. Deutsch, *et al.*, A guided tour of the Trans-Proteomic Pipeline, *Proteomics*, 2010, **10**(6), 1150–1159.
44. H. Choi and A. I. Nesvizhskii, Semisupervised model-based validation of peptide identifications in mass spectrometry-based proteomics, *J. Proteome Res.*, 2008, **7**(1), 254–265.
45. Y. Ding, H. Choi and A. I. Nesvizhskii, Adaptive discriminant function analysis and reranking of MS/MS database search results for improved peptide identification in shotgun proteomics, *J. Proteome Res.*, 2008, **7**(11), 4878–4889.
46. J. K. Eng, A. L. McCormack and J. R. Yates, An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database, *J. Am. Soc. Mass Spectrom.*, 1994, **5**(11), 976–989.
47. M. Spivak, *et al.*, Improvements to the percolator algorithm for Peptide identification from shotgun proteomics data sets, *J. Proteome Res.*, 2009, **8**(7), 3737–3745.
48. P. Yang, *et al.*, Improving X!Tandem on peptide identification from mass spectrometry by self-boosted Percolator, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 2012, **9**(5), 1273–1280.
49. M. Xu, Z. Li and L. Li, Combining percolator with X!Tandem for accurate and sensitive peptide identification, *J. Proteome Res.*, 2013, **12**(6), 3026–3033.
50. V. Granholm, *et al.*, Fast and accurate database searches with MS-GF +Percolator, *J. Proteome Res.*, 2014, **13**(2), 890–897.
51. S. Kim and P. A. Pevzner, MS-GF+ makes progress towards a universal database search tool for proteomics, *Nat. Commun.*, 2014, **5**, 5277.
52. M. Brosch, *et al.*, Accurate and sensitive peptide identification with Mascot Percolator, *J. Proteome Res.*, 2009, **8**(6), 3176–3181.
53. J. C. Wright, *et al.*, Enhanced peptide identification by electron transfer dissociation using an improved Mascot Percolator, *Mol. Cell. Proteomics*, 2012, **11**(8), 478–491.
54. B. Wen, *et al.*, The OMSSAPercolator: an automated tool to validate OMSSA results, *Proteomics*, 2014, **14**(9), 1011–1014.
55. M. Sturm, *et al.*, OpenMS–an open-source software framework for mass spectrometry, *BMC Bioinf.*, 2008, **9**, 163.
56. A. Frank, *et al.*, Peptide sequence tags for fast database search in mass-spectrometry, *J. Proteome Res.*, 2005, **4**(4), 1287–1295.

57. N. Li, *et al.*, PepDistiller: A quality control tool to improve the sensitivity and accuracy of peptide identifications in shotgun proteomics, *Proteomics*, 2012, **12**(11), 1720–1725.
58. M. Vaudel, *et al.*, PeptideShaker enables reanalysis of MS-derived proteomics data sets, *Nat. Biotechnol.*, 2015, **33**(1), 22–24.
59. C. D. Wenger, *et al.*, COMPASS: a suite of pre- and post-search proteomics software tools for OMSSA, *Proteomics*, 2011, **11**(6), 1064–1074.
60. R. J. Chalkley, *et al.*, In-depth analysis of tandem mass spectrometry data from disparate instrument types, *Mol. Cell. Proteomics*, 2008, **7**(12), 2386–2398.
61. A. K. Yadav, *et al.*, ProteoStats–a library for estimating false discovery rates in proteomics pipelines, *Bioinformatics*, 2013, **29**(21), 2799–2800.
62. M. Choi, *et al.*, MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments, *Bioinformatics*, 2014, **30**(17), 2524–2526.
63. A. K. Shanmugam and A. I. Nesvizhskii, Effective Leveraging of Targeted Search Spaces for Improving Peptide Identification in Tandem Mass Spectrometry Based Proteomics, *J. Proteome Res.*, 2015, **14**(12), 5169–5178.
64. G. Hart-Smith, *et al.*, Large-scale mass spectrometry-based identifications of enzyme-mediated protein methylation are subject to high false discovery rates, *Mol. Cell. Proteomics*, 2015, **15**(3), 989–1006.
65. B. Cooper, The problem with peptide presumption and the downfall of target-decoy false discovery rates, *Anal. Chem.*, 2012, **84**(22), 9663–9667.
66. B. Teng, T. Huang and Z. He, Decoy-free protein-level false discovery rate estimation, *Bioinformatics*, 2014, **30**(5), 675–681.
67. A. I. Nesvizhskii, *et al.*, A statistical model for identifying proteins by tandem mass spectrometry, *Anal. Chem.*, 2003, **75**(17), 4646–4658.
68. O. Serang, M. J. MacCoss and W. S. Noble, Efficient marginalization to compute protein posterior probabilities from shotgun mass spectrometry data, *J. Proteome Res.*, 2010, **9**(10), 5346–5357.
69. A. Ramos-Fernandez, *et al.*, Generalized method for probability-based peptide and protein identification from tandem mass spectrometry data and sequence database searching, *Mol. Cell. Proteomics*, 2008, **7**(9), 1748–1754.
70. L. Reiter, *et al.*, Protein identification false discovery rates for very large proteomics data sets generated by tandem mass spectrometry, *Mol. Cell. Proteomics*, 2009, **8**(11), 2405–2417.
71. M. M. Savitski, *et al.*, A Scalable Approach for Protein False Discovery Rate Estimation in Large Proteomic Data Sets, *Mol. Cell. Proteomics*, 2015, **14**(9), 2394–2404.
72. O. Serang and L. Kall, Solution to Statistical Challenges in Proteomics Is More Statistics, Not Less, *J. Proteome Res.*, 2015, **14**(10), 4099–4103.
73. J. J. Howbert and W. S. Noble, Computing exact p-values for a cross-correlation shotgun proteomics score function, *Mol. Cell. Proteomics*, 2014, **13**(9), 2467–2479.
74. J. Griss, Spectral Library Searching in Proteomics, *Proteomics*, 2015, **16**(5), 729–740.

CHAPTER 5

Protein Inference and Grouping

ANDREW R. JONES[a]

[a]Institute of Integrative Biology, University of Liverpool, UK
*E-mail: andrew.jones@liv.ac.uk

5.1 Background

In large-scale proteomics studies employing LC-MS/MS, "*shotgun studies*" (see Table 5.1 for a summary of terminology used) has become the prevalent method, in which proteins are digested into peptides early in the workflow. In many such experiments, no separation is performed in protein space, and LC is used with a long gradient, say 2 or more hours, to separate and limit the number of different peptides arriving at the MS instrument at the same point in time. These approaches are popular due to their technical simplicity, the lack of laborious sample handling (*e.g.* compared with the use of gel-based techniques for protein separation), and the fact that modern MS instruments can identify/quantify thousands of proteins in a single LC-MS run. However, one of the consequences of shotgun proteomics is that there is no direct link from a given peptide and the protein from which it was derived. As such, most proteomic identification software or workflows perform a second step, after the identification of peptides, to determine which proteins have been identified in a process known as "protein inference"[1] or protein grouping.

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

Table 5.1 A summary of terminology and definitions used in the chapter.

Term	Definition
Unique peptide	A peptide that can be assigned to only a single database protein
Shared peptide	A peptide that can be assigned to more than one database protein
Resolved peptide	A shared peptide that can be assigned to a group of proteins in which there are only same-set or sub-set relationships. Such peptides may be used with reasonable confidence for quantitation in some workflows
Conflicted peptide	A shared peptide that cannot easily be assigned to a single protein group, and could be assigned to more than one group. Such peptides are often considered dubious for quantitation purposes
Razor peptide	An assignment of a shared peptide to the database protein with the most other supporting evidence
Database protein	A single protein in the database that was searched, *e.g.* derived from a FASTA file loaded into the search engine
Protein group	A group of database proteins sharing some evidence in common
Same-set proteins	A set of database proteins with the same set of supporting evidence (peptide or spectral depending on the approach taken), which are then assumed to be indistinguishable
Sub-set protein	A protein with a sub-set of evidence (peptide or spectral) compared with one or more other proteins, and thus is usually considered not to have been identified by most protein inference approaches
Multiply subsumed protein	A type of sub-set protein, where it is identified based on peptide or spectral evidence where all the evidence is also contained within more than one protein – each of which has more evidence than the multiply subsumed protein
Protein cluster (family)	A collection of protein groups related *via* conflicted peptides
Representative protein	A single protein accession taken to represent a protein group, in some cases chosen arbitrarily *e.g.* from same-set proteins
Group leader protein	A single protein accession taken to represent a group, where it is assumed (but not always enforced by software), that it has more evidence than other group members

The methods discussed in this chapter are largely concerned with protein inference following peptide identification following a sequence database search (Chapter 3), but also apply to peptide identification following a spectral library search from which similar scores and statistical values for confidence in peptide identification are produced. Approaches for *de novo* sequencing of peptides (Chapter 2) are usually applied where there is no database of proteins available, for example if the sample is derived from a species with no sequenced genome. As such, most of the methods described here are not directly applicable to *de novo* sequencing, and specialised approaches are needed, for example using BLAST-like tools to query full or partial peptide sequences against protein databases from other species.

5.1.1 Assignment of Peptides to Proteins

The first step in most protein inference approaches is to determine which peptides have been confidently identified, and then assign these to proteins in the searched database, based on the assumed digestion that has taken place. For example, if the search was specified with "full trypsin cleavage" (cleavage after K or R, not followed by P), then peptide to protein assignment would only consider the peptide could be derived from proteins where the preceding residue is K, R or the N-terminus of the protein, and the residue after the peptide is not P. Search engines tend to export a list of all possible proteins in which a given peptide can be found, given the digestion constraints (Figure 5.1).

The assignment of a peptide to a parent protein is a relatively trivial process for any peptide that can uniquely be assigned to a single protein in the search database (a "*unique peptide*"). Assuming the peptide has been identified with

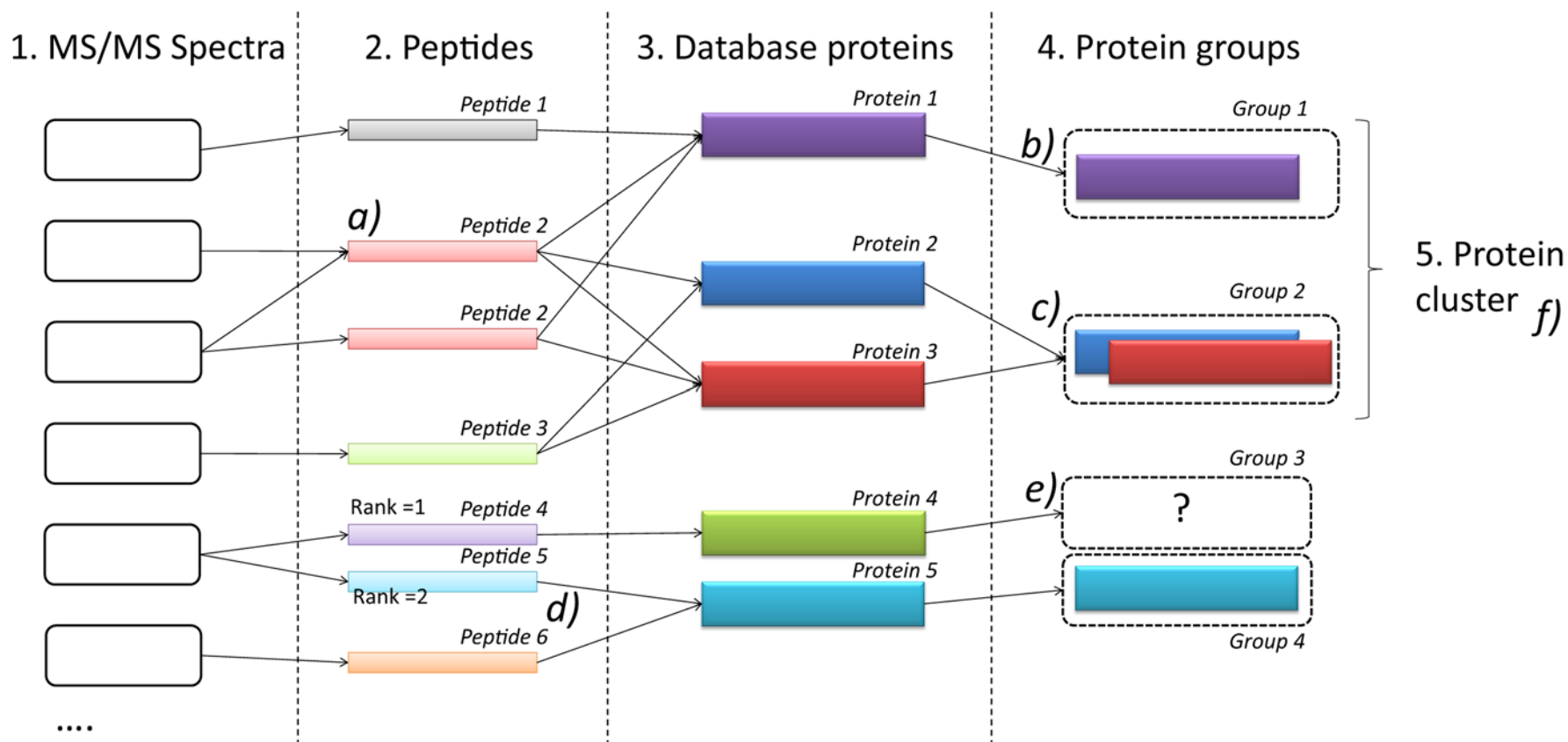

Figure 5.1 Overview of the different levels involved in protein inference. A database search engine is often used to perform stages 1–3, *i.e.* identifying peptides from spectra, and reporting all possible proteins from which they could have been derived. Protein inference, grouping and clustering (stages 4 and 5) are described in this chapter. Several key points that will be addressed are highlighted: (a) it is common for the same peptide to be identified by more than one spectrum, for the purpose of protein inference and scoring these entities are often collapsed to a single data point; (b) Protein 1 has one unique peptide (Peptide 1) and one shared peptide (Peptide 2), and thus forms its own group based on having a unique peptide; (c) Proteins 2 and 3 have the same-set of peptides and thus are reported together in one group; (d) it is common for the search engine to report multiple ranked peptides for a given spectrum and in some cases a lower ranked peptide can contribute to protein inference; (e) depending on the protein inference algorithm and scores (not shown), Protein 4 might be assigned to its own group or be assigned to a group with Protein 5; (f) a protein cluster can be formed linking Groups 1 and 2, due to Peptide 2 being a conflicted assignment.

sufficient confidence, the score or probability value associated with the peptide identification can contribute directly to a score or probability that the parent protein has been identified (see Section 5.3). However, it is common that many peptides identified cannot be uniquely assigned to a single protein, and can be found within several proteins within the database (a "*shared peptide*" or degenerate peptides in ref. 2).

Shared peptides arise for a number of different reasons:

1. *Paralogues*: in many species gene families are common, causing paralogous proteins (with similar sequences) to be present in the search database. Since peptides identified in proteomics tend to be short (say 6–25 amino acids), it is common that 100% peptide sequence identity is sometimes observed between paralogues (Figure 5.2).
2. *Alternative splicing*: some protein databases, such as UniProt,[3] contain different protein sequences resulting from alternative splicing of a single gene, causing different protein-level records in the searched

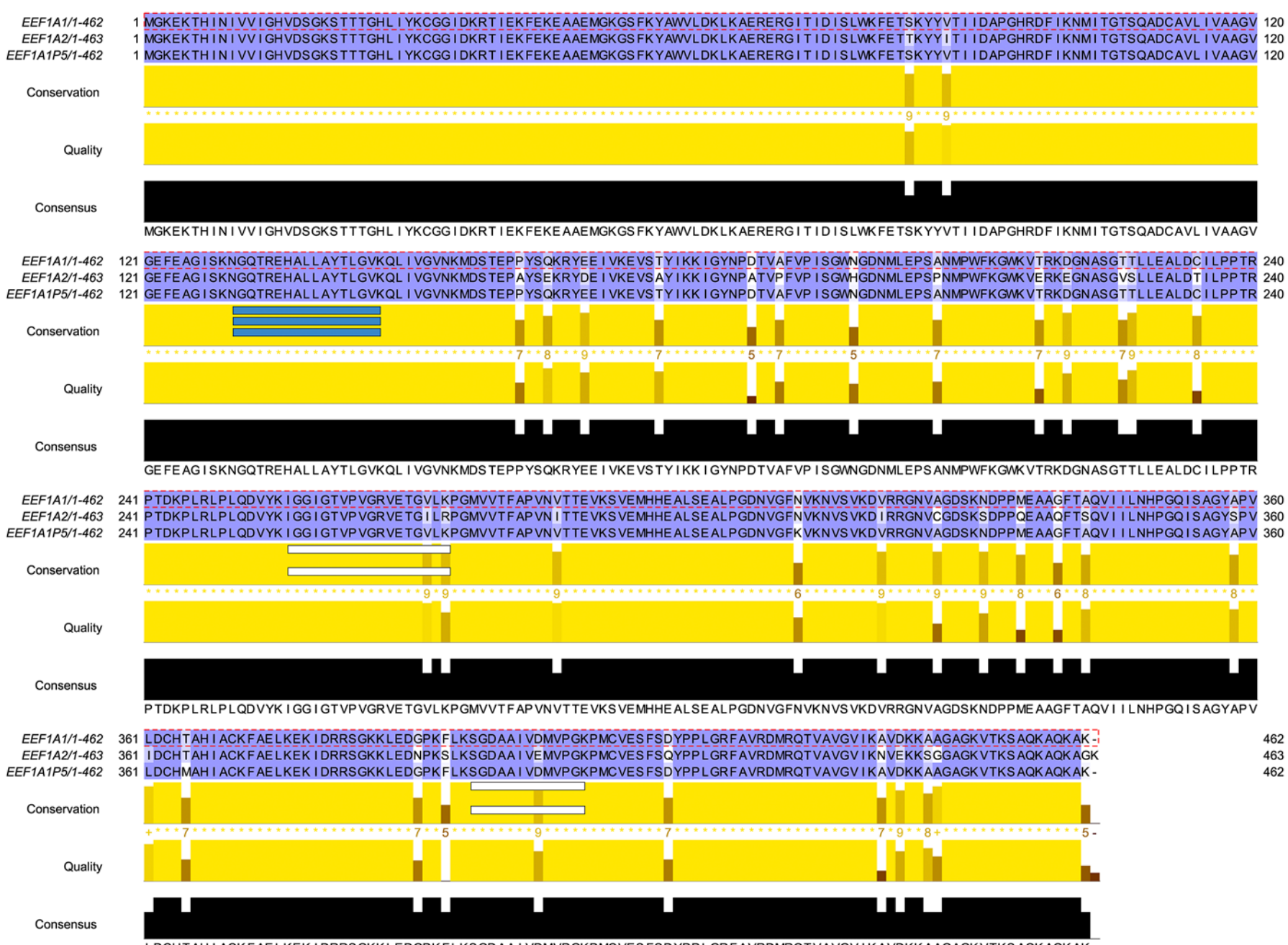

Figure 5.2 A multiple sequence alignment of three human paralogues (Elongation factor 1-alpha) with high sequence similarity. Any peptides mapping to the shaded regions of the proteins would be mapped to all three proteins. Three peptides have been identified. The shaded peptide can be found in all three proteins. The unshaded peptides can only be found in proteins 1 and 3. As such, a same-set group of protein 1 and 3 would be formed, with protein 2 as a sub-set protein. Protein 1 has been selected as a group representative.

database. Such protein records have shared peptides covering the exons shared between different predicted splice products.

3. *Alleles*: search databases can contain additional protein variants derived from the same gene, such as those caused by different alleles identified within individuals, in some cases differing by only a single amino acid polymorphism, in which cases multiple protein records can have all but one peptide in common.
4. *Redundant database merges*: in some studies, the search engine uses a database merged from different sources, such as different sets of predicted gene models (*e.g.* in "proteogenomics" approaches – see Chapter 15), to increase coverage and in efforts to find supporting evidence for particular gene models. Often such approaches lead to considerable levels of peptide-level sequence identity between different records in the searched database.
5. *Orthologues*: in studies on more than one organism, for example host-parasite proteomics, it is common to search a database merged from source databases from the different species. In these cases, it is possible for orthologous proteins to have peptide sequences in common.
6. *Chance matches*: it is possible for different proteins with no functional relationship or shared evolutionary origin to have short peptides in common by chance, although this rarely occurs for peptides ~>7 amino acids or so.

Any shared peptide thus cannot straightforwardly be assigned to a single "*database protein*", and has to be treated as providing evidence towards the identification of any or all of the proteins from which it could have been derived.

5.1.2 Protein Groups and Families

In the early days of shotgun proteomics, there was a tendency for studies to report all proteins for which there was any peptide-level evidence, including the common case of multiple proteins with the same-set of shared peptides. These approaches led to inflated protein lists being reported, especially for species containing extensive gene families (which give rise to paralogous protein sequences), or if databases containing protein variants were used in the search. This is a problem for several reasons. First, this could introduce biases in downstream data analysis, such as pathway mapping or functional enrichment. Any pathway or functional grouping in which there is a higher than average number of paralogues, alternative splicing or genetic variants in the search database would artificially appear "enriched" leading to incorrect biological conclusions, not supported by the data. Second, when examining evidence at the level of individual proteins, for example, following quantitative approaches for differential expression analysis or biomarker identification, it is crucial to know the actual

strength of evidence for a given protein's identification. In cases, where all of the peptides assigned to a protein X can also be assigned to a different protein Y with stronger evidence (more peptides), it is important to know that there is no independent evidence for protein X's identification, and it is quite possible it was not present in the sample at all. Third, there can be a tendency for labs to engage in competition over the number of proteins identifiable from a given sample or protocol. Without intelligent protein inference, a longer protein list can be produced simply by increasing the level of redundancy in the search database, which is evidently not an optimal scientific outcome.

In more recent years, it has been common to employ some rules of parsimony and report only the list of protein entities for which there is independent evidence supporting identification. Studies that report all possible protein identifications from a given peptide set would not generally pass peer review. To achieve a parsimonious result set, a variety of algorithmic approaches can be employed (as detailed in Section 5.3), most of which lead to the primary unit reported being the "*protein group*". The concept of a protein group represents the set of database proteins within which none of the group members (each database protein) have any independent, substantive evidence.

For many algorithms and software, a protein group in practice means sets of proteins supported by the same-set of peptides ("*same-set proteins*"). If a protein is supported by a sub-set of peptides, compared with one or more proteins in the group, in some approaches these "*sub-set proteins*" are included in the group but flagged in some way that they have probably not been identified (Figures 5.2 and 5.3). Where there are same-set proteins, *e.g.* Protein_a and Protein_b in a group, the software performing protein inference usually has no evidence to distinguish whether Protein_a, Protein_b or both have been identified. As such, an entity of [Protein_a, Protein_b] will be reported, and for the purposes of counting the number of proteins identified, the group would be counted only once. In some approaches, a single protein is selected to act as a "*representative protein*", based on some arbitrary rule such as alphabetical order (further discussed in Section 5.2.6). A more intelligent approach for selecting a representative protein is theoretically possible, for example an algorithm could give preference to a canonical gene model or commonly observed protein over a *de novo* gene prediction, which has never previously been observed. Such approaches are sometimes used in proteogenomics (Chapter 15), but source information on database or protein-level preference is not usually available to the software in most protein inference approaches.

The purpose of a representative protein is to simplify downstream data analysis, such as functional or pathway-based approaches that cannot usually handle the concept of protein groups, although with the obvious caveat that if Protein_a and Protein_b have different functions or map to different pathways, this can introduce a source of error into such analyses.

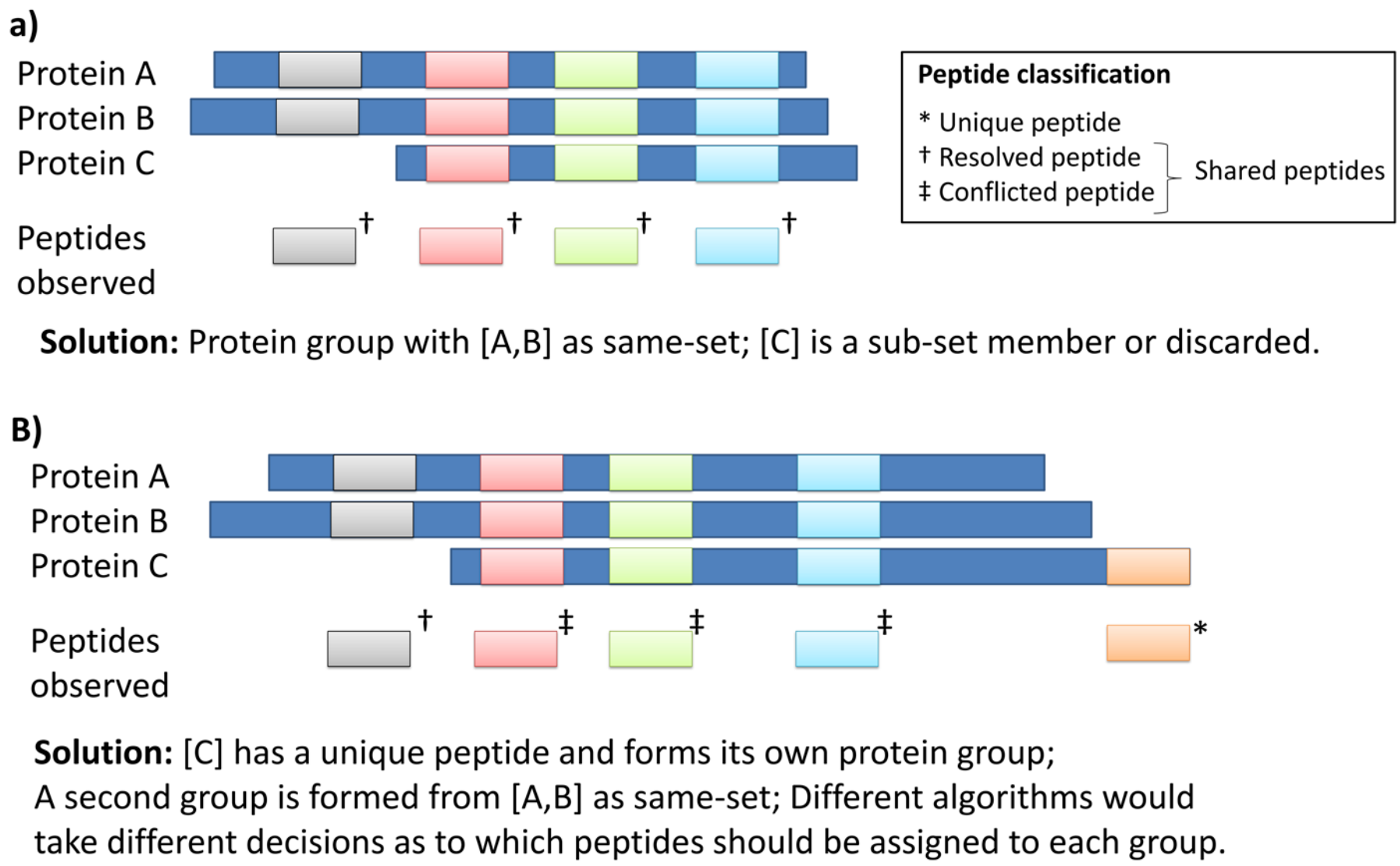

Figure 5.3 (a) A simple protein grouping scenario where proteins A and B are supported by the same-set of peptides, protein C has a sub-set of peptides; (b) a more complicated grouping scenario where Proteins A and B have a single "resolved peptide" and thus form one protein group; protein C has a single unique peptide and thus forms another group; three peptides are conflicted and cannot straightforwardly be assigned to either resulting group.

In some protein inference approaches, a further concept of a "*protein cluster*" ("*protein family*" in the Mascot software)[4] has been introduced (Figure 5.1). A protein cluster represents a wider concept than a protein group, capturing a set of protein groups that are linked through having peptides in common. Such peptides are sometimes called "*conflicted peptides*", as they cannot easily be assigned as belonging to one group or the other. This situation commonly arises when analysing proteome data resulting from gene families containing many paralogues with variable sequence identity amongst members. The concept of a protein cluster can be useful for visualising relationships in the data between protein groups, which would otherwise be lost if conflicted peptides are simply discarded from the analysis altogether. As shown in Figure 5.1, Protein Groups 1 and 2 share Peptide 2 in common, say because the proteins within these two groups all belong to a single gene family and are paralogues. Once groups have been formed, without the concept of a protein cluster, or a visualisation showing conflicted peptides across groups, a researcher would not necessarily be aware that there is any relationship between Protein Groups 1 and 2 in the data.

5.2 Theoretical Solutions and Protein Scoring

5.2.1 Protein Grouping Based on Sets of Peptides

One of the most commonly applied approaches is to perform protein grouping, based on same-set and sub-set peptide relationships amongst proteins. The following general steps may be followed (Figure 5.3):

1. Determine the list of peptides that have been confidently identified at a given threshold, such as false discovery rate (FDR) < 0.01, and discard all peptides not passing this threshold.
   a. Note: some software may also choose to discard any peptides that while passing the given FDR threshold are not the top ranking candidate for a given spectrum (Figure 5.1d). There is debate in the field as to which is the most appropriate action. On the one hand, it is plausible for more than one peptide to be correctly identified from a single spectrum, due to co-isolation of peptides with similar mass/charge. On the other hand, rank = 2 or greater peptides are often alternative hypotheses explaining many of the same fragment ions as the rank = 1 hypothesis, in which case there is no evidence to suggest that *both* rank = 1 and rank = 2 (or higher ranks) have been observed. Spectral-based protein inference approaches can handle this problem intrinsically.
2. The protein inference software then assigns all peptides (passing threshold in step 1) to all proteins in which they can be observed.
3. The software then traverses the list of proteins to discover same-set and sub-set relationships, by analysing the peptides assigned to each protein.
4. Protein groups can be formed under the following conditions (Figure 5.3):
   a. If a database protein has one or more unique peptides, it will form a protein group, acting as the group leader.
   b. If two or more database proteins share the same-set of peptides, and no other proteins exist (with more peptides assigned) to which the same peptides can be assigned, then a new group is formed.
5. Database proteins with shared peptides but existing in sub-set relationships (*i.e.* all peptides can also be assigned to another database protein with more peptides), are then assigned to the appropriate group and flagged as a sub-set protein, or discarded altogether, depending on the algorithm.
6. Following group formation, peptides can be assigned as resolved (existing in same-set relationships) or conflicted (potentially assigned to more than one group) – Figure 5.3(b).

These steps do not cover all possible cases that might occur, which if not appropriately handled will lead to some independent evidence for the

observation of additional proteins being wrongly discarded. As shown in Figure 5.4, a not uncommon scenario is for peptides to be distributed across several proteins, but not resulting in straightforward same-set and sub-set relationships. In the case shown in Figure 5.4, there are three database proteins, supported by three peptides. The evidence points to at least two protein entities having been observed *i.e.* under rules of parsimony, the evidence cannot be explained by only a single protein in the sample. There are several possible solutions, in order of perceived validity:

1. First, assign peptides to proteins based on rank ordering of proteins *e.g.* by protein score, then form groups. In the example, this would result in a group being formed from the top scoring protein B, and claiming peptides 2 and 3. A second group would be formed by protein C, claiming the remaining peptide 1. Protein A could then be assigned a "*multiply subsumed*" member of either or both groups.
2. Form two groups and arbitrarily assign group membership.
3. Report three protein groups (non-parsimonious solution).
4. Report a single protein group containing A, B and C – under-representing the evidence.

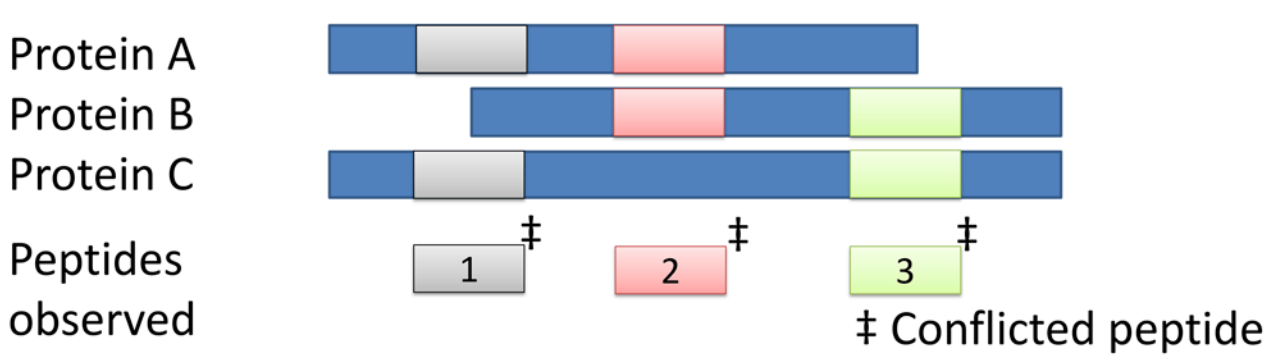

Solution: The evidence supports at least two different proteins having been identified. Based on peptide assignment only, an algorithm cannot easily determine groupings. Taking into account a protein score (table below), an algorithm could suggest two protein groups: [B] and [C], and A is "multiply subsumed" by groups [B] and [C].

Peptides Proteins	1 *(35)*	2 *(45)*	3 *(65)*	Protein score
A	X	X		80
B		X	X	110
C	X		X	100

Figure 5.4 An example grouping scenario where three proteins are supported by three conflicted peptides. The solution to the grouping problem can be solved in some algorithms by taking peptide and protein scores into account. Peptide scores (italic) go from low to high quality (low = weak identification; high = strong identification).

There are other variations on this concept involving more complex groupings and sets of conflicted peptides. The classification algorithm can be achieved simply by ordering proteins based on some preference (*e.g.* most peptides, then score, then alphabetical order), and then assigning every peptide to the first protein in the list. Shared peptides assigned to a single protein based on a protein-level preference are sometimes called "razor peptides" (after Occam's razor) – following the terminology in ref. 5. Such an algorithm will ensure that all evidence is captured, and that stable protein groups can be formed, based on same-set, sub-set and multiple subsumed relationships. However, as shown in Figure 5.4, caution needs to be taken in result interpretation. One implementation of an algorithm might suggest that Protein A has been multiply subsumed and thus probably not identified. However, an alternative explanation of the data might suggest that Protein A is just as likely to have been identified as Proteins B and C. It is important that such nuances are well communicated by software and in data formats (Section 5.4), enabling researchers to at least be aware that caution needs to be taken when interpreting results, and more work still needs to be done in this space.

5.2.2 Spectral-Focussed Inference Approaches

A spectral-focussed approach differs from a peptide-focussed approach in that lower ranked peptides potentially identified from a given spectrum can be intrinsically considered (Figure 5.5). From the data presented in Figure 5.5, a peptide-focussed approach would do one of two things. If only rank = 1 peptides were allowed, then Peptide A would be assigned to Protein X as a unique peptide, forming one protein group. Peptides E, G and I would be assigned to Protein Y forming a second protein group, no evidence for Protein Z would be considered. If the peptide-focussed approach allowed rank ≥ 1 to be considered, then Peptide J would be assigned to Protein Z forming a third protein group. A spectral-focussed approach would take a different approach. First, all plausible peptide identifications are assigned to proteins, including those with rank > 1. Those shown in italic with very low scores are here discarded, but are included in some implementations (with a very low weighting factor). Next a protein grouping approach is taken following the same algorithm as for peptide-focussed approaches, but instead forming same- and sub-set groupings based on spectra rather than peptides. When examining proteins X, Y and Z, it is discovered that the data only supports the parsimonious identification of a single protein group – in which protein Y has most evidence, and proteins X and Z are classified as "spectral sub-set proteins".

It is important to note that based on identical search engine results from only four spectra, different apparently "parsimonious" inference algorithms could arrive at one, two or three proteins being identified.

Spectral-focussed inference approaches have the advantage of intrinsically weighing the evidence when two or more peptide candidates produce very similar or identical scores from one spectrum. Identical scores occur

Published on 15 November 2016 on http://pubs.rsc.org | doi:10.1039/9781782626732-00093

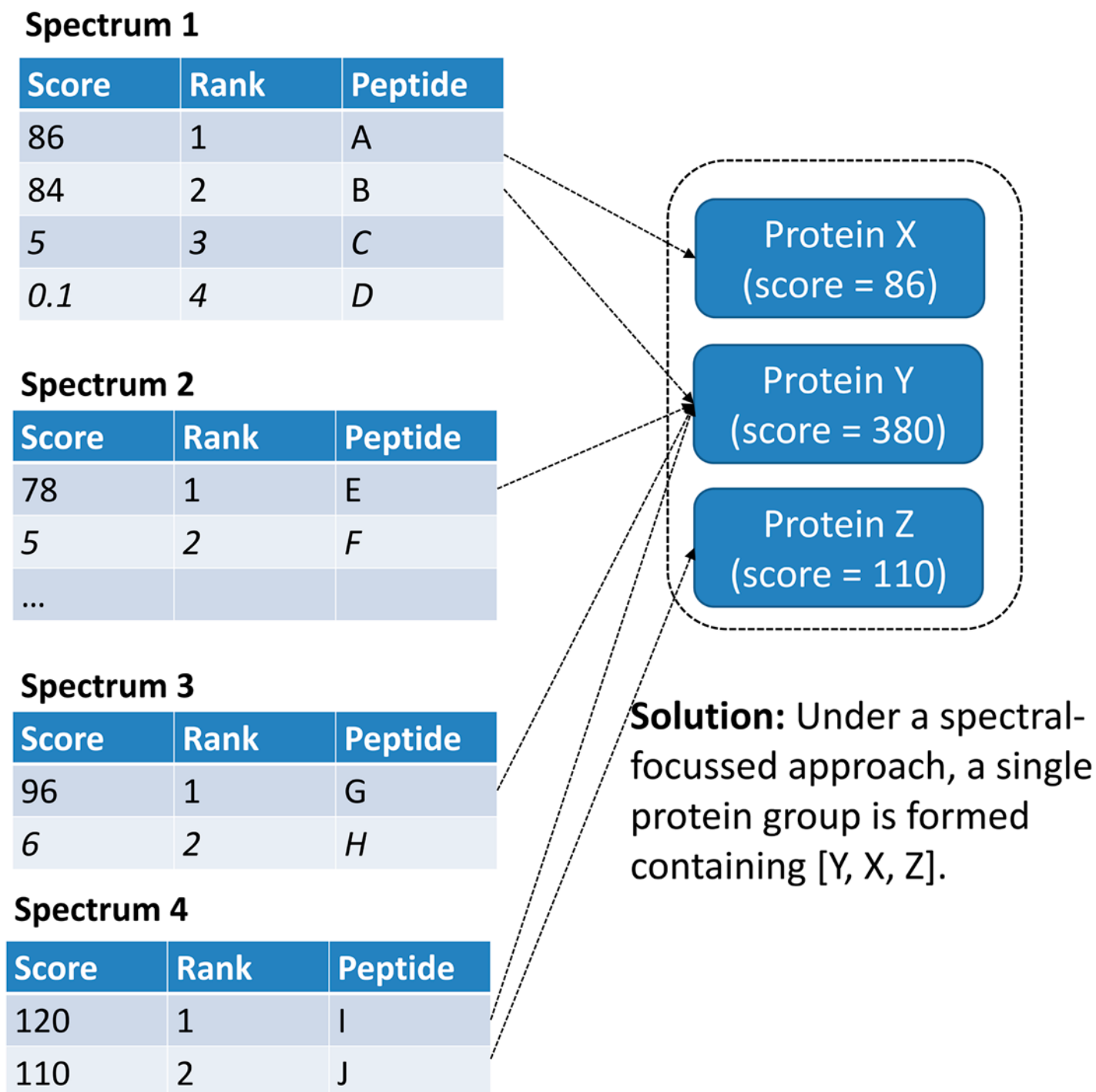

Figure 5.5 An example demonstrating how a spectral-focussed inference algorithm can take into account different ranked identifications from the input spectra – forming a spectrum same-set group. Peptide scores go from low to high quality (low = weak identification; high = strong identification). A peptide-focussed algorithm would form two or three protein groups depending on how lower ranked peptides are handled.

whenever two peptide candidates contain isoleucine or leucine (indistinguishable by traditional LC-MS workflows), or for example when certain modifications are tested for: deamidation of asparagine is chemically identical to aspartic acid. Furthermore, even with peptide hypotheses that are technically distinguishable, identical scores can be achieved when two different peptide sequences have considerable sequence identity. Identical scoring peptides (both rank = 1 for a given spectrum) are problematic for peptide-focussed inference algorithms, since neither possible approach (accept both peptides or arbitrarily accept only one) correctly reflects the evidence. In terms of similar scoring peptide candidates, spectral-focussed approaches are also theoretically superior to peptide-focussed approaches, since they handle the case in a sensible manner where two candidates have essentially almost identical evidence – as in Figure 5.5 where there is no strong evidence that Protein X is actually present in the sample.

While there are theoretical advantages to spectral-focussed inference approaches, their actual benefits are very challenging to demonstrate in practice to biological researchers. In many cases, peptide-level or spectral-level

inference approaches will produce highly similar lists and counts of identified proteins. Spectral-level approaches can be more conservative, as shown in Figure 5.5, producing only a single protein group, where a peptide-focussed approach would produce two or three groups depending on the implementation. As such, an end user applying a spectral-focussed software package would potentially observe only that a shorter protein list had been determined than if processing the data with a peptide-focussed package. In many cases, particularly with commercial software, the search engine and protein inference algorithm are integrated in a single step. The overall count of peptides and proteins, say at 1% FDR for both, can be taken as a measure of software performance given the same data, where a higher count is typically preferred. However, in the case of protein inference, the ability to produce a shorter (or more parsimonious) protein group list from a given set of input peptide-spectrum matches (PSMs) may in fact be a better measure of success. Benchmarking protein inference algorithms is a very challenging task for this reason. The Association for Biomolecular Research Facilities (ABRF) – Proteome Informatics Research Group (iPRG) generate yearly benchmarking studies in which different labs/groups analyse the same data sets, to test software performance. The iPRG2008 study investigated this phenomenon, designing a complex scenario in which there were extensive shared peptides, and where the number of protein groups was known in advance. To our knowledge, the full analysis of study results has not been published, although it is described in studies that re-used the data including ref. 4 and 6. The initial (unpublished) results appeared to indicate that best performance (fewest false positive protein groups) was achieved by Pro Group from SCIEX, an algorithm that is spectral-focussed. It is currently largely unknown how much variability in proteomics workflows is introduced in practice through the use of different inference algorithms, since in most cases these cannot be decoupled from the search engine, although in theory spectral-focussed algorithms appear superior.

5.2.3 Considerations of Protein Length

The decision as to whether to exclude low scoring PSMs from inference can impact on downstream results, and there is some dependence on the protein length. For example, a naïve approach in which a peptide of any score could contribute to inference could have unexpected consequences in the case of very large proteins. The largest human protein is titin, with a molecular weight of 3.8 million Daltons (~30 000 amino acids). By chance, many very low scoring peptides could suggest proteins such as titin have been identified when in fact no single *significant* PSM had been made. As such, an algorithm could have an additional requirement that only proteins with at least one significant peptide identification should be considered for inference, which would get around this problem to some extent. More generally however, in any protein inference approach, there is a potential for bias towards longer proteins, since they have more chances to be identified than short

proteins. Some inference approaches attempt to correct for this *e.g.* ref. 7. However, while comparing unrelated proteins it seems intuitive to correct a protein-level score or probability for protein length, this can have undesired consequences for related proteins. For example, where two proteins have been identified in a (peptide or spectral) same-set relationship, it does not seem intuitive to suggest that the shorter protein is more likely to have been identified, as would result if protein scores are normalised by protein length. Most protein scoring approaches (Table 5.2) do not appear to correct for length at the present time.

5.2.4 Handling Sub-Set and Same-Set Proteins within Groups

Under either a spectral or peptide-focussed inference approach it is common to identify sub-set proteins *i.e.* those supported only by evidence (spectra or peptides) that can also be assigned to other proteins that have a greater amount of evidence. Such sub-set proteins can either be assigned to the protein group with which most evidence is shared, and flagged as sub-set proteins, or discarded altogether. In either case, most researchers would ignore these entities in downstream data analysis as probably not being present in the sample. However, such an approach could lead to some unintended consequences. Consider a case where the database sequence of Protein X has a true sub-set of tryptic peptides as Protein Y *e.g.* if X and Y were derived from alternative splicing of a single gene (X has exons 1, 2, 4 and Y has exons 1, 2, 3, 4; and peptides crossing splice junctions are not good tryptic candidates for identification.). In a shotgun approach, if both Protein X and Y were truly present in the sample, Protein X would always be identifiable only as a sub-set protein assuming high sequence coverage. Even if only Protein X was present in the sample, it could only ever be identified as a same-set protein, unless a normalisation approach is used based on protein length, which is not standard practice. At present there is no obvious solution to this problem, and to our knowledge, few, if any, studies have attempted to measure the extent of this potential issue in practice.

With (peptide or spectral) same-set proteins in groups, software either report them as a group *e.g.* [A, B, C] as a single unit or as a representative protein [A] with same-set members [B, C]. The distinction can be important depending on downstream data analysis that is performed. As for sub-set proteins, the same potential problem exists where two or more proteins (*e.g.* Protein A and B) have the same-set of identifiable tryptic peptides *e.g.* due to identical or very similar protein sequences. If the protein inference algorithm always favours Protein A (say on alphabetical order as a tiebreaker), Protein B would always be unidentifiable in a given workflow.

The upshot from both same-set and sub-set scenarios is that (a) researchers should be aware that manual interpretation of results may still be required and (b) developers of software and data formats should endeavour to communicate the remaining ambiguity in protein grouping in a clear manner. From a given protein group, it is possible that all of the members were present in

Table 5.2 A summary of protein scoring and inference approaches used in some popular software. Note, this is not an exhaustive list, and there are many other packages available, which mostly use a method similar to those described.

Software package	Availability and/or reference	Inference approach	Protein scoring	Notes
Mascot	Commercial, Matrix Science	Peptide-focussed	Protein score summing peptide ion scores to give a protein score	The original Mascot grouping approaches used simple peptide same- and sub-sets. This was later updated to a protein family approach[4] in which hierarchical clustering is performed over shared peptides (using distance based on scores from non-shared matches) to assist users in visualising structure across protein groups
MaxQuant	Free[5]	Peptide-focussed	Protein-level PEP	MaxQuant assigns peptides to protein groups using the so-called "razor peptide" approach – assigning a shared peptide to the protein that has the most evidence, but are reported for all groups in which they could occur. Protein PEP is derived by multiplication of peptide-level PEPs, taking best PEP in case of multiple PSMs per peptide, and using PEPs from resolved peptides only. Users can choose whether to quantify from unique peptides only, unique and razor (shared) peptides or all peptides
Protein-Prophet	Free[2]	Peptide-focussed	Protein probability	Protein probability is calculated directly from peptide probabilities, as discussed in the main text following standard probability theory for independent events. Conflicted peptides can contribute to protein probabilities *via* a weighting scheme
Scaffold	Commercial, Proteome Software	Spectral-focussed	Protein probability	Scaffold protein-level scoring (probability calculation) is built upon ProteinProphet with a slight adaptation to the method so that it runs more quickly. Protein grouping uses a tripartite graph: spectra-peptides-proteins, as described in ref. 16
Protein pilot	Commercial, SCIEX	Spectral-focussed	"ProtScore" approach	A spectral-focussed algorithm that enables lower ranked peptides to contribute to protein-level scoring, but no spectral-level data can contribute to different protein groups *i.e.* spectral same-set and sub-set relationships are established

PEAKS	Commercial, Bioinformatics Solutions Inc.	Peptide-focussed	Protein scores, derived from summing −10 log (*p*-values)	PEAKS follows a protein grouping approach similar to the algorithm described in Section 5.2.1. Protein scores are formed by summing values derived from peptide-level *p*-values, using −10 log(*p*-value) to transform them to a positive scale. A weighting factor is introduced to account for lower ranked peptides from a given spectrum, where the weight is 1/rank × −10 log(*p*-value)
Progenesis QI for proteomics	Commercial, Waters	Peptide-focussed	Protein confidence score	A protein grouping approach similar to that described in Section 5.2.1 is used. Resolved peptides are used for quantitation but conflicted peptides are not. A protein confidence score is formed by summing all peptide-level scores derived from the source search engine, and various input search engines with different score types are supported
Proteome discoverer	Commercial, Thermo	Spectral-focussed	Protein probability	A Bayesian statistical inference model has been implemented in Proteome Discoverer 2.0, as described in,[17] using a so-called “convolution tree” approach. The approach is similar to ProteinProphet, in that the evidence associated with conflicted peptides is shared amongst candidates proteins, iteratively updating protein-level probabilities until a stable result is obtained

the sample, or that only one of the members was present. Even if a sub-set protein appears at first look to have significantly less evidence than other group members, this may be due to it being a considerably shorter protein or lower abundance (in both cases leading to fewer identifiable peptides).

5.2.5 Assignment of Representative or Group Leader Proteins

Many protein grouping approaches attempt to assign a single protein as representative of a given group – sometimes called a representative protein or group leader ("anchor protein" in Mascot, Matrix Science). This has become common since pathway or functional enrichment approaches were generally designed for gene expression data, and cannot handle the concept of a protein group as a single entity. A group leader would often be assumed to be the protein within the group that has the most evidence (highest count of peptides, highest score). In the case of same-set group, a group leader cannot be unambiguously assigned, and thus caution must be taken when interpreting results if the software does not clearly distinguish these two cases. The term representative protein is sometimes used instead implying the concept that a protein has been chosen as a single accession for the group, but that the selection criteria might be arbitrary, such as alphabetical order. An ideal classification and software output would make it clear to end users which proteins fall into these two groupings, although this is not common.

5.2.6 Importance of Peptide Classification to Quantitative Approaches

Shared or razor peptides (shared peptides assigned to proteins with most evidence) can be further classified into those that participate in same-set relationships, which could be called "*resolved peptides*" or those could belong to more than one protein "*conflicted peptides*" (or bridge peptides in ref. 8). The distinction between resolved peptides or conflicted peptides is actually not necessary in the peptide-centric protein grouping approach described, since both types are assigned to the protein with most evidence or based on arbitrary preference. However, it becomes important for many quantitative software packages that generally use identification data from search engines for inferring the protein groups present and quantifiable. Some quantitative packages choose to quantify using: (1) unique peptides only; (2) unique + resolved peptides; (3) all peptides. The downside of using unique peptides only is that same-set relationships are exceptionally common, particularly when searching (for example) a UniProt proteome for human or other animals, which contains protein variants from the same gene, as well as paralogues. As such, a large proportion of proteins could not be quantified using unique peptides only. A pragmatic approach is thus to use category 2 peptides for quantification, since conflicted peptides are comparatively rare. Conflicted peptides are potentially highly problematic

for quantification, since the peptide signal observed by LC-MS will usually be derived from a mixture of at least two different source proteins, which may well have different abundance profiles across sample conditions. In theory, it could also be argued that where same-set relationships exist (resolved peptides), the same problem could occur, and potentially all peptide ions give mixed signals from different source proteins. However, for at least some resolved peptides, there may indeed only be a single protein in the sample (we just do not know which one), and thus reliable quantitation values are achievable. For conflicted peptides, we have concrete evidence that they are derived from a mixed signal and likely only add noise to the analysis.

5.2.7 Scoring or Probability Assignment at the Protein-Level

In all workflows employing protein inference, some form of protein scoring is used (Table 5.2), for ordering the final protein list, and in some cases for performing global statistics, such as False Discovery Rate (FDR) analysis. Most search engines produce a PSM or peptide-level score, often as an integer or floating point number where usually higher scores are better. These are sometimes produced by performing a minus log 10 transformation on e-values or *p*-values. Mascot for example creates an "ion score" based on: $-10\,\mathrm{Log}(P)$, where P is the probability that the PSM is a random event (*p*-value). Ion scores for reliable PSMs thus range from around 45 to >100 for commonly observed *p*-values. In Mascot, a protein score is formed by summing the ion scores values for all peptides that can be assigned to a protein (taking only the best ion score per distinct peptide). Depending on the size of the input data set (number of spectra), Mascot has two different modes – regular scoring and MudPIT scoring. In the former, all possible peptide assignments are considered (including low scoring peptides below the peptide significance threshold, and rank > 1 PSMs). When the size of the input data set becomes large (beyond an internal threshold), "MudPIT scoring" is used instead, whereby only peptide identifications above the significance threshold contribute to the protein score. Such an approach is needed since for large scale searches chance low scoring matches can start to accumulate, particularly for peptides assigned to high molecular weight proteins. Since it is log-based, summing the ion scores has the same effect as multiplying *p*-values. Performing an inverse operation on the protein score thus equates to a protein-level *p*-value, which is a valid assumption if the peptide identifications are considered independent events.

A similar approach is taken by ProteinProphet,[2] which rather than estimating the *p*-value for a protein (probability that it is a random false positive with such a score), it instead calculates the probability that a protein has been identified. It could be argued that the probability of identification is a more intuitive and useful value for researchers than a *p*-value. The input data is the set of peptide-level identification probabilities $P_{\mathrm{pep}(1..n)}$ (*i.e.*

the probability of identification again rather than *p*-values), calculated by PeptideProphet.[9] Following basic statistical theory, it is straightforward to calculate the probability that the protein has *not* been identified, by multiplying together all peptide level values of: $(1-P_{\text{pep}(1..n)})$. One minus this value, then gives the protein-level probability. This approach is fairly simple in the case of unique peptides, but falls down for shared peptides. ProteinProphet implements an iterative algorithm, which weights assignments of peptides to proteins based on protein-level probabilities. In simple terms, if a shared peptide A could be assigned to protein X that is well supported by other evidence or protein Y that has little other evidence, a higher weighted probability for A will go to X over Y. The process is repeated in iterations until a stable result is obtained.

In both Mascot and ProteinProphet, peptide-level scores/probabilities are used as input rather than PSM-level *i.e.* by taking the only best scoring or most probable PSM for each peptide in the (common) case of multiple PSMs identifying the same peptide. More generally, there is a move towards using peptide-level scores, *p*-values or probabilities for protein-level scoring, instead of PSM-level scores. The rationale is that multiple PSMs for the same peptide do not constitute independent evidence, and can lead to systematic error. For example, consider a case where the top ranking result for 10 spectra is peptide X with a relatively weak score; where in fact all 10 spectra were derived from peptide Y, which was not included in the search, but for various reasons produces a similar pattern of fragment ions to peptide X. This phenomenon can occur due to a missing or incorrect gene model annotation or if peptide Y contains a modification that was not included in the search. Without applying such a correction, 10 weak scores for peptide X could lead to the parent protein incorrectly having a high score or apparent statistically significant result. Taking only the best score for peptide X from all 10 PSMs, avoids this potential bias.

Beyond the examples given for Mascot and ProteinProphet, protein-level scores can be calculated in a variety of mechanisms based on different peptide or PSM-level scores, including *p*-values, q-values or FDR values, posterior error probabilities, Bonferroni corrected *p*-values and so on (see Table 5.2). In many approaches, the protein list will be ordered by the resulting protein-score, and the target-decoy method can be re-applied (see Chapter 4) to estimate the FDR. The target-decoy approach is valid, so long as the search and protein inference process has not introduced any biases that would favour targets over decoys. Approaches that up-weight the scores of peptides, based on protein-level information (as performed by ProteinProphet), have the potential to break the central assumption of the target-decoy FDR method that targets and decoys are equally likely, which can lead to under-estimation, at least of peptide or PSM-level FDR (There is some discussion of this issue in ref. 10). As such, before a target-decoy approach is applied, some understanding of the approach taken for protein inference and scoring is needed.

5.2.8 Handling "One Hit Wonders"

The concept of a protein identification supported by a single peptide (so called "one hit wonders") has been the subject of much debate in the field, in terms of whether they are acceptable or not. In any shotgun study, there will typically be a large proportion of the potentially identifiable protein set, where the proteins are each supported by only a single peptide. This is because once proteins are ranked by score or probability, those proteins truly in the sample but of lower abundance or containing fewer peptides that ionise well, will have decreasing numbers of peptides identified, and indeed the majority of proteins actually in the sample will have zero peptides identified, and thus will not be observed. By including one hit wonders, a researcher will be able to extend the list of proteins identified, potentially increasing the impact of the publication, or the ability to make downstream conclusions. On the flip side, however, most false positive identifications are proteins identified by a single peptide. As a result, false positives can be almost completely eliminated by setting a threshold that a protein identification must have at least two distinct peptide identifications, so long as sensible thresholding (see Chapter 4) had already been performed on the peptide list.

For the purposes of quantitative studies, it is generally agreed that quantification from a single peptide is not recommended practice, since there is no independent evidence that a reliable quantitation has been achieved, and beyond the identification stage, various other factors can also go wrong (feature detection, mapping across replicates, normalisation *etc.*). For identification studies, there is no clear consensus on whether to trust one hit wonders. If a sensible protein inference and scoring approach has been taken, from a statistical point of view, there is no reason to trust a protein (group) with a weaker score but two peptides, over one with a higher score but only a single peptide.

Many groups now choose to perform target-decoy FDR calculation at the protein-level, and allow any protein (regardless of the number of peptides) passing at 1% FDR into the final set. While this approach is widespread, it is not without its limitations. For example, if there are only ~100 proteins identified in a given study, the appearance in the ranked list of the first or second decoy protein, will be enough to push the estimated FDR over the 1% level, where the cut-off is made. As such, there is considerable randomness in where the cut-off line is ultimately drawn. More generally, this is a limitation of protein-level target-decoy approaches, in which the score of a small number of decoys can have considerable effects on the position of the threshold. This issue is less problematic for peptide-level target-decoy, since the overall number of peptides is larger, and thus estimates of FDR are more robust. The overall conclusion is that different levels of trust should be placed in those proteins supported by large amounts of evidence, and those with weak evidence, which have only just passed the threshold used. Ideally, downstream analysis (such as pathway-based analyses) would incorporate the probability of protein identification intrinsically into the tools, but we are not aware that such tools are generally available.

5.3 Support for Protein Grouping in Data Standards

As described in Chapter 11, the Proteomics Standards Initiative (PSI) has been working since 2002 to improve efforts in data sharing, by developing reporting requirements documents, standard data formats, and supporting software. The main format for identification data, for example produced as output by a search engine, is mzIdentML.[11] mzIdentML is an XML-based format, capturing a detailed trace of the process of peptide identification, including input parameters, software used, output scores and statistics. The format can also capture the results of protein inference either performed alongside peptide identification in a single stage, or by a second stage post-processing algorithm or package. At the time of writing, the stable version of mzIdentML is version 1.1, as described in ref. 11, which has remained unchanged since 2012. Extensions to the format can be achieved by adding new terminology to the PSI-Mass Spectrometry Controlled Vocabulary (PSI-MS CV), as described in ref. 12. A valid mzIdentML file must conform to the XML Schema as originally released, and pass so-called "semantic validation",[6,13] which checks that correct and valid CV terms have been used in the correct locations of the file. The format can thus remain stable in terms of the core schema, but while allowing for new terms to be added, such as new scores or statistics from search engines as they are produced. In mzIdentML 1.1, a two-level hierarchy of results can be captured in the following file elements: ProteinAmbiguityGroup (PAG) and ProteinDetectionHypothesis (PDH).

Within the overall list of protein results, it was intended that each PAG could contain 1 or more PDH elements, where each PDH element mapped to a single database protein that had been identified, and the PAG corresponded to the concept of protein group as defined in this chapter. A set of CV terms was also included in the PSI-MS covering various possible relationships, such as peptide same-set, spectral same-set and so on, and the term "anchor protein" (from Mascot, Matrix Science) to describe a representative protein for the group. While mzIdentML 1.1 overall has become a widely used and stable standard, it has now been acknowledged that the protein grouping aspect was not specified sufficiently tightly to ensure that different groups implementing the standard, would always use the core features – PAG, PDH and CV terms in a consistent way. In addition, mzIdentML 1.1 lacked a clear mapping between the count of identified proteins, which an investigator might wish to report in a manuscript, and an attribute of a given file *e.g.* one could count PDH or PAG elements. In response, an update to mzIdentML is under review, as described in ref. 14, from which mzIdentML 1.2 will emerge. The new aspects of mzIdentML 1.2 include mandatory terminology being added to protein groups, following many of the concepts described in this chapter. There is also now a clear expectation that when one wishes to count the number of identified proteins, this is derived from a count of the protein groups (PAGs) in mzIdentML. The count of database proteins (PDH) in mzIdentML is largely irrelevant and should not usually be reported in a manuscript, for the reasons described in Section 5.1. For groups wishing to implement

mzIdentML now, it is possible to include all of the terminology described in ref. 14 in both mzIdentML 1.1 and 1.2 – as this has been done in a backwards compatible manner. The extra terminology for protein grouping is enforced in mzIdentML 1.2 but not in mzIdentML 1.1, but would be considered best practice.

A second standard format from PSI is mzTab,[15] which is a simpler representation of either identification or quantification results, in a flat-file (tab-separated text) format, suitable for visualisation in a Spreadsheet or statistical software. In mzTab support for protein grouping has been included to the extent that each row of results (in the Protein section of a file), has a nominated protein accession – assumed to be the representative protein of the group. A second cell of data can be included called "ambiguity members" where accessions for same-set proteins can be reported, with the implication that the row of data is reported for a group made up of the main accession and those ambiguity members together.

5.4 Conclusions

This chapter has summarised current algorithms and implementations for inference of protein identification, from peptide identification results in LC-MS/MS proteomics workflows. It is now accepted that all high quality studies should include some form of intelligent (*e.g.* parsimonious) protein grouping, unless extensive pre-separation of proteins has occurred (such as the use of two-dimensional gel electrophoresis). Proteomics researchers should have a reasonable level of awareness of the process performed by the main software packages, as this stage does have impact on the final list of proteins produced, and thus implications for downstream conclusions to be drawn from the data. Unlike the stage of peptide identification where straightforward metrics can be used to compare performance between different software packages (such as counting peptide identifications at 1% FDR, assuming unbiased calculation), in protein inference comparing the quality of different approaches is more difficult. In some situations, for example in the case of bacterial proteomics where gene families are less common (and there is no alternative splicing), the choice of protein inference approaches will make little or no difference on the final results, as long as high-quality gene models exist. However, in proteome analyses on complex protein sets with high sequence redundancy due to extensive alternative splicing or paralogues, considerable differences in the results could be introduced by the choice of protein inference engine alone, even accounting for the same search engine being used in different workflows. While higher counts of PSMs or peptides at a fixed FDR gives some indication of search engine performance (*i.e.* higher is probably better), given a fixed count of peptide identifications, it should not be assumed that a higher count of protein identifications is a desirable facet of protein identification software. Bioinformatics groups and proteomics research labs should strive to produce the most

parsimonious explanation of a given set of peptide identifications, to avoid biasing downstream data analysis or quantitative analysis. From the alternative approaches for protein inference and grouping presented in this chapter, there is no clear optimal solution for all approaches. However, it is a reasonable assumption that those which handle multiple levels of inference (spectral, peptide, protein, groups) in a true statistical framework are likely to be theoretically superior, assuming they have been implemented and parameterised optimally.

Acknowledgements

The author gratefully acknowledges funding from BBSRC for research in proteome informatics and computational mass spectrometry [BB/L024128/1, BB/M025705/1, BB/L005239/1, BB/K01997X/1].

References

1. A. I. Nesvizhskii and R. Aebersold, Interpretation of Shotgun Proteomic Data: The Protein Inference Problem, *Mol. Cell. Proteomics*, 2005, **4**, 1419–1440.
2. A. I. Nesvizhskii, A. Keller, E. Kolker and R. Aebersold, A Statistical Model for Identifying Proteins by Tandem Mass Spectrometry, *Anal. Chem.*, 2003, **75**, 4646–4658.
3. The UniProt Consortium, UniProt: a hub for protein information, *Nucleic Acids Res.*, 2015, **43**, D204–D212.
4. V. R. Koskinen, P. A. Emery, D. M. Creasy and J. S. Cottrell, Hierarchical Clustering of Shotgun Proteomics Data, *Mol. Cell. Proteomics*, 2011, **10**(6), M110.003822.
5. J. Cox and M. Mann, MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification, *Nat. Biotechnol.*, 2008, **26**, 1367–1372.
6. F. Ghali, R. Krishna, P. Lukasse, S. Martínez-Bartolomé, F. Reisinger, H. Hermjakob, J. A. Vizcaíno and A. R. Jones, Tools (Viewer, Library and Validator) that Facilitate Use of the Peptide and Protein Identification Standard Format, Termed mzIdentML, *Mol. Cell. Proteomics*, 2013, **12**, 3026–3035.
7. N. Gupta and P. A. Pevzner, False Discovery Rates of Protein Identifications: A Strike against the Two-Peptide Rule, *J. Proteome Res.*, 2009, **8**, 4173–4181.
8. K. Meyer-Arendt, W. M. Old, S. Houel, K. Renganathan, B. Eichelberger, K. A. Resing and N. G. Ahn, IsoformResolver: A Peptide-Centric Algorithm for Protein Inference, *J. Proteome Res.*, 2011, **10**, 3060–3075.
9. A. Keller, A. I. Nesvizhskii, E. Kolker and R. Aebersold, Empirical Statistical Model To Estimate the Accuracy of Peptide Identifications Made by MS/MS and Database Search, *Anal. Chem.*, 2002, **74**, 5383–5392.

10. M. W. Bern and Y. J. Kil, Two-Dimensional Target Decoy Strategy for Shotgun Proteomics, *J. Proteome Res.*, 2011, **10**, 5296–5301.
11. A. R. Jones, M. Eisenacher, G. Mayer, O. Kohlbacher, J. Siepen, S. Hubbard, J. Selley, B. Searle, J. Shofstahl, S. Seymour, R. Julian, P.-A. Binz, E. W. Deutsch, H. Hermjakob, F. Reisinger, J. Griss, J. A. Vizcaino, M. Chambers, A. Pizarro and D. Creasy, The mzIdentML data standard for mass spectrometry-based proteomics results, *Mol. Cell. Proteomics*, 2012, **11**, M111.014381.
12. G. Mayer, L. Montecchi-Palazzi, D. Ovelleiro, A. R. Jones, P.-A. Binz, E. W. Deutsch, M. Chambers, M. Kallhardt, F. Levander, J. Shofstahl, S. Orchard, J. Antonio Vizcaíno, H. Hermjakob, C. Stephan, H. E. Meyer and M. Eisenacher, The HUPO proteomics standards initiative- mass spectrometry controlled vocabulary, *Database*, 2013, **2013**.
13 L. Montecchi-Palazzi, S. Kerrien, F. Reisinger, B. Aranda, A. R. Jones, L. Martens and H. Hermjakob, The PSI semantic validator: A framework to check MIAPE compliance of proteomics data, *Proteomics*, 2009, **9**, 5112–5119.
14. S. L. Seymour, T. Farrah, P. A. Binz, R. J. Chalkley, J. S. Cottrell, B. C. Searle, D. L. Tabb, J. A. Vizcaino, G. Prieto, J. Uszkoreit, M. Eisenacher, S. Martinez-Bartolome, F. Ghali and A. R. Jones, A standardized framing for reporting protein identifications in mzIdentML 1.2, *Proteomics*, 2014, **14**, 2389–2399.
15. J. Griss, A. R. Jones, T. Sachsenberg, M. Walzer, L. Gatto, J. Hartler, G. G. Thallinger, R. M. Salek, C. Steinbeck, N. Neuhauser, J. Cox, S. Neumann, J. Fan, F. Reisinger, Q. W. Xu, N. Del Toro, Y. Perez-Riverol, F. Ghali, N. Bandeira, I. Xenarios, O. Kohlbacher, J. A. Vizcaino and H. Hermjakob, The mzTab data exchange format: communicating mass-spectrometry-based proteomics and metabolomics experimental results to a wider audience, *Mol. Cell. Proteomics*, 2014, **13**, 2765–2775.
16. B. C. Searle, Scaffold: A bioinformatic tool for validating MS/MS-based proteomic studies, *Proteomics*, 2010, **10**, 1265–1269.
17. O. Serang, The Probabilistic Convolution Tree: Efficient Exact Bayesian Inference for Faster LC-MS/MS Protein Inference, *PLoS One*, 2014, **9**, e91507.

CHAPTER 6

Identification and Localization of Post-Translational Modifications by High-Resolution Mass Spectrometry

RUNE MATTHIESEN*[a] AND ANA SOFIA CARVALHO[a]

[a]Computational and Experimental Biology Group, CEDOC - CHRONIC DISEASES RESEARCH CENTER, NOVA Medical School, Faculdade de Ciências Médicas, Universidade Nova de Lisboa, Campo dos Mártires da Pátria, 130, 1169-056 Lisboa, Portugal
*E-mail: rune.matthiesen@nms.unl.pt

6.1 Introduction

Current estimates suggest that there are more than 200 possible *in vivo* post-translational modifications (PTMs).[1,2] In addition, an even larger number of chemical modifications can be introduced either intentionally (*e.g.* for stable isotope based quantitation[3,4]) or unintentionally during sample preparation for mass spectrometry (MS) analysis (*e.g.* oxidation of methionine, deamidation of asparagine and glutamine). Methionine oxidation and deamidation constitute chemical modifications which in most cases should be

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

taken into account in database dependent searches. More modifications are being discovered on a regular basis such as crotonylation,[5] succinylation,[6] 3-phosphoglyceryl-lysine[7] and prokaryotic ubiquitin-like protein (Pub).[8]

PTMs introduce either a negative or a positive mass shift to the canonical peptide. This leads to a delta mass shift in MS1 scans. In MS2 scans the fragment ions that still contain the modification will likewise be shifted by a delta mass. It is the delta mass shift that makes mass spectrometry a powerful methodology for studying PTMs. For example, for the acetylated peptide ASTEVEacK all the b-ions (N-terminal fragments) from b1 to b6 will not be shifted whereas all the y-ions (C-terminal fragments) and b7 will be shifted by a delta mass of 42.0106 m/z. Mass spectrometry analysis is therefore able to site localized PTMs in proteins if the ion series (typically a, b and y-ions for higher energy collision dissociation (HCD) and collision induced dissociation (CID) data) covers the modification site and ideally the full peptide sequence. Note that it is not necessary to obtain a full y-ion or b-ion series to determine site localization. Frequently the combined assignment of the ion series provides fairly good sequence coverage. The theoretical delta mass introduced by different PTMs is available from several databases (see Subsection 6.3.2). A subset of PTMs such as phosphorylation are labile in CID and HCD which means that a CID–HCD MS2 spectrum of a phosphopeptide often contains a large neutral loss peak at m/z: $[M + nH]^{n+}/n - 79.9663$ or $[M + nH]^{n+}/n - 97.9769$, where $[M + nH]^{n+}/n$ is the n protonated parent ion mass. The consequence of this neutral loss of the phospho-group or phospho-group plus H_2O gives a-, b- and y-ion series that correspond to the canonical peptide and therefore provide no information about the phospho- site localization. Acquiring complementary ECD–ETD spectra can resolve the stability related issue but often at the cost of lower sensitivity. A site's occupancy or stoichiometry of a modification, defined as the fraction of protein molecules that are modified at a specific site by a specific modification, ideally requires that both the modified and a number of unmodified peptides from the specific protein are observed together with known concentrations of stable isotope labeled versions of the peptides. However, relative changes in PTM occupancy across samples can be detected by label-free quantitation by comparing ratios of ion counts from the modified peptide *versus* the canonical peptide from different samples.

PTM patterns change upon external stimuli, development stages, diseases, genotype, subcellular localization and even more fluctuate in a spatiotemporally regulated manner in specific subcellular processes (*e.g.* autophagy and proteasome degradation). A subset of PTMs changes occupancy on a short time scale upon external stimuli[9] (*e.g.* acetylation, phosphorylation, ubiquitination, *etc.*) whereas others change on a longer time frame such as glycosylation in cancer and some glycosylation patterns that are fixed for life such as blood types, which is genotype specific.

Many diseases demonstrate modulation of the occupancy of a large number of *in vivo* post-translational modifications,[10] *e.g.* proteolytic cleavages,[11] ubiquitin, acetylation,[12–14] methylation,[15,16] phosphorylation,[17,18] glycosylation,[15,19] redox modifications[20,21] and prenylation[10] have all been implicated in cancer (see Table 6.1). In other words PTMs constitute potential

Table 6.1 Post-translational modifications and their association with different cancer types.

Post-translational modification	Prominent examples
Acetylation	Histone deacetylase inhibitors
Methylation	Arginine methyltransferases (PRMTs) and protein lysine methyltransferases and demethylases
Proteolytic cleavage	Proteasome inhibitors (*e.g.* Bortezomib); ADAMs in drug resistance, cancer stem cells, cell migration and invasion
Ubiquitin and SUMOylation	DUBs and E3 ligases
Phosphorylation	Kinase inhibitors (Receptor tyrosine kinases/PI 3-kinase/Akt/mTOR/Ras/Raf/MEK/ERK, MEKK/MKK/JNK, and JAK/STAT)
Glycosylation	GP73, CD44, galectins,CA125, CA19-9, MUC1, MUC4, MUC16, prostate-specific antigen, osteopontin, Sialyl Lewis A and Lewis X
Redox modification	Aberrant induction of signaling networks triggered by reactive oxygen species in *e.g.* cardiovascular diseases and cancer
Prenylation	Activation of GTPases such as Ras, Rho, and G-proteins coupled
Poly(ADP-ribosyl)ation	Genotoxic stress, cell division and survival

biomarkers and surrogate markers to complement clinical methods which strictly only use proteins or biomarkers from genomic platforms. Furthermore, a number of pharmaceutical drugs target specific PTMs emphasizing the importance of PTMs.

6.2 Sample Preparation Challenges

A plethora of technical challenges is associated with identification and site localization of PTMs. These challenges must be taken into consideration when performing computational analysis and therefore will be shortly reviewed here. Table 6.2 provides an overview of some of the challenges associated with PTM identification and site localization and suggestions to alleviate the problems from an experimental or computational point of view.

It is becoming increasingly clear that MS-based proteomics challenges the original idea of few individually important post-translational modifications being key regulators of a biological process such as cell cycle, development or detection and response to stimulus (For example, MS nowadays provides thousands of phosphorylation, acetylation and ubiquitylation sites in a single study.).[1] Clearly, these gear our perspectives toward the importance of studying groups of entities rather than key regulators, which consequently requires the development of new experimental and computational methods.

N-linked glycosites and glycosylphosphatidylinositol (GPI) anchors occur at high stoichiometries and are typically irreversible. In contrast, reversible PTMs such as acetylation,[22] phosphorylation,[23,24] ubiquitinylation[25] and SUMOylation[26] are typically reported to display low stoichiometries which is linked to the

Table 6.2 Overview of challenges in MS-based PTM research. The first column lists technical problems and the second column lists technologies that alleviate the problem.

Challenge	Methods to alleviate problem
Near isobaric masses	Synthetic peptides, Gaussian weight, alternative fragmentation methods
Substochiometry	PTMs enrichment protocols (antibodies TiO_2, lectins, TUBEs, *etc.*)
Sample degradation	Protease inhibitors, TUBEs, PhosSTOP™, Histone deacetylace inhibitors
Stability in gas phase especially in fragmentation step	Combine CID–HCD with ETD, alternative protein digestion methods
Artifical modifications	Chemical modifications (*e.g.* cystein modifications such as iodoacetamide)
Detectability	Enrichment, artificial modifications, enzymatic and chemical cleavage
Complexity	Enzymatic and chemical cleavage of modifications
Size	Enzymatic and chemical cleavage of modifications
Peptide solubility	Consider buffer properties used for enrichment and reverse phase material
Site localization	Gaussian weight, ETD, high mass accuracy and resolution in MS and MS/MS
Cross talk between PTMs	Open or semi-open search combined with label-free quantitation, serial enrichment

fact that they play a key regulatory role. The low stoichiometries require specialized enrichment strategies prior to MS analysis. Enrichment-based strategies increase significantly the number of identified sites but have the drawback that occupancy is not easily determined and the interplay with other modifications is lost. Nevertheless, serial enrichment allows integrated enrichment and analysis of phosphorylation, ubiquitination and acetylation.[27]

Additional difficulties with PTM identification are caused by complexity and stability of specific modifications. PTMs such as GPI anchors,[28] Poly(ADP-ribosyl)ation[29] and phosphorylation are labile within the mass spectrometer, especially in the fragmentation step impairing modification site determination. Combining CID–HCD and ETD can enable site location of labile modifications. The use of Gaussian weights in scoring functions also improves site location of PTMs and certainty of correct identification.[30] In a similar way, the use of high mass accuracy in both MS and tandem mass spectrometry (MS/MS) also improves identification and site location.

Proteins can be modified by other proteins *e.g.* ubiquitylation, SUMOylation and NEDDylation. Recently it has become evident that cross branching between ubiquitin and ubiquitin-like proteins is possible and additionally ubiquitin and ubiquitin-like proteins can become modified by, for example, phosphorylation adding to the complexity of PTM studies.[31] Glycosylation also forms highly complex structures with heterogeneous compositions of glycosylation units and multiple possibilities for branching.[32] Furthermore, the PTM component of the protein can cause the total mass of the peptide to be outside the measureable range within the instrument settings used for

large scale proteomics. This means that enzymatic removal leaving a small residual component of the full modification can be a strategy that allows site mapping of such modifications.[33] This strategy of cleaving of a major component of the modification has been used for glycosylation, GPI-anchors, ubiquitin and ubiquitin-like modifiers.

Different buffers used for cellular lysis and subcellular fractionation can also affect the final outcome of enrichment strategies and is worth considering.[34] This can for example be caused by a buffer's effect on solubility of peptides and affinity to C18 resins (*e.g.* phospho peptides are in general more hydrophilic than canonical peptides).

Chemical modifications generated during sample preparation can cause ion suppression leading to lower identification rate. In the worst case scenario the spectra from these chemical modifications can be matched to incorrect peptides if not considered during the analysis. PTMs that occur transiently in the cell such as intermediates in redox reactions can be stabilized by chemical reagents, for example, free thiols at cysteine can be blocked (*e.g.* with NEM, MMTS or iodoacetamide). Subsequently S-nitrosylation is reduced with ascorbate and the newly exposed thiols labeled with a thiol-reactive biotin which can be used for enrichment prior to MS analysis.[35,36]

Enrichment strategies for phosphorylation are in general more developed than for other PTMs. For example, antibodies raised against specific phosphorylation motifs, reflecting phosphorylation sites from a specific kinase, can be used for enrichment.[37] However, most studies still report the whole phosphoproteome commonly using titanium dioxide (TiO_2) chromatography for selective phosphopeptide enrichment.[38]

In conclusion, MS-based study of PTMs still holds major challenges making it an interesting research topic. Efficient experimental protocols and enrichment strategies require development for a large number of modifications.

6.3 Identification and Localization of Post-Translational Modifications

6.3.1 Computational Challenges

PTM identification faces several challenges from computational and experimental sides. Solving these challenges involves iteration between both the computational and experimental sides since they are directly linked. The diverse experimental methodologies require specific computational methods optimized for the specific tasks. Frequently this is not available and adaptation of computational methods optimized under other conditions and assumptions are adapted in the best possible way to provide pragmatic solutions. Furthermore, few reference data sets exist to validate experimental and computational methods, although, a reference data set of synthetic phosphopeptides analyzed by MS/MS is available.[39]

Matching peptides against experimental spectra is an optimization process aiming at the best possible solution given predefined constraints. The number of peptide sequences to score against a spectrum to identify the PTM site in a given peptide sequence can be formulated as:

$$N = \prod_{i=1}^{m} 2^{M_i}$$

where N is the number of peptides and i iterates over possible modifications m for the given peptide. M_i is the number of possible positions for the modification i., *e.g.* a peptide with one phosphorylation and 10 possible sites (typical Ser, Thr and Tyr) results in $2^{10} = 1024$ possible peptides to match a given spectrum. For a small number of modifications, less than five, the iteration shown is possible but becomes computationally intensive when more than five modifications are considered. However, for more modifications the combinatorial problem becomes intractable and consequently an MS-Alignment-based algorithm for "blind" spectral search has been proposed.[40,41] The central idea of the MS-Alignment algorithm is based on allowing mass shifts, corresponding to modifications or mutations, to obtain a best possible match between experiment and theoretical spectra. These solutions can efficiently be found by adapting dynamic programming algorithms used to align sequence data. Computational methods based on "semi-blind–open" searches have also been proposed.[42] The scoring of a peptide to a spectrum consists of matching theoretical fragment ions (see Section 6.3.4) against the observed fragments ions in MS2 scans using a mass interval in m/z or ppm to define matches. Many scoring functions have been proposed and can roughly be divided in functions that only consider the observed fragment ions matches and functions that both consider matched and unmatched peaks.[43,44] A simple scoring function is the cross correlation score which basically counts the number of matched ions, such as the correlation score.[45] More advanced scoring functions include terms that consider the intensity patterns,[46] the continuation of the ion series,[47] Gaussian weights for the accuracy of fragment ion matches[43] and the isobaric ambiguity of the matched masses.[43,44] In addition to assigning a score to the peptide spectrum match (PSM) some software also provide a site location score or probability which provides a measure for how likely the correct position of the modification in the peptide is based on spectral information.[48] Gaussian weights in scoring functions also improve the correct assignment of PTMs.[30]

One challenging problem is to statistically estimate the accuracy of the modification site. FDR estimation using reverse, permutated or random protein sequence databases is now generally accepted as a way to compare different search engine results for peptide and protein identifications[49] (see Chapter 4). Unfortunately this concept does not translate well to site localizations of modifications. Although some methods have been proposed to estimate false localization rate (FLR) they have not proved practically useful or achieved general acceptance.[50,51] The role of peak picking on the accuracy of modification site determination has been discussed[52] but using raw

Table 6.3 Overview of databases containing post-translational modifications assigned to protein amino acid residue positions. The annotation in these databases focus on the biological impact of the modifications.

Database	Webpage
dbPTM	http://dbptm.mbc.nctu.edu.tw/
UniProtKB	http://www.uniprot.org/
PhosphoSitePlus	http://www.phosphosite.org

data is preferable. Providing probability estimates for site localization is difficult because ion series frequently do not cover the full peptide sequence or the part of the sequence containing the potential modification sites. Even if the ion series covers all the potential modification sites the score difference between two potential sites are often minor and can be attributed to assigning a single extra fragment ion which could be noise or a fragment from another peptide that was co-fragmented.[52] Strategies for estimating the reliability of modification site localization can be divided into two main strategies: 1) assess the chance of a given peak that allows site determination to have been matched at random[53–58] and 2) calculate a search engine score difference between peptide identifications with different site localizations.[50–52]

Protein PTM databases are useful to compare obtained results from an MS study and can also serve as a way to minimize the search space by restricting the search of protein residue sites to annotated modifications. Although computationally simpler it prevents assignment of unknown sites. Table 6.3 provides a list of commonly used databases containing PTMs assigned to proteins. dbPTM integrates data from 14 different public databases of post-translational modifications and is therefore currently the most complete database of PTMs.

The information deposited in PTM databases is confined commonly to the modified site in the protein sequence, however a site's occupancy or stoichiometry of a modification, defined as the fraction of protein molecules that are modified at a specific site by a specific modification should be provided in case this information can be obtained in the study.[9,59] Consequently, PTM databases should include information on the data source in such cases.

6.3.2 Annotation of Modifications

Table 6.4 provides an overview of databases containing annotation of modifications relevant for MS identifications. For example, important information about delta mass of the modification, diagnostic ions, neutral losses and the chemical compositions (useful for calculating isotope distributions) are available. Additionally, UniMod and PSI-MOD[60] (see Chapter 11) also provides information on which amino acid residues a given modification can occur on and if the modification is biologically relevant, a chemical artefact or intentionally introduced modifications to either stabilize a residue or for quantitative purposes.

Table 6.4 Databases containing mass spectrometry relevant annotation of protein modifications. The annotations in these databases focus on technical aspects that are necessary to correctly annotate modified peptides to spectra.

Modification databases	Webpage
UniMod database	http://www.unimod.org/
UniProt	http://www.uniprot.org/docs/ptmlist
ResID	http://pir.georgetown.edu/resid/
PSI-MOD	http://psidev.cvs.sourceforge.net/viewvc/psidev/psi/mod/data/PSI-MOD.obo

As explained in Chapter 3, the identification of proteins from MS/MS spectra and their associated modifications is usually carried out by matching spectra data against a protein sequence database using database-dependent search engines. Several individual efforts have been made to provide the research community with algorithms for searching MS/MS data such as MaxQuant,[61] Sipros,[62] X!tandem[63] and mVEMS.[42] We have been able to successfully test the aforementioned programs and we use them on a regular basis to search MS data. This list of software projects is not complete and many more programs have been proposed in the literature.[64] However, we can state that according to our searches this list of software provides similar identification given similar search settings and false discovery rate cut off.

6.3.3 Common Post-Translational Modifications Identified by Mass Spectrometry

The aim of the analysis described herein was to define a set of common modifications that typically can be identified in MS/MS data sets from instruments using high resolution in MS and MS/MS. The search was performed using mVEMS, which is under development to be provided to the research community.[42,65] Although the presented results were obtained by analyzing unlabeled proteomes that have not been subjected to any type of enrichment for post-translational modifications we expect that the obtained result is also relevant for other experimental methodologies. For example, the most abundant modifications identified would still be relevant to search for in combination with phosphorylation in a study where enrichment of phosphorylated peptides is performed prior to MS analysis.

We observed a considerable number of modifications that are considered as artefacts from sample preparation such as deamidation, methionine oxidation, carbamoylation and carbamidomethylation (CAM) of methionine, lysine and N-terminus (see Figure 6.1). Some of these artefact modifications are even more frequent than methionine oxidation. Single amino acid polymorphisms were also considered in the analysis and we observe that especially lysine ↔ arginine substitutions occur frequently. Serine and threonine formylation are also among the most frequently occurring. These are also most likely artefacts

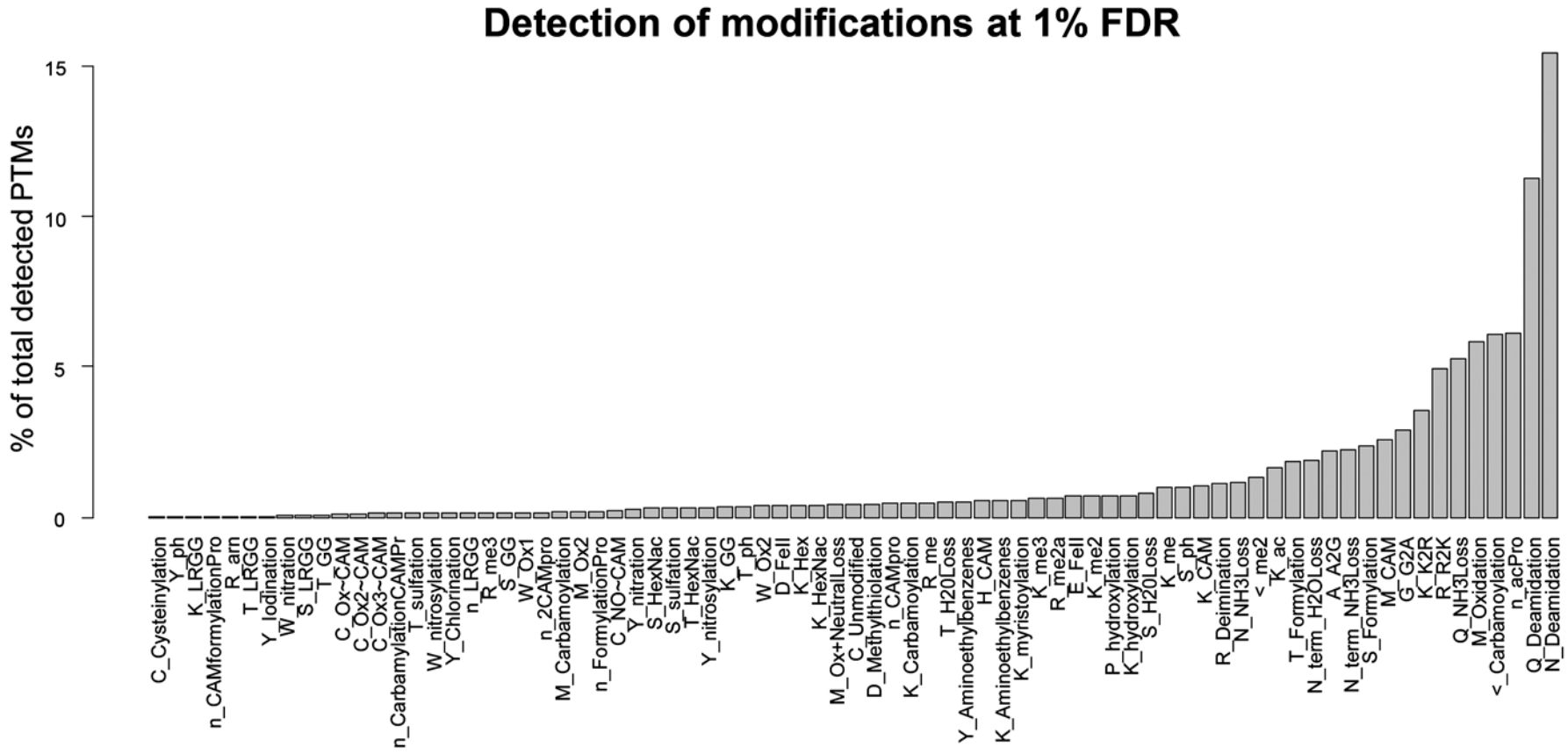

Figure 6.1 Relative identification frequencies of post-translational modifications and single amino acid mutations obtained by analyzing the cytosol fraction of a cancer cell line. A 1% FDR cut-off was applied using the peptide score distribution from the correct protein sequences and amino acid permutated protein sequences.

however it has recently been suggested that MS identified lysine formylation on histone proteins might not be fully explained as an artefact.[66]

6.3.4 Validation of Results

The output of a database dependent search is often two matrices with quantitative values for all identified peptides across the analyzed samples. That is, one for spectral counts and one for extracted ion counts. These matrices can conveniently be analyzed in statistical software such as R, SASS or even Microsoft Excel. If a modified peptide is found significantly regulated then this might be of interest. However, it is often a good idea to validate the quality of a modified peptide's spectral assignment before other types of validation or follow up experiments are carried out, since we find that especially modified peptides in the result files outputted from database dependent search engines are not always reliable. Chapter 4 discusses a large number of statistical methods for validating the quality of PSMs. Although statistical methods development is essential most of these statistical methods are either functions of the specific PSM score or functions of the distribution of the PSM scores. This means they correlate with the score and therefore multiple statistical models with slightly different assumptions provide little additional value in terms of deciding the correctness of a modified peptide assignment to a spectrum. However, for PSMs representing modified peptides additional statistical models that measure the probability that the corrections of the assigned location of the modifications are justified. The binomial function has been proposed for calculating probability of correctness of site location. However, the binomial

Published on 15 November 2016 on http://pubs.rsc.org | doi:10.1039/9781782626732-00116

function applied on assigned fragment masses will again correlate with any other PSM score. Ideally probabilities for modification sites location should be calculated from the score distribution of alternative modification sites obtained by scoring the raw spectrum. The gold standard for validating a peptide assigned MS spectrum is by performing MS analysis of the chemical synthesized peptide including the modification. Additionally, Bunkenborg *et al.*[2] lists a number of matters that can be considered for validating spectra assignments and we will therefore not discuss this topic in detail but highly recommend reading this chapter as well. Nevertheless, we provide a brief review herein. A quick validation test is to see if the parent ion mass of peptide spectra assignments fits with the expected number of basic residues (A short peptide with only one basic residue (*e.g.* one Lys or Arg) is unlikely to be detected as a triple charged parent ion.). Plotting the raw annotated spectrum is also highly recommended and journals such as Molecular and Cellular Proteomics often ask authors to provide raw annotated spectra of all spectra assignments (see Figure 6.2). Theoretical ions can be calculated by the following equations (more details and mass tables can be found in ref. 2).

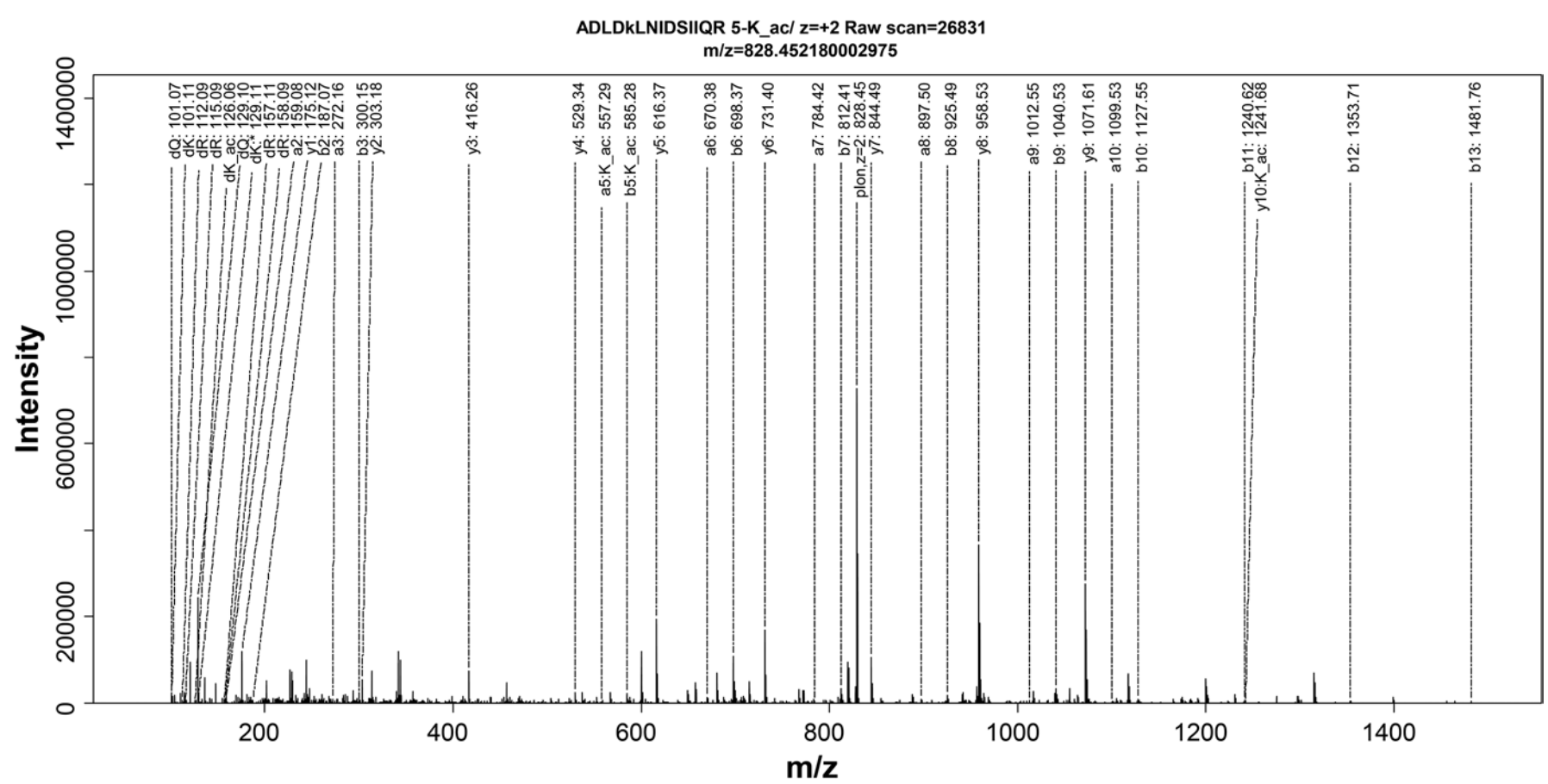

Figure 6.2 The peptide ADLDacKLNIDSIIQR containing an acetylated lysine annotated to a raw MS spectrum (intensity counts *versus* m/z). The labeling of fragment ions assumes single charged fragments unless otherwise provided in the labels (*e.g.* the parent ion is labeled "pIon, $z = 2$" to indicated that this is a doubly charged ion). Labels starting with "d" are diagnostic ions for either amino acids or amino acid modifications. The theoretical masses of the ion series a-, b- and y-ions were matched against the observed masses using a threshold of 0.005 m/z as well as the correct charge state of the peak was validated automatically to define a successful match for a fragment ion. Note y-ions are numbered from the C-terminal side whereas a- and b-ions are numbered from the N-terminal side. This means that y1 corresponds to the ion "R" (see equation in Section 6.3.4 for CID–HCD fragments).

The parent ion $[M + H]^+$ is given by the sum

$$[M+H]^+ = m_N + m_C + m(H^+) + \sum_{i=1}^{n} m_i$$

where m_N (normally hydrogen) and m_C (normally hydroxyl group) are the mass of the N- and C-terminal group respectively. $m(H^+)$ is the mass of the proton and m_i are the residue masses. The "mass-to-charge ratio" of the monoisotopic peak of an n multiple protonated peptide is given by $[M + nH]^{n+}/n$.

The theoretical fragment ions in MS/MS for MS scans labeled as collision induced dissociation (CID) or high-energy collision dissociation (HCD) can be calculated using the following equations:

a-ions (normally $K = 26.9864$):

$$a_i = m_N - m(CO) - m(e^-) + \sum_{i=1}^{n} m_i = K + \sum_{i=1}^{n} m_i$$

b-ions (normally $K = 1.007276$):

$$b_i = m_N - m(e^-) + \sum_{i=1}^{n} m_i = K + \sum_{i=1}^{n} m_i$$

y-ions (normally $K = 19.01784$):

$$y_i = m_C - m(H) - m(H^+) + \sum_{i=1}^{n} m_i = K + \sum_{i=1}^{n} m_i$$

where i iterates over the peptide sequence with length n. Note the indexing starts from the N-terminal for a- and b-ions and from the C-terminal for y-ions. The following equation can be used to calculate the corresponding b- or y-ion given the parent ion mass and either one of the other ions.

$$b_i + y_{n-i} = MH^+ + m(H^+) = MH^+ + 1.007276$$

Fragment ions in MS/MS scans labeled as electron-capture dissociation (ECD) or electron-transfer dissociation (ETD) can be calculated using the following equations:

c-ions (normally $K = 18.03383$):

$$c_i = m_N + m(H^+) + m(NH_2) + \sum_{i=1}^{n} m_i = K + \sum_{i=1}^{n} m_i$$

z + 1 -ions (normally $K = 2.99912$):

$$z_i + 1 = m_C + m(H^+) - m(NH) + \sum_{i=1}^{n} m_i = K + \sum_{i=1}^{n} m_i$$

ETD–ECD spectra often have charge reduced ions with no further fragmentation.[2] For example, $[M + 3H]^{3+}/3$ can have corresponding charged reduced ions at m/z $[M + 3H]^{2+}/2$ and $[M + 3H]^{+\bullet}$, The dot indicates that it is an odd electron ion.

In addition to the ion series MS/MS spectra of peptides also contain a number of diagnostic ions (peaks with labels starting with "d" in Figure 6.2, *e.g.* dK_Ac at *m*/*z* 126.06) which are fragments from specific amino acids and modifications. Diagnostic ions specific for amino acids can be found in ref. 2 and diagnostic ions for modifications can be found in databases such as UniMod (see Table 6.4). UniMod also annotates neutral losses for modifications *e.g.* neutral loss from methionine sulfoxide of 63.998 can frequently be observed in both MS and MS/MS scans.

Especially complex modifications provide complex fragmentation patterns in MS/MS that serve as additional validation information. Unfortunately computational algorithms are unable to match the quality of a skilled expert in mass spectrometry in terms of validating spectra assignments. This fact justifies proteomics specialized journals' request to provide annotated raw spectra. We briefly mentioned that applying Gaussian weights in scoring functions improve site location of PTMs and the reliability of the assignment.[30] To illustrate this we plotted the delta mass between experimental and theoretical fragments from the MS2 spectrum from Figure 6.2 (see Figure 6.3).

Linear regression on assigned fragment masses and delta mass from both b- and y-ions gives similar regression lines for the b- and y-ions. Furthermore, the delta masses are evenly distributed around the regression lines. We next assigned the tri-methylated peptide "ADLDme3KLNIDSIIQR" to the same raw spectrum in Figure 6.2 and repeated the analysis of delta masses (see Figure 6.4). The mass difference between acetylation and tri-methylation is 0.03639 (*m*/*z*).

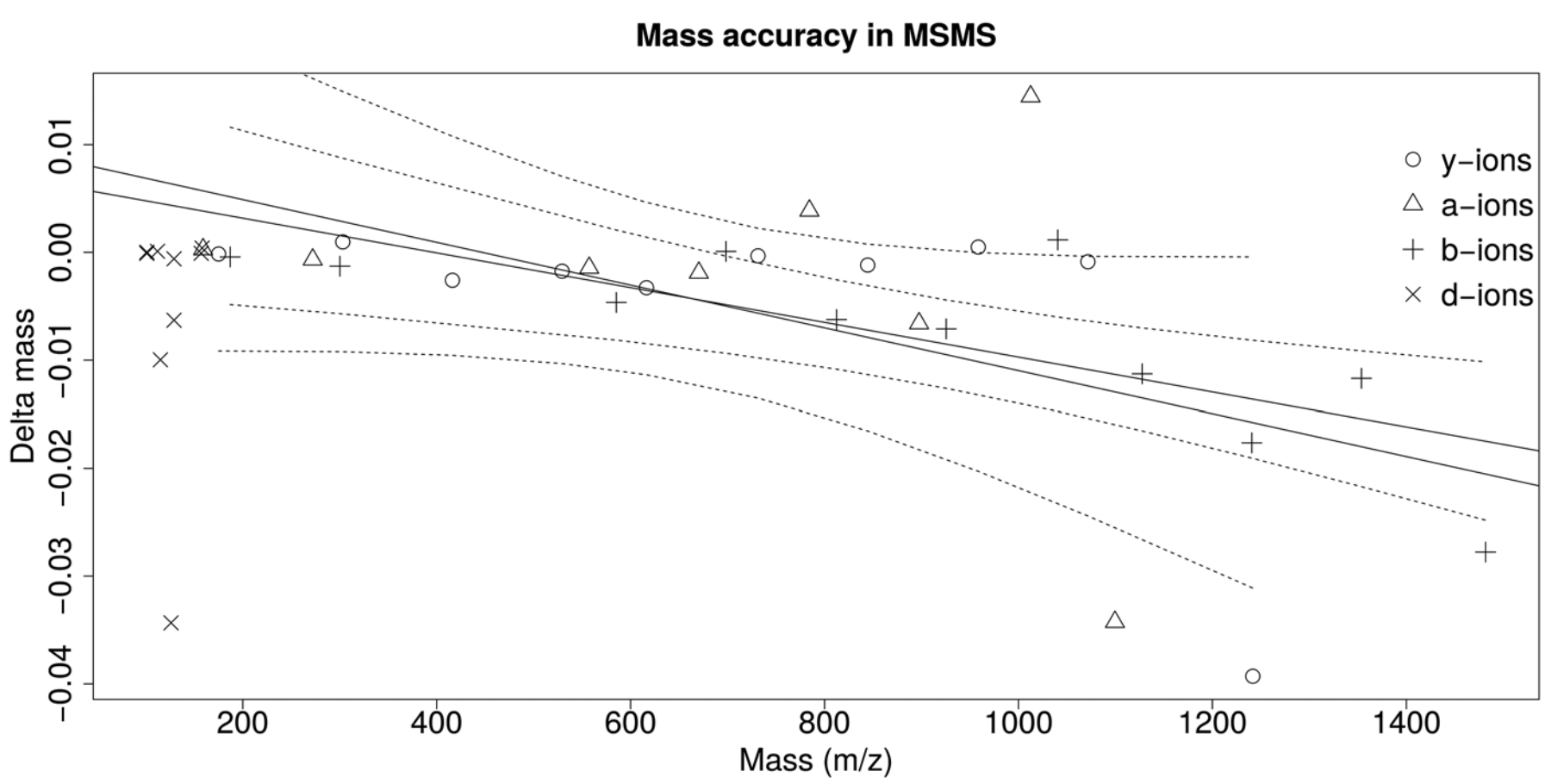

Figure 6.3 Delta mass *versus* *m*/*z* for assigned ions in MS/MS spectrum for the assignment provided in Figure 6.2. Different ion types are depicted with different symbols as indicated in the legend. Linear regression on masses *versus* delta masses was performed for both b-ion and y-ion series.

We observed that the regression lines for the b- and y-ions are now different, and especially, the delta masses for the b-ions have increased (mainly the b-ions containing the modification). Precisely how the linear regressions based on the b- and y-ions are affected depends on the position of the modifications within the peptide. A similar diagnostic plot is the density of delta masses for the two candidate peptide assignments (see Figure 6.5).

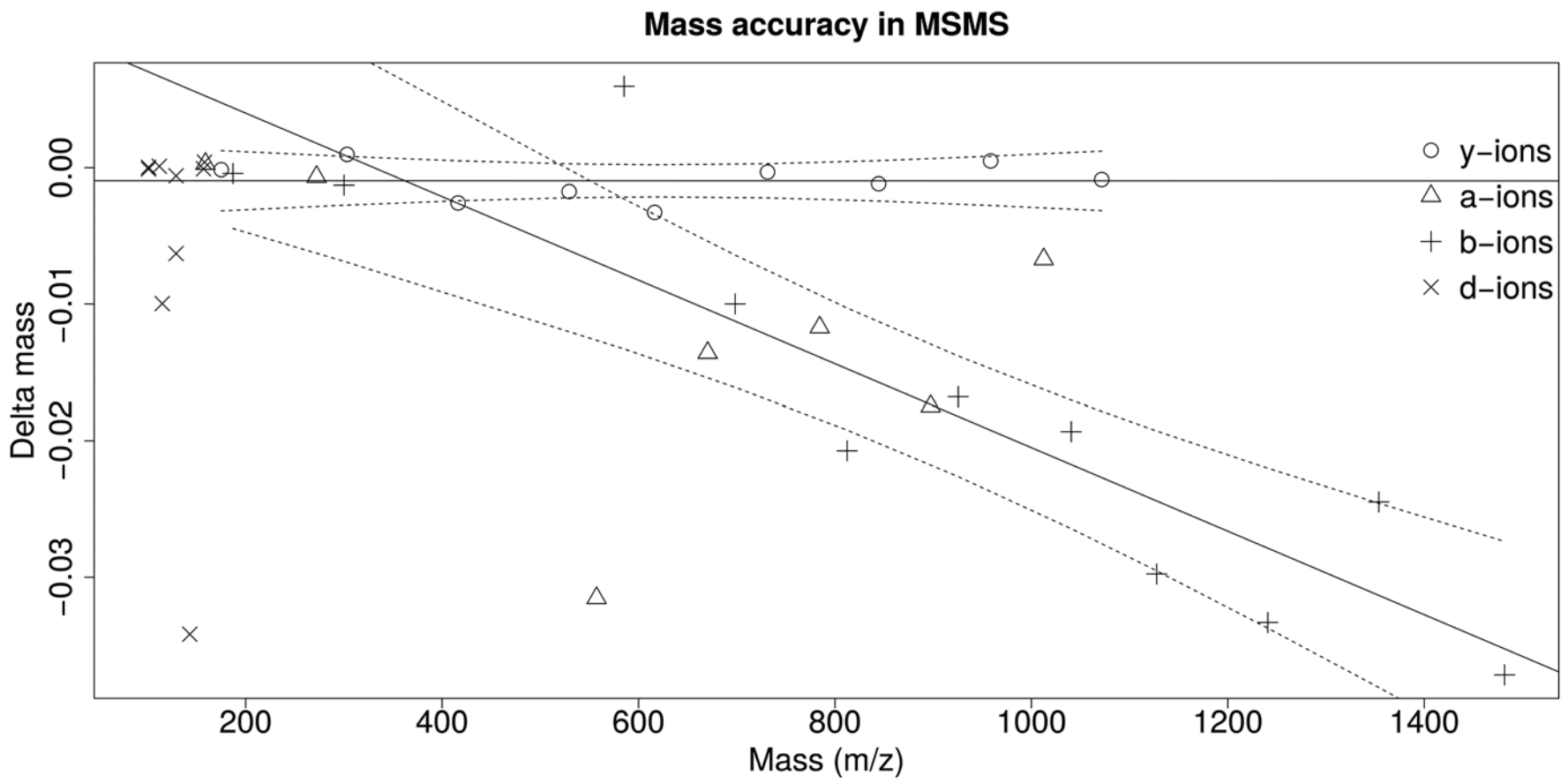

Figure 6.4 Delta masses *versus* m/z for assigned ions in MS/MS spectrum for the assignment provided in Figure 6.2 but the modification was artificially changed *in silico* to tri-methyl instead of acetylation. Different ion types are depicted with different symbols as indicated in the legend. Linear regression was performed for both b-ion and y-ion series.

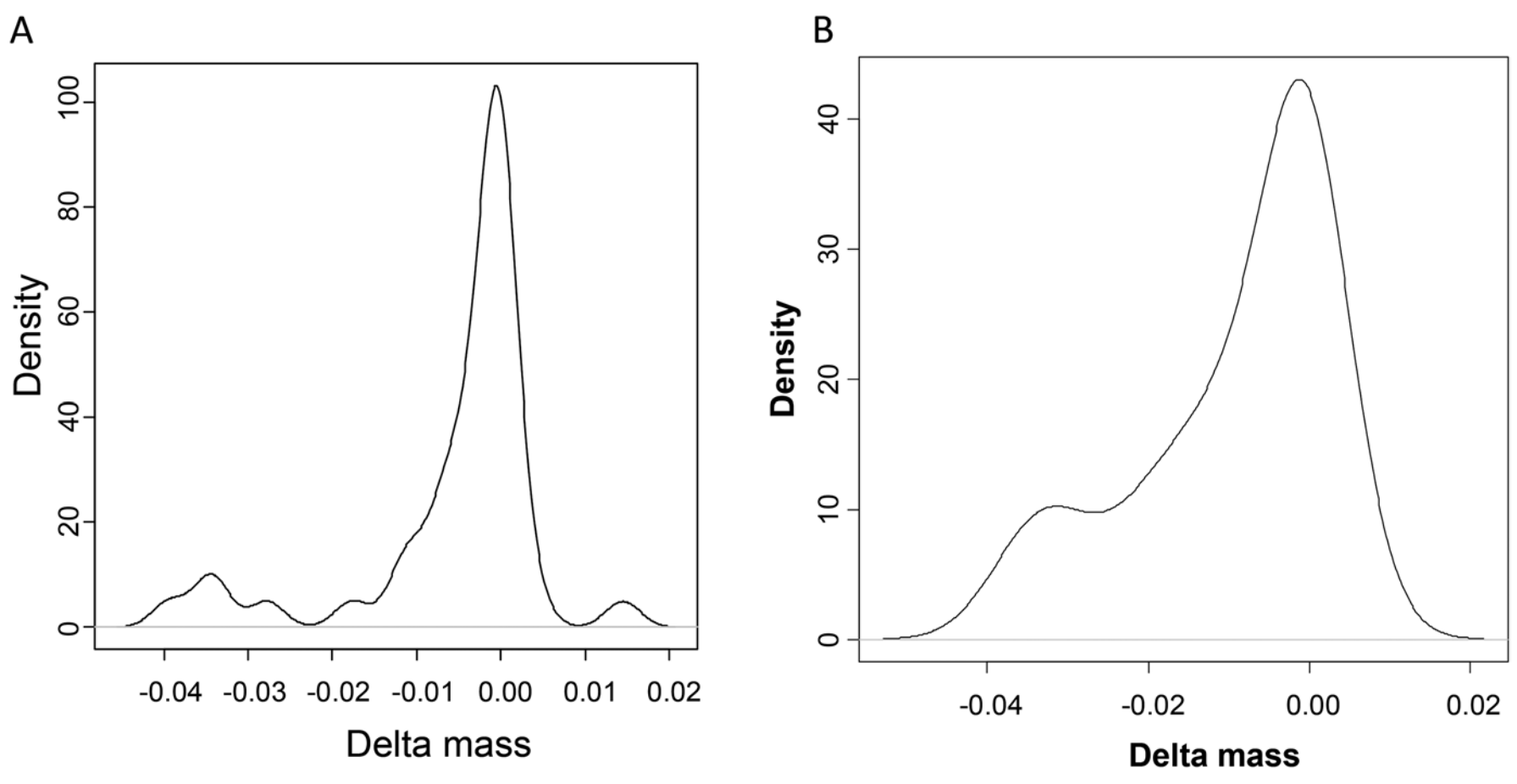

Figure 6.5 Density of delta masses obtained by assigning (A) ADLDacKLNIDSIIQR and (B) ADLDme3KLNIDSIIQR to the raw spectrum in Figure 6.2.

Published on 15 November 2016 on http://pubs.rsc.org | doi:10.1039/9781782626732-00116

The delta mass distributions are centered more or less on zero as expected for well calibrated data. It is evident that the delta mass distribution based on the tri-methylated peptide (Figure 6.5(B)) is considerably broader compared to the one based on the acetylated peptide (Figure 6.5(A)) which suggests that acetylation is the best solution.

6.4 Conclusion

We foresee that in the future it will become standard to perform database dependent searches of MS/MS spectra considering large numbers of modifications. We are currently involved in efforts to provide software applications able to perform database searches, as described in this chapter, of data obtained by high resolution mass spectrometers. The result presented clearly indicates that there is a need to consider more modifications in standard proteomics studies. It is however unclear if all modifications identified are assigned correctly to delta masses or if it is a result of other isobaric or near isobaric modifications. Even if the additional identified modifications are of no interest from a biological point of view they are important in terms of controlling the level of artefact modifications by specific experimental methodologies. Furthermore, excluding the most abundant artefact modifications in the search parameters will increase the risk of assigning the corresponding spectra to erroneous peptides.

Acknowledgements

This work was supported by the Portuguese Foundation for Science and Technology (EXPL/DTP-PIC/0616/2013). R.M. is supported by FCT investigator program 2012. A.S.C. is supported by the Portuguese Foundation for Science and Technology (FCT), financed by the European Social Funds (COMPETE-FEDER) and national funds of the Portuguese Ministry of Education and Science (POPH-QREN) fellowship SFRH/85569/2012.

References

1. J. V. Olsen and M. Mann, *Mol. Cell. Proteomics*, 2013, **12**, 3444–3452.
2. J. Bunkenborg and R. Matthiesen, *Methods Mol. Biol.*, 2013, **1007**, 139–171.
3. R. Matthiesen and A. S. Carvalho, *Methods Mol. Biol.*, 2010, **593**, 187–204.
4. R. Matthiesen and A. S. Carvalho, *Methods Mol. Biol.*, 2013, **1007**, 183–217.
5. M. Tan, H. Luo, S. Lee, F. Jin, J. S. Yang, E. Montellier, T. Buchou, Z. Cheng, S. Rousseaux, N. Rajagopal, Z. Lu, Z. Ye, Q. Zhu, J. Wysocka, Y. Ye, S. Khochbin, B. Ren and Y. Zhao, *Cell*, 2011, **146**, 1016–1028.
6. Z. Zhang, M. Tan, Z. Xie, L. Dai, Y. Chen and Y. Zhao, *Nat. Chem. Biol.*, 2011, **7**, 58–63.
7. R. E. Moellering and B. F. Cravatt, *Science*, 2013, **341**, 549–553.

8. M. J. Pearce, J. Mintseris, J. Ferreyra, S. P. Gygi and K. H. Darwin, *Science*, 2008, **322**, 1104–1107.
9. A. S. Carvalho, D. Penque and R. Matthiesen, *Proteomics*, 2015, **15**, 1789–1792.
10. K. E. Krueger and S. Srivastava, *Mol. Cell. Proteomics*, 2006, **5**, 1799–1810.
11. G. Murphy, *Nat. Rev. Cancer*, 2008, **8**, 929–941.
12. H. C. Beck, E. C. Nielsen, R. Matthiesen, L. H. Jensen, M. Sehested, P. Finn, M. Grauslund, A. M. Hansen and O. N. Jensen, *Mol. Cell. Proteomics*, 2006, **5**, 1314–1325.
13. S. Ropero and M. Esteller, *Mol. Oncol.*, 2007, **1**, 19–25.
14. M. A. Glozak and E. Seto, *Oncogene*, 2007, **26**, 5420–5432.
15. Y. Yang and M. T. Bedford, *Nat. Rev. Cancer*, 2013, **13**, 37–50.
16. Y. He, I. Korboukh, J. Jin and J. Huang, *Acta Biochim. Biophys. Sin.*, 2012, **44**, 70–79.
17. J. Reimand, O. Wagih and G. D. Bader, *Sci. Rep.*, 2013, **3**, 2651.
18. A. S. Dhillon, S. Hagan, O. Rath and W. Kolch, *Oncogene*, 2007, **26**, 3279–3290.
19. J. H. Rho, J. R. Mead, W. S. Wright, D. E. Brenner, J. W. Stave, J. C. Gildersleeve and P. D. Lampe, *J. Proteomics*, 2014, **96**, 291–299.
20. T. C. Jorgenson, W. Zhong and T. D. Oberley, *Cancer Res.*, 2013, **73**, 6118–6123.
21. W. Yang, L. Zou, C. Huang and Y. Lei, *Drug Dev. Res.*, 2014, **75**, 331–341.
22. C. Choudhary, B. T. Weinert, Y. Nishida, E. Verdin and M. Mann, *Nat. Rev. Mol. Cell Biol.*, 2014, **15**, 536–550.
23. J. Villen and S. P. Gygi, *Nat. Protoc.*, 2008, **3**, 1630–1638.
24. E. López, R. Matthiesen, I. López, K. Ashman, J. Mendieta, J. Wesselink, P. Gómez-Puertas and A. Ferreira, *J. Integr. OMICS*, 2010, DOI: 10.5584/jiomics.v1i1.40.
25. N. D. Udeshi, P. Mertins, T. Svinkina and S. A. Carr, *Nat. Protoc.*, 2013, **8**, 1950–1960.
26. I. A. Hendriks, R. C. D'Souza, J. G. Chang, M. Mann and A. C. Vertegaal, *Nat. Commun.*, 2015, **6**, 7289.
27. P. Mertins, J. W. Qiao, J. Patel, N. D. Udeshi, K. R. Clauser, D. R. Mani, M. W. Burgess, M. A. Gillette, J. D. Jaffe and S. A. Carr, *Nat. Methods*, 2013, **10**, 634–637.
28. M. J. Omaetxebarria, F. Elortza, E. Rodriguez-Suarez, K. Aloria, J. M. Arizmendi, O. N. Jensen and R. Matthiesen, *Proteomics*, 2007, **7**, 1951–1960.
29. V. Schreiber, F. Dantzer, J. C. Ame and G. de Murcia, *Nat. Rev. Mol. Cell Biol.*, 2006, **7**, 517–528.
30. R. Matthiesen, M. B. Trelle, P. Hojrup, J. Bunkenborg and O. N. Jensen, *J. Proteome Res.*, 2005, **4**, 2338–2347.
31. F. Koyano, K. Okatsu, H. Kosako, Y. Tamura, E. Go, M. Kimura, Y. Kimura, H. Tsuchiya, H. Yoshihara, T. Hirokawa, T. Endo, E. A. Fon, J. F. Trempe, Y. Saeki, K. Tanaka and N. Matsuda, *Nature*, 2014, **510**, 162–166.

32. K. W. Moremen, M. Tiemeyer and A. V. Nairn, *Nat. Rev. Mol. Cell Biol.*, 2012, **13**, 448–462.
33. P. Hagglund, R. Matthiesen, F. Elortza, P. Hojrup, P. Roepstorff, O. N. Jensen and J. Bunkenborg, *J. Proteome Res.*, 2007, **6**, 3021–3031.
34. S. S. Jensen and M. R. Larsen, *Rapid Commun. Mass Spectrom.*, 2007, **21**, 3635–3645.
35. J. Pan and K. S. Carroll, *Biopolymers*, 2014, **101**, 165–172.
36. C. I. Murray and J. E. Van Eyk, *Circ.: Cardiovasc. Genet.*, 2012, **5**, 591.
37. K. Engholm-Keller and M. R. Larsen, *Proteomics*, 2013, **13**, 910–931.
38. T. E. Thingholm, T. J. Jorgensen, O. N. Jensen and M. R. Larsen, *Nat. Protoc.*, 2006, **1**, 1929–1935.
39. H. Marx, S. Lemeer, J. E. Schliep, L. Matheron, S. Mohammed, J. Cox, M. Mann, A. J. Heck and B. Kuster, *Nat. Biotechnol.*, 2013, **31**, 557–564.
40. S. Tanner, H. Shu, A. Frank, L. C. Wang, E. Zandi, M. Mumby, P. A. Pevzner and V. Bafna, *Anal. Chem.*, 2005, **77**, 4626–4639.
41. D. Tsur, S. Tanner, E. Zandi, V. Bafna and P. A. Pevzner, *Nat. Biotechnol.*, 2005, **23**, 1562–1567.
42. A. S. Carvalho, H. Ribeiro, P. Voabil, D. Penque, O. N. Jensen, H. Molina and R. Matthiesen, *Mol. Cell. Proteomics*, 2014, **13**, 3294–3307.
43. R. Matthiesen, *Methods Mol. Biol.*, 2013, **1007**, 119–138.
44. R. Matthiesen, L. Azevedo, A. Amorim and A. S. Carvalho, *Proteomics*, 2011, **11**, 604–619.
45. J. K. Eng, A. L. McCormack and J. R. Yates, *J. Am. Soc. Mass Spectrom.*, 1994, **5**, 976–989.
46. D. L. Tabb, A. Saraf and J. R. Yates 3rd, *Anal. Chem.*, 2003, **75**, 6415–6421.
47. D. B. Kristensen, J. C. Brond, P. A. Nielsen, J. R. Andersen, O. T. Sorensen, V. Jorgensen, K. Budin, J. Matthiesen, P. Veno, H. M. Jespersen, C. H. Ahrens, S. Schandorff, P. T. Ruhoff, J. R. Wisniewski, K. L. Bennett and A. V. Podtelejnikov, *Mol. Cell. Proteomics*, 2004, **3**, 1023–1038.
48. J. V. Olsen, B. Blagoev, F. Gnad, B. Macek, C. Kumar, P. Mortensen and M. Mann, *Cell*, 2006, **127**, 635–648.
49. J. E. Elias, W. Haas, B. K. Faherty and S. P. Gygi, *Nat. Methods*, 2005, **2**, 667–675.
50. M. M. Savitski, S. Lemeer, M. Boesche, M. Lang, T. Mathieson, M. Bantscheff and B. Kuster, *Mol. Cell. Proteomics*, 2011, **10**, M110 003830.
51. P. R. Baker, J. C. Trinidad and R. J. Chalkley, *Mol. Cell. Proteomics*, 2011, **10**, M111 008078.
52. R. J. Chalkley and K. R. Clauser, *Mol. Cell. Proteomics*, 2012, **11**, 3–14.
53. J. Cox, N. Neuhauser, A. Michalski, R. A. Scheltema, J. V. Olsen and M. Mann, *J. Proteome Res.*, 2011, **10**, 1794–1805.
54. S. A. Beausoleil, J. Villen, S. A. Gerber, J. Rush and S. P. Gygi, *Nat. Biotechnol.*, 2006, **24**, 1285–1292.
55. C. P. Albuquerque, M. B. Smolka, S. H. Payne, V. Bafna, J. Eng and H. Zhou, *Mol. Cell. Proteomics*, 2008, **7**, 1389–1396.

56. C. M. Bailey, S. M. Sweet, D. L. Cunningham, M. Zeller, J. K. Heath and H. J. Cooper, *J. Proteome Res.*, 2009, **8**, 1965–1971.
57. D. H. Phanstiel, J. Brumbaugh, C. D. Wenger, S. Tian, M. D. Probasco, D. J. Bailey, D. L. Swaney, M. A. Tervo, J. M. Bolin, V. Ruotti, R. Stewart, J. A. Thomson and J. J. Coon, *Nat. Methods*, 2011, **8**, 821–827.
58. T. Taus, T. Kocher, P. Pichler, C. Paschke, A. Schmidt, C. Henrich and K. Mechtler, *J. Proteome Res.*, 2011, **10**, 5354–5362.
59. Z. Li, Y. Wang, Q. Yao, N. B. Justice, T. H. Ahn, D. Xu, R. L. Hettich, J. F. Banfield and C. Pan, *Nat. Commun.*, 2014, **5**, 4405.
60. L. Montecchi-Palazzi, R. Beavis, P. A. Binz, R. J. Chalkley, J. Cottrell, D. Creasy, J. Shofstahl, S. L. Seymour and J. S. Garavelli, *Nat. Biotechnol.*, 2008, **26**, 864–866.
61. J. Cox and M. Mann, *Nat. Biotechnol.*, 2008, **26**, 1367–1372.
62. D. Hyatt and C. Pan, *Bioinformatics*, 2012, **28**, 1895–1901.
63. R. Craig and R. C. Beavis, *Rapid Commun. Mass Spectrom.*, 2003, **17**, 2310–2316.
64. L. N. Mueller, M. Y. Brusniak, D. R. Mani and R. Aebersold, *J. Proteome Res.*, 2008, **7**, 51–61.
65. A. S. Carvalho, H. Molina and R. Matthiesen, *Sci. Rep.*, 2016, **6**, 18826.
66. J. R. Wisniewski, A. Zougman, S. Kruger and M. Mann, *Mol. Cell. Proteomics*, 2007, **6**, 72–87.

Section II

Protein Quantitation

CHAPTER 7

Algorithms for MS1-Based Quantitation

HANQING LIAO[a], ALEXANDER PHILLIPS[a], ANDRIS JANKEVICS[a] AND ANDREW W. DOWSEY*[a]

[a]University of Liverpool, Department of Electrical Engineering and Electronics, Brownlow Hill, Liverpool, L69 3GJ, UK
*E-mail: andrew.dowsey@liverpool.ac.uk

7.1 Introduction

In LC-MS/MS-based discovery studies, many thousands of peptides can be determined from complex samples, most appearing as multiple signal features due to differing numbers of protonation. Raw analytical data are significantly complex, on the order of gigabytes, with many thousands of MS1 spectra collected across a single liquid chromatogram. Each mass spectrum contains 100–1000s of unique features, with each feature being separated across multiple mass spectra. The first aim of MS1 quantitation is to directly estimate the intensity of each feature by integrating the ion count under its chromatographic peak shape or extracted ion chromatogram (XIC). Assuming detector linearity, this intensity is proportional to the ion count, which is proportional to peptide abundance. Since ionisation efficiency varies between features, only relative abundance changes across treatment groups can be established, by inferring ratios of how abundance differs between those samples. Nevertheless, some level of absolute quantification is possible if we possess

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

information about peptide 'detectability', either through synthetic standards or the machine learning of empirical evidence.[1] As well as ion efficiency, it is important to mention ion suppression effects. Less volatile compounds can change the efficiency of droplet formation or droplet evaporation during ionisation, thus affecting the distribution of ions reaching the detector and hence the comparison of peptide abundances across samples.

The established approach to MS1-quantitation is reductionist, converting the raw spectrum data into a mass peak representation at an early stage, before assigning each peak to a feature, and in turn each feature to a peptide. Instrumental drift in relation to retention time (RT) is observed across samples analysed in single or multiple analytical batches. Hence to compare relative quantification across samples at the feature, peptide or protein level, RT alignment and matching of corresponding features across runs is necessary. An in-depth assessment of MS reveals that systematic and random variation inherent in MS data comes from a number of known sources and in a number of cases these have been characterised.[2]

(1) Poisson distributed uncertainty due to the limited number of ions recorded at the detector, exhibited as multinomial variation between isotope peaks.
(2) The range of isotopic patterns exhibited by the range of compounds and their adducts (*e.g.* post-translational modifications).
(3) With quadrupole and time-of-flight (ToF) instruments, non-linear effects at high intensities appear due to detector saturation. This may be mitigated at the loss of some sensitivity by automatic gain control if implemented within the instrument.
(4) Artefact peaks observed with Fourier Transform (FT) instruments.
(5) Changes in electrical and thermal operating properties (*e.g.* time-of-flight expansion and constriction).
(6) A chemical baseline, periodic over ~1 Dalton, composed of contaminants, fragmented peptides and numerous coalesced peptides at very low abundances.[3]
(7) Biological variation, often approximated by a log-normal distribution in proteomics studies.

If your experiment involves analysing cell populations cultivated in cell culture, more sensitive differential analysis can result by utilising an isotopic labelling workflow such as stable isotope labelling by amino acids in cell culture (SILAC).[4] Here, in one of the populations natural 'light' amino acids are replaced by 'heavy' SILAC versions by incorporation through the growth medium *e.g.* arginines can be replaced with those labelled by six carbon-13 atoms. Light and heavy samples are then mixed and run together through LC-MS, with an extension of the approach allowing three protein populations to be compared simultaneously. In the resulting MS1 spectra, the peptide features of each sample can be detected and quantified separately, and linked together by the mass difference caused by the heavy labelling.

In brief, there are five components to an MS1-quantitation pipeline:

- *Feature detection and quantitation* – extraction and quantification of peptide features, with the monoisotopic peak and charge state determined through deisotoping and decharging respectively.
- *Peptide and protein-level identification* – each peak-picked MS2 spectrum, together with the detected *m*/*z* and charge state of its associated MS1 precursor feature, is submitted for peptide identification using methods described in earlier chapters. Protein-level grouping and false discovery rate corrections at peptide and protein level are then performed on the peptide identifications as a group, as described in Chapters 4 and 5.
- *Chromatogram alignment* – grouping corresponding features across experimental runs and propagating peptide identifications when available. Chromatographic deformation can be significantly variable and deteriorates with column age, with complete signal dropouts possible.
- *Abundance normalisation* – primarily, the correction of sample-loading differences, and more recently, the correction of ionisation fluctuations during runs. These both lead to greatly improved differential quantitation sensitivity.
- *Protein-level differential analysis* – methods to determine protein quantitation from constituent peptide measurements, and methods to statistically determine significant regulation across corresponding protein quantifications between multiple treatment groups. Recent methodology performs both these functions in one step, and can handle specialised or complex experimental designs (*e.g.* repeated measures or mixing technical and biological replicates).

Whether you employ a labelling approach or a label-free workflow, a number of biological samples should be run per condition so that statistical inferences can be made regarding population-level differences. Hence in each case, multiple LC-MS runs are usually acquired and so chromatogram alignment and intensity normalisation is invariably a requirement.

In the following three sections we review the state-of-the-art for the four components stated. Note that not all software pipelines follow the previous order shown. Moreover, since the landscape is complex and not conveniently modular, in each section we explain the methodology of the established MaxQuant pipeline[5] and compare it to competing approaches and cutting edge research.

7.2 Feature Detection and Quantitation

For a broad and detailed review of the commonly used feature detection techniques we refer the reader to ref. 2, 3, 6 and 7. A representative feature detection pipeline is illustrated in Figure 7.1a and can be divided into two

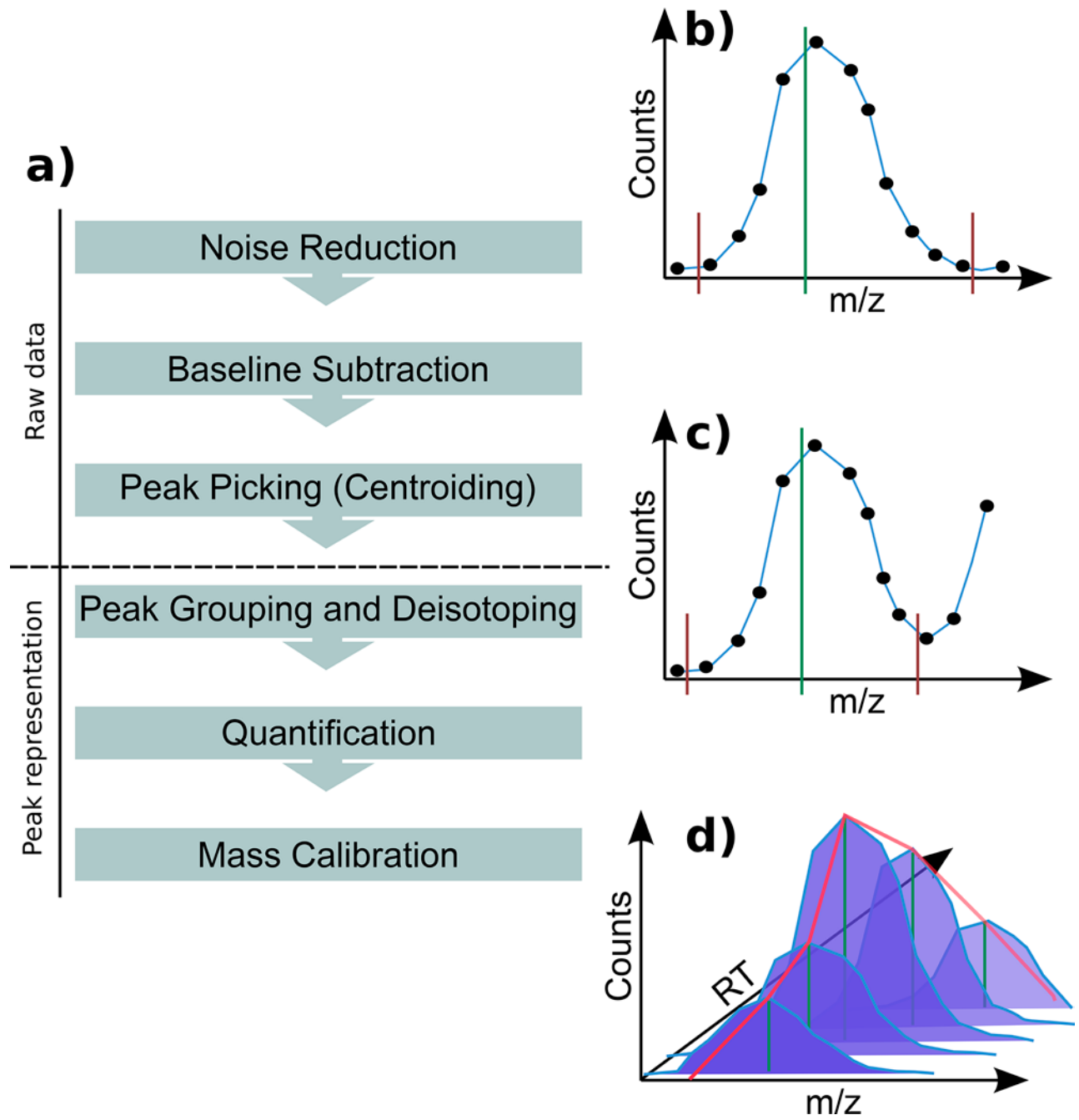

Figure 7.1 (a) The typical informatics pipeline for MS1 feature quantitation. Noise reduction and baseline subtraction are first performed, before a centroiding step performs the conversion from raw data to a symbolic peak-based representation. Isotopic peaks are then grouped to form peptide features. At this stage, the masses of the detected features may inform a mass re-calibration step. (b) Illustration of how MaxQuant performs centroiding by establishing the local maximum and boundaries of a peak. (c) If there are two overlapping signals, the boundary is set to their mutual local minimum. (d) Centroids are then linked across neighbouring spectra to establish the chromatographic peak, with quantitation performed by integrating under this shape. This figure is adapted from ref. 5 with permission from Macmillan Publishers Ltd. Copyright 2008.

steps; the first acting to extract peaks from the raw data, and the second to group isotopic peaks from the same peptide features together so that monoisotopic peaks and charge states can be established. In the following discussion, we will present conventional approaches and also some new methods that detect features from the raw data directly.

7.2.1 Conventional Feature Detection

In the first step, each raw spectrum containing profile mode continuum measurements is converted into a list of (m/z, intensity) pairs representing the salient peak centres. This step is called centroiding or peak picking. In MaxQuant, centroids are detected by locating the local maxima of the intensities

within each MS spectrum and then extending the *m*/*z* values below and above this maximum until a minimum is reached (Figure 7.1b, where the middle vertical line represents the local maximum, and the other vertical lines represent minima). If there are two overlapping signals, the peak boundaries are set at their mutual local minimum (Figure 7.1c). Subsequently, a Gaussian peak shape is fitted to the three data points nearest the maximum intensity, in order to interpolate the centroid's *m*/*z*. The peak is then quantified as the sum of the intensities of all the data points within.

MaxQuant is predominantly designed for Thermo Orbitrap® data, for which they assume no extra pre-processing steps are required prior to maxima–minima detection. In general, spectrum de-noising is first performed, for which the most common techniques are signal to noise ratio (SNR) filtering and smoothing approaches such as Gaussian filtering, LOWESS and Savitzky-Golay. In general, SNR algorithms either use a predefined threshold or noise model to estimate signal noise levels.[8] In this approach, signals with a local maximum above the estimated SNR and within an expected peak width are considered as peaks. Baseline subtraction can also be performed at this point *e.g.* median window or top hat filtering.

In step two, centroids are assembled into two-dimensional features by constructing a chromatographic peak profile XIC for each across neighbouring spectra, and connecting isotope peaks. MaxQuant decomposes this task into two, first constructing each XIC by linking centroids from neighbouring spectra that lie within a 7 ppm window, as shown in Figure 7.1d. Centroids that cannot be matched to any neighbouring spectra are discarded as noise. A windowed mean filter is then applied in the RT direction to robustly detect any local minima within each XIC, for splitting co-eluting peaks with similar *m*/*z* (*e.g.* from isomers). Finally, the mass of each peak is re-estimated as the mean of the per-spectrum centroids weighted by their intensity.

To connect isotope peaks, and from that perform deisotoping, it is essential to understand how isotopes affect the MS signal. Since MS measures *m*/*z*, on high resolution instruments each feature is seen as a set of peaks separated by an *m*/*z* interval of approximately 1 Dalton divided by the feature's number of protonations *z*, *i.e.* the charge state. The peak composed of only the isotopes of greatest natural abundance is termed the monoisotopic peak. With proteins, this is usually the peak of lowest mass. The second isotope peak (of monoisotopic mass plus one) is actually a compound peak comprised of all the possibilities where one atom in the peptide has an extra neutron *e.g.* for a typical peptide comprised of carbon, hydrogen, nitrogen, oxygen and sulfur, the second peak is the sum intensity of peptides with a single ^{13}C, ^{2}H, ^{15}N, ^{17}O or ^{33}S atom. The number of these combinations increases geometrically as the isotope peaks increase in mass.[9] Using ultra-high resolution FT-MS, the mass defect allows some of this isotopic fine detail to be resolved into a further number of distinct peaks in their own right.[10] The relative signal intensity for each isotopic variant (the 'isotope distribution') follows a multinomial distribution with parameters given by the relative (usually naturally occurring) abundances of the elemental isotopes. Apart from the fact

that peaks may overlap or be corrupted by noise, we also in general do not know the underlying molecular formula in advance, and as aforementioned, multiple isotopes coincide in the spectrum. It is therefore not possible to perform a simple hypothesis test to assess if a sequence of peaks represents a peptide. It is also infeasible to find the goodness of fit between the peak set and the expected isotope distribution of every possible peptide that lies within the calibration error of the instrument. Rather, the generally adopted method is to fit a representative isotope distribution such as the averagine[11] to the peak set, allowing for some error. In MaxQuant, extracted 2D peaks are clustered into features by constructing an undirected graph with the peaks as the nodes and connecting those together that could represent neighbouring isotopes. The graph is then decomposed into connected sub-graphs each representing a single candidate peptide, which is then filtered by correlating the isotope distribution to the averagine.

7.2.2 Recent Approaches Based on Sparsity and Mixture Modelling

Sparse approaches rely on the ability to decompose a signal parsimoniously into a linear combination of elementary signals, or 'atoms', from a 'dictionary' of such signals. Since noise cannot be decomposed parsimoniously, a very successful strategy is to design a data transformation that represents MS signal sparsely. After transformation, the interesting MS signal is concentrated within a few data points, whilst noise is spread across all data points. Hence, thresholding the transformed data will remove most of the noise but alter the signal only very slightly. As in many fields, a popular transformation is the wavelet transform, which converts the data into components related to both position and size ('scale' *i.e.* coarse to fine detail). In MS, so called 'wavelet denoising' techniques are typically used in combination with SNR peak detection,[12] with adaptive thresholding used to compensate for dependencies between neighbouring scales.[13,14]

In contrast to these denoising approaches based on conventional centroiding, where peak shape is more or less ignored, other approaches have attained greater detection sensitivity and specificity by defining a specialised wavelet that represents the shape of a peak or even whole peptide feature. Here, the continuous wavelet transform (CWT) is employed, to directly extract candidate peaks from the data while being robust to a low frequency baseline. A successful wavelet choice has been the Gaussian 2nd derivative (popularly called the Mexican Hat), whose response is proportional to the height of Gaussian-shaped peaks of a particular width.[15] An interesting approach is offered by the OpenMS Isotope Wavelet,[16] which was designed to respond directly to the full averagine peptide isotope distribution in m/z space, hence detecting features directly from the raw data.

In comparison to SNR techniques, CWT-based algorithms are reported to perform better on data sets with variable noise levels, yet still fail to detect distorted or overlapping peaks.[3] In the CWT, a linear convolution filter is

used to approximate the solution of an inverse problem. While optimal for detecting a peak among additive white noise, the optimality assumptions do not hold when multiple peak signals overlap, resulting in spurious peaks and other artefacts. These can be reduced if the analysis is restricted to only a single filter that transforms the known peak shape function into a narrower version of itself, or if a geometric mean of multiple linear filters is used to better approximate a non-linear filter.[17] Nevertheless, in most cases an involved post-processing step is required to remove false-positives. For example, it has been shown that detecting zero-crossings of the Gaussian 1st derivative wavelet transform, with peak height and width estimated with the zero-crossings of the Gaussian 2nd derivative, provides a more robust solution.[15] Methods to select the set of analysis scales from the data have been reported,[18] as have methods that utilise the resulting patterns observed across scales, which include zero-crossings, ridges and valleys.[19]

Mixture modelling is an alternative or complementary technique that can be used to improve feature detection specificity. Here, parametric peak models are arranged on a set of previously detected peaks and fitted to the data by optimising their coefficients with a non-linear iterative technique such as Expectation-Maximisation[20] or graph-based integer linear programming.[21] With this approach, the adducts and charge states for each peptide can be modelled together, thus adding constraints that allow poorly resolved or coincident features to be detected and quantified reliably by borrowing strength from their relations in other parts of the spectrum. Mixture modelling can also be performed *via* stochastic methods such as Markov Chain Monte Carlo (MCMC), which adds robustness against converging to sub-optimal solutions.[22] Notably, these techniques sample the range of possible outcomes and therefore also output uncertainty information. However, MCMC techniques are computationally expensive, so remain tractable only if the number of unknowns is kept small.

A class of methods based on sparse regression are emerging, which consist of elements from both wavelet and mixture modelling approaches. Like CWT approaches, peak or feature templates are compared to the data at regular small intervals in the spectrum. However, like mixture modelling, iterative nonlinear estimation is employed, though in this case it is used to fit a weighted sum of these templates to the data. Sparsity is the key goal, so that only the subset of templates representing true peaks or features are assigned nonzero weights, *i.e.* are 'active'. An established statistical approach for computing sparse solutions is the least absolute shrinkage and selection operator (LASSO),[23] which was first applied to MS for assigning isotope distributions across multiple charge states to a set of detected peaks.[24] A popular means to estimate the LASSO trace (the set of solutions from when all templates are active, to when only one template is active) is with least angle regression stagewise (LARS). LARS iteratively adds the template with maximal correlation to the residual to the active set. The weighting coefficient for each template in the active set is updated at each iteration to fit the data, whilst maintaining equiangular correlation with the residual. The NITPICK

method for MS1 feature detection and quantitation extended LARS with non-negativity constraints on the weights and employed Bayesian Information Criterion (BIC) model selection to terminate when the optimum number of active templates is reached *i.e.* before over-fitting.[25] NITPICK works on the raw data by pre-convolving the isotope distribution templates with the known instrument peak shape. More recently, an analogous approach based on mixture modelling has also been proposed.[26]

7.3 Chromatogram Alignment

In modern MS, *m*/*z* measurements are precise and reproducible, but chromatogram RT is known to be variable across different runs due to many factors including variations in temperature, flow rate, and properties of the column such as the gradient of the mobile phase. Since the same peptide feature will be found at different RTs in each run, this hinders the establishment of feature correspondences across runs, resulting in problems such as missing or incorrect linkage in a subset of runs. This has significant impact both on quantification and on the propagation of peptide identifications to unidentified features in other runs. The key goal of chromatogram alignment is to adjust all the runs to a common RT coordinate system such that corresponding features will have highly similar RT. Neighbourhood matching schemes are then employed to link the same features across runs to create ‘consensus’ features. Each resulting consensus feature RT may then be adjusted to some robust average of the original feature RTs in order to provide a calibrated estimate that can be compared with the predicted RTs of peptides stored in a database. This can provide an extra level of discriminatory power in the identification phase. These predicted RTs are learnt from prior experiments by regression or kernel learning from the set of reliable identifications and sequence information.[27,28]

Broadly, there are two main methodological categories of alignment algorithm. In feature-based approaches, corresponding features are brought into alignment by comparing the spatial pattern exhibited by each run. In the raw profile alignment approach, the raw data for each run is summarised into either a total ion chromatogram or base peak chromatogram (TIC or BPC), or image representation, and directly warped to align corresponding signals. Some techniques, such as the commercial Progenesis QI® package (Waters Inc.) perform chromatogram alignment in this fashion before a single consensus feature detection on the stacked set of aligned runs. Whether a feature-based or raw profile alignment is employed, there are further commonalities. A model needs to be chosen to translate the RT alignment problem into a mathematical criterion, often including a mechanism to warp the RT dimension, *e.g.* a global or piecewise linear shift, cubic-spline or thin-plate spline shift, or non-linear smoothing. After that, a practical optimisation strategy is employed to derive the warp for an optimal alignment.

Two points can be made comparing feature-based with raw profile alignment. Firstly, there is a loss of information by only utilising features in the

alignment stage, rather than the full MS signal, and moreover, feature-based strategies need to be robust to erroneous features. Nevertheless, this may be balanced out by the higher quality information contained within true features, as specialised biochemical knowledge is utilised during feature extraction. Feature-based approaches also have reduced computational complexity, enabling them to scale to consider large numbers of runs in an unbiased way. The following discussion is far from an exhaustive list of methods, rather, descriptions are given to the evolution of each main branch of algorithm. For those readers who are interested in exploring more into this topic, two review papers are recommended.[2,29]

7.3.1 Feature-Based Pattern Matching

Feature-based alignment methods utilise retention times, *m*/*z* ratios, charge states, *etc.* of either peaks or whole features extracted by the feature detection stage. A typical approach is to attempt to establish candidate correspondences between features across runs based on mass and/or identification information and then use curve fitting on RT differences between these to estimate the deformation. In more involved techniques, after correcting for this deformation by warping feature RTs, the correspondence estimation is renewed and the process iterates until convergence.

Reference mapping algorithms, as the name suggests, require an LC-MS run to be selected as reference. Corresponding features from the rest of the runs are mapped to this reference. For example, pose clustering has been employed for robust pair-wise alignment.[30] Through an affine transformation, each pair of features in a run is mapped to every pair of features in the reference within a realistic range. An overall affine transformation for each run is then computed through a voting scheme.

With group-wise mappings, the reference may be updated during processing, up to a full pair-wise strategy that maps each run to every other. In this way reliance on a single reference is avoided, so features not present in the reference run can still be matched. In MaxQuant's label-free quantification component MaxLFQ,[31] runs are organised into a hierarchical tree structure. The alignment begins by establishing pair-wise correspondences between the most similar runs, then moves on to align less similar runs with the runs already aligned, with the path of grouping runs forming a hierarchical tree. For each pair-wise alignment, a non-linear warp is computed by Gaussian kernel smoothing on the scatterplot containing the difference in RTs between features with similar mass in the two runs.

7.3.2 Raw Profile Alignment

Dynamic time warping (DTW)[32] and correlation optimised warping (COW)[33] are the earliest raw profile alignment algorithms applied to LC-MS data. A dissimilarity criterion is defined that quantitatively assesses how close two runs match at particular points in their RT profile, with zero meaning a

perfect match. Due to computational complexity, the first approaches compared only TICs or BPCs. However, DTW and COW based solely on TICs or BPCs often fail to align regions with significant differential expression, and so a complementary component detection algorithm (CODA) has been developed that uses only XICs reproducible across runs.[34] Other work utilises spectral details or detected features.[35] In the case of DTW, a similarity matrix is established with rows representing the spectra of one run and columns representing the spectra of the other run, while the cells contain the respective pair-wise similarities of those spectra. As illustrated in Figure 7.2, the optimal warp is defined as the 'shortest' path from the first row–column to the last

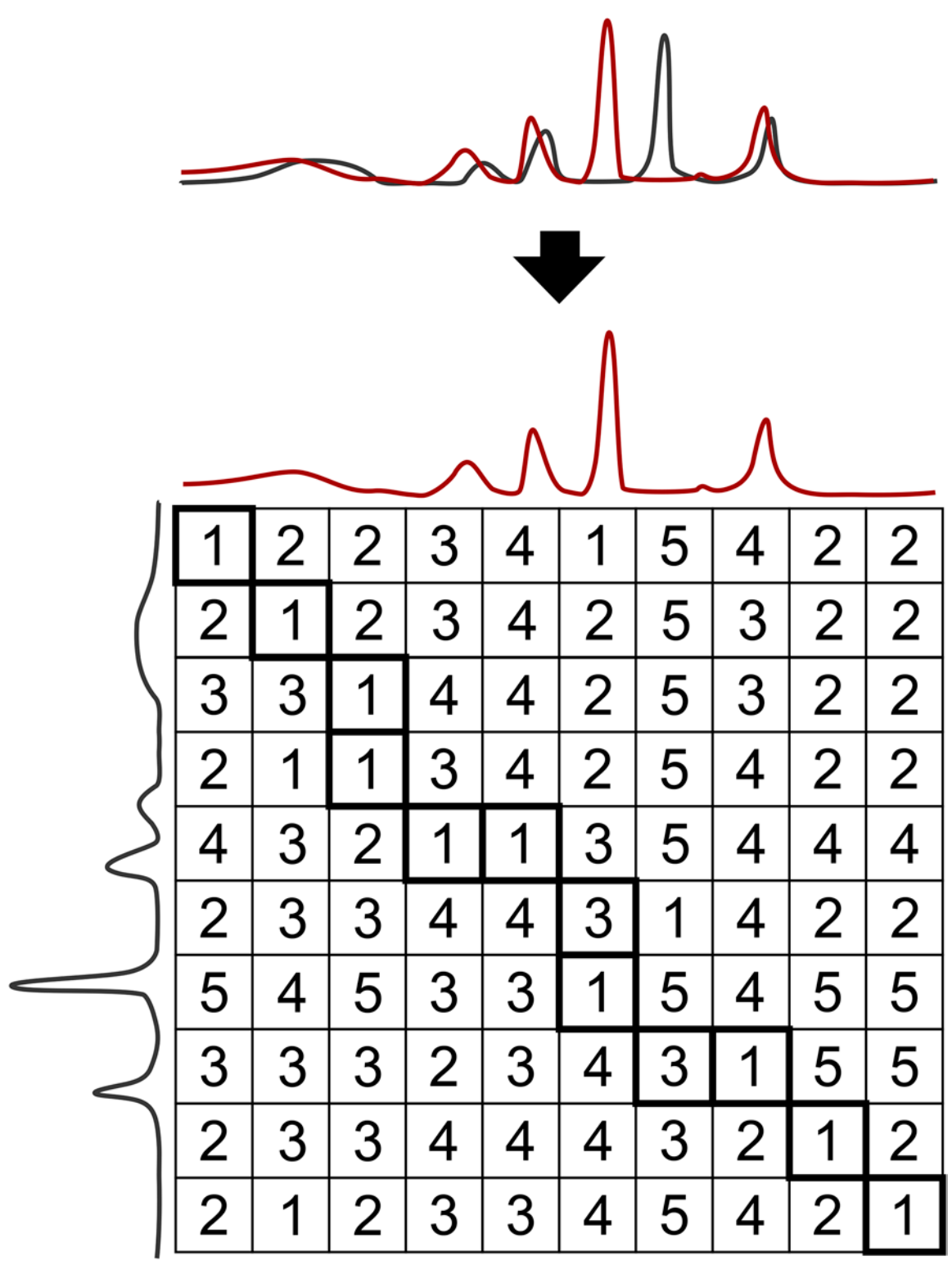

Figure 7.2 Schematic of the dynamic time warping approach for aligning two chromatograms. A dissimilarity matrix is generated, with rows representing the spectra of one chromatogram, and columns representing the spectra of the other chromatogram (For clarity, each chromatogram here has only 10 spectra.). Each cell in the matrix is computed as the dissimilarity between the two corresponding spectra. The dissimilarity measure could just be the difference between the respective TICs or BPCs at that point, or some measure of the peak pattern dissimilarity between them. Dynamic programming is then employed to efficiently find the shortest path from the first to last spectra (highlighted in bold), which represents the rough discretised warp that will realise the best alignment.

row–column *i.e.* the path with minimum dissimilarity, computed by dynamic programming. DTW has been progressively developed and is widely used for RT alignment. Since DTW solutions do not produce smooth warps, interpolation can be used to derive a bijective warp that can be applied to either of the two chromatograms to get a smooth RT warp function.[35] COW utilises a correlation criterion with a similar dynamic programming solution, but breaks down the chromatograms into sections. One chromatogram is used as reference and the other is warped by section-wise linear transformations.

Stochastic models have also been applied to RT profile alignment. The continuous profile model (CPM) enables group-wise alignment of multiple TICs through a reformulation of DTW using a hidden Markov model (HMM).[36] It models discrete shifts in alignment and changes in intensity between observed and consensus TICs as Markov processes, the consensus TIC profile as hidden states, and utilises the expectation-maximisation (EM) algorithm to probabilistically determine the most likely state changes and hence alignments. In a later study, they extended the model to use spectral information discretised into four mass bins, reporting improved alignment performance but at a significant computational cost.[37] More recently, a related strategy was presented using a Bayesian MCMC approach. In this, the authors were able to explicitly define alignment and intensity changes as piece-wise linear (with the number of pieces inferred from the data), the consensus TIC as a piece-wise B-spline curve, and provide uncertainty estimates for the alignments. A follow-up paper enabled retention time standards to be incorporated as prior information, and provided a way to determine and utilise multiple reliable XICs.[38]

Finally, we finish this section by mentioning the image registration approach, which was originally developed for the alignment of medical images, but has come to LC-MS *via* its use in 2D gel electrophoresis *e.g.* Progenesis QI® (Waters Inc.). As in other raw profile approaches, image registration utilises a dissimilarity criterion, but in image registration it is defined as the similarity between the runs given a specific alignment. Unlike DTW, COW and CPM, a continuous and smoothly realistic warping transformation can be used for the alignment (*e.g.* B-spline or thin plate spline curve), with its parameters optimised until similarity is maximised. This scheme is more computationally efficient than DTW, but in its basic form only features that have some overlap will be brought into correspondence. To mitigate this limitation, multi-resolution schemes are employed, where approximate alignment is first performed on heavily blurred images containing only gross spatial details, and then iteratively refined to account for finer and finer signal structure. Complementary feature-based information can also be incorporated into the dissimilarity criterion, and the technique scales efficiently to truly group-wise alignment strategies. In one recent example, a hybrid feature and raw profile alignment approach using robust M-estimation and a non-Euclidean similarity metric in a multi-resolution framework was demonstrated.[39] Robustness to reference image selection was achieved by registration to an evolving consensus image.

7.4 Abundance Normalisation

After features have been detected and quantified on each run, and corresponding features matched between runs, the remainder of the pipeline is concerned with their relative quantification across runs. However, the sample volume injected into the LC-MS instrument is not stable across runs, and electrospray performance and transmission efficiency fluctuate during each run. With SILAC, the mixing proportions of light and heavy samples may also not be perfect. These phenomena must be retrospectively corrected by abundance normalisation.

Firstly, an arbitrary sample is selected as reference, and then a different per-sample scaling factor is applied to the feature quantifications in the other samples. This can be achieved by spiking an internal standard of known quantity into each LC-MS sample. Upon identifying these internal standard features, they can be used to derive the normalising factors that equalise the quantification of these features in each sample with respect to the reference sample, and scaling all the other features accordingly. The disadvantage is that any systematic bias in the spike-in feature quantifications is propagated to the rest of the experiment. To mitigate this, the normalisation can be based on features assumed to be unchanged in the experiment, either explicitly through “housekeeping” proteins, or implicitly by assuming that the majority of proteins remain unregulated between samples. In the latter case, the simplest technique in widespread use is to calculate the ratio of intensities between the reference sample and the other samples, and from these compute the per-sample normalisation factors as the median ratio across all features.[40] For example, in MaxQuant the peptide ratios between light–heavy SILAC pairs are normalised so that the median peptide ratio is 1 : 1 in each run.

We must note that special treatment is required when samples have been pre-fractionated before acquisition in multiple LC-MS runs. This case is trivial in SILAC as the intensity ratios between light and heavy samples are uncorrupted as they are fractionated together. However, in label-free acquisition, establishing which peptides do not change relies on prior knowledge of the per-fraction normalisation factors, and *vice-versa*. MaxQuant's MaxLFQ modules addresses this by a ‘delayed normalisation’ approach that analyses quantifications across all fractions simultaneously. Here, an optimisation technique is employed to determine the normalisation factors that minimise the pair-wise intensity ratios between samples for all peptides across the experiment.[31]

When effects such as ion suppression, detector saturation or instrumental drift occur, the LC-MS signal intensity is no longer in a linear relationship with peptide abundance. In order to address these issues, non-linear normalisation techniques developed for microarray analysis have been applied, such as quantile normalisation.[41] More complex statistical models have also been investigated. Karpievich *et al.* adapted the Surrogate Variable Analysis technique of Leek and Storey, performing a Singular Value Decomposition

on the residuals of the fitted differential analysis model to determine if any significant systematic biases exist (*e.g.* batch effects) that have not already been specified in the differential analysis model.[42] Ranjbar *et al.* propose a Bayesian MCMC approach that determines and corrects for instrumental drift effects that cause smoothly changing systematic fluctuations in intensity related to acquisition time order.[43] Crucially, it was recently reported that these fluctuations are also highly noticeable during each run, contributing to significant differences in the intensity ratios between peptides of the same protein that elute at differing points in time.[44] The authors propose to apply median normalisation within an RT sliding window to better account for these time-varying effects in LC-MS signal intensity.

7.5 Protein-Level Differential Quantification

At this final stage in the pipeline we have a list of identified consensus peptide features, each with a separate (possibly missing) quantification per-sample. Through protein grouping (Chapter 5), each feature will also either be annotated as a subsequence of one protein, or as a 'shared' peptide that is a subsequence of more than one protein (Figure 7.3). We now wish to infer the relative abundance ratios of the parent proteins between those same samples, and either from these or directly, the higher-level experimental effects *e.g.* differences between treatment groups. While there are simple methods for deriving protein-level ratios between pairs of samples, more advanced statistical methods intrinsically incorporate protein-level quantitation as part of statistical differential expression analysis models that use the feature quantifications across all samples simultaneously. In this chapter, we will cover these methods predominantly to illustrate their merits for protein-level quantitation. Due to variations in the stochasticity of peptide cleavage, in general a protein-level quantitation is more accurate when many peptide quantifications support it.[45] Statistical methods hold a key advantage here, in that they can additionally utilise information on peptide quantification variability across samples.

MaxQuant calculates protein ratios as the median of all ratios of peptides belonging to that protein.[5] This reduces the effect of any outlying peptide ratios that could be corrupted by interferences or digestion issues, but fails

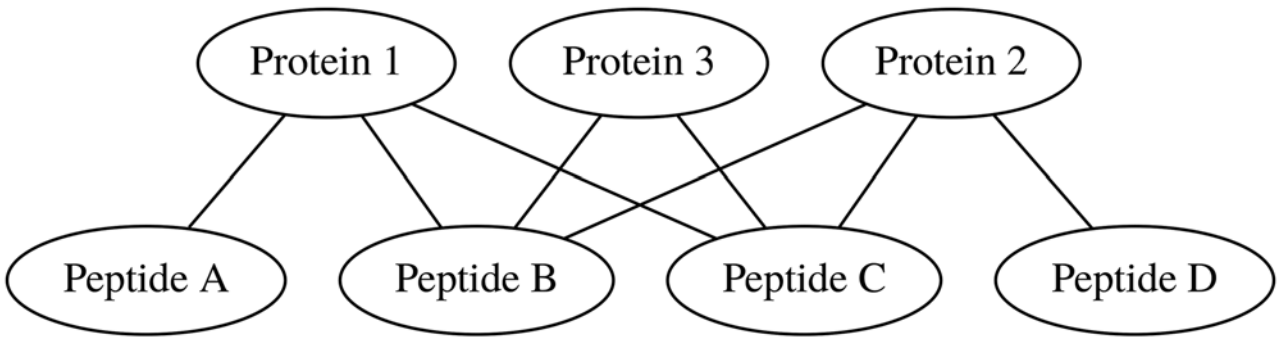

Figure 7.3 Peptides B and C, which are sub-sequences of more than one protein, are referred to as shared or degenerate peptides. Shared peptides are often discarded as their quantification pattern represents a mixture of the quantification patterns of their parent proteins.

to use all the available information. Under the assertion that ion count measurements in MS follow Poissonian statistics, Carrillo *et al.* investigated a number of similar schemes including averaging the peptide ratios, computing the ratio after summing the peptide quantifications in each sample, and using linear regression to compute the slope of the line fitting one sample's peptide quantifications to the other.[46] They note the effectiveness of using the ratio after summing peptide quantifications, and a modified 'total least squares' regression that minimises the orthogonal distance between the peptide ratios and line of best fit, which accounts for error in both samples' peptide quantifications. These methods are effective because errors in the relative fold–change decrease as intensity increases (as would be expected in Poissionian statistics), hence summing peptide quantifications before ratio calculation leads to proportional weighting of the more intense features. The regression approach adds a form of outlier rejection, in that intensity values are down-weighted according to their distance from the line of best fit, and therefore errors in the fold-change estimate are further reduced. The authors noted that the sum of peptide quantifications method performed marginally better, however, with the added benefit of being significantly less computationally expensive than the total least squares method.

For these simpler methods, an additional stage is needed to determine whether relative changes in protein abundance across treatment groups can be deemed significant.[47] Since biological variation is regularly assumed to be log-normally distributed in proteomics studies, normalised quantifications or quantification ratios are often log-transformed for statistical analysis. However, some studies have argued for the use of alternative variance-stabilising transformations that account for an additional additive component approximating instrument and ion counting noise.[48] For case-control experimental design, typically Student's *t*-test is then used to determine the statistical significance of the difference in abundances. Note that for clinical studies, Welch's *t*-test should be considered, since we cannot assume that cases and controls have the same population variance. In either case, we are then left with a *p*-value for each protein, the probability that the observed data or more extreme data would occur if the null hypothesis (no difference in abundance) were true. Since we are testing multiple proteins (multiple hypotheses), it is essential that the *p*-values are then adjusted to control the False Discovery Rate (FDR), which is the expected proportion of false positives in the set that we declare to be significant. Among other approaches, the Benjamini-Hochberg procedure is often employed.[49] For further information, discussion of appropriate experimental designs and an expanded treatment of statistical testing methodology, we refer the reader to ref. 47.

7.5.1 Statistical Methods

Fitting statistical models to all the feature-level data has the potential for more accurate quantification through joint inference of peptide reliability and differential quantitation.[50] This comes at a cost of being more

computationally intensive. These statistical models attempt to account for the inherent variability in the observed log-transformed intensities due to random experimental variation, with protein digestion being a major component. The most popular modelling framework underpinning these tools is the *mixed-effect* model, which generalises a large cross-section of statistical models including the *t*-test, linear regression and multi-way ANOVA. In this framework, predictors are termed 'fixed effects'. A fixed effect is one which we consider to be systematic *e.g.* the effect of a particular protein, peptide or condition on the fold change of a feature, or a batch effect between two batches. Unlike simpler models, mixed-effect models also support 'random effects'. Random effects represent stochastic fluctuations that occur within larger populations and are represented as log-normal distributions with unknown variance *e.g.* biological variation causing per-sample random deviations from the population mean, or batch deviations across many batches. When protein-level quantifications have already been derived, the resulting test for assessing differential expression (*e.g. t*-test) models biological variation as a log-normally distributed residual. In protein-level quantitation performed by a mixed-effects model, the log-normally distributed residual models technical variation at the feature level instead, so it is crucial to also fit a random effect to model protein-level biological variation.

Tools such as MSstats[51] fit the mixed-effects model on a per-protein basis, employing standard methods for fitting based on Restricted Maximum Likelihood (ReML). As illustrated in Figure 7.4, each log-transformed feature is modelled as a linear combination of peptide feature, condition and sample effects, with feature and condition assigned as fixed effects

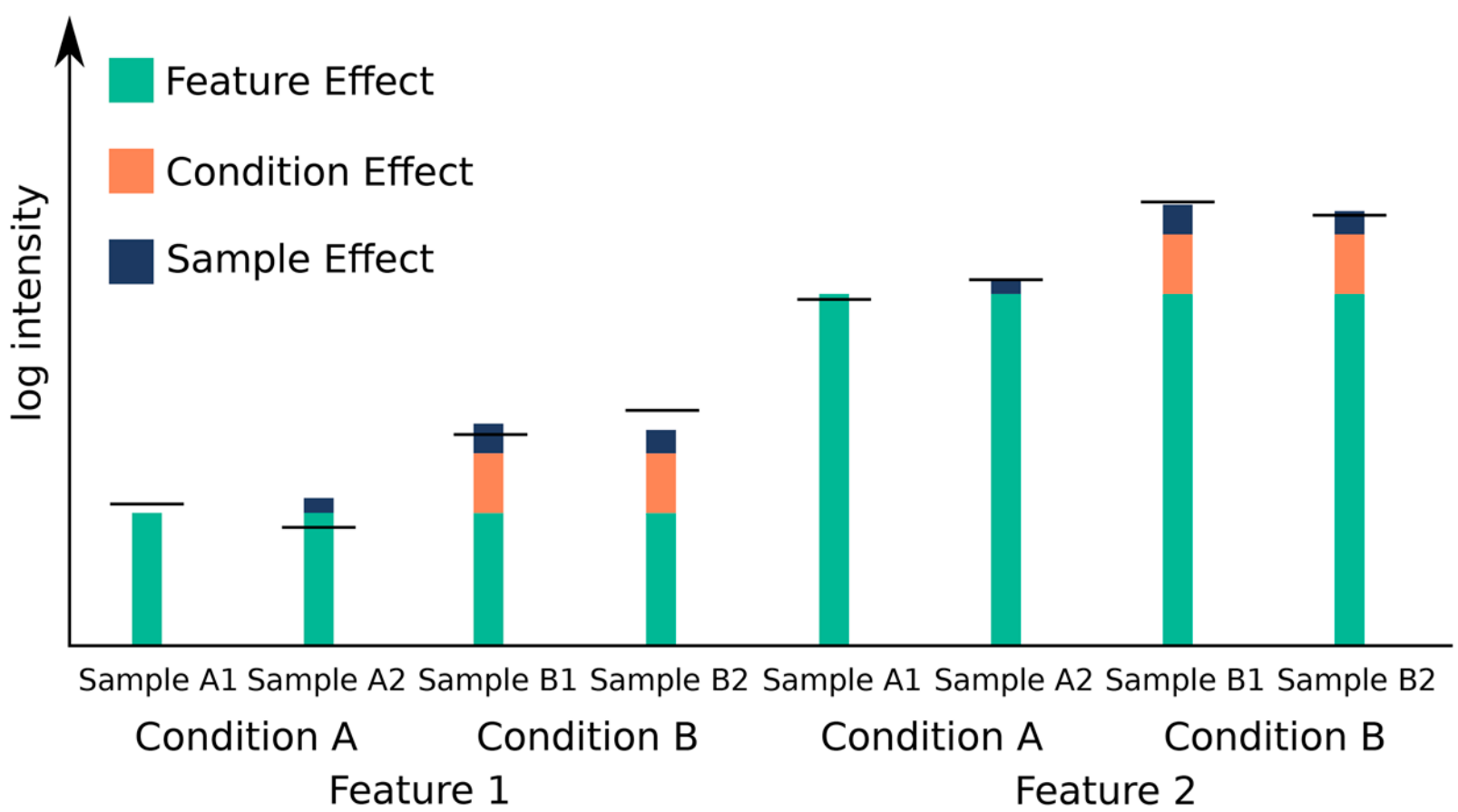

Figure 7.4 Protein-level quantitation based on the linear or mixed-effects model decomposes the log intensity of each feature quantification (horizontal bars) into per-feature, per-condition and per-sample contributions. Here, the residuals for feature 1 are greater than feature 2, hence for models that infer per-feature residual variances, peptide 2 will influence the protein-level quantifications to a greater extent.

and sample as a random effect (if the sample size is adequate). MSstats also includes an optional interaction effect between feature and condition which models feature-specific signal interferences that only appear in one condition.[52] A popular approach is to estimate a separate residual variance for each feature, which has the effect of weighting each feature's contribution to the protein-level quantitation by the reciprocal of its inferred residual variance.[53] However, the authors of MSstats advise that ReML will over-fit such a model, advocating that a non-linear relationship between a feature's abundance and its variance should be enforced to avoid overfitting.[52] This is achieved by ReML fitting of a single residual variance to all features, but with the variance weighted on a per-feature basis. Through a technique called iterative reweighted least squares, the weights are initially set to unity and are then iteratively refined by rounds of LOESS curve fitting to the model residuals against predicted feature abundance followed by ReML model refitting.

Goeminne *et al.* recently presented three improvements to increase the robustness of the mixed-effects model approach:[54] (i) 'Ridge regression' is adopted to reduce over-fitting by penalising the feature effect. This is achieved by assuming peptide features from the same protein have log-normally distributed fluctuations in fold change within each sample. Since estimating this variance can only be achieved with significant uncertainty when a protein is supported by only a few quantified peptides, statistical testing becomes more conservative in these cases. (ii) Rather than estimate per-feature residual variances, an M-estimation approach with Huber weights is used to down-weight individual outlier quantifications. (iii) Through Empirical Bayes, the residual variance estimates of proteins with few observations are made more reliable by borrowing strength from the variance estimates of other proteins in the experiment.

We conclude this section by discussing the missing data issue. The proteome informatics pipeline will result is many consensus features missing quantifications for one or more samples. There are two main mechanisms for this missingness: (i) low intensity features are much more likely to be missed due to insensitivity in the feature detection method *i.e.* these quantifications are 'censored'. (ii) Features can be missed at random, due to a combination of technical (*e.g.* ion-suppression effects) and informatics issues (*e.g.* failure to deconvolute co-eluting interferences). Ignoring all missing data will reduce the sensitivity of differential expression analysis, as protein quantifications will be over-estimated in conditions with greater numbers of censored values. Conversely, setting all missing data to zero will both over-estimate differential expression and bias quantitation where the missingness is completely at random. Karpievitch *et al.* have presented a mixed-effects model that can compensate for these missingness mechanisms, and optionally impute the missing data.[53] Given a study-wide heuristic estimate of the probability a missing quantification is at random, their model estimates feature-specific censoring thresholds and hence the distribution of intensity values each missing quantification could have represented.

7.5.2 Statistical Models Accounting for Shared Peptides

A peptide may be present in, and therefore represent more than one protein. These are referred to as shared or degenerate peptides. In most cases these are discarded as being uninformative. However, Figure 7.3 considers the case where proteins 1 and 2 are represented in results by peptides A, B and C, and B, C and D respectively. Ignoring shared peptides B and C we can infer the presence of both proteins 1 and 2 by the presence of peptides A and D. Then suppose that protein 3 is typified by peptides B and C only; ignoring shared peptides would mean that protein 3 is undetected. Shared peptides are often discarded since using the relative abundance of a constituent peptide as the relative abundance of the parent protein is only viable in the case where that peptide is unique to that protein.[45] However, in a typical protein database shared peptides can account for as much as 50% of the peptides in the database;[55] we are effectively discarding half of the information that we might use to identify and quantify those proteins. In the cases where a protein has no unique peptide, we would be totally unable to infer its presence or quantification. Models which account for shared peptides in some way include that proposed by Blein-Nicolas *et al.*[56] A non-linear model is proposed where the measured log transformed quantification of a peptide is equal to the sum of: the log-sum of the quantifications of the proteins it is a part of, a peptide random effect, random error due to biological variation, random error due to technical variation, and the residual error. Crucially, as this model is evaluated across all proteins at once, and uses all the available information to calculate the protein quantifications, the authors demonstrate that this improves the accuracy of the estimations of parameters compared to a model quantifying one protein at a time. This comes at significant computational expense, however, as Bayesian MCMC is utilised for inference.

7.6 Discussion

The final result of the presented MS1 quantitation pipeline is a list of proteins which we have identified as being differentially expressed according to our experimental design, along with a measure of our confidence in this assertion, the FDR, and a measure of how much we believe them to have changed *i.e.* their ratios or fold changes. These results could then be used to present a set of proteins for further analysis, whether by pathway analysis or as candidates for developing a biomarker prediction panel. Other biological applications may call for different downstream analysis, but the core quantitation algorithms are typically the same.

One of the key limitations of current strategies for MS1 quantitation is that they make deterministic decisions at each stage of the workflow. Since no algorithm has been presented that is 100% reliable and sensitive, erroneous decisions are inevitable, leading both to incorrect knowledge extraction (false positives) and missed knowledge extraction (false negatives). Most stages of the pipeline attempt to be robust to false positives in their input, but

again are not 100% successful. Hence in many cases errors accumulate as they flow through the pipeline. The use of rigorous stochastic sampling (*e.g.* MCMC) or more computationally efficient optimisation approaches (*e.g.* Approximate Bayesian Computation)[57] that capture uncertainty in the results and propagating this information downstream is an area of significant future promise. Nevertheless, this methodology will not directly influence the false negative rate. The key to greater sensitivity is to utilise prior information earlier in the pipeline before the raw data is discarded (*e.g.* the NITPICK sparse regression approaches that perform deisotoping on the raw data), or on the other side of the same coin, preserve the raw data throughout the pipeline. As a proof of concept, Liao *et al.* recently proposed a pipeline of image denoising, registration and functional mixed-effects modelling that performs differential analysis directly on the denoised and aligned raw data.[39] With this approach, they discovered a number of differentially expressed peptide features that were missed by Waters Progenesis® at the feature detection stage.

Acknowledgements

The support of the UK BBSRC [BB/M024954/1] and MRC [MR/L011093/2] are acknowledged.

References

1. P. Kelchtermans, W. Bittremieux, K. De Grave, S. Degroeve, J. Ramon, K. Laukens, D. Valkenborg, H. Barsnes and L. Martens, *Proteomics*, 2014, **14**, 353–366.
2. A. W. Dowsey, J. A. English, F. Lisacek, J. S. Morris, G.-Z. Yang and M. J. Dunn, *Proteomics*, 2010, **10**, 4226–4257.
3. C. Bauer, R. Cramer and J. Schuchhardt, *Data Mining in Proteomics*, 2011, vol. 696, pp. 341–352.
4. S.-E. Ong and M. Mann, *Nat. Protoc.*, 2007, **1**, 2650–2660.
5. J. Cox and M. Mann, *Nat. Biotechnol.*, 2008, **26**, 1367–1372.
6. R. Hussong and A. Hildebrandt, *Proteome Bioinformatics, Methods in Molecular Biology*, 2010, vol. 604, pp. 145–161.
7. J. Zhang, E. Gonzalez, T. Hestilow, W. Haskins and Y. Huang, *Curr. Genomics*, 2009, **10**, 388–401.
8. P. Du, G. Stolovitzky, P. Horvatovich, R. Bischoff, J. Lim and F. Suits, *Bioinformatics*, 2008, **24**, 1070–1077.
9. J. Meija, *Anal. Bioanal. Chem.*, 2006, **385**, 486–499.
10. M. L. Toumi and H. Desaire, *J. Proteome Res.*, 2010, **9**, 5492–5495.
11. M. W. Senko, S. C. Beu and F. W. McLaffertycor, *J. Am. Soc. Mass Spectrom.*, 1995, **6**, 229–233.
12. J. S. Morris, K. R. Coombes, J. Koomen, K. A. Baggerly and R. Kobayashi, *Bioinformatics*, 2005, **21**, 1764–1775.
13. D. Kwon, M. Vannucci, J. J. Song, J. Jeong and R. M. Pfeiffer, *Proteomics*, 2008, **8**, 3019–3029.

14. F. Mo, Q. Mo, Y. Chen, D. R. Goodlett, L. Hood, G. S. Omenn, S. Li and B. Lin, *BMC Bioinf.*, 2010, **11**, 219.
15. N. Nguyen, H. Huang, S. Oraintara and A. Vo, *Bioinformatics*, 2010, **26**, i659–i665.
16. R. Hussong, B. Gregorius, A. Tholey and A. Hildebrandt, *Bioinformatics*, 2009, **25**, 1937–1943.
17. D. I. Malyarenko, W. E. Cooke, E. R. Tracy, R. R. Drake, S. Shin, O. J. Semmes, M. Sasinowski and D. M. Manos, *Rapid Commun. Mass Spectrom.*, 2006, **20**, 1670–1678.
18. P. Wang, P. Yang, J. Arthur and J. Y. H. Yang, *Bioinformatics*, 2010, **26**, 2242–2249.
19. Z.-M. Zhang, X. Tong, Y. Peng, P. Ma, M.-J. Zhang, H.-M. Lu, X.-Q. Chen and Y.-Z. Liang, *Analyst*, 2015, **140**, 7955–7964.
20. M. Dijkstra and R. C. Jansen, *Proteomics*, 2009, **9**, 3869–3876.
21. C. Bielow, S. Ruzek, C. G. Huber and K. Reinert, *J. Proteome Res.*, 2010, **9**, 2688–2695.
22. Y. Sun, J. Zhang, U. Braga-Neto and E. R. Dougherty, *BMC Bioinf.*, 2010, **11**, 490.
23. T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*, Springer New York, New York, NY, 2009.
24. P. Du and R. H. Angeletti, *Anal. Chem.*, 2006, **78**, 3385–3392.
25. B. Y. Renard, M. Kirchner, H. Steen, J. A. J. Steen and F. A. Hamprecht, *BMC Bioinf.*, 2008, **9**, 355.
26. W. J. Browne, I. L. Dryden, K. Handley, S. Mian and D. Schadendorf, *J. R. Stat. Soc. Ser. C*, 2010, **59**, 617–633.
27. N. Pfeifer, A. Leinenbach, C. Huber and O. Kohlbacher, *BMC Bioinf.*, 2007, **8**, 468.
28. K. Shinoda, M. Tomita and Y. Ishihama, *Bioinformatics*, 2008, **24**, 1590–1595.
29. M. Vandenbogaert, S. Li-Thiao-Té, H.-M. Kaltenbach, R. Zhang, T. Aittokallio and B. Schwikowski, *Proteomics*, 2008, **8**, 650–672.
30. E. Lange, C. Gröpl, O. Schulz-Trieglaff, A. Leinenbach, C. Huber and K. Reinert, *Bioinformatics*, 2007, **23**, i273–i281.
31. J. Cox, M. Y. Hein, C. A. Luber, I. Paron, N. Nagaraj and M. Mann, *Mol. Cell. Proteomics*, 2014, **13**, 2513–2526.
32. A. Kassidas, J. F. MacGregor and P. A. Taylor, *AIChE J.*, 1998, **44**, 864–875.
33. N.-P. V. Nielsen, J. M. Carstensen and J. Smedsgaard, *J. Chromatogr. A*, 1998, **805**, 17–35.
34. C. Christin, H. C. J. Hoefsloot, A. K. Smilde, F. Suits, R. Bischoff and P. L. Horvatovich, *J. Proteome Res.*, 2010, **9**, 1483–1495.
35. J. T. Prince and E. M. Marcotte, *Anal. Chem.*, 2006, **78**, 6140–6152.
36. J. Listgarten, R. M. Neal, S. T. Roweis and A. Emili, *Advances in Neural Information Processing Systems*, MIT Press, 2005, pp. 817–824.
37. J. Listgarten, R. M. Neal, S. T. Roweis, P. Wong and A. Emili, *Bioinformatics*, 2007, **23**, e198–e204.
38. T.-H. Tsai, M. G. Tadesse, C. D. Poto, L. K. Pannell, Y. Mechref, Y. Wang and H. W. Ressom, *Bioinformatics*, 2013, **29**, 2774–2780.

39. H. Liao, E. Moschidis, I. Riba-Garcia, Y. Zhang, R. D. Unwin, J. S. Morris, J. Graham and A. W. Dowsey, *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, 2014, pp. 1332–1335.
40. W. Wang, H. Zhou, H. Lin, S. Roy, T. A. Shaler, L. R. Hill, S. Norton, P. Kumar, M. Anderle and C. H. Becker, *Anal. Chem.*, 2003, **75**, 4818–4826.
41. S. J. Callister, R. C. Barry, J. N. Adkins, E. T. Johnson, W. Qian, B.-J. M. Webb-Robertson, R. D. Smith and M. S. Lipton, *J. Proteome Res.*, 2006, **5**, 277–286.
42. Y. V. Karpievitch, T. Taverner, J. N. Adkins, S. J. Callister, G. A. Anderson, R. D. Smith and A. R. Dabney, *Bioinformatics*, 2009, **25**, 2573–2580.
43. M. R. Nezami Ranjbar, Y. Zhao, M. G. Tadesse, Y. Wang and H. W. Ressom, *Proteome Sci.*, 2013, **11**, 1–12.
44. Y. Lyutvinskiy, H. Yang, D. Rutishauser and R. A. Zubarev, *Mol. Cell. Proteomics*, 2013, **12**, 2324–2331.
45. B. Dost, N. Bandeira, X. Li, Z. Shen, S. P. Briggs and V. Bafna, *J. Comput. Biol.*, 2012, **19**, 337–348.
46. B. Carrillo, C. Yanofsky, S. Laboissiere, R. Nadon and R. E. Kearney, *Bioinformatics*, 2010, **26**, 98–103.
47. A. L. Oberg and O. Vitek, *J. Proteome Res.*, 2009, **8**, 2144–2156.
48. M. Anderle, S. Roy, H. Lin, C. Becker and K. Joho, *Bioinformatics*, 2004, **20**, 3575–3582.
49. Y. Benjamini and Y. Hochberg, *J. R. Stat. Soc. Ser. B*, 1995, **57**, 289–300.
50. L. J. E. Goeminne, A. Argentini, L. Martens and L. Clement, *J. Proteome Res.*, 2015, **14**, 2457–2465.
51. M. Choi, C.-Y. Chang, T. Clough, D. Broudy, T. Killeen, B. MacLean and O. Vitek, *Bioinformatics*, 2014, btu305.
52. T. Clough, S. Thaminy, S. Ragg, R. Aebersold and O. Vitek, *BMC Bioinf.*, 2012, **13**, S6.
53. Y. Karpievitch, J. Stanley, T. Taverner, J. Huang, J. N. Adkins, C. Ansong, F. Heffron, T. O. Metz, W.-J. Qian, H. Yoon, R. D. Smith and A. R. Dabney, *Bioinformatics*, 2009, **25**, 2028–2034.
54. L. J. E. Goeminne, K. Gevaert and L. Clement, *Mol. Cell. Proteomics*, 2015, **15**, 657–668.
55. K. Podwojski, M. Eisenacher, M. Kohl, M. Turewicz, H. E. Meyer, J. Rahnenfuehrer and C. Stephan, *Expert Rev. Proteomics*, 2010, **7**, 249–261.
56. M. Blein-Nicolas, H. Xu, D. de Vienne, C. Giraud, S. Huet and M. Zivy, *Proteomics*, 2012, **12**, 2797–2801.
57. A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari and D. B. Rubin, *Bayesian Data Analysis*, CRC Press, 3rd edn, 2013.

CHAPTER 8

MS2-Based Quantitation

MARC VAUDEL[a]

[a]Proteomics Unit, Department of Biomedicine, University of Bergen, Norway
*E-mail: marc.vaudel@uib.no

8.1 MS2-Based Quantification of Proteins

In gel-free proteomics, mass spectrometry signals specific to proteins are used to infer their abundance.[1] This can be achieved in a targeted or untargeted manner. In the latter, so-called shotgun proteomics, scientists aim at the quantification of entire proteomes.[2] As outlined in Chapter 1, due to the analytical challenges posed by the analysis of intact proteins, such approaches rely in their vast majority on the identification and quantification of proteins *via* their peptides, obtained after proteolytic digestion. In tandem mass spectrometry, the mass over charge ratios [m/z] of peptides are recorded, as well as the [m/z] of their fragment ions, obtained by fragmentation. The fragmentation can target specific peptides, so-called Data Dependent Acquisition (DDA), or fragment all the peptides ionized at a given time point in a mass window, so-called Data Independent Acquisition (DIA).

From the intensity of the signal recorded, it is possible to infer the abundance of ions measured, the abundance of peptides from MS spectra in MS1-based quantification, and the abundance of fragment ions from MS/MS spectra in MS2-based quantification. Specialist software tools were thus designed to extract protein specific intensities from mass spectra, allowing the estimation of protein abundances for entire proteomes. These quantification

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

approaches can be used to compare protein levels between samples (relative quantification), or to estimate the actual protein abundance in a given sample (absolute quantification). More details on protein quantification can be found in Chapter 7 and in specialist reviews.[2–5]

In this chapter, we focus on the case of DDA shotgun quantification based on MS/MS spectra. The two most encountered quantification methods relying mainly on MS2 spectra are spectrum counting techniques, where the number of spectra or peptides recorded for a given protein is used as a proxy to estimate its abundance, and reporter ion techniques, where peptides are labeled with an isobaric reagent releasing specific ions upon fragmentation, allowing the relative quantification of a peptide between multiple samples from a single spectrum. For each of these, we detail the rationale of the approach, identify the main pitfalls, and demonstrate its application using user-friendly open source software. Note that the concepts and techniques introduced in this chapter are generic, and can be transposed to other methods and software tools.

With this chapter, we aim at providing the reader with understanding needed to conduct sound MS2-based protein quantification and critically interpret the results. The feasibility and reliability of peptide quantification inferred from MS2 signals are critically discussed throughout the text. Finally, we provide elements to guide the decision between MS1 or MS2 quantification during experimental design.

8.2 Spectral Counting

As illustrated in Figure 8.1, spectral counting relies on the rationale that high intensity peptides have a higher probability to be selected for fragmentation, and therefore a higher chance of generating peptide spectrum matches (PSMs). Assuming that the intensity of a peptide is proportional to its abundance, and in turn to the original protein abundance, such methods use the number of PSMs for a given protein as a proxy to estimate its abundance. Various spectrum counting indexes are available, and can be categorized by the way spectra are counted: (1) all spectra are counted for a given protein, as in the NSAF index,[6] or (2) only one spectrum is counted per peptide, as in the emPAI index.[7] Because longer proteins are likely to produce more peptides, and thus more PSMs, the NSAF index further normalizes the spectral count assigned to a protein N_{spectra} to the protein length in amino acids, l_{protein}:

$$\text{NSAF} = \frac{N_{\text{spectra}}}{l_{\text{protein}}} \tag{8.1}$$

Similarly, for emPAI, the number of peptides observed, N_{observed}, is normalized to the number of theoretically observable peptides, $N_{\text{observable}}$, prior to exponential transformation:

$$\text{emPAI} = 10^{\frac{N_{\text{observed}}}{N_{\text{observable}}}} - 1 \tag{8.2}$$

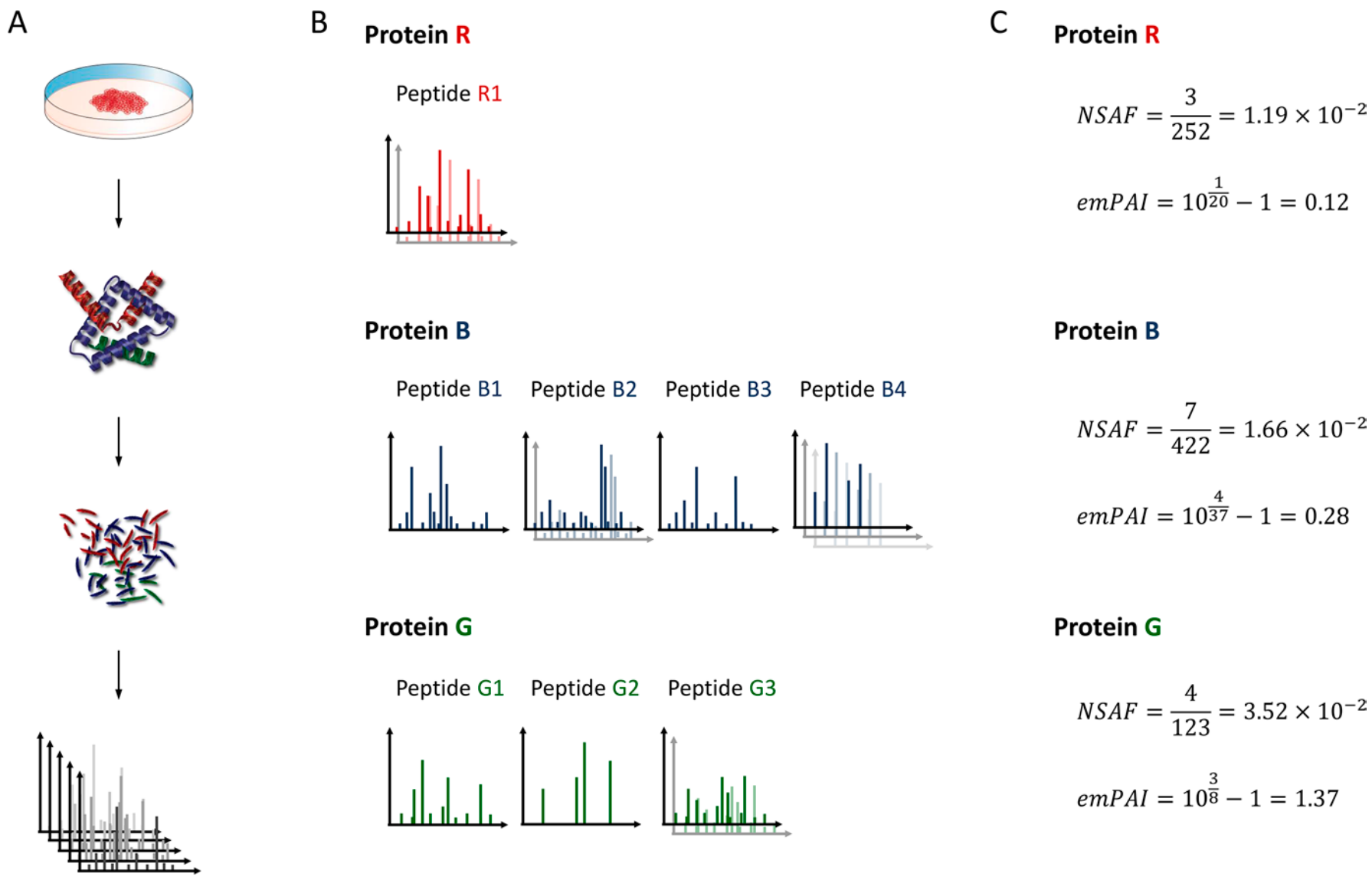

Figure 8.1 Spectrum counting quantification approaches rely on peptide or peptide-to-spectrum matches (PSM) counts to derive abundance indexes of proteins. (A) These methods do not require any additional sample preparation procedure; samples undergo the canonical proteomic workflow where proteins are proteolytically digested into peptides and analyzed by tandem mass spectrometry coupled to liquid chromatography, LC-MS/MS. (B) Upon identification of the fragment ion spectra acquired, a list of peptides and PSMs are available for every protein, as illustrated here with three proteins R, B, G, supported by the identification of 1, 4, and 3 peptides respectively. Multiple spectra supported the identification of the peptides R1, B2, B4, and G1, 3, 2, 3, and 2 spectra, respectively. Consequently, the proteins R, B, and G have 3, 7, and 4 PSMs, respectively. (C) As detailed in the text, the spectrum counting indexes NSAF and emPAI are defined by the formulas $\mathrm{NSAF} = \frac{N_{\mathrm{spectra}}}{l_{\mathrm{protein}}}$ and $\mathrm{emPAI} = 10^{\frac{N_{\mathrm{observed}}}{N_{\mathrm{observable}}}} - 1$. In this example, the proteins R, B, and G are of length 252, 422, and 123, respectively, and their *in silico* estimated number of observable peptides is 20, 37, and 8, respectively. Replacing the values in the formulas gives the protein abundance indexes for these proteins in this experiment.

$N_{\mathrm{observable}}$ is computed *in silico* from the protein sequence and search parameters. The detectability of a peptide depends on multiple factors, including its length, enzymatic cleavage status, amino acid composition, carried post-translational modifications, *etc.* While solutions have emerged to refine the predictability of the detection of peptides,[8,9] they remain challenging to integrate in desktop bioinformatic applications. Consequently, simpler estimators are used to compute $N_{\mathrm{observable}}$ from the *in silico* digestion of the protein sequence, returning the number of fully tryptic non-modified peptides within a given size range, typically 6 to 30 amino acids.

8.2.1 Implementations

Spectrum counting indexes are readily implemented in most proteomic platforms, and are also directly embedded in some identification search engines, as for example the emPAI index in Mascot[10] (Matrix Science, www.matrixscience.com). Note that the freedom left in the implementation of spectrum counting indexes makes these poorly comparable between bioinformatic tools. This is especially the case with emPAI, where the definition of a peptide and its conditions of observability are left to the interpretation of the developer.

In this chapter, we illustrate the simple estimation of spectrum counting indexes using the freely available PeptideShaker[11] software package. PeptideShaker supports various proteomic search engines, notably *via* SearchGUI[12] which allows the user friendly harnessing of eight search engines at time of writing: X!Tandem,[13] MyriMatch,[14] MS Amanda,[15] MS-GF+,[16] OMSSA,[17] Comet,[18,19] Tide,[20] and Andromeda.[21] For detailed instructions on how to operate these tools, please refer to the CompOmics Proteomics Bioinformatics tutorials[22] (compomics.com/bioinformatics-for-proteomics). Upon creation of a project, the spectrum counting index is displayed and visualized using a sparkline[23] for every protein in the protein table of the *Overview* tab as illustrated in Figure 8.2.

In the PeptideShaker implementation, since peptides can be shared between proteins, and appear multiple times in a single protein, the count of spectra for a peptide is weighted by its multiplicity among proteins and in protein sequences. Also, since large parts of the sequence of some proteins cannot be observed, mostly due to the dynamic range of the measurement and the complexity of spectra obtained from peptides with high charges, the normalization of the NSAF index is based on the observable length of the protein instead of the total length. The observable length is estimated based on the number of amino acids likely to be observed given the distance between two consecutive cleavage sites. The spectrum counting value for every protein in the result set is subsequently normalized to an absolute abundance using a reference total amount of proteins set by the user, or to a relative abundance in percent or ppm.

The user can select between the emPAI and NSAF techniques, and set preferences *via* the *Edit → Project Settings* menu. The spectrum counting indexes can be exported in text or Microsoft Excel .xls formats *via* the *Export → Identification Features* menu. There, the user can select preformatted reports, or design his or her own, choosing the inclusion of raw and normalized emPAI or NSAF indexes for every protein. These reports can readily be exported from the command line, and can thus be generated in batches. The inclusion of PeptideShaker in Galaxy[24,25] (see Chapter 13) and in the distributed computing platform Pladipus[26] allows the design of high throughput workflows.

8.2.2 Conclusion on Spectrum Counting

Due to differences in digestion, separation, ionization, and fragmentation efficiency of proteins and peptides, it can be anticipated that the number of spectra recorded for a protein will strongly depend on its physical and

Proteins (1193/1670 - 596 confident, 597 doubtful)
Peptides (34/36 - 34 confident, 0 doubtful)
Peptide Spectrum Matches (2/2 - 2 confident, 0 doubtful)
Spectrum & Fragment Ions (HR - NH2-AALAHSEEVTASQVAATK-COOH - TL 2+ 892.46 m/z)
Protein Sequence Coverage (10.67% - 10.2% confident, 0.0% doubtful, 0.47% not validated - 89.65% possible - 4684 AA)
MS2 Quant.
Overview
Spectrum IDs
Fractions
Modifications
3D Structures
Annotation
GO Analysis
Validation
QC Plots

Figure 8.2 The *Overview* tab of PeptideShaker displays identification results in a top-down view listing identified proteins at the top, peptides and PSMs of the selected protein and peptide, respectively, to the left, the annotated spectrum of the selected PSM to the right, and at the bottom the identified peptides annotated on the protein sequence. In the protein table at the top, the *MS2 Quant*. column, here highlighted with dashes, displays the spectrum counting index for every protein. For improved readability, the index is displayed using a sparkline (note the log scale).

chemical properties, and not only on its abundance. Consequently, inaccuracies in spectrum counting indexes are expected. Figure 8.3, plots the NSAF indexes obtained previously against the iBAQ values obtained by MS1 feature-based quantification using MaxQuant[27] (see Chapter 7). One can see that the iBAQ values span a much wider range than the NSAF; 95% of the data being comprised within 3.4 orders of magnitude for iBAQ against 1.7 for NSAF. The spectrum counting indexes thus have a narrower dynamic range than the intensity-based quantification.

While both indexes correlate linearly, the NSAF values vary by a ratio of 2.5 : 1 compared to the iBAQ values, for 50% of the values, and by a ratio of 15 : 1 for 95% of the values, illustrating the low precision of spectrum counting approaches. These are thus generally used to estimate the order of magnitude of the abundance and not for accurate quantification.

The quantification of low abundant proteins, and proteins yielding few peptides, relies on low counts of peptides and spectra, up to the extreme case of quantification based on a single spectrum. In such situations where the identification of an additional spectrum practically doubles the spectral count, the spectrum counting indexes also lack accuracy and robustness. To optimize the performance of spectrum counting quantification, the scientist needs to avoid the presence of false positive hits, while maximizing the number of spectra identified. This is achieved by tuning the stringency of the validation, which can be done in PeptideShaker in the *Validation* tab by

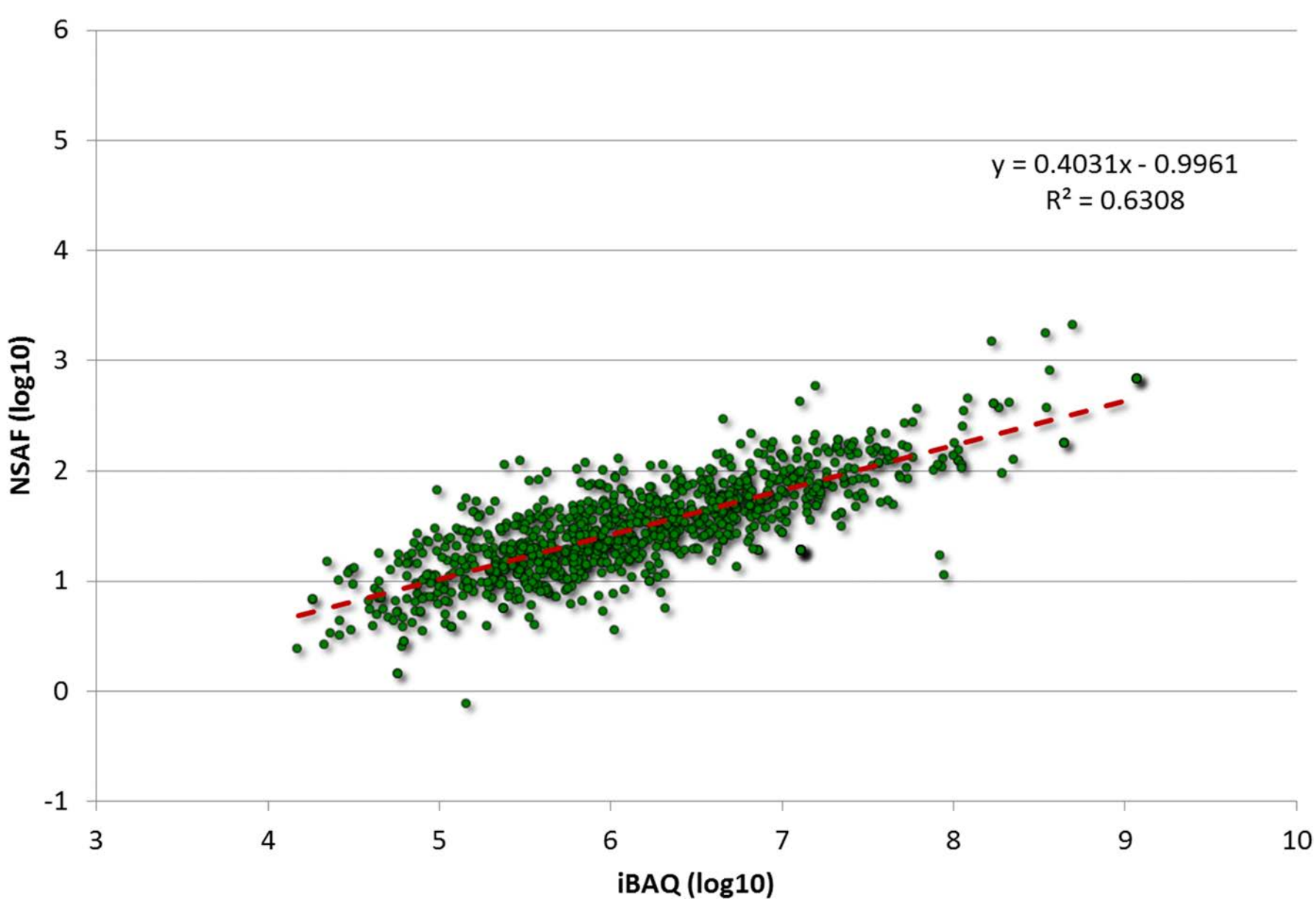

Figure 8.3 The NSAF indexes obtained from PeptideShaker are plotted against the iBAQ values obtained when processing the same dataset with MaxQuant. Only proteins common to both approaches are considered. Note that axes are in base 10 logarithmic scale.

balancing between the false positive and negative rates, respectively termed FDR and FNR.

The simplicity of the computation of spectrum counting indexes, and the fact that they do not generate additional costs, make them ideal candidates for rapid and simple evaluation of protein abundances. However, results should be interpreted in light of the performance. Consequently, spectrum counting indexes are mainly used to evaluate the order of magnitude of protein abundances. With the current instrumentation, whenever accurate absolute quantification is needed, the use of spiked-in standards and intensity-based quantification is required.

8.3 Reporter Ion-Based Quantification

As illustrated in Figure 8.4, in reporter ion-based quantification, chemical tags are used to label peptides from different samples, and multiplex them after digestion. As illustrated in Figure 8.5, the tags consist of three parts: the reporter group, the mass balancer group, and the reactive group. During labeling, the balancer and reporter groups are bound to peptides. Interestingly, as detailed in Tables 8.1–8.5, the mass of the reporter group is specific to every reagent, whereas the balancer counterpart ensures that the different reagents are isobaric.

As listed in Tables 8.1–8.5, at the time of writing, two reporter ion-based quantification techniques are available: iTRAQ® allows the multiplexing of four or eight samples,[28] and TMT the multiplexing of two, six, or ten samples.[29] Importantly, 6-plex and 10-plex TMT kits consist of the combination of two series of 3 and 5 reagents, respectively, where the mass difference is encoded using a ^{13}C or a ^{15}N. These so-called C and N series of ions are distinguishable by a mass difference corresponding to the difference in mass defect between C and N, 6.32 mDa:[30]

$$\left(^{13}\mathrm{C} - {}^{12}\mathrm{C}\right) - \left(^{15}\mathrm{N} - {}^{14}\mathrm{N}\right) = 6.32\ \mathrm{mDa} \quad (8.3)$$

Multiple samples are hence subjected to the same workflow, thereby reducing experimental variability between samples. The different samples remain indistinguishable until data interpretation, where comparing the relative intensities of the fragmented reporter groups, so-called reporter ions (sample specific fragment ions in MS2 spectra) allows the relative quantification of the peptide in the different multiplexed samples. By aggregating the relative intensities of reporter ions of all the spectra recorded for one protein, bioinformatic tools provide relative abundances for the proteins identified in a dataset.

Most proteomics bioinformatic platforms propose solutions for reporter ion-based quantification. Here again, the data processing only relies on identified spectra, and is thus conducted after identification of the dataset. We will illustrate the processing of such data using Reporter (http://compomics.github.io/projects/reporter.html), a user friendly interface to perform reporter ion-based quantification of data processed using PeptideShaker.

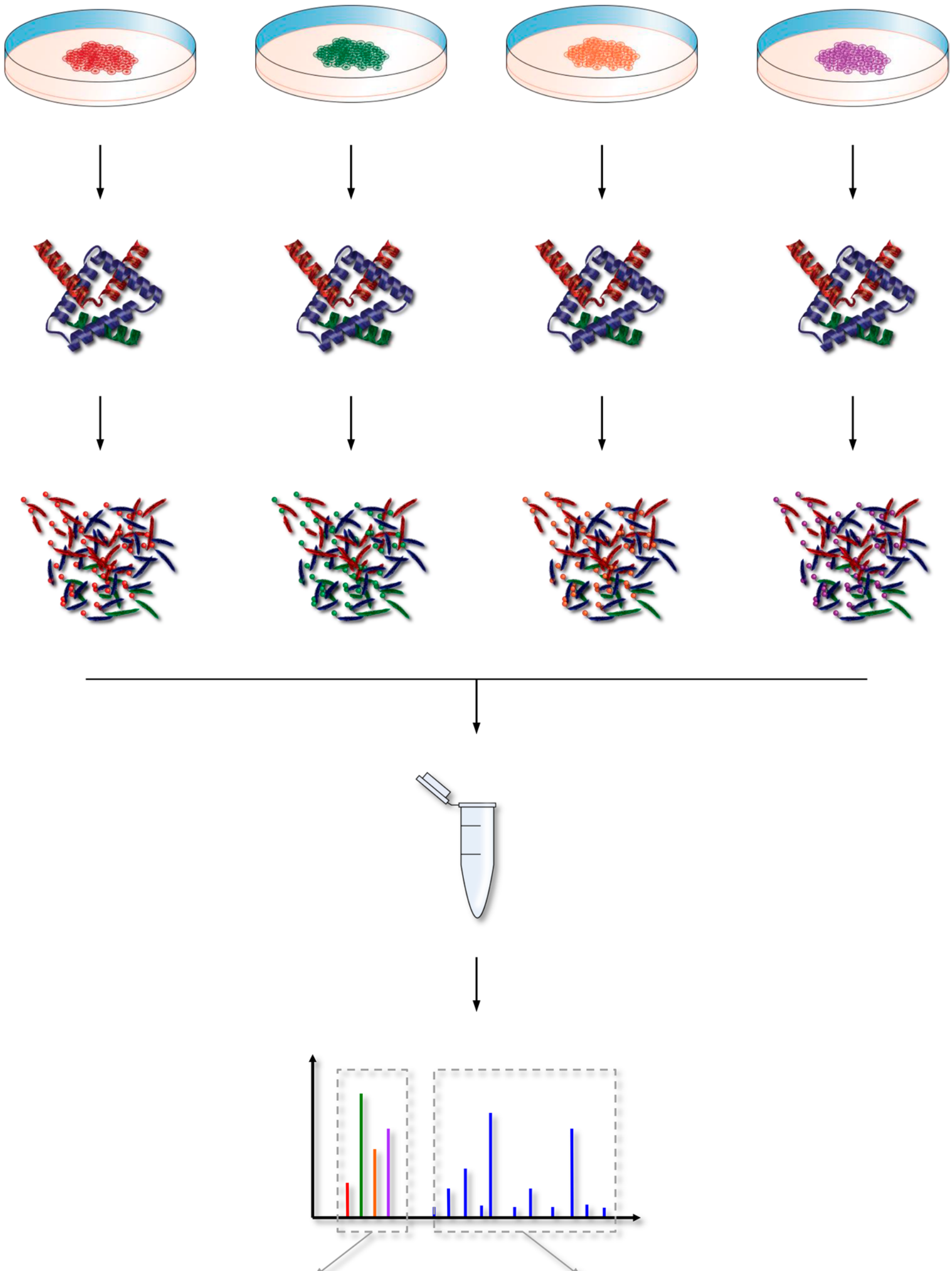

Reporter Intensities				Sequence
4	20	10	15	NLLDEELQR

Figure 8.4 In reporter ion-based quantification, different samples are digested in parallel, and the peptides obtained are labeled using isobaric tags, as illustrated here with four samples. After labeling, the peptides are pooled and undergo the same experimental workflow. Upon acquisition by tandem mass spectrometry coupled to liquid chromatography, LC-MS/MS, the MS2 spectra acquired after isolation and fragmentation of a peptide present sample specific reporter ions in their low mass region. As illustrated in the table at the bottom, the intensity of these peaks is used to estimate the abundance of the peptide in the different samples while the other peaks are used to infer the peptide sequence.

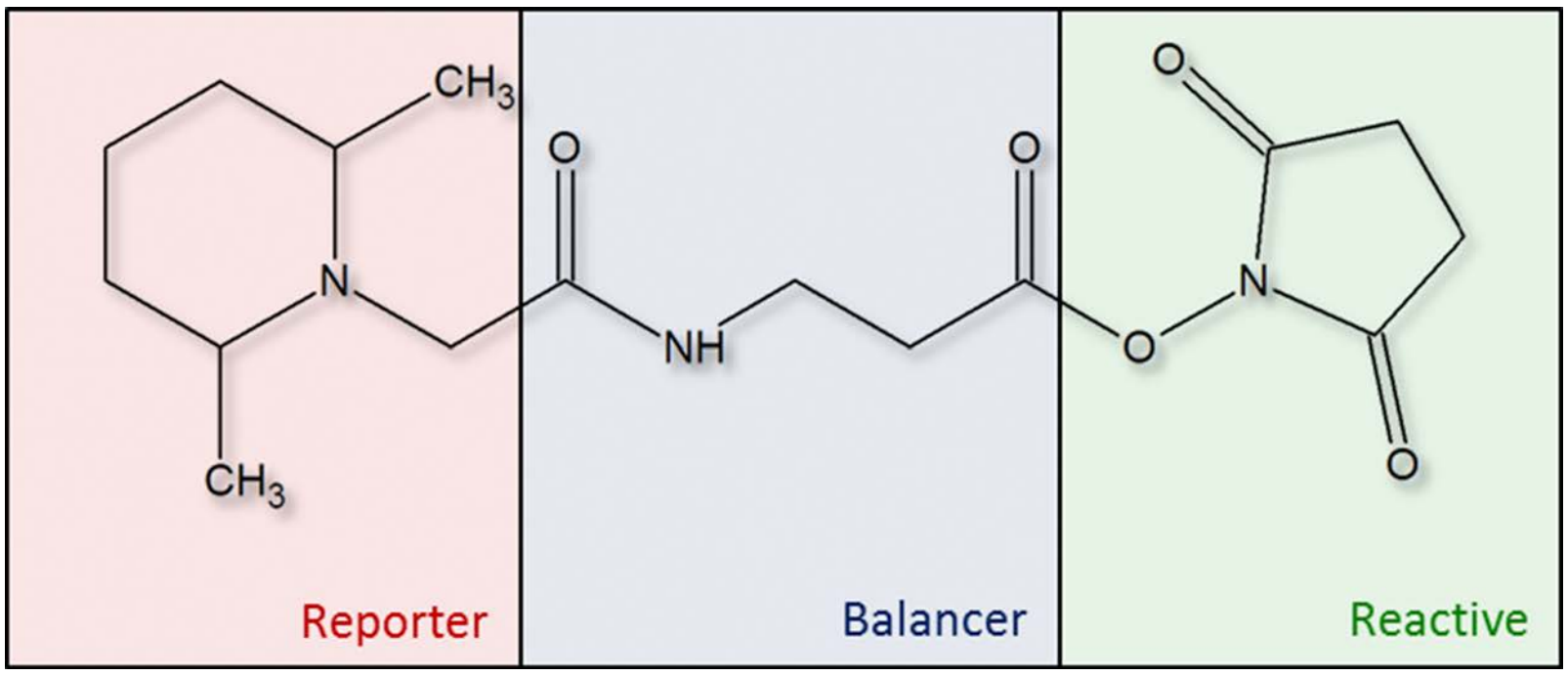

Figure 8.5 As exemplified here with a Tandem Mass Tag™ (TMT), see main text for details, the chemical tags used for peptide labeling consist of three parts: the reporter group to the left, the balancer group in the center, and the reactive group to the right. As detailed in Tables 8.1–8.5, the mass of the reporter groups varies between reagents through the incorporation of isotopes, but the respective balancer groups compensate the mass differences ensuring that all tags are isobaric.

Table 8.1 The iTRAQ 4-plex labeling kit allows the multiplexing of up to four different samples. The adduct mass is used as modification during the search, while the reporter ion masses are used to find the quantitative information in spectra. Note that the actual composition of the iTRAQ labels is not publicly available. The values given here are the ones used by default in the compomics-utilities package[52] for identification and quantification of iTRAQ datasets and are given without guarantee.

Label	Composition (reporter + balancer)	Adduct mass (Da)	Protonated reporter *m*/*z*
iTRAQ 4-plex 114	$C_5{}^{13}CH_{12}N_2 + {}^{13}C^{18}O$	144.10592	114.11068
iTRAQ 4-plex 115	$C_5{}^{13}CH_{12}N^{15}N + C^{18}O$	144.09960	115.10771
iTRAQ 4-plex 116	$C_4{}^{13}C_2H_{12}N^{15}N + {}^{13}CO$	144.10206	116.11107
iTRAQ 4-plex 117	$C_3{}^{13}C_3H_{12}N^{15}N + CO$	144.10206	117.11442

Table 8.2 The iTRAQ 8-plex labeling kit allows the multiplexing of up to eight different samples. The adduct mass is used as modification during the search, while the reporter ion masses are used to find the quantitative information in spectra. Note that the actual composition of the iTRAQ labels is not publicly available. The values given here are the ones used by default in the compomics-utilities package[52] for identification and quantification of iTRAQ datasets and are given without guarantee.

Label	Composition (reporter + balancer)	Adduct mass (Da)	Protonated reporter *m*/*z*
iTRAQ 8-plex 113	$C_6H_{12}N_2 + C_2{}^{13}C_6H_{12}{}^{15}N_2O_3$	304.19904	113.10732
iTRAQ 8-plex 114	$C_5{}^{13}CH_{12}N_2 + C_3{}^{13}C_5H_{12}{}^{15}N_2O_3$		114.11068
iTRAQ 8-plex 115	$C_5{}^{13}CH_{12}N^{15}N + C_3{}^{13}C_5H_{12}N^{15}NO_3$		115.10771
iTRAQ 8-plex 116	$C_4{}^{13}C_2H_{12}N^{15}N + C_4{}^{13}C_4H_{12}N^{15}NO_3$		116.11107
iTRAQ 8-plex 117	$C_3{}^{13}C_3H_{12}N^{15}N + C_5{}^{13}C_3H_{12}N^{15}NO_3$		117.11442
iTRAQ 8-plex 118	$C_3{}^{13}C_3H_{12}{}^{15}N_2 + C_5{}^{13}C_3H_{12}N_2O_3$		118.11146
iTRAQ 8-plex 119	$C_2{}^{13}C_4H_{12}{}^{15}N_2 + C_6{}^{13}C_2H_{12}N_2O_3$		119.11481
iTRAQ 8-plex 121	${}^{13}C_6H_{12}{}^{15}N_2 + C_8H_{12}N_2O_3$		121.12152

Table 8.3 The TMT^0 labeling kit is for method development and targeted quantification. The adduct mass is used as modification during the search, while the reporter ion masses are used to find the quantitative information in spectra.

Label	Composition (reporter + balancer)	Adduct mass (Da)	Protonated reporter *m*/*z*
TMT^0 126	$C_8H_{15}N + C_4H_5NO_2$	224.15248	126.12773

Table 8.4 The TMT^2 labeling kit is for the multiplexing of two samples. The adduct mass is used as modification during the search, while the reporter ion masses are used to find the quantitative information in spectra.

Label	Composition (reporter + balancer)	Adduct mass (Da)	Protonated reporter *m*/*z*
TMT^2 126	$C_8H_{15}N + C_3{}^{13}CH_5NO_2$	225.15583	126.12773
TMT^2 127	$C_7{}^{13}CH_{15}N + C_4H_5NO_2$		127.13108

Table 8.5 The TMT^6 and TMT^{10} labeling kits are for the multiplexing of six and ten samples, respectively. The adduct mass is used as modification during the search, while the reporter ion masses are used to find the quantitative information in spectra.

Label	Composition (reporter + balancer)	Adduct mass (Da)	Protonated reporter *m*/*z*
$TMT^{6/10}$ 126	$C_8H_{15}N + {}^{13}C_4H_5{}^{15}NO_2$	229.16293	126.12773
$TMT^{6/10}$ 127N	$C_8H_{15}{}^{15}N + {}^{13}C_4H_5NO_2$		127.12476
TMT^{10} 127C	$C_7{}^{13}CH_{15}N + C^{13}C_3H_5{}^{15}NO_2$		127.13108
TMT^{10} 128N	$C_7{}^{13}CH_{15}{}^{15}N + C^{13}C_3H_5NO_2$		128.12812
$TMT^{6/10}$ 128C	$C_6{}^{13}C_2H_{15}N + C_2{}^{13}C_2H_5{}^{15}NO_2$		128.13444
$TMT^{6/10}$ 129N	$C_6{}^{13}C_2H_{15}{}^{15}N + C_2{}^{13}C_2H_5NO_2$		129.13147
TMT^{10} 129C	$C_5{}^{13}C_3H_{15}N + C_3{}^{13}CH_5{}^{15}NO_2$		129.13779
TMT^{10} 130N	$C_5{}^{13}C_3H_{15}{}^{15}N + C_3{}^{13}CH_5NO_2$		130.13483
$TMT^{6/10}$ 130C	$C_4{}^{13}C_4H_{15}N + C_4H_5{}^{15}NO_2$		130.14115
$TMT^{6/10}$ 131	$C_4{}^{13}C_4H_{15}{}^{15}N + C_4H_5NO_2$		131.13818

8.3.1 Identification

Prior to quantification, iTRAQ or TMT spectral datasets undergo identification similarly as other proteomic datasets, producing a set of PSMs that are grouped to proteins. The only difference is the setting of modifications, which should account for the presence of the isobaric tags: a modification of +144.1 Da or +304.19904 Da on peptide N-termini (fixed), Lys (fixed), and Tyr (variable) for iTRAQ 4-plex and 8-plex, respectively, and of +225.158 Da or +229.1629 Da on peptide N-termini and Lys (both fixed) for TMT 2-plex

and 6 or 10-plex, respectively. Note that iTRAQ 4-plex tags are not completely isobaric, and a consensus mass is used for the search, either based on the average mass or by selecting one of the reagent masses.[31] This notably does not make iTRAQ particularly suited for high resolution mass spectrometry. Conversely, TMT requires a resolution of >50 000 at 150 *m*/*z* to resolve the C- and N-ion series of TMT 10-plex kits, and it is advised to use the same resolution to resolve the isotopic patterns of the C- and N-ion series of TMT 6-plex kits. At such high resolutions, most search engines will provide enhanced identification rates when using a stringent MS2 *m*/*z* tolerance (≤0.01 Da or 10 ppm); it is thus recommended to optimize this search setting and not reuse search parameters designed for lower resolution.[32] Upon identification, the reporter ion peaks are visible in the low *m*/*z* range of the spectrum, as illustrated in Figure 8.6.

8.3.2 Reporter Ion Intensities, Interferences and Deisotoping

The first step of the quantification procedure is to extract sample specific reporter ion intensities from spectra. Spectra result from the measurement of compound intensities over the entire mass range of the instrument where peaks are represented as a bell-shape curve resulting from technical variability in the mass detection. This profile needs to be centroided (integrated and transformed into a single peak) prior to processing using a search engine, resulting in peak lists, easier to handle by identification algorithms and more compact in terms of memory. The quality of the *m*/*z* centroiding will mainly affect identification efficiency, while the quality of the intensity integration will affect the accuracy of the quantification.[33] This so-called peak picking step can be conducted by the instrument signal processing module, or using third party software.[33,34]

Due to the presence of isotopes in reagents, mostly ^{13}C, the reporter ions from a sample distribute on multiple masses. The software thus needs to deconvolute the intensities present over the different masses to infer the contribution of every sample. For this, it is necessary to account for the coefficients of purity provided by the manufacturer with the kit as illustrated in Figure 8.7 with TMT 10-plex reagents. Note that for the sake of reproducibility, it is recommended to provide this information along with the data when publishing the results.

When isolating a peptide for fragmentation, the mass spectrometer might also include co-eluting peptides.[35] While these are barely noticeable in the identified spectra, because less abundant, their contribution in the reporter ion masses will introduce a background intensity and thus impair the quality of the quantification. This effect is called precursor ion interference in the literature and results in a compression of relative abundances towards a 1 : 1 ratio,[36–39] 0 in logarithmic scale. Several experimental procedures exist to alleviate this problem,[40–43] the simplest being to reduce MS1 complexity by fractionation.[44,45] Additionally, several tools including the

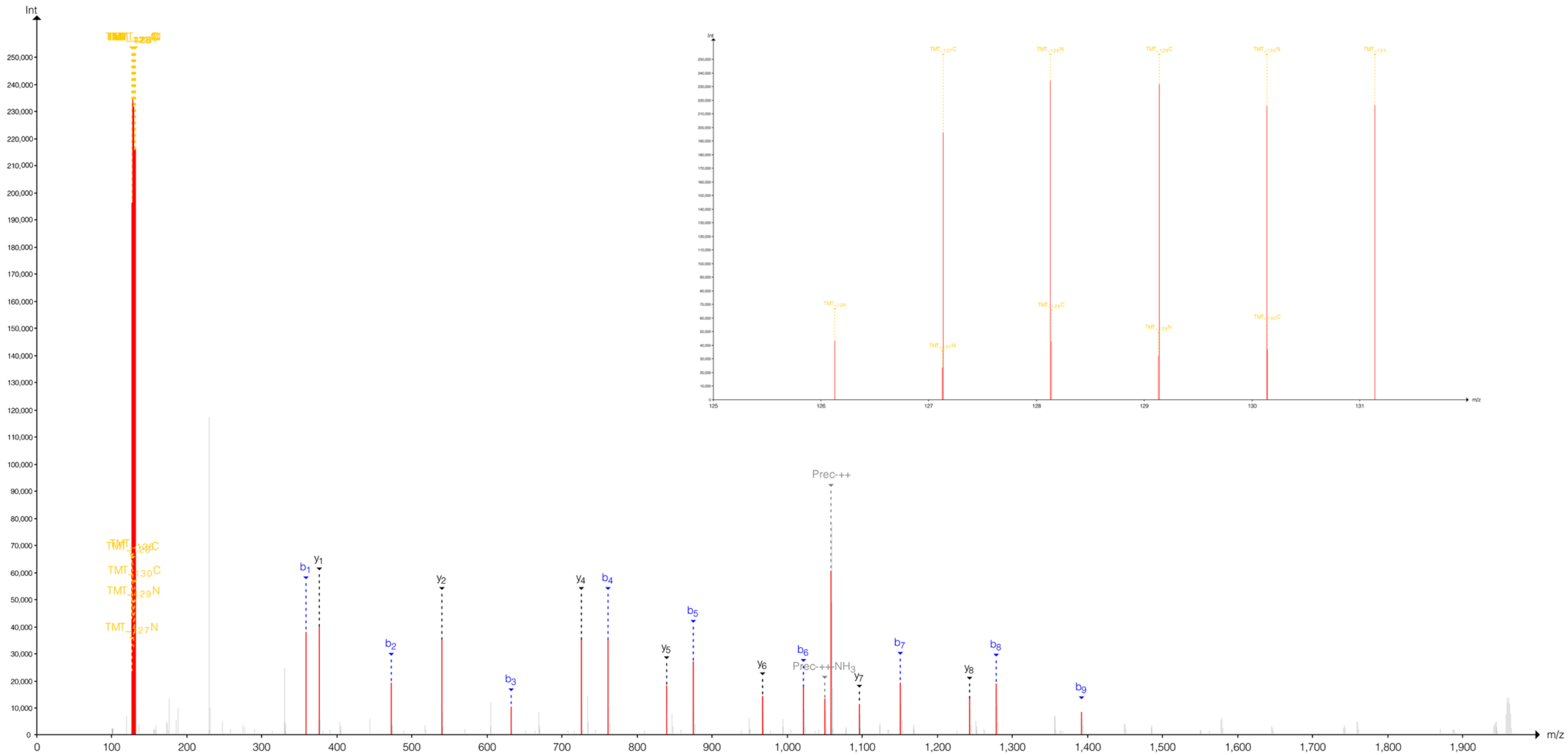

Figure 8.6 The fragment ion spectra of isobaric tag labeled peptides consist of peptide fragment ions with reporter ions in the low *m*/*z* range as illustrated here with a spectrum recorded during a TMT 10-plex experiment. The b- and y-ions obtained from the fragmentation of the spectrum are annotated. The TMT reporter ions are visible in the 126–131 *m*/*z* range. A zoom on this region displayed on the upper right corner shows the N and C reporter ion series with different intensities. These intensities are used to estimate the abundance of the peptide in the multiplexed samples.

ThermoFisher SCIENTIFIC

The world leader in serving science

PRODUCT DATA SHEET

TMT10plex™ Label Reagent Set

Product Number: 90110B **Lot Number:** OJ190396B

Form: The TMT10plex™ Label Reagents are supplied dried, 0.8 mg/tube. Make a stock solution by reconstituting each tube with 41 µl acetonitrile.

Note: Please refer to Table 3 of the instruction booklet for cross-referenced TMTsixplex products, reporter ion masses or mass tolerance window in your data analysis software.

**Reporter ion isotopic distributions are for informational use only and are not required isotope correction factors for Proteome Discoverer software TMT10plex quantitation method. Reporter ion isotopic distributions (-2, -1, +1, +2) are primarily for carbon isotopes with reporter ion interference for each mass tag shown in parentheses.

****Reporter Ion Isotopic Distributions:**

Mass Tag	Reporter Ion	-2	-1	Monoisotopic	+1	+2
TMT^{10}-126	126.127726	0	0	**100%**	4.69 (127C)	0 (128N)
TMT^{10}-127N	127.124761	0	0.4	**100%**	6.5 (128N)	0 (128C)
TMT^{10}-127C	127.131081	0	0.2 (126)	**100%**	4.6 (128C)	0.3 (129N)
TMT^{10}-128N	128.128116	0	0.9 (127N)	**100%**	4.7 (129N)	0.2 (129C)
TMT^{10}-128C	128.134436	0.10 (126)	0.53 (127C)	**100%**	2.59 (129C)	0 (130N)
TMT^{10}-129N	129.131471	0 (127N)	0.73 (128N)	**100%**	2.49 (130N)	0 (130C)
TMT^{10}-129C	129.137790	0 (127C)	1.3 (128C)	**100%**	2.5 (130C)	0 (131)
TMT^{10}-130N	130.134825	0 (128N)	1.2 (129N)	**100%**	2.8 (131)	2.7
TMT^{10}-130C	130.141145	0.1 (128C)	2.9 (129C)	**100%**	2.9	0
TMT^{10}-131	131.138180	0 (129N)	2.36 (130N)	**100%**	1.43	0

Stability: One year from date of product receipt.

Storage: Store at -20°C.

Pierce Biotechnology
3747 N. Meridian Road
PO Box 117
Rockford, IL 61105
USA
(800) 874-3723
(815) 968-0747
(815) 968-7316 fax
www.thermo.com/pierce

Figure 8.7 Example of product sheet for TMT 10-plex reagents. The table in the centre displays purity coefficients for the different reagents. These coefficients need to be provided to the analysis software that will proceed to an isotopic correction of the measured reporter ion intensities.

widely used MaxQuant software,[27] allow filtering peptides presenting risks of ion interference based on the signal to noise level in the isolation window of the precursor ion.

8.3.3 Ratio Estimation and Normalization

In order to convey the quantitative information to the protein level, reporter ion data interpretation tools aggregate the intensities recorded in all spectra of all peptides ascribed to a given protein. Spectral intensities are, however, not directly comparable from one spectrum to another due to the differences in peptide ionization, isolation and fragmentation. A ratio of the different deisotoped peak intensities to a reference is then established for every spectrum, as illustrated in eqn (8.4) with r_{Spectrum_k} the ratio of a reagent k between the respective deisotoped intensity I_{0k} and $I_{\text{ref}_{\text{Spectrum}}}$, the reference intensity for this spectrum.

$$r_{\text{Spectrum}_k} = \frac{I_{0_k}}{I_{\text{ref}_{\text{Spectrum}}}} \tag{8.4}$$

The reference intensity I_{ref} can be one peak's intensity, a combination of different peak intensities, or any reference intensity in the spectrum. In the latter case, a noise level can be selected as reference, and the ratio will be equivalent to a signal-to-noise ratio as implemented in the latest versions of Proteome Discoverer™ (Thermo Scientific™). The choice of this reference will have a direct impact on the final calculation as variability on this reference will propagate to the downstream processing.[46]

It is important to note that ratios do not distribute normally; as a result, most statistical estimators designed for normally distributed populations, are not suitable to work on ratios. A simple approach to alleviate this problem is to perform a logarithmic transformation before working on ratios, as reviewed in ref. 47. Throughout the chapter, ratios thus implicitly refer to the logarithmically transformed ratios.

As illustrated in eqn (8.5) and (8.6), for every reagent k, the spectrum level ratios, r_{Spectrum}, are subsequently aggregated by the software to estimate peptide level ratios, r_{Peptide}, and in turn, protein level ratios, r_{Protein}. An estimator f is used to draw a representative peptide level ratio from the distribution of n spectrum level ratios obtained from the PSMs ascribed to this peptide. Similarly, it is used to draw a representative protein level ratio from the distribution of the p peptide level ratios ascribed to this protein.

$$r_{\text{Peptide}_k} = f\left(r_{\text{Spectrum}_{k_1}}, \ldots, r_{\text{Spectrum}_{k_n}}\right) \tag{8.5}$$

$$r_{\text{Protein}_k} = f\left(r_{\text{Peptide}_{k_1}}, \ldots, r_{\text{Peptide}_{k_p}}\right) \tag{8.6}$$

Several estimators exist to combine spectrum and peptide level ratios into peptide and protein level ratios, respectively, with different levels of accuracy and robustness toward outliers. The most encountered are median – robust but not accurate; mean – accurate but not robust; or other statistical functions like maximum likelihood estimators which offer a good balance between accuracy and robustness.[37]

Additionally, it is possible to remove outliers in order to reduce their influence on the final ratio. Specific spectra can also be excluded from the peptide ratio calculation, or their contribution can be weighted, depending on spectrum quality metrics like intensity levels or a signal to noise ratio. Spectra can also be excluded from peptide ratio estimation based on the presence of other peaks surrounding the precursor in the MS1 spectrum, hinting at possible peptide interference. Similarly, it is possible to exclude peptides from protein ratio calculation based on their modification or cleavage statuses. Since their ratio is the result of the contribution of multiple proteins, peptides shared between different proteins also induce ratio distortion.[48] It is therefore preferable to rely on peptides unique to a protein or a protein ambiguity group. Conversely, relying on fewer peptides can reduce the robustness of the ratio estimation procedure.

Ultimately, the software provides such ratios for every peptide and protein. Despite sample equalization prior to labeling, due to difference in the handling of samples, offsets can be observed on a given channel, resulting in the distribution of ratios not being centred around 1 : 1, 0 in logarithmic scale. These are typically dilution effects prior to labeling or errors in peptide or protein total amount measurement, and can be corrected by normalization on all the ratios of a channel. The normalization can be done both at the protein and peptide level, in order to correct for biases introduced before and after digestion. It needs to be adapted to a hypothesis of stability: if the majority of proteins are presumed to be stable, the median of ratios should correspond to a 1 : 1 ratio, on the other hand if specific proteins are known to be stable and the background to vary, like after immunoprecipitation or when working with some biofluids,[49] then these stable proteins should be used as reference, and the median of all ratios will vary between channels. Good examples of proteins that are typically not stable between samples and should be excluded prior to normalization are contaminants.

8.3.4 Implementation

In this chapter, we demonstrate the estimation of peptide and protein ratios from isobaric tags using Reporter (http://compomics.github.io/projects/reporter.html). Reporter operates directly on PeptideShaker projects saved as *.cpsx* files. For details on how to create a PeptideShaker project, please refer to the CompOmics Proteomics Bioinformatics tutorials[22] (compomics.com/bioinformatics-for-proteomics). After starting Reporter by double-clicking the *.jar* file in the Reporter folder, click on *New Project*. The dialog of Figure 8.8 allows the creation of a project.

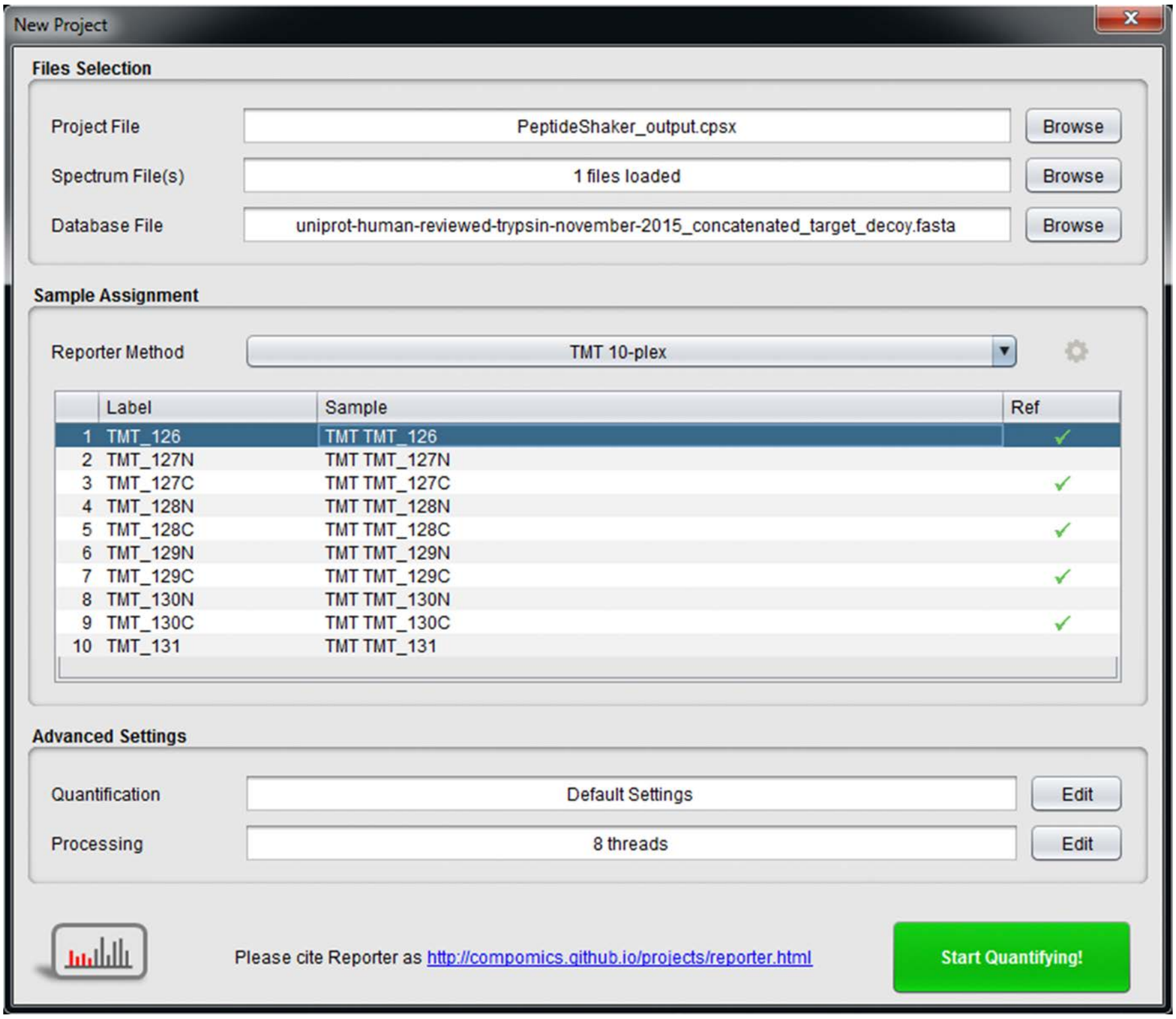

Figure 8.8 The *New Project* dialog of Reporter makes it possible to set up the quantification procedure of a PeptideShaker project. At the top, the *Files Selection* panel allows selecting the PeptideShaker project as *.cpsx* file as well as the spectrum and protein database files. In the *Sample Assignment* panel, the user can select the quantification method in the drop down menu and set the purity coefficients (see Figure 8.7) after clicking on the cogwheel. The table allows assigning samples to labels, and selecting reference samples for reporter intensities normalization (see main text). Quantification and processing settings can be edited in the *Advanced Settings* panel. Once the project is set, clicking the *Start Quantifying!* button launches the quantification process.

In the *Files Selection* panel at the top of the dialog, the PeptideShaker project file can be selected. The *Sample Assignment* panel in the centre allows selecting the quantification method, assigning samples to labels, and selecting references. The reference intensities are used to normalize spectral intensities, and if none is selected, a median of all non-null intensities will be used. Clicking the cogwheel next to the method selection drop down menu allows setting the isotope coefficients of the kit used for the experiment as displayed in Figure 8.9.

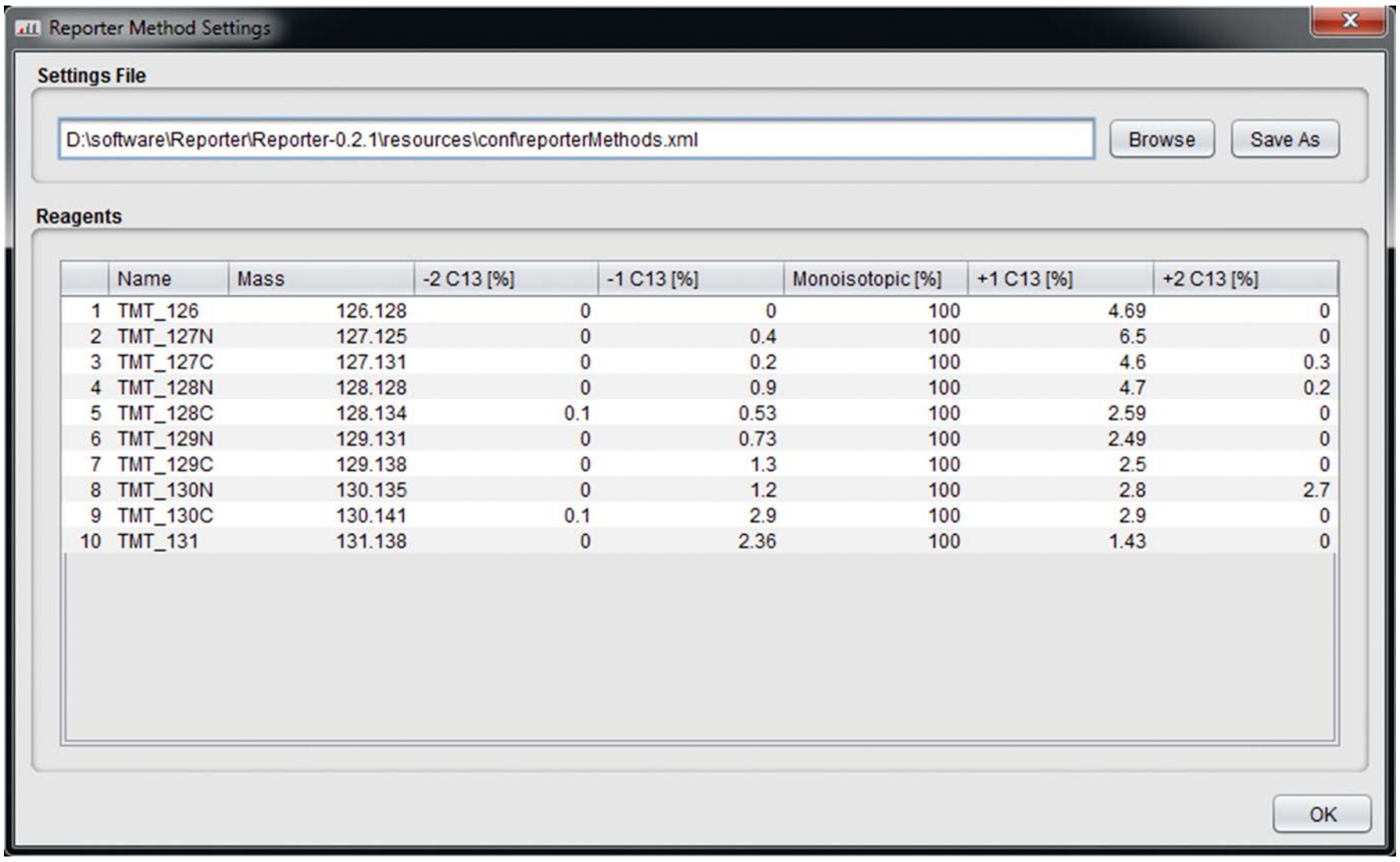

Figure 8.9 When setting up a new project (see Figure 8.8 and main text), it is important to provide the purity coefficients of the kit used to label the samples. The coefficients can be set in the table and saved in a file.

Finally, the *Advanced Settings* panel at the bottom allows setting quantification and processing settings. The quantification settings are separated into three categories, as displayed in Figure 8.10(A): (1) *Reporter Ions* settings, (2) *Ratio Estimation* settings, and (3) *Normalization* settings. As illustrated in Figure 8.10(B), the reporter ions settings allow the specification of an *m*/*z* tolerance for the matching of reporter ions, and the setting of a precursor *m*/*z* and retention time window in case the reporter ions are not recorded in the spectrum used for identification.[50] As illustrated in Figure 8.10(C), the ratio estimation settings allow selecting proteins, peptides, and PSMs according to their identification validation status in order to retain only high scoring hits. It is furthermore possible to parametrize the maximum likelihood estimator used to aggregate ratios at the peptide and protein level, and exclude modified or miscleaved peptides as detailed in ref. 37. As illustrated in Figure 8.10(D), the normalization settings allow selecting the statistical estimator to use for channel normalization at the PSM, peptide, and protein level. It is also possible to provide lists of stable proteins and contaminants in the FASTA format. Note that the common Repository of Adventitious Proteins, cRAP, from the Global Proteome Machine[51] (thegpm.org/crap) is selected as the default list of contaminants.

After clicking the *Start Quantifying!* button, Reporter will estimate ratios according to the specified settings, and display the protein ratios clustered using k-means clustering as implemented in the compomics-utilities

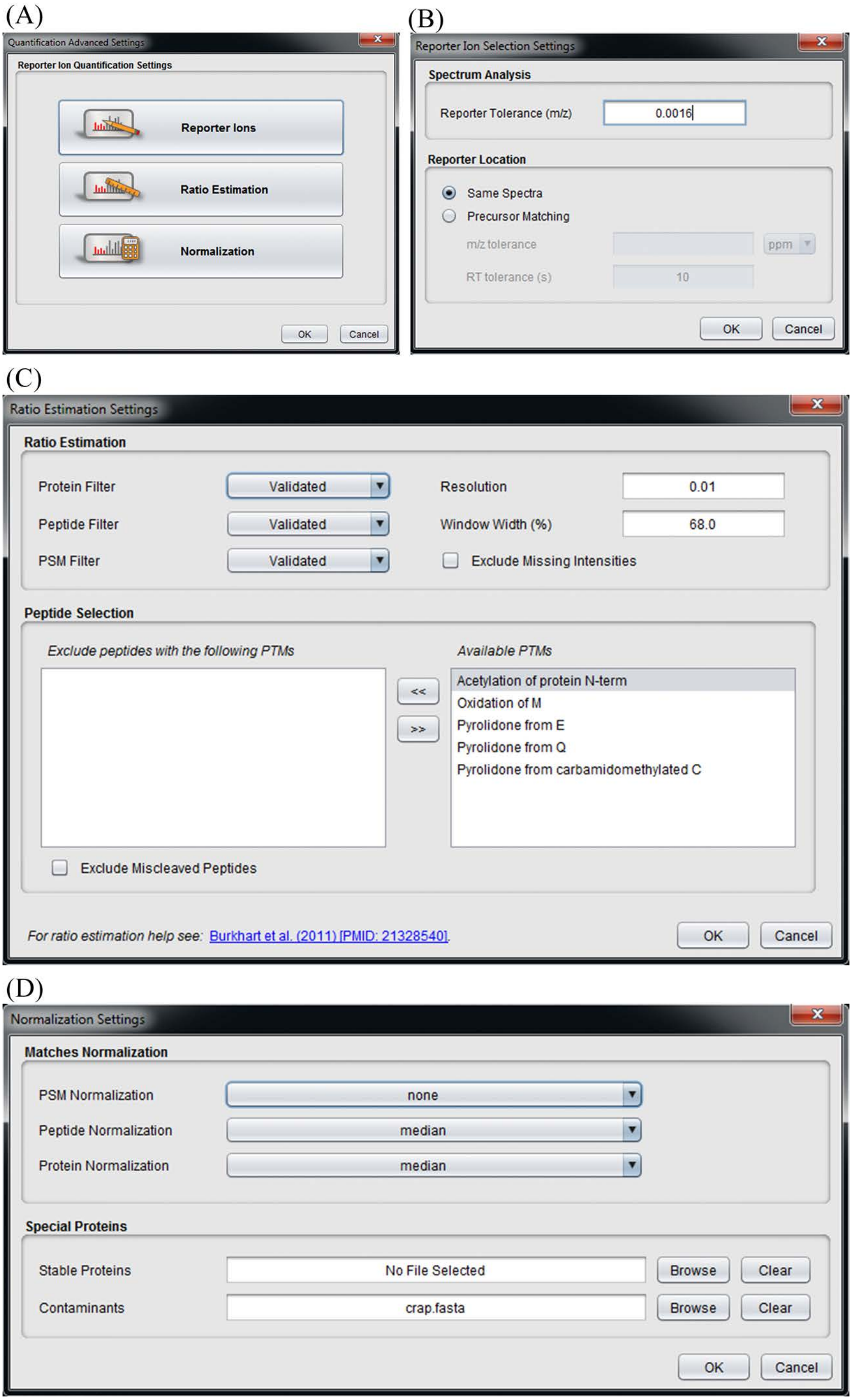

Figure 8.10 The quantification settings can be categorized into (1) *Reporter Ions* settings, (2) *Ratio Estimation* settings, and (3) *Normalization* settings. (A) The *Quantification Settings* can be edited *via* the *Quantification*

package.[52] As illustrated in Figure 8.11, it is possible to navigate the clusters and visualize the protein profiles. Finally, PSM, peptide, and protein level ratios can be exported *via* the *Export → Quantification Features* menu.

8.3.5 Conclusion on Reporter Ion-Based Quantification

As mentioned, one of the main shortcomings of reporter ion quantification is the inaccuracy of the obtained ratios, notably due to the problem of ion interference during co-isolation of peptides. Another critical problem is the reduced identification rate observed in reporter ion-labeled samples in comparison to label-free experiments. This can be attributed to the increased MS1 complexity due to sample multiplexing, to altered fragmentation of peptides due to the label, and to the increased precursor charge.[53] If the experiment allows it, when high accuracy quantification or high sample coverage are needed, it is thus recommended to proceed to targeted quantification with spiked labeled peptides or to label free intensity-based quantification.

Nevertheless, the ability to multiplex samples, the simplicity of the protocol, and the straightforward data interpretation make reporter ion quantification methods extremely competitive. They are notably of high interest whenever the sample preparation procedure includes steps that could introduce variability between samples. This is, for example, the case in post-translational modification analyses, where the low reproducibility of the enrichment procedures for modified peptides can impair the label free comparison of peptide intensities. The ability to multiplex up to ten samples makes reporter ion-based quantification attractive for large scale discovery proteomic studies. By combining multiple kits using a common reference as done in super-SILAC approaches,[54] it is possible to drastically reduce the sample preparation and acquisition time.

Advanced Settings dialog available when creating a new project (see Figure 8.8 and main text.). Clicking each category opens the respective dialogs displayed in B, C, and D. (B) It is possible to set the m/z tolerance to use when selecting the reporter ions as well as an m/z and retention time window in case the reporter ions are not in the same spectrum as the peptide. (C) The ratio estimation settings allow filtering proteins, peptides, and PSMs based on the quality of their identification using the three drop down menus to the left of the *Ratio Estimation* panel. It is possible to parametrize the maximum likelihood estimator to use for peptide and protein ratio estimation, and notably set its resolution and window width. It is "also possible to exclude missing intensities from the ratio calculation. The Peptide Selection panel allows excluding peptides based on their modification and cleavage statuses. (D) The normalization for each sample can be done at the PSM, peptide, and protein level and selected in the *Matches Normalization* panel. Lists of proteins to consider stable or contaminants can be provided in the *Special Proteins* panel.

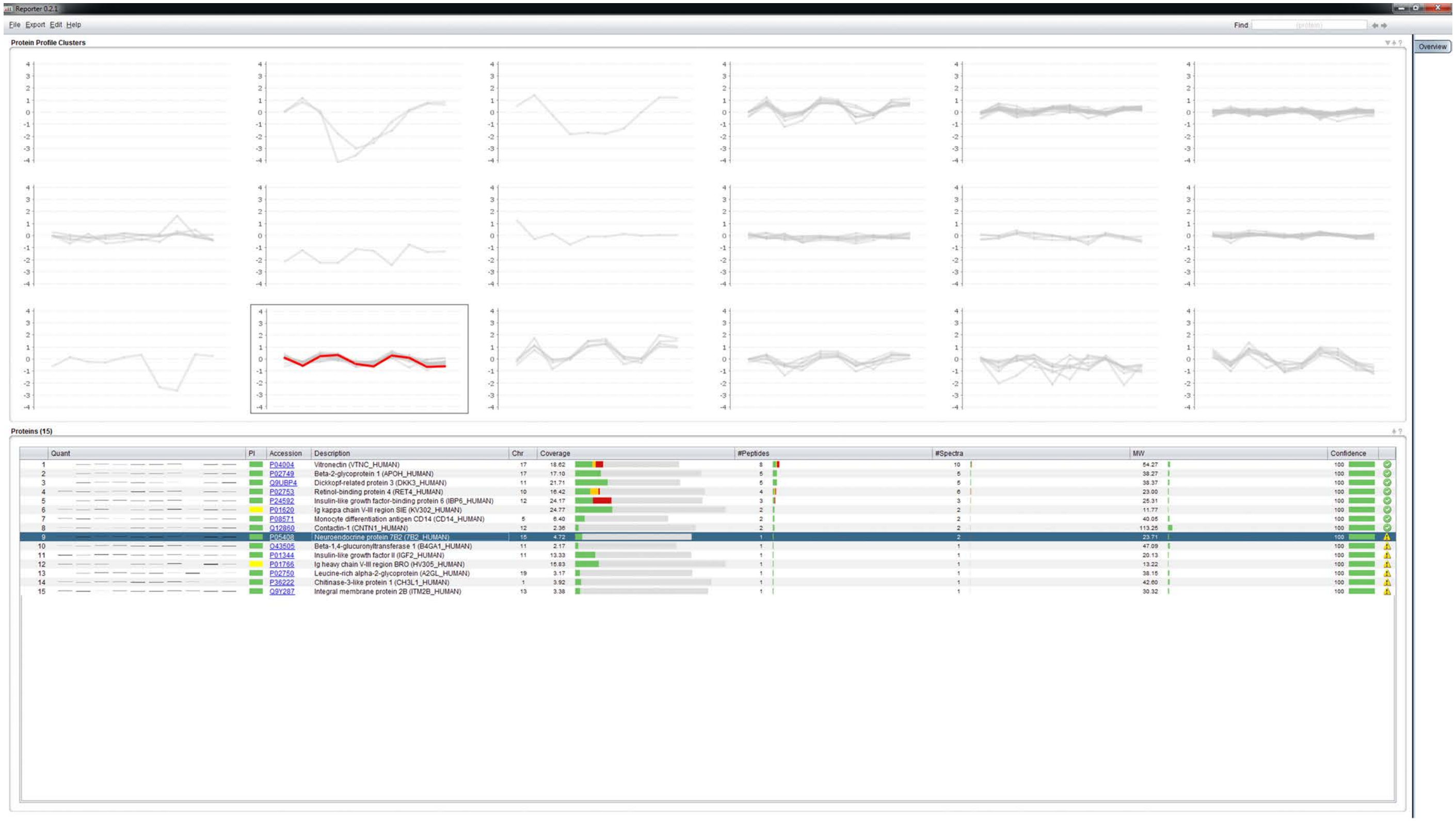

Figure 8.11 After loading of the project, the protein profiles appear after clustering in the top panel of the main interface. After clicking on a specific cluster, the list of proteins in this cluster is displayed at the bottom. Selecting a protein of interest will highlight it in red in the cluster as displayed here with the bold line in the second cluster of the bottom row. At the top right of the cluster panel, it is possible to change the parameters of the clustering algorithm.

Acknowledgements

The authors declare no competing financial interests.

References

1. R. Aebersold and M. Mann, *Nature*, 2003, **422**, 198–207.
2. B. Domon and R. Aebersold, *Science*, 2006, **312**, 212–217.
3. M. Bantscheff, M. Schirle, G. Sweetman, J. Rick and B. Kuster, *Anal. Bioanal. Chem.*, 2007, **389**, 1017–1031.
4. M. Bantscheff, S. Lemeer, M. M. Savitski and B. Kuster, *Anal. Bioanal. Chem.*, 2012, **404**, 939–965.
5. M. Vaudel, A. Sickmann and L. Martens, *Proteomics*, 2010, **10**, 650–670.
6. A. C. Paoletti, T. J. Parmely, C. Tomomori-Sato, S. Sato, D. Zhu, R. C. Conaway, J. W. Conaway, L. Florens and M. P. Washburn, *Proc. Natl. Acad. Sci. U. S. A.*, 2006, **103**, 18928–18933.
7. Y. Ishihama, Y. Oda, T. Tabata, T. Sato, T. Nagasu, J. Rappsilber and M. Mann, *Mol. Cell. Proteomics*, 2005, **4**, 1265–1272.
8. T. Fannes, E. Vandermarliere, L. Schietgat, S. Degroeve, L. Martens and J. Ramon, *J. Proteome Res.*, 2013, **12**, 2253–2259.
9. E. Vandermarliere and L. Martens, *Proteomics*, 2013, **13**, 1028–1035.
10. D. N. Perkins, D. J. Pappin, D. M. Creasy and J. S. Cottrell, *Electrophoresis*, 1999, **20**, 3551–3567.
11. M. Vaudel, J. M. Burkhart, R. P. Zahedi, E. Oveland, F. S. Berven, A. Sickmann, L. Martens and H. Barsnes, *Nat. Biotechnol.*, 2015, **33**, 22–24.
12. M. Vaudel, H. Barsnes, F. S. Berven, A. Sickmann and L. Martens, *Proteomics*, 2011, **11**, 996–999.
13. R. Craig and R. C. Beavis, *Bioinformatics*, 2004, **20**, 1466–1467.
14. D. L. Tabb, C. G. Fernando and M. C. Chambers, *J. Proteome Res.*, 2007, **6**, 654–661.
15. V. Dorfer, P. Pichler, T. Stranzl, J. Stadlmann, T. Taus, S. Winkler and K. Mechtler, *J. Proteome Res.*, 2014, **13**, 3679–3684.
16. S. Kim and P. A. Pevzner, *Nat. Commun.*, 2014, **5**, 5277.
17. L. Y. Geer, S. P. Markey, J. A. Kowalak, L. Wagner, M. Xu, D. M. Maynard, X. Yang, W. Shi and S. H. Bryant, *J. Proteome Res.*, 2004, **3**, 958–964.
18. J. K. Eng, T. A. Jahan and M. R. Hoopmann, *Proteomics*, 2013, **13**, 22–24.
19. J. K. Eng, M. R. Hoopmann, T. A. Jahan, J. D. Egertson, W. S. Noble and M. J. MacCoss, *J. Am. Soc. Mass Spectrom.*, 2015, **26**, 1865–1874.
20. B. J. Diament and W. S. Noble, *J. Proteome Res.*, 2011, **10**, 3871–3879.
21. J. Cox, N. Neuhauser, A. Michalski, R. A. Scheltema, J. V. Olsen and M. Mann, *J. Proteome Res.*, 2011, **10**, 1794–1805.
22. M. Vaudel, A. S. Venne, F. S. Berven, R. P. Zahedi, L. Martens and H. Barsnes, *Proteomics*, 2014, **14**, 1001–1005.
23. H. Barsnes, M. Vaudel and L. Martens, *Proteomics*, 2015, **15**, 1428–1431.
24. B. Giardine, C. Riemer, R. C. Hardison, R. Burhans, L. Elnitski, P. Shah, Y. Zhang, D. Blankenberg, I. Albert, J. Taylor, W. Miller, W. J. Kent and A. Nekrutenko, *Genome Res.*, 2005, **15**, 1451–1455.

25. J. Boekel, J. M. Chilton, I. R. Cooke, P. L. Horvatovich, P. D. Jagtap, L. Kall, J. Lehtio, P. Lukasse, P. D. Moerland and T. J. Griffin, *Nat. Biotechnol.*, 2015, **33**, 137–139.
26. K. Verheggen, D. Maddelein, N. Hulstaert, L. Martens, H. Barsnes and M. Vaudel, *J. Proteome Res.*, 2016, **15**, 707–712.
27. J. Cox and M. Mann, *Nat. Biotechnol.*, 2008, **26**, 1367–1372.
28. P. L. Ross, Y. N. Huang, J. N. Marchese, B. Williamson, K. Parker, S. Hattan, N. Khainovski, S. Pillai, S. Dey, S. Daniels, S. Purkayastha, P. Juhasz, S. Martin, M. Bartlet-Jones, F. He, A. Jacobson and D. J. Pappin, *Mol. Cell. Proteomics*, 2004, **3**, 1154–1169.
29. A. Thompson, J. Schafer, K. Kuhn, S. Kienle, J. Schwarz, G. Schmidt, T. Neumann, R. Johnstone, A. K. Mohammed and C. Hamon, *Anal. Chem.*, 2003, **75**, 1895–1904.
30. G. C. McAlister, E. L. Huttlin, W. Haas, L. Ting, M. P. Jedrychowski, J. C. Rogers, K. Kuhn, I. Pike, R. A. Grothe, J. D. Blethrow and S. P. Gygi, *Anal. Chem.*, 2012, **84**, 7469–7478.
31. D. Linke, C. W. Hung, L. Cassidy and A. Tholey, *J. Proteome Res.*, 2013, **12**, 2755–2763.
32. M. Vaudel, J. M. Burkhart, A. Sickmann, L. Martens and R. P. Zahedi, *Proteomics*, 2011, **11**, 2105–2114.
33. E. Lange, C. Gropl, K. Reinert, O. Kohlbacher and A. Hildebrandt, *Pac. Symp. Biocomput. 2006*, 2006, 243–254.
34. W. R. French, L. J. Zimmerman, B. Schilling, B. W. Gibson, C. A. Miller, R. R. Townsend, S. D. Sherrod, C. R. Goodwin, J. A. McLean and D. L. Tabb, *J. Proteome Res.*, 2015, **14**, 1299–1307.
35. A. Michalski, J. Cox and M. Mann, *J. Proteome Res.*, 2011, **10**, 1785–1793.
36. S. Y. Ow, M. Salim, J. Noirel, C. Evans, I. Rehman and P. C. Wright, *J. Proteome Res.*, 2009, **8**, 5347–5355.
37. J. M. Burkhart, M. Vaudel, R. P. Zahedi, L. Martens and A. Sickmann, *Proteomics*, 2011, **11**, 1125–1134.
38. M. Bantscheff, M. Boesche, D. Eberhard, T. Matthieson, G. Sweetman and B. Kuster, *Mol. Cell. Proteomics*, 2008, **7**, 1702–1713.
39. N. A. Karp, W. Huber, P. G. Sadowski, P. D. Charles, S. V. Hester and K. S. Lilley, *Mol. Cell. Proteomics*, 2010, **9**, 1885–1897.
40. L. Ting, R. Rad, S. P. Gygi and W. Haas, *Nat. Methods*, 2011, **8**, 937–940.
41. C. D. Wenger, M. V. Lee, A. S. Hebert, G. C. McAlister, D. H. Phanstiel, M. S. Westphall and J. J. Coon, *Nat. Methods*, 2011, **8**, 933–935.
42. M. M. Savitski, G. Sweetman, M. Askenazi, J. A. Marto, M. Lang, N. Zinn and M. Bantscheff, *Anal. Chem.*, 2011, **83**, 8959–8967.
43. M. Wuhr, W. Haas, G. C. McAlister, L. Peshkin, R. Rad, M. W. Kirschner and S. P. Gygi, *Anal. Chem.*, 2012, **84**, 9214–9221.
44. S. Y. Ow, M. Salim, J. Noirel, C. Evans and P. C. Wright, *Proteomics*, 2011, **11**, 2341–2346.
45. M. Vaudel, J. M. Burkhart, S. Radau, R. P. Zahedi, L. Martens and A. Sickmann, *J. Proteome Res.*, 2012, **11**, 5072–5080.

46. M. Vaudel, A. Sickmann and L. Martens, *Biochim. Biophys. Acta*, 2014, **1844**, 12–20.
47. M. Vaudel, H. Barsnes, R. Bjerkvig, A. Bikfalvi, F. Selheim, F. S. Berven and T. Daubon, *Curr. Pharm. Biotechnol.*, 2016, **17**, 105–114.
48. A. I. Nesvizhskii and R. Aebersold, *Mol. Cell. Proteomics*, 2005, **4**, 1419–1440.
49. E. Aasebo, J. A. Opsahl, Y. Bjorlykke, K. M. Myhr, A. C. Kroksveen and F. S. Berven, *PLoS One*, 2014, **9**, e90429.
50. T. Kocher, P. Pichler, M. Schutzbier, C. Stingl, A. Kaul, N. Teucher, G. Hasenfuss, J. M. Penninger and K. Mechtler, *J. Proteome Res.*, 2009, **8**, 4743–4752.
51. R. Craig, J. P. Cortens and R. C. Beavis, *J. Proteome Res.*, 2004, **3**, 1234–1242.
52. H. Barsnes, M. Vaudel, N. Colaert, K. Helsens, A. Sickmann, F. S. Berven and L. Martens, *BMC Bioinf.*, 2011, **12**, 70.
53. T. E. Thingholm, G. Palmisano, F. Kjeldsen and M. R. Larsen, *J. Proteome Res.*, 2010, **9**, 4045–4052.
54. T. Geiger, J. Cox, P. Ostasiewicz, J. R. Wisniewski and M. Mann, *Nat. Methods*, 2010, **7**, 383–385.

CHAPTER 9

Informatics Solutions for Selected Reaction Monitoring

BIRGIT SCHILLING[a], BRENDAN MACLEAN[b], JASON M. HELD[c] AND BRADFORD W. GIBSON*[a,d]

[a]Buck Institute for Research on Aging, 8001 Redwood Blvd, Novato, California 94945, USA; [b]Department of Genome Sciences, University of Washington School of Medicine, Foege Building S113, 3720 15th Ave NE, Seattle, Washington 98195, USA; [c]Departments of Medicine and Anesthesiology, Washington University School of Medicine, 660 South Euclid Avenue, St. Louis, Missouri 63110, USA; [d]Department of Pharmaceutical Chemistry, University of California, San Francisco, CA 94143, USA
*E-mail: bgibson@buckinstitute.org

9.1 Introduction

9.1.1 SRM – General Concept and Specific Bioinformatic Challenges

Quantitative mass spectrometry-based proteomics plays an important role in many aspects of biological science from basic research to clinical analysis. Selected reaction monitoring (SRM) is a type of mass spectrometric data acquisition in which specific analytes are targeted for quantification in a triple quadrupole mass spectrometer. In contrast to data-dependent acquisition (DDA) used for unbiased discovery experiments that chose detected

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

ions for fragmentation and analysis, SRM is data-independent and targets specific peptide analytes that are pre-defined prior to data acquisition. SRM acquisition on triple quadrupole instruments utilizes the first quadrupole, Q1, for peptide precursor ion (P) selection, typically using a mass selection window width of 0.7 *m*/*z*. Subsequently, the isolated precursor ion is fragmented in the collision cell, Q2, followed by specific fragment ion (F) selection in the third quadrupole, Q3, also using a mass selection window width of 0.7 *m*/*z*. These precursor–fragment ion pairs are called Q1/Q3 *transitions*, and are monitored either during the entire chromatographic gradient or at pre-determined, scheduled retention time windows (scheduled SRM). Typically, 3–5 transitions are monitored per peptide precursor ion and two or more peptides per protein are monitored for robust quantification of protein expression. Each Q1/Q3 transition is selected for a specific 'dwell time', the length of time each transition is acquired, typically in the range of 10–100 milliseconds. The time needed to monitor all assay transitions at a specific retention time (RT) is referred to as cycle time. The number of selected transitions, the assigned dwell time, whether an acquisition is RT-scheduled or not, related RT window size, and chromatographic resolution all influence the sensitivity and robustness of an SRM assay. Parameter selection typically requires trade-offs, with optimization required to maximize dwell times for maximum sensitivity, shorter cycle times to measure more data points across a chromatographic peak to increase quantitative accuracy, and increase of the number of transitions to maximize multiplexing. Decisions regarding these parameters during assay development and appropriate experimental design are crucial for reliable, highly sensitive, and highly accurate quantification.

Since SRM mass spectrometry was first applied for peptides and protein expression analysis[1] it has become increasingly popular. Today, SRM is considered the gold standard for mass spectrometric quantitation and was declared 'Method of the Year 2012' by Nature Methods.[2] The increased MS scan speed and sensitivity of modern triple quadrupole instrumentation has enabled highly multiplexed SRM studies[3,4] in complex biological matrices, such as plasma and tissue lysates. For this reason, targeted SRM techniques and their associated workflows have gained considerable popularity in biomedical applications, as well as for systems biology and translational medicine (as recently reviewed[5–7]). Recent studies have demonstrated that limits of quantification (LOQs) can be achieved in the ng mL^{-1} to low µg mL^{-1} range while maintaining high assay reproducibility and low coefficients of variation (CVs) $< 20\%$, demonstrating that SRM assays can be applied in verification studies in the context of clinical or biological experiments.[3,4,8–10]

SRM has been adopted for a wide range of experimental goals that have varying requirements for assay accuracy, repeatability, sensitivity and robustness. A prior report by Carr *et al.*[11] outlines three different tiers describing experimental design and assay characteristics for targeted assays, such as SRM assays, and the corresponding requirements for publication. This report also details the appropriate presentation of data quality, performance metrics, and data transparency. Given these developments, it is essential

to choose reliable and multi-functional software capable of addressing the diverse requirements necessary for successful implementation and data analysis of targeted SRM assays.

SRM assays pose several specific bioinformatics challenges including assay development (choosing transitions and collision energies), generating acquisition methods (determining dwell times and retention time scheduling), and data processing (peak integration). These requirements are very different from typical software solutions used for data-dependent acquisitions, such as discovery or shotgun proteomics and will be discussed in more detail. Furthermore, SRM is often coupled to experimental designs using stable isotope dilution SRM mass spectrometry workflows (SID-SRM-MS) that utilize one or more stable isotope versions of the analyte as internal standards. SID-SRM-MS requires the 'light' and 'heavy' forms of the peptide analytes to be analyzed in a uniform way to determine peak areas. Ideal SRM software would also be able to accommodate label-free SRM assays. In addition, the software used for SRM experiments should be able to facilitate assay development and optimization steps, including refining peptide and transition selection, retention time scheduling for high multiplexing and high throughput, collision energy optimization *etc.* At the same time visual software tools should be available to review the acquired data or to potentially manually adjust incorrect peak picking. An SRM software algorithm also needs to provide tools that assess false discovery rates for peak picking and provide confidence metrics that a certain analyte peak group was selected correctly in the chromatograms, for example using a tool matching SRM peak groups back to original MS/MS spectral libraries of the same analyte to identify it. Lastly, it may be advantageous to have a software algorithm that performs SRM system suitability testing 'on the fly', especially when assays are retention-time scheduled, and when RT starts to drift, so that the software can immediately and automatically correct the instrument acquisition method, *e.g.*, RT re-scheduling.

9.1.2 SRM-Specific Bioinformatics Tools

Instrument vendors are one source of SRM software, providing an array of commercial products including PinPoint™ (Thermo), MultiQuant™ (SCIEX), MassHunter™ (Agilent) and MassLynx™/TargetLynx™ (Waters). In addition, other sources have contributed software tools for the design and analysis of large scale SRM proteomic datasets.[12,13] Of these alternative sources, Skyline,[14] an open-source software suite of tools for SRM analysis, has emerged as the most widely used platform (http://proteome.gs.washington.edu/software/skyline) and will be the focus of this chapter. Skyline is a freely available, comprehensive tool with high versatility for SRM assay development and subsequent processing of data acquired on the triple quadrupole mass spectrometers from Agilent, SCIEX, Shimadzu, Thermo and Waters. Skyline can be used for peptide and transition selection, assay

optimization, SRM instrument method export, peak detection and integration, signal processing, and integration with statistical External Tools and algorithms to generate quantitative results for peptides and proteins (Figure 9.1). A number of recent, large-scale multi-laboratory studies have processed their SRM data using Skyline.[3,4,8,15] To highlight some of the Skyline SRM functionalities, we describe features including important visual displays and statistical tools, Skyline's support in accomplishing large multi-laboratory studies involving different sites and instruments, and its use in generating custom results reports for data sharing. We will discuss the integration of various, 'External Tools' into the Skyline user interface for additional data processing that can be downloaded from the Skyline Tool Store (http://skytools.maccosslab.org) and that install automatically onto the Skyline Tools menu.[16] Finally, an easy, point-and-click strategy is presented that supports dissemination of SRM data processed in Skyline to the Panorama web data repositories.[17] Most of the Skyline features discussed in this chapter have existing tutorials, for step-by-step instructions, that are published on the Skyline website along with recorded webinars presenting key Skyline functionalities.

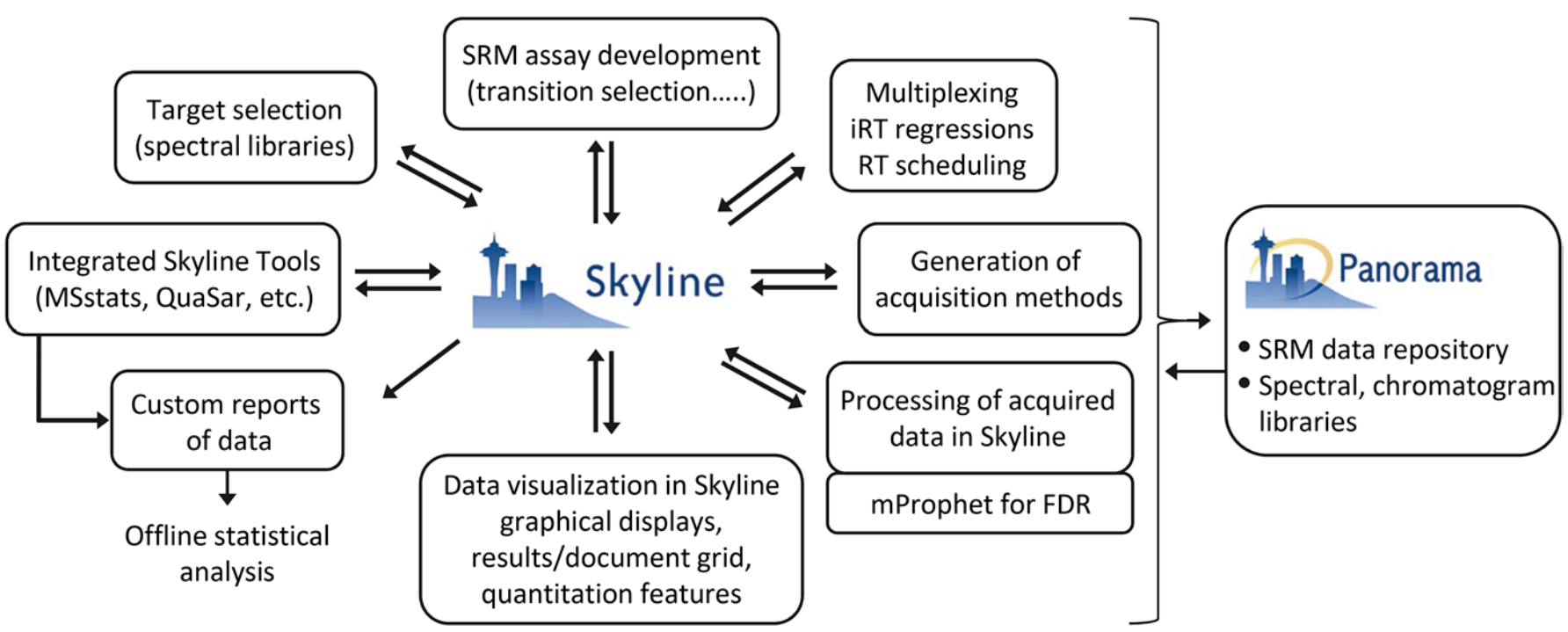

Figure 9.1 Skyline as integral part of a targeted SRM assay and subsequent data analysis. A typical SRM workflow is shown starting with target selection from spectral libraries or other resources, SRM assay development and optimization, and SRM assay multiplexing. Instrument SRM assay acquisition methods are directly exported from Skyline, and acquired data are imported into Skyline for data processing, including the use of the mProphet algorithm to perform false discovery rate (FDR) analysis for correct peak detection. Many graphical and visual features in Skyline allow for fast assessments of data quality and initial data results reports inside Skyline. For further statistical analysis custom reports can be generated and processed post-Skyline analysis, or data can be processed using Skyline's integrated External Tools available under the Skyline tool options, *i.e.*, MSstats, QuaSar. All data processed through Skyline can be directly uploaded from Skyline using a 'point-and-click' feature to Panorama, a web-based data repository containing spectral and chromatogram libraries and quantitative data sets.

9.2 SRM Assay Development

9.2.1 Target and Transition Selection, Proteotypic and Quantotypic Peptides

The Skyline software supports a variety of strategies to select peptide targets and MS/MS fragment ions, also known as a SRM 'transitions'. For SRM quantification, researchers typically select 'proteotypic' peptides as analytes and targets. Proteotypic peptides are defined as peptides that uniquely identify each protein and are consistently observed when a sample mixture is interrogated by a mass spectrometer. In addition, high peptide MS response factor and good ionization is preferable to make an assay as sensitive as possible. While proteotypic peptides can be predicted computationally[18] (for additional details see), selection of optimal peptides for targeted SRM assays is often based on empirical mass spectrometric data. Indeed, the most common approach used is to select target peptides and transitions from previously generated data obtained by data-dependent acquisitions (DDA) and corresponding MS/MS spectral libraries, or from existing SRM chromatogram libraries. MS/MS spectral libraries can be easily built within Skyline from DDA data by importing the corresponding database search engine outputs,[19] and Skyline supports the majority (~20) of common MS/MS search algorithms. Vast amounts of proteomic data are now uploaded to public data repositories providing valuable data resources. MS2 data can be downloaded and added to Skyline as described, or SRM chromatogram libraries from Panorama can be added to Skyline directly *via* an 'Edit Library' form. Essentially any data or library format can be added independent of what type of triple quadrupole instrument is used to generate the SRM assays. Repositories of proteomic data are provided by the ProteomeXchange consortium such as Pride[20] for tandem MS/MS data, and PASSEL–Peptide Atlas[21,22] and SRM Atlas (www.srmatlas.org),[23] as an interface for selection of SRM transitions. In addition, other major resources include MassIVE (http://massive.ucsd.edu), GPMdb,[24] NIST libraries (http://Peptide.NIST.gov), Panorama,[17] and the CPTAC Assay Portal, the latter provides a repository of targeted proteomic assays (http://assays.cancer.gov).[25]

In addition to selecting proteotypic peptides, ideal peptides chosen for analysis are also quantotypic. The latter comprises aspects such as choosing peptides that are stable during workup and assay acquisitions, selecting peptides that were generated with 'complete proteolysis', *e.g.* avoiding peptides with missed cleavages or semi-tryptic peptides, avoiding peptides ending in KK, RR, KR or RR, avoiding N-terminal proline cleavage, *e.g.*, KP, RP. Peptides are also often not selected when they are prone to chemical, artificial modifications, such as methionine oxidation, or deamidation of Asn or Gln residues, which are particularly prone to deamidation when followed by Gly, *e.g.*, NG or QG. Potential N-terminal cyclization of Gln and Glu residues should also be considered. Lastly, peptides containing known PTM motifs, such as the N-X-S/T glycosylation motif are often avoided when performing regular protein quantification.

Once spectral–SRM libraries are in Skyline, and the Skyline target tree is populated with analyte peptides, SRM transitions can be selected automatically. This is typically prioritized based on the library intensity ranking of the transitions. Skyline Transition Settings allow flexibility in defining 'Transition Selection Filters', *e.g.*, choosing the top 5, highest ranked y-ions, or y- and also b-ions based on the library information available per target peptide. The Skyline peptide target tree is subsequently and automatically populated with peptides and transitions as first steps of the SRM assay development and general workflow (see Figure 9.1).

The vendor neutrality of Skyline facilitates the use of public spectral or SRM libraries which are often acquired on instruments across a range of vendors and proprietary data acquisition software. This is also advantageous when performing large multi-laboratory studies where participants often employ different instrument platforms. In addition to data analysis, Skyline also facilitates generation of SRM data acquisition methods. Specifically, Skyline template files pre-populated with target peptides and transitions can be distributed to the different study sites, and, if necessary, the templates and assay transitions can be further refined at the individual study sites depending on SRM instrument platform. Transition lists or native instrument acquisition methods can also be exported from Skyline for the SRM assays, as recently demonstrated by a large-scale, NCI-CPTAC sponsored study measuring cancer-relevant proteins in plasma[8] and a large, international study configuring and validating 645 novel multiplexed MRM assays representing 319 proteins expressed in human breast cancer.[4]

At times, no peptides from a target protein of interest are found in DDA analysis or public spectral and SRM libraries. In these cases, other strategies can be chosen for peptide selection, such as software algorithms that predict proteotypic peptides (PTPs) unique to the target proteins of interest.[18,26,27] Caution is urged, however, in using these tools trained on DDA data, based on recent results in developing the Skyline External Tool 'Peptide Response Predictor' named PREGO, which can also be incorporated directly into Skyline.[28] PREGO is a software tool, trained on targeted and data-independent acquisition (DIA) MS/MS data, that predicts high responding peptides for SRM experiments, followed by convenient population of the Skyline target tree with the predicted assay target peptides. Alternatively, a less frequently used approach referred to as empirical transition refinement is based on theoretical prediction of peptides and transitions,[29] which then can be refined by data acquisition from study samples,[29] or recombinant expressed proteins in water,[30] followed by further processing in Skyline.

9.2.2 Spikes of Isotopically Labeled Peptides and Protein Standards and Additional Assay Development Steps

When SRM assays are conducted in complex matrix backgrounds,[31] such as plasma or urine, targeted SRM assays often apply *s*table *i*sotope *d*ilution strategies, SID-SRM-MS, to improve dynamic range and selectivity.[32]

In these settings, it is essential that the SRM data analysis software tool supports flexible stable isotope peptide setup, both for assay development and data analysis and further processing. An NCI sponsored workshop published fit-for-purpose guidelines addressing design of targeted SRM studies and proposed a three-tier system for different study types.[11] The most stringent quantitative SRM assay requirements are given for clinical or diagnostic assays, which should be designed with (i) 'labeled internal standards' (spiked heavy-labeled peptides for each analyte peptide), and (ii) 'reference standards' (standards very similar to native protein targets, *i.e.*, heavy labeled proteins undergoing all processing steps). Skyline software fully supports these types of SID-SRM-MS assays, and was used in a recent NCI-CPTAC verification group study that successfully monitored 27 cancer-relevant proteins across 11 laboratories and 14 triple quadrupole instruments from 4 different instrument manufacturers.[8] As shown in Figure 9.2, 27 fully (and uniformly) labeled ^{15}N proteins were spiked into the plasma samples at the beginning of sample processing, while the 125 heavy labeled synthetic peptides containing ^{13}C–^{15}N labeled amino residues were spiked in pre- or post-desalting after sample processing and digestion. Skyline analysis using this multiple isotope labeling strategy can be set up in the Peptide Settings by defining 'isotope label types' and 'isotope modifications' and by choosing 'internal standard types'. After the import of raw data, extracted ion chromatograms will show peptide groups of corresponding light and heavy peptide counterparts (Figure 9.2).

Additional assay development steps performed in Skyline include features to test for auto-interferences that ensure that y- and b-ions for any given target peptide do not contain any *m*/*z* overlap in their various isotopic forms. If necessary, collision energies (CE) can be optimized using a Skyline algorithm to further increase Q1/Q3 transition responses and assay sensitivity.[33] A key assay development step for high throughput, highly multiplexed SRM assays typically consists of generating time-schedule acquisition methods, monitoring target peptides only during a time window in which they elute from the HPLC.[10,34] As discussed in the following section, this retention time (RT) scheduling method enables increasing the number of SRM transitions that can be monitored in a single LC-MS run and is now becoming more routine[3,4,8] (Figure 9.2).

9.2.3 Retention Time Regressions and Retention Time Scheduling

The faster scan speeds, improved sensitivity and dynamic range and more flexible instrument acquisition software of modern triple quadrupole instruments has enabled a dramatic increase in the number of targeted SRM analytes that can be multiplexed in a single experiment. For example, retention time scheduling of SRM transitions with defined acquisition time windows around known or predicted analyte retention times has become widely

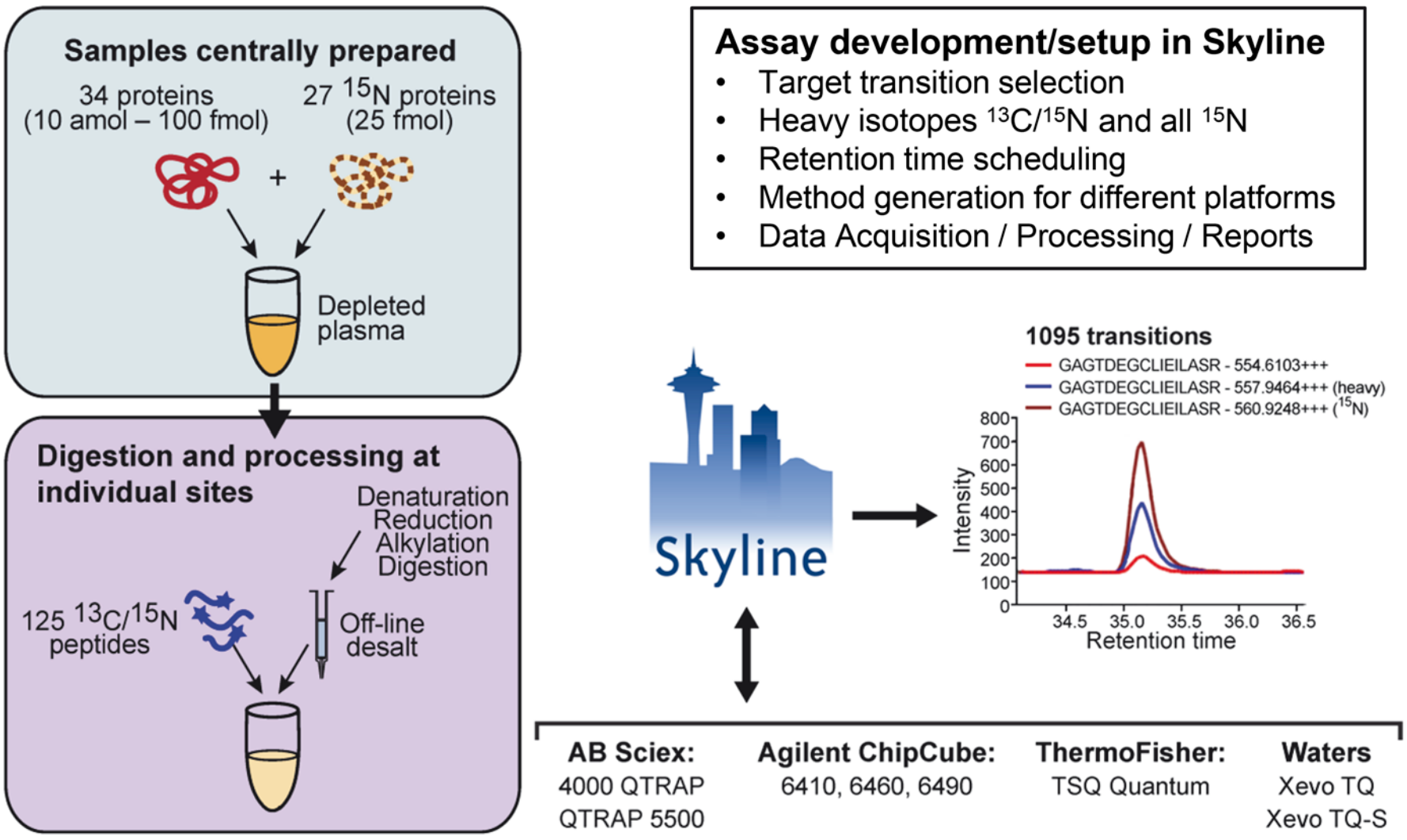

Figure 9.2 Schematic of the experimental design for CPTAC SID-SRM-MS study. The CPTAC Study Phase III introduced unlabeled (light) and uniformly ^{15}N-labeled proteins into the workflow, which were spiked into depleted plasma to generate a nine-point response curve. Samples were further processed at the individual sites to denature, reduce, alkylate, desalt, and reconstitute the samples with ^{13}C and ^{15}N peptide standards for LC-MRM-MS analysis, resulting in a total of 1095 transitions for each method. Skyline was integral from Phase I through Phase III for transition selection, method building, retention time scheduling, and data integration across the different vendor platforms. (This research was originally published in Molecular Cellular Proteomics. Abbatiello, S. E.; Schilling, B. *et al.* Large-Scale Interlaboratory Study to Develop, Analytically Validate and Apply Highly Multiplexed, Quantitative Peptide Assays to Measure Cancer-Relevant Proteins in Plasma. *Molecular Cellular Proteomics*. 2015; 14: 2357–2374. © the American Society for Biochemistry and Molecular Biology).

adopted.[10,34] Skyline has implemented algorithms that can estimate how many concurrent SRM transitions can be grouped into an SRM assay for each scheduling time window. The time windows are constructed to maximize MS/MS accumulation time per scan yet obtain >10 measurement points across an eluting targeted peptide for optimal ion statistics.

To set up scheduled SRM analysis, the retention time of each peptide must be determined. To do this, the user typically imports several non-scheduled SRM acquisitions into the Skyline assay document in which each run monitored a fraction of the analytes during the entire LC-MS run. Subsequently, Skyline generates a combined, single RT scheduled instrument method measuring all transitions in a single acquisition. A well-documented example of such an application was recently demonstrated by Abbatiello *et al.* where precise details were provided in the supplemental SOPs.[8] Alternatively

or in addition, the user can take advantage of Skyline's indexed retention time (iRT) features,[35] that can use previously generated SRM acquisitions, such as those from a different laboratory or when using a different gradient, to re-calibrate for the current chromatographic conditions. The iRT strategy requires a common set of (iRT) peptides, *e.g.*, 10–20 standard peptides, that are analyzed on both systems in a single acquisition, and that are processed using the iRT calculator. Ideally, the iRT peptides elute across the full range of the LC gradient. An 11 peptide mix (available from Biognosys, Pierce and Sigma Aldrich) is commonly used to calibrate chromatographic retention times for reversed phase, C18 columns (Figure 9.3(A)). Once the targeted peptides are analyzed on a given system or gradient they can be added to the iRT regression (Figure 9.3(B)). A different laboratory (or the same laboratory with a changed chromatographic setup) can now use these iRT regressions by first acquiring data just for the 11 iRT standard peptides on their own LC-SRM-MS system. These data are then used to predict scheduled retention times for all peptide analytes and to export a single RT-scheduled acquisition method, initially using a wider (~5 min) scheduled RT window (Figure 9.3(C)). The latter acquisition can be used to refine the iRT regression including all analyte peptides, and based on the newly measured RTs using the new gradient–system, acquisition methods can subsequently be designed with tighter scheduled RT windows, *i.e.*, ~1–2 min or less, depending on the chromatographic conditions.

In summary, the iRT technique allows for previously measured peptide retention times to be stored for reuse across multiple runs, laboratories, instruments and even gradient changes. Only a single calibration run is required to estimate analyte retention times that can greatly simplify the generation of RT scheduled methods for use in higher multiplexed targeted experiments. Recent work by Parker *et al.*[36] demonstrated the use of peptides endogenous to the sample matrix to be used for indexed RT regressions. Specifically, peptide sequences were chosen that are conserved across most eukaryotic species, termed *C*ommon *i*nternal *R*etention *T*ime standards (CiRT). In addition, Skyline has incorporated features that allow fast 'RT *re-scheduling*' in cases where operators may experience a chromatographic shift, such as in assays conducted over several weeks or when introducing a new column.

9.2.4 Method Generation for MS Acquisitions

Downstream of the assay development and optimization, Skyline provides options to generate (multiplexed) methods for SRM-MS acquisitions. The user can either export an SRM transition list in a format appropriate for the selected triple quadrupole instrument platform–vendor, which can be imported by the vendor acquisition software. Alternatively, Skyline can directly generate a native instrument method on the MS instrument computer that can be immediately used. Interfacing of the mass spectrometer

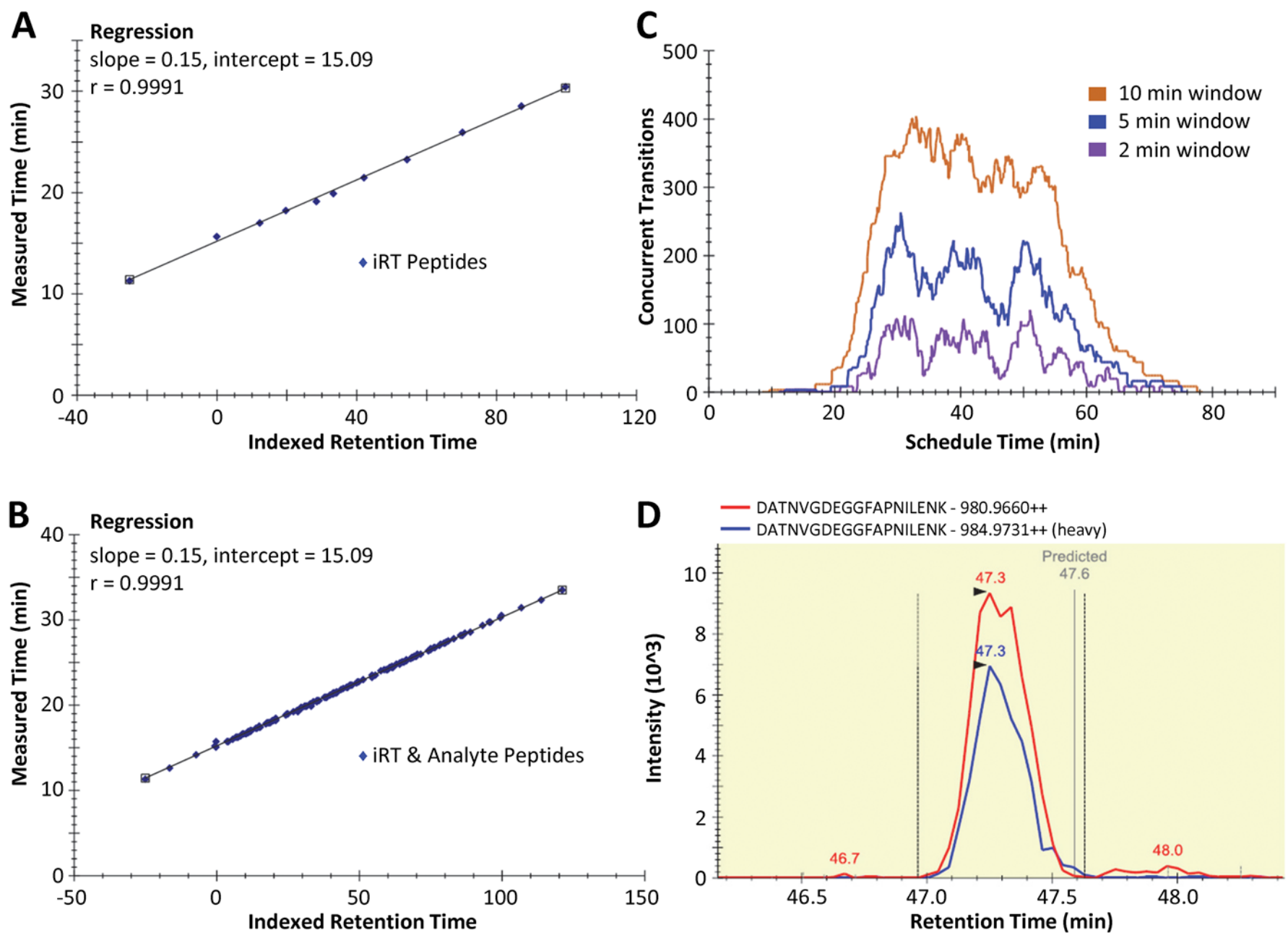

Figure 9.3 Retention time regressions and time-scheduled acquisitions. (A) Retention time regressions are based on an iRT calculator using an eleven peptide mixture (Biognosys) eluting across a given LC-SRM-MS gradient. Measured retention times for peptides are displayed on the *y*-axis while the *x*-axis shows a calibrated indexed retention time (iRT–C18) scale. (B) Peptides measured during SRM assays can be added to the iRT regression itself. Using the iRT calculator, the 147 human peptide analytes have obtained calibrated indexed retention time values (iRT–C18 values, *x*-axis). The iRT regression now contains values from the original 11 iRT standard peptides and the calibrated 147 SRM peptide analytes. (C) iRT regressions enable quick adjustment of the SRM assay to new gradients or a different chromatographic system, and only a simple acquisition of just the 11 iRT standard peptides using the new gradient or system is necessary. The iRT calculator will predict RTs for all peptide analytes/SRM transitions present in the Skyline document that can subsequently employ RT scheduling using an appropriate scheduling window according to the estimated concurrent transitions as shown in (C). Here the 5 min scheduling window is initially chosen. (D) Example for a light and heavy peptide pair showing the measured elution profile and the predicted RT.

acquisition software directly with Skyline can be very convenient, particularly for high throughput studies. A related algorithm, SkylineRunner.exe, is a command line interface that automates many processes such as import of acquired SRM data into Skyline and peak processing steps. Assay workflows can be highly automated using these Skyline capabilities including the final result reports.

9.3 System Suitability Assessments

One important component of any quantitative SRM study for the instrument operator is to ensure system suitability for both the chromatographic and mass spectrometric systems. A system suitability study by Abbatiello *et al.* has established a set of performance metrics for multi-site studies, including the use of Skyline to monitor data quality during an ongoing SRM study.[37] Graphical displays in the Skyline environment allow the user to quickly assess performance visually and be alerted for any deviation or metrics variance. Some examples of suboptimal SRM performance are shown in Figure 9.4, in which a system suitability standard sample was acquired before, and during the SRM target assays.[8,37] Common performance issues include the loss of signal intensity for late eluting peptides, typically a problem with the LC system, and increased peak area coefficient of variation (CV), which indicates decreased repeatability. Figure 9.4 demonstrates how system suitability can uncover performance issues, and system suitability metrics plots are shown before (top) and after (bottom) troubleshooting and repairing an HPLC problem.

Bereman *et al.* have further refined system suitability monitoring through process controls with their 'External Tool' called Statistical Process Control in Proteomics, SProCoP.[38] This program can be downloaded from the Skyline tool store for installation directly into Skyline and used to monitor system performance. Additional system performance monitoring has also been developed by the Skyline team in collaboration with LabKey Software, that automates assessment of data quality and system performance using Skyline and the web interface Panorama (unpublished results – http://skyline.gs.washington.edu/labkey/webinar11.url).

9.4 Post-Acquisition Processing and Data Analysis

9.4.1 mProphet False Discovery Analysis, Peak Detection and Peak Picking

Subsequent to data acquisition, raw mass spectrometric files can be directly imported into the same Skyline document that was used for SRM assay development and previous method exports. Skyline has continually improved SRM peak detection and peak picking algorithms which are crucial for reliable quantification of peptides. Algorithms are available that attempt to accurately choose peaks for peptide SRM data and that assign statistical metrics for the confidence that assigned peaks differ significantly from those produced by random chance. One such algorithm that has been recently implemented in Skyline, mProphet,[39] evaluates peaks based on training a linear combination of scores related to co-elution, peak shape, ion intensity, spectral library relative product ion abundance correlation, predicted retention time, isotopic standards, and several other factors. Confident and advanced peak picking is especially important when handling very large and highly multiplexed data

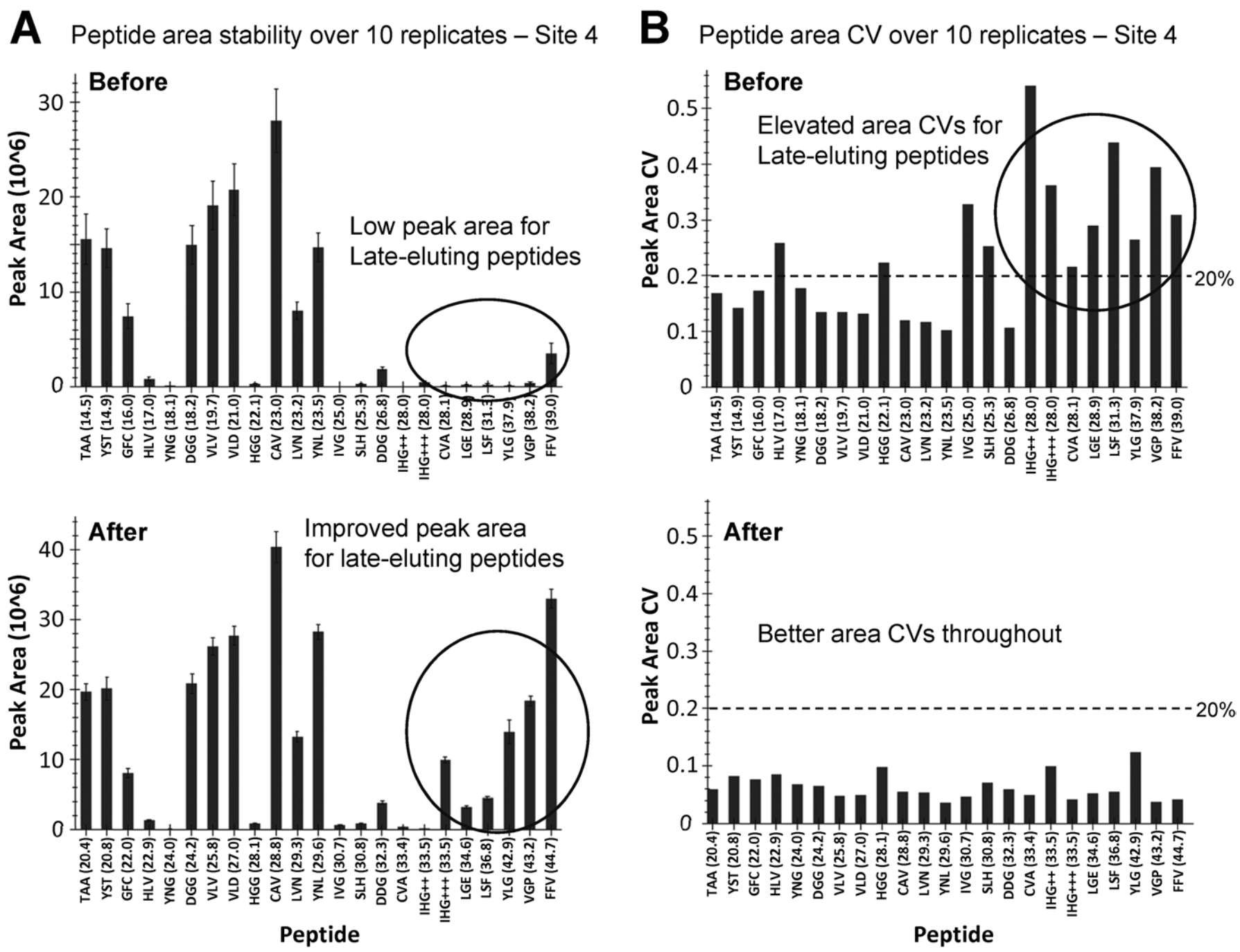

Figure 9.4 Before and After Plots of System Suitability Data from Individual Sites with System Performance Problems. (A) Before and after view of the Peak Area view in Skyline for Site 4. The last 5 peptides of the system suitability sample showed very low signal as compared with the other sites involved in the study. When Site 4 re-calibrated the flow meter on their organic solvent pump, the peak areas for the later eluting peptides increased and were observed to be more similar to the other sites. (B) The same before and after data set from Site 4 but using the Peak Area CV view in Skyline. The later eluting peptides all have elevated CV values for Peak Area in the "before" case. When the system was fixed, all CVs improved, but the effect was largest on the later eluting peptides. (This research was originally published in Molecular Cellular Proteomics. Abbatiello, S. E.; Mani, D. R.; Schilling, B. *et al.* Design, implementation and multisite evaluation of a system suitability protocol for the quantitative assessment of instrument performance in liquid chromatography–multiple reaction monitoring-MS (LC-MRM-MS). *Molecular Cellular Proteomics*. 2013; 12: 2623–2639. © the American Society for Biochemistry and Molecular Biology.)

sets where visual inspection of all target peptides is challenging. Figure 9.5(A) shows how Skyline uses decoy peptides with reversed peptide sequences to assess the false discovery rate. Alternatively, training of the model can also be performed using 'second best peaks'. As shown in Figure 9.5(B), individual feature scores are used with the given weight and contribution assigned for the specific model, and subsequently composite scores are calculated similar to the mProphet reports as described by Reiter *et al.*[39] Finally, each target

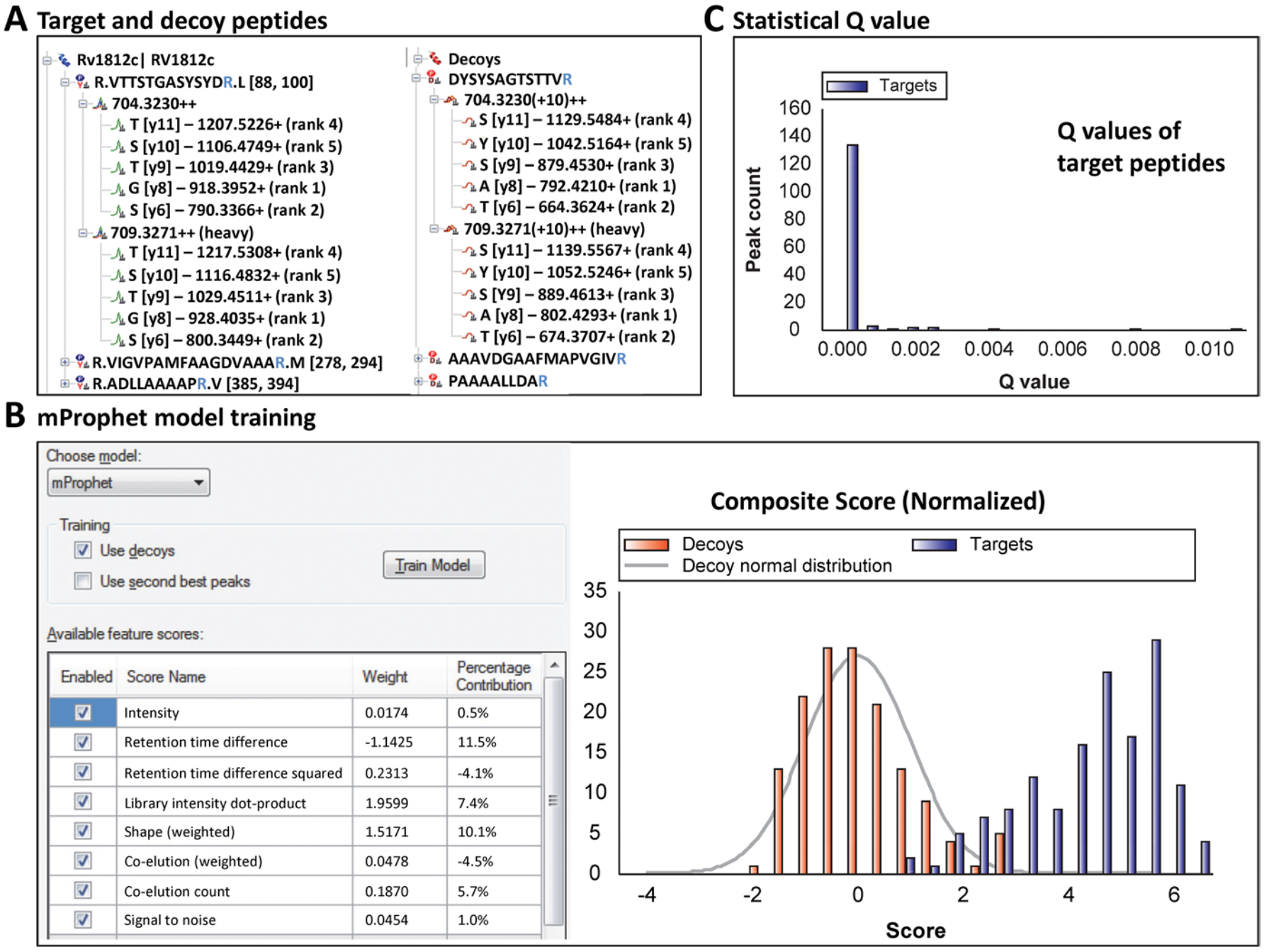

Figure 9.5 mProphet feature in Skyline for SRM peak detection and target FDR analysis. (A) Skyline target tree showing target and decoy peptides. Decoy peptides show reversed sequences, however, the K or R residue is kept constant at the C-terminus. All precursor *m*/*z* values shift by '+10 *m*/*z*', while fragment ions are held constant in position (*e.g.*, y11 remains y11) but usually changes in product *m*/*z* according to the reversed decoy sequence. (B) Training the mProphet Peak Scoring model for the SRM data set using decoy peptides. (C) Skyline mProphet model for correct peak detection and peak integration showing statistical *q* value distribution for target peptides.

peptide is assigned a statistical *p* value that reflects the confidence in peak detection per each replicate acquisition. These *p* values are then adjusted to *q* values reflecting expected false-discovery rates (Figure 9.3(C)). Researchers can subsequently apply a *q* value cutoff threshold, to only provide quantitative data for confidently measured target peptides, and thus improve study results and conclusions.

The Skyline target tree provides access to several additional parameters that are displayed for easy assessment of data processed in Skyline. For each peptide target within each replicate acquisition, the dot product (dotp) compares the extracted SRM signal and transition distribution to the original MS/MS spectral library; transition ranking is displayed, as well as light to heavy ratios per transition when stable isotope standards are present. Overall, data imported into Skyline is automatically processed with statistically solid peak detection algorithms, and extracted ion chromatograms are visualized

in informative displays. However, Skyline also allows the user to manually adjust peak picking if necessary.

9.4.2 Data Viewing and Data Management: Custom Annotation, Results and Document Grids, Group Comparisons

Handling and organizing big data sets can be challenging. This is especially true for studies with multiple laboratories from different Institutes, or when multiple instruments across different platforms and vendors are used. One Skyline feature, referred to as 'Custom Annotation', has facilitated downstream statistical analysis of large data sets. Custom annotations allow instrument operators to annotate and document observations. Common annotations include indicating interferences for transitions, weak signal, peptide signal that was cut off or drifted out of time-scheduled acquisition windows, *etc.* The controlled vocabulary that is provided in Skyline can be tailored for each study, allowing statisticians to account for irregularities and to potentially exclude certain data points from being used. In addition, uniform data reports can be exported across all study sites in the same format to expedite further post-Skyline statistical data processing.[8] Custom annotations are also possible on the transition level, peptide level or data replicate level, and can be included into all custom data reports.

During data processing, users can display Skyline Results Grids that list target peptide peak areas and many other measured parameters in real time. Recently, additional fields have been introduced into the Skyline Document Settings that allow users to enter detailed sample information, *i.e.*, 'disease' or 'control' type samples, sample annotation for 'condition A' or 'condition B', *etc.* These annotations can also be displayed, sorted and filtered 'on the fly' in Skyline's Document Grid during Skyline data processing and visualizations. A new Skyline feature called Group Comparisons enables the user to define groups of different sample subsets; *i.e.*, a control group and a group compared against the control. The Skyline Document Grid can then display calculated fold-change and statistical inference comparing the two sample subsets. Sample annotation can be used to link sample origin and biological information, a feature that is especially critical to organizing data during large-scale studies and in facilitating downstream processing with additional statistical tools.

9.4.3 Data Reports, LOD–LOQ Calculations and Statistical Processing, Use of Skyline External Tools

As outlined in Figure 9.1, Skyline can generate comprehensive custom reports of processed data files. Relevant data fields, such as sample replicate information, transition m/z values, peak areas, annotations and a multitude of other parameters can be selected for data export using specific

report templates tailored for each dataset. These defined data reports can be used for post-Skyline statistical data processing, such as '*au*tomated *d*etection of *i*naccurate and imprecise *t*ransitions' (AuDIT) that identifies potentially inaccurate SRM transition data based on the presence of interfering signal or inconsistent recovery among replicate samples.[40] QuaSar (http://genepattern.broadinstitute.org) is another algorithm that can be used to plot peptide response–calibration curves and to calculate limits of detection (LOD) and limits of quantitation (LOQ) for SID-SRM-MS data sets.[8,41]

To make interfacing with external statistical programs even more convenient for users, Skyline provides a uniform interface to format so-called External Tools for installation directly into Skyline.[16] Tool developers can widely share their tools with proteomics researchers using Skyline and users obtain point-and-click access directly from the Skyline Tools menu for additional statistical analysis. The Skyline Tool Store (http://skytools.maccosslab.org) contains several programs applicable for downstream SRM statistical data processing, including QuaSar, and MSstats,[42] but also other external tools helpful during assay development (PREGO,[28] BiodiversityPlugin) or for quality control (SProCoP[38]).

9.4.4 Group Comparisons and Peptide & Protein Quantification

As mentioned, the Skyline environment allows for some initial quantitative group comparisons. For example, after annotation and grouping of sample types *via* the document grid, one can obtain (limited) statistical ratios for a rapid assessment of changes between conditions. Subsequently, a more in-depth statistical analysis of SRM data can be performed using the Skyline External Tool MSstats 2.0[42] for peptide and protein quantification, and to ultimately detect differentially abundant proteins or peptides. MSstats also includes user options to automatically filter out interferences or 'poor quality features'. Details how to use MSstats in conjunction with Skyline documents are provided in tutorials and webinars on the Skyline website.

New features in Skyline have addressed common requests for protein/peptide quantitation using SRM assays, specifically for SID-SRM-MS assays. In the Skyline Peptide Settings under the Quantification tab, certain parameters will be defined including regression curve fit and regression weighting that will be applied to the new calibration curve calculations. In the Document Grid acquisition, replicate samples can be defined as 'Standard' (samples that are points of the calibration curve), 'Blank', 'Quality Control' (typically some light or heavy peptide mixtures at defined concentrations), and 'Unknown' representing unknown study samples that are investigated, and for which concentrations will be calculated as part of this new peptide quantitation process. The user can also enter 'multiplication factors' that reflect dilution steps performed for the unknown samples during processing, so that in the end concentrations for the unknown samples can be provided as measured 'on-column' concentrations, as well as concentrations reflecting the original sample. Skyline first calculates, and now can also graphically

display, the calibration curves. For example, for the mentioned NCI CPTAC plasma study (Phase II)[8] calibration curves were acquired in 4 replicates for 125 peptides ranging from 1 Amol μL^{-1} to 100 fmol μL^{-1}, and blinded samples, or unknown samples, were also acquired. Peptide concentrations for the different blinded samples were spiked at 72.0, 19.4, 1.75, and 0.105 fmol μL^{-1}, respectively; however those levels were unknown to the user. Skyline generates calibration curves for each peptide analyte according to the spiked calibration curve standards, generates slope, intercepts, and R^2-values for the weighted linear regression lines for each peptide target; and subsequently can determine concentrations of the unknown samples. Unknown samples are marked with 'x' in the calibration curve graphics indicating concentration measurements for the blinded samples (http://skyline.gs.washington.edu/labkey/webinar13.url). Curve graphics displayed in Skyline are interactive and clicking on any concentration point (calibration curve standard sample or unknown sample) will show the extracted chromatograms or peak area views in the typical Skyline visualizations. Skyline quantification features and calculations are implemented currently for SID SRM-MS assays, and will still be further refined for more complicated normalization strategies and more variable heavy peptide spike level matrices.

9.4.5 Easy Data Sharing and SRM Resources – Panorama

Skyline algorithms are closely interfaced with Panorama,[17] a freely-available, open-source repository server application for targeted proteomics assays that integrates into the Skyline proteomics workflow. Security settings are implemented that allow for flexibility in data sharing pre- and post-publication of proteomic and SRM studies. Figure 9.6(A) demonstrates the simple one-click feature that enables the upload of Skyline documents, *i.e.*, SRM studies directly into a user project folder on Panorama, which then populates a web-interface, as shown here for a system suitability example (Figure 9.6(B–D)), with assay peptide and transition information, extracted chromatogram displays, peak area views and more. More details of these features are described by Sharma *et al.*,[17] as well as in tutorials on the Panorama website (www.panoramaweb.org).

Panorama provides interesting applications, not only as repository and resource for other laboratories but it also allows for interactive, large scale multi-site data management. Users can view data on Panorama through the web interface or also download the underlying Skyline files and spectral/chromatogram libraries.

9.5 Post-Translational Modifications and Protein Isoforms or Proteoforms

Processing SRM data for peptides containing post-translational modifications or for peptides derived from protein isoforms or proteoforms pose their own challenges for a bioinformatics algorithm. While challenging, PTM-related signaling pathways and PTM crosstalk, are often particularly interesting

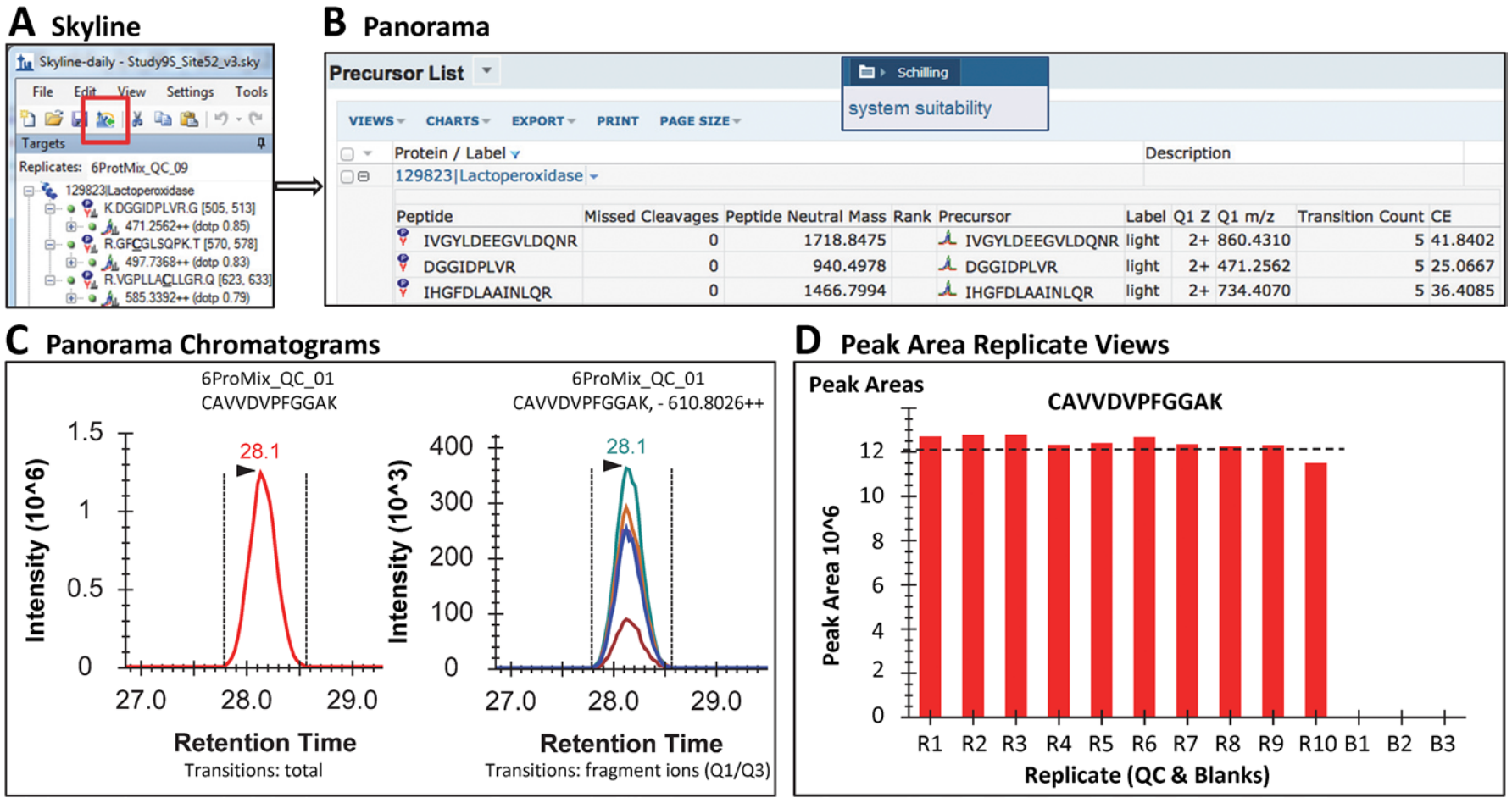

Figure 9.6 Panorama – repository software for targeted proteomics assays from Skyline. (A) Targeted SRM data processed in Skyline was uploaded to Panorama using a 'one-click' feature in the Skyline menu bar. (B) The web-based Panorama interface allows for easy navigation of SRM data results within a given project, here from a system suitability study.[37] (C) Chromatogram views of all uploaded replicate acquisitions are available, as well as (D) peak area replicate views (ten QC replicate acquisitions R1–R10 and three blank acquisitions B1-–B3). The dotted line indicates the average of the peak area across the 10 QC replicates. Views in Panorama are very similar to those in Skyline.

and relevant in diseases. Several studies have reported targeted SRM work for a variety of post-translational modifications (PTMs), for example, the use of SRM for phosphorylation,[43–47] for monitoring glycosylation sites[34] or even providing an *N*-glycoprotein SRM Atlas which can serve as a resource of mass spectrometric assays for *N*-glycosites and multiplexed protein quantification for clinical applications.[48] In addition, SRM has also been used for acylation analysis,[49] or for PTMs on histone, which can become particularly complicated due to the many different modifications and sites possible.[50]

As mentioned, challenges for PTM SRM work for a software algorithm are different than for general protein quantification where one can choose from many proteolytic peptides and find the best proteotypic and quantotypic peptides. For PTM analysis one is typically limited to a specific peptide that contains the PTM site. The scenario becomes even more complicated when a monitored proteolytic peptide contains more than one PTM site, which often occurs for phosphorylation but also for acylation and other modifications. The flexibility of Skyline, however, allows one to overcome most of the bioinformatics difficulties that some other software packages encounter when designing PTM SRM assays and processing acquired PTM data files. For example, Skyline enables simulation of neutral loss for phosphopeptides, loss of H_3PO_4, that often occurs when fragmenting a phosphopeptide.[51] Another challenge is just to simulate different variations of PTM

containing peptides, particularly when they contain more than one PTM site. However, Skyline can import any peptide sequence with any modification at one of multiple sites either from a peptide list or directly from the DDA search engine results and generated spectral libraries. PTM's can be predefined in Skyline under peptide settings, modifications, structural modifications, and Skyline either offers UniMod naming nomenclature or also accepts custom naming from the user. As part of the peptide target tree in Skyline the 'Edit Modification Table' allows for very easy adjustments to the modifications in Skyline. Once set up in Skyline, SRM assays for PTM peptides are handled the same as for any other peptides. It is interesting to mention that particularly for PTM analysis some recent reports have emerged by Jaffe *et al.*[52,53] that take advantage of a targeted assay, referred to as parallel reaction monitoring (PRM) and subsequent processing in Skyline. PRM is similar to SRM, however PRM acquires full scan MS/MS data for the targeted precursor ions and it is typically acquired on high resolution mass spectrometers, allowing for higher multiplexing and monitoring of all fragment ions per peptide.

One additional challenge in quantitative targeted mass spectrometry is to assess different proteoforms or protein isoforms. A large variety of proteoforms can be generated resulting in post-translationally modified proteins as discussed, but also protein isoforms with highly similar protein sequences, proteins from alternatively spliced RNA transcripts (splice variants) or proteins from DNA that featured single nucleotide polymorphisms (SNPs). The application of SRM assays regarding some of these aspects were recently reviewed.[54] For example Costenoble *et al.* developed SRM assays to monitor abundance differences of more than 200 proteins, including a family of isoenzymes with highly similar amino acid sequences as part of a study in *S. cerevisiae* investigating central carbon metabolism.[55] Peptides had to be selected for the SRM assays that are unique to the specific protein isoform. While challenging, SRM assays can begin to assess and quantify proteoforms as shown in a recent study that used SRM to quantify multiple proteoforms of a single protein allowing the quantification of allelic expression of a particular sequence polymorphism.[56] In addition, a study by Vegvari *et al.* identified a novel proteoform of prostate specific antigen (SNP-L132I) in clinical samples by multiple reaction monitoring.[57] In any proteoform SRM project Skyline can assist the operator particularly when generating the SRM assay. Skyline features such as 'remove duplicate peptides' will ensure that only peptides are used that are specific for a particular protein isoform to be used for final quantification.

9.6 Conclusion and Future Outlook

For targeted SRM assays, Skyline provides a complete software solution or toolbox that enables a complete workflow to be set up, acquired and processed. Skyline allows relatively easy interfacing with other resources, such as spectral or chromatogram libraries as well as a full suite of additional statistical External Tools. SRM assays using Skyline can connect to other

workflows, such as DIA/SWATH that can also be processed in Skyline. Skyline's vendor neutral algorithm allows for easy study logistics and for uniform data formats, thus providing a major advantage for Skyline users when performing multi-laboratory studies. Skyline software fully complies with journal requirements for SRM targeted MS assay workflows and quantification, allowing for rapid deposition of properly formatted data sets for publications.

Acknowledgements

We acknowledge the support from the NIH grant RO1-GM103551 (to Michael MacCoss).

References

1. D. M. Desiderio and M. Kai, *Biomed. Mass Spectrom.*, 1983, **10**, 471–479.
2. Editorial, *Nat. Methods*, 2013, **10**, 1.
3. M. W. Burgess, H. Keshishian, D. R. Mani, M. A. Gillette and S. A. Carr, *Mol. Cell. Proteomics*, 2014, **13**, 1137–1149.
4. J. J. Kennedy, S. E. Abbatiello, K. Kim, P. Yan, J. R. Whiteaker, C. Lin, J. S. Kim, Y. Zhang, X. Wang, R. G. Ivey, L. Zhao, H. Min, Y. Lee, M. H. Yu, E. G. Yang, C. Lee, P. Wang, H. Rodriguez, Y. Kim, S. A. Carr and A. G. Paulovich, *Nat. Methods*, 2014, **11**, 149–155.
5. H. A. Ebhardt, E. Sabido, R. Huttenhain, B. Collins and R. Aebersold, *Proteomics*, 2012, **12**, 1185–1193.
6. M. A. Gillette and S. A. Carr, *Nat. Methods*, 2013, **10**, 28–34.
7. P. Picotti and R. Aebersold, *Nat. Methods*, 2012, **9**, 555–566.
8. S. E. Abbatiello, B. Schilling, D. R. Mani, L. J. Zimmerman, S. C. Hall, B. MacLean, M. Albertolle, S. Allen, M. Burgess, M. P. Cusack, M. Gosh, V. Hedrick, J. M. Held, H. D. Inerowicz, A. Jackson, H. Keshishian, C. R. Kinsinger, J. Lyssand, L. Makowski, M. Mesri, H. Rodriguez, P. Rudnick, P. Sadowski, N. Sedransk, K. Shaddox, S. J. Skates, E. Kuhn, D. Smith, J. R. Whiteaker, C. Whitwell, S. Zhang, C. H. Borchers, S. J. Fisher, B. W. Gibson, D. C. Liebler, M. J. MacCoss, T. A. Neubert, A. G. Paulovich, F. E. Regnier, P. Tempst and S. A. Carr, *Mol. Cell. Proteomics*, 2015, **14**, 2357–2374.
9. T. A. Addona, S. E. Abbatiello, B. Schilling, S. J. Skates, D. R. Mani, D. M. Bunk, C. H. Spiegelman, L. J. Zimmerman, A. J. Ham, H. Keshishian, S. C. Hall, S. Allen, R. K. Blackman, C. H. Borchers, C. Buck, H. L. Cardasis, M. P. Cusack, N. G. Dodder, B. W. Gibson, J. M. Held, T. Hiltke, A. Jackson, E. B. Johansen, C. R. Kinsinger, J. Li, M. Mesri, T. A. Neubert, R. K. Niles, T. C. Pulsipher, D. Ransohoff, H. Rodriguez, P. A. Rudnick, D. Smith, D. L. Tabb, T. J. Tegeler, A. M. Variyath, L. J. Vega-Montoto, A. Wahlander, S. Waldemarson, M. Wang, J. R. Whiteaker, L. Zhao, N. L. Anderson, S. J. Fisher, D. C. Liebler, A. G. Paulovich, F. E. Regnier, P. Tempst and S. A. Carr, *Nat. Biotechnol.*, 2009, **27**, 633–641.

10. R. Huttenhain, M. Soste, N. Selevsek, H. Rost, A. Sethi, C. Carapito, T. Farrah, E. W. Deutsch, U. Kusebauch, R. L. Moritz, E. Nimeus-Malmstrom, O. Rinner and R. Aebersold, *Sci. Transl. Med.*, 2012, **4**, 142ra194.
11. S. A. Carr, S. E. Abbatiello, B. L. Ackermann, C. Borchers, B. Domon, E. W. Deutsch, R. P. Grant, A. N. Hoofnagle, R. Huttenhain, J. M. Koomen, D. C. Liebler, T. Liu, B. MacLean, D. R. Mani, E. Mansfield, H. Neubert, A. G. Paulovich, L. Reiter, O. Vitek, R. Aebersold, L. Anderson, R. Bethem, J. Blonder, E. Boja, J. Botelho, M. Boyne, R. A. Bradshaw, A. L. Burlingame, D. Chan, H. Keshishian, E. Kuhn, C. Kinsinger, J. S. Lee, S. W. Lee, R. Moritz, J. Oses-Prieto, N. Rifai, J. Ritchie, H. Rodriguez, P. R. Srinivas, R. R. Townsend, J. Van Eyk, G. Whiteley, A. Wiita and S. Weintraub, *Mol. Cell. Proteomics*, 2014, **13**, 907–917.
12. M. Y. Brusniak, C. S. Chu, U. Kusebauch, M. J. Sartain, J. D. Watts and R. L. Moritz, *Proteomics*, 2012, **12**, 1176–1184.
13. C. M. Colangelo, L. Chung, C. Bruce and K. H. Cheung, *Methods*, 2013, **61**, 287–298.
14. B. MacLean, D. M. Tomazela, N. Shulman, M. Chambers, G. L. Finney, B. Frewen, R. Kern, D. L. Tabb, D. C. Liebler and M. J. MacCoss, *Bioinformatics*, 2010, **26**, 966–968.
15. S. Surinova, M. Choi, S. Tao, P. J. Schuffler, C. Y. Chang, T. Clough, K. Vyslouzil, M. Khoylou, J. Srovnal, Y. Liu, M. Matondo, R. Huttenhain, H. Weisser, J. M. Buhmann, M. Hajduch, H. Brenner, O. Vitek and R. Aebersold, *EMBO Mol. Med.*, 2015, **7**, 1166–1178.
16. D. Broudy, T. Killeen, M. Choi, N. Shulman, D. R. Mani, S. E. Abbatiello, D. Mani, R. Ahmad, A. K. Sahu, B. Schilling, K. Tamura, Y. Boss, V. Sharma, B. W. Gibson, S. A. Carr, O. Vitek, M. J. MacCoss and B. MacLean, *Bioinformatics*, 2014, **30**, 2521–2523.
17. V. Sharma, J. Eckels, G. K. Taylor, N. J. Shulman, A. B. Stergachis, S. A. Joyner, P. Yan, J. R. Whiteaker, G. N. Halusa, B. Schilling, B. W. Gibson, C. M. Colangelo, A. G. Paulovich, S. A. Carr, J. D. Jaffe, M. J. MacCoss and B. MacLean, *J. Proteome Res.*, 2014, **13**, 4205–4210.
18. P. Mallick, M. Schirle, S. S. Chen, M. R. Flory, H. Lee, D. Martin, J. Ranish, B. Raught, R. Schmitt, T. Werner, B. Kuster and R. Aebersold, *Nat. Biotechnol.*, 2007, **25**, 125–131.
19. B. Frewen and M. J. MacCoss, *Curr. Protoc. Bioinformatics*, 2007, chap 13, unit 13 17.
20. L. Martens, H. Hermjakob, P. Jones, M. Adamski, C. Taylor, D. States, K. Gevaert, J. Vandekerckhove and R. Apweiler, *Proteomics*, 2005, **5**, 3537–3545.
21. E. W. Deutsch, J. K. Eng, H. Zhang, N. L. King, A. I. Nesvizhskii, B. Lin, H. Lee, E. C. Yi, R. Ossola and R. Aebersold, *Proteomics*, 2005, **5**, 3497–3500.
22. E. W. Deutsch, H. Lam and R. Aebersold, *EMBO Rep.*, 2008, **9**, 429–434.
23. F. Desiere, E. W. Deutsch, N. L. King, A. I. Nesvizhskii, P. Mallick, J. Eng, S. Chen, J. Eddes, S. N. Loevenich and R. Aebersold, *Nucleic Acids Res.*, 2006, **34**, D655–D658.
24. R. Craig, J. P. Cortens and R. C. Beavis, *J. Proteome Res.*, 2004, **3**, 1234–1242.

25. J. R. Whiteaker, G. N. Halusa, A. N. Hoofnagle, V. Sharma, B. MacLean, P. Yan, J. A. Wrobel, J. Kennedy, D. R. Mani, L. J. Zimmerman, M. R. Meyer, M. Mesri, H. Rodriguez, C. Clinical Proteomic Tumor Analysis and A. G. Paulovich, *Nat. Methods*, 2014, **11**, 703–704.
26. V. A. Fusaro, D. R. Mani, J. P. Mesirov and S. A. Carr, *Nat. Biotechnol.*, 2009, **27**, 190–198.
27. B. Kuster, M. Schirle, P. Mallick and R. Aebersold, *Nat. Rev. Mol. Cell Biol.*, 2005, **6**, 577–583.
28. B. C. Searle, J. D. Egertson, J. G. Bollinger, A. B. Stergachis and M. J. MacCoss, *Mol. Cell. Proteomics*, 2015, **14**, 2331–2340.
29. M. S. Bereman, B. MacLean, D. M. Tomazela, D. C. Liebler and M. J. MacCoss, *Proteomics*, 2012, **12**, 1134–1141.
30. A. B. Stergachis, B. MacLean, K. Lee, J. A. Stamatoyannopoulos and M. J. MacCoss, *Nat. Methods*, 2011, **8**, 1041–1043.
31. H. A. Ebhardt, A. Root, C. Sander and R. Aebersold, *Proteomics*, 2015, **15**, 3193–3208.
32. H. Keshishian, T. Addona, M. Burgess, D. R. Mani, X. Shi, E. Kuhn, M. S. Sabatine, R. E. Gerszten and S. A. Carr, *Mol. Cell. Proteomics*, 2009, **8**, 2339–2349.
33. B. MacLean, D. M. Tomazela, S. E. Abbatiello, S. Zhang, J. R. Whiteaker, A. G. Paulovich, S. A. Carr and M. J. MacCoss, *Anal. Chem.*, 2010, **82**, 10116–10124.
34. J. Stahl-Zeng, V. Lange, R. Ossola, K. Eckhardt, W. Krek, R. Aebersold and B. Domon, *Mol. Cell. Proteomics*, 2007, **6**, 1809–1817.
35. C. Escher, L. Reiter, B. MacLean, R. Ossola, F. Herzog, J. Chilton, M. J. MacCoss and O. Rinner, *Proteomics*, 2012, **12**, 1111–1121.
36. S. J. Parker, H. Rost, G. Rosenberger, B. C. Collins, L. Malmstrom, D. Amodei, V. Venkatraman, K. Raedschelders, J. E. Van Eyk and R. Aebersold, *Mol. Cell. Proteomics*, 2015, **14**, 2800–2813.
37. S. E. Abbatiello, D. R. Mani, B. Schilling, B. MacLean, L. J. Zimmerman, X. Feng, M. P. Cusack, N. Sedransk, S. C. Hall, T. Addona, S. Allen, N. G. Dodder, M. Ghosh, J. M. Held, V. Hedrick, H. D. Inerowicz, A. Jackson, H. Keshishian, J. W. Kim, J. S. Lyssand, C. P. Riley, P. Rudnick, P. Sadowski, K. Shaddox, D. Smith, D. Tomazela, A. Wahlander, S. Waldemarson, C. A. Whitwell, J. You, S. Zhang, C. R. Kinsinger, M. Mesri, H. Rodriguez, C. H. Borchers, C. Buck, S. J. Fisher, B. W. Gibson, D. Liebler, M. MacCoss, T. A. Neubert, A. Paulovich, F. Regnier, S. J. Skates, P. Tempst, M. Wang and S. A. Carr, *Mol. Cell. Proteomics*, 2013, **12**, 2623–2639.
38. M. S. Bereman, R. Johnson, J. Bollinger, Y. Boss, N. Shulman, B. MacLean, A. N. Hoofnagle and M. J. MacCoss, *J. Am. Soc. Mass Spectrom.*, 2014, **25**, 581–587.
39. L. Reiter, O. Rinner, P. Picotti, R. Huttenhain, M. Beck, M. Y. Brusniak, M. O. Hengartner and R. Aebersold, *Nat. Methods*, 2011, **8**, 430–435.
40. S. E. Abbatiello, D. R. Mani, H. Keshishian and S. A. Carr, *Clin. Chem.*, 2010, **56**, 291–305.
41. D. R. Mani, S. E. Abbatiello and S. A. Carr, *BMC Bioinf.*, 2012, **13**(suppl. 16), S9.

42. M. Choi, C. Y. Chang, T. Clough, D. Broudy, T. Killeen, B. MacLean and O. Vitek, *Bioinformatics*, 2014, **30**, 2524–2526.
43. D. M. Cox, F. Zhong, M. Du, E. Duchoslav, T. Sakuma and J. C. McDermott, *J. Biomol. Tech.*, 2005, **16**, 83–90.
44. A. Wolf-Yadlin, S. Hautaniemi, D. A. Lauffenburger and F. M. White, *Proc. Natl. Acad. Sci. U. S. A.*, 2007, **104**, 5860–5865.
45. L. L. Jin, J. Tong, A. Prakash, S. M. Peterman, J. R. St-Germain, P. Taylor, S. Trudel and M. F. Moran, *J. Proteome Res.*, 2010, **9**, 2752–2761.
46. J. M. Held, D. J. Britton, G. K. Scott, E. L. Lee, B. Schilling, M. A. Baldwin, B. W. Gibson and C. C. Benz, *Mol. Cancer Res.*, 2012, **10**, 1120–1132.
47. A. M. Zawadzka, B. Schilling, J. M. Held, A. K. Sahu, M. P. Cusack, P. M. Drake, S. J. Fisher and B. W. Gibson, *Electrophoresis*, 2014, **35**, 3487–3497.
48. R. Huttenhain, S. Surinova, R. Ossola, Z. Sun, D. Campbell, F. Cerciello, R. Schiess, D. Bausch-Fluck, G. Rosenberger, J. Chen, O. Rinner, U. Kusebauch, M. Hajduch, R. L. Moritz, B. Wollscheid and R. Aebersold, *Mol. Cell. Proteomics*, 2013, **12**, 1005–1016.
49. M. J. Rardin, J. M. Held and B. W. Gibson, *Methods Mol. Biol.*, 2013, **1077**, 121–131.
50. A. Darwanto, M. P. Curtis, M. Schrag, W. Kirsch, P. Liu, G. Xu, J. W. Neidigh and K. Zhang, *J. Biol. Chem.*, 2010, **285**, 21868–21876.
51. S. D. Sherrod, M. V. Myers, M. Li, J. S. Myers, K. L. Carpenter, B. MacLean, M. J. MacCoss, D. C. Liebler and A. J. Ham, *J. Proteome Res.*, 2012, **11**, 3467–3479.
52. A. L. Creech, J. E. Taylor, V. K. Maier, X. Wu, C. M. Feeney, N. D. Udeshi, S. E. Peach, J. S. Boehm, J. T. Lee, S. A. Carr and J. D. Jaffe, *Methods*, 2015, **72**, 57–64.
53. J. G. Abelin, J. Patel, X. Lu, C. M. Feeney, L. Fagbami, A. L. Creech, R. Hu, D. Lam, D. Davison, L. Pino, J. W. Qiao, E. Kuhn, A. Officer, J. Li, S. Abbatiello, A. Subramanian, R. Sidman, E. Y. Snyder, S. A. Carr and J. D. Jaffe, *Mol. Cell. Proteomics*, 2016, **15**, 1622–1641.
54. A. Maiolica, M. A. Junger, I. Ezkurdia and R. Aebersold, *J. Proteomics*, 2012, **75**, 3495–3513.
55. R. Costenoble, P. Picotti, L. Reiter, R. Stallmach, M. Heinemann, U. Sauer and R. Aebersold, *Mol. Syst. Biol.*, 2011, **7**, 464.
56. Q. Fu, Z. Chen, S. Zhang, S. J. Parker, Z. Fu, A. Tin, X. Liu and J. E. Van Eyk, *Methods Mol. Biol.*, 2016, **1410**, 249–264.
57. A. Vegvari, K. Sjodin, M. Rezeli, J. Malm, H. Lilja, T. Laurell and G. Marko-Varga, *Mol. Cell. Proteomics*, 2013, **12**, 2761–2773.

CHAPTER 10

Data Analysis for Data Independent Acquisition

PEDRO NAVARRO[†a]*, MARCO TREVISAN-HERRAZ[†b] AND HANNES L. RÖST[†c]

[a]Institute for Immunology, University Medical Centre of the Johannes Gutenberg University Mainz, 55131 Mainz, Germany; [b]Centro Nacional de Investigaciones Cardiovasculares, 28029 Madrid, Spain; [c]Department of Genetics, Stanford University, Stanford, CA 94305, USA
*E-mail: pnavarro@uni-mainz.de

10.1 Analytical Methods

10.1.1 Motivation

As is apparent from the other chapters in this book, the most popular workflow in proteomics is the so-called *shotgun proteomics* approach where proteins are enzymatically cleaved to produce a mixture of peptides that are then separated by online liquid chromatography (LC) coupled to tandem mass spectrometry (MS/MS). Within this workflow, data independent acquisition (DIA) and its counterpart data dependent acquisition (DDA) are included.

[†]These authors contributed equally.

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

Since the development of mass spectrometry-based proteomics, a great research effort was made to improve spectrometers and technology towards better and faster ways to identify the proteins present in a sample. Between 2000 and 2010 the main techniques were focused on taking advantage of the available data. This led to the development of high throughput proteomics and different flavours of DDA. During the same decade DIA arose,[1] but the germinal DIA workflows were not competitive with DDA at that moment. The introduction of better instruments, as well as the development of bioinformatics, gave way to an explosion in the amount of data produced, which allowed DIA to come into the proteomics arena with competitive approaches.

These new methods arose as an innovative way to address the main drawbacks of DDA or selected reaction monitoring (SRM, see previous chapter), for both targeted and shotgun proteomics, allowing the characterisation of practically all proteins in complex samples without discrimination.

10.1.2 Background: Other MS Methods

In DDA, in an effort to subject as many peptide precursors as possible to sequencing, the mass spectrometer first performs an MS1 survey scan, which is used to select the most abundant precursor ions for fragmentation. The selected precursor ion then undergoes fragmentation and the resulting fragments are recorded in an MS2 fragment ion scan which provides extensive amino acid composition information for the selected precursor at a specific time point. This is repeated for other precursor ions for a determined number of candidates, then a new MS1 survey scan is performed and the process is repeated.[2] The number of candidates isolated on each duty cycle depends on the acquisition frequency at which the mass spectrometer can acquire a fragment spectrum with sufficient quality to characterise the peptide sequence.

This strategy, known as data dependent acquisition (DDA), is a highly efficient method to obtain fragment ion information since it samples precursor ions at positions with high MS1 intensity and thus increased likelihood of obtaining a high-quality fragment ion spectrum. When applied to whole cell lysates, shotgun proteomics provides fast enumeration of the most abundant peptide species present in the sample which allows for exploratory data analysis and achieves identification of previously unknown proteins. However, while DDA allows discovery-driven research and offers high throughput, its sensitivity may be limited by undersampling issues in complex samples and it suffers from ambiguity in spectra assignments to peptides and inconsistent identification reproducibility across samples due to the on-the-fly precursor selection by the data dependent algorithm.

In order to address these issues, alternative data acquisition strategies were developed, aiming for higher reproducibility. Most prominent among these methods is SRM (see Chapter 9), a targeted method that uses sensitive mass spectrometric assays to selectively monitor a set of pre-selected peptides.[3–6]

In SRM, data is recorded repeatedly across the LC-time dimension at predefined precursor and fragment ion mass-over-charge (*m*/*z*) pair values (so-called transitions) and identification specificity is usually achieved by assessing the co-elution of the time-resolved traces of multiple transitions of the same peptide.[7,8] This strategy exploits the capacity of triple quadrupole mass spectrometers to selectively isolate specific precursor ions, fragment them and monitor their fragment's intensities across chromatographic time. While SRM offers high reproducibility, dynamic range, sensitivity and good signal-to-noise ratio, it comes at the cost of significantly lower throughput and requires time-consuming assay development which pre-determines and limits the number of hypotheses that can be tested in an experiment.[6,9,10] Compared to shotgun proteomics, the sampling procedure employed in SRM is significantly less efficient. In order to ensure that a signal is recorded reproducibly, the mass analyser has to monitor a set of transitions repeatedly for several minutes, only to finally record a signal of an LC-elution span of several seconds in length. Therefore, most of the measurement time is lost recording data at positions in the MS1–MS2 space where no signal is present.

With the choice between shotgun and targeted proteomics methods, the proteomics researcher is thus faced with a decision between obtaining snapshots of extensive fragment ion data of a population of peptides sampled from the pool of available peptides (shotgun) or obtaining time-resolved fragment ion intensities for a lower number of predetermined peptides (targeted).[4] In terms of scanning efficiency, it may be argued that shotgun is "too efficient" to the point of irreproducibility whereas SRM is "too conservative" to the point where most of the measurement time is spent recording noise in order not to miss any signal.

10.1.3 DIA Concept

As an alternative to data dependent shotgun proteomics and targeted SRM, researchers have been studying data independent acquisition (DIA) as a method for high throughput proteomic analysis.[11–23] In DIA mode, the instrument fragments all precursors generated from a sample that are within a predetermined *m*/*z* and retention time range, regardless of the presence of precursors in that region (hence the terminology "data independent"). Usually, the instrument cycles through the precursor ion *m*/*z* range in segments of specified width.

Thus, DIA does not explicitly target single precursors but rather fragments whole bands of the precursor ion range simultaneously. Generally, a set of *n* mass isolation ranges are chosen to cover most doubly and triply charged peptide precursors (for example, mass isolation ranges of 25 *m*/*z* width covering the mass range of 400–1200 *m*/*z* in 32 steps) which gives the researcher large flexibility in data analysis. In practice, the instrument will usually generate an MS1 scan at the beginning of each cycle. Next, however, instead of isolating specific precursors from the survey scan as in shotgun, it will isolate

all precursors in a specific precursor isolation window, subject them to fragmentation and acquire a complete (high resolution) fragment ion spectrum. The instrument then proceeds to the next precursor mass range (also called swath) and repeats the process, thus stepping through a set of isolation windows. In the example mentioned, the instrument would initially co-fragment all precursors between 400 and 425 *m*/*z* and perform a high resolution fragment ion scan in the first cycle, then proceed to the 425 to 450 *m*/*z* window *etc.*, until the end of the cycle is reached. After reaching the last window, a new cycle is started with another MS1 and a set of fragment ion scans, generated in the same order as the previous cycle. Note that if the number of windows and time per scan is constant (as it usually is), the scheme will produce a set of *n* fragment ion scans of exactly the same precursor isolation window every few seconds (for example, every 3.3 seconds for 32 windows with 100 ms acquisition time per scan).

Multiple variations of the DIA theme have been described with different instrument types and setups, duty cycles and window widths. Methods like MS^E fragment all precursors (basically implementing a single, very large isolation window and the duty cycle consists of alternating MS1 and MS2 scans)[13] while others such as PAcIFIC use precursor selection windows as small as 2.5 *m*/*z*. (See Law *et al.*[24] and Chapman *et al.*[25] for recent reviews of different DIA approaches.)

However, common to all DIA approaches is that they use a fixed, deterministic acquisition scheme and that they acquire full (usually high resolution) fragment ion spectra. As in the case of SRM, the deterministic acquisition strategy makes DIA highly reproducible and therefore repeat injections of the same sample will produce highly similar data. The acquisition of full fragment ion data provides additional information compared to SRM, especially if high resolution scans are acquired. Compared to shotgun approaches which only provide single snapshot fragment ion spectra, DIA additionally provides information about the elution profile of each fragment ion. When all scans from the same isolation window are aggregated, a DIA-map (signal intensities coming from one isolation window, represented along *m*/*z* and retention time axes) is generated that is continuous in time *and* fragment ion intensity, and can be analysed in both dimensions. These maps contain the full fragment ion signal for a specific isolation window sampled at regular intervals and can thus be considered a complete digital representation of a proteomic sample. Every fragment ion signal above the limit of detection will be recorded in the corresponding map, given that the sampling frequency is dense enough and that the precursor falls into one of the predetermined isolation windows. It is thus always possible to re-examine any DIA measurement for evidence of a specific peptide precursor and its fragment ions as new hypotheses are generated. It can thus be argued that DIA methods attempt to find a compromise in scanning efficiency between shotgun and targeted proteomics. The highly efficient shotgun scheme of producing fragment ion data only when a promising precursor is detected is replaced with a strategy of frequent fragment ion acquisition but with a large enough

precursor isolation window to ensure capture of one or more precursor ions. At the same time, the precursor isolation window is not as small as in SRM where many MS cycles are spent isolating and fragmenting an area of the precursor space where no precursor is eluting and only a small portion of all precursors can be covered, limiting throughput.

10.1.4 Theoretical Considerations

One way to understand the strength and challenges of the different methods discussed is with regard to the structure of the output data. In most quantitative proteomic studies, the goal is to measure protein concentrations across multiple samples (experimental conditions, time series, patient samples *etc.*). Especially for systems biology studies, obtaining quantitative measurements of the analyte concentrations is crucial as it allows researchers to understand the systems' behaviour on a molecular level. In these studies, the measurement output is generally a two-dimensional data matrix containing quantitative measurement values of specific analytes (first dimension) across multiple samples (second dimension). For successful downstream data analysis, the comprehensiveness and accuracy of the data matrix in *both* dimensions is equally important.

The data produced by shotgun proteomics poses significant challenges with regard to the proteomic data matrix. While shotgun allows measurements to be performed with high throughput and coverage, the data generally has low comprehensiveness. In the resulting data matrices, the data is often only complete for the most intense peptides of high abundance proteins but contains missing values for proteins of lower abundance.[26] In addition, the more samples are analysed and the more biologically diverse the samples are, the lower the number of complete rows; due to the intensity-dependence of the sampling and undersampling issues for complex samples, the missing values will generally not be missing completely at random.[27]

On the other hand, data matrices generated by SRM are much more complete than those produced by shotgun proteomics, but generally contain one to two orders of magnitude fewer proteins.[28] Since the proteins to be measured have to be pre-selected, the measurements tend to be biased by prior hypotheses and may not cover all biologically relevant cellular processes and pathways.[29] Therefore, SRM has been mostly used in studies where large sample numbers are required and only few proteins were under investigation.

However, DIA has the potential to address both dimensions of the data matrix at the same time and thus allows true systems analysis on protein measurements. When analysed using a targeted extraction strategy, as in SWATH-MS (a DIA method based on sequential windowed acquisition of all theoretical fragments), the approach combines the strength of SRM (high reproducibility and quantitative accuracy) with the high throughput of shotgun proteomics, thus focusing on both analyte and sample dimension of the

data matrix at the same time. The targeted analysis approach to DIA data produces multiple highly reproducible fragment ion chromatograms for each peptide in each measured sample, which are conceptually similar to SRM traces. However, unlike SRM, the number of extractable analytes in DIA is not constrained by instrument speed and thus substantially higher throughput can be achieved. It is not uncommon to extract several tens of thousands of peptide chromatograms from a single injection, and the technique is able to achieve comparable proteome coverage as shotgun. Thus, DIA is able to produce data matrices that are quantitatively accurate and qualitatively complete, allowing researchers to track protein quantities across many samples in high throughput.

Another way to conceptualise the differences between the three acquisition methods in LC-MS/MS-based proteomics can be understood in terms of sampling efficiency. The problem to be solved can be posed as how to sample best from a two-dimensional MS1-RT space if only a limited amount of samples can be taken due to instrument speed.

In shotgun proteomics, small sampling windows are used (small arrows of width 1-2 *m*/*z* displayed in top and middle panels of Figure 10.1) but sampled at positions of highest signal density with the hope of capturing signal with high information content – using small sampling windows facilitates subsequent data analysis and establishes a clear relationship between precursor ion and fragment ions. Targeted proteomics methods such as SRM opt to sample only a very limited part of the MS1-RT space (and even do not acquire full fragment ion spectra; not shown in the figure). The deterministic sampling scheme of SRM makes the method highly reproducible across repeated measurements (Figure 10.2). If a peptide precursor with a given mass elutes during the experiment, it will be fragmented and measured in every measurement even if its intensity is low compared to other analytes in the sample (whereas in shotgun proteomics, depending on the number of co-eluting species, it might get selected for fragmentation in some measurement runs but not in others). This "guarantee" of detection and quantification in SRM (given the analyte is present in sufficient amount) makes the method very attractive to replace antibody-based methods for protein quantification, such as western blotting, in the laboratory and in the clinics. Unfortunately, SRM does not easily lend itself for systems biology analysis since only a small portion of the system can be observed in a single measurement run. SWATH-MS now extends targeted proteomics methods by implementing a deterministic acquisition scheme but covering the whole MS1-RT space, thus implementing a high throughput targeted proteomics method. The method allows the extension of the targeted proteomics approach and generation of chromatographic signals for peptide analytes *in silico*; in theory it is thus able to investigate the presence of fragment ion signals for any analyte even after acquisition of the data. SWATH-MS data therefore constitute a complete digital record of all fragment ions produced from a biological sample capturing the complete analyte fragmentation information obtainable from such a sample in one single experiment.

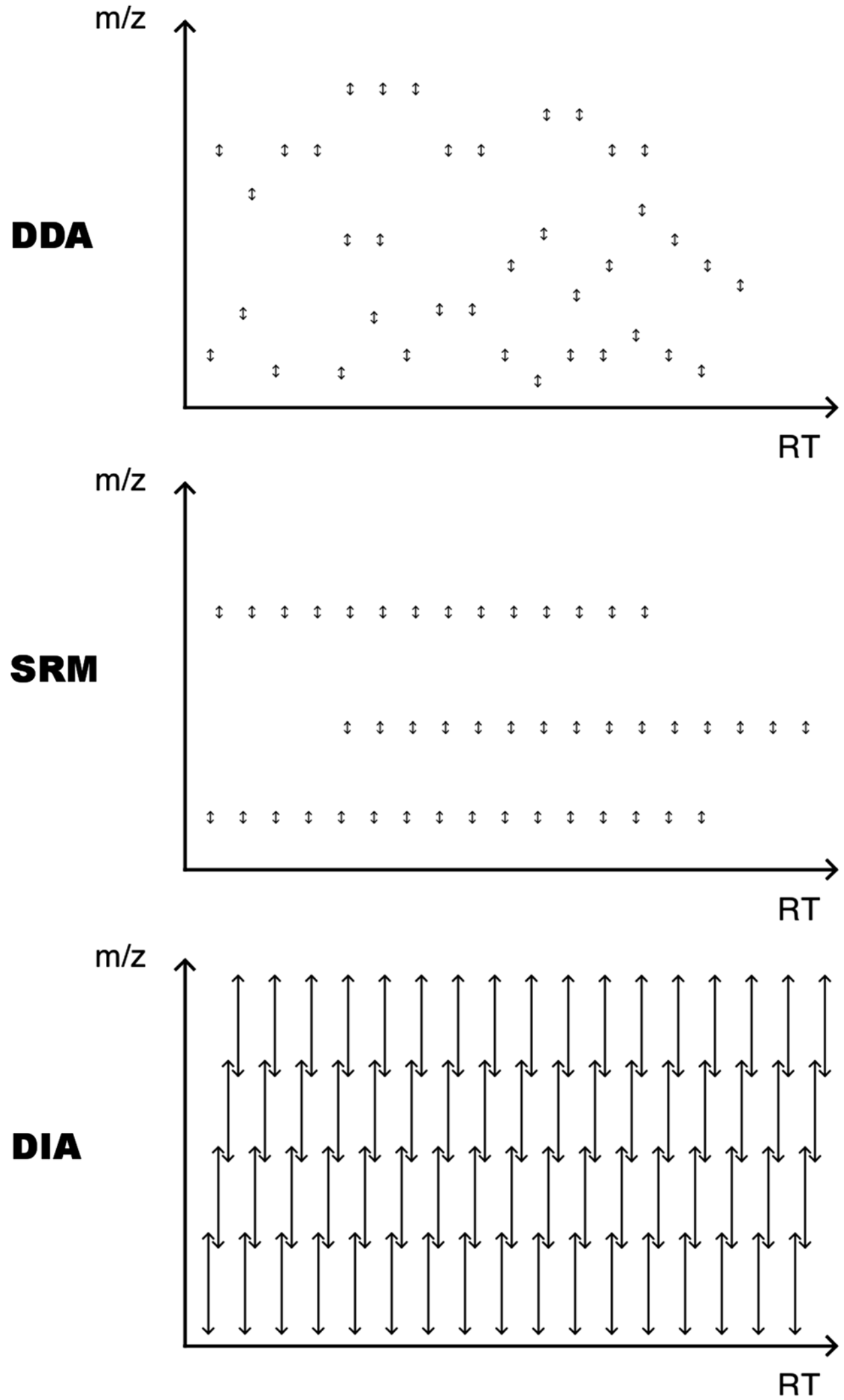

Figure 10.1 Schematic representation of the main acquisition modes in a mass spectrometer. Displayed are (i) shotgun acquisition or DDA (data dependent acquisition), (ii) targeted proteomics or SRM acquisition and (iii) SWATH-MS or DIA (data independent acquisition). The two axes represent precursor ion *m*/*z* space and chromatography retention time in an LC-MS/MS experiment. Arrows depict precursor isolation windows selected for fragmentation. Note how (i) and (ii) cannot cover the whole space while SWATH-MS fragments every single possible precursor at the expense of larger isolation windows. Only SRM and SWATH-MS are deterministic methods while shotgun is data dependent, fragmenting preferably at positions of high ion density.

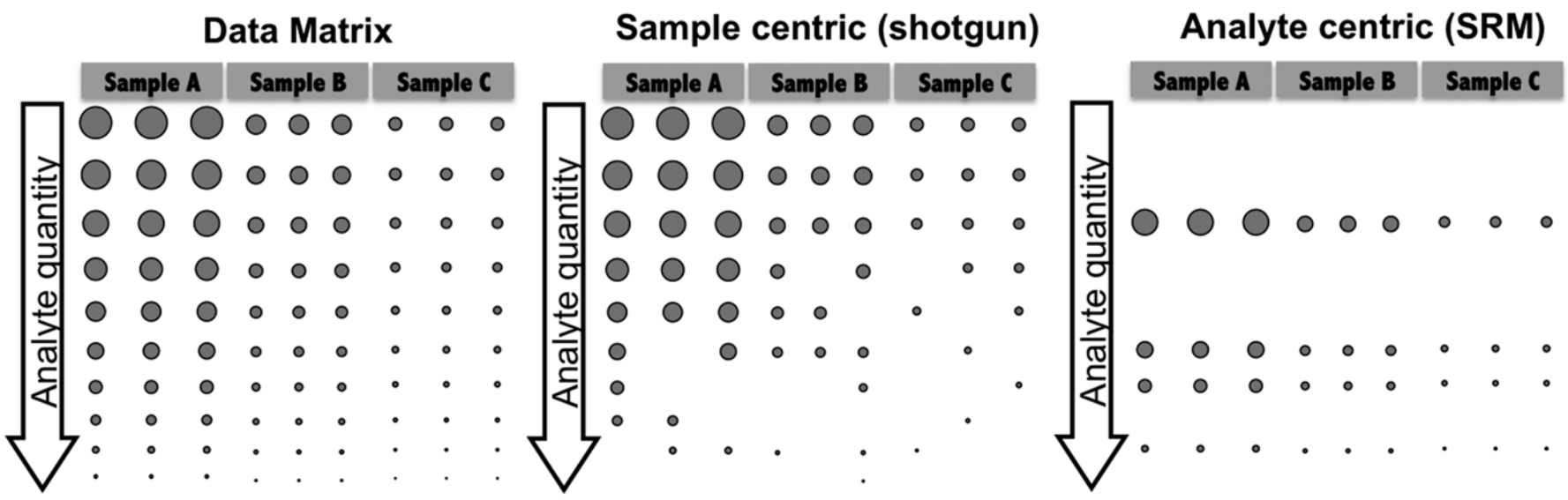

Figure 10.2 Representation of a theoretical data matrix in a proteomics experiment, and how DDA and SRM record this matrix using circles to indicate protein quantity. (i) The ideal data matrix contains quantitative values for analytes measured across multiple samples, achieving high throughput (large number of quantified analytes) consistently across many samples (experimental conditions, perturbations, or patient samples). (ii) Sample-centric workflows (such as discovery proteomics or shotgun proteomics) use data dependent acquisition to achieve high number of identifications per sample. However, they sacrifice sampling consistency and usually not all analytes can be quantified in every single sample. (iii) Analyte-centric workflows (such as SRM and other low-throughput targeted proteomics techniques), on the other hand, achieve highly consistent quantification across many samples. However, these techniques only cover a few, carefully selected analytes.

10.1.5 Main DIA Methods

10.1.5.1 PRM

See a summary of DIA methods in Table 10.1. Parallel reaction monitoring[30] (PRM) is a targeted proteomics method which substitutes the third quadrupole in a typical QQQ setup with a high resolution mass analyser (for example an Orbitrap™). This allows PRM to obtain a complete recording of all fragment ions produced by a specific precursor (instead of a pre-selected number as in SRM). In addition, the high mass resolution (parts per million compared to 0.2–1 m/z) in conjunction with the ability to monitor a virtually unlimited number of fragment ions allows for higher specificity in PRM compared to traditional SRM. Specialised software is then used to extract individual fragment ions from all high resolution MS2 spectra associated with a given precursor in order to reconstruct fragment ion traces. Note that while the fragment ion traces produced by PRM look similar to SRM traces, PRM records substantially more data and allows computational re-analysis, for example using a different set of transitions if an interference is detected in the initial set of transitions. Compared to SRM, PRM allows for reduced assay development time and may produce accurate quantification over a larger dynamic range than SRM.[30] However, the limits in throughput (number of peptides that can be measured per run) are not substantially different than in SRM, making PRM a low-throughput DIA method.

Table 10.1 Comparison of several commonly used DIA methods. The table compares six current DIA methods based on duty cycle, analytical separation as well as data analysis strategy. The table illustrates the tradeoffs available when running DIA in terms of duty cycle, acquisition time and isolation window.

	PRM	MSE/AIF	HDMSE	PAcIFIC	SWATH-MS	MSX
Analyte separation methods	HPLC +, mass isolation +++	UHPLC ++	UHPLC ++, ion mobility +	HPLC +, mass isolation +++	HPLC +, mass isolation ++	HPLC +, mass isolation ++
Duty cycle	~3 s	~1 s	~1 s	~2.5–3.5 s	~3 s	~2 s
Isolation window size	1.0–2.0 *m*/*z*	No isolation	No isolation	2.5 *m*/*z*, with 1.0 *m*/*z* overlap	Medium–big (8 *m*/*z*–30 *m*/*z*)	20 *m*/*z*, computationally reduced to 4 *m*/*z*
Acquisition time – number of samples needed	>1 day (>10 injections)	2–3 hours (1 injection)	2–3 hours (1 injection)	>1 day (>10 injections)	2–3 hours (1 injection)	2–3 hours (1 injection)
Identification strategy	Targeted	Untargeted	Untargeted	Untargeted	Both	Targeted
Quantification	Both	MS1	MS1	MS1	MS2	MS2

10.1.5.2 MS^E/HDMS^E/AIF

MS^E is a data acquisition method performed in a quadrupole–TOF mass spectrometer that alternates two different collision energy modes (low-energy and elevated-energy) in order to acquire — practically — continuous data of precursor ions entering the mass spectrometer and the fragment ions produced by those precursors. The absence of any isolation process (except for a low mass filter, typically of about 350 *m*/*z*) yields a fast analytical process with little sample loss.[31] Peptide characterisation performance relies on a great chromatography separation (typically based on columns filled with sub 2 μm particles, and that therefore needs ultra high performance liquid chromatography, UHPLC). When an ion mobility separation is used in combination with MS^E, the method is called High Definition MS^E (HDMS^E).[19] The separation provided by ion mobility allows the number of characterised peptides to rise from the order of hundreds (MS^E) to several thousands (HDMS^E).[32] Ion mobility can be also coordinated with the collision energy utilised at the elevated-energy spectra to improve the fragmentation efficiency.[32]

A very similar acquisition scheme can also be applied to high resolution trap-based systems like Orbitrap™,[23] where the acquisition of fragments with no fragment ion isolation is called All-Ion Fragmentation (AIF). Orbitrap™ can offer a better mass resolution than time-of-flight instruments (100 000 compared with 10 000 resolving power),[23] and though the analyser is slower than time-of-flight, the simultaneous use of several traps (like the C-trap and the analyser Orbitrap™) and cells (like the collision cell) ensures a good sample preservation.

In practice, these methods provide two retention time–mass maps (or 2D data arrays), that correspond to the locations of the precursors and fragments. Features found in these two maps can be related by their similarity in retention time and eventually in ion mobility.

10.1.5.3 PAcIFIC

Precursor Acquisition Independent from Ion Count (PAcIFIC) is an acquisition method based on isolation and fragmentation of contiguous narrow *m*/*z* intervals (as in Panchaud *et al.*,[14] 2.5 *m*/*z* isolation widths with 1.0 *m*/*z* overlaps). The narrow isolation windows used in PAcIFIC determine the peptide precursor mass without further signal alignment with precursor spectra, and therefore generated spectra can be matched to peptide sequences by using conventional (from data dependent acquisition) database search engines. Acquisition of this practically monoplexed fragment spectra is time consuming, and thus the different isolation windows that should be acquired are usually distributed in several injections of the same sample. The number of injections needed per sample depends on the total *m*/*z* range the researcher desires to cover (this range varies typically from 400 *m*/*z* to 1200–1400 *m*/*z* in a proteomics experiment), and the sampling frequency of the mass spectrometer. Panchaud *et al.* reported results by using cycles of 10 consecutive isolation windows of 1.5 *m*/*z* effective coverage each, requiring 167 injections

to aim for a total acquisition range of 400–1400 *m*/*z*.[14] The larger number of samples required is the biggest limitation of the method. One interesting computational challenge of this approach is the optimisation of the different *m*/*z* range widths and spectra acquisition cycles (*i.e.* the number of isolations per cycle) in order to reduce the acquisition time and the number of samples required, as the PAcIFIC authors successfully showed in a subsequent publication.[15] In this regard, the continuous development of mass spectrometers in terms of sampling rate will further facilitate in the future the reduction of the number of injections needed to cover a full *m*/*z* range.

10.1.5.4 SWATH-MS

SWATH-MS uses the high mass resolution of recent instruments in a similar fashion as PRM but additionally employs multiplexing to achieve higher throughput.[33] In PRM, a single targeted peptide precursor ion is selected, fragmented and its fragment ions recorded in a high resolution MS2 scan. SWATH-MS does not explicitly target single precursors but rather fragments whole bands (swathes) of the precursor ion range simultaneously and then uses downstream software to computationally create ion traces, in a similar fashion to PRM. After acquisition, researchers can decide which peptides and which transitions to extract from the dataset and are not limited to the set of recorded precursor–fragment ion pairs (as in SRM) or to a restricted set of recorded precursors (as in PRM). For example, if a researcher would like to analyse a peptide with a precursor mass of 410 *m*/*z* and two fragment ions at 500 and 600 *m*/*z*, the analysis software would collect all high-resolution fragment ion spectra from the swath 400–425 *m*/*z*. In those spectra, the analysis software would then extract the signal at the *m*/*z* of the two requested fragment ions at 500 and 600 *m*/*z*.

10.1.5.5 MSX

MSX is a targeted proteomics strategy that works similarly to SWATH-MS but includes a de-multiplexing strategy for the highly multiplexed fragment ion spectra produced in DIA. In MSX, multiple precursors from non-consecutive isolation windows (five, for example) are co-fragmented in each cycle and co-fragmenting different precursors in each cycle together allows for deconvolution of the different isolation window during post-processing. In this manner a 20 *m*/*z* isolation window can be deconvoluted into five "virtual" isolation windows that have an effective width of only 4 *m*/*z*, allowing for much higher specificity. In this manner, the MSX approach has the potential to combine the throughput of SWATH-MS with the specificity of PRM.[34]

10.1.6 Analyte Separation Methods

Proteomics and mass spectrometry fall into the field of analytical chemistry, the art of classifying analytes by separating them. In mass spectrometry, several separation methods are simultaneously applied to each regular MS

acquisition: mass to charge separation (in two variants: through precursor ion isolation, and through ion detection), elution in liquid chromatography separation, and ion mobility separation are the most used. We will discuss now their ability to separate the analytes and their independency (or orthogonality) among them.

The most obvious analyte separation in mass spectrometry happens in the mass (to charge) dimension. Mass spectrometers classify the ions according to their mass to charge ratio. This separation occurs in two stages: a first, optional stage of ion isolation (and subsequent fragmentation of these precursor ions), and secondly at the ion detector. The precision in the mass to charge dimension in these two stages is uneven: isolation is generally resolved by quadrupole mass filter that are very fast in selection, but less precise than mass to charge detectors. At the time of writing, high precision detectors are prevalently used in DIA, allowing a precision of less than 10–15 parts per million. In peptide characterization, a higher precision does not typically separate analytes better due to the reduced number of elements (20 amino acids) that contribute to the building block structure of peptides.

The separation in elution time (chromatography) depends on several physical parameters. The most important are: analysis column length, column material (especially the particle size), solvents composition, and gradient applied to the solvent mixing. All of them affect the precision and sensitivity of the chromatography, and should be taken into account for quality benchmarking. It is of special interest as a computational challenge the fact that peptide elution order depends on the column material, making harder to collect libraries of peptide elution times that can be used to limit the retention time range in which a peptide should be found. There exist predictive models of peptide elution for several column materials,[35] but mass spectrometrists tend towards the use of libraries of measured elution times. A typical precision obtained by High Performance Liquid Chromatography (HPLC) and its improved Ultra-Performance Liquid Chromatography (UPLC) is about 30 s and 5–10 s respectively in gradients of 2 hours. Chromatographic separation has some correlation with peptide mass, and thus is not absolutely orthogonal to precursor isolation separation.

The last separation method we discuss, ion mobility (IM), consists of a gas-phase separation based on the collision cross-section of the analytes.[34,36] Ion mobility can be coupled to a time of flight analyser, making it compatible with mass spectrometry. Ions are separated by their interaction with a buffer gas through a collision cell (also called mobility cell), and can then be examined in the mass analyser. This essentially adds one more dimension to the analysis which can be used to deconvolute the resulting multiplexed fragment ion spectra. Similarly to peptide liquid chromatography, ion mobility shares a strong correlation with mass, but it has an interesting advantage compared to liquid chromatography: since spectra can be deconvoluted by collision cross-sections, it allows the separation of ions by charge state, since ions with similar *m*/*z* value but different mass (and thus also different charge state) will show very different collision cross-section values. Classification of ions by charge state can help to peptide characterization in two ways: it

reduces the search space among the different peptide candidates, and it also diminishes spectral noise by removing singly charged ions, since most of peptides generated in regular shotgun experiments (using trypsin as digestion enzyme) are doubly or triply charged.

Identification in DDA analyses relies on parent ion isolation of the peptides subjected to be identified. This isolation is performed in a narrow *m*/*z* range that grants the central assumption of most DDA analysis pipelines: the subsequent fragment ions spectrum is a consequence of an individual compound, and therefore ions above the noise level can be assigned to a single peptide. Although some fractions of DDA MS fragment spectra are chimeric (*i.e.* contain fragment ions from multiple peptide analytes), parent ion isolation remains a powerful source of analyte separation. DIA methodologies, on the other hand, cover wide *m*/*z* ranges in the aim of collecting data from all precursors in a sample, and most of these methods (with the omission of PAcIFIC) do not rely on individual isolation of precursor ions. Instead, in DIA the other analyte separation methods assume a more decisive role, and mass spectrometrists generally agree that DIA requires a higher effort in achieving good chromatography than DDA.

10.2 Data Analysis Methods

10.2.1 DIA Data Analysis

The obvious advantage of DIA methods is that they create a highly reproducible record of the fragment ion signal of all precursors in a sample, therefore combining the high throughput of shotgun proteomics with the high reproducibility of SRM. The resulting data are continuous in time *and* fragment ion intensity, thus increasing the dimensionality of shotgun proteomics data where fragment ion intensities are recorded only at selected time points or SRM data where continuous time profiles are acquired but only for a few selected fragment ions. However, to limit analysis time (*i.e.* number of LC injections) and sample amount, larger precursor isolation windows than in shotgun proteomics or SRM are typically used. This leads to highly complex, composite fragment ion spectra from multiple precursors and thus to a loss of the direct relationship between a precursor and its fragment ions, making subsequent data analysis non-trivial.

To analyse these highly complex data, two main strategies have emerged. In the first approach, the spectrum-centric approach, the multiplexed spectra from DIA data are searched using traditional shotgun MS search engines either directly[12] or after computation of pseudo-spectra containing fragments assigned to a precursor based on their co-elution profiles.[16,17,19–21] However, the spectrum-centric approach suffers from the high complexity of the data and the fact that errors in the generation of pseudo-spectra will propagate through the analysis workflow. In the second approach, the chromatogram-centric approach, the data are first reduced in complexity by extracting fragment ion chromatograms (XICs) of the most abundant fragment ions for

each peptide of interest. These XICs are then scored and analysed similar to SRM data using multiple fragment ion traces ("transitions") per peptide. This approach has proved to be very successful in high throughput settings, however it relies heavily on *a priori* knowledge in the form of spectral libraries which contain the fragment ion coordinates for each peptide and require considerable effort to generate.

10.2.2 Untargeted Analysis, Spectrum-Centric

Untargeted analysis in DIA is referred to approaches that do not initially filter MS acquired data by using a prior knowledge derived from a set of peptides that we try to identify in the sample. In other words, the paradigm of DIA untargeted analysis is to process, organise, and clean DIA data with no external information influence, and to use this processed data to compare to a knowledge previously acquired. This knowledge in the case of proteomics is a set of proteins, and from these proteins we might know only their amino acid sequences, or we might also have prior spectral information. The fundamental difference among DIA targeted and untargeted analyses is the order in which we try to relate acquired data from a sample (ion signals) to information about its content (proteins or peptides) (Figure 10.3).

10.2.2.1 Signal Clustering

In DIA untargeted analyses it is common to use a "pseudo-spectrum" approach. The main task of this procedure consists of resolving the chimericity of the sample, deconvoluting it by separating the signals according to their peptide precedence. This signal deconvolution yields a set of MS ion fragment "pseudo" or "*in silico*" spectra (where a MS spectrum is a set of paired m/z and intensity values), which are also related to a parent ion if MS1

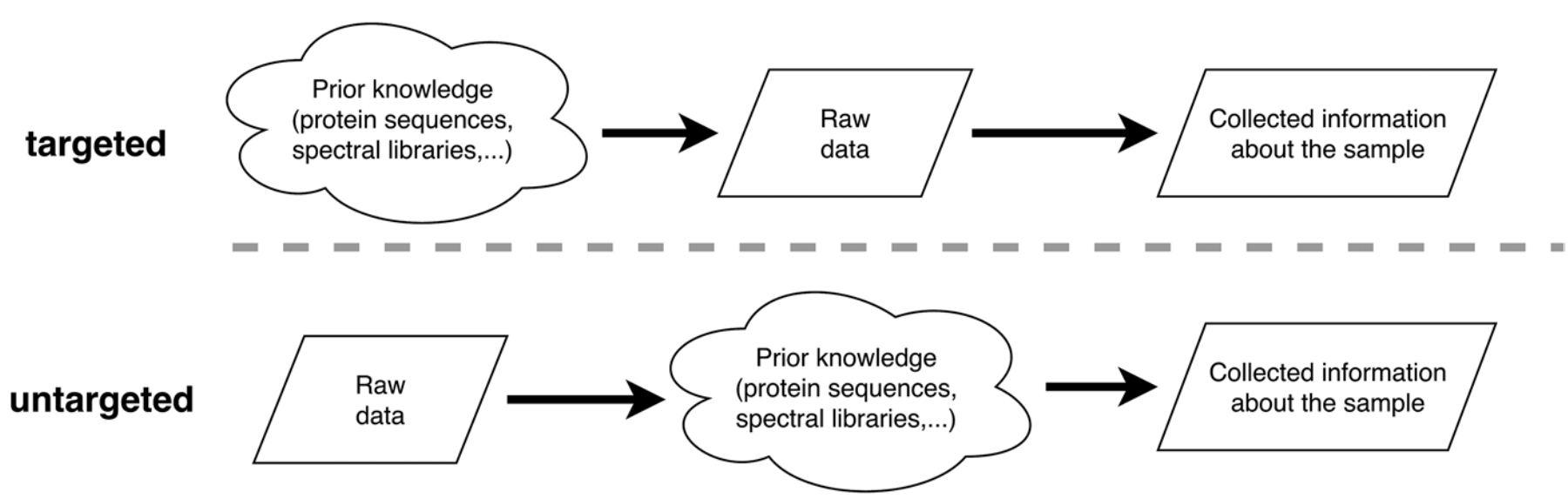

Figure 10.3 Data analysis logics of targeted and untargeted approaches. The fundamental difference between targeted and untargeted approaches is highlighted by the fact that a targeted approach generally needs highly specific prior knowledge about the samples to be analysed whereas untargeted approaches use the raw data directly to identify signal patterns likely corresponding to peptide analytes.

spectra have been acquired. Signal deconvolution is achieved through signal clustering. Fragment ion *m*/*z*–intensity pairs are grouped by using correlation in values of any analyte separation method used (with the obvious exception of the detected mass, which is a nominal part of the *m*/*z*–intensity pairs we want to group).

Precursor ion isolation (like in PAcIFIC or SWATH-MS) can be used as a simple data classification into *n* different datasets for *n* given different precursor ion isolation ranges. This also facilitates the parallelization of the entire analysis process. Since precision of precursor ion isolation is lower than ion *m*/*z* detection, it is common to program the isolations with some overlap between isolation ranges. In the case of untargeted analysis, we should take this into account when relating a set of fragment ion *m*/*z*–intensity pairs to isolation ranges.

Peptide elution time and ion mobility (both together, or just one of them) are used for the final ion grouping. Since fragmentation occurs after peptide elution and ion mobility separations, all ions product of the same precursor should have the same elution time and ion mobility profiles, and they can be clustered by profile correlation, after an intensity normalisation. The clustered sets of fragment ion *m*/*z*–intensity pairs can as well be related to one precursor ion by using these profiles (or several, if more than one candidate seems to correlate), but in the case of ion mobility it is important to consider that for fragmenting, ions are accelerated, causing a slight shift in ion mobility values to faster values of the ion mobility drift times. Due to the extreme complexity of proteome samples, it is very likely that several unrelated ions are added to clustered pseudo-spectra, impairing peptide identification. Some ways to increase the cluster selectivity are: implementing an intensity threshold, and using theoretical models of fragmentation, which estimate an approximate number of ions generated by fragmentation of a precursor of a certain mass (or number of amino acids).

10.2.2.2 Pseudo-Spectra Identification

The set of pseudo-spectra produced in the preceding step can be searched in conventional DDA database search engines, producing very good results in cases of lower complexity samples and excellent analyte separation. However, when the sample complexity is high, pseudo-spectra have an increased chimericity, and additional strategies should be taken into consideration to improve peptide identification. There are several proposed algorithms based on iterative database searching of the fragment ions set, which are able to identify multiple co-fragmenting peptides in one (pseudo) spectrum.[37] These algorithms subtract the fragments ions identified in previous iterations for further searches. Another strategy to improve identification is to select a subset of the clustered ions based on known peptide physical properties. Some of these properties are evident: for example, fragment ion masses cannot be bigger than the assigned precursor mass. In this case, we can discard a precursor mass candidate (if more than one is related to a cluster), or discard

those fragments. We can also use the correlation between peptide mass and peptide elution time to discard those clusters (or precursor candidates) with non-correspondent mass and elution time values. It is also known that peptide precursors produce fewer fragments at the *m*/*z* range over the precursor *m*/*z* value. In most DIA methods, the high number of co-fragmenting precursors makes the *m*/*z* region below 350 *m*/*z* not useful in practice, due to the huge number of small fragments — of three amino acids or less — of very similar (or exactly equal) mass. It is also interesting to observe that the intensity of the precursor should be related to the sum of intensities of the fragment ions, and you can apply simple linear models to all the clusters of an experiment in order to determine this relationship, and filter clusters that do not match the model.

These filtering steps may be applied before identification, or used as additional scores to improve selectivity of the identification of peptide-spectrum matches (after the database search of the clusters in a conventional search engine), as some current software tools do.[20]

Complete annotation of all peptide precursors present in an MS injection is a challenging task, and still today a great proportion of detectable peptide precursors in LC-MS/MS runs are not annotated.[10] One way to increase the rate of identified features in DIA runs is to complement the DIA identification workflow with DDA runs conveniently aligned with the DIA runs. DDA identifications show some orthogonality with DIA identifications at the peptide level, and good similitude at the protein level,[38] therefore its combination increases protein sequence coverage, and such a pipeline is already part of current software tools like DIA-Umpire.[38]

10.2.2.3 Peptide and Protein Quantification

You can estimate the peptide quantity in the sample using either precursor or fragment ion signals, both quantities correlate well. In the matter of DIA methods that use mass filter separation, like SWATH-MS or PAcIFIC, it could be more desirable to use fragment ion signals, as they are supposed to be "cleaner" after mass isolation. If fragment ion signals are used, due to probable chimericity it looks logical to use only annotated fragment ions and to limit the number of them in order to reduce the risk of adding intensity values of different species. Also, fragment selection is important in order to make consistent comparisons between different runs, and we should ensure that the chosen quantitation model does not introduce biased data. In that instance, you can select an *N* number of most intense fragment ions from the total pool of annotated fragment ions among all runs containing the peptide.

10.2.3 Targeted Analysis, Chromatogram-Centric

The chromatogram-centric approach to analyse DIA proteomic data has been proposed in 2004 by Venable *et al.*,[12] and put successfully into practice using SWATH-MS in 2012 by Gillet *et al.*[33] The approach (also known as the targeted

approach) is based on computational strategies commonly employed in targeted proteomics. Here, the DIA data are viewed as a large-scale targeted proteomics experiment where individual chromatographic traces are not produced at acquisition time but can be extracted from the data at analysis time. Thus, once the data are acquired, the chromatographic extraction can be repeated multiple times for different sets of target peptides. However, compared to SRM, in DIA much larger precursor isolation windows are used and many more peptides are analysed than in typical SRM experiments. Therefore, automated tools that implement appropriate statistical scoring and error rate estimation are crucial for data analysis. With *OpenSWATH*, Röst *et al.*[39] described the first such automated algorithm in the literature, but since then multiple alternative software tools have been created or existing SRM tools have been adopted to perform targeted analysis of DIA data.

Even though DIA targeted methods record high resolution fragment ion data to achieve high specificity, a fragment ion trace (or transition) may not be specific for a particular peptide, especially in a complex sample. Other peptides with similar precursor and fragment masses may produce interfering, non-specific signals. These peptides might either produce identical fragment ions due to (partial) sequence similarity or simply by chance. Since relatively large precursor isolation windows are usually employed, like in SWATH-MS, the fragment ion space is highly crowded and a fragment ion m/z does not usually map uniquely to a single peptide.[8] In addition, non-canonical protein isoforms, post-translational modifications and the natural isotope distribution increase the likelihood of such interferences occurring. Without automated, unbiased evaluation and scoring, such signals can easily be mistaken for the true signal, which could lead to the quantification of the wrong signal (*e.g.* signal unrelated to the analyte) which would greatly impact the accuracy of the quantitative data matrix.

A typical chromatogram-centric analysis workflow may be organised in five distinct steps, which will be discussed here in greater detail: (i) retention time normalisation, (ii) chromatogram extraction, (iii) peak group scoring, (iv) error rate estimation and (v) optional cross-run alignment. As in any targeted workflow, the analysis relies on highly specific assay coordinates (fragment ion intensity and m/z as well as retention times for each peptide analyte). A discussion on generating such assay libraries would be beyond the scope of this chapter but can be found for example in Schubert *et al.* (Figure 10.4).[40]

10.2.3.1 Retention Time Normalisation

During retention time (RT) normalisation, a (linear) function is computed to transform the retention time space of an individual run into normalised retention time. For this, a set of anchor points (normalisation peptides) are chosen so that coordinates are known in both spaces, usually a standardised set of spiked-in or endogenous peptides is used for this step.[41,42] First, the dataset is investigated for the normalisation peptides by extracting the traces for these peptides and identifying the best overall peak

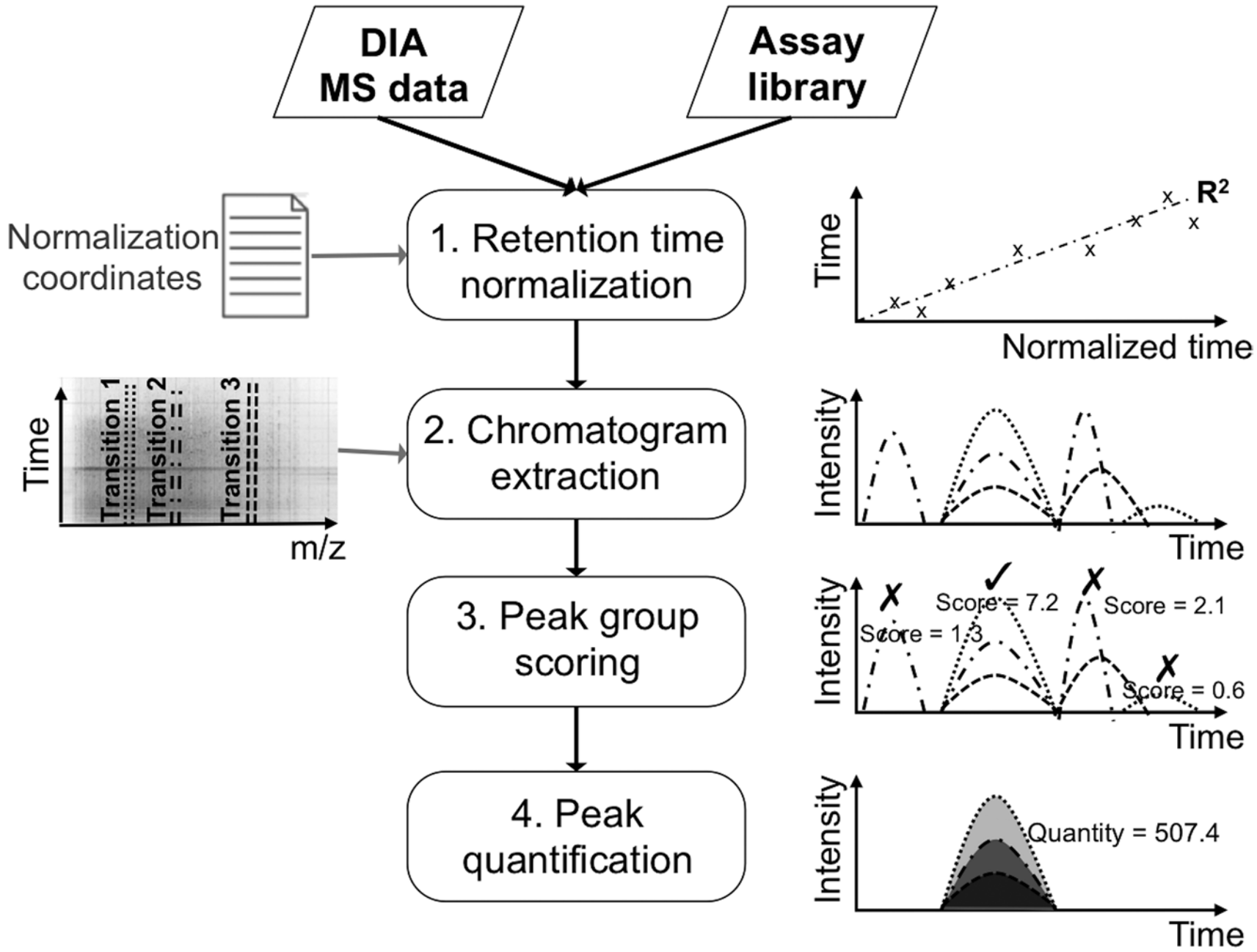

Figure 10.4 Analysis steps in a DIA targeted workflow. Generally speaking, a targeted workflow uses *a priori* data (assay library) and raw data together to extract chromatogram traces at the fragment *m*/*z* and retention time space where the analyte is most likely to elute. The workflow may include a (linear) retention time normalisation step (1), chromatogram extraction (2) and subsequent peak group scoring (3) which automatically reports peak groups above a given score cut-off followed by quantification (4).

group. (Note that the coordinates in the normalised space are given.) It is thus assumed that the peptides used for the RT normalisation are easy to spot for the algorithm (since they have to be searched over the whole retention time space). Using noise-prone transitions or low-intensity transitions is thus not advisable for the retention time normalisation peptides. Next, an outlier detection algorithm, such as RANSAC or Chauvenet's criterion may be used to remove outliers.[42] Then, a (linear) function is computed to transform the experimental retention time into the normalised retention time space of the assay library, making the library retention times applicable to the current run.[41]

10.2.3.2 *Chromatogram Extraction*

After RT normalisation, the assay coordinates containing the fragment ion *m*/*z* and retention times for each peptide are used to extract fragment ion chromatograms from the raw data. For each peptide, the appropriate

SWATH-map is chosen for extraction (*e.g.* the map with the corresponding precursor *m*/*z* of the peptide). Usually, a tolerance window in *m*/*z* and RT is applied (for example 50 ppm and 10 minutes) and extraction is performed by applying a convolution function to each spectrum in the RT window. Generally, the convolution function is a top-hat function centred at the fragment ion *m*/*z*, which has the effect of simply adding up all signal within a square window. The result of the transformation is recorded at each chromatographic time point, leading to an extracted ion chromatogram (XIC) in the fragment ion domain. Note that generally each peptide precursor has multiple fragment chromatograms, one for each fragment ion present in the assay library.[33]

10.2.3.3 Peak Group Scoring

Peak group scoring is the next step in the analysis where chromatographic peaks are identified, grouped together by precursor and scored.[39] The purpose of this step is to identify potential points of elution for each peptide in the chromatograms extracted in the previous step and provide information about the quality of each potential elution point (peak). This step is performed in two distinct steps:

10.2.3.3.1 Peak Picking. The aim of peak picking is to identify potential peak candidates (points of elution) for each peptide in the fragment XICs. This can be done in multiple ways, the simplest one is to first identify peaks in each fragment ion chromatogram independently and then group the individual peaks at the peptide level to obtain peptide peaks spanning multiple chromatograms (termed "peak groups" from here on). Peak picking on the one-dimensional XIC data may be performed by initial smoothing and identification of maxima in the smoothed data. Alternatively, the XICs may be aggregated first using an appropriate function (such as a correlation score with the expected intensities) and peak picking may be performed on the aggregated trace.[43] In either case, the result of this step is a single list of peak candidates consisting of a retention time and potentially a start and end point of the peak.[39]

10.2.3.3.2 Peak Scoring. The algorithm next operates on the peak group candidates found in the previous step and computes a set of scores for each candidate. While no software tool uses the same set of scores, the commonly used scores can be classified in three groups: (i) chromatogram-based scores which operate on the XICs alone, mostly taking cross-correlation and shape of the traces into account, (ii) library-based scores which compute the agreement of the peak candidate with the assay library in terms of retention time and fragment ion intensity and (iii) spectrum-based scores which rely on the full high resolution fragment ion spectrum recorded at the peak apex, computing *m*/*z* deviation and agreement with the expected isotopic pattern.[39] Additional scores based on the MS1 signal within the peak boundaries and

scoring schemes relying on statistical models have recently been proposed as well.[44,45] In SRM, usually an additional score is generated which computes the signal correspondence to an isotopically-labelled spike-in standard.[46] However, in SWATH-MS often no spiked-in standard is available and therefore this score may only be available for certain datasets.

10.2.3.4 Peak Quantification

At this point, the peak candidates are also quantified. This is usually done by integrating the area under the chromatographic signal, but other metrics such as apex intensities may alternatively be reported.[39] More sophisticated quantification methods may also attempt to remove background signal or interferences (noise signals that co-elute with the target signal). For example, Teleman *et al.*[44] and Keller *et al.*[45] both suggested using the relative fragment ion intensities from the assay library to identify and remove interfering signal and report a corrected quantification value.

10.2.3.5 Error Rate Estimation

In this step, the previously computed individual scores for each peak group are combined into a single discriminant score and a global error rate estimation is performed on the result. Several statistical and machine-learning techniques are employed to determine how to combine the individual scores in an optimal fashion such that all high-quality peaks obtain a high score while low-quality peaks are assigned a low score. Generally, a set of "decoy assays" is generated and added to the assay library at the beginning of the analysis. Decoys are generated by perturbing the input assays (for example, shuffling the peptide sequence or moving fragment m/z by a random number).[46,47] The score distribution of these decoy assays can then be used to perform semi-supervised learning and obtain a final discriminant score as first suggested by Käll *et al.*[48] for shotgun proteomics data, and implemented for targeted proteomics data by Reiter *et al.*[46] A Bayesian approach similar to the one employed in PeptideProphet[49] for shotgun proteomics data can then be used for posterior error probability (PEP) estimation and computation of q-values[50] (see also Chapter 4, and Section 2.4). Specifically, the *mProphet* algorithm first allowed such automated analysis of targeted proteomics data employing linear discriminant analysis (LDA) to separate true and false peaks.[46] Since then, the algorithm has also been implemented in other programming languages and software tools.[44,51] Additional information about these concepts can be found in Section 10.2.4.

10.2.3.6 Alignment

Most currently published algorithms for analysis of targeted proteomics data or for targeted analysis of DIA data operate on a single dataset at a time and do not take information from multiple runs into account. However,

performing a post-analysis integration and consolidation step is crucial in order to obtain consistent and accurate proteomic data matrices. Otherwise, the quantified protein values may be inconsistent across multiple MS runs or may result in missing values in certain experimental conditions. This makes any downstream analysis where dozens to hundreds of protein samples have to be meaningfully compared, such as case-control studies (*e.g.* biomarker studies, perturbation experiments or affinity purifications) or time-course series highly challenging.

There are multiple advantages of an experiment-wide approach to identification and error control. It could boost identification confidence if other, related peak groups could be compared to the peak group at hand to decide whether it represents a true signal or a noise signal. This is based on the fact that it is much less likely to observe a noise signal consistently across many different samples than a true signal belonging to the target peptide.

Such a strategy could also help to resolve ambiguous identifications as in some cases, two or more suitable peak group candidates emerge from the scoring previously described and it is unclear which corresponds to the target peptide. This may be due to some co-elution of interference signal in one of the runs or increased noise which may not be present in other runs. The signal in other runs with different noise profiles could be used to resolve these ambiguous cases and select the appropriate peak group. Additionally, such an approach may also help to remove wrong identifications, as certain configurations are highly implausible (*e.g.* a peak that is substantially different in one run compared to the rest of the experiment). Finally, using cross-run alignment methods may help to increase the completeness of the data matrix as peaks with lower confidence could now be "rescued" if they are consistent with more confident peaks in other runs.

So far, no multi-level integration algorithm specific for targeted proteomics has been described in the literature. In addition, multiple statistical methods have been applied to large datasets which perform experiment-wide analysis. The *mProphet* algorithm, for example, can be run on the complete dataset instead of running it on a sample-by-sample basis. Also, more recently the original *iProphet* algorithm[52] has been modified by Keller *et al.*,[45] to also support SWATH-MS data, which allows integrative analysis on a statistical level.

10.2.4 FDR

Currently, the FDR, or false discovery rate, is the tool of choice to ensure the trustworthiness of a dataset. Its extensive use makes it a good parameter to compare results across research groups. A significant amount of research in statistics deals with measuring the certainty of scientific statements, as has already been discussed in Chapter 4. In the following discussion, we will thus focus on questions related to statistical evaluation of DIA data. In DIA data analysis, we can consider each assignment of mass spectrometric signal to a specific peptide is associated with a certain error and in the following, we will discuss how this error can be estimated in DIA and the total number of false positives reported in the result (the FDR) can be controlled.

The dominant approach for statistical evaluation in DIA consists of calculating the probability that a given signal found in the data was generated by random chance and not by the analyte in question. This is usually done by comparing the signal to so-called "decoy" signals which define the null hypothesis of the experiment, and which need to be chosen with great care. There are several questions we can try to answer by interrogating the MS data matrix generated in a DIA experiment: is this group of signals we are looking at the product of a fortuitous event? Is it a peptide signal? Or, more specifically, does this group of signals indicate the presence of a definite peptide (a defined sequence)? How these decoys are defined is going to define which of these questions can be answered. Starting from the most ambiguous of these questions, we can generate decoys by extracting signals from the data matrix in shifted or randomly chosen coordinates (compared to the target signals). It is important to notice that this decoy definition does not include any factor related to a peptide sequence, and thus it does not answer if signal groups rejecting the null hypothesis defined by these decoys are peptides or any other kind of analyte. The third question, ("Does this group of signals indicate the presence of a defined sequence?") may be answered by including particular characteristics of the sequence in decoys, like *m*/*z* coordinate combinations that may only be attributed to a particular sequence.

The case of untargeted analyses of DIA data can be treated similarly to DDA analysis, since pseudo-spectra generated can be searched in conventional DDA experiments, and therefore decoys are usually defined as reversed, pseudo-reversed or scrambled sequences of protein sequences. The strengths and weaknesses of these decoy generation methods have been largely discussed in the literature.[53,54] On the other hand, DIA targeted analyses are very similar to SRM experiments and related statistical methods as in SRM data analysis can be used.[46] Identification inference can be seen as a two step process: first, the interrogated coordinates (defined at the assay library) should define univocally the sequences we aim to identify. This defines that such signal combination can only come from a defined peptide sequence. Second, we want to ensure that the signal is not just a random event caused by noise signals. Then we may use random combinations of *m*/*z* and retention time coordinates searched across the data matrix, taking into consideration the number of elements of each combination, which should be equivalent to the target peptides in our library, *i.e.* one is less likely to find by chance six randomly chosen *m*/*z* coordinates at the same retention time than only two.

Comparing untargeted DIA *vs.* DDA experiments, and considering the different ways to calculate the FDR, it is interesting to note that one critical issue for calculating the FDR in DDA is usually the size of the precursor window used, as narrow windows lead to a small number of sequence candidates.[54] Using large *m*/*z* precursor windows for searching, in the order of 800 ppm or wider, solves the problem of obtaining a reasonable number of sequence candidates, and the resulting false positives with a precursor mass far from the experimental mass can be filtered out in a second step;[55] while wide precursor windows are not commonly used in DDA, for DIA experiments these (and larger) sizes are the norm, hence the number of sequence candidates is

also large enough to provide reliable statistics. Consequently, as the number of sequence candidates is not a substantial issue in DIA experiments, the decision about how the FDR should be calculated is limited to other criteria, such as the FDR formula, decoy database generation and concatenated *versus* separated search.[56] There are still issues about the scoring functions, as some search engines use biased scores (for example, SEQUEST's XCorr tends to be higher for longer peptides) requiring a score normalisation.[57] A good normalisation for these cases consists of using probabilistic scores, such as *p*-values.

The different approaches used to calculate the FDR exposes the uncertainty of a parameter that is used, in turn, to calculate the uncertainty of other results. Sometimes the researchers hesitate about using quantities as different as 1% or 2% to present their results. In fact, taking into account the big picture, *p*-values and *q*-values are themselves random variables, and spending time looking for high precision may distract from the main goal, which is having an approximate figure to be able to compare the different data obtained. An alternative to setting specific FDR thresholds could use a local FDR strategy, computing the number of true positives and false positives at each score threshold and selecting an appropriate cutoff, for example when the number of true positives outnumber the false positives.

10.2.5 Results and Formats

Proteomics mass spectrometry acquisitions, and particularly data independent acquisitions, produce massive output files. A DIA raw file typically occupies several gigabytes of disk space (typically triple the space needed for an equivalent DDA analysis). These massive files cause important challenges in the field: file format accessibility, file storage, and file input–output condition most of the current analysis pipelines. As explained in Chapter 11, the HUPO–PSI initiative[58,59] continuously works on the development of open file formats for the information exchange among the proteomics community. The vast majority of these formats are based on extensible markup language (XML), a human readable file format that permits tree data organisation. However, XML does not favour good input–output pipelines. It is size inefficient, and its tree organisation is counterproductive for some tasks. In the case of DIA, it is relevant to have an adequate method to access *m*/*z* values across retention time. There are several ways to circumvent this problem. You can improve the accession to raw files in standard XML formats by mixing conveniently the use of random and sequential access, as some libraries already do.[60] In a different approach, you can develop your own intermediary format, ensuring read–write efficiency. This also has its own obstacles: you need to write APIs able to convert from most of proprietary vendors formats, or force users to perform multiple format conversions. Currently there are also some other file format alternatives for raw files, which are more efficient in storage size and in read–write access. One of the most promising is mz5,[61] an MS raw file format based on HDF5. HDF5 files offer several advantages

over XML files that are relevant to computational proteomics. Its internal file structure integrates multidimensional array datasets that can be interrogated as matrices. The best example in the case of DIA is that MS data can be organised in a three-dimensional array dataset with the following dimensions: *m*/*z*, retention time, and intensity. This arrangement allows a fast read access of the intensity values that correspond to an *m*/*z* value range across the retention time axis. In practice the task of monitoring the intensities of a fragment ion mass of *e.g.* 500 *m*/*z* with a given *m*/*z* tolerance of *e.g.* 5 parts per million across a predicted retention time range of *e.g.* 25 to 28 minutes is performed by slicing in two dimensions (*m*/*z* and retention time) a three-dimensional array. This extraction process in slices is even more effective when raw data is previously normalised across dimensions. Specifically, the fragment ion monitoring of the previous example would be performed faster if the *m*/*z* value of the fragment ion of interest is constant across the retention time axis.

The best format for results in many cases depends strongly on the needs of the researcher and their context. Not just in DIA, but for proteomics in general, there is not an established best format, the main reason being the fact that the typical file size and the standard information needed for an experiment is still increasing year by year. This is a situation that could change in the next few years, with further implementation of standard formats in mass spectrometry laboratories. Developing of efficient standard formats, which fit to the many MS acquisition flavours is yet a major work field in computational proteomics, as formats are subjected to continuous improvement.

10.3 Challenges

For DIA data analysis, and specifically for targeted analysis, many computational challenges remain. The methods in the field are just emerging and there are many opportunities for improvement of the existing tools in terms of performance and usability. Each of the steps described for the targeted analysis of DIA data provides room for improvement and presents several unsolved questions.

Specifically, retention time normalisation is performed on a linear scale in most algorithms even though it has been shown that considering non-linear effects is important in LC-MS/MS-based proteomics.[62] Improving the retention time normalisation step could help to narrow the search window for candidate peak groups and thus improve algorithm performance. Additionally, current software tools do not take into account the theoretical order of the peptides eluting, which might be used for discarding—or penalising—peptide detections, which violate the expected elution order. Similarly, a normalisation step for the *m*/*z* domain (mass calibration) as sometimes performed for shotgun proteomics could help to improve specificity.

Selection of the fragment masses from the library is a critical task in targeted analyses. It is common to use the top *n* most intense transitions of the library (where *n* may vary from 3 to ~10). In many cases, these top *n*

transitions do not ensure a univocal characterisation of a peptide, or in a peak group where one of the transitions is not detected may cause an ambiguity. The use of all available transitions is computationally expensive, but may provide increases in specificity especially if several, closely related analytes are present in the biological sample (as is often the case with modified proteins).

Currently, most tools extract chromatograms using a simple top-hat filter where all signal within a square window is summed up. However, using more sophisticated methods that model peaks in m/z dimension and attempt peak deconvolution may increase specificity and may be able to distinguish fragment ions that are close in m/z but are not completely baseline-separated.

The peak group scoring is currently at the heart of most targeted analysis workflows and multiple, orthogonal scores have been proposed for this step.[44–46] However, so far no unbiased investigation has evaluated the contribution of the different scores or attempted to create a synthesis of multiple scoring approaches in order to improve peak picking performance. Similarly, improvements in peak group quantification removing noise, interferences and background could lead to more precise measurements and thus improvements in the biological conclusions.

However, in both main analysis approaches (targeted and untargeted), some of the main challenges lie in the currently under-explored areas of error rate computation and cross-run alignment. Most currently available tools in these areas were either developed for low-throughput targeted proteomics or for shotgun proteomics. Thus, in most cases the algorithms need to be adapted to the needs of high throughput technologies.

References

1. A. Doerr, *Nat. Methods*, 2014, **12**, 35.
2. R. Aebersold and M. Mann, *Nature*, 2003, **422**, 198–207.
3. S. A. Gerber, J. Rush, O. Stemman, M. W. Kirschner and S. P. Gygi, *Proc. Natl. Acad. Sci. U. S. A.*, 2003, **100**, 6940–6945.
4. V. Lange, P. Picotti, B. Domon and R. Aebersold, *Mol. Syst. Biol.*, 2008, **4**, 222.
5. A. Wolf-Yadlin, S. Hautaniemi, D. A. Lauffenburger and F. M. White, *Proc. Natl. Acad. Sci. U. S. A.*, 2007, **104**, 5860–5865.
6. P. Picotti, B. Bodenmiller, L. N. Mueller, B. Domon and R. Aebersold, *Cell*, 2009, **138**, 795–806.
7. J. Sherman, M. J. McKay, K. Ashman and M. P. Molloy, *Mol. Cell. Proteomics*, 2009, **8**, 2051–2062.
8. H. Rost, L. Malmstrom and R. Aebersold, *Mol. Cell. Proteomics*, 2012, **11**, 540–549.
9. B. Domon and R. Aebersold, *Nat. Biotechnol.*, 2010, **28**, 710–721.
10. A. Michalski, J. Cox and M. Mann, *J. Proteome Res.*, 2011, **10**, 1785–1793.

11. S. Purvine, J.-T. Eppel, E. C. Yi and D. R. Goodlett, *Proteomics*, 2003, **3**, 847–850.
12. J. D. Venable, M.-Q. Dong, J. Wohlschlegel, A. Dillin and J. R. Yates, *Nat. Methods*, 2004, **1**, 39–45.
13. R. S. Plumb, K. A. Johnson, P. Rainville, B. W. Smith, I. D. Wilson, J. M. Castro-Perez and J. K. Nicholson, *Rapid Commun. Mass Spectrom.*, 2006, **20**, 1989–1994.
14. A. Panchaud, A. Scherl, S. A. Shaffer, P. D. von Haller, H. D. Kulasekara, S. I. Miller and D. R. Goodlett, *Anal. Chem.*, 2009, **81**, 6481–6488.
15. A. Panchaud, S. Jung, S. A. Shaffer, J. D. Aitchison and D. R. Goodlett, *Anal. Chem.*, 2011, **83**, 2250–2257.
16. M. Bern, G. Finney, M. R. Hoopmann, G. Merrihew, M. J. Toth and M. J. MacCoss, *Anal. Chem.*, 2010, **82**, 833–841.
17. J. W. H. Wong, A. B. Schwahn and K. M. Downard, *BMC Bioinf.*, 2009, **10**, 244.
18. P. C. Carvalho, X. Han, T. Xu, D. Cociorva, M. da G. Carvalho, V. C. Barbosa and J. R. Yates 3rd, *Bioinformatics*, 2010, **26**, 847–848.
19. S. J. Geromanos, J. P. C. Vissers, J. C. Silva, C. A. Dorschel, G.-Z. Li, M. V. Gorenstein, R. H. Bateman and J. I. Langridge, *Proteomics*, 2009, **9**, 1683–1695.
20. G.-Z. Li, J. P. C. Vissers, J. C. Silva, D. Golick, M. V. Gorenstein and S. J. Geromanos, *Proteomics*, 2009, **9**, 1696–1719.
21. K. Blackburn, F. Mbeunkui, S. K. Mitra, T. Mentzel and M. B. Goshe, *J. Proteome Res.*, 2010, **9**, 3621–3637.
22. X. Huang, M. Liu, M. J. Nold, C. Tian, K. Fu, J. Zheng, S. J. Geromanos and S.-J. Ding, *Anal. Chem.*, 2011, **83**, 6971–6979.
23. T. Geiger, J. Cox and M. Mann, *Mol. Cell. Proteomics*, 2010, **9**, 2252–2261.
24. K. P. Law and Y. P. Lim, *Expert Rev. Proteomics*, 2013, **10**, 551–566.
25. J. D. Chapman, D. R. Goodlett and C. D. Masselon, *Mass Spectrom. Rev.*, 2014, **33**, 452–470.
26. E. Sabidó, N. Selevsek and R. Aebersold, *Curr. Opin. Biotechnol.*, 2012, **23**, 591–597.
27. R. Bruderer, O. M. Bernhardt, T. Gandhi, S. M. Miladinović, L.-Y. Cheng, S. Messner, T. Ehrenberger, V. Zanotelli, Y. Butscheid, C. Escher, O. Vitek, O. Rinner and L. Reiter, *Mol. Cell. Proteomics*, 2015, **14**, 1400–1410.
28. P. Picotti, M. Clément-Ziza, H. Lam, D. S. Campbell, A. Schmidt, E. W. Deutsch, H. Röst, Z. Sun, O. Rinner, L. Reiter, Q. Shen, J. J. Michaelson, A. Frei, S. Alberti, U. Kusebauch, B. Wollscheid, R. L. Moritz, A. Beyer and R. Aebersold, *Nature*, 2013, **494**, 266–270.
29. A. M. Edwards, I. Ruth, G. D. Bader, S. V. Frye, T. M. Willson and F. H. Yu, *Nature*, 2011, **470**, 163–165.
30. A. C. Peterson, J. D. Russell, D. J. Bailey, M. S. Westphall and J. J. Coon, *Mol. Cell. Proteomics*, 2012, **11**, 1475–1488.
31. J. C. Silva, M. V. Gorenstein, G.-Z. Li, J. P. C. Vissers and S. J. Geromanos, *Mol. Cell. Proteomics*, 2006, **5**, 144–156.
32. U. Distler, J. Kuharev, P. Navarro, Y. Levin, H. Schild and S. Tenzer, *Nat. Methods*, 2014, **11**, 167–170.

33. L. C. Gillet, P. Navarro, S. Tate, H. Röst, N. Selevsek, L. Reiter, R. Bonner and R. Aebersold, *Mol. Cell. Proteomics*, 2012, **11**, O111.016717.
34. J. D. Egertson, A. Kuehn, G. E. Merrihew, N. W. Bateman, B. X. MacLean, Y. S. Ting, J. D. Canterbury, D. M. Marsh, M. Kellmann, V. Zabrouskov, C. C. Wu and M. J. MacCoss, *Nat. Methods*, 2013, **10**, 744–746.
35. O. V. Krokhin and V. Spicer, UNIT 13.14 Predicting Peptide Retention Time for Proteomics, in *Current Protocols in Bioinformatics*, John Wiley and Sons, Inc., 2010, pp. 13.14.1–13.14.15.
36. G. F. Verbeck, B. T. Ruotolo, H. A. Sawyer, K. J. Gillig and D. H. Russell, *J. Biomol. Tech.*, 2002, **13**, 56–61.
37. N. Zhang, X.-J. Li, M. Ye, S. Pan, B. Schwikowski and R. Aebersold, *Proteomics*, 2005, **5**, 4096–4106.
38. C.-C. Tsou, D. Avtonomov, B. Larsen, M. Tucholska, H. Choi, A.-C. Gingras and A. I. Nesvizhskii, *Nat. Methods*, 2015, **12**, 258–264.
39. H. L. Röst, G. Rosenberger, P. Navarro, L. Gillet, S. M. Miladinović, O. T. Schubert, W. Wolski, B. C. Collins, J. Malmström, L. Malmström and R. Aebersold, *Nat. Biotechnol.*, 2014, **32**, 219–223.
40. O. T. Schubert, L. C. Gillet, B. C. Collins, P. Navarro, G. Rosenberger, W. E. Wolski, H. Lam, D. Amodei, P. Mallick, B. MacLean and R. Aebersold, *Nat. Protoc.*, 2015, **10**, 426–441.
41. C. Escher, L. Reiter, B. MacLean, R. Ossola, F. Herzog, J. Chilton, M. J. MacCoss and O. Rinner, *Proteomics*, 2012, **12**, 1111–1121.
42. S. J. Parker, H. Rost, G. Rosenberger, B. C. Collins, L. Malmström, D. Amodei, V. Venkatraman, K. Raedschelders, J. E. Van Eyk and R. Aebersold, *Mol. Cell. Proteomics*, 2015, **14**, 2800–2813.
43. C. R. Weisbrod, J. K. Eng, M. R. Hoopmann, T. Baker and J. E. Bruce, *J. Proteome Res.*, 2012, **11**, 1621–1632.
44. J. Teleman, H. L. Röst, G. Rosenberger, U. Schmitt, L. Malmström, J. Malmström and F. Levander, *Bioinformatics*, 2015, **31**, 555–562.
45. A. Keller, S. L. Bader, D. Shteynberg, L. Hood and R. L. Moritz, *Mol. Cell. Proteomics*, 2015, **14**, 1411–1418.
46. L. Reiter, O. Rinner, P. Picotti, R. Hüttenhain, M. Beck, M.-Y. Brusniak, M. O. Hengartner and R. Aebersold, *Nat. Methods*, 2011, **8**, 430–435.
47. H. Lam, E. W. Deutsch and R. Aebersold, *J. Proteome Res.*, 2010, **9**, 605–610.
48. L. Käll, J. D. Canterbury, J. Weston, W. S. Noble and M. J. MacCoss, *Nat. Methods*, 2007, **4**, 923–925.
49. A. Keller, A. I. Nesvizhskii, E. Kolker and R. Aebersold, *Anal. Chem.*, 2002, **74**, 5383–5392.
50. J. D. Storey and R. Tibshirani, *Proc. Natl. Acad. Sci. U. S. A.*, 2003, **100**, 9440–9445.
51. B. MacLean, D. M. Tomazela, N. Shulman, M. Chambers, G. L. Finney, B. Frewen, R. Kern, D. L. Tabb, D. C. Liebler and M. J. MacCoss, *Bioinformatics*, 2010, **26**, 966–968.
52. D. Shteynberg, E. W. Deutsch, H. Lam, J. K. Eng, Z. Sun, N. Tasman, L. Mendoza, R. L. Moritz, R. Aebersold and A. I. Nesvizhskii, *Mol. Cell. Proteomics*, 2011, **10**, M111.007690.

53. J. E. Elias and S. P. Gygi, *Nat. Methods*, 2007, **4**, 207–214.
54. K. Jeong, S. Kim and N. Bandeira, *BMC Bioinf.*, 2012, **16**(suppl. 13), S2.
55. E. Bonzon-Kulichenko, F. Garcia-Marques, M. Trevisan-Herraz and J. Vázquez, *J. Proteome Res.*, 2015, **14**, 700–710.
56. A. R. Jones, J. A. Siepen, S. J. Hubbard and N. W. Paton, *Proteomics*, 2009, **9**, 1220–1229.
57. S. Martinez-Bartolome, P. Navarro, F. Martin-Maroto, D. Lopez-Ferrer, A. Ramos-Fernandez, M. Villar, J. P. Garcia-Ruiz and J. Vazquez, *Mol. Cell. Proteomics*, 2008, **7**, 1135–1145.
58. S. Orchard, *Biochim. Biophys. Acta*, 2014, **1844**, 82–87.
59. S. Orchard, P.-A. Binz, C. Borchers, M. K. Gilson, A. R. Jones, G. Nicola, J. A. Vizcaino, E. W. Deutsch and H. Hermjakob, *Proteomics*, 2012, **12**, 2767–2772.
60. H. L. Röst, U. Schmitt, R. Aebersold and L. Malmström, *PLoS One*, 2015, **10**, e0125108.
61. M. Wilhelm, M. Kirchner, J. A. J. Steen and H. Steen, *Mol. Cell. Proteomics*, 2012, **11**, O111.011379.
62. H. Weisser, S. Nahnsen, J. Grossmann, L. Nilse, A. Quandt, H. Brauer, M. Sturm, E. Kenar, O. Kohlbacher, R. Aebersold and L. Malmström, *J. Proteome Res.*, 2013, **12**, 1628–1644.

Section III

Open Source Software Environments for Proteome Informatics

CHAPTER 11

Data Formats of the Proteomics Standards Initiative

JUAN ANTONIO VIZCAÍNO*[a], SIMON PERKINS[b], ANDREW R. JONES[b] AND ERIC W. DEUTSCH[c]

[a]European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Trust Genome Campus, Hinxton, Cambridge, CB10 1SD, UK; [b]Institute of Integrative Biology, University of Liverpool, UK; [c]Institute for Systems Biology, 401 Terry Ave N, Seattle, WA 98109, USA
*E-mail: juan@ebi.ac.uk

11.1 Introduction

There is a huge variety of academic and commercial software that is routinely used for the analysis, visualisation, and storage of proteomics data. Mass spectrometers routinely output the mass spectrometry (MS) raw data in vendor-specific proprietary binary formats. In addition, every analysis tool or data analysis pipeline produces their own output data formats for reporting the identification and quantification results. Thus, there is a wide variety of data formats, each of them inherently complex, with its own peculiarities. This can hinder the further development of the field due to the effort that must be invested in supporting multiple heterogeneous data formats – particularly computational groups developing new analysis or

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

statistical packages or databases. See elsewhere[1] for a high-level review of all formats used in conjunction with MS-based proteomics.

Indeed, it is broadly recognised that common data standards are a crucial element of advancing a research field.[2,3] Among the many existing benefits, they can enhance interoperability between software tools, and increase usability of tools across different instrument vendors and computer operating systems. Additionally, the formats can also increase the ease of sharing, reusing data produced by others, and depositing data in public data repositories.[4] Furthermore, the adoption of standard data formats can facilitate the validation, reproducibility and comparability of the experimental results produced by different groups, potentially using different instrument and software platforms. Finally, the existence of broadly used data standards allows programmer resources to be concentrated on algorithm development rather than infrastructure to support a fleet of different data formats.

The HUPO (Human Proteome Organisation) PSI (Proteomics Standards Initiative, http://www.psidev.info) was formed in 2002 to coordinate the efforts of developing data standards in the field of proteomics. The PSI, with broad participation by academics and representatives from industry and journals, was originally tasked with developing open and common data standards which could be used in the different stages of the typical proteomics analysis workflow. See Deutsch *et al.*,[3] for a high-level overview of the history, activities, and products of the PSI. In addition to the standard data formats themselves, the PSI develops and maintains two other types of products:

(i) Controlled vocabularies (CVs), which are needed for providing values or descriptions within the standard formats. A CV is essentially a list of standardly agreed terms, accompanied by a definition, and sometimes a data type and unit which may accompany the term. The main PSI CVs[5] are the PSI-MS[6] (for MS and proteomics informatics related information), PSI-MOD (for protein modifications)[7] and PSI-MI (for molecular interaction information). These CVs are available in the widely used OBO format (http://www.geneontology.org/GO.format.shtml) and are usually updated whenever new terms are needed.

(ii) Minimum information guidelines, called MIAPE (Minimum Information About a Proteomics Experiment) documents.[8] These guidelines aim to show the information that needs to be reported by researchers to enable critical analysis or even reproduction of a given experiment. The main existing MIAPE guidelines are MIAPE-MS[9] (for MS data), MIAPE MSI (for peptide/protein identifications), MIAPE-Quant[10] (for quantification experiments) and MIMIx[11] (for molecular interaction data), although there are others (See http://www.psidev.info/miape for more information.).

It is important to highlight that every data format or MIAPE guidelines document produced by the PSI undergoes a thorough review process, called the PSI document process,[12] which is analogous to the typical manuscript review in scientific journals. At the end of the process, usually after a few

iterations involving the authors, reviewers, and the editor, the specification is approved and it becomes an official standard of the PSI. In addition, every time there is a substantial update in the documents, the standards need to go through the same process.

Proteomics repositories heavily rely on the existence of data standards. The ProteomeXchange Consortium,[13] comprising some of the main resources in the field, was set up to overcome the existing challenges in MS proteomics public data sharing and dissemination by implementing standard submission and dissemination pipelines. At present the consortium includes the PRoteomics IDEntifications (PRIDE) database,[14] PeptideAtlas[15] and the related resource PeptideAtlas SRM Experiment Library (PASSEL), and MassIVE (http://massive.ucsd.edu/). All ProteomeXchange resources promote the use and implementation of PSI standards.[16]

In this chapter, we focus on the PSI data standards related to MS, namely mzML, mzIdentML, mzQuantML, mzTab and TraML (For a summary overview, see Table 11.1.). Most of them are based in XML (Extensible Markup Language) schemas. Only mzTab was designed as a tab-delimited file. We will here outline the main characteristics of each format, the current implementations, and describe any current efforts to update each standard in order to support new use cases. In addition, we will also mention some additional data formats that, although not formal PSI data standards, were inspired by PSI standards and have become popular in a particular area. Finally, we will briefly outline other non-MS related PSI data standards.

11.2 mzML

11.2.1 Data Format

mzML is an XML-based format used for storing MS data (and associated metadata), which constitute the output from the mass spectrometers.[17] The current version (1.1.0) was released in 2009 (http://www.psidev.info/mzml) and retains the best attributes of two previously developed formats called mzData (XML format developed also by the PSI in its early days) and mzXML[18] developed by the Institute for Systems Biology (Seattle, USA). At the time, it was agreed that having two different formats was not beneficial for the field and therefore, the unified mzML format was developed.

mzML was originally designed as a flexible format able to cope with a variety of cases and adapt to new scenarios. This flexibility, like in the other PSI formats, is achieved mainly through the use of CVs (in this concrete case, mostly the PSI-MS CV). If new pieces of information need to be included in the format, the XML schema does not need to be changed to support new attributes. This was one of the main limitations of mzXML. Instead, new CV terms can be used to encode any novel information required. However, to avoid the formats becoming too flexible (the main issue of mzData), it was necessary to develop a "semantic validator", a software tool for checking formal usage of CV terms in a correct location in the file. The same semantic validator principles are used in the other XML-based PSI data standards.[19]

Table 11.1 Main characteristics of the mass spectrometry related HUPO PSI data standards.

	mzML[17]	mzIdentML[27]	mzQuantML[36]	mzTab[38]	TraML[42]
Types of information	Mass spectra and chromatograms	Peptide and protein identification data	Peptide, protein and small molecule quantification data	Peptide, protein and small molecule identification and quantification data	Transition lists for SRM, and inclusion lists for targeted MS/MS
Current stable version (March 2016)	1.1.0	1.1.1	1.0.1	1.0	1.0
Type of file	XML	XML	XML	Tab-delimited	XML
Reference for the specific MIAPE guidelines	MIAPE MS	MIAPE MSI	MIAPE Quant	Not applicable	Not developed yet
CV mainly used	PSI-MS	PSI-MS PSI-MOD/Unimod	PSI-MS PSI-MOD/Unimod	PSI-MS PSI-MOD/Unimod	PSI-MS PSI-MOD/Unimod
URL	http://www.psidev.info/mzml	http://www.psidev.info/mzidentml	http://www.psidev.info/mzquantml	https://github.com/HUPO-PSI/mzTab	http://www.psidev.info/traml

Therefore, each XML format has, in addition to its XML schema, a set of associated semantic rules.

The mzML format is designed to contain one MS run per file, including metadata about the spectra plus all the spectra themselves, either in centroided (peak list) or profile mode. Optionally, the file can also contain the corresponding chromatograms. At the top of the XML schema, there is space for some basic metadata: first the <cvList> element (< > denotes an element in an XML file), a common element in many PSI standards, contains information about all the CVs referenced in the file. Then, the <fileDescription> element contains information about the type of spectra. The following two elements are optional. First of all, the <referenceableParamGroupList> element contains a list of the groups of CV terms that are used frequently and may simply be defined once and referenced in the file thereafter. The <sampleList> element contains information about samples, that are again referenced throughout the file.

Next, the <instrumentConfiguration> element contains information about the instrument used in the MS run (in more than one configuration in the case of hybrid instruments). The <softwareList> and <dataProcessingList> elements provide the information related to data processing that may have occurred since the acquisition of the raw data. Finally in this section, an optional <acquisitionSettingsList> element can hold special input parameters to mass spectrometers such as inclusion lists.

These elements are followed in the schema by the actual spectra and optionally the chromatograms. Both spectra and chromatograms are represented in binary format encoded into base64 strings. Figure 11.1 contains an example of a tandem MS spectrum encoded in an mzML file. Finally, it should be noted that mzML was designed such that the main part of an mzML document does not contain an index, but that the document may be enclosed in a wrapper schema that includes an index. In this context an index is a lookup table of spectrum identifiers and scan numbers pointing to specific offsets within the file, enabling software to seek directly to a location within a file. This was designed in this manner to enable fast retrieval of individual spectra.

11.2.2 Software Implementations

mzML is now widely used and its adoption keeps growing. There are multiple implementations, including a number of libraries, analysis and visualisation tools. A comprehensive list is available at http://www.psidev.info/mzml. A widely used set of tools are those implemented by the ProteoWizard project.[20] ProteoWizard is the reference implementation, written in C++, enabling the access and conversion of the raw proprietary formats coming from the main vendors into open formats such as mzML. Due to the fact that the vendor software libraries are developed for running in Microsoft Windows®, the main limitation is that the ProteoWizard msconvert tool must run on Windows to support vendor file format conversions, or potentially

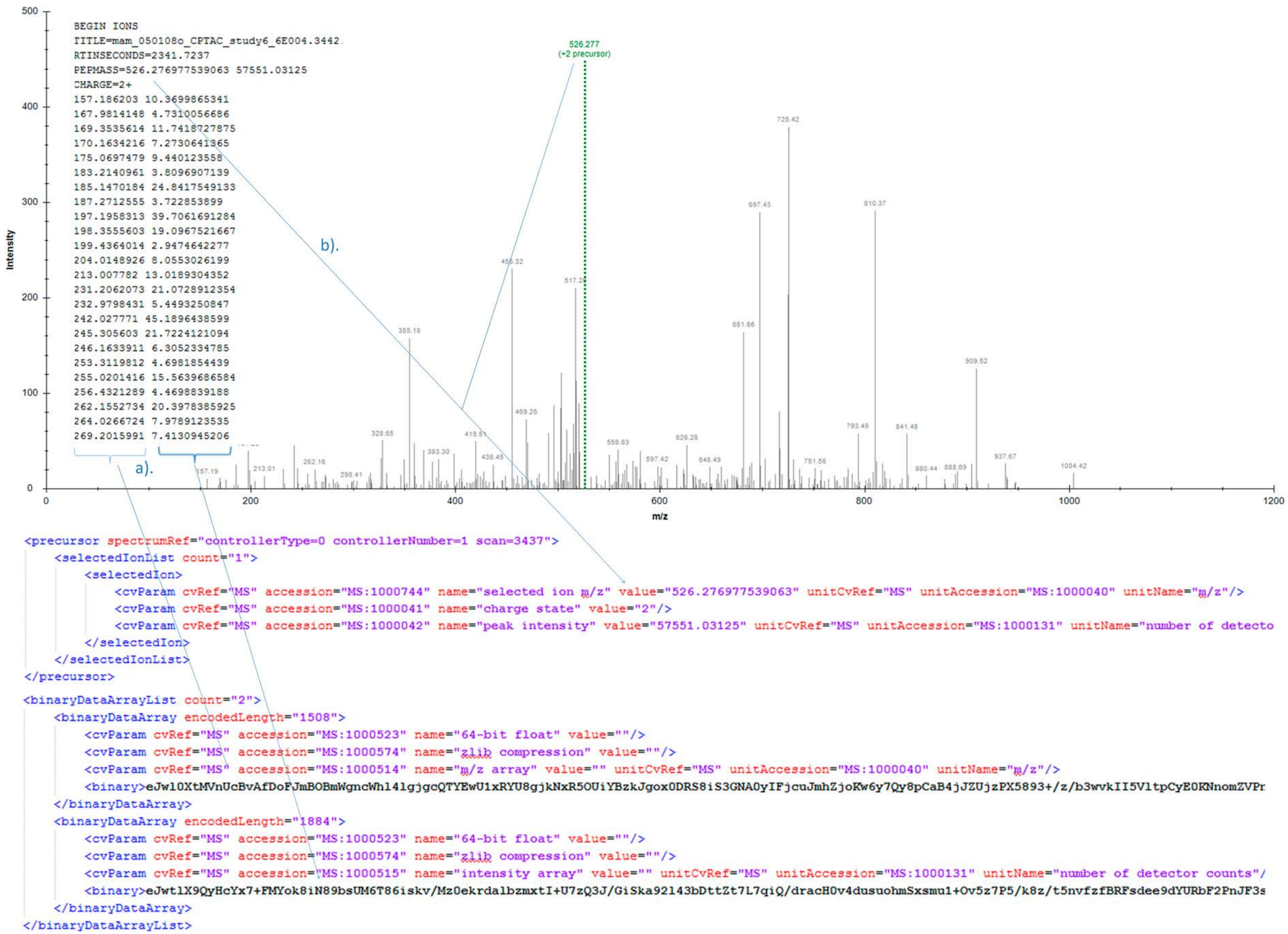

Figure 11.1 Tandem MS spectrum encoded in mzML. The list of fragment *m*/*z* values and their respective abundances (a) are encoded in mzML in separate 'BinaryDataArray' elements. The binary nature of these values is encoded using the base64 encoding scheme. Optionally the values may be compressed using one of the recommended data compression schemes (In this example the data have been compressed with the zlib library before base64 encoding.). Other aspects of a spectrum may be encoded. For example, as it can be seen in (b) the precursor (peptide) of the spectrum is also encoded in mzML.

Windows emulators (although these may not function in all scenarios as intended, see http://tools.proteomecenter.org/wiki/index.php?title=Msconvert_Wine). However, if conversion to mzML is performed, then only the conversion must happen on the Windows-based computer, but all downstream processing may occur under other operating systems.

In addition, most of the search engines and post-processing software support mzML as the input format for the search (*e.g.* X!Tandem, Mascot, MyriMatch, OpenMS, the Trans-Proteomic Pipeline, and many others). For software developers, there are open source libraries available in different programming languages such as Java (jmzml,[21] https://github.com/PRIDE-Utilities/jmzml), R (mzR package in BioConductor, http://bioconductor.org/packages/release/bioc/html/mzR.html, see Chapters 15 or 16) and Python (pymzML[22]). Finally, it should be noted that most proteomics repositories including all the members of ProteomeXchange support submission of MS data in the mzML format.

11.2.3 Current Work

Similarly to MS proteomics, the MS-based metabolomics field is also advancing fast. In parallel with the continuous development of the instrumentation and analysis approaches, it has also been acknowledged that there is the need to formalise data standards in the field. In this context, mzML is being logically adopted as the format of choice for the output of the mass spectrometers. Again, due to the flexible design of mzML, no schema changes were necessary to accommodate this extension, as it could be achieved using CV terms instead.

One of the known issues of the mzML format is that files can become very large, for instance when compared with the data proprietary “raw” files produced by the different instrument vendors, mainly due to the verbosity introduced by the use of XML tags. This issue is becoming more and more important since modern instruments produce increasingly larger files. For instance, the largest mzML files are produced at present by the Waters ion mobility instruments where each MS run can have an average size of around 20 GB. Consequently, different attempts have now started to produce smaller mzML files by compressing the MS information. One of such attempts is the development of a family of numerical compression algorithms called *MS-Numpress*.[23] It is expected that the next iteration of the format, version 1.2, will also support this set of algorithms for encoding the mass spectra in binary format, in addition to the already supported ones. Otherwise, no schema changes are expected.

11.2.4 Variations of mzML

Also in recent years, some variations of the mzML format have been developed outside the PSI umbrella. These formats have been driven to address specific use cases, mainly due to the limitations of storing and accessing

large amounts of data in mzML files (as in any XML format). First of all, the MS imaging field developed a format called imzML (http://www.imzml.org/).[24] The data are split in two files linked by a universally unique identifier. The experimental details are stored in an XML file based on the mzML schema, whereas the spectra are stored in a binary file in order to allow an efficient storage.

The format mz5[25] was designed to achieve smaller files and faster random access *via* reimplementation of mzML based on the HDF5 binary file format system, which is optimised for storage of complex numerical data – heavily used in astrophysics and related fields. Finally, the mzDB format,[26] based on an SQLite format, has been recently developed to enable an efficient extraction of the signals used to identify specific target peptides in the case of large datasets coming from MS/MS workflows and from the data independent acquisition (DIA) SWATH-MS approach. The PSI regularly evaluates potential alternatives to the XML-based formats, taking into account performance *versus* the need for a universal access to data. There are no short term plans to move away from an XML-based mzML, but alternatives will be considered on a longer time scale, as MS data volumes continue to grow, and the scalability of a pure XML format is unclear.

11.3 mzIdentML

11.3.1 Data Format

mzIdentML is an XML format developed for reporting the search parameters and results of peptide and protein identification data, derived from spectrum identification algorithms (*e.g.* search engines) (http://www.psidev.info/mzidentml).[27] One of the main aims of the format is to support the full trace of evidence in a typical shot-gun proteomics experiment, including scores or statistics associated with peptide-spectrum matches (PSMs), proteins inferred from those PSMs, and protein groups – due to ambiguity in the unique assignment of some peptides to proteins. The current version (1.1) was released in 2011.

mzIdentML does not support the reporting of quantitative information, which can be provided in other formats such as mzQuantML and mzTab (see next sections). In addition, mzIdentML does not contain the original spectra identified. Instead, the file contains references to mass spectra in external files. This is the same approach used in mzQuantML and mzTab, using an established mechanism that depends on the file format in which the searched mass spectra are stored (*e.g.* ideally mzML, but there is also support for other formats such as mzXML, raw proprietary formats, or peak list spectra files such as mgf, dta, dta, pkl or apl). The design decision was made to avoid redundant storage of information across different PSI formats (since mzML can fully handle peak lists), but requires some extra effort from mzIdentML implementers, as full support requires code for reading mzIdentML, as well as peak lists in alternative formats. As in the case of

mzML, mzIdentML makes heavy use of the PSI-MS CV. Additionally, for the reporting of protein modifications (natural and artifactual), both the PSI-MOD and Unimod (http://www.unimod.org/) CVs are supported. Analogously to mzML, mzIdentML was originally designed to include the search results coming from one MS run (but see Section 11.3.3 for planned changes to this).

It is acknowledged that the mzIdentML XML schema is complex and can be challenging to implement since it includes many internal cross-references, to avoid redundantly storing the same information multiple times in each file (such as peptide or protein sequences). Each file must contain one or more instances of the <SpectrumIdentificationList> element (the set of PSMs) and must contain zero or one <ProteinDetectionList> elements (the set of protein identifications inferred from the PSMs). This means it is valid to create mzIdentML files containing only PSMs without including the protein identifications, but not *vice versa*.

At the top of the file schema, general metadata can be reported. First, the <cvList> element, as in the case of mzML and other formats, contains information about all the CVs referenced in the file. Then <AnalysisSoftwareList> includes the software used (*e.g.* the search engine that generated the file) and the optional element <AnalysisSampleCollection> can provide information about the biological samples used. The following highlighted element in the schema, <SequenceCollection> includes all the peptide (<Peptide> elements) and protein sequences from the search database (<DBSequence> elements) reported (and crucially, the correspondence between peptides and proteins). Next, the element <AnalysisCollection> includes the list of inputs and outputs of the analysis, where the protocols are applied. The <AnalysisProtocolCollection> contains all the parameters used in the analysis, both for the spectra identification (<SpectrumIdentificationProtocol>) and protein detection (<ProteinDetectionProtocol>).

The last element in the schema is <DataCollection>, which first lists the database, spectra searched and the input file converted to mzIdentML (within the element <Inputs>). Then, it contains a second element called <AnalysisData>, by far the most data-rich section of mzIdentML files. Within <AnalysisData> there are again two elements. The first one is called <SpectrumIdentificationList>. The core element in the list is <SpectrumIdentificationResult> – representing all the PSMs found from a single spectrum searched. Each <SpectrumIdentificationResult> references the spectrum from which identifications have been made in an external file. Each PSM within the <SpectrumIdentificationResult> is captured in an ordered list of elements called <SpectrumIdentificationItem> (Figure 11.2). Scores or statistical values for the PSM are described by CV terms, specific to each search engine, as well as (optionally) more general terms such as local or global false discovery rate, or posterior error probability (PEP).

While most implementations of mzIdentML contain the results of sequence database-based searches, it is straightforward to include the results of spectral library searches as well. All spectral library algorithms export scores and/or statistical values for peptide identifications, which can

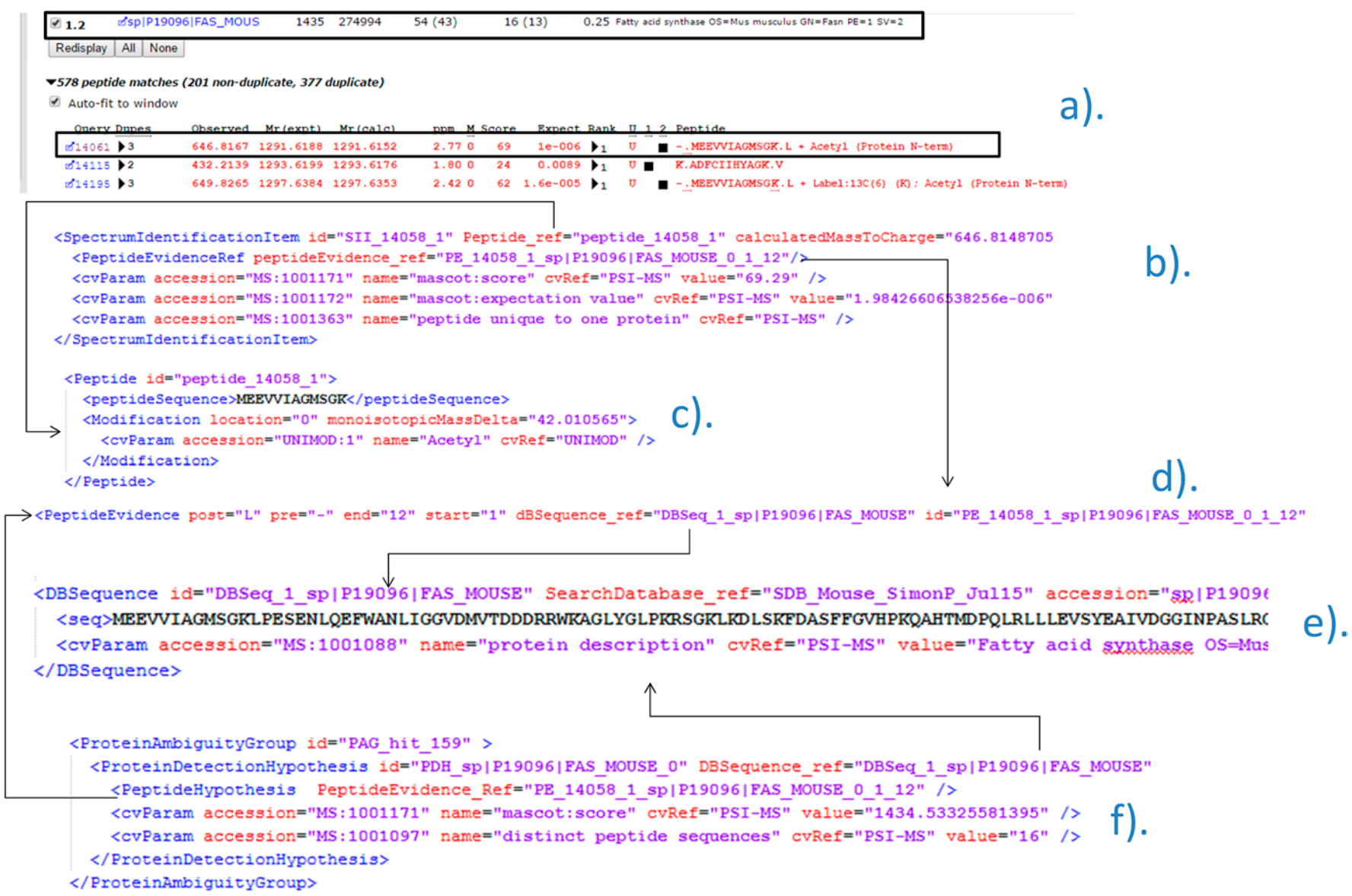

Figure 11.2 Excerpts from an mzIdentML file generated from Mascot. (a) The PSM itself is in the black box, and the parent proteins can be seen for context. The match between a given spectrum and a candidate peptide (*i.e.* a PSM) can be encoded in a 'SpectrumIdentificationItem' element (b), which can describe the quality of the match. The 'SpectrumIdentificationItem' element references a reusable 'Peptide' (c) element containing the peptide sequence and modifications. A 'PeptideEvidence' (d) element describes the relationship between a given peptide sequence and the parent protein(s) in which it can be found. The 'DBSequence' element (e) represents a protein sequence within the searched database and can be referenced from various places in the file. The 'ProteinAmbiguityGroup' element (f) groups together candidate protein identifications ('ProteinDetectionHypothesis') in which there are shared peptides/spectra in common. Various scores can be added at the level of the group or each individual protein (Here a Mascot score is shown for a single protein.).

be included under <SpectrumIdentificationItem>. Any metadata associated with the library entry can be added to the referenced <Peptide> element as additional <cvParam> or <userParam> elements.

For the reporting of protein identifications (which is optional), again the schema enables that the ambiguity derived from the protein inference is communicated. Namely, also within <AnalysisData>, the protein identifications are stored under <ProteinDetectionList>, where each <ProteinDetectionHypothesis> element represents a putative identification of one protein accession from the search database (Figure 11.2). A <ProteinAmbiguityGroup> sits above in the hierarchy, acting as a way to group related <ProteinDetectionHypothesis>, for example where the peptide sequences identified provide

supporting evidence for more than one protein identification. There is further discussion around the issue of protein grouping, and the implementation within mzIdentML within Chapter 5.

In addition to including instructions about how to encode the typical protein sequence database-based analysis search, the mzIdentML format specification (at http://www.psidev.info/mzidentml) includes how to encode a few specific cases such as spectral library-based searches (see previous text), *de novo* searches, the use of nucleotide sequence databases, and how to report the use of multiple search engines in the same analysis. For encoding *de novo* sequencing results, the requirement to reference the database proteins from which a peptide was derived, *via* the <PeptideEvidence> element, will be removed for the next version of the standard mzIdentML 1.2, when the file is flagged as being derived from *de novo* sequencing only (see Section 11.3.3). This means that an identification must only state which peptide sequence has been found, and the associated scores or statistical values.

11.3.2 Software Implementations

Although the format is still relatively young, the adoption of mzIdentML is growing steadily. At the moment of writing, mzIdentML is exported by many of the most popular proteomics search engines and post-processing tools, including the open source tools X!Tandem (from PILEDRIVER version, 04/2015), MS-GF+, MyriMatch, the commercial software Mascot (from version 2.4), ProteinPilot, PEAKS, Scaffold, and the stand-alone open-source PeptideShaker post-processing tool,[28] which integrates different open source search engines such as X!Tandem, MS Amanda, OMSSA, Tide and Comet. Furthermore, it is becoming more common that analysis pipelines produce mzIdentML files as the final file output. One of such examples is ProteoAnnotator[29] (http://www.proteoannotator.org/), covered in Chapter 16. For software developers, there are also some open source reader libraries such as the Java-based jmzIdentML[30] (reference implementation, https://github.com/PRIDE-Utilities/jmzIdentML) and the mzidLibrary.[31] There are also some libraries in other languages such as the previously mentioned mzR package in BioConductor.

The ProteomeXchange resources PRIDE and MassIVE fully support mzIdentML as data submission format to enable the full integration and visualisation of the data in these resources (the so-called "Complete" submissions).[32] For the visualisation of the files (ideally together with the external referenced MS files), the open source and free to use PRIDE Inspector stand-alone tool[33] (https://github.com/PRIDE-Toolsuite/pride-inspector) was recently updated[34] to support the format. It also supports mzML and the rest of the open peak list spectra files (*e.g.* mzXML, mgf, pkl, ms2, dta, apl). A comprehensive list of the mzIdentML implementations can be found at http://www.psidev.info/tools-implementing-mzidentml.

11.3.3 Current Work

At the moment of writing, the next iteration of the format (version 1.2) is work in progress. Some of the novel features that will be supported are the improvement in the reporting of protein inference related information, at the protein level[35] (see Chapter 5), the possibility of including results from different MS runs in the same file, support for peptide-level scores (as groups of PSMs with the same sequence), ambiguity in the protein modification position, and reporting of cross-linking experiments. The expected schema changes (compared to version 1.1) are expected to be minimal. However, new ways of encoding this novel information will need to be taken into account by implementers, making the format potentially more challenging for those developing mzIdentML reading software.

11.4 mzQuantML

11.4.1 Data Format

mzQuantML is an XML format developed for reporting the search parameters and results of a quantitative analysis at the peptide and protein level.[36] Version 1.0 of the format was released in 2013 (http://www.psidev.info/mzquantml). mzQuantML is designed around a common core that can be extended to support particular quantitative techniques through different sets of semantic rules that are tailored for the different experimental approaches. Originally, mzQuantML supported quantitative techniques including: intensity-based MS^1 label-free, spectral counting, MS^1 label-based (such as SILAC, Chapter 7) and MS^2 tag-based (such as iTRAQ or TMT, Chapter 8). In the latest version of the format (1.0.1), support for selected reaction monitoring (SRM) approaches has also been added.[37]

The format supports the reporting at two levels: (i) the final quantification results (at the peptide and/or protein level) without detailed information about all the features taken into account for the data processing, or (ii) including all the fine-grained information at the feature level that can enable the full recreation of the results. In addition, mzQuantML enables a quite detailed description of the experimental design, an essential piece of information in quantitative studies. Like in the case of mzIdentML and mzML, mzQuantML makes heavy use of the PSI-MS CV. In addition, for the reporting of the protein modifications, both the PSI-MOD and Unimod CVs are allowed.

The top part of the mzQuantML XML schema is devoted, as in other formats, to general metadata information. First, the file captures the CVs used in the element <CvList> (common to other formats), and optionally the provider of the document (<Provider>) and their contact details (<AuditCollection>). Very importantly, a semantically valid file must contain particular CV terms included within the <AnalysisSummary> element, describing the type of data represented in the file (either intensity-based MS^1 label-free, spectral counting, MS^1 label-based, MS^2 tag-based or SRM techniques) and

whether the software is reporting values for features, peptides, proteins and/or protein groups (used to represent the ambiguity coming from the protein inference).

The <InputFiles> element captures references to the data files used for analysis, including raw MS data files (*e.g.* in mzML format), identification data (*e.g.* in mzIdentML format), the sequence database (or spectral library) from which peptides/proteins have been identified (this is not required for identification by *de novo* sequencing) and the configuration or methods files required for the analysis. There is no dependency on any particular input format, so long as they can be externally referenced by a URI (Uniform Resource Identifier). Next, similarly to other formats, mzQuantML captures a description of the software and version used in <SoftwareList>, and the analysis steps performed in the <DataProcessingList> element.

As mentioned, the experimental design is well modelled in mzQuantML. An <Assay> element typically represents the analysis of a single sample by MS (one MS run). Additional replicate analyses of the same sample are modelled as extra <Assay> elements within an <AssayList>. For quantitative techniques in which multiple samples have been compared within a single MS run, multiple <Assay> elements are defined which all refer to the same raw MS data file(s) (included within <InputFiles>, as explained). For label-free techniques, there is typically a one-to-one mapping from an <Assay> to a raw file. In label or tag-based techniques, the <Assay> must also capture the label or tag used (*e.g.* the iTRAQ or SILAC reagents used). In any case, <StudyVariable> elements are used to apply logical groupings to sets of <Assay> elements, for which quantitative values may be reported (Figure 11.3). A typical example of a study variable could be a set of biological or technical replicates, for which the software has calculated average quantitative values across <Assay> elements, then representing replicate analyses of the same sample. <StudyVariable> elements are grouped under <StudyVariableList> (Figure 11.3).

One of the key elements of the mzQuantML schema is a matrix-based element called <QuantLayer>, which is designed to be very flexible to accommodate many different scenarios and be economical in storage space. A <QuantLayer> holds a two-dimensional matrix of data values. There are various sub-types of <QuantLayer> elements, which are named according to the part of the experimental design for which data values are included (assays, study variables, ratios, global values among others), which form the columns of the data matrix. In addition, the location of the <QuantLayer> within the mzQuantML file defines the type of <QuantLayer> object for which data are reported (protein groups, proteins, peptides or features), which form the rows of the data matrix (Figure 11.3). As a concrete example, an <AssayQuantLayer> within the <ProteinList> element contains a <DataMatrix> where the columns reference <Assay> elements and the rows reference <Protein> elements. There are multiple combinations possible taking into account the experimental design and the location of the element within the file (Figure 11.3).

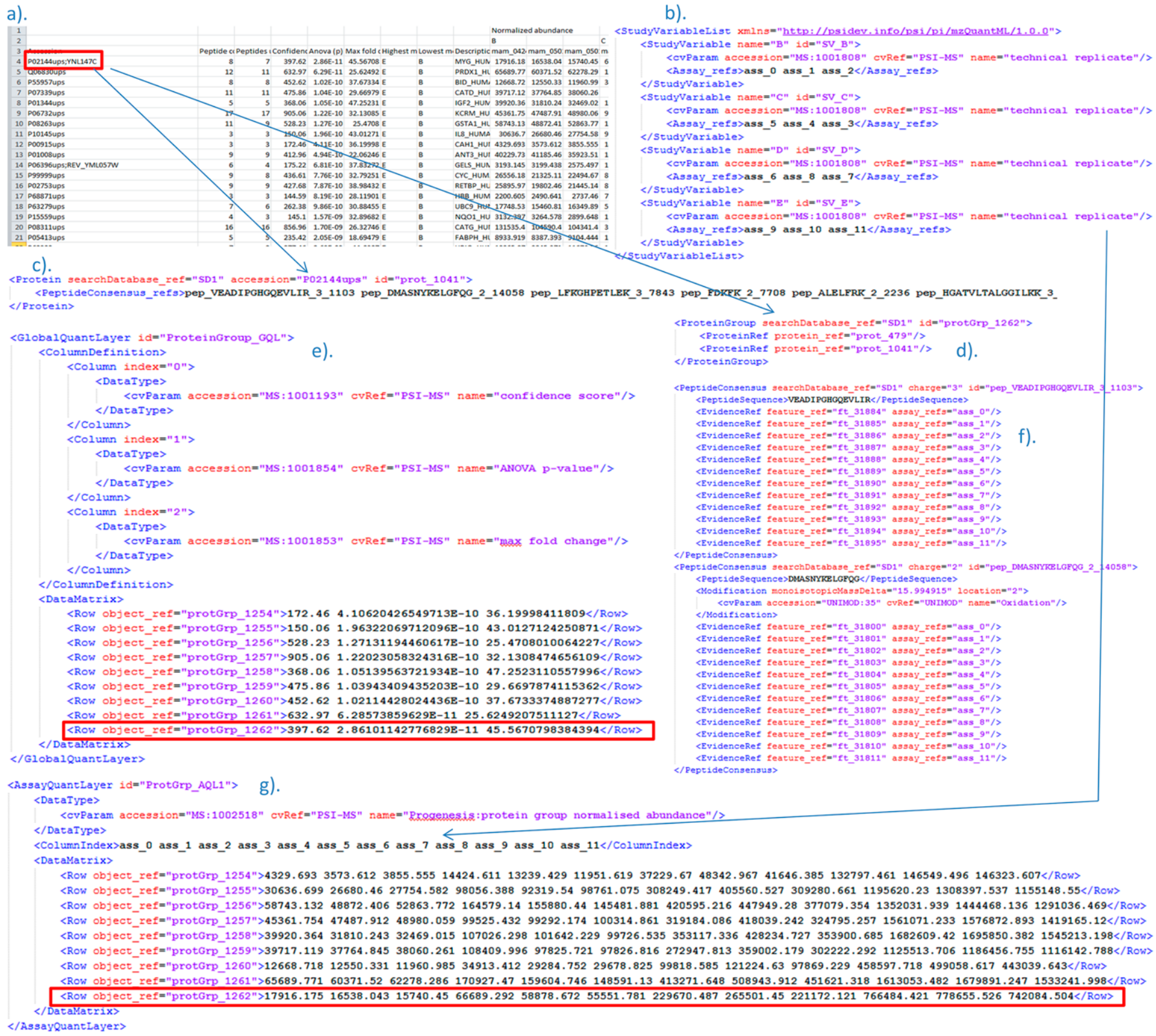

Figure 11.3 Excerpts from an mzQuantML file, generated from Progenesis QI data. The Progenesis data in tabular form (a) are shown for context here. The 'StudyVariableList' (b) contains a logical grouping of assays which can be used to make sense of the data. Proteins quantified (*e.g.* see (c)) can be grouped together into 'ProteinGroup' elements (d), these protein groups can be the basis for quantification and related statistical metrics. A 'GlobalQuantLayer' (e) can be used to describe statistical metrics for a particular protein group, and an 'AssayQuantLayer' (g) can be used to store normalised or raw quantification values for the protein groups previously described. As the quantification values are ordered by assay as described in the 'ColumnIndex', this element together with the previously described 'StudyVariableList' can be used to calculate statistical metrics as well as basic fold change values. The format can also contain a list of 'PeptideConsensus' elements (f), which can be used to list the peptides quantified in each protein and describe the original feature evidence from which these peptides were identified/quantified.

For each analysis of a given raw file (or group of raw files), a <FeatureList> can optionally be available. A <FeatureList> contains a list of positions in the 2D LC-MS space that have been quantified, called <Feature> elements. A minimal <Feature> definition includes the *m*/*z* value, the predicted charge, the retention time (if applicable) and a unique identifier within the file.

Final quantitative results can be reported as <ProteinGroup> (within a <ProteinGroupList>), <Protein> (within a <ProteinList>) or <PeptideConsensus> elements (within a <PeptideConsensusList>). The <PeptideConsensus> contains the peptide sequences (Figure 11.3).

For implementers of the formats, as mentioned before, it is important to take into account that mass spectra available in external files can be referenced (as raw files or in other formats, analogously to mzIdentML), and that protein/peptide identification information can also be referenced in external files (*e.g.* mzIdentML files). Therefore, potentially, to have all the information needed in a given quantitative experiment (mass spectra, identification and quantification), it may be necessary to handle three types of files together, something that can be quite challenging.

The mzQuantML format specification includes detailed information about how to encode the different quantification techniques in the file format. Apart from the specification document, there is a "20-minute guide to mzQuantML" document aimed to facilitate the work of implementers (http://www.psidev.info/mzquantml).

11.4.2 Software Implementations

Since the format is still relatively new, not many implementations are yet available. For software developers, there are open source reader libraries such as the Java-based jmzQuantML (reference implementation) and the recently developed mzqLibrary and mzqViewer libraries.[31] The mzqLibrary contains several converters, including converters to the format from OpenMS, Progenesis LC-MS and MaxQuant, and exporters from mzQuantML to other file formats such as mzTab. The open source ProteoSuite toolkit (http://www.proteosuite.org/), also written in Java, can output and as a key functionality, it is the only tool at present that can visualise mzQuantML files.

ProteomeXchange resources support submission of the format as a quantification output file (tagged as 'QUANT' in the submission process). The files are made available to download but no web visualisation of the results is enabled at present.

11.4.3 Current Work

Support for other quantification approaches is ongoing through the extension of the existing semantic rules and the addition of new CV terms to the PSI-MS CV, and current work is focused on providing more implementations of the format. It is also important to highlight that the mzQuantML schema formally includes support for the reporting of small molecules coming from MS metabolomics experiments (<SmallMolecule> elements within a <SmallMoleculeList>). However, this functionality has not been used in practise so far since the metabolomics community working in the development of data standards has decided to give priority to the extension of the mzTab format first (see next section).

11.5 mzTab

11.5.1 Data Format

mzTab is the latest PSI data standard developed. Version 1.0 was formalised in 2014.[38] As opposed to the other main data formats covered in this chapter, mzTab is a tab-delimited text file (https://github.com/HUPO-PSI/mzTab). During the development of mzIdentML and mzQuantML the focus was put on storing a comprehensive representation of the data. This resulted in relatively complex XML file schemas, which could potentially make it difficult for data consumers to access the information. Many "end user" data consumers are only concerned about having access to the results of a study in an easily accessible format that is compatible with tools such as Microsoft Excel® or the R programming language, among others. For this reason, mzTab was aimed at making MS proteomics and metabolomics results available to the wider biological community, beyond the field of MS. The microarray community is one example of a similar solution where the format MAGE-TAB[39] is widely used, since it can cover the main use cases, and for the sake of simplicity, is often preferred to the corresponding previously developed XML standard format called MAGE-ML.[40]

The main principle behind the development of mzTab was then to provide a flexible tab-delimited file format, to report proteomics and metabolomics results derived from MS experiments, including both identification and quantification data. In fact, mzTab enables the reporting of results at different levels, ranging from a simple summary or subset of the complete information (they could be labelled as the *final results*) up to fairly comprehensive representation of the results including a detailed experimental design.

An mzTab file can have up to five different sections: *metadata*, *protein*, *peptide*, *psm* and *small molecule* (Figure 11.4). While the *protein*, *peptide*, *PSM* and *small molecule* sections are classical table-based structures containing a header line, the *metadata* section contains one tab-separated key-value pair per row. It is important to highlight that only the *metadata* section is strictly mandatory, since there are some metadata related fields that always need to be present in the file such as "mzTab-version", "mzTab-mode" and "mzTab-type" and "description". The other four sections are then optional.

There are two types (*mzTab-type*) of files: "Identification" (including peptide, protein, and/or small molecule identifications) and "Quantification" (containing quantification results, but it may contain identification results as well). In addition, there are two supported levels of details (*mzTab-mode*) for reporting: "Summary" and "Complete". The "Summary" mode can be used to report the *final results* of a study, for example reporting data averaged from different replicates. The "Complete" mode is used if detailed experimental information coming from each individual assay and/or replicate is provided. Therefore, there are four different flavours of mzTab files, when combining the different mzTab types and modes.

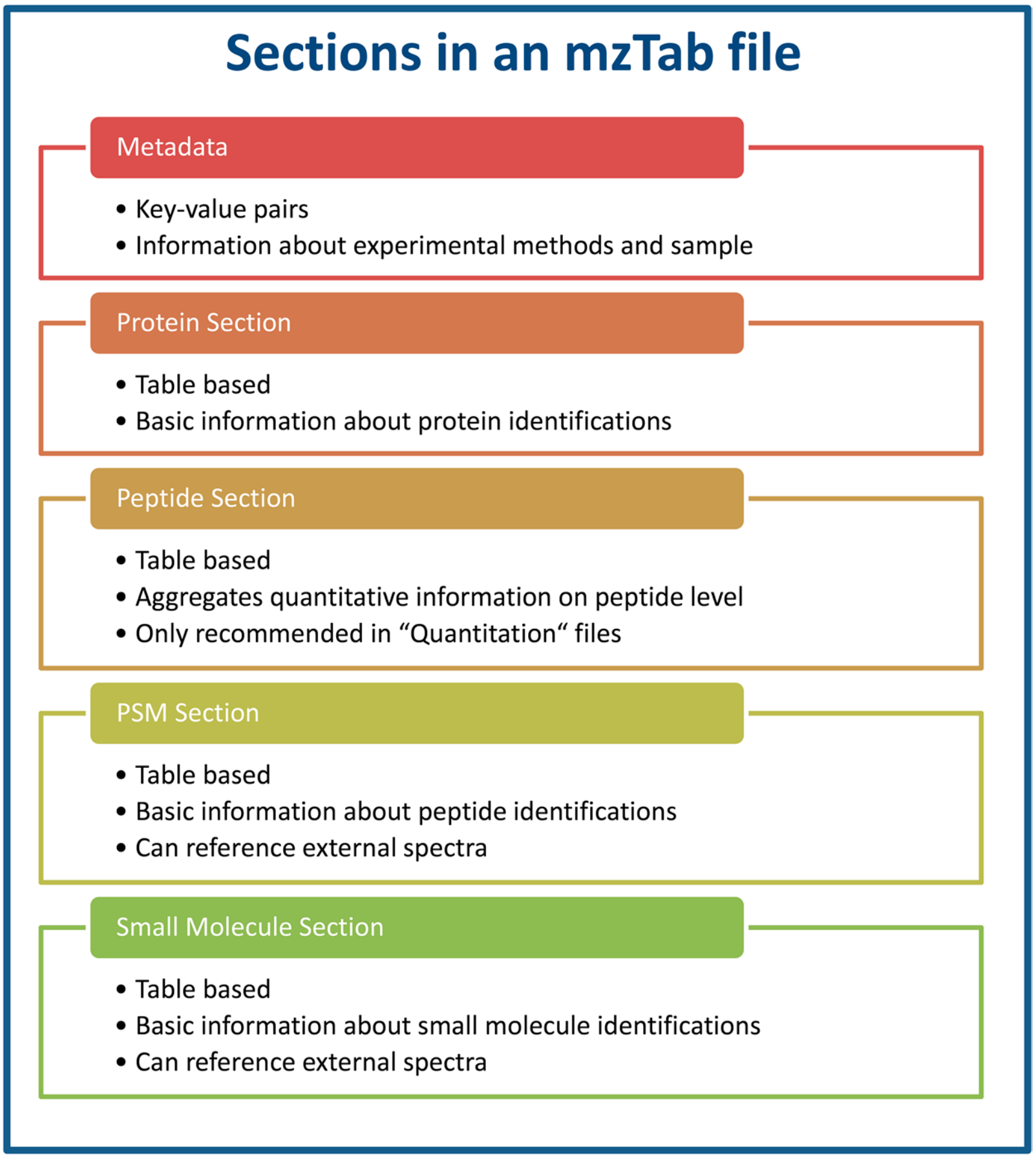

Figure 11.4 High-level overview of the data model for mzTab. Figure is reused with permission from this publication.[38]

Every line in a given file starts with a three letter code indicating the type of information captured: "MTD" (metadata section), "PRH" (protein section header), "PRT" (proteins), "PEH" (peptide section header), "PEP" (peptides), "PSH" (the PSM section header), "PSM" (peptide spectrum matches), "SMH" (small molecule section header), "SML" (small molecules), and "COM" (for comment type lines).

For a detailed list of all the fields included in the different sections, please see the specification document (http://www.psidev.info/mztab). Apart from the fields included in each of the sections, it is always possible to add customised extra columns using CV terms. First of all, the *metadata* section was deliberately kept flexible and the majority of fields are optional. Therefore, it is possible to report different levels of experimental annotation depending on the interest of the producer of the files: ranging from basic annotations up to the complete metadata defined in the MIAPE guidelines. Protein and

peptide identifications are reported in the *Protein* and *PSM* sections, respectively. The *Peptide* section is only used to report aggregated quantification data based on groups of PSMs containing the sequence. Its use is therefore not recommended in 'Identification' files.

To simplify the format, it was decided to change the modelling of protein inference in mzTab when compared to mzIdentML, excluding detailed data on how the ambiguity was actually resolved. Protein entries in mzTab files contain the column "ambiguity_members". The protein accessions listed in this field should identify proteins that were also identified through the same set of peptides or spectra, but without providing extra information. Finally, metabolomics results are reported in the *small molecule* section. Different identifiers are supported for small molecules including identifiers in different resources, Simplified Molecular-Input Line-Entry System (SMILES), and/or IUPAC International Chemical Identifier (InChI) identifiers.

As in the case of mzIdentML and mzQuantML, in mzTab it is possible to reference the corresponding mass spectra in external files and other files containing the original identification and/or quantification results (*e.g.* mzIdentML files). The experimental design related information (optional in the metadata section) is modelled in a similar way to mzQuantML, including the elements "study_variable", "assay", "ms_run", and "sample". Like in the previously mentioned formats the PSI-MS, PSI-MOD and Unimod CVs are used in mzTab.

Some specific use cases are explained in detail in the specification document. Additionally, there is a "20-minute guide to mzTab" document aimed to facilitate the work of implementers at https://github.com/HUPO-PSI/mzTab.

11.5.2 Software Implementations

The format is quite new so only a few implementations exist at present. The search engine Mascot can export the format from version 2.5. In addition, the already mentioned new version of the PRIDE Inspector tool can be used to visualise the files (both identification and quantification information, together with the referenced mass spectra). Other proteomics and metabolomics tools (*e.g.* OpenMS) have implemented an initial export to mzTab that still needs to be refined. For software developers, there is an open source library called jmzTab[41] (reference implementation, https://github.com/PRIDE-Utilities/jmzTab), written in Java.

Among the ProteomeXchange resources, MassIVE currently supports mzTab as a submission format. The PRIDE team plans to support it in the near future. All processed results files submitted to PRIDE Archive included in "Complete" submissions (both identification and quantification, *e.g.* mzIdentML and mzQuantML files) are now converted to mzTab and will be provided in this format to the users (as well as the originally submitted files). In addition, mzTab is already used heavily by PRIDE Archive as the model used for its backend storage.

11.5.3 Current Work

As already mentioned, there is interest in the metabolomics community of extending mzTab for improving the reporting of small molecule identification and quantification results. It is expected that the current *small molecule* section devoted to small molecules is extended to two or three different sections. In addition, there has also been recent interest in mzTab coming from the glycomics community. Quite likely, this will result in the near future in the existence of a core part of the format and different extensions available for the different data types (proteomics, metabolomics, glycomics and potentially others, each of them containing specific sections apart from the generic metadata section). Metabolomics and glycomics repositories are expected to formally support the format as well in the future. One of such resources is the MetaboLights database (http://www.ebi.ac.uk/metabolights/).

11.6 TraML

11.6.1 Data Format

The last PSI data standard covered in detail in this chapter is TraML, an XML-based format for encoding transition lists (and associated metadata), used in targeted proteomics approaches such as SRM (see Chapter 9).[42] The current version (1.0) was released at the end of 2011 (http://www.psidev.info/traml).

A high-level overview of the TraML XML schema is included in Figure 11.5, organised into ten highlighted top-level sets of information. The first elements in the file are, as usual, related to metadata information. The first one, called <SourceFileList>, is optional and enables the listing of the data files from which the transitions contained in the TraML file are derived. Next, <CvList> is again a required element containing the list of the CVs referenced in the file. The following elements are optional: first <ContactList>, provides a list of the people involved in the generation, validation, and/or optimisation of the transitions. The element <PublicationList> contains the publications from which the transitions are derived. Next, <InstrumentList> contains one or more instruments that can be referenced in the context of the validation and optimisation information for the transitions. And finally in this part of the schema the element <SoftwareList>, like in other formats, describes the software programs that were used to predict, validate, and/or optimise the transitions.

Following these initial metadata containers is the seventh element called <ProteinList>, an optional list of protein identifiers that may be referenced by the peptide entries. Following this is the <CompoundList>, which may contain any number of peptide or compound entries. As in mzML, mzQuantML and mzTab, TraML was designed to support encoding of metabolomics data as well. In fact, a compound is used in the format to represent not only peptides, but also chemical compounds and metabolites. These peptide or compound elements are then referenced in the subsequent transition or target lists.

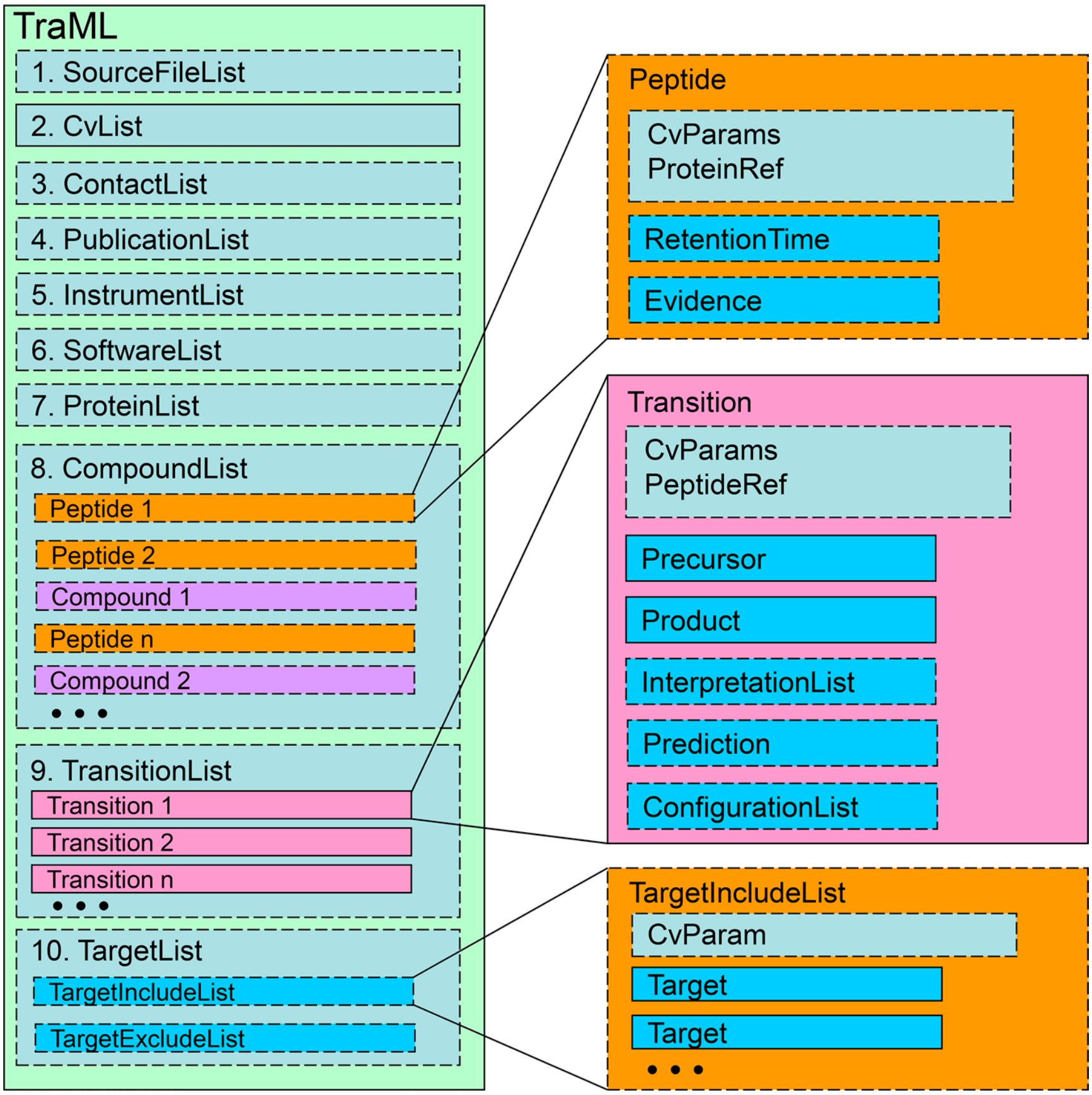

Figure 11.5 High-level overview of the XML elements included in the TraML schema. Each box represents an XML element, nested within other elements as shown. Figure reused with permission from this publication.[42]

The <TransitionList> is the next element and constitutes the main core of the schema. Each <Transition> contained in this element must at minimum contain the information about the precursor and product *m*/*z* value, but may also contain information about interpretations, predictions, and instrument configurations on which the transition has been tested or optimised. Finally, the last (optional) element is the general <TargetList>, which may contain an inclusion list and/or an exclusion list. Each of these lists contains individual targets with at minimum a precursor *m*/*z*, but optionally also retention times and other attributes. This final component was added to manage and exchange ordinary inclusion or exclusion precursor *m*/*z* lists. Like in the previously mentioned formats, the PSI-MS, PSI-MOD and Unimod CVs can be used by TraML.

11.6.2 Software Implementations

For software developers, there is an open source reader library written in Java called jtraML[43] (reference implementation, https://github.com/compomics/jtraml), as well as a C++ implementation in OpenMS. The jtraML package is not just a library, but also includes a converter tool that can convert TraML to several of the vendor-specific tab-delimited input formats and *vice versa*. The Anubis software[44] also supports TraML. The widely used Skyline tool[45] does not yet support TraML, but this would be a great benefit.

In the context of proteomics data repositories, PASSEL (maintained by the PeptideAtlas team) is the ProteomeXchange resource devoted to the storage of SRM/MRM data. As such, TraML is supported as one of the submission formats. At the moment of writing, TraML is considered to be stable, and at least at present, there is no ongoing work to extend or update the format.

11.7 Other Data Standard Formats Produced by the PSI

The PSI has also developed data standards that are widely used in the molecular interactions field (*e.g.* protein–protein interactions), such as the PSI-MI format.[46] It is an XML-based format that enables the representation of MI (molecular interactions) between different types of molecules, like for example proteins, nucleic acids and chemical compounds. PSI-MI (current version is 2.5, released in 2006, although version 3.0 is well under development at present) enables the description of highly detailed molecular interaction data and facilitates the data exchange between existing protein interaction databases from the IMEX (International Molecular EXchange) Consortium (http://www.imex.org),[47] led by the IntAct database (http://www.ebi.ac.uk/intact/). In addition, there is also a simpler, tab-delimited format called MI-TAB (there are different versions available at present, 2.5, 2.6 and 2.7), built for those users who require less detailed information and simpler parsing, following the same reasoning explained before for MAGE-TAB and mzTab. Multiple implementations of these molecular interaction standards are available. For an updated list, see http://www.psidev.info/groups/molecular-interactions.

The PEFF (PSI Extended Fasta Format) format (http://www.psidev.info/peff) is based on the widely used FASTA format, but enforces a flexible yet cleanly parsable header for each entry, in which extra information that can potentially be used for analysis software, such as post-translational modifications (PTMs) and sequence variants, can be encoded and used. The development of the format initially started in 2007, but its development was stalled due to the lack of agreement between the authors and the reviewers during the PSI document process. However, resources such as neXtProt (http://www.nextprot.org/) have provided protein sequence data using the preliminary version of the format and some visualisation software supporting PEFF has also been developed. In 2015, efforts have been restarted to finalise and formalise the format.

11.8 Conclusions

The development and maintenance of a data standard is a collaborative and generally quite resource-intensive task. In fact, the actual development represents only the very first step, since wide adoption usually takes a long time, if it happens at all. The existence of easy-to-use and preferably free-to-use software is essential to enable the adoption of the formats. For this reason, the PSI has also spent considerable efforts in developing different application programming interfaces (APIs) that implement the different standard data formats.

However, the continuous evolution and inherent complexity of the proteomics analysis data workflows, together with the developments in instrumentation demand that the current data standards and related software are in continual evolution. Clearly the right balance needs to be found since it is well-known that a data format needs to be stable for quite some time before it will be widely implemented and adopted, especially in commercial software, which often has a long lead time and requires documented justification for adding features. This is the reason why the latest developments in a given field can take some time to get incorporated into the standards.

In the coming years, it is expected that collaboration between the PSI and the MS metabolomics community will continue in order to leverage existing products and experiences and encourage greater interoperability among software tools used in these two fields. Currently it seems likely that mzML will be widely adopted in both fields, and the extension of the mzTab format for encoding MS metabolomics identification and quantification results will be completed. Another future topic of interest will be the extension of existing data standards for the increasingly used DIA approaches such as SWATH-MS and MS^E. In this context but also in others, data compression will be an active field of research and of possible interest to the PSI.

Finally, proteogenomics is a fast-growing field still in need of suitable data standards and reporting guidelines, so it is expected that some future efforts will also be devoted to this task. In any case, new contributors to the PSI activities are always welcome. If you are willing to contribute, the best way to start would be to join the PSI mailing lists and participate in the annual PSI Spring meetings (http://www.psidev.info/).

Abbreviations

API	Application Programming Interface
CV	Controlled Vocabulary
DIA	Data Independent Acquisition
(HUPO) PSI	(Human Proteome Organisation) Proteomics Standards Initiative
IMEX	International Molecular EXchange
InChI	International Chemical Identifier
MIAPE	Minimum Information About a Proteomics Experiment
MI	Molecular Interactions

MRM	Multiple Reaction Monitoring
PASSEL	PeptideAtlas SRM Experiment Library
PEFF	PSI Extended Fasta Format
PRIDE	PRoteomics IDEntifications (database)
PSM	Peptide Spectrum Match
PTM	Post-Translational Modification
SMILES	Simplified Molecular-Input Line-Entry System
SRM	Selected Reaction Monitoring
URI	Uniform Resource Identifier
XML	Extensible Markup Language

Acknowledgements

J.A.V. acknowledges funding from the Wellcome Trust [grant number WT101477MA] and from the BBSRC 'PROCESS' grant [BB/K01997X/1]. A.R.J. also acknowledges funding from BBSRC [BB/L024128/1 and BB/K01997X/1]. S.P. acknowledges funding from BBSRC [BB/I00095X/1]. E.W.D acknowledges the NIGMS grant R01GM087221 and the NIBIB grant U54EB020406. The authors want to acknowledge all the contributors to the activities of the Proteomics Standards Initiative.

References

1. E. W. Deutsch, File formats commonly used in mass spectrometry proteomics, *Mol. Cell. Proteomics*, 2012, **11**, 1612–1621.
2. C. Brooksbank and J. Quackenbush, Data standards: a call to action, *OMICS*, 2006, **10**, 94–99.
3. E. W. Deutsch, J. P. Albar, P. A. Binz, M. Eisenacher, A. R. Jones, G. Mayer, G. S. Omenn, S. Orchard, J. A. Vizcaino and H. Hermjakob, Development of data representation standards by the human proteome organization proteomics standards initiative, *J. Am. Med. Inform. Assoc.*, 2015, **22**, 495–506.
4. Anonymous, Democratizing proteomics data, *Nat. Biotechnol.*, 2007, **25**, 262.
5. G. Mayer, A. R. Jones, P. A. Binz, E. W. Deutsch, S. Orchard, L. Montecchi-Palazzi, J. A. Vizcaino, H. Hermjakob, D. Oveillero, R. Julian, C. Stephan, H. E. Meyer and M. Eisenacher, Controlled vocabularies and ontologies in proteomics: overview, principles and practice, *Biochim. Biophys. Acta*, 2014, **1844**, 98–107.
6. G. Mayer, L. Montecchi-Palazzi, D. Ovelleiro, A. R. Jones, P. A. Binz, E. W. Deutsch, M. Chambers, M. Kallhardt, F. Levander, J. Shofstahl, S. Orchard, J. A. Vizcaino, H. Hermjakob, C. Stephan, H. E. Meyer, M. Eisenacher and H.-P. Group, The HUPO proteomics standards initiative- mass spectrometry controlled vocabulary, *Database*, 2013, **2013**, bat009.
7. L. Montecchi-Palazzi, R. Beavis, P. A. Binz, R. J. Chalkley, J. Cottrell, D. Creasy, J. Shofstahl, S. L. Seymour and J. S. Garavelli, The PSI-MOD community standard for representation of protein modification data, *Nat. Biotechnol.*, 2008, **26**, 864–866.

8. C. F. Taylor, N. W. Paton, K. S. Lilley, P. A. Binz, R. K. Julian Jr., A. R. Jones, W. Zhu, R. Apweiler, R. Aebersold, E. W. Deutsch, M. J. Dunn, A. J. Heck, A. Leitner, M. Macht, M. Mann, L. Martens, T. A. Neubert, S. D. Patterson, P. Ping, S. L. Seymour, P. Souda, A. Tsugita, J. Vandekerckhove, T. M. Vondriska, J. P. Whitelegge, M. R. Wilkins, I. Xenarios, J. R. Yates 3rd and H. Hermjakob, The minimum information about a proteomics experiment (MIAPE), *Nat. Biotechnol.*, 2007, **25**, 887–893.
9. C. F. Taylor, P. A. Binz, R. Aebersold, M. Affolter, R. Barkovich, E. W. Deutsch, D. M. Horn, A. Huhmer, M. Kussmann, K. Lilley, M. Macht, M. Mann, D. Muller, T. A. Neubert, J. Nickson, S. D. Patterson, R. Raso, K. Resing, S. L. Seymour, A. Tsugita, I. Xenarios, R. Zeng and R. K. Julian Jr., Guidelines for reporting the use of mass spectrometry in proteomics, *Nat. Biotechnol.*, 2008, **26**, 860–861.
10. S. Martinez-Bartolome, E. W. Deutsch, P. A. Binz, A. R. Jones, M. Eisenacher, G. Mayer, A. Campos, F. Canals, J. J. Bech-Serra, M. Carrascal, M. Gay, A. Paradela, R. Navajas, M. Marcilla, M. L. Hernaez, M. D. Gutierrez-Blazquez, L. F. Velarde, K. Aloria, J. Beaskoetxea, J. A. Medina-Aunon and J. P. Albar, Guidelines for reporting quantitative mass spectrometry based experiments in proteomics, *J. Proteomics*, 2013, **95**, 84–88.
11. S. Orchard, L. Salwinski, S. Kerrien, L. Montecchi-Palazzi, M. Oesterheld, V. Stumpflen, A. Ceol, A. Chatr-aryamontri, J. Armstrong, P. Woollard, J. J. Salama, S. Moore, J. Wojcik, G. D. Bader, M. Vidal, M. E. Cusick, M. Gerstein, A. C. Gavin, G. Superti-Furga, J. Greenblatt, J. Bader, P. Uetz, M. Tyers, P. Legrain, S. Fields, N. Mulder, M. Gilson, M. Niepmann, L. Burgoon, J. De Las Rivas, C. Prieto, V. M. Perreau, C. Hogue, H. W. Mewes, R. Apweiler, I. Xenarios, D. Eisenberg, G. Cesareni and H. Hermjakob, The minimum information required for reporting a molecular interaction experiment (MIMIx), *Nat. Biotechnol.*, 2007, **25**, 894–898.
12. J. A. Vizcaino, L. Martens, H. Hermjakob, R. K. Julian and N. W. Paton, The PSI formal document process and its implementation on the PSI website, *Proteomics*, 2007, **7**, 2355–2357.
13. J. A. Vizcaino, E. W. Deutsch, R. Wang, A. Csordas, F. Reisinger, D. Rios, J. A. Dianes, Z. Sun, T. Farrah, N. Bandeira, P. A. Binz, I. Xenarios, M. Eisenacher, G. Mayer, L. Gatto, A. Campos, R. J. Chalkley, H. J. Kraus, J. P. Albar, S. Martinez-Bartolome, R. Apweiler, G. S. Omenn, L. Martens, A. R. Jones and H. Hermjakob, ProteomeXchange provides globally coordinated proteomics data submission and dissemination, *Nat. Biotechnol.*, 2014, **32**, 223–226.
14. J. A. Vizcaino, R. G. Cote, A. Csordas, J. A. Dianes, A. Fabregat, J. M. Foster, J. Griss, E. Alpi, M. Birim, J. Contell, G. O'Kelly, A. Schoenegger, D. Ovelleiro, Y. Perez-Riverol, F. Reisinger, D. Rios, R. Wang and H. Hermjakob, The PRoteomics IDEntifications (PRIDE) database and associated tools: status in 2013, *Nucleic Acids Res.*, 2013, **41**, D1063–D1069.
15. E. W. Deutsch, H. Lam and R. Aebersold, PeptideAtlas: a resource for target selection for emerging targeted proteomics workflows, *EMBO Rep.*, 2008, **9**, 429–434.

16. Y. Perez-Riverol, E. Alpi, R. Wang, H. Hermjakob and J. A. Vizcaino, Making proteomics data accessible and reusable: current state of proteomics databases and repositories, *Proteomics*, 2015, **15**, 930–949.
17. L. Martens, M. Chambers, M. Sturm, D. Kessner, F. Levander, J. Shofstahl, W. H. Tang, A. Rompp, S. Neumann, A. D. Pizarro, L. Montecchi-Palazzi, N. Tasman, M. Coleman, F. Reisinger, P. Souda, H. Hermjakob, P. A. Binz and E. W. Deutsch, mzML–a community standard for mass spectrometry data, *Mol. Cell. Proteomics*, 2011, **10**, R110 000133.
18. P. G. Pedrioli, J. K. Eng, R. Hubley, M. Vogelzang, E. W. Deutsch, B. Raught, B. Pratt, E. Nilsson, R. H. Angeletti, R. Apweiler, K. Cheung, C. E. Costello, H. Hermjakob, S. Huang, R. K. Julian, E. Kapp, M. E. McComb, S. G. Oliver, G. Omenn, N. W. Paton, R. Simpson, R. Smith, C. F. Taylor, W. Zhu and R. Aebersold, A common open representation of mass spectrometry data and its application to proteomics research, *Nat. Biotechnol.*, 2004, **22**, 1459–1466.
19. L. Montecchi-Palazzi, S. Kerrien, F. Reisinger, B. Aranda, A. R. Jones, L. Martens and H. Hermjakob, The PSI semantic validator: a framework to check MIAPE compliance of proteomics data, *Proteomics*, 2009, **9**, 5112–5119.
20. M. C. Chambers, B. Maclean, R. Burke, D. Amodei, D. L. Ruderman, S. Neumann, L. Gatto, B. Fischer, B. Pratt, J. Egertson, K. Hoff, D. Kessner, N. Tasman, N. Shulman, B. Frewen, T. A. Baker, M. Y. Brusniak, C. Paulse, D. Creasy, L. Flashner, K. Kani, C. Moulding, S. L. Seymour, L. M. Nuwaysir, B. Lefebvre, F. Kuhlmann, J. Roark, P. Rainer, S. Detlev, T. Hemenway, A. Huhmer, J. Langridge, B. Connolly, T. Chadick, K. Holly, J. Eckels, E. W. Deutsch, R. L. Moritz, J. E. Katz, D. B. Agus, M. MacCoss, D. L. Tabb and P. Mallick, A cross-platform toolkit for mass spectrometry and proteomics, *Nat. Biotechnol.*, 2012, **30**, 918–920.
21. R. G. Cote, F. Reisinger and L. Martens, jmzML, an open-source Java API for mzML, the PSI standard for MS data, *Proteomics*, 2010, **10**, 1332–1335.
22. T. Bald, J. Barth, A. Niehues, M. Specht, M. Hippler and C. Fufezan, pymzML–Python module for high-throughput bioinformatics on mass spectrometry data, *Bioinformatics*, 2012, **28**, 1052–1053.
23. J. Teleman, A. W. Dowsey, F. F. Gonzalez-Galarza, S. Perkins, B. Pratt, H. L. Rost, L. Malmstrom, J. Malmstrom, A. R. Jones, E. W. Deutsch and F. Levander, Numerical compression schemes for proteomics mass spectrometry data, *Mol. Cell. Proteomics*, 2014, **13**, 1537–1542.
24. T. Schramm, A. Hester, I. Klinkert, J. P. Both, R. M. Heeren, A. Brunelle, O. Laprevote, N. Desbenoit, M. F. Robbe, M. Stoeckli, B. Spengler and A. Rompp, imzML–a common data format for the flexible exchange and processing of mass spectrometry imaging data, *J. Proteomics*, 2012, **75**, 5106–5110.
25. M. Wilhelm, M. Kirchner, J. A. Steen and H. Steen, mz5: space- and time-efficient storage of mass spectrometry data sets, *Mol. Cell. Proteomics*, 2012, **11**, O111 011379.

26. D. Bouyssie, M. Dubois, S. Nasso, A. Gonzalez de Peredo, O. Burlet-Schiltz, R. Aebersold and B. Monsarrat, mzDB: a file format using multiple indexing strategies for the efficient analysis of large LC-MS/MS and SWATH-MS data sets, *Mol. Cell. Proteomics*, 2015, **14**, 771–781.
27. A. R. Jones, M. Eisenacher, G. Mayer, O. Kohlbacher, J. Siepen, S. J. Hubbard, J. N. Selley, B. C. Searle, J. Shofstahl, S. L. Seymour, R. Julian, P. A. Binz, E. W. Deutsch, H. Hermjakob, F. Reisinger, J. Griss, J. A. Vizcaino, M. Chambers, A. Pizarro and D. Creasy, The mzIdentML data standard for mass spectrometry-based proteomics results, *Mol. Cell. Proteomics*, 2012, **11**, M111 014381.
28. M. Vaudel, J. M. Burkhart, R. P. Zahedi, E. Oveland, F. S. Berven, A. Sickmann, L. Martens and H. Barsnes, PeptideShaker enables reanalysis of MS-derived proteomics data sets, *Nat. Biotechnol.*, 2015, **33**, 22–24.
29. F. Ghali, R. Krishna, S. Perkins, A. Collins, D. Xia, J. Wastling and A. R. Jones, ProteoAnnotator–open source proteogenomics annotation software supporting PSI standards, *Proteomics*, 2014, **14**, 2731–2741.
30. F. Reisinger, R. Krishna, F. Ghali, D. Rios, H. Hermjakob, J. A. Vizcaino and A. R. Jones, jmzIdentML API: A Java interface to the mzIdentML standard for peptide and protein identification data, *Proteomics*, 2012, **12**, 790–794.
31. F. Ghali, R. Krishna, P. Lukasse, S. Martinez-Bartolome, F. Reisinger, H. Hermjakob, J. A. Vizcaino and A. R. Jones, Tools (Viewer, Library and Validator) that facilitate use of the peptide and protein identification standard format, termed mzIdentML, *Mol. Cell. Proteomics*, 2013, **12**, 3026–3035.
32. T. Ternent, A. Csordas, D. Qi, G. Gomez-Baena, R. J. Beynon, A. R. Jones, H. Hermjakob and J. A. Vizcaino, How to submit MS proteomics data to ProteomeXchange via the PRIDE database, *Proteomics*, 2014, **14**, 2233–2241.
33. R. Wang, A. Fabregat, D. Rios, D. Ovelleiro, J. M. Foster, R. G. Cote, J. Griss, A. Csordas, Y. Perez-Riverol, F. Reisinger, H. Hermjakob, L. Martens and J. A. Vizcaino, PRIDE Inspector: a tool to visualize and validate MS proteomics data, *Nat. Biotechnol.*, 2012, **30**, 135–137.
34. Y. Perez-Riverol, Q. W. Xu, R. Wang, J. Uszkoreit, J. Griss, A. Sanchez, F. Reisinger, A. Csordas, T. Ternent, N. Del-Toro, J. A. Dianes, M. Eisenacher, H. Hermjakob and J. A. Vizcaino, PRIDE Inspector Toolsuite: moving towards a universal visualization tool for proteomics data standard formats and quality assessment of ProteomeXchange datasets, *Mol. Cell. Proteomics*, 2016, **15**, 305–317.
35. S. L. Seymour, T. Farrah, P. A. Binz, R. J. Chalkley, J. S. Cottrell, B. C. Searle, D. L. Tabb, J. A. Vizcaino, G. Prieto, J. Uszkoreit, M. Eisenacher, S. Martinez-Bartolome, F. Ghali and A. R. Jones, A standardized framing for reporting protein identifications in mzIdentML 1.2, *Proteomics*, 2014, **14**, 2389–2399.
36. M. Walzer, D. Qi, G. Mayer, J. Uszkoreit, M. Eisenacher, T. Sachsenberg, F. F. Gonzalez-Galarza, J. Fan, C. Bessant, E. W. Deutsch, F. Reisinger,

J. A. Vizcaino, J. A. Medina-Aunon, J. P. Albar, O. Kohlbacher and A. R. Jones, The mzQuantML data standard for mass spectrometry-based quantitative studies in proteomics, *Mol. Cell. Proteomics*, 2013, **12**, 2332–2340.

37. D. Qi, C. Lawless, J. Teleman, F. Levander, S. W. Holman, S. Hubbard and A. R. Jones, Representation of selected-reaction monitoring data in the mzQuantML data standard, *Proteomics*, 2015, **15**, 2592–2596.
38. J. Griss, A. R. Jones, T. Sachsenberg, M. Walzer, L. Gatto, J. Hartler, G. G. Thallinger, R. M. Salek, C. Steinbeck, N. Neuhauser, J. Cox, S. Neumann, J. Fan, F. Reisinger, Q. W. Xu, N. Del Toro, Y. Perez-Riverol, F. Ghali, N. Bandeira, I. Xenarios, O. Kohlbacher, J. A. Vizcaino and H. Hermjakob, The mzTab data exchange format: communicating mass-spectrometry-based proteomics and metabolomics experimental results to a wider audience, *Mol. Cell. Proteomics*, 2014, **13**, 2765–2775.
39. T. F. Rayner, P. Rocca-Serra, P. T. Spellman, H. C. Causton, A. Farne, E. Holloway, R. A. Irizarry, J. Liu, D. S. Maier, M. Miller, K. Petersen, J. Quackenbush, G. Sherlock, C. J. Stoeckert Jr., J. White, P. L. Whetzel, F. Wymore, H. Parkinson, U. Sarkans, C. A. Ball and A. Brazma, A simple spreadsheet-based, MIAME-supportive format for microarray data: MAGE-TAB, *BMC Bioinf.*, 2006, **7**, 489.
40. P. T. Spellman, M. Miller, J. Stewart, C. Troup, U. Sarkans, S. Chervitz, D. Bernhart, G. Sherlock, C. Ball, M. Lepage, M. Swiatek, W. L. Marks, J. Goncalves, S. Markel, D. Iordan, M. Shojatalab, A. Pizarro, J. White, R. Hubley, E. Deutsch, M. Senger, B. J. Aronow, A. Robinson, D. Bassett, C. J. Stoeckert Jr. and A. Brazma, Design and implementation of microarray gene expression markup language (MAGE-ML), *Genome Biol.*, 2002, **3**, RESEARCH0046.
41. Q. W. Xu, J. Griss, R. Wang, A. R. Jones, H. Hermjakob and J. A. Vizcaino, jmzTab: a java interface to the mzTab data standard, *Proteomics*, 2014, **14**, 1328–1332.
42. E. W. Deutsch, M. Chambers, S. Neumann, F. Levander, P. A. Binz, J. Shofstahl, D. S. Campbell, L. Mendoza, D. Ovelleiro, K. Helsens, L. Martens, R. Aebersold, R. L. Moritz and M. Y. Brusniak, TraML–a standard format for exchange of selected reaction monitoring transition lists, *Mol. Cell. Proteomics*, 2012, **11**, R111 015040.
43. K. Helsens, M. Y. Brusniak, E. Deutsch, R. L. Moritz and L. Martens, jTraML: an open source Java API for TraML, the PSI standard for sharing SRM transitions, *J. Proteome Res.*, 2011, **10**, 5260–5263.
44. J. Teleman, C. Karlsson, S. Waldemarson, K. Hansson, P. James, J. Malmstrom and F. Levander, Automated selected reaction monitoring software for accurate label-free protein quantification, *J. Proteome Res.*, 2012, **11**, 3766–3773.
45. B. MacLean, D. M. Tomazela, N. Shulman, M. Chambers, G. L. Finney, B. Frewen, R. Kern, D. L. Tabb, D. C. Liebler and M. J. MacCoss, Skyline: an open source document editor for creating and analyzing targeted proteomics experiments, *Bioinformatics*, 2010, **26**, 966–968.

46. H. Hermjakob, L. Montecchi-Palazzi, G. Bader, J. Wojcik, L. Salwinski, A. Ceol, S. Moore, S. Orchard, U. Sarkans, C. von Mering, B. Roechert, S. Poux, E. Jung, H. Mersch, P. Kersey, M. Lappe, Y. Li, R. Zeng, D. Rana, M. Nikolski, H. Husi, C. Brun, K. Shanker, S. G. Grant, C. Sander, P. Bork, W. Zhu, A. Pandey, A. Brazma, B. Jacq, M. Vidal, D. Sherman, P. Legrain, G. Cesareni, I. Xenarios, D. Eisenberg, B. Steipe, C. Hogue and R. Apweiler, The HUPO PSI's molecular interaction format–a community standard for the representation of protein interaction data, *Nat. Biotechnol.*, 2004, **22**, 177–183.
47. S. Orchard, S. Kerrien, S. Abbani, B. Aranda, J. Bhate, S. Bidwell, A. Bridge, L. Briganti, F. S. Brinkman, G. Cesareni, A. Chatr-aryamontri, E. Chautard, C. Chen, M. Dumousseau, J. Goll, R. E. Hancock, L. I. Hannick, I. Jurisica, J. Khadake, D. J. Lynn, U. Mahadevan, L. Perfetto, A. Raghunath, S. Ricard-Blum, B. Roechert, L. Salwinski, V. Stumpflen, M. Tyers, P. Uetz, I. Xenarios and H. Hermjakob, Protein interaction data curation: the International Molecular Exchange (IMEx) consortium, *Nat. Methods*, 2012, **9**, 345–350.

CHAPTER 12

OpenMS: A Modular, Open-Source Workflow System for the Analysis of Quantitative Proteomics Data

LARS NILSE[a]

[a]Institute of Molecular Medicine and Cell Research, University of Freiburg, D-79104 Freiburg, Germany
*E-mail: lars.nilse@mol-med.uni-freiburg.de

12.1 Introduction

Mass-spectrometry-based proteomics has greatly evolved over the past decade, and is now an integral part of many biological and clinical studies. New techniques have emerged. New mass spectrometry hardware allows for ever-increasing throughput and accuracy. In the same way, the bioinformatics tools for the analysis of mass spectrometry data have become more sensitive, reliable and easier to use. Researchers can now choose from a wide range of software solutions, either academic or commercial. Their choice can have a major impact on the final results and conclusions of their studies.[1]

OpenMS is one of these software solutions. It allows for the flexible and transparent analysis of both proteomics and metabolomics data. The underlying library is implemented in C++ but can also be accessed *via* a Python

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

interface.[2] The code is openly accessible at GitHub https://github.com/OpenMS and described by an extensive documentation http://openms.de/documentation. Many external software solutions such as common search engines and protein inference algorithms can be integrated by the provided wrappers. An important distinguishing feature is its modularity. OpenMS is not a single, monolithic program but rather a collection of over 100 individual tools. These tools can be combined to simple or increasingly complex analysis workflows. A small subset of roughly 30 tools is sufficient to construct most standard workflows, Tables 12.1–12.3. In this chapter, we will present four example workflows and in the process discuss various aspects of OpenMS. We will start with a simple peptide identification workflow in 12.2, and explain how to integrate external search engines into OpenMS workflows. Next, we will have a closer look at a complete iTRAQ protein quantification workflow in 12.3. We will discuss how proteins are inferred from peptide sequence information, and which of these proteins are then quantified. In the following Section 12.4, we will focus on the quantification of dimethyl-labeled samples. In this workflow, the protein quantification relies on the correct detection of peptide features at MS1 spectrum level. As a final example, we will discuss a label-free quantification workflow in 12.5. Peptide sequence identification and peptide feature detection are performed as in the previously discussed workflows. The challenge of this analysis is the correct alignment and linking of corresponding peptide features in the independently measured mass spectrometry runs.

OpenMS installers for all major operating systems can be downloaded at http://openms.de/download. The four workflows described in detail in this chapter are available from the OpenMS workflow repository at http://openms.de/workflows. The repository contains a collection of standard workflows, which are already optimized for specific mass spectrometry machines. The

Table 12.1 Tools for identification in OpenMS.

Tool	Description	Section
DecoyDatabase	Appends decoy sequences to a target database	12.2
MascotAdapter, XTandemAdapter, MSGFPlusAdapter, …	Search engine adapters. Enables the integration of external search engines such as Mascot, X!Tandem and MS-GF+	12.2
PeptideIndexer	Adds target-decoy annotation and protein references for all peptides	12.2
FalseDiscoveryRate	Calculates false discovery rates for target-decoy searches on peptide and protein level	12.2
IDPosteriorError-Probability	Calculates posterior error probabilities	12.3
ConsensusID	Combines multiple search results and determines the best PSM for each MS2 spectrum	12.4
HighResPrecursor-MassCorrector	Corrects precursor positions to the nearest peptide feature or the nearest centroided MS1 peak	12.4
FidoAdapter	Adapter for the protein inference engine Fido[28]	12.3

Table 12.2 Tools for quantification in OpenMS.

Tool	Description	Section
PeakPickerHiRes, PeakPickerWavelet	Detects peaks in high- and low-resolution profile mass spectrometry data	12.4
FeatureFinderMultiplex, FeatureFinderCentroided	Detects peptide features in profile or centroided LC-MS data	12.4
SpectraMerger	Averages or merges neighboring spectra. Averaging results in smoothing and noise reduction in RT direction	12.4
NoiseFilterSGolay, NoiseFilterGaussian	Savitzky-Golay and Gaussian smoothing and noise reduction of individual spectra *i.e.* in m/z direction	12.4
ITRAQAnalyzer	Extracts and normalizes iTRAQ reporter ion intensities	12.3
IDMapper	Assigns sequences to peptide features	12.3
IDConflictResolver	Resolves ambiguous feature annotations. If multiple sequences are assigned to the same feature, only the sequence with the highest score is retained	12.3
MapAlignerPoseClustering	Corrects relative retention time shifts between mass spectrometry runs	12.5
MapRTTransformer	Applies retention time transformations to a mass spectrometry run	12.5
FeatureLinkerUnlabeledQT	Links corresponding peptide features in different mass spectrometry runs	12.5
ProteinQuantifier	Determines protein abundances from peptide level abundances and protein inference information	12.3

Table 12.3 Tools for file handling in OpenMS.

Tool	Description	Section
FileConverter	Converts files between different formats. For example, mzML to mgf	12.2
FileMerger	Merges several files. For example, peptide features from multiple fractions	12.3
FileFilter	Extracts subsets from raw files or quantitative results	12.4
IDFileConverter	Converts peptide and protein identifications between different file formats. For example, pepXML to mzid	12.2
IDMerger	Merges several peptide and protein identification results into one file. For example, identifications from multiple search engine runs	12.3
IDFilter	Extracts subsets from search results. For example, all peptides from a specific protein	12.2
MzTabExporter	Exports peptide and protein level results to mzTab	12.2
FileInfo	Summary for mzML, featureXML and consensusXML files	12.2

combination of OpenMS and its workflow repository allows for a speedy and optimal analysis of the experimental data.

12.2 Peptide Identification

In a typical OpenMS data analysis, the user works with two different programs: the workflow editor TOPPAS[3,4] for design and execution of workflows, and the data viewer TOPPView for the inspection of experimental data and intermediate results, Figures 12.1 and 12.2. Before any analysis workflow can be run, two preparatory steps are necessary: the conversion of the experimental LC-MS/MS data to an open non-proprietary file format, and the construction of a target-decoy protein database.

OpenMS requires the experimental mass spectrometry data in an open file format such as mzML[5,6] or one of the legacy formats mzXML and mzData. If the vendor acquisition software does not support export to an open format, we recommend ProteoWizard's msConvert tool[7,8] for the conversion from the proprietary formats to mzML. The ProteoWizard toolkit ships with continuously updated, vendor-provided libraries from Sciex, Thermo Fisher Scientific, Agilent, Bruker and Waters. This single converter therefore bridges the vendor-specific formats and open community standards, and allows OpenMS to analyze data from all major mass spectrometry vendors.

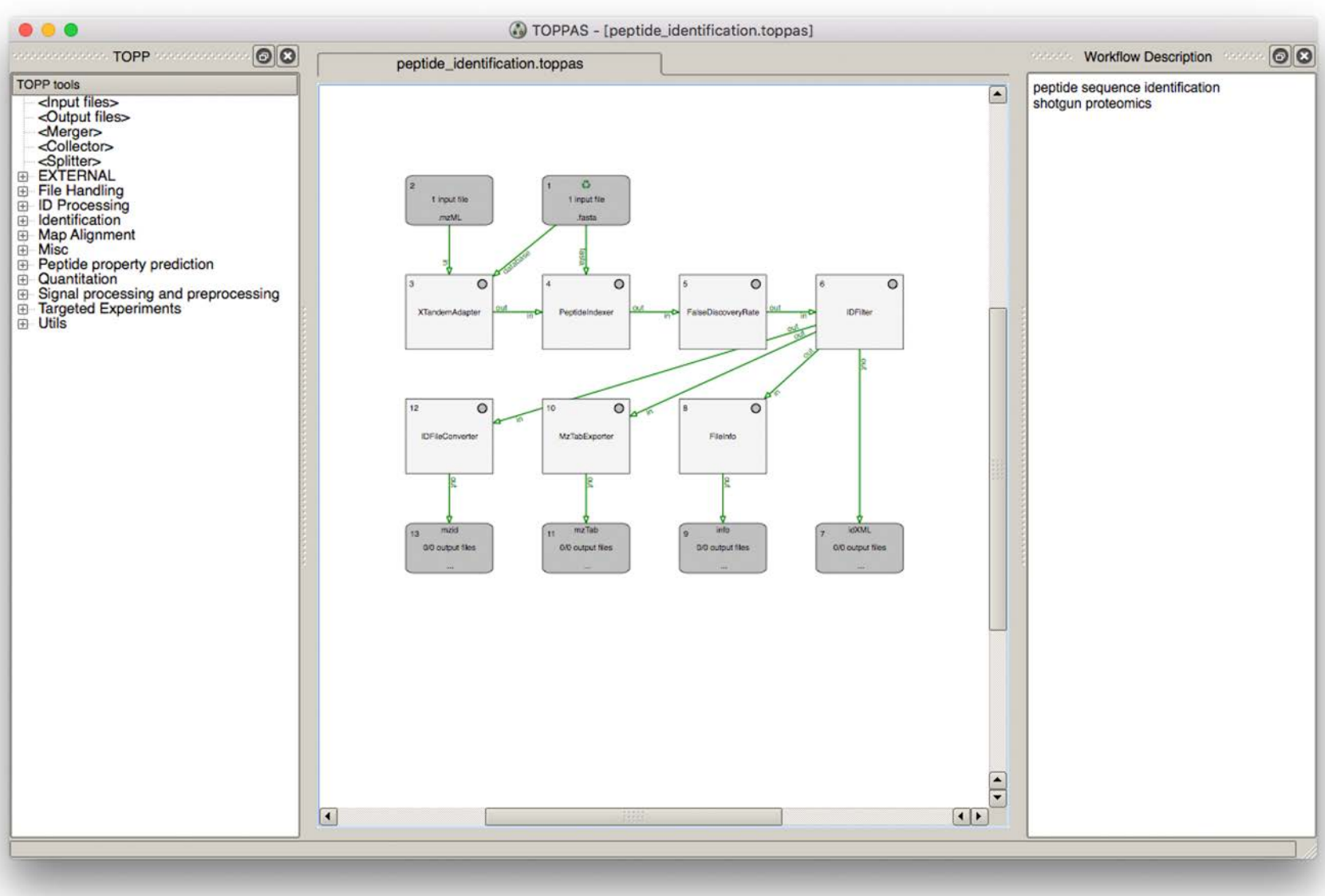

Figure 12.1 Workflow editor TOPPAS. The left panel contains the list of all individual tools. The central space contains tabs for multiple workflows. The right panel is reserved for a description and notes.

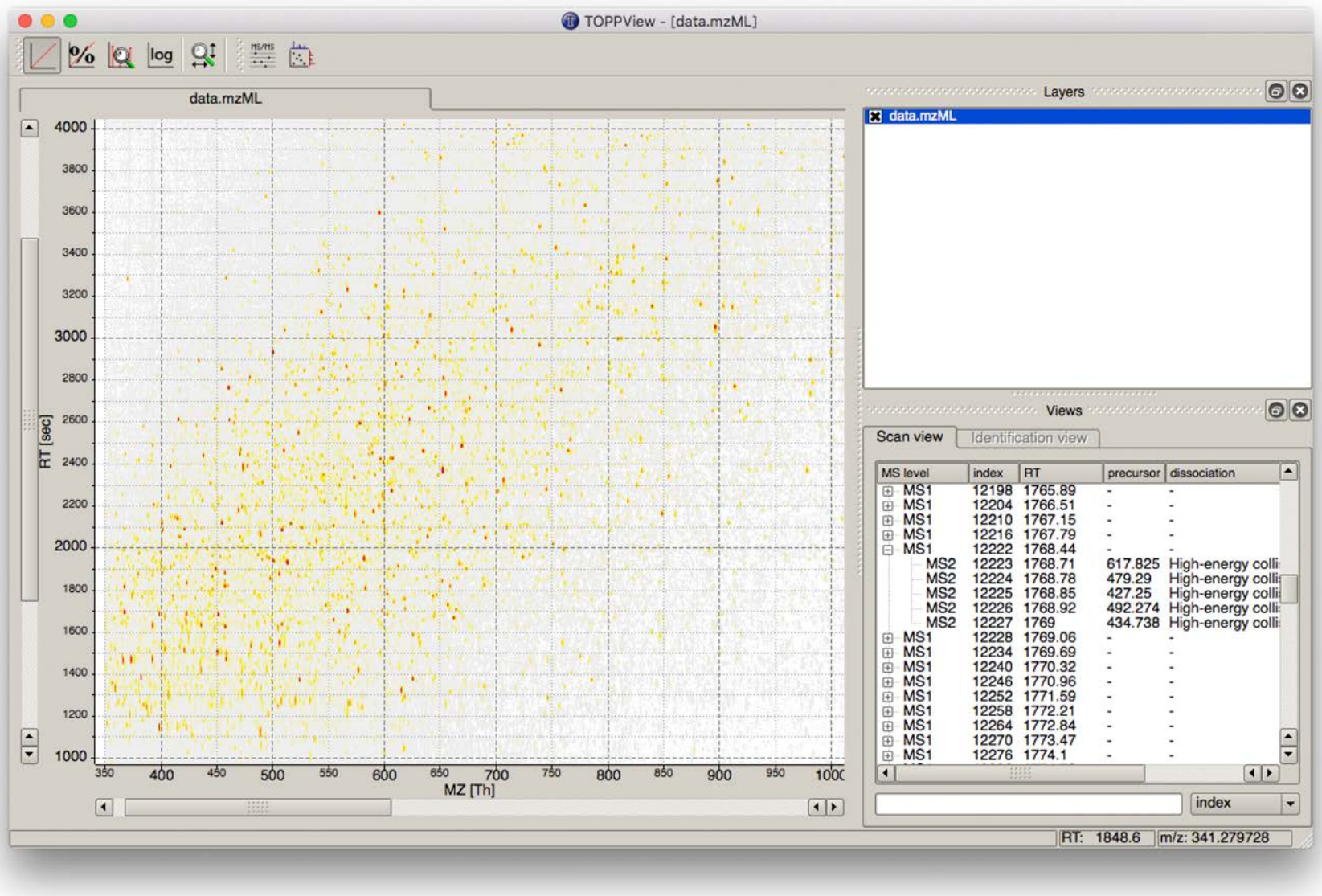

Figure 12.2 Data viewer TOPPView. The main part visualizes experimental data in the *m*/*z*-RT plane. The right panel contains the list of opened raw data files (mzML) and results (idXML, featureXML and consensusXML) as well as a list of all spectra.

The conversion to mzML can lead to a significant increase in file size and read–write time. This fact can be alleviated by using MS-Numpress compression of mzML[9] which is supported by both ProteoWizard and OpenMS. Recent improvements in parsing speeds and support for indexed mzML have further decreased the mzML file access times.[10]

For the peptide identification workflow we require an appropriate target-decoy database as input (see Chapter 4). Using the DecoyDatabase tool we can easily construct one, Figure 12.3(A). As input we specify a fasta file containing all possible target protein sequences for the search. The tool either shuffles or reverses these sequences and appends them to the input file. The resulting output file now contains both target and randomized decoy sequences, with decoys denoted by a specific prefix or suffix in their name.

In the peptide identification workflow, we can now specify both required inputs, the mass spectrometry data in mzML format and a target-decoy fasta database, and start the workflow, Figure 12.3(B). The data are first sent to an external search engine, in this case X!Tandem. The XTandemAdapter tool is a simple wrapper and provides access to all parameter settings of the search engine. The tool returns a number of peptide-spectrum-matches (PSM) and their corresponding scores for each submitted MS2 level spectrum. At this stage, the sequences in the output are not matched to their corresponding protein(s). Neither is it clear whether the sequence stems

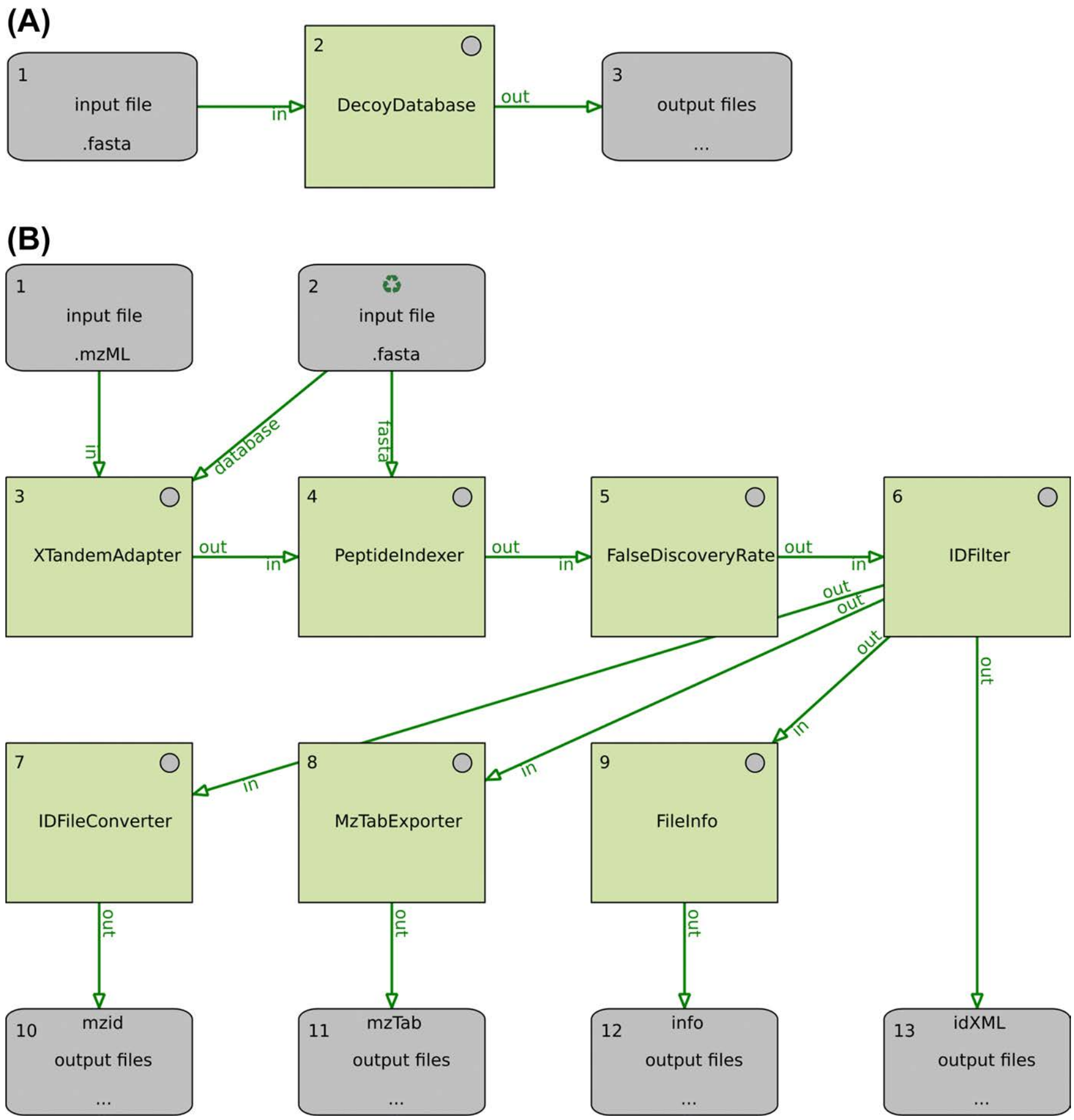

Figure 12.3 (A) Workflow for the construction of a target-decoy protein database (B) Workflow for peptide sequence identification and export to mzIdentML (mzid) and mzTab.

from a target or decoy entry. The following PeptideIndexer tool adds this information by referencing the sequences against the original search database. In the next step, the FalseDiscoveryRate tool can calculate the score distributions for target and decoy hits, and determine the false discovery rates (FDR) at both peptide and protein level. Finally, the PSMs can be filtered for specific peptide and protein FDR cut-offs in the IDFilter tool. The results of the workflow are stored in the OpenMS idXML format, Table 12.4. Alternatively, the final peptide identifications can be converted to the open standard mzIdentML[11,12] using the IDFileConverter tool, Table 12.5. In this format, the results can be validated in tools such as PRIDE Inspector[13] and

Table 12.4 Primary file formats supported by OpenMS.

Format	Description
mzML	Raw mass spectrometry data.[5] Numpress compression[9] and indexed mzML[10] supported ((HUPO–PSI format))
idXML	Peptide and protein level identifications (OpenMS specific format)
featureXML	Individual peptide features (OpenMS specific format)
consensusXML	Groups of quantified peptide features. For example, SILAC peptide pairs or groups of corresponding peptides in label-free quantifications (OpenMS specific format)
trafoXML	Retention time transformations (OpenMS specific format)
fasta	Protein sequence databases
toppas	OpenMS workflows and parameter settings (OpenMS specific format)

Table 12.5 Further supported file formats in OpenMS.

Format	Description
mzXML, mzData	Legacy formats for raw mass spectrometry data[67,68]
mgf	Mascot generic format, raw mass spectrometry data
mzTab	Text format for mass-spectrometry-based proteomics and metabolomics results[17,18] (HUPO–PSI format)
mzIdentML	XML format for identification results of mass-spectrometry-based proteomics studies.[11,12] Used for PRIDE submissions (HUPO–PSI format)
mzQuantML	XML format for quantitative results of mass-spectrometry-based proteomics studies[69] (HUPO–PSI format)
qcML	XML format for quality control metrics from mass-spectrometry experiments[59] (HUPO–PSI standard format)
TraML	XML format for Selected Reaction Monitoring (SRM) transition lists[70] (HUPO–PSI standard)
pepXML	XML format for peptide level results of mass-spectrometry-based proteomics studies[71,72] (Trans-Proteomic Pipeline format)
protXML	XML format for protein level results of mass-spectrometry-based proteomics studies[71,72] (Trans-Proteomic Pipeline format)

then uploaded to public repositories such as the Proteomics Identifications (PRIDE) database[14,15] and other projects in the ProteomeXchange Consortium.[16] For further post-processing of the sequences a simple text-based format is the better choice. The MzTabExporter tool can write the sequence information to the open mzTab standard.[17,18] This file can be viewed in any text editor and further analyzed in tools such as R or Windows Excel®. Often a quick overview instead of the complete results is sufficient. The FileInfo tool can generate a short summary with the main statistics of an analysis (number of PSMs, number of unique sequences *etc.*). The tool can generate summaries not only for identifications but any of the first four primary OpenMS file formats listed in Table 12.4.

The described workflow can easily be modified. For example, we can replace X!Tandem with another search engine, by simply replacing the

search engine adapter. OpenMS provides adapters for Mascot,[19] MS-GF+,[20,21] OMSSA,[22] MyriMatch[23] and InsPecT.[24] The remaining workflow does not need to be changed. As we will see later on in the chapter, often we can re-use complete and well-tested modules in other workflows and projects, and therefore speed up the development time of bioinformatics solutions.

The modular workflow design provides us not only with great flexibility, but transparency. Each intermediate step of an analysis can be inspected and therefore easily optimized. Each protein identification, each reported fold change in the final result can be traced back to the specific set of spectra they originate from. For example, after opening the mzML spectral data and the idXML identification result in TOPPView, we can link each precursor fragmentation to the corresponding peptide sequence in the result, Figures 12.4 and 12.5.

12.3 iTRAQ Labeling

After the simple peptide identification example in the last section, we now turn to a complete proteome comparison workflow and re-analyze a previously published quantitative proteomics dataset. In a recent study,[25] Subbannayya *et al.* used 4-plex iTRAQ labeling[26] (see Chapter 8) for the identification of gastric adenocarcinoma biomarkers in blood serum. Serum samples from ten patients and ten controls were pooled and subsequently trypsin digested. The control and carcinoma samples were labeled with iTRAQ reagents (114, 115) and (116, 117) respectively. After strong cation exchange (SCX) fractionation, the samples were measured on an LTQ Orbitrap Velos mass

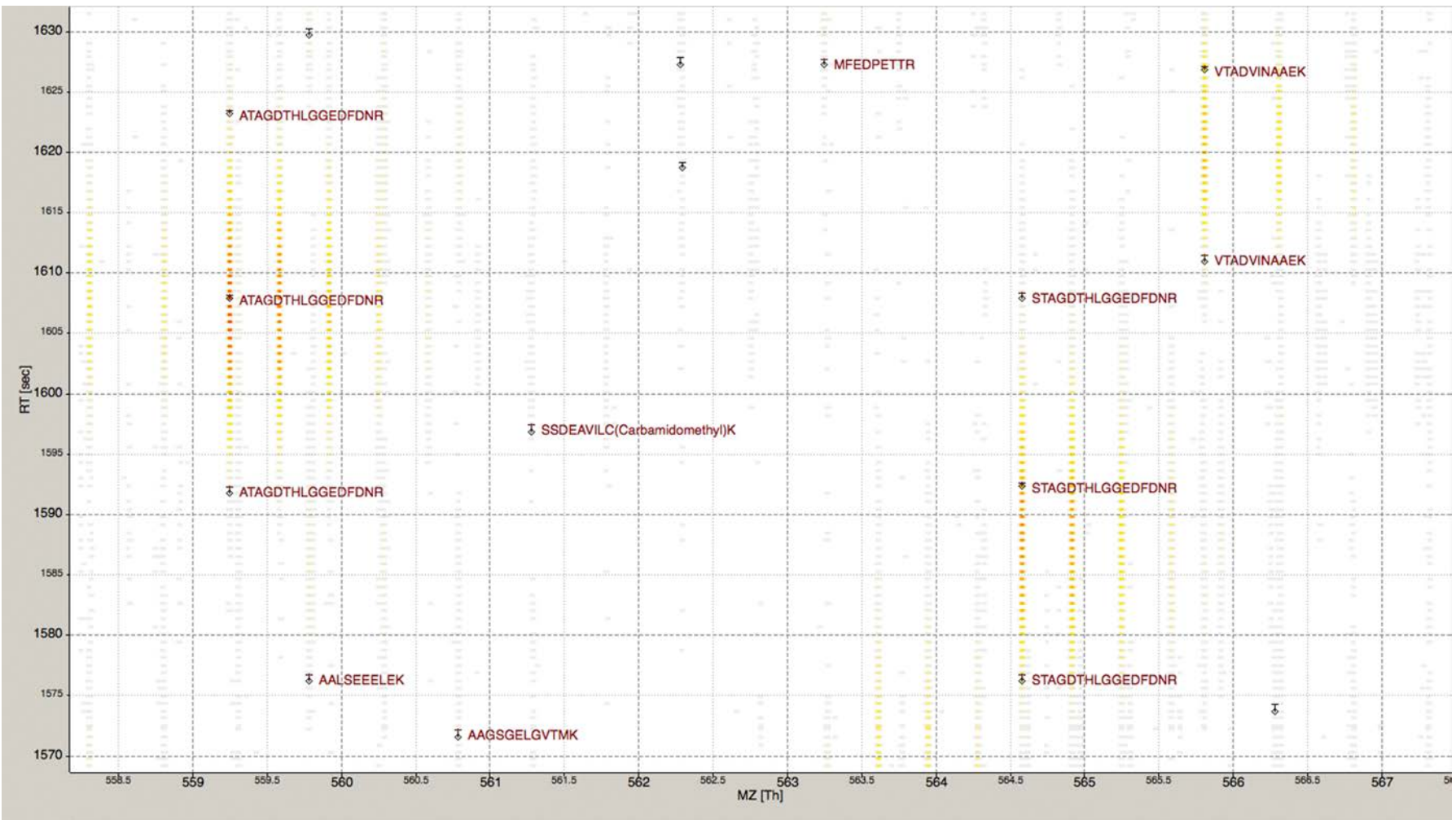

Figure 12.4 Experimental data (mzML) and peptide identifications (idXML) in TOPPView. Note that three of the peptides were fragmented multiple times leading to the same peptide sequence.

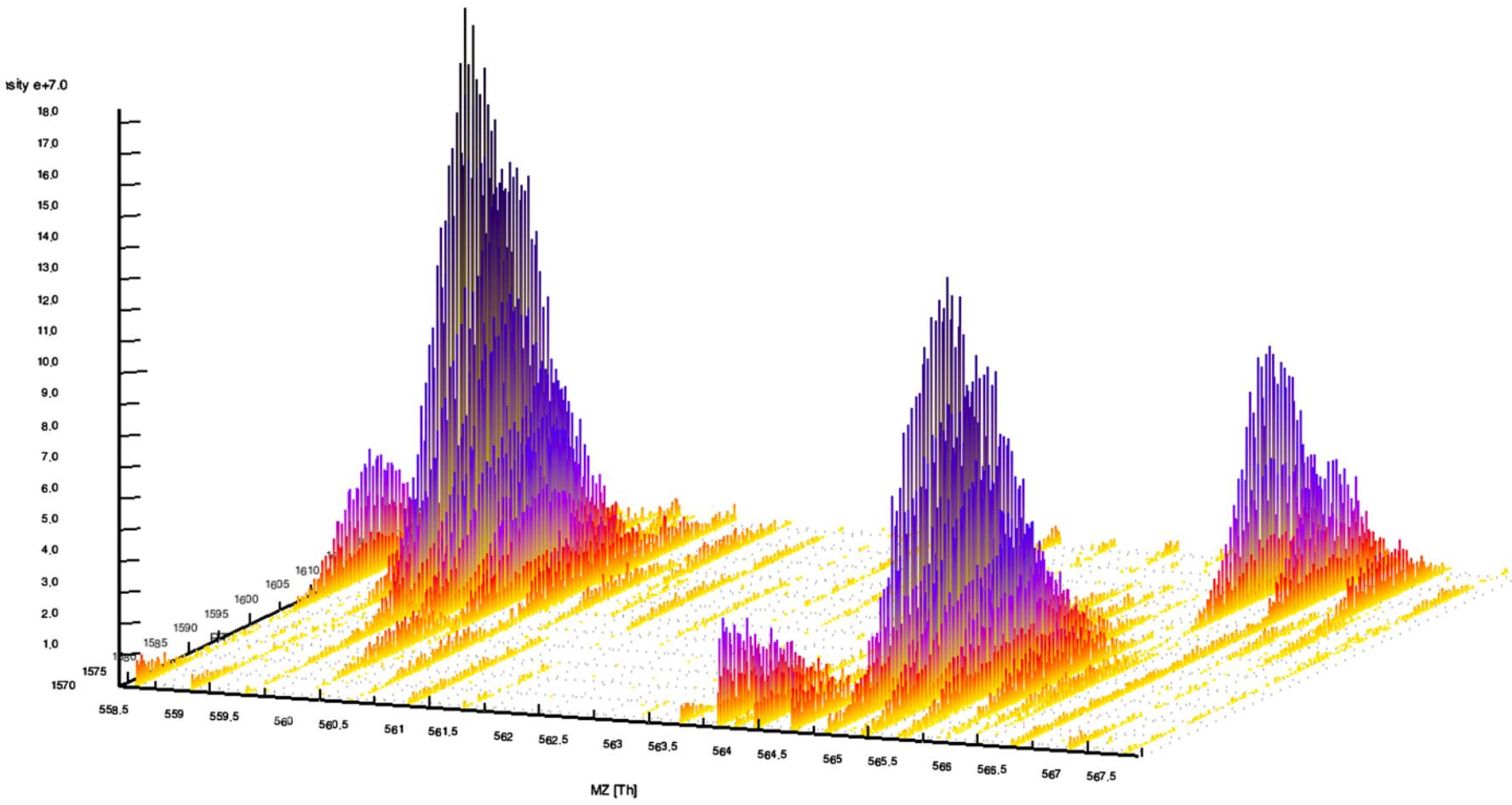

Figure 12.5 The same experimental data as in Figure 12.4 in 3D view in TOPPView.

spectrometer (Thermo Scientific). The mass spectrometry data are available *via* the PRIDE[14,15] archive under identifier PXD001265.

The authors analyzed the data with Proteome Discoverer® (Thermo Scientific, version 1.3.0.339) in combination with the two search engines Mascot (version 2.2) and Sequest-HT®.[27] A 1% FDR cut-off at peptide level was applied. The analysis resulted in 643 quantified proteins, 48 of them up-regulated (fold change, fc > 1) and 11 of them down-regulated (fc < −1). Among the up-regulated proteins, the authors identified seven biomarkers of particular interest: ITIH4, MBL2, SHBG, SAA1, ORM1, SOD3 and IGFBP2.

The re-analysis of the data combines OpenMS components with the external search engine MS-GF+[20,21] and the protein inference algorithm Fido,[28,29] Figure 12.6. The OpenMS workflow quantifies 722 proteins. Forty of them show an up-regulation (fc > 1) and 8 of them a down-regulation (fc < −1). Six out of the seven up-regulated biomarkers in ref. 25 are corroborated by the new analysis, Figure 12.7A. The protein IGFBP2 was not quantified. In conclusion, the re-analysis quantifies +12% additional proteins with a more conservative fold change distribution.

The OpenMS workflow comprises of tools for identification, numbered (3) to (12), and quantification, numbered (13) to (18), Figure 12.6. The peptide identification module, (3) to (6), is nearly identical to the previously discussed workflow. Merely the search engine was interchanged. In steps (7) to (12), we will now determine the complete set of identified proteins in the sample. In preparation for the protein inference algorithm, we first calculate the posterior error probabilities[30] for both good and bad PSMs. Different fractions are then combined and sent *via* an adapter to the external protein inference algorithm Fido. Its performance is similar or better than the performance of the well-established ProteinProphet tool.[29,31] For a more detailed discussion, let us consider the simple example in Figure 12.7B. From a set of identified

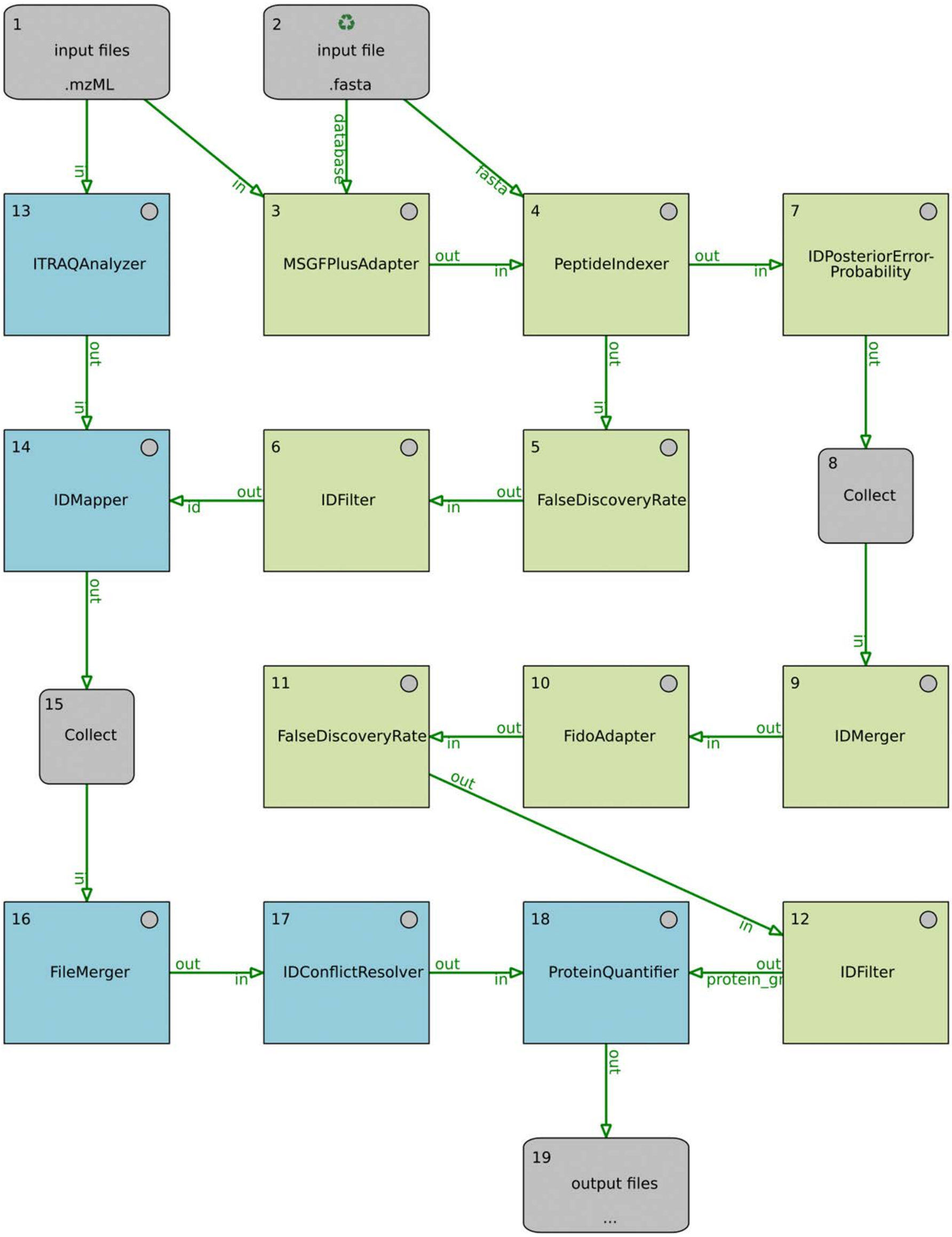

Figure 12.6 Workflow for the relative protein quantification of iTRAQ labeled samples. It combines tools for peptide and protein identification (3–12) and tools for quantification (13–18).

peptide sequences 1 to 8, the algorithm infers the presence of proteins A to E in the sample. Solid and dotted lines link peptides and their corresponding proteins. Peptides 1, 2, 4, 6 and 7 are proteotypic *i.e.* their sequence appears in only one of the five proteins. Two proteins are members of the same protein group if they share one or more peptide sequences. In our example, this results in two groups [A, B, C] and [D, E] of identified proteins.

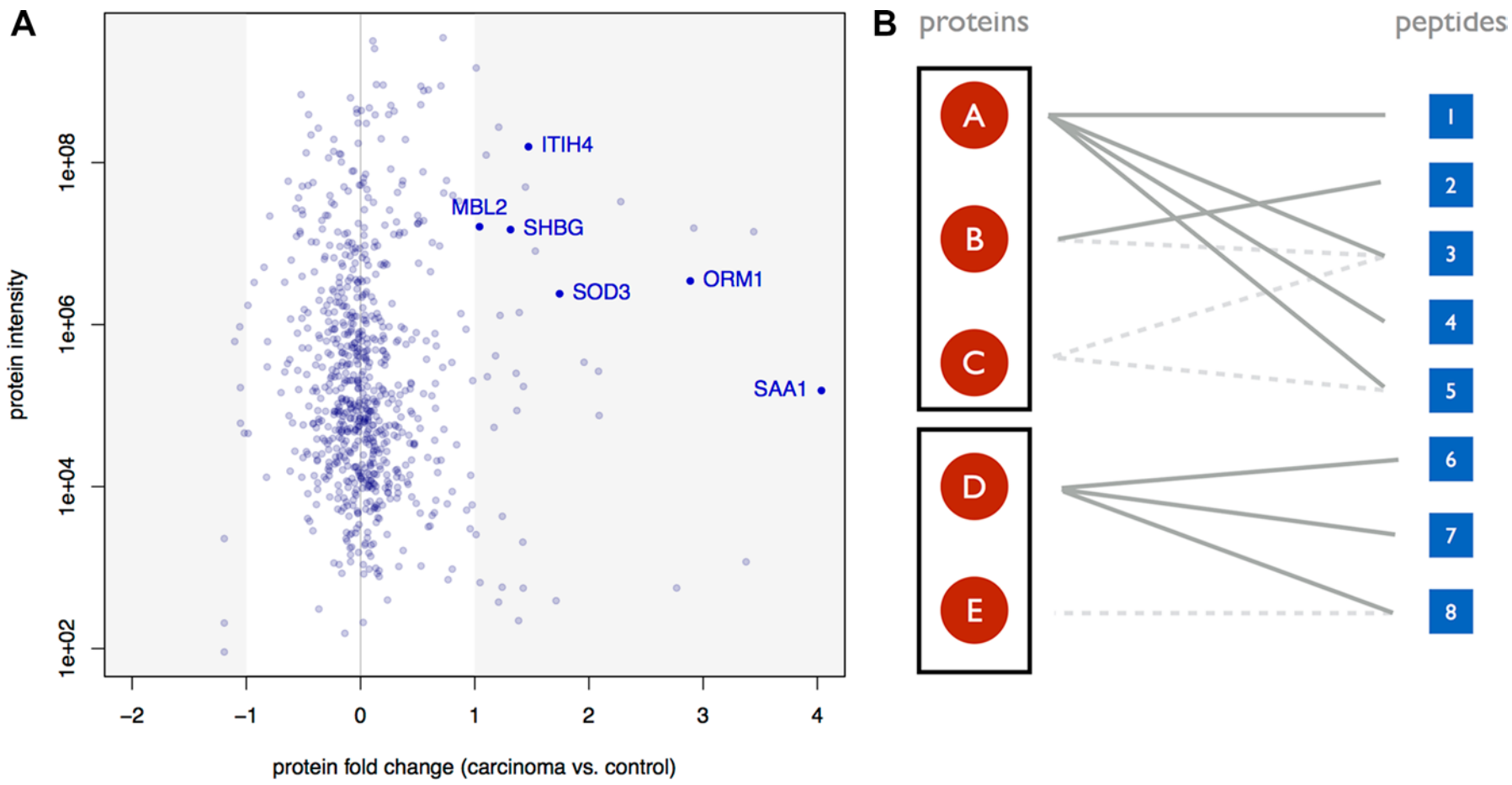

Figure 12.7 (A) Protein fold changes *vs.* protein intensities. (B) Protein inference and quantification.

The identification of these proteins does not automatically imply that all of the proteins are also quantified. Here the user can choose from three different options: *all*, *greedy* and *proteotypic*. In the first case, all proteins are quantified based on the quantitative information from all corresponding peptides. For example, protein A is quantified based on abundances of peptides 1, 3, 4 and 5, and C based on 3 and 5. All five proteins are quantified. But this approach has a downside. Many protein quantifications will be unreliable due to shared peptides such as sequence 3.

On the other hand, the *proteotypic* option is very reliable. In this case, only proteotypic peptides are used for quantification. For example, protein A is quantified based on peptides 1 and 4, B based on 2, and D based on 6 and 7. Proteins C and E are not quantified. A clear disadvantage is the lower number of proteins quantified. Many of the peptides, such as sequence 3, will not contribute to the protein quantification at all, although they might have been reliably identified and quantified.

The second option strikes a good compromise between these two cases. In the *greedy* method, each of the peptides is used exactly once; see solid lines in Figure 12.7B. Let proteins A to E be ranked by their protein score, *i.e.* A is the most and E the least reliable of the protein identifications. Starting with A, each protein uses as many peptide quantifications as are still available. For example, peptide 5 is used for the quantification of A, but not C, since we are more confident that A is present in the sample than C. In this way, the entire set of peptides is used but not all of the identified proteins are also quantified. This *greedy* approach was adopted for our re-analysis of the adenocarcinoma dataset.[25]

Both the protein inference *via* Fido and the *greedy* post-processing of the resulting protein list are performed in step (10) of the workflow. After FDR filtering at protein level in (11) and (12), the proteins and their corresponding

peptide evidences are handed over to the ProteinQuantifier tool. The protein quantification tool requires a second input: the peptide quantifications from steps (13) to (17). The iTRAQAnalyzer detects the reporter ions at 114, 115, 116 and 117 Th[†] in all MS2 level spectra, and adjusts their intensities using a correction matrix provided by Sciex. In step (14), the IDMapper tool then maps the reporter ion intensities to the corresponding peptide identifications from the same MS2 spectra. After combining all fractions, the peptide sequences and abundances are then passed to the ProteinQuantifier tool. Finally, we have all necessary information and compile a list of quantified proteins.

The described iTRAQ workflow is optimized for data from LTQ Orbitrap Velos mass spectrometers and available for download from the repository http://openms.de/workflows. It can easily be adjusted for data from other machines by changing the parameter settings in the search engine adapter. Alternatively, it can be turned into a Tandem Mass Tag (TMT)[32] workflow by replacing the iTRAQAnalyzer by the TMTAnalyzer tool. The workflows in the repository should be considered as templates, which users can easily adapt for their specific projects and needs.

12.4 Dimethyl Labeling

In the two workflows discussed so far, both identifications and quantifications were based on MS/MS spectral data. We will now focus on MS-based quantification techniques with stable isotope labeling. Here we have chosen dimethyl labeling,[33] but with minimal changes this workflow will work equally well for SILAC (stable isotope labeling by amino acids in cell culture)[34] and ICPL (isotope-coded protein label) data.[35,36]

Our test dataset is simple. We used formaldehyde-based dimethylation of primary amines, *i.e.* peptide N-termini and lysine side chains, for the light (+28 Da) and heavy (+34 Da) labeling of a human embryonic kidney (HEK) cell lysate. The two differently labeled samples were mixed with a fixed ratio of heavy : light = 1 : 4, *i.e.* a fold change, fc = −2. Without prior fractionation, the sample was analyzed on a Q Exactive Orbitrap mass spectrometer (Thermo Scientific). As in the previously discussed iTRAQ workflow, we use the MS-GF+ search engine[20] and the Fido protein inference algorithm,[28] Figure 12.8. The analysis results in 532 protein quantifications with fold changes clustering around the expected value of fc = −2, Figure 12.9.

SILAC, ICPL and dimethyl labeling are widely used techniques for quantitative discovery proteomics studies.[37,38] The shotgun approach provides a good coverage of the entire proteome, MS1-based quantifications yield reliable protein fold changes which are often confirmed by multi-reaction-monitoring (MRM) follow-ups. Key, in these experiments, is the sensitive and accurate detection of peptide features in the mass spectrometry data.[39] In the previous discussion of Section 12.2, we already encountered peptide features. Figures 12.4 and 12.5 show three clear features, each eluting for between 20 and

---

[†] [m/z] = Th = Da/e, see https://en.wikipedia.org/wiki/Thomson_(unit).

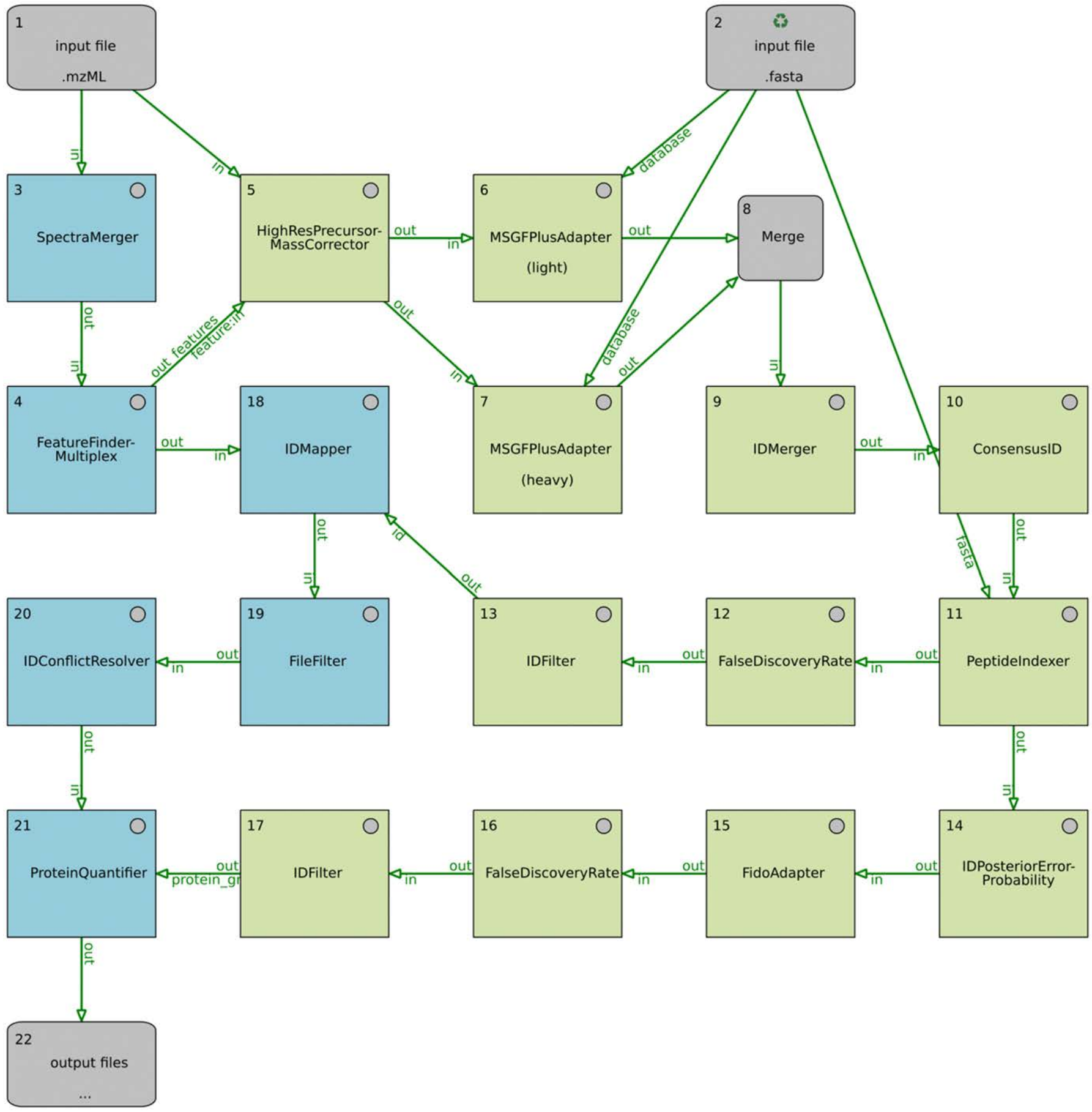

Figure 12.8 Workflow for the quantification of dimethyl labeled samples.

30 seconds and with three to five peaks in their isotopic patterns. We focused on their MS2 fragment spectra and corresponding amino acid sequences, and for the analysis in this section we will also detect them at MS1 level. Their abundances will form the basis of the final protein quantifications.

The peptide features in our test dataset appear in pairs, due to the light and heavy dimethyl labeling. The TOPPView screenshot in Figure 12.10 shows two clear peptide feature pairs with amino acid sequences VLQATV-VAVGSGSK and YSQVLANGLDNK. In both cases, the N-termini and lysines are tagged with either light or heavy dimethyl modifications. Since both peptides are doubly charged, this results in a relative m/z shift of (6.031 Da + 6.031 Da)/2 C = 6.031 Th. The heavy partners show a lower intensity than their light counterparts, due to the fixed mixing ratio of 1 : 4. The TOPPView screenshot shows three different layers: the original spectral data (mzML), the FDR-filtered peptide sequences (idXML output of step (13)) as well as the

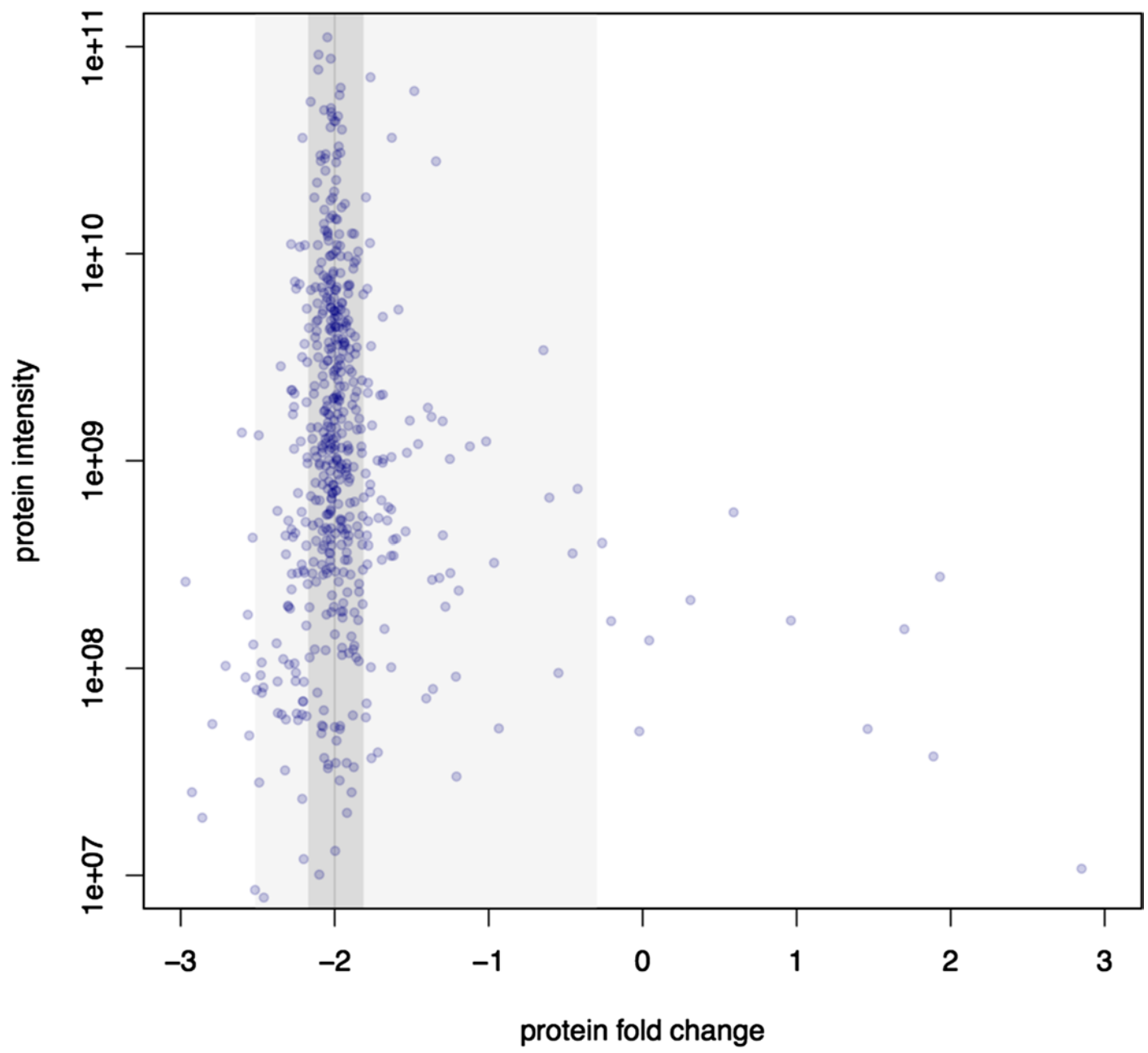

Figure 12.9 Protein fold changes *vs.* protein intensities dimethyl labeled HEK cell lysate sample with fixed ration of fc = −2. One and two sigma deviations in the fold change are shaded in dark and light grey respectively.

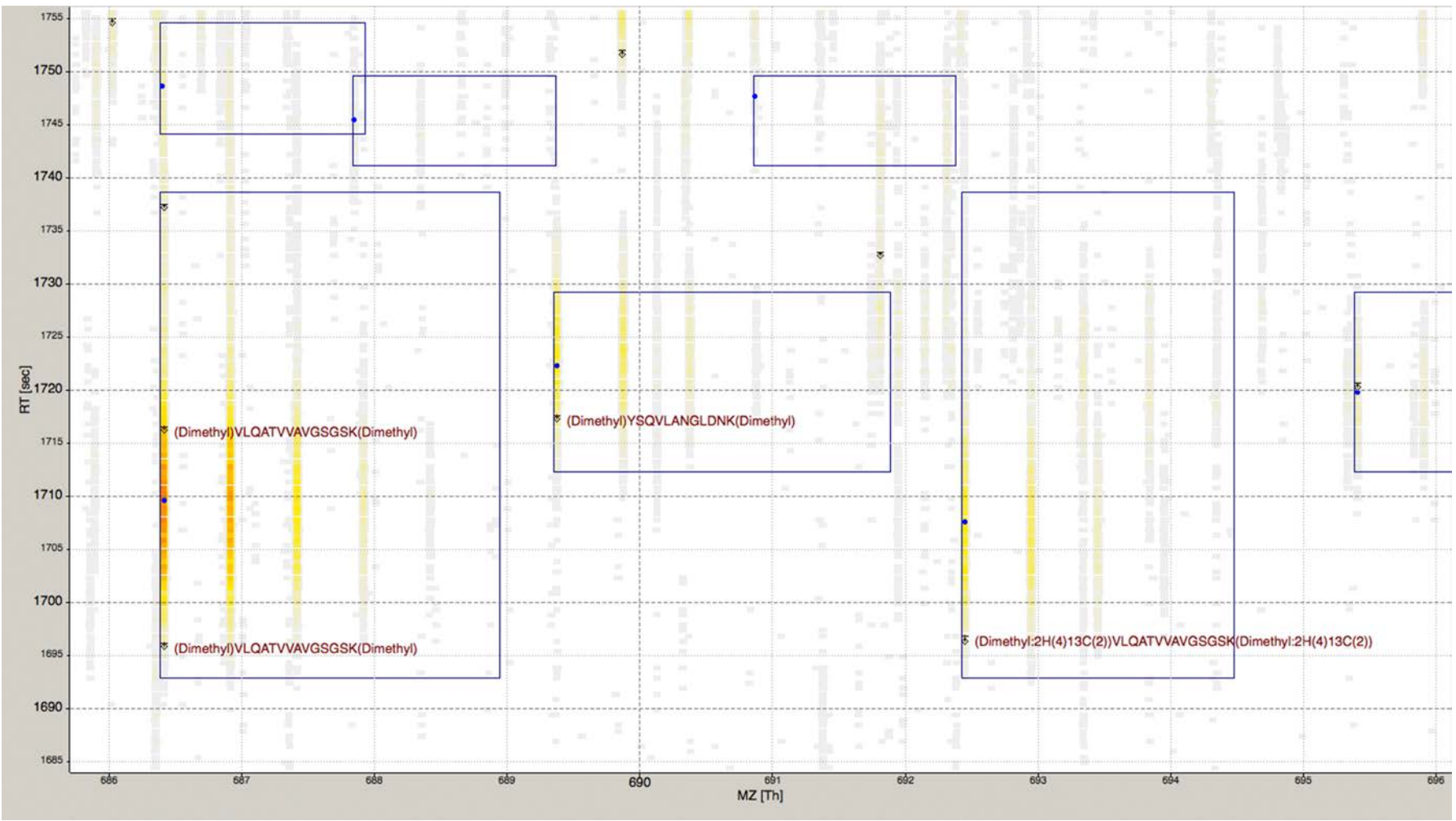

Figure 12.10 Two peptide feature pairs with relative m/z shifts of 6 Th. The TOPPView screenshot shows the raw spectral data (mzML), FDR filtered peptide sequences (idXML) and the detected peptide features (featureXML).

detected features (featureXML output of step (4)). We will start our discussion of the workflow with the feature detection and return to the sequence identification later on.

The SpectraMerger tool in step (3) prepares the spectral data for the subsequent feature detection step. The tool averages neighboring MS1 spectra within a specified retention time (RT) range. Signals from true peptides reappear in neighboring spectra and are amplified, while background noise fluctuates and is suppressed. The moving average increases the signal-to-noise ratio and results in smoother peptide elution profiles in RT direction. For the feature detection, OpenMS provides a number of different algorithms. Here we have chosen the FeatureFinderMultiplex tool.[39–41] It can detect single peptides, but also peptide pairs, triplets *etc.* in the spectral data (hence multiplex). Searching directly for pairs of peptides instead of single peptides has an advantage. Two isotopic peak patterns with a fixed relative shift provide a clearer search pattern than a peak pattern from a single peptide. False positive detection from the background noise is therefore less likely and the search more sensitive. In step (18), the peptide features are then annotated with peptide sequences from the identification part of the workflow. The following steps resemble the ones in the iTRAQ workflow. Features without sequence annotation are removed, and multiple sequence annotations resolved. The remaining peptide feature pairs, each now annotated with a single sequence, are then passed to the protein quantification. Note that the workflow contains no FileMerger tool. The sample was not fractionated and consequently the merging of multiple mass spectrometry runs is not necessary.

The identification branch of the workflow begins with a pre-processing step. In shotgun experiments, the most intense peaks of each MS1 spectrum, typically 10, are fragmented resulting in MS2 spectra from which the peptide sequences are deduced. The isolation window around the precursor ion positions is relatively large, about ± 2 Th.[42] Small errors in the precursor position are therefore not crucial. The mono-isotopic peak of a peptide will be fragmented and its fragments part of the MS2 spectrum. On the other hand, many search engines assume the precursor mass to be identical with the mono-isotopic parent mass. Deviations between the two can lead to missing identifications or incorrect scoring. Some search engines therefore provide correction parameters in order to account for this possible discrepancy. In our analysis workflow, we can correct any mistaken precursor positions before the search. From the feature detection we already know the position of all peptides and their mono-isotopic masses. The HighResPrecursorMassCorrector tool in step (5) simply shifts any incorrect precursors to the true mono-isotopic peak positions.

The search strategy we follow in this workflow differs slightly from what we have encountered in the previous two analyses. We already know that each peptide in our sample is modified. The N-termini and lysines are either lightly or heavily dimethylated. One possible option is to run a single search and specify light and heavy dimethylation as variable modifications. But this

approach is not ideal since it also allows for entirely unmodified peptide sequences in the result. Instead, we run two separate searches with different modifications: one with a fixed light dimethylation and one with a fixed heavy dimethylation. An additional benefit is the decrease in processing time, since fixed searches are significantly faster than variable ones. The two search results are then combined in steps (8) to (10). In case both searches report a peptide hit for the same MS2 spectrum, the ConsensusID tool[43] assigns the PSM with the better score to that spectrum.

But the tool is even more versatile. In our analysis, we merged two different searches from the same MS-GF+ search engine. ConsensusID is also able to combine search results from entirely different search algorithms. Using sequence similarity scoring mechanism, the algorithm estimates scores for sequences that are not reported in one of the results. From this complete score matrix it can then construct a carefully weighted consensus score for each of the reported peptide sequences. Combining results from multiple search algorithms in this way can further increase the number of peptide and protein identifications. The basic concept behind ConsensusID is similar to that of iProphet, MSblender or PepArML.[44–47] The remainder of the workflow is identical to what we have encountered and discussed in the previous two examples. In steps (11) to (13), the peptide sequences are FDR-filtered and subsequently mapped to the detected peptide feature pairs. Apart from the minor precursor correction, the peptide identification and quantification are performed independently from each other. Unlike in other approaches, the feature detection is not based on any sequence information. The mapping in step (18) serves as an important cross-check. Only peptides, for which identifications and quantifications conform, Figure 12.10, are passed on to the protein quantification. Finally in steps (14) to (17), a protein list is constructed, again FDR-filtered and then provided to the protein quantification. As we have shown, the reported protein fold changes reflect the fixed mixing ratio of our test sample, Figure 12.9.

The depicted workflow in Figure 12.9 contains a single output node in (22). Note that we can add further outputs at any step of the workflow and thereby check the intermediate results. For example, the result of node (13) can be piped to an additional output or exported to mzTab and mzIdentML, as done in our first discussed workflow, Figure 12.3(A). In this way, the FDR-filtered peptide identifications can be inspected in TOPPView (idXML) or any text editor (mzTab). Similarly, we can have a closer look at the peptide level quantifications. The output of node (20) contains all peptide feature pairs that will contribute to the protein quantification (consensusXML). Often the user is interested in not all but one specific protein. We can further filter the output of node (20) using the simple workflow in Figure 12.11. The FileFilter tool is very versatile and contains among other parameters a protein accession whitelist. Only peptides that occur in the sequence of one of these whitelist proteins will pass the filter. The evidence for each individual reported protein fold change can therefore be conveniently checked and traced back to the original spectral data.

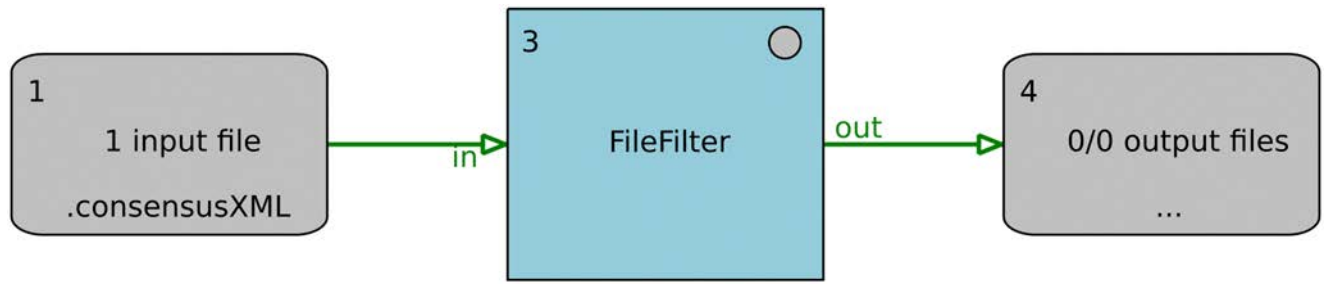

Figure 12.11 Workflow for the filtering of peptide level quantifications. Only peptides belonging to specified whitelist proteins can pass the filter.

12.5 Label-Free Quantification

In our final protein quantification workflow, we dispense with labels altogether. In the label-free approach, multiple samples are measured in independent mass spectrometry runs. The technique relies on the fact that identical peptides will elute at the same or similar times in each of the runs. In the subsequent data analysis, corresponding peptide features are matched, and their abundances in the different runs can be compared. The approach has several advantages over the previously discussed techniques: a less complex sample preparation, an increased number of samples that can be compared simultaneously, and a more complete coverage of the proteome.[37,48] A peptide needs to be fragmented and reliably identified in only one of the shotgun runs in order to contribute to the protein quantification. On the other hand, the data processing is more challenging and can have a major impact on the final results.[1] The alignment and correct matching of corresponding peptide features,[49] and the subsequent normalization of the feature intensities are two crucial steps in the analysis.[50,51] We will focus on these two points in the following discussion of the OpenMS workflow, Figure 12.12.

As proof of concept, we re-analyze a previously published spike-in dataset.[52] The authors prepared two whole cell lysates: one from a bacterial cell line (*Streptococcus pyogenes*, strain SF370) and one from a human cell line (fetal lung fibroblasts, HFL-1). The two samples were mixed in six different ratios, Table 12.6, with increasing concentration of the bacterial sample and decreasing concentration of the human sample. The six mixtures were then measured on an LTQ Orbitrap XL instrument (Thermo Scientific) with a 75 min gradient. The OpenMS analysis results in a total of 496 quantified proteins. They display the expected linear correlation between sample concentration and reported protein intensities. Two examples with medium protein abundance and good correlations $R > 0.96$ are shown in Figure 12.13(A). The correlation and average protein abundances are plotted in Figure 12.13(B). As expected, the detection of the linear behaviour becomes more difficult as the protein intensities decrease.

The label-free quantification workflow is the most complex of the four workflows we present in this chapter, Figure 12.12. Fortunately, it is also the easiest one to explain since most of its components were already discussed previously. The analysis starts with a database search using the MS-GF+ search engine and a target-decoy database containing both human and bacterial

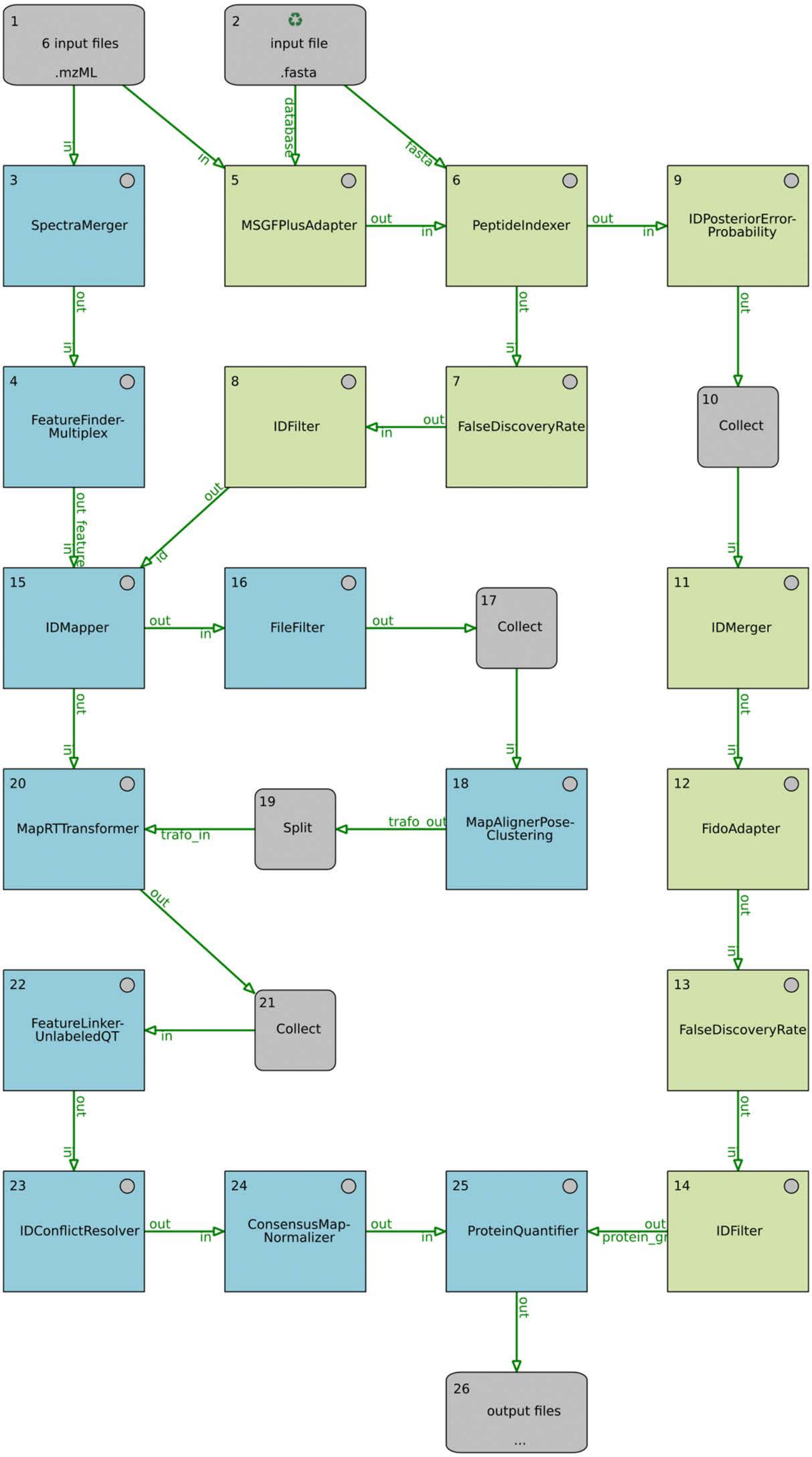

Figure 12.12 Workflow for label-free protein quantification.

Table 12.6 Label-free test dataset. Relative concentrations of the bacterial and human samples in the six mixtures.

Streptococcus pyogenes (%)	Human (%)
0	100
20	80
40	60
60	40
80	20
100	0

proteome. A correction of precursor positions is not necessary for LTQ Orbitrap data. FDR-filtered peptide identifications are generated in steps (5) to (8). In steps (9) to (14), the unfiltered PSMs from the six runs are first combined and then a set of proteins inferred and FDR-filtered. The quantitative part of the workflow starts again with the smoothing of the MS1 spectra in retention time direction using a simple moving average. The peptide features are then detected and annotated with the sequence information. After step (15), we have six sets of peptide features (featureXML), one for each sample.

We now face the challenging task of matching corresponding peptide features between the six different runs. Some of the peptide features will have a sequence annotation, but many of them will not. This is because their MS2 fragment spectra are too sparse for a reliable identification, or the features were not fragmented in the first place. Hence, we cannot make use of the peptide sequence annotation to find corresponding features. Instead we will rely entirely on the retention time (RT) and mass-to-charge ratio (m/z) of each peptide feature. Modern high-resolution instruments can measure m/z positions very precisely. The deviation between mass spectrometry runs is minute. On the other hand, the time at which a peptide elutes from the column can vary significantly from run to run. We need to correct for these RT shifts by first aligning the six mass spectrometry runs. For this task only high-intensity features, which are repeatedly detected in all six runs, hold valuable information. We filter for these high-intensity peptides in step (16) and thereby simplify the alignment problem. A much clearer feature pattern is now repeated in each of the six feature sets. These sets are then collectively passed to the alignment algorithm that determines the relative RT shifts between the runs (trafoXML). These transformations are then applied to the entire, unfiltered feature sets in step (20). Corresponding peptide features now have the same m/z-RT position in each of the six runs. Matching features are then linked to so-called consensus groups (consensusXML). Each group contains between one and six features. Some features in a group are annotated with a sequence, others are not. If both alignment and linking worked perfectly, the consensus groups contain no conflicting sequence annotations. Any remaining conflicts are resolved in step (23) by retaining only the highest-scoring identification in each group.

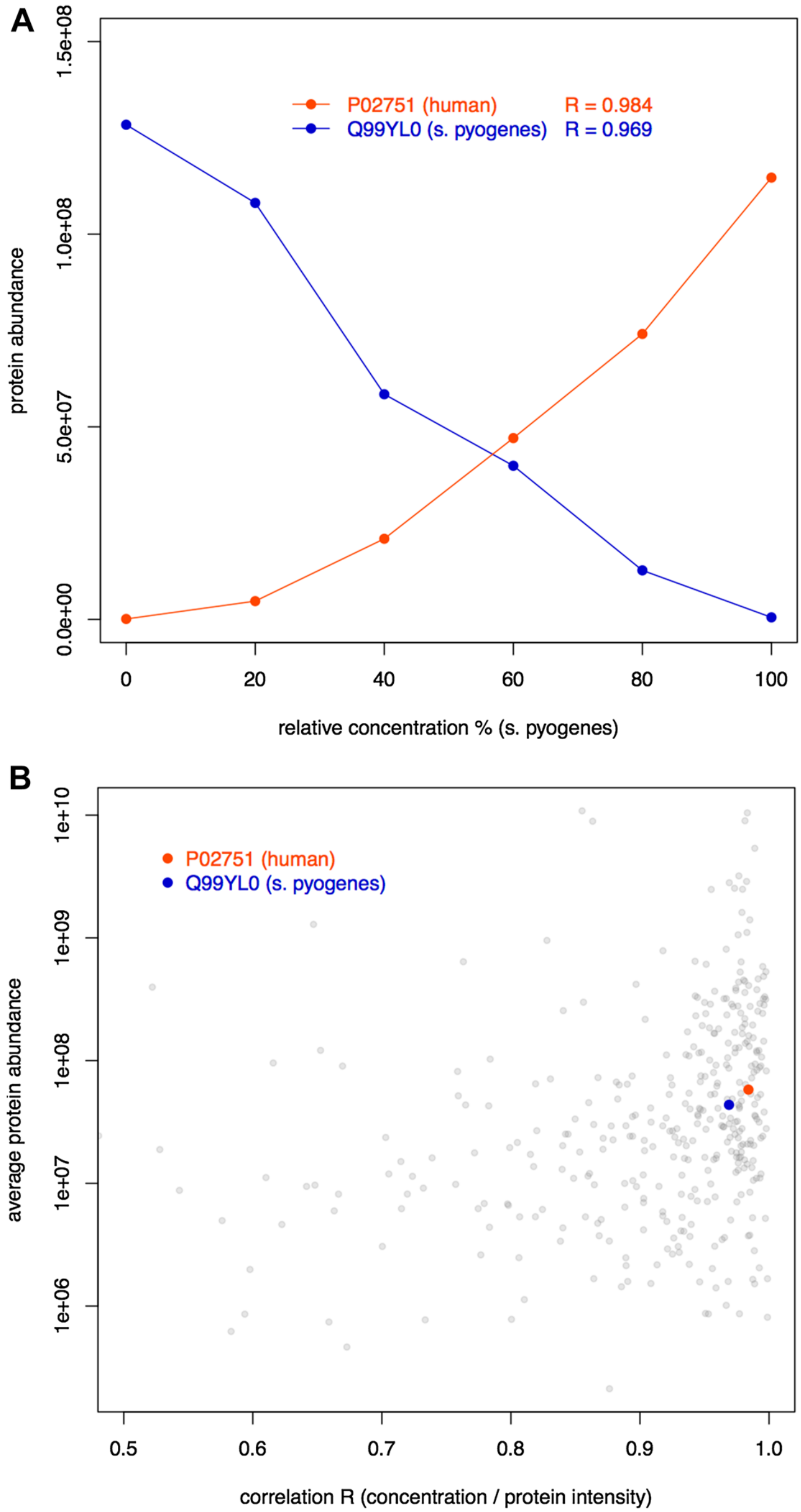

Figure 12.13 (A) Protein abundance *vs.* relative spike-in concentration % (*Streptococcus pyogenes*) for proteins P02751 and Q99YL0, see Table 12.6. (B) Average protein abundance *vs.* correlation (concentration/protein abundance) for all quantified proteins.

Unlike in labeled quantification techniques, the peptide features within a consensus group originate from different mass spectrometry runs. Changes in the liquid chromatography and spray instabilities can easily result in additional peptide intensity variations, which do not reflect the peptide abundances in the samples. The normalization of the peptide intensities is therefore an important part of any label-free data analysis. Various approaches have been proposed and implemented.[50,51,53] The ConsensusMapNormalizer tool offers three different options: median correction, robust regression and quantile normalization. Median correction is the most simple and conservative of the three. The algorithm first determines the median intensity for the run with the most features. The intensities in the remaining runs are then scaled to the same median intensity. In robust regression, the feature intensities are normalized pair-wise relative to the run with the most features. Given two runs, features are classified as outliers and non-outliers. From the non-outliers an average fold change is calculated and used for normalization. The final quantile normalization is identical to the approach used in many microarray data analyses.[54] In step (24) of our workflow, we apply a simple median correction to all peptide feature intensities. Finally, the protein quantification step combines the quantitative information at peptide level and the inferred protein list. The final output is a list of 496 proteins and their abundances. As we have seen in Figure 12.13(B), these abundances show the clear linearity that we expect from our spike-in test dataset.

12.6 Conclusion

In this chapter, we described in detail four typical data analyses using the OpenMS workflow system. Our discussion focused on the flexibility and modularity of the analysis workflows. We hope the four use cases help new users to understand the basic principles such that they are then able to design new workflows on their own. On the other hand, all of the complexity can be readily ignored when using OpenMS workflows as each workflow is simply a black box with typically two inputs, the spectral data and a database, and one output, the list of quantified proteins. The repository at http://openms.de/workflows provides a range of already optimized workflows for specific tasks and specific mass spectrometry machines for download. For example, the previously discussed workflow in Figure 12.6 is optimized for 4-plex iTRAQ analyses of LTQ Orbitrap Velos data.

The workflow repository is continuously extended and updated by the OpenMS community. As new mass spectrometry machines and experimental techniques become available, new workflows will be uploaded to the repository. Often only minor changes are necessary to adapt an existing workflow for a different use. For example, the workflow in Figure 12.6 can be used for the analyses of iTRAQ data from an entirely different machine by simply adjusting the search parameters in the MS-GF+ adapter in step (3). Similarly, replacing the node (13) by a TMTAnalyzer tool turns the same workflow into a TMT analysis pipeline. In many other cases, merely changing the workflow

parameters is not sufficient. For example in the workflow of Figure 12.8, we added the two additional nodes (3) and (5). The smoothing of the MS1 spectra and the correction of the precursors are not necessary, but these two pre-processing steps clearly improve the analysis of the Q Exactive data. As a second example, consider the search strategy in steps (6) to (10) of the same workflow. We could have simply performed a single search with the dimethyl labels as variable modifications. Instead, we run two separate fixed modification searches and subsequently combine the results. In this way, we can ensure that the search result does not include any unmodified peptide sequences. Even in the context of well-established experimental techniques, a modular workflow system is of great advantage. Some components early on in the workflow can be fine-tuned in order to work optimally for a specific LC-MS, while later modules can be re-used in many other situations.

The benefits become even more apparent when we turn to novel experimental techniques. As an example let us consider a recent study[55] in which the authors identify binding sites in RNA–protein complexes using a combination of photo-cross-linking and high-resolution mass spectrometry. The mass spectrometry data were analyzed with an OpenMS workflow together with the search engine OMSSA.[22] Key, in the data analysis, was the generation of a list of precursor mass variants. The masses of all possible nucleotides were subtracted from the experimentally observed precursor masses. A subsequent database search of all MS2 spectra using all precursor mass variants would have required enormous processing power. Instead the MS2 spectra were carefully pre-filtered in order to reduce processing time. For a detailed discussion we refer to the manuscript.[55] Here we simply want to point out that many components in this workflow are identical to the ones we have previously discussed in this chapter. Examples are peptide feature detection, retention time alignment or sequence identifications. By re-using established and well-tested modules, the developers were able to focus on the novel aspects of their analysis pipeline.

In this chapter, we have presented only a small fraction of the over 100 tools in OpenMS. For example, the previously mentioned RNA–protein cross-linking workflow includes a number of important tools that we have not discussed. In what follows, we want to give a very brief overview of the tools that are not covered in this introductory chapter. They fall broadly into three major areas of application: SWATH data analyses, metabolomics and quality control reports. In all four of the discussed workflows, we analyzed data from typical shotgun LC-MS experiments. In this so-called data-dependent acquisition (DDA) approach, only high-intensity molecules are fragmented and subsequently identified. The data-independent acquisition (DIA) method[56] (see Chapter 10) is fundamentally different in that it fragments all ionized molecules in a sample. It is also known as Sequential Windowed Acquisition of All Theoretical Fragment Ion Mass Spectra (SWATH MS). The approach allows for an improved coverage of the analyte, but also increases the complexity of the data analysis. OpenSWATH[57] is a set of OpenMS tools for the automated analysis of DIA mass spectrometry data.

Liquid chromatography coupled to high-resolution mass spectrometry is not restricted to the field of proteomics. The technology is increasingly applied in metabolomics studies and the study of other small molecules. OpenMS provides a set of specialized tools for the detection of metabolite features in LC-MS data.[58] These tools can be readily combined with the label-free quantification modules discussed in detail in Section 12.5. The resulting workflows are well-suited for the fully-automated analysis of large-scale clinical metabolomics studies.

Finally, OpenMS offers a set of tools for the generation of modular quality reports. These reports can list the basic statistics of a dataset, and a wide range of different quality control metrics. For example, they can include total ion chromatograms (TIC), injection times, mass accuracy distributions, charge state distributions and nominal *vs.* fractional mass plots. The complete information of metadata, tables and plots can be exported to the qcML file format[59] and conveniently viewed in any web browser.

With this set of over 100 different tools, OpenMS provides a lot of functionality but anyone can contribute new tools and algorithms to the project if needed. OpenMS is an open-source software project hosted on the software repository GitHub https://github.com/OpenMS. The OpenMS library contains over 1300 C++ classes with extensive documentation. Stringent code review, continuous integration and a multitude of functional and unit tests ensure the robustness and continued support of the code. All data structures and algorithms of the library are also accessible over an interactive Python interface, pyOpenMS.[2] This interface provides users with basic programming skills, further flexibility beyond the workflow system, and is particularly suited for fast prototyping. OpenMS is covered by the permissive BSD license, and can therefore be used in both academic and commercial projects.

Not all steps of a data analysis need to be processed within the OpenMS framework itself. OpenMS collaborates and is integrated with many other software projects. As discussed in Section 12.2, OpenMS relies entirely on the ProteoWizard project to bridge between the proprietary vendor formats and the open mzML standard. For peptide identification, OpenMS relies on proven external search engines rather than a single internal solution. Alternatively, identifications can be directly imported from any academic or commercial software package that supports the pepXML, protXML or mzIdentML file standards. In the same way, the support of open file format standards facilitates the downstream statistical analysis of the OpenMS results. Users can choose from a range of tools such as R packages (see Chapter 15) MSnbase,[60] MSStats[61] and aLFQ,[62] or Skyline.[63] At no stage of the data analysis is the user locked in.

Throughout this chapter, we used the TOPPAS application for the design and execution of the analysis workflows, Figure 12.1. TOPPAS is a so-called workflow engine. It is independent of the individual OpenMS tools, and merely ensures their correct order of execution within a workflow. That means TOPPAS can be replaced by other workflow engines without altering

the workflows themselves. One possible alternative is the Konstanz Information Miner (KNIME).[64] OpenMS is fully integrated with KNIME *via* community nodes. The KNIME project provides further nodes for statistical data analysis, visualization, machine learning and various scripting languages such as R. That enables users to combine OpenMS analyses and statistical post-processing steps in a single, fully automated workflow.[65] A second alternative is the Galaxy workflow engine.[66] As in TOPPAS and Knime, the OpenMS tools are represented by individual nodes and can be combined to complex workflows. The Galaxy project was initially conceived as a platform for the analysis of genomics data but is gradually being extended to proteomics (see Chapter 14). With the addition of proteomics tools such as OpenMS, it is now an ideal framework for the data analysis of complex multiomics studies. In summary, OpenMS solutions can be deployed on a variety of different platforms. An OpenMS workflow designed on a small laptop computer can also be executed on a big KNIME server or a powerful Galaxy computing cluster. This scalability makes OpenMS solutions ideally suited for academic, clinical as well as pharmacological applications.

Modularity has been a reoccurring theme in this chapter. It underlies many characteristics of the OpenMS software framework. Data analyses become transparent, because intermediate steps can be inspected and optimized. Complete modules of tools can be re-used in different projects, and thereby reduce development times. External software solutions can be easily integrated. The software modules can be automatically tested, and therefore much better supported in the long run. Finally, the modularity can also be ignored, and workflows from the repository readily be used for standard data analyses. OpenMS provides a good platform for the analysis of today's proteomics and metabolomics data, and is flexible enough to adapt to changing requirements in the future.

Acknowledgements

The author would like to thank Oliver Schilling and Yasset Perez-Riverol for helpful discussions and valuable comments on the manuscript.

References

1. A. Chawade, M. Sandin, J. Teleman, J. Malmström and F. Levander, Data processing has major impact on the outcome of quantitative label-free LC-MS analysis, *J. Proteome Res.*, 2015, **14**, 676–687.
2. H. L. Röst, U. Schmitt, R. Aebersold and L. Malmström, pyOpenMS: a Python-based interface to the OpenMS mass-spectrometry algorithm library, *Proteomics*, 2014, **14**, 74–77.
3. O. Kohlbacher, K. Reinert, C. Gröpl, E. Lange, N. Pfeifer, O. Schulz-Trieglaff and M. Sturm, TOPP–the OpenMS proteomics pipeline, *Bioinformatics*, 2007, **23**, e191–e197.

4. J. Junker, C. Bielow, A. Bertsch, M. Sturm, K. Reinert and O. Kohlbacher, TOPPAS: a graphical workflow editor for the analysis of high-throughput proteomics data, *J. Proteome Res.*, 2012, **11**, 3914–3920.
5. L. Martens, M. Chambers, M. Sturm, D. Kessner, F. Levander, J. Shofstahl, W. H. Tang, A. Römpp, S. Neumann and A. D. Pizarro, *et al.*, mzML–a community standard for mass spectrometry data, *Mol. Cell. Proteomics*, 2011, **10**, R110.000133.
6. E. W. Deutsch, Mass spectrometer output file format mzML, *Methods Mol. Biol.*, 2010, **604**, 319–331.
7. D. Kessner, M. Chambers, R. Burke, D. Agus and P. Mallick, ProteoWizard: open source software for rapid proteomics tools development, *Bioinformatics*, 2008, **24**, 2534–2536.
8. M. C. Chambers, B. MacLean, R. Burke, D. Amodei, D. L. Ruderman, S. Neumann, L. Gatto, B. Fischer, B. Pratt and J. Egertson, *et al.* A cross-platform toolkit for mass spectrometry and proteomics, *Nat. Biotechnol.*, 2012, **30**, 918–920.
9. J. Teleman, A. W. Dowsey, F. F. Gonzalez-Galarza, S. Perkins, B. Pratt, H. L. Röst, L. Malmström, J. Malmström, A. R. Jones and E. W. Deutsch, *et al.*, Numerical compression schemes for proteomics mass spectrometry data, *Mol. Cell. Proteomics, Am. Soc. Biochem. Mol. Biol.*, 2014, **13**, 1537–1542.
10. H. L. Röst, U. Schmitt, R. Aebersold and L. Malmström, Fast and Efficient XML Data Access for Next-Generation Mass Spectrometry, *PLoS ONE*, 2015, **10**, e0125108.
11. A. R. Jones, M. Eisenacher, G. Mayer, O. Kohlbacher, J. Siepen, J. N. Selley, B. C. Searle, J. Shofstahl, S. L. Seymour and R. Julian, *et al.* The mzIdentML data standard for mass spectrometry-based proteomics results, *Mol. Cell. Proteomics*, 2012, **11**, M111.014381.
12. S. L. Seymour, T. Farrah, P.-A. Binz, R. J. Chalkley, J. S. Cottrell, B. C. Searle, D. L. Tabb, J. A. Vizcaíno, G. Prieto and J. Uszkoreit, *et al.* A standardized framing for reporting protein identifications in mzIdentML 1.2., *Proteomics*, 2014, 2389–2399.
13. Y. Pérez-Riverol, Q.-W. Xu, R. Wang, J. Uszkoreit, J. Griss, A. Sanchez, F. Reisinger, A. Csordas, T. Ternent and N. Del Toro, *et al.* PRIDE Inspector Toolsuite: Moving Toward a Universal Visualization Tool for Proteomics Data Standard Formats and Quality Assessment of ProteomeXchange Datasets, *Mol. Cell. Proteomics*, 2016, **15**, 305–317.
14. J. A. Vizcaíno, R. Côté, F. Reisinger, H. Barsnes, J. M. Foster, J. Rameseder, H. Hermjakob and L. Martens, The Proteomics Identifications database: 2010 update, *Nucleic Acids Res.*, 2010, **38**, D736–D742.
15. J. A. Vizcaíno, R. G. Côté, A. Csordas, J. A. Dianes, A. Fabregat, J. M. Foster, J. Griss, E. Alpi, M. Birim and J. Contell, *et al.* The PRoteomics IDEntifications (PRIDE) database and associated tools: status in 2013, *Nucleic Acids Res.*, 2013, **41**, D1063–D1069.
16. J. A. Vizcaíno, E. W. Deutsch, R. Wang, A. Csordas, F. Reisinger, D. Ríos, J. A. Dianes, Z. Sun, T. Farrah and N. Bandeira, *et al.* ProteomeXchange

provides globally coordinated proteomics data submission and dissemination, *Nat. Biotechnol.*, 2014, **32**, 223–226.
17. J. Griss, A. R. Jones, T. Sachsenberg, M. Walzer, L. Gatto, J. Hartler, G. G. Thallinger, R. M. Salek, C. Steinbeck and N. Neuhauser, *et al.* The mzTab Data Exchange Format: Communicating Mass-spectrometry-based Proteomics and Metabolomics Experimental Results to a Wider Audience, *Mol. Cell. Proteomics*, 2014, **13**, 2765–2775.
18. J. A. Vizcaíno and O. Griss, *The twenty minute guide to mzTab*, 2014, http://www.psidev.info/mztab.
19. D. N. Perkins, D. J. Pappin, D. M. Creasy and J. S. Cottrell, Probability-based protein identification by searching sequence databases using mass spectrometry data, *Electrophoresis*, 1999, **20**, 3551–3567.
20. S. Kim and P. A. Pevzner, MS-GF+ makes progress towards a universal database search tool for proteomics, *Nat. Commun.*, 2014, **5**, 5277.
21. V. Granholm, S. Kim, J. C. F. Navarro, E. Sjölund, R. D. Smith and L. Käll, Fast and accurate database searches with MS-GF+Percolator, *J. Proteome Res.*, 2014, **13**, 890–897.
22. L. Y. Geer, S. P. Markey, J. A. Kowalak, L. Wagner, M. Xu, D. M. Maynard, X. Yang, W. Shi and S. H. Bryant, Open mass spectrometry search algorithm, *J. Proteome Res.*, 2004, **3**, 958–964.
23. D. L. Tabb, C. G. Fernando and M. C. Chambers, MyriMatch: highly accurate tandem mass spectral peptide identification by multivariate hypergeometric analysis, *J. Proteome Res.*, 2007, **6**, 654–661.
24. S. Tanner, H. Shu, A. Frank, L.-C. Wang, E. Zandi, M. Mumby, P. A. Pevzner and V. Bafna, InsPecT: identification of posttranslationally modified peptides from tandem mass spectra, *Anal. Chem.*, 2005, **77**, 4626–4639.
25. Y. Subbannayya, S. A. Mir, S. Renuse, S. S. Manda, S. M. Pinto, V. N. Puttamallesh, H. S. Solanki, H. C. Manju, N. Syed and R. Sharma, *et al.* Identification of differentially expressed serum proteins in gastric adenocarcinoma, *J. Proteomics*, 2015, **127**, 80–88.
26. P. L. Ross, Y. N. Huang, J. N. Marchese, B. Williamson, K. Parker, S. Hattan, N. Khainovski, S. Pillai, S. Dey and S. Daniels, *et al.* Multiplexed protein quantitation in Saccharomyces cerevisiae using amine-reactive isobaric tagging reagents, *Mol. Cell Proteomics*, 2004, **3**, 1154–1169.
27. D. L. Tabb, The SEQUEST Family Tree, *J. Am. Soc. Mass Spectrom*, 2015, **26**, 1814–1819.
28. O. Serang, M. J. MacCoss and W. S. Noble, Efficient marginalization to compute protein posterior probabilities from shotgun mass spectrometry data, *J. Proteome Res.*, 2010, **9**, 5346–5357.
29. O. Serang, Concerning the accuracy of Fido and parameter choice, *Bioinformatics*, 2013, **29**, 412.
30. L. Käll, J. D. Storey, M. J. MacCoss and W. S. Noble, Posterior error probabilities and false discovery rates: two sides of the same coin, *J. Proteome Res.*, 2008, **7**, 40–44.
31. A. I. Nesvizhskii, A. Keller, E. Kolker and R. Aebersold, A statistical model for identifying proteins by tandem mass spectrometry, *Anal. Chem.*, 2003, **75**, 4646–4658.

32. A. Thompson, J. Schäfer, K. Kuhn, S. Kienle, J. Schwarz, G. Schmidt, T. Neumann, R. Johnstone, A. K. A. Mohammed and C. Hamon, Tandem mass tags: a novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS, *Anal. Chem.*, 2003, **75**, 1895–1904.
33. P. J. Boersema, R. Raijmakers, S. Lemeer, S. Mohammed and A. J. R. Heck, Multiplex peptide stable isotope dimethyl labeling for quantitative proteomics, *Nat. Protoc.*, 2009, **4**, 484–494.
34. S.-E. Ong, B. Blagoev, I. Kratchmarova, D. B. Kristensen, H. Steen, A. Pandey and M. Mann, Stable isotope labeling by amino acids in cell culture, SILAC, as a simple and accurate approach to expression proteomics, *Mol. Cell. Proteomics*, 2002, **1**, 376–386.
35. A. Schmidt, J. Kellermann and F. Lottspeich, A novel strategy for quantitative proteomics using isotope-coded protein labels, *Proteomics*, 2005, **5**, 4–15.
36. F. Lottspeich and J. Kellermann, ICPL labeling strategies for proteome research, *Methods Mol. Biol.*, 2011, **753**, 55–64, Humana Press, Totowa, NJ.
37. H. L. Röst, L. Malmström and R. Aebersold, Reproducible quantitative proteotype data matrices for systems biology, *Mol. Biol. Cell*, 2015, **26**, 3926–3931.
38. A. F. M. Altelaar, C. K. Frese, C. Preisinger, M. L. Hennrich, A. W. Schram, H. T. M. Timmers, A. J. R. Heck and S. Mohammed, Benchmarking stable isotope labeling based quantitative proteomics, *J. Proteomics*, 2013, **88**, 14–26.
39. L. Nilse, F. C. Sigloch, M. L. Biniossek and O. Schilling, Toward improved peptide feature detection in quantitative proteomics using stable isotope labeling, *Proteomics: Clin. Appl.*, 2015, **9**, 706–714.
40. L. Nilse, M. Sturm, D. Trudgian, M. Salek, P. F. Sims, K. M. Carroll and S. J. Hubbard, *SILACAnalyzer–A Tool for Differential Quantitation of Stable Isotope Derived Data. Lecture Notes in Computer Science*, ed. F. Masulli, L. E. Peterson and R. Tagliaferri, Springer, Berlin, Heidelberg, 2010, vol. 6160, pp. 45–55.
41. K. Bartkowiak, M. Kwiatkowski, F. Buck, T. M. Gorges, L. Nilse, V. Assmann, A. Andreas, V. Müller, H. Wikman and S. Riethdorf, *et al.* Disseminated Tumor Cells Persist in the Bone Marrow of Breast Cancer Patients through Sustained Activation of the Unfolded Protein Response, *Cancer Res.*, 2015, **75**, 5367–5377.
42. A. Michalski, E. Damoc, J. P. Hauschild, O. Lange, A. Wieghaus, A. Makarov, N. Nagaraj, J. Cox, M. Mann and S. Horning, Mass Spectrometry-based Proteomics Using Q Exactive, a High-performance Benchtop Quadrupole Orbitrap Mass Spectrometer, *Mol. Cell. Proteomics*, 2011, **10**, M111.011015.
43. S. Nahnsen, A. Bertsch, J. Rahnenführer, A. Nordheim and O. Kohlbacher, Probabilistic consensus scoring improves tandem mass spectrometry peptide identification, *J. Proteome Res.*, 2011, **10**, 3332–3343.
44. D. Shteynberg, A. I. Nesvizhskii, R. L. Moritz and E. W. Deutsch, Combining Results of Multiple Search Engines in Proteomics, *Mol. Cell Proteomics*, 2013, **12**, 2383–2393.

45. D. Shteynberg, E. W. Deutsch, H. Lam, J. K. Eng, Z. Sun, N. Tasman, L. Mendoza, R. L. Moritz, R. Aebersold and A. I. Nesvizhskii, iProphet: multi-level integrative analysis of shotgun proteomic data improves peptide and protein identification rates and error estimates, *Mol. Cell. Proteomics*, 2011, **10**, M111.007690.
46. T. Kwon, H. Choi, C. Vogel, A. I. Nesvizhskii and E. M. Marcotte, MSblender: A probabilistic approach for integrating peptide identifications from multiple database search engines, *J. Proteome Res.*, 2011, **10**, 2949–2958.
47. N. Edwards, X. Wu and C.-W. Tseng, An Unsupervised, Model-Free, Machine-Learning Combiner for Peptide Identifications from Tandem Mass Spectra, *Clin. Proteomics*, 2009, **5**, 23–36.
48. M. Sandin, A. Chawade and F. Levander, Is label-free LC-MS/MS ready for biomarker discovery?, *Proteomics: Clin. Appl.*, 2015, **9**, 289–294.
49. M. Sandin, A. Ali, K. Hansson, O. Månsson, E. Andreasson, S. Resjö and F. Levander, An adaptive alignment algorithm for quality-controlled label-free LC-MS, *Mol. Cell. Proteomics*, 2013, **12**, 1407–1420.
50. J. Cox, MaxLFQ allows accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction, *Mol. Cell. Proteomics*, 2014, 1–37.
51. A. Chawade, E. Alexandersson and F. Levander, Normalyzer: a tool for rapid evaluation of normalization methods for omics data sets, *J. Proteome Res.*, 2014, **13**, 3114–3120.
52. H. Weisser, S. Nahnsen, J. Grossmann, L. Nilse, A. Quandt, H. Brauer, M. Sturm, E. Kenar, O. Kohlbacher and R. Aebersold, *et al.* An automated pipeline for high-throughput label-free quantitative proteomics, *J. Proteome Res.*, 2013, **12**, 1628–1644.
53. Y. V. Karpievitch, A. R. Dabney and R. D. Smith, Normalization and missing value imputation for label-free LC-MS analysis, *BMC Bioinf.*, 2012, **13**(Suppl. 16), S5.
54. B. M. Bolstad, R. A. Irizarry, M. Astrand and T. P. Speed, A comparison of normalization methods for high density oligonucleotide array data based on variance and bias, *Bioinformatics*, 2003, **19**, 185–193.
55. K. Kramer, T. Sachsenberg, B. M. Beckmann, S. Qamar, K.-L. Boon, M. W. Hentze, O. Kohlbacher and H. Urlaub, Photo-cross-linking and high-resolution mass spectrometry for assignment of RNA-binding sites in RNA-binding proteins, *Nat. Methods*, 2014, **11**, 1064–1070.
56. L. C. Gillet, P. Navarro, S. Tate, H. Röst, N. Selevsek, L. Reiter, R. Bonner and R. Aebersold, Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: a new concept for consistent and accurate proteome analysis, *Mol. Cell. Proteomics*, 2012, **11**, O111.016717.
57. H. L. Röst, G. Rosenberger, P. Navarro, L. Gillet, S. M. Miladinović, O. T. Schubert, W. Wolski, B. C. Collins, J. Malmström and L. Malmström, *et al.* penSWATH enables automated, targeted analysis of data-independent acquisition MS data, *Nat. Biotechnol.*, 2014, **32**, 219–223.

58. E. Kenar, H. Franken, S. Forcisi, K. Wörmann, H.-U. Häring, R. Lehmann, P. Schmitt-Kopplin, A. Zell and O. Kohlbacher, Automated label-free quantification of metabolites from liquid chromatography-mass spectrometry data, *Mol. Cell. Proteomics*, 2014, **13**, 348–359, American Society for Biochemistry and Molecular Biology.
59. M. Walzer, L. E. Pernas, S. Nasso, W. Bittremieux, S. Nahnsen, P. Kelchtermans, P. Pichler, H. W. P. van den Toorn, A. Staes and J. Vandenbussche, *et al.* qcML: an exchange format for quality control metrics from mass spectrometry experiments, *Mol. Cell. Proteomics*, 2014, **13**, 1905–1913.
60. L. Gatto and K. S. Lilley, MSnbase-an R/Bioconductor package for isobaric tagged mass spectrometry data visualization, processing and quantitation, *Bioinformatics*, 2012, **28**, 288–289.
61. M. Choi, C.-Y. Chang, T. Clough, D. Broudy, T. Killeen, B. MacLean and O. Vitek, MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments, *Bioinformatics*, 2014, **30**, 2524–2526.
62. G. Rosenberger, C. Ludwig, H. L. Röst, R. Aebersold and L. Malmström, aLFQ: an R-package for estimating absolute protein quantities from label-free LC-MS/MS proteomics data, *Bioinformatics*, 2014, **30**, 2511–2513.
63. B. MacLean, D. M. Tomazela, N. Shulman, M. Chambers, G. L. Finney, B. Frewen, R. Kern, D. L. Tabb, D. C. Liebler and M. J. MacCoss, Skyline: an open source document editor for creating and analyzing targeted proteomics experiments, *Bioinformatics*, 2010, **26**, 966–968.
64. M. R. Bertho;ld, N. Cebron, F. Dill and T. R. Gabriel, KNIME: the Konstanz Information Miner in *Data Analysis, Machine Learning and Applications*, ed. C. Preisach, H. Burkhardt, L. Schmidt-Thieme and R. Decker, Springer, Heidelberg, 2008.
65. S. Aiche, T. Sachsenberg, E. Kenar, M. Walzer, B. Wiswedel, T. Kristl, M. Boyles, A. Duschl, C. G. Huber and M. R. Berthold, *et al.* Workflows for automated downstream data analysis and visualization in large-scale computational mass spectrometry, *Proteomics*, 2015, **15**, 1443–1447.
66. J. Goecks, A. Nekrutenko, J. Taylor and The Galaxy Team, Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences, *Genome Biol.*, 2010, **11**, R86.
67. P. G. A. Pedrioli, J. K. Eng, R. Hubley, M. Vogelzang, E. W. Deutsch, B. Raught, B. Pratt, E. Nilsson, R. H. Angeletti and R. Apweiler, *et al.* A common open representation of mass spectrometry data and its application to proteomics research, *Nat. Biotechnol.*, 2004, **22**, 1459–1466.
68. S. Orchard, P. Jones, C. Taylor, W. Zhu, R. K. Julian, H. Hermjakob and R. Apweiler, Proteomic data exchange and storage: the need for common standards and public repositories, *Methods Mol. Biol.*, 2007, **367**, 261–270.
69. M. Walzer, D. Qi, G. Mayer, J. Uszkoreit, M. Eisenacher, T. Sachsenberg, F. F. Gonzalez-Galarza, J. Fan, C. Bessant and E. W. Deutsch, *et al.* The

mzQuantML data standard for mass spectrometry-based quantitative studies in proteomics, *Mol. Cell. Proteomics*, 2013, **12**, 2332–2340.
70. E. W. Deutsch, M. Chambers, S. Neumann, F. Levander, P.-A. Binz, J. Shofstahl, D. S. Campbell, L. Mendoza, D. Ovelleiro and K. Helsens, *et al.* TraML–a standard format for exchange of selected reaction monitoring transition lists, *Mol. Cell. Proteomics*, 2012, **11**, R111.015040.
71. E. W. Deutsch, L. Mendoza, D. Shteynberg, T. Farrah, H. Lam, N. Tasman, Z. Sun, E. Nilsson, B. Pratt and B. Prazen, *et al.* A guided tour of the Trans-Proteomic Pipeline, *Proteomics*, 2010, **10**, 1150–1159.
72. D. K. Han, J. Eng, H. Zhou and R. Aebersold, Quantitative profiling of differentiation-induced microsomal proteins using isotope-coded affinity tags and mass spectrometry, *Nat. Biotechnol.*, 2001, **19**, 946–951.

CHAPTER 13

Using Galaxy for Proteomics

CANDACE R. GUERRERO[a], PRATIK D. JAGTAP[a,b], JAMES E. JOHNSON[c] AND TIMOTHY J. GRIFFIN*[a,b]

[a]Department of Biochemistry, Molecular Biology and Biophysics, University of Minnesota, 321 Church St SE/6-155 Jackson Hall, Minneapolis, MN 55455, USA; [b]Center for Mass Spectrometry and Proteomics, University of Minnesota, 1479 Gortner Avenue, St. Paul, MN 55108, USA; [c]Minnesota Supercomputing Institute, University of Minnesota, 512 Walter Library, 117 Pleasant Street SE, Minneapolis, MN 55455, USA
*E-mail: tgriffin@umn.edu

13.1 Introduction

Computation has become an essential component to the current era of biomedical research, distinguished by its large-scale and system-wide approaches as exemplified by the 1000 Genomes Project,[1] the growth of personalized molecular medicine, and systems biology. Key to this new era of research is the use of high-throughput technologies enabling genome-, proteome- or metabolome-wide studies. Common to all these technologies is the generation of large datasets requiring informatics solutions, including software for complex analyses and other computational resources for capturing, processing, annotating and disseminating data and experimental procedures.

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

As it has in other '-omic' fields, the reliance on computation and informatics has created a bottleneck in mass spectrometry (MS)-based proteomics.[2] Despite much improvement over the last decade and an increased attention to design, researchers still face many challenges in this area. Although many software applications exist, these often remain difficult for researchers to access and implement, especially for investigators with limited computational expertise seeking to conduct large-scale studies. Additionally, ensuring that complex analyses using multiple software programs are documented and communicated in a manner adhering to scientific standards, and done with process transparency to enable reproducibility by others is a challenge rarely met. Consequently, the richness of results obtained by researchers using high throughput technologies suffers, hindering the translation of information into knowledge and limiting new discoveries of critical importance to biology.

Inherent to MS-based proteomics informatics, specifically those used for "bottom-up" proteomics approaches based on tandem mass spectrometry (MS/MS) and sequence database searching,[3] are stepwise analytical workflows, wherein each step many times requires accessing a single-standing software program. These software programs are targeted at the numerous levels of information that biomedical researchers seek from proteomics studies, such as protein identification, characterization of post-translational modification (PTMs), and quantification of protein abundance. Numerous effective, single standing software programs exist to meet these needs, both commercial and open-source. Unfortunately, although proteomics researchers can intuitively envision how these could be assembled into analytical pipelines of workflows, many do not have the necessary skill sets or training to do so. Thus, researchers are forced to partner with computer experts, or, more often, resigned to operating these on their own, one software at a time in a non-integrated fashion.

In response to these limitations in MS-based proteomic informatics, some groups have developed customized "pipelines" that link together a number of software programs. The Trans Proteomic Pipeline (TPP)[4] and SearchGUI–PeptideShaker[5,6] are some notable examples. However, despite the effectiveness of some of these integrated pipelines they have significant constraints. For one, these pipelines can be limited in the choices of software programs they offer. Thus while facing many requests from the community for new functionality, the responsibility for the framework and analytical application maintenance and development rests with the small group of developers alone. Consequently, framework integration of powerful new software can be lagging. Furthermore, none of these provide an environment that facilitates complete sharing of workflows, including experimental information and exact parameters used for each software program within the workflow. This limits the ability of others to reproduce results employing sophisticated procedures for analysis of large-scale MS-based proteomics data – contributing to the recognized issue of lacking reproducibility in the field.[7]

13.2 The Galaxy Framework as a Solution for MS-Based Proteomic Informatics

With a focus on flexibility, usability, process transparency and reproducibility, Galaxy is a freely-available, open, web-based bioinformatics platform or workbench.[8] Galaxy was initiated as a solution for the genomics research community, which over the years has encountered many of the same informatics challenges as described previously for MS-based proteomics. *Via* the internet and through a consistent and simple interface, users have at their fingertips access to a series of analytical programs and on demand tutorials guiding them through the process of multiple computational analyses and bioinformatics processing tasks (http://main.g2.bx.psu.edu/). Using data provenance information and user activity tracking within the Galaxy space, history logs are recorded in stepwise increments that can be saved for future reference and shared with any or all Galaxy users or exported for publications. Utilizing a sniffer function and XML configuration files for tracking software input file format requirements and compatibility, Galaxy also works as an invisible guide to users on what next steps are possible in an analytical process or pipeline. Galaxy also acts as an enterprise solution, providing centralized coordination of resources such as the creation of reference genome indexes or proteomics reference libraries or other shared data libraries, in addition to data analysis software tools. Lastly, Galaxy is built with scalability in mind, amenable to deployment on high performance computing infrastructure[8] to aid in handling large data volumes.

The Galaxy framework benefits from a wide user community, as well as a dedicated team of developers who work to maintain the core platform and make it usable by the broader biological research community. As the framework has matured over the last decade, so have the accessible resources for training new users in its operation. The core Galaxy team has developed "Galaxy 101" which helps to introduce new users to the framework (found at https://wiki.galaxyproject.org/Learn#Galaxy_101). This site contains basic information on user operations in Galaxy, as well as more targeted tutorials focusing mostly on use of the software for analysis of Next Generation Sequencing data for genomic and transcriptomic characterization. Here we describe some of the basic operations in Galaxy that are important for its use in proteomic data analysis.

13.2.1 The Web-Based User Interface

Galaxy is web-based, such that the user interface operates through any web browser. Figure 13.1 shows a snapshot of the user interface. This shows the basic view as seen by the users, which includes a Tool Menu, the main viewing window and the active history. The Tool menu provides a listing of available software that can be utilized to process data of interest. The structure of this menu can be customized to any given Galaxy instance. The main viewing

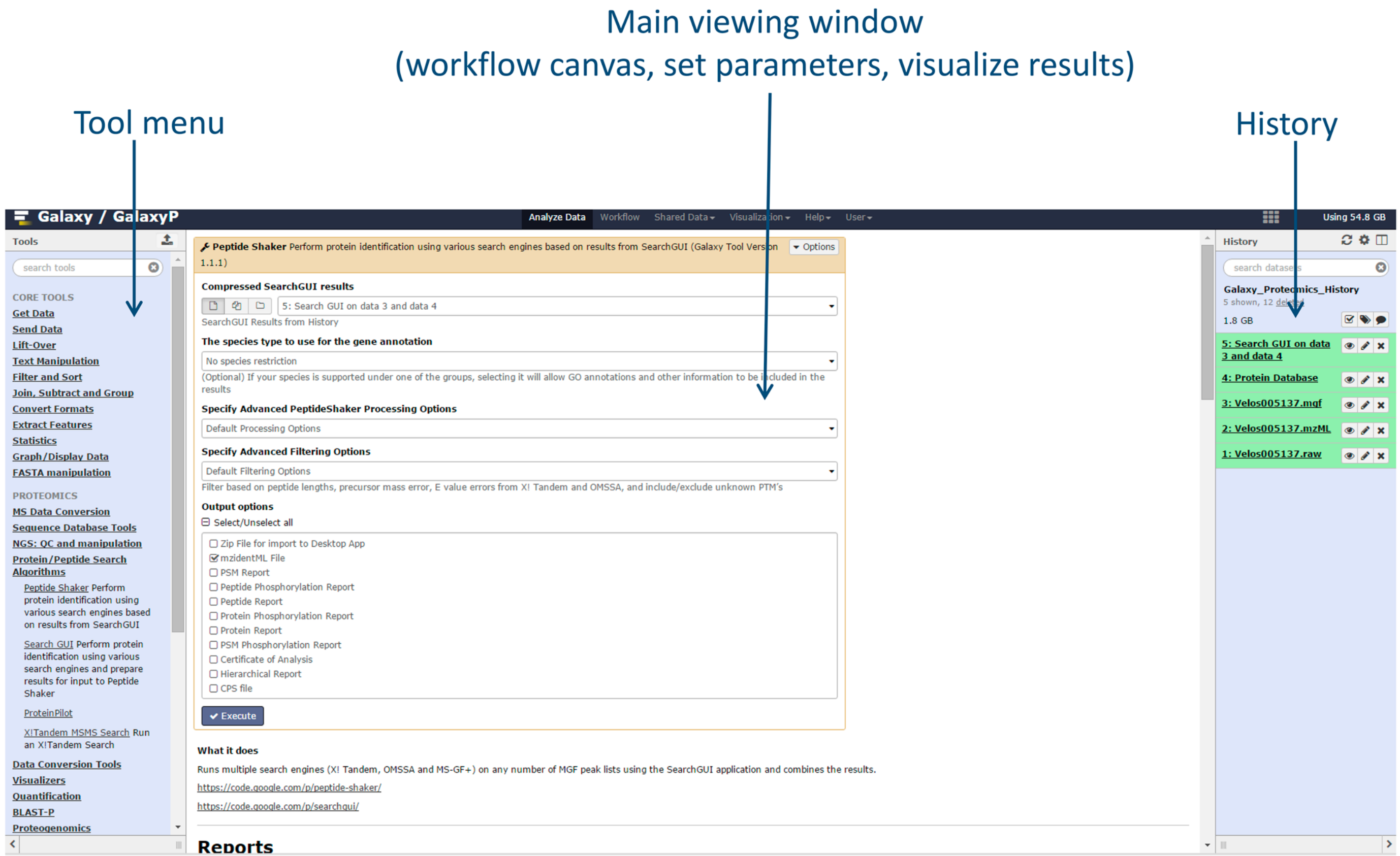

Figure 13.1 The web-based user interface to Galaxy.

window enables a number of different user operations. Parameters are set here when selecting software from the Tool Menu and building a workflow; it also serves as the canvas for editing a workflow and the window for viewing results using Galaxy-based visualization tools. The History section shows the operations and their status when running an analysis. Histories are discussed in more detail in the following section. The History also provides access to results generated at any point along the way when using a workflow containing multiple processing steps.

13.2.2 Galaxy Histories

When a user uploads data, executes software tools for its analysis, and creates processed data outputs, Galaxy creates a record of all these steps. This record is archived and called a History. In addition to all input, intermediate and final outputted datasets, the History also records all parameters and settings used for every software tool across the entire process. As such, a History is a complete archive of any data analysis carried out in Galaxy.

Figure 13.2 shows an example History that represents a relatively simple MS-based proteomics workflow. Here, there are multiple inputs, such as a raw MS data file (step 1), the processed raw data in the form of a peaklist for database searching (step 2 and 3) and a protein sequence database used to match peptide sequences to MS/MS data (step 4). The software tool for analysis of the input data is then selected, in this case the sequence database search software SearchGUI[5] deployed as a Galaxy tool (step 5), which works on the peaklist from step 3 and the database from step 4. After the analysis is complete, in this case matching MS/MS to peptide sequences, an output file is created that becomes part of the History. Galaxy helps guide users through the building of the History, as it will only allow for selection of compatible data types from prior steps in a History when selecting a new software tool to perform a data analysis. Compatible tools for visualization of results also can be selected when clicking on a results file in the History.

Histories provide users with flexibility in viewing outputted data and adding further analysis steps. The History contains not only the input data and final output data, but any intermediate datasets that were generated or utilized in a multi-step data analysis process. The user can access, view and/or download any of the datasets contained in the History. Also, steps can be added to the History and saved either as the same History file, or copied and renamed as a new file. For example, if the user wanted to further analyze the results from the sequence database search shown in the History (Figure 13.2 (step 5)), he or she could select another software tool from the Tool menu and add an additional step for processing the data.

13.2.3 Galaxy Workflows

Workflows are related to Histories, with some important differences. A Workflow consists of analysis and processing steps run in a particular sequence that are used in a History, but it does not contain any specific datasets

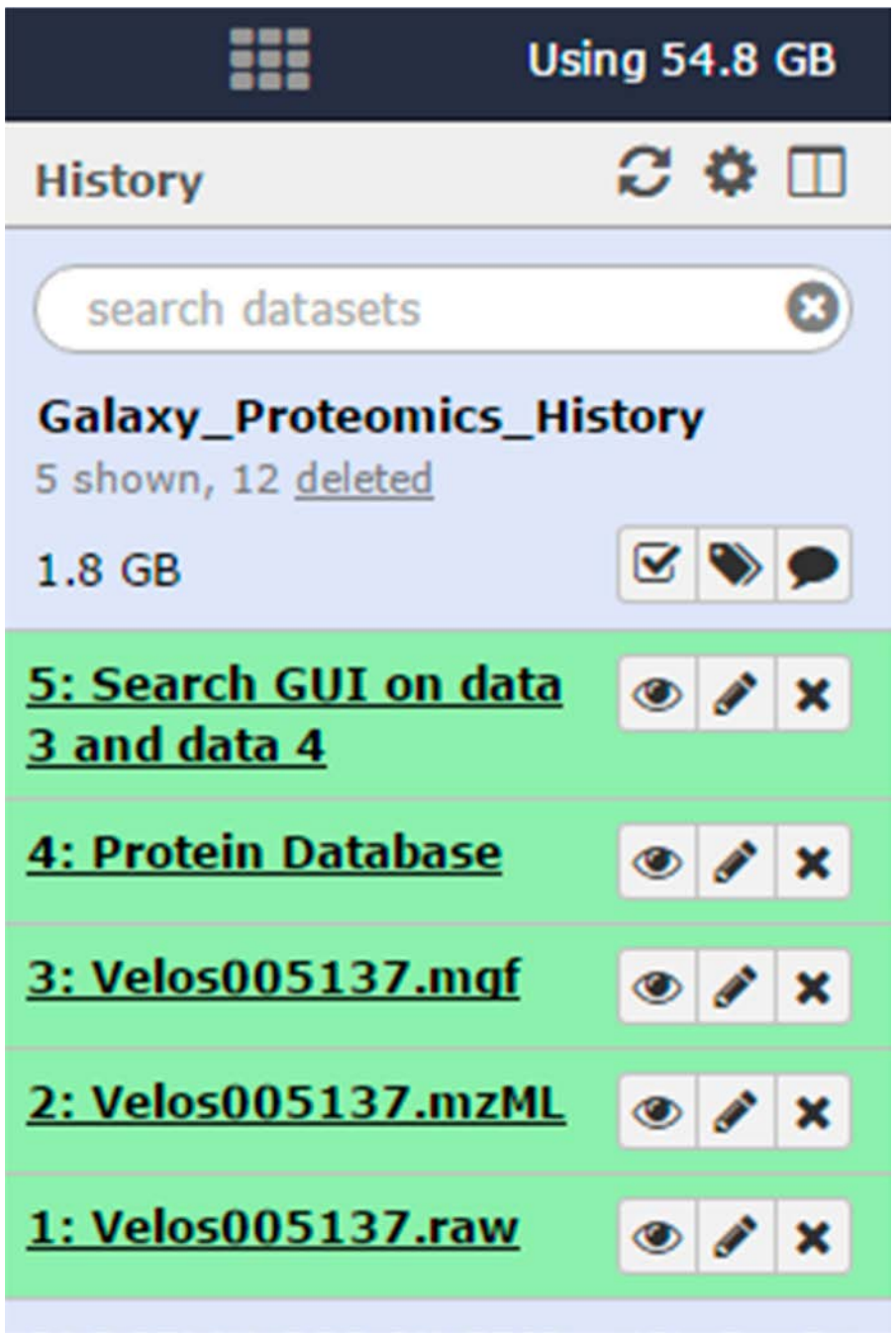

Figure 13.2 An example of a Galaxy History for protein identification from MS/MS data *via* sequence database searching. The History contains the raw data uploaded (step 1), the processed peak list(s) (steps 2 and 3), the protein sequence database used (step 4) and the sequence database search software used with settings (step 5).

(inputs or outputs) like those found in a History. Thus, a Workflow contains all the data analysis steps, including software settings and parameters for each subsequent step, but no data to serve as an input. The Workflow can be published and shared and appropriate data inputted by other users to create a new History.

Workflows can be created and modified in a number of ways. Once a History has been created, a Workflow can be created simply by using the "Extract Workflow" command. Here, the analysis steps along with all relevant parameters and settings, with their ordering preserved, are extracted from the History and saved to a Workflow. Galaxy also contains a Workflow editing function, which can be used to modify an existing Workflow or even build one from scratch. Figure 13.3 shows a snapshot of the Workflow canvas. Here, compatible software tools and processing steps can be linked together *via* a graphical interface to build a multi-step Workflow. As with the creation of Histories, Galaxy helps guide users through the creation of Workflows by only allowing the linkage of software tools that produce data compatible with analysis by the next software tool selected. As such, the intelligent structure of Galaxy helps users avoid generating Workflows that will contain incompatible tools and steps that would need to be corrected later.

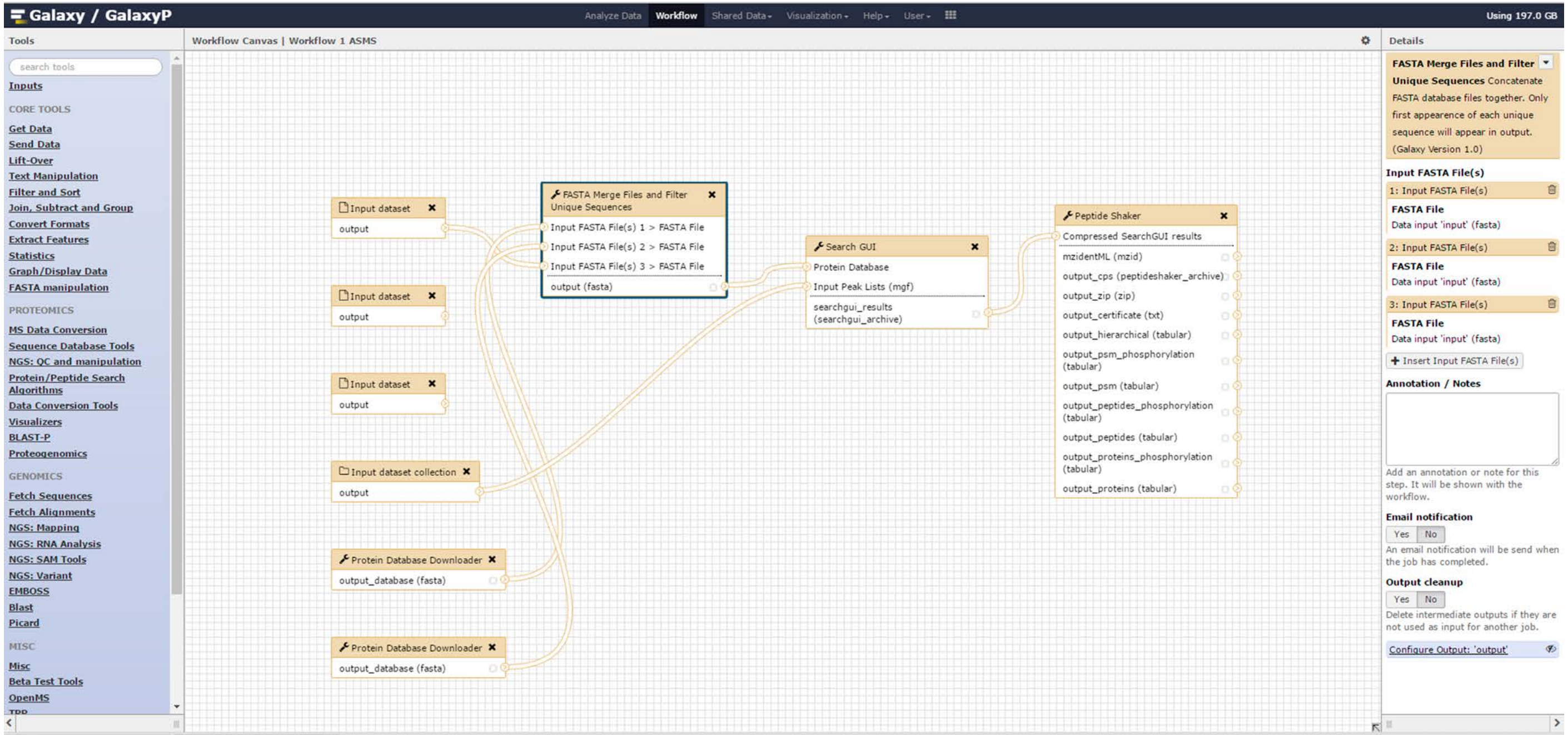

Figure 13.3 The Workflow editing canvas in Galaxy. The graphical interface enables building of multi-step workflows, guiding users through the use of compatible tools for specific input and output data.

13.2.4 Sharing Histories and Workflows in Galaxy

Arguably the most powerful and unique feature of the Galaxy framework is the ability to share not only software tools, but also complete Histories and Workflows between users. This sharing functionality offers several powerful features: dissemination of even complex workflows and easy use by others, as all software settings and parameters critical to optimal operation are shared in the Workflow or History, obviating the need for re-creating these analytical workflows from scratch; reproducibility between different laboratories, an issue that has been difficult to master in the MS-based proteomics community where each laboratory uses its own software and settings; establishment of optimized standards in data analysis that can be shared between laboratories and used on a variety of different data sets.

Sharing is accomplished in a number of ways. Figure 13.4 shows a screenshot of how these are used *via* the user interface. First, Workflows or Histories can simply be made public to all users of the Galaxy instance *via* choosing a simple "Publish History" option. The Workflow or History is then listed with all others that have been published, and are now accessible by other users. Second, rather than publishing as a publically available Workflow or History, the user can choose to create a URL link, which, when clicked, will open the process *via* the locally installed Galaxy instance, making it immediately usable within another user's account. The user can choose to share with only selected other researchers, by sending a link *via* email. Lastly, in the case of Workflows, these can be saved as a Galaxy-formatted file (.ga file type). This can be imported by other users to make use of the analytical process.

13.3 Extending Galaxy for New Data Analysis Applications

13.3.1 Deploying Software as a Galaxy Tool

A core mission in the original conception of Galaxy was to make a platform that was flexible and extensible, amenable to integration of a wide variety of software tools for different analysis applications. Galaxy can be extended to a new area of analysis by adding tools, datatypes, workflows, and interactive plugins such as a visualizations.

A Galaxy tool is a configuration file that the Galaxy server can transform into a web form and that specifies a command line that can be executed by a Galaxy job runner. Thus, any software program that can be run on a command line without requiring interaction by a user can be incorporated into Galaxy as a tool. Galaxy presents a tool to the user *via* the web-based interface as a web form, in which the user can select any inputs and options required by the tool to generate its command line. Clicking the 'Execute' button in the web form submits the tool to Galaxy to be queued for execution as a job. A tool job produces one or more output files that are recorded as datasets in the active History.

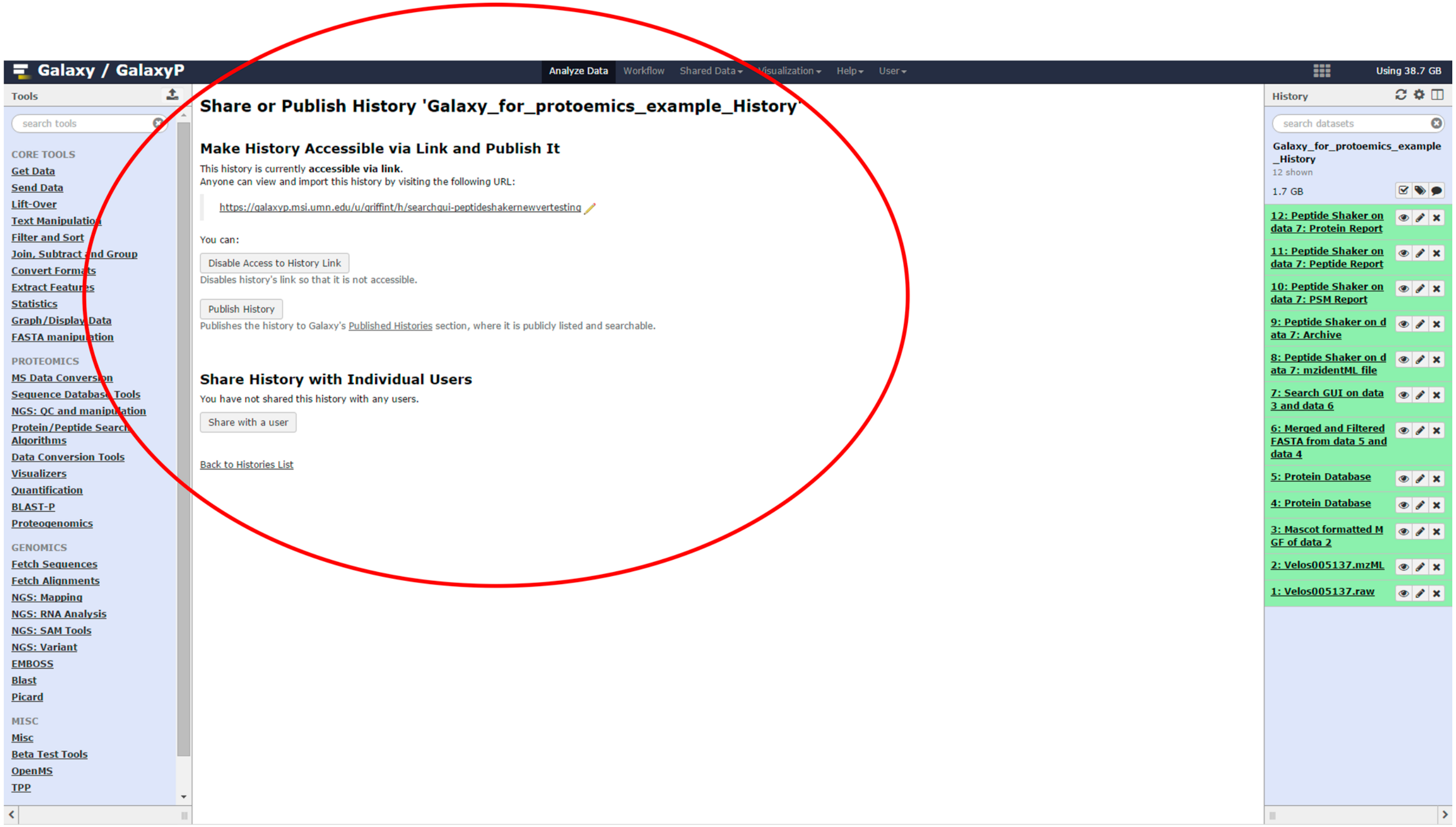

Figure 13.4 A screenshot of the functionality for sharing a History in Galaxy. Here, a URL for the History can be created for sharing with other users, or, optionally, the URL can be emailed to a single, selected user.

Every dataset in Galaxy is assigned a datatype, which informs Galaxy about the format of the file. Galaxy datatypes represent a hierarchical class system for file formats. For example, a gene feature file, GFF or GTF, is a tabular file with a specific number of columns, each with a particular meaning. A tabular file is a text file in which each field in a line is separated by a TAB character. Correctly specifying the input and output datatypes associated with any given Galaxy tool prevents users from providing inappropriate input files to an application. For example, the "Select first lines" tool in Galaxy can operate on any text file with lines including tabular and GTF formats, whereas the "Cut columns" tool requires lines of a file to be separated into fields by the TAB character. Specifying the input to the "Cut columns" tool as a tabular datatype prevents a user from providing it an input dataset that does not contain TAB-separated fields. These are examples of how Galaxy helps guide users through the process of workflow generation, helping them avoid tool combinations that are not compatible with each other.

A new tool is added to Galaxy by providing a tool configuration file. This XML-based file specifies the options to present to the user in a web form, a template for generating the command line, and the output datasets that are produced. The following example is the configuration file for a simple tool that retains the beginning of a file and removes the remaining lines:

```
<tool id = "Show beginning1" name = "Select first" version =
"1.0.0">
  <description>lines from a dataset</description>
  <command interpreter = "perl">headWrapper.pl $input $lineNum
  $out_file1</command>
  <inputs>
        <param name = "lineNum" size = "5" type = "integer"
        value = "10" label = "Select first" help = "lines"/>
        <param format = "txt" name = "input" type = "data"
        label = "from"/>
  </inputs>
  <outputs>
        <data format = "input" name = "out_file1" metadata_
        source = "input"/>
  </outputs>
</tool>
```

The *id* and *version* attributes in the tool tag provide a unique identifier that Galaxy uses for tracking data provenance and providing reproducibility. The *inputs* tag contains options to present to the user in the web form. In this example, the "input" parameter will allow the user to select any textual dataset in the active History (The "format" parameter limits the selection of datasets to those that derive from datatype "txt".). The "lineNum" parameter lets the user enter the number of lines of the input file to copy to the output. The *command* tag contains a template for the command line to execute.

The command template contains variable names, denoted with the '$' prefix, that refer to the inputs and outputs. Galaxy generates the command line by substituting the values of the named parameters for the variable names in the command template. Finally, the *outputs* tag specifies any output files generated by the command line that should be retained as datasets in the Galaxy history.

While the preceding example demonstrates the essential aspects of a tool configuration file, many software programs require more sophisticated configurations. Fortunately, Galaxy provides documentation that can help. The tool configuration specification (https://wiki.galaxyproject.org/Admin/Tools/ToolConfigSyntax) provides a rich set of tags from which to define a tool. Some software applications require a configuration file for settings rather than options that can be expressed on the command line. Just as the command tag contains a template for generating a command line, configfile tags can specify templates for generating temporary files with parameter value substitution. The tool configuration syntax provides the means for conditionally including parameter options. By carefully constructing the tool configuration, the tool developer can limit the selection of options for a software program to eliminate combinations that would cause the program to fail.

As discussed, a Galaxy Workflow is a network of tools in which the output dataset of a tool may be connected as an input dataset for a tool operating in the next step of the Workflow. When a Workflow is executed, Galaxy manages the execution of each individual tool specified in the sequence of analysis steps. A tool in the Workflow is queued as a job as soon as all of its input datasets are available.

13.3.2 Galaxy Plugins and Visualization

While tools and Workflows are designed to run without user intervention, interactive plugins provide the means for users to interactively explore the datasets in a Galaxy History. For example, the scatterplot visualization plugin allows the user to generate and manipulate scatterplots for any tabular file in the history, enabling the user to view the correspondence of any two numerical columns. Numerous visualization plugins tailored toward more viewing of more specific datatypes (*e.g.* assembled DNA or RNA sequences) have also been developed for use in Galaxy.[9]

Developing a visualization plugin for Galaxy requires three elements:

1. A configuration file that specifies which datasets the plugin will operate on.
2. A template for a webpage that will be generated for the visualization. The template can contain variables that reference Galaxy elements, such as the active History and dataset that are to be viewed.
3. A set of Javascript code that the web browser will use to provide the user interaction and provide the connectivity to the Galaxy server for retrieving more data or initiating jobs.

13.4 Publishing Galaxy Extensions

An advantage of using Galaxy for analysis is the large and growing set of available tools, which are available in Galaxy Tool Sheds. Administrators of a Galaxy server can browse or search Galaxy Tool Sheds for tools and even constructed Workflows to extend their server. When the administrator selects a tool to install, the Galaxy server downloads the tool from the Tool Shed and adds it to the tool panel making it available to users. Although many groups maintain their own Tool Shed, the core Galaxy project operates the popular "main" Tool Shed at https://toolshed.g2.bx.psu.edu/. When installing a Galaxy server, its default configuration includes this main Tool Shed as a source of tools.

A Galaxy Tool Shed is a web server that administers a source code control system. To make a tool publicly available, a tool developer creates a tool repository in a Tool Shed and uploads the tool files to that repository. The developer can provide updates to the tool as needed. The Tool Shed assigns a new version to each tool update, while maintaining a copy of the previous versions. This enables Galaxy users to select exact version of tools enabling them to replicate experiments.

A Galaxy Tool Shed can manage software dependencies. Tool Shed repositories may contain a tool_dependencies.xml file that can specify other Tool Shed repositories that need to be installed in order to run a Galaxy tool and can include the installation recipe for a piece of software. When a tool is installed on Galaxy, the server also installs all of the tool dependencies and it records those tool dependencies. It uses those tool dependency records to construct the execution environment with specific software versions when a tool is run.

The Galaxy project provides a command line application, planemo (https://github.com/galaxyproject/planemo), to aid the development of Galaxy tools. The planemo application can initialize a tool configuration form, evaluate the tool as it is being developed, and publish the tool to any Galaxy Tool Shed.

Given the open-source and collaborative nature of the Galaxy project, many tool developers choose to host their tool development on GitHub. This provides a forum for discussing issues with tools and suggesting changes. The major GitHub sites for Galaxy tool development are: https://github.com/galaxyproject and https://github.com/galaxyproteomics/tools-galaxyp.

13.5 Scaling Galaxy for Operation on High Performance Systems

There are many venues for accessing and using a Galaxy server. A number of publicly accessible Galaxy servers are available for general use, including https://usegalaxy.org/ operated by the Galaxy project. However, these public servers usually offer a restricted set of tools and resources. They generally lack the infrastructure needed for users with large volumes of data. Thus, many users turn to operating their own Galaxy server, installed on hardware infrastructure that can support their data volumes. A local server also

provides the ability to customize the platform, adding new tools and capabilities specific to required data analyses.

One of the easier ways to access a personalized Galaxy server is to use a Galaxy project CloudMan image (described at https://wiki.galaxyproject.org/Cloud). The CloudMan image can be operated in the cloud *via* Amazon Web Services (AWS). Using a CloudMan image, the user has administrative privileges to add tools to Galaxy and to scale up the compute resources as needed. Of course, AWS is a commercial service, so fees must be paid that scale with the amount of compute resources used.

Galaxy is also installable on one's own, local hardware resources. The default Galaxy download can be installed and immediately run on a unix-like system, such as Linux or Mac OS X. Many tool developers will run a Galaxy server on their laptop computer for testing during development.

The default Galaxy configuration is sufficient for single user environment, but multi-user, enterprise installations (also called production servers) will need to be enhanced in order to provide the best performance. In these cases, Galaxy is usually run on a distributed, high performance computing infrastructure. The Galaxy project provides a set of incremental enhancements and configuration changes for production servers https://wiki.galaxyproject.org/Admin/Config/Performance/ProductionServer (https://wiki.galaxyproject.org/Admin/Config/Performance/ProductionServer) which includes these steps:

- Use a front end Apache or Nginx proxy webserver to manage user authentication and to serve static data page requests.
- Delegate the relational database to another server such as PostgreSQL. This is accomplished by providing a connection URL in the galaxy.ini configuration file.
- Configure Galaxy to run tool jobs on remote, heterogeneous compute nodes. This is particularly important in a field like proteomics in which some required applications are only available for the Windows operating system. The instructions for configuring Galaxy to use cluster resources are at https://wiki.galaxyproject.org/Admin/Config/Performance/Cluster.

13.6 Windows-Only Applications in a Linux World

Unlike the genomics field, many MS-based proteomics software applications, particularly from equipment vendors, are only available for the Microsoft Windows operating system. A Windows-only application can still be offered as a tool within Galaxy if the application can be executed as a batch script. There are two options for deploying such applications: they can either be run on Linux within a Windows emulation program such as Wine or Mono, or the tool job can be routed to a Windows server to be executed using the Pulsar job runner, a distributed job execution runner made for Galaxy (https://github.com/galaxyproject/pulsar).

Pulsar is a Python web server application that allows a Galaxy server to run jobs on remote systems (including Windows) without requiring a shared, mounted file system. Input files, scripts, and config files are transferred to the remote system, the job is executed, and the results are transferred back to the Galaxy server (ref: https://wiki.galaxyproject.org/Admin/Config/Pulsar). There are some caveats to executing Windows jobs through Galaxy, one being that a tool that is intended to run on a remote Windows server needs to avoid any absolute file paths that would be undefined on that server.

13.7 MS-Based Proteomic Applications in Galaxy

As mentioned previously, Galaxy offers many features beneficial to MS-based proteomics analyses. Large-scale proteome characterization using MS can require multiple single-standing software programs for the numerous steps necessary – from data pre-processing, to sequence database searching to results filtering and visualization. Some experienced labs with computational proteomics expertise have developed platforms for data analysis.[4–6] Despite their usefulness, these platforms do not generally offer the flexibility that many users require, namely workflows that can be designed to fit their analysis needs from start to finish. The flexibility of Galaxy for integration of disparate tools, as well as scalability to handle large data volumes, has many researchers turning toward the platform as a solution for MS-based informatics.

The main Galaxy Tool Shed offers a variety of tools for MS-based proteomics from a number of groups from around the world (See https://toolshed.g2.bx.psu.edu/ under the section "Proteomics".). These tools cover the basic four modules that make up the current paradigm for most MS-based proteomic data analysis workflows. These modules include: (1) raw data conversion and pre-processing; (2) protein sequence database generation; (3) sequence database searching; and (4) results filtering and visualization. In this section, we will discuss examples of Galaxy-based tools within each of these modules, and their role in the overall proteomic data analysis workflow.

13.7.1 Raw Data Conversion and Pre-Processing

Typical MS-based proteomic data outputted from a mass spectrometer are in the form of vendor-specific, Windows-based raw files. Traditionally, accessing the information encoded in these files was dependent upon vendor-produced software, which usually created processed results that were compatible with downstream, vendor-specific software applications. Fortunately, the research community was able to develop tools for converting these vendor-specific results files into a generic format compatible with downstream tools, using programs such as msConvert[10] and MGF Formatter.[11] Since raw data from all mass spectrometers utilize Windows-based dynamic link libraries (DLLs),

their processing requires Windows programs to convert the initial raw data into a generic format for downstream applications.

The msConvert tool operates on Windows systems, processing the raw file information into an mzML file format, the PSI standard spectral file format used by the MS research community[12] (see Chapter 11). This conversion process entails multiple steps such as peak detection and intensity measurement, noise removal, baseline correction, monoisotopic peaks correction (for high resolution data), charge state determination, *etc.* The resulting mzML files are many times referred to as "peaklists", which the subsequent search algorithms will use as input files.

Some sequence database search algorithms directly read mzML as inputs; however a number of other programs use the Mascot Generic Format (MGF) format.[11] MGF is a generic format that encodes multiple experimental MS/MS spectra in a single file with mass-to-charge (*m*/*z*) values and associated intensities separated by headers. The header corresponding to each scan collected by the instrument generating an MS/MS spectrum also has information about the peptide mass, charge state, scan number, *etc.*

The main Galaxy Tool Shed has both msConvert and MGF formatter tools available as base tools for raw data conversion. One caveat with msConvert is that it must run on a Windows operating system to convert raw data, due to the Windows-based DLLs that are associated with the raw files specific to any type of commercially available mass spectrometers. Thus, the Pulsar job runner is needed to set up these Windows jobs from a Galaxy instance.

One characteristic unique to MS-based proteomic data as compared to other '-omic' datatypes is the "one-to-many" relationship of a sample to raw data produced in proteomics. This is a result of the common experimental practice when analyzing complex protein digest mixtures of fractionating the sample prior to LC-MS analysis.[3] This means a single starting sample produces tens of individual raw files (one per fraction), which need to be analyzed as a batch. The results from each raw file (peptide sequences matched to MS/MS spectra) then need to be combined together on the backend to obtain an overall result of identified proteins for the starting sample. This creates potential complications when using Galaxy, which was originally designed for analysis of a single data file generated for a single sample. One could set up multiple histories, one per raw file, and run these separately, although this would be very inefficient and also cause difficulties in combining results. Fortunately the core Galaxy team has provided a solution. Galaxy utilizes a function called "Dataset collections" wherein multiple files of the same type can be defined as a collection, and processed together throughout the proteomic data analysis workflow. This capability simplifies the analysis process as the search engine processes the individual peak list files during the sequence database search and then groups the results together as one output. Typically, a Dataset Collection is defined on processed results (mzML or mgf files) after raw file conversion using msConvert and/or MGF Formatter.

13.7.2 Generation of a Reference Protein Sequence Database

Sequence database searching is a core approach for "bottom-up" proteomics, in which MS/MS spectra are matched with peptide sequences contained in the database.[3] Typically, sequence database searches are performed using databases generated from known protein sequences, such as UniProt's reference proteomes. In Section 13.8.1, we will discuss the generation of protein databases using genomic–transcriptomic data for proteogenomic assessments. To accurately match peptides and subsequently infer identities of proteins within the sample, a database must be as comprehensive as possible to maximize the peptide matches to MS/MS spectra. The database is also selected to specifically match the organism being studied.

Galaxy offers a custom tool, Protein Database Downloader, that utilizes the UniProt Knowledgebase (www.uniprot.org) containing numerous reference proteome sequences for an assortment of organisms. Proteins within UniProt have been manually curated and annotated to minimize redundancies and provide a rich menu of information on each protein (*e.g.* structural, biological function *etc.*) for researchers. A reference proteome of your selection can be chosen within the tool and it directly loads the desired database (in FASTA format) into the current history. Using the Database Downloader tool, sequence databases can be selected from a pre-defined list of organisms, or a URL pointing to the location of the database at the UniProt site can be inserted for organisms not in the list. The tool downloads the most updated version of the database imported into the History from the UniProt site. Earlier versions of databases can be stored locally or accessed remotely if a URL is available.

One of Galaxy's strengths is the manipulation of text files, such as the FASTA formatted sequence databases used for proteomics. Tools exist for merging different FASTA files into a single file, or even creating a decoy database[13] (made up of reversed or scrambled sequences), that allows for estimations of false discovery rate (FDR) of putatively identified peptides and proteins.

13.7.3 Sequence Database Searching

As explained in Chapter 3, in order to infer the proteins present in a complex mixture of enzymatically cleaved peptides, a sequence database searching program is typically employed as a means to match MS/MS spectra to peptide sequences contained within a sequence database. Although there are similarities in many cases, each search program uses a slightly different algorithm to obtain peptide sequence matches (PSMs) for each MS/MS spectrum. Each of these programs then uses a scoring function to assess the confidence of the PSM. Scores are calculated on such parameters as number of ions matched to expected peptide sequence fragments, mass accuracy of the m/z values recorded for intact peptides and fragments detected in MS/MS, as well as a plethora of other parameters. There is not one perfect solution for

sequence database searching, and it is well-established that different search engines usually provide at least slightly different results in proteins identified.[14] Galaxy is able to capitalize on this reality, by deploying many different search programs and providing users a choice to use one or many of these in their workflows. We describe some of these options here, focusing on both open and free software as well as commercial choices.

Galaxy currently offers several open source, non-commercial options for sequence database searching. The standalone tool X!Tandem[15] is one of the most widely used programs, and was the first deployed in Galaxy. More recently, the SearchGUI[5] platform has been deployed in Galaxy. SearchGUI capitalizes on the distinct and complementary algorithms used by different search programs enabling users to analyze their MS/MS data using a suite of different open source software (currently X!Tandem, MyriMatch, MS-Amanda, MS-GF+, OMSSA, COMET, and TIDE). The output from all these search programs can then be tied together and viewed *via* the companion PeptideShaker program[6] described in the next section. Although both SearchGUI and PeptideShaker can be installed as a stand-alone desktop application, their implementation in Galaxy provides scalability for larger datasets as well as linkage to other Galaxy functionalities, such as workflow sharing and integration with other pre- and post-processing tools.

Galaxy is also compatible with commercial software for protein database searching. Although costly, these options can be good solutions as continuous support and maintenance of the software comes with the price. The search program ProteinPilot[16] sold by SCIEX has been used extensively in Galaxy by our group.[17–19] ProteinPilot utilizes a well-developed search algorithm, which provides options for PTM analysis and quantitative proteomics using isobaric tags (iTRAQ) or other isotopic labeling methods such as SILAC. It is a Windows-based program, thus jobs are run through the Pulsar functions in Galaxy. The ability to run these jobs is also dependent on the software being installed on a Windows server with all appropriate commercial licenses to enable operation of the software. Because of this, commercial software is limited to local Galaxy servers for groups who have purchased the licenses, and cannot be used in public servers without permission of vendors.

13.7.4 Results Filtering and Visualization

A final, critical part of the MS-based proteomic workflow is filtering the data to ensure quality and confidence, including the inference of protein identities from PSMs, as well as visualization of results.

Galaxy houses PeptideShaker, which provides a means for organizing and filtering outputted PSM data from SearchGUI, inferring protein identities and providing a means for visualization. With PeptideShaker, a user can stipulate specific parameters for combining the individual database searches performed by SearchGUI and build customized reports (outputted as tabular text files) reporting on PSMs, inferred proteins, FDR levels *etc.* PeptideShaker makes use of the complementary database search programs used

in SearchGUI, assigning increased confidence to the PSMs that were identified by multiple programs. In its current implementation in Galaxy, a zipped file containing all related files for the database search can also be outputted for visualization of spectra, protein sequence coverage, and other features using the stand-alone PeptideShaker viewer program. PeptideShaker also outputs the data in the community standard mzIdentML format, which can be utilized by other Galaxy tools (such as the PSME tool described in the following section).

Somewhat analogous to PeptideShaker, output from the commercial ProteinPilot software, in the form of a .group file, can be viewed in the standalone ProteinPilot viewer program. This is a sophisticated viewing program that provides many assessments of FDR levels, grouping of results by inferred proteins as well as quantitative analysis for different methods, such as iTRAQ isobaric peptide labeling. ProteinPilot also generates a ProteinPilot Descriptive Statistics Template (PDST) that offers quality control metrics about sample preparation, mass spectrometry and identification statistics.

Our group has also focused on developing Galaxy-based tools for visualizing and filtering outputted results. The Peptide Spectrum Match Evaluator (PSME)[18,19] is a Galaxy tool that allows the user to view and evaluate MS/MS spectra matched to peptide sequences and confirm quality of the PSM (Figure 13.5). The PSME tool also allows users to filter PSMs using different quality metrics, which provides a means for extended, stringent filtering of results that goes beyond the scores assigned by the database search engine.

To summarize, Galaxy offers tools that fall into the four main modules central to bottom-up MS-based proteomics data analysis. Galaxy's inherent features allow for modular Workflows to be developed, with each module being a Workflow on its own, or a complete Workflow that combines tools across the modules. As we described at the outset, these Workflows contain a complete record of all parameters used across every tool, making it easier for sharing and reproducing any results derived from their use. This inherent quality of Galaxy becomes even more important for more highly complex workflows, such as those described in the following section, for multi-omic applications.

13.8 Integrating the '-omic' Domains: Multi-Omic Applications in Galaxy

Technological advancements have enabled generations of molecular profiles at the genomic, transcriptomic, proteomic and metabolic levels. Traditionally, the technologies and resulting insights from generated data were restricted to each respective domain. For example, RNA-Seq[20] has made identification and quantification of novel gene transcript rearrangements and variants possible. For proteomics, high-resolution and accurate mass instruments[21] have made it possible to identify and quantify proteins and peptides across nearly the entire dynamic range of abundance. The ability of

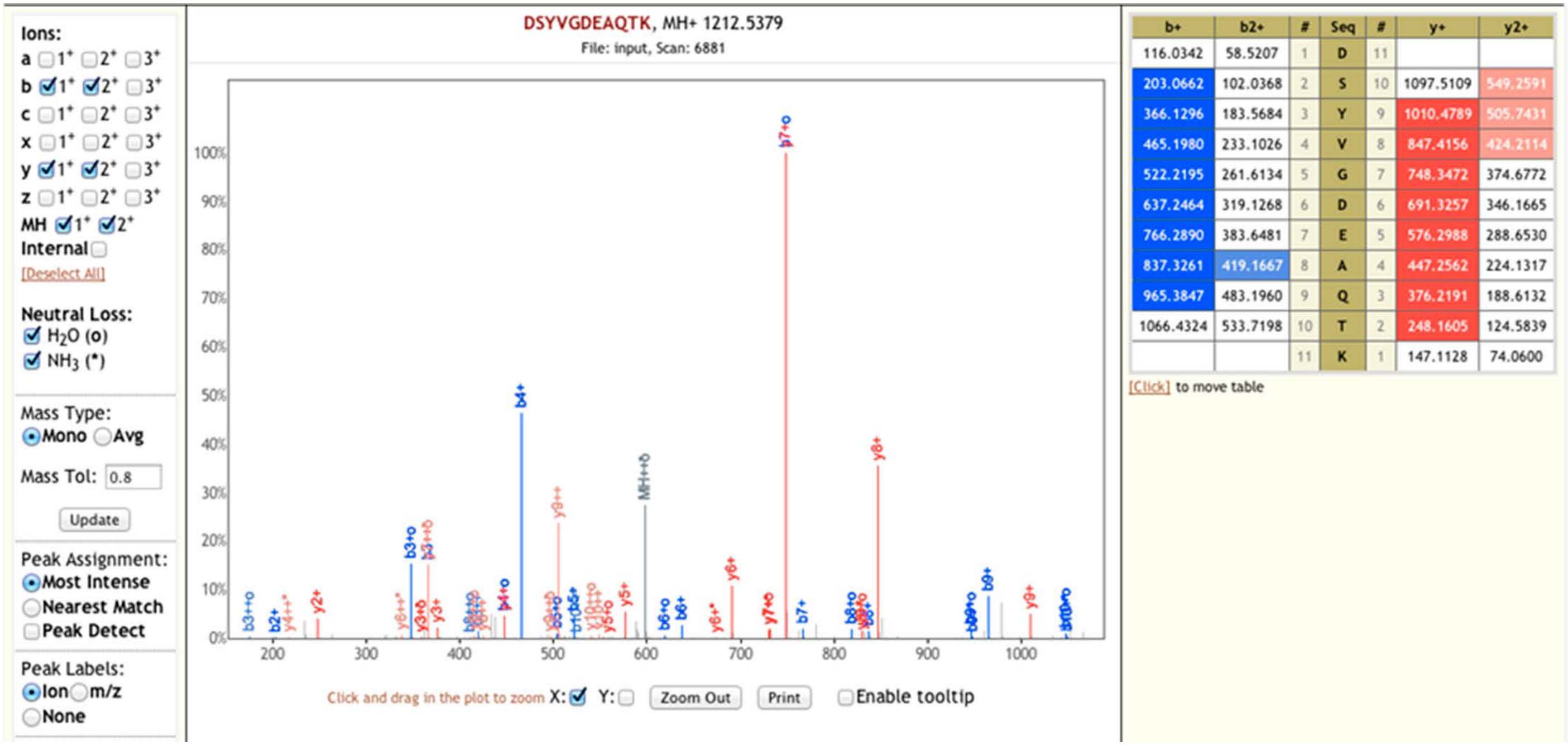

Figure 13.5 The Peptide Sequence Match Evaluator (PSME) tool for visualizing and filtering PSMs outputted from Galaxy proteomic workflows.[18] The tool provides user-input fields to select for different, expected fragment ion types from MS/MS spectra matched to peptides. Ions are annotated based on expected masses of various fragment ion types.

researchers to generate "multi-omic" information has opened up opportunities wherein outputs from one domain can be used to complement and inform findings from another. Meanwhile, methodological advances continue to be explored in each field to increase the depth and accuracy of results – which in turn confers better comparisons between profiles of gene structures, RNA transcripts, proteins or metabolites. For example, quantitative transcriptome profiles can now be more thoroughly compared to quantitative proteome profiles in the same sample, gaining insights into post-transcriptional regulation pathways.[22]

There are multiple challenges presented by multi-omic data analysis. For one, as previously mentioned, most bioinformatic tools have been developed for use within each 'omic domain, targeting users with expertise in analyzing domain-specific data. Moreover each 'omic field has a set of tools that work optimally in a particular environment along with its dependencies. For example, for the conversion of proteomic data acquired by mass spectrometry, data analysis software generally requires Windows-based software along with vendor-specific library dependencies. Conversely RNA-Seq software tools usually work in Linux environments, with their own specific dependencies. The lack of a common interface to access these disparate tools creates a major hindrance to analysis in the field of multi-omic analysis.

The Galaxy framework turns out to be an ideal platform for multi-omic data analysis. As it was initially developed for genomic and transcriptomic analysis, the main Galaxy Tool Shed offers abundant software tools for sequence analysis, variant analysis, genome assembly, FASTA manipulation, metagenomics, transcriptomics analysis *etc.* Thus, it already houses the software needed for genomic and transcriptomic applications, two of the pillars of multi-omic applications. In the last few years, the Tool Shed has expanded into proteomics, as we have already described, and metabolomics (see https://toolshed.g2.bx.psu.edu/ under the section "Metabolomics"). Therefore, the framework stands poised to integrate once disparate data analysis tools between the different 'omic domains. Notably, the intuitive interface and ability to weave different tools into analytical workflows promotes usage by wet-bench researchers. Moreover, these workflows can be creatively modified according to customized needs and can be shared along with the history (as in Figure 13.4).

Despite Galaxy being well poised for multi-omics, development of workflows for these applications still requires careful thought to the design and knowledge of software. In particular, selection and packaging of the appropriate tool from a plethora of choices can require extensive planning and communication between the developers and end users. Next, it is important to define the expected end results for analysis and come up with a blueprint for the analysis. Lastly, if the goal is to develop a robust workflow – testing, refinement and segmenting of the workflow into workable modules is beneficial. Development of modules allows an end user to perform analysis as well as monitor the progress of the workflow by assessing the quality of results during intermediate steps. Once developed, a robust workflow has

many benefits – allowing for sharing and immediate use by others, and use in high-throughput settings where replicate datasets may need analysis.[19]

In the next few sections, we describe the development and use of analytical workflows for two important multi-omic applications – proteogenomics and metaproteomics. Proteogenomics (discussed in more detail in Chapter 15) has emerged as an approach that integrates genomic or transcriptomic data with proteomic data, for improved protein identification and genome annotation. Genomic sequencing reveals gene rearrangements or variants that may not be represented in current annotations of genomes or proteomes. As a result, matching of MS-based proteomics MS/MS data to standard sequence databases misses variant peptide sequences, derived from novel proteoforms,[23] and emanating from gene variations. The ability to translate *in-silico* potential protein sequences derived from genomic sequences,[24–28] cDNA sequences[18,29–32] or RNA-Seq data[19,33–35] has made it possible to generate sample-specific FASTA protein sequence databases. Matching MS/MS data to these databases offers the ability to identify peptides corresponding to novel proteoforms. A caveat to this analysis is that greater scrutiny needs to be applied to these sequences, as they are generally resulting from a single PSM which increases potential for false positives.[36]

Metaproteomics is the study of identification and functional characterization of the complement of proteins expressed as a collection of organisms, usually populations of microbes, within a single sample.[37,38] Metaproteomics benefits from the metagenomic or genomic annotations of organisms under study. In particular, metaproteomic analysis relies on using the protein sequence FASTA database comprised of proteins expressed by the organisms present in the sample, usually determined by metagenomic analysis. Metaproteomics expands information that can be gleaned from metagenomic analysis, in that it offers a snapshot of the proteins expressed by the community, giving direct insight into the biochemical functional state of the system, in addition to taxonomical analysis of the sample.[39]

13.8.1 Building Proteogenomic Workflows in Galaxy

Proteogenomics, as described earlier, integrates genomic or transcriptomic data with MS-based proteomics data, to identify variant sequences belonging to novel proteoforms and better annotate coding regions within genomes. Figure 13.6 provides a blueprint of the modules required for proteogenomics, when starting with assembled RNA-Seq or genomic sequences and MS/MS data.

The database generation in a proteogenomics workflow involves conversion of a nucleic-acid-based sequence database into a protein database. Some of the nucleic acid sequencing data used for these applications is publically available. For example complementary DNA (cDNA) databases (derived from RNA sequences in public databases) are stored at EnsEMBL (the latest version of human cDNA database can be found at: ftp://ftp.ensembl.org/pub/release-82/fasta/homo_sapiens/cdna/). The Galaxy-wrapped tool getORF

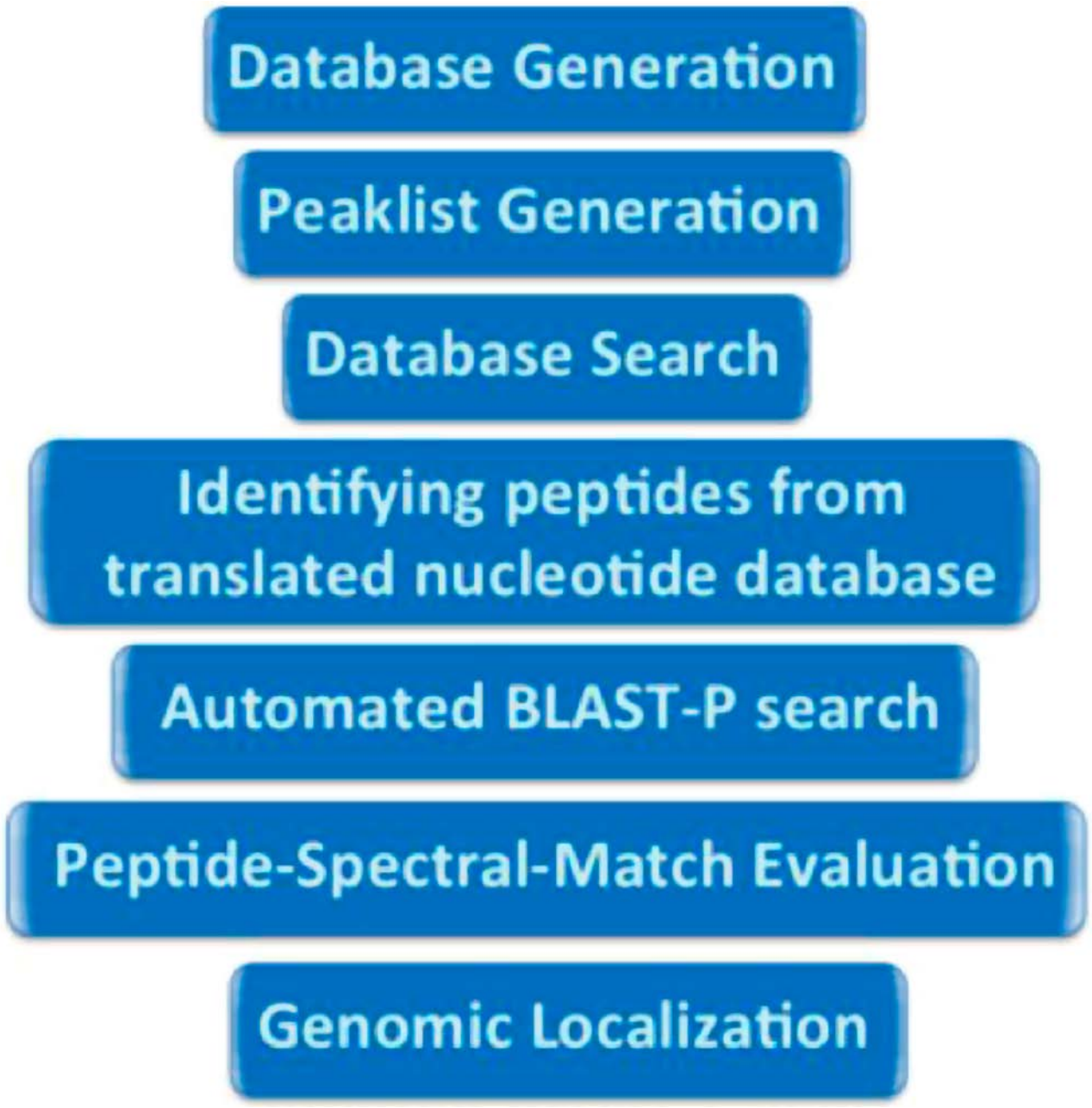

Figure 13.6 Steps involved in a proteogenomic workflow.

from the EMBOSS software suite converts the cDNA database into a protein database.

For RNA-Seq databases, we have developed sophisticated workflows in Galaxy that generate three types of proteomic databases.[40] These workflows use assembly tools such as Tophat and other software implemented in Galaxy to analyze assembled RNA sequences. The RNA sequences are filtered for specific types of transcript variants, and used as a template to create protein sequence databases *via in-silico* translation. The different types of protein sequence databases generated (Figure 13.7) include: (a) peptide sequences with novel single amino acid polymorphisms (SAPs); (b) peptide sequences with novel splice junction sequences; (c) peptide sequences derived from high confidence RNA sequences expressed above a quantitative threshold. These databases can be used for matching to MS/MS data.

Figure 13.8 provides an expanded view of the steps involved in a proteogenomics workflow, once appropriate protein sequence databases have been generated. For processing the raw data, creating peaklists and conducting sequence database searches, the software described in the previous section for proteomics applications are used for proteogenomics as well (*e.g.* msConvert for data conversion and SearchGUI/PeptideShaker for sequence database searching).

Proteogenomics presents some unique requirements and challenges that necessitate utilization of additional tools. Many times the protein sequence

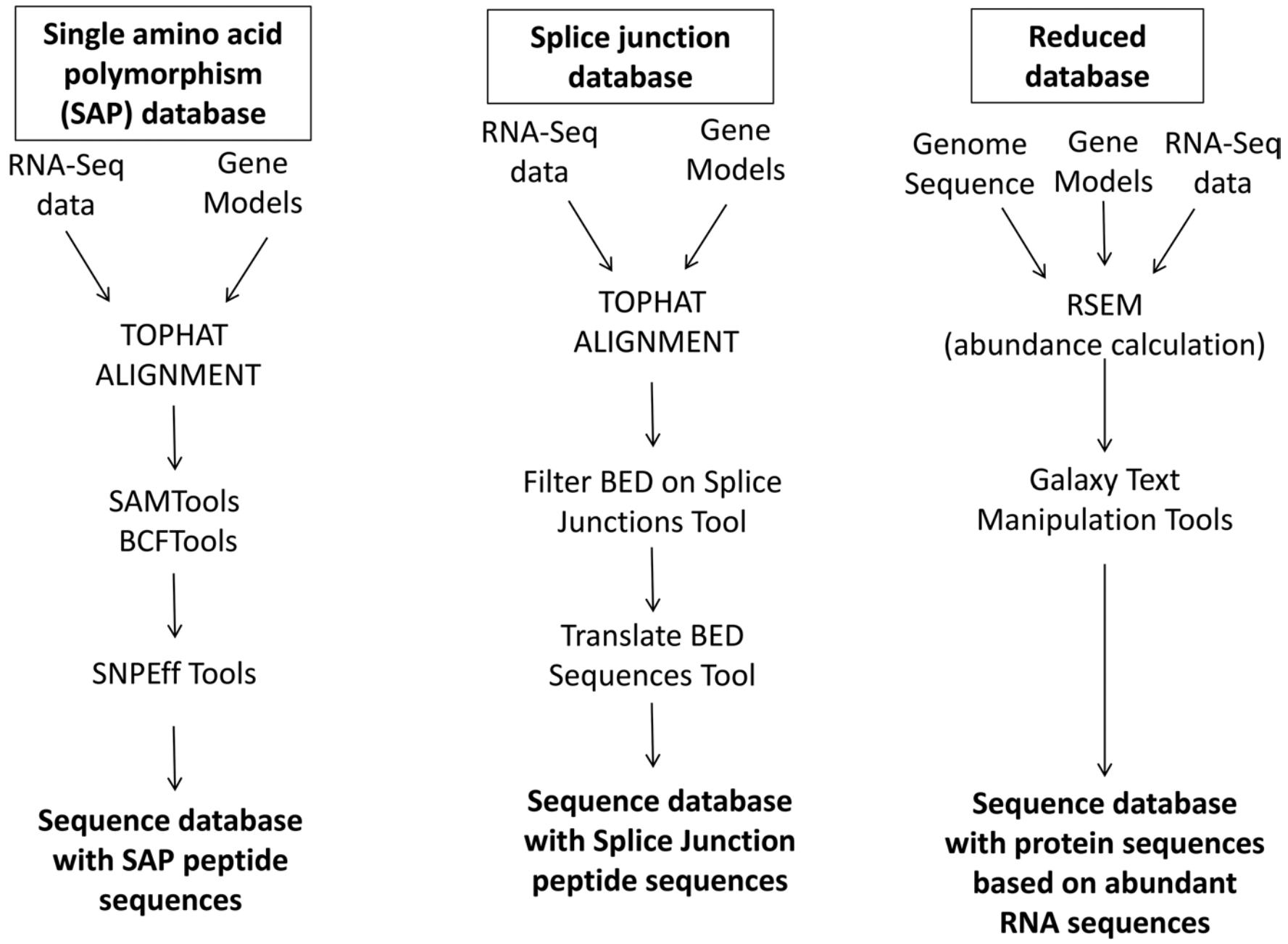

Figure 13.7 Workflows in Galaxy for generating protein sequences databases for identifying novel peptide sequence variants *via* proteogenomics. These are described in detail in ref. 40.

databases used for proteogenomics are significantly larger than those used in normal proteomics studies. For example, the RNA-Seq derived transcript sequences or cDNA sequences may be translated in three coding frames to account for all possible encoded proteins, which quickly create a very large number of protein sequences. Large database searches generally result in a low sensitivity of peptide identifications due to increased potential for false positive identifications.[41,42] Galaxy workflows that use various text processing tools (such as cut, sort, join, *etc.*) can be used to generate smaller, customized databases that maximize PSM identifications.[17,18] In our studies, we have also incorporated the "Minnesota two-step" method which improves sensitivity in peptide spectrum matches.[42] Galaxy's text manipulation tools for merging and creating new FASTA databases are well-suited for automating such methods within workflows. Galaxy should also be well-suited for other emerging methods to improve proteogenomic results, such as the two-stage method[43] or the multi-stage search method.[44]

For database searches, the nucleic acid-derived databases may be appended with the annotated UniProt proteome database. This ensures that the MS/MS data are searched against the standard, known protein sequences from the organism as well as novel sequences derived from the genomic or transcriptomic data. Once the database search is performed, outputted PSMs are further processed using Galaxy tools. Text processing tools parse out PSMs

Published on 15 November 2016 on http://pubs.rsc.org | doi:10.1039/9781782626732-00289

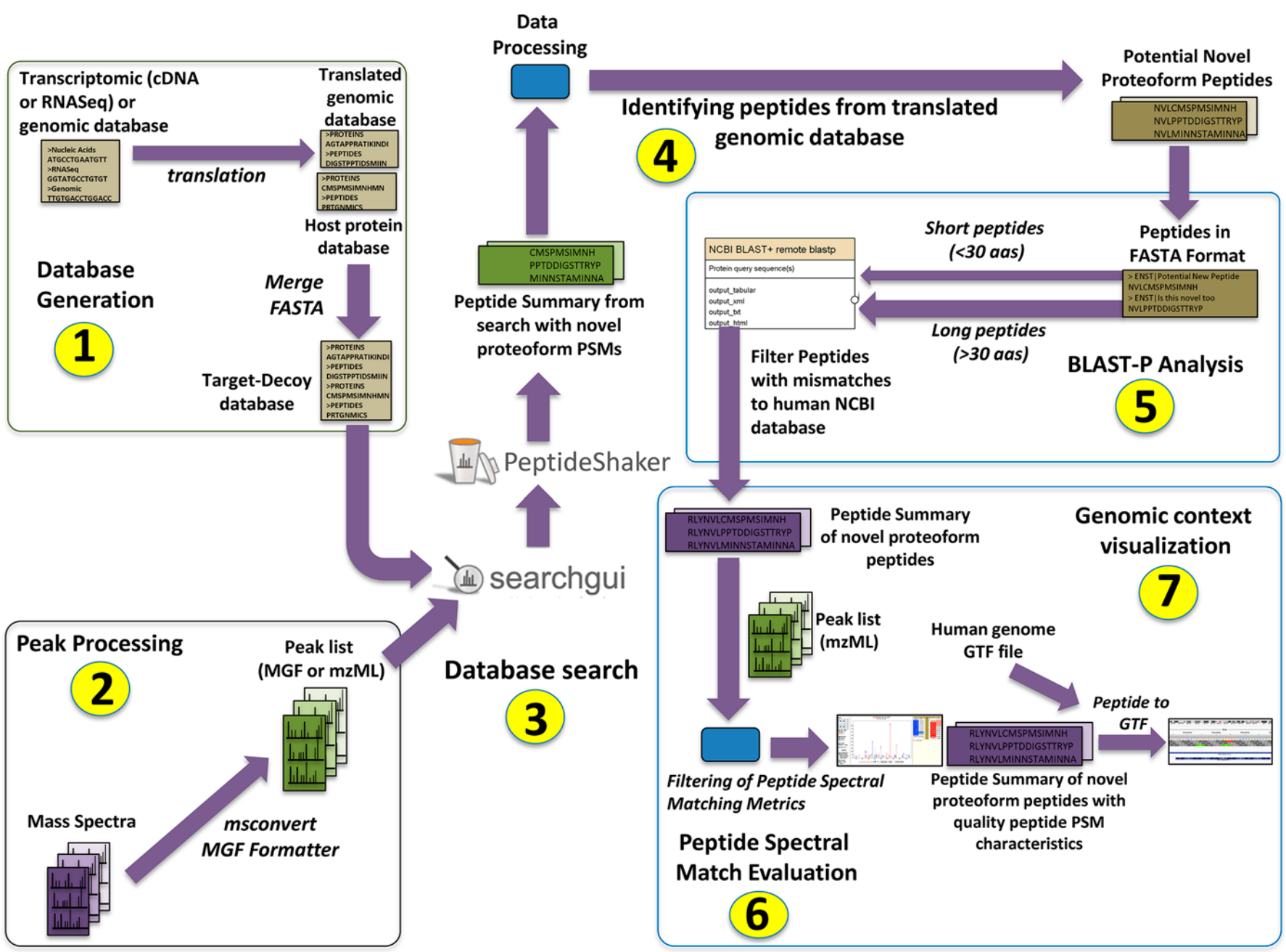

Figure 13.8 A detailed view of a modular proteogenomic Workflow in Galaxy, numbered in order. (1) Genomic or transcriptomic data are translated *in-silico* to generate a protein sequence database; (2) raw mass spectrometry data are converted to peak lists; (3) MS/MS peak lists are matched against the protein database; (4) results are processed and possible novel sequence variants identified; (5) peptide sequences of interest are filtered to verify novelty; (6) quality of PSMs of interest are assessed and visualized; (7) verified novel peptide sequences are mapped to reference genomic sequences and visualized. These workflows are discussed in detail in ref. 18.

corresponding to novel peptide variant sequences carrying scores that are above acceptable FDR thresholds. In order to ascertain that the peptides identified from the translated nucleic databases do indeed correspond to novel sequences, we utilized an elaborate BLAST-P search workflow.[18,19] Briefly, this workflow searches peptides against the latest version of NCBI non-redundant (nr) protein database looking for matches to known sequences. Sequences with gaps or mismatches against the known proteins are selected as putatively novel peptide sequences corresponding to novel proteoforms worthy of further analysis.

In order to further ensure that the quality of peptide spectral matches is of acceptable quality, the PSME tool described is used. *Via* this tool, users can select novel peptide sequences of interest and generate a tabular form of metrics associated with the PSM. These metrics can be used to filter and select for PSMs of highest quality. The PSM tool can also be used to launch the ProtVis application (https://bitbucket.org/Andrew_Brock/proteomics-

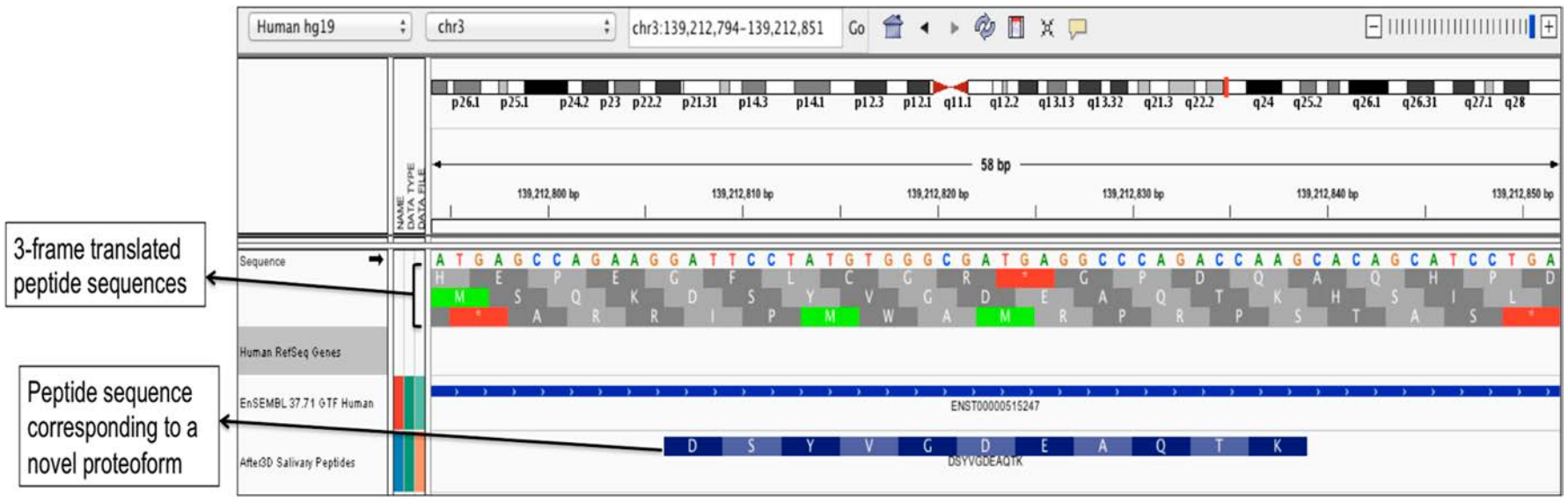

Figure 13.9 A snapshot of the Integrative Genome Viewer (IGV) used for mapping peptide sequences to their coding region in a reference genome.

visualise) which enables visualization of annotated PSMs and their MS/MS spectra for confirmation of the match.

The last step in the proteogenomics workflow is to map the identified peptides onto the coding location in the genome of the organism under study. This enables researchers to understand the nature of potentially novel peptide sequences identified in their study (*e.g.* splice isoforms, SAPs, frameshifts *etc.*). In Galaxy, the peptides to GFF tool generates a GFF file that can be used to upload onto the Integrative Genomics Viewer[45] (IGV, https://www.broadinstitute.org/igv/). The browser can be used to visualize genomic coding localization for any given peptide sequence of interest (see Figure 13.9).

The modules comprising the blueprint workflow for proteogenomics analysis shown in Figure 13.8 have been successfully used for multiple studies by our group.[18,19] However, improvements can always be made, and our group, along with others, is continuing to develop new Galaxy-based tools for these applications. For example, we are working on a Galaxy plugin for improved, interactive viewing of proteomic and multi-omic data. Enhancements such as automated launch of the IGV tool from the Galaxy workflow could also provide a more user-friendly platform. Proteogenomics continues to emerge, with many groups working on software that is either amenable to Galaxy implementation, or already implemented. Some examples here include the Chromosome-Assembled human Proteome browsER,[46] tools for coupling polysome RNA-seq (RiboSeq) data and proteomics[47] and for Proteomics Informed by Transcriptomics described by the Bessant group and implemented in Galaxy.[48]

13.8.2 Metaproteomics Applications in Galaxy

Metaproteomic data analysis has many similarities to proteogenomics, with some important distinctions. With the addition of some tools, the workflow modules generated for proteogenomics can be modified to accommodate metaproteomics analysis (See Figure 13.10 showing a detailed workflow.). Tools for steps such as peaklist conversion and sequence database searching remain the same across all these applications.

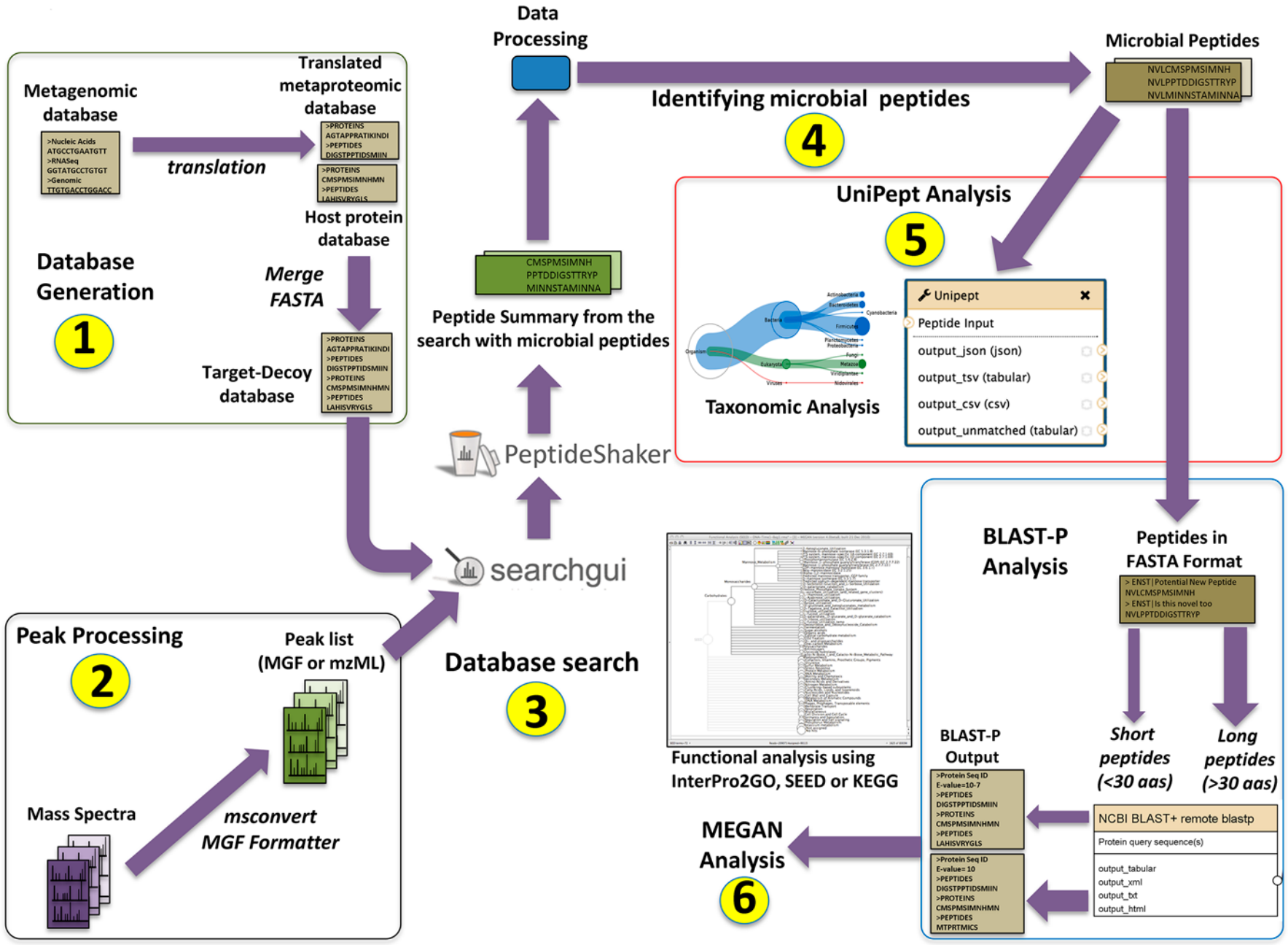

Figure 13.10 A detailed view of a modular metaproteomic Workflow in Galaxy, numbered in order. (1) Metagenomic data are translated *in-silico* to generate a protein sequence database; (2) raw mass spectrometry data are converted to peak lists; (3) MS/MS peak lists are matched against the protein database; (4) results are processed and microbial peptides selected; (5) peptide sequences of interest are assigned to taxonomies and verified; (6) verified peptides are optionally analyzed using tools such as MEGAN, providing taxonomic information as well as functional annotation. These workflows are discussed in detail in ref. 17.

One aspect unique to metaproteomics is the generation and selection of a protein sequence database for matching to MS/MS data. Metaproteomics relies on a protein database, which is generated by merging the protein sequences from many organisms (*e.g.* bacterial species), usually totaling into the hundreds or thousands. The selection of organisms is many times based on metagenomic data which identifies the bacteria present in the sample. In some cases, metagenomic analysis has already been carried out, creating a reference of all species present in the sample being studied. For example, in metaproteomic studies of the oral microbiome by our group, we have used the established Human Oral Microbiome Database (HOMD, http://www.homd.org),[17,42,49,50] translating the genomic sequences from the microbes in this database into expected protein products to create a protein sequence database. In other cases, metagenomics must be performed on the sample of interest to identify organisms present and create the appropriate database.

Regardless of how the database is generated, a distinguishing feature of most metaproteomic sequence databases is their large size – an order of magnitude or more greater than most conventional databases.[51] As with proteogenomics, these large databases present the challenge of maximizing sensitivity for PSMs while minimizing false positives. To meet this challenge, we have also utilized the two-step method within our Galaxy workflows to generate a reduced database that results in increased peptide identifications for metaproteomics.[17,42,49,50]

Some metaproteomics studies, such as those from human samples, require a database that also contains the protein sequences of the host organism, in addition to the community of non-host organisms. In these sample types, the metaproteomic database (*e.g.* the sequences derived from the HOMD data) is appended with the UniProt database of the host proteome (*e.g.* human). Once the sequence database search is performed, important processing steps of the results must be carried out. Text processing tools are used in Galaxy to parse out PSMs that score above thresholds to ensure an acceptable, estimated FDR level. Because identified peptides are not from only one organism, identified peptides must be analyzed *via* taxonomic software. For example, microbial peptides can be submitted to the web-based UniPept[52,53] for phylogenetic analysis. UniPept performs taxonomic assignments of tryptic peptides by using the lowest common ancestor approach. UniPept has now been incorporated into Galaxy thus making it easier for users to submit the peptide inputs and generate outputs as part of a workflow.[17]

While UniPept offers taxonomic analysis of metaproteomics samples, we have found that use of other external bioinformatic tools such as MEGAN5 (http://ab.inf.uni-tuebingen.de/software/megan5/) can also be used for additional functional analysis.[17,49] For this, several text manipulation tools and a powerful BLAST-P workflow in Galaxy is used to generate a text-formatted output compatible with MEGAN analysis. We have submitted as many as 60 000 microbial peptide sequences for processing the downstream MEGAN analysis in a single batch using these workflows.[50]

Our group has successfully used the metaproteomics workflows described, however enhancements can be made to improve these workflows. For example, a suite of tools named Metaproteome Analyzer (MPA),[54] which can perform PSM annotation, functional annotation (at the protein level) and taxonomic assignment has been recently described. Integration of this tool suite into Galaxy would provide even more powerful resources for researchers seeking to pursue studies in this emerging area of metaproteomics.

13.9 Concluding Thoughts and Future Directions

It is clear that Galaxy has a number of advantages to offer the world of MS-based proteomic informatics. Work by our group and others from around the world[47,48,55,56] has demonstrated the potential for Galaxy not only

for proteomic applications, but also for emerging multi-omic data analysis. Coupling the already rich selection of tools for genomic and transcriptomic applications with the growing options for MS-based proteomics makes Galaxy a very attractive option for multi-omic studies integrating analysis of these different datatypes. Metabolomics is also an area of growth in Galaxy,[57] with tools for LC-MS analysis of metabolites growing in the Tool Shed. These tools will add another layer to the possibilities of multi-omic data analysis in Galaxy.

Although proteomic informatics in Galaxy is maturing, there are still a number of enhancements that can be made in this area. One area of importance is tools for visualization and interpretation of outputted data. Although some basic visualization tools are in place, as we describe, there is much room for expansion of their functionality. For example, development of fast-responding, interactive functions within visualization tools could provide users more informative ways to view their data outputs. We are exploring new ways to generate output datatypes that can be read by visualization tools with quick response times, offering users interactive options for "Google maps-like" viewing of results and filtering. We also envision the visualization tools automatically opening other tools for visualization, such as the IGV interface for viewing and mapping peptide sequences onto reference genomes, or querying web-services of appropriate knowledge bases (*e.g.* UniProt) to view known information on proteins of interest. Such tools would put powerful resources for interpreting results from Galaxy workflows at the fingertips of users, allowing them to better generate hypotheses from their data for further testing.

New technologies in MS-based proteomics also continue to emerge that offer new opportunities for Galaxy-based data analysis. Targeted proteomics,[58] using either selected reaction monitoring (SRM) MS approaches or the data-independent acquisition (DIA) approaches require customized software that could benefit from Galaxy deployment. In the case of DIA, the software available is still maturing and seems well-suited for implementation in Galaxy. For example, OpenSWATH[59] and DIAUmpire[60] are platforms that tie together a number of different software tools and modules (see Chapter 10). These different tools are used to make ion libraries from annotated MS/MS spectra matched to peptides in a sample, and then search the DIA data to these libraries in order to extract information on peptides of interest from a complex sample. Other tools are then used for quantifying the amount of signal for peptides of interest in the sample and conducting statistical analysis when comparing results across different samples. The multi-faceted nature of DIA data analysis using disparate software tools makes this application an ideal candidate for deployment in Galaxy, where these tools could be easily weaved together into workflows. Galaxy's scalability is also well-suited for the large datasets produced in DIA.

One last area of continued development is providing easier access to Galaxy instances for the broader research community. Traditionally, to install a local Galaxy server on scalable hardware has taken a fair amount

of technical expertise, limiting the access of some labs to this software. The emergence of new technologies such as Docker (www.docker.com) has provided a solution. Docker provides a "container" in which all software and dependencies are contained in a lightweight package that runs on the existing infrastructure's operating system. A Docker container is easily installed on scalable infrastructure, including cloud-based environments, with advantages over more traditionally used virtual machine images. Using the Docker technology, a customized Galaxy server, containing all tools necessary for proteomics data analysis or other multi-omics applications, can be installed on local infrastructure in a much easier fashion. Alternatively, a Docker container can be easily deployed in cloud infrastructure, such as on Amazon Web Services, allowing for scalable deployment of these tools. These new informatics technologies should provide access for more researchers to the powerful MS-based proteomic data analysis tools being developed in Galaxy – in turn helping to increase the biological discoveries made from the analysis of proteomic and other '-omic' data.

Acknowledgements

The authors wish to acknowledge the National Science Foundation and grants 1147079 and 1458524 that have supported our work developing and using Galaxy for proteomic and multi-omic applications.

References

1. W. Zhang and M. E. Dolan, *Pharmacogenomics*, 2010, **11**, 249–256.
2. J. C. Wright and S. J. Hubbard, *Comb. Chem. High Throughput Screening*, 2009, **12**, 194–202.
3. M. R. Roe and T. J. Griffin, *Proteomics*, 2006, **6**, 4678–4687.
4. E. W. Deutsch, L. Mendoza, D. Shteynberg, J. Slagel, Z. Sun and R. L. Moritz, *Proteomics: Clin. Appl.*, 2015, **9**, 745–754.
5. M. Vaudel, H. Barsnes, F. S. Berven, A. Sickmann and L. Martens, *Proteomics*, 2011, **11**, 996–999.
6. M. Vaudel, J. M. Burkhart, R. P. Zahedi, E. Oveland, F. S. Berven, A. Sickmann, L. Martens and H. Barsnes, *Nat. Biotechnol.*, 2015, **33**, 22–24.
7. D. L. Tabb, L. Vega-Montoto, P. A. Rudnick, A. M. Variyath, A. J. Ham, D. M. Bunk, L. E. Kilpatrick, D. D. Billheimer, R. K. Blackman, H. L. Cardasis, S. A. Carr, K. R. Clauser, J. D. Jaffe, K. A. Kowalski, T. A. Neubert, F. E. Regnier, B. Schilling, T. J. Tegeler, M. Wang, P. Wang, J. R. Whiteaker, L. J. Zimmerman, S. J. Fisher, B. W. Gibson, C. R. Kinsinger, M. Mesri, H. Rodriguez, S. E. Stein, P. Tempst, A. G. Paulovich, D. C. Liebler and C. Spiegelman, *J. Proteome Res.*, 2010, **9**, 761–776.
8. J. Goecks, A. Nekrutenko, J. Taylor and T. Galaxy, *Genome Biol.*, 2010, **11**, R86.

9. J. Goecks, C. Eberhard, T. Too, T. Galaxy, A. Nekrutenko and J. Taylor, *BMC Genomics*, 2013, **14**, 397.
10. D. Kessner, M. Chambers, R. Burke, D. Agus and P. Mallick, *Bioinformatics*, 2008, **24**, 2534–2536.
11. E. W. Deutsch, *Mol. Cell. Proteomics*, 2012, **11**, 1612–1621.
12. L. Martens, M. Chambers, M. Sturm, D. Kessner, F. Levander, J. Shofstahl, W. H. Tang, A. Rompp, S. Neumann, A. D. Pizarro, L. Montecchi-Palazzi, N. Tasman, M. Coleman, F. Reisinger, P. Souda, H. Hermjakob, P. A. Binz and E. W. Deutsch, *Mol. Cell. Proteomics*, 2011, **10**, R110 000133.
13. J. Peng, J. E. Elias, C. C. Thoreen, L. J. Licklider and S. P. Gygi, *J. Proteome Res.*, 2003, **2**, 43–50.
14. J. K. Eng, B. C. Searle, K. R. Clauser and D. L. Tabb, *Mol. Cell. Proteomics*, 2011, **10**, R111 009522.
15. R. Craig and R. C. Beavis, *Bioinformatics*, 2004, **20**, 1466–1467.
16. P. Jagtap, S. Bandhakavi, L. Higgins, T. McGowan, R. Sa, M. D. Stone, J. Chilton, E. A. Arriaga, S. L. Seymour and T. J. Griffin, *Proteomics*, 2012, **12**, 1726–1730.
17. P. D. Jagtap, A. Blakely, K. Murray, S. Stewart, J. Kooren, J. E. Johnson, N. L. Rhodus, J. Rudney and T. J. Griffin, *Proteomics*, 2015, **15**, 3553–3565.
18. P. D. Jagtap, J. E. Johnson, G. Onsongo, F. W. Sadler, K. Murray, Y. Wang, G. M. Shenykman, S. Bandhakavi, L. M. Smith and T. J. Griffin, *J. Proteome Res.*, 2014, **13**, 5898–5908.
19. K. L. Vermillion, P. Jagtap, J. E. Johnson, T. J. Griffin and M. T. Andrews, *J. Proteome Res.*, 2015, **14**, 4792–4804.
20. Y. Han, S. Gao, K. Muegge, W. Zhang and B. Zhou, *Bioinf. Biol. Insights*, 2015, **9**, 29–46.
21. J. Cox and M. Mann, *Annu. Rev. Biochem.*, 2011, **80**, 273–299.
22. S. Haider and R. Pal, *Curr. Genomics*, 2013, **14**, 91–110.
23. L. M. Smith, N. L. Kelleher and P. Consortium for Top Down, *Nat. Methods*, 2013, **10**, 186–187.
24. N. E. Castellana, S. H. Payne, Z. Shen, M. Stanke, V. Bafna and S. P. Briggs, *Proc. Natl. Acad. Sci. U. S. A.*, 2008, **105**, 21034–21038.
25. I. R. Cooke, D. Jones, J. K. Bowen, C. Deng, P. Faou, N. E. Hall, V. Jayachandran, M. Liem, A. P. Taranto, K. M. Plummer and S. Mathivanan, *J. Proteome Res.*, 2014, **13**, 3635–3644.
26. D. Fermin, B. B. Allen, T. W. Blackwell, R. Menon, M. Adamski, Y. Xu, P. Ulintz, G. S. Omenn and D. J. States, *Genome Biol.*, 2006, **7**, R35.
27. H. Pawar, S. Renuse, S. N. Khobragade, S. Chavan, G. Sathe, P. Kumar, K. N. Mahale, K. Gore, A. Kulkarni, T. Dixit, R. Raju, T. S. Prasad, H. C. Harsha, M. S. Patole and A. Pandey, *OMICS*, 2014, **18**, 499–512.
28. J. D. Volkening, D. J. Bailey, C. M. Rose, P. A. Grimsrud, M. Howes-Podoll, M. Venkateshwaran, M. S. Westphall, J. M. Ane, J. J. Coon and M. R. Sussman, *Mol. Cell. Proteomics*, 2012, **11**, 933–944.
29. R. Menon and G. S. Omenn, *Cancer Res.*, 2010, **70**, 3440–3449.
30. R. Menon and G. S. Omenn, *Methods Mol. Biol.*, 2011, **696**, 319–326.

31. R. Menon, Q. Zhang, Y. Zhang, D. Fermin, N. Bardeesy, R. A. DePinho, C. Lu, S. M. Hanash, G. S. Omenn and D. J. States, *Cancer Res.*, 2009, **69**, 300–309.
32. G. S. Omenn, A. K. Yocum and R. Menon, *Dis. Markers*, 2010, **28**, 241–251.
33. A. de Groot, D. Roche, B. Fernandez, M. Ludanyi, S. Cruveiller, D. Pignol, D. Vallenet, J. Armengaud and L. Blanchard, *Genome Biol. Evol.*, 2014, **6**, 932–948.
34. H. D. Li, R. Menon, G. S. Omenn and Y. Guan, *Trends Genet.*, 2014, **30**, 340–347.
35. S. Woo, S. W. Cha, G. Merrihew, Y. He, N. Castellana, C. Guest, M. MacCoss and V. Bafna, *J. Proteome Res.*, 2014, **13**, 21–28.
36. A. I. Nesvizhskii, *Nat. Methods*, 2014, **11**, 1114–1125.
37. P. Wilmes and P. L. Bond, *Environ. Microbiol.*, 2004, **6**, 911–920.
38. P. Wilmes and P. L. Bond, *Trends Microbiol.*, 2006, **14**, 92–97.
39. P. Wilmes, A. Heintz-Buschart and P. L. Bond, *Proteomics*, 2015, **15**, 3409–3417.
40. G. M. Sheynkman, J. E. Johnson, P. D. Jagtap, M. R. Shortreed, G. Onsongo, B. L. Frey, T. J. Griffin and L. M. Smith, *BMC Genomics*, 2014, **15**, 703.
41. B. J. Cargile, J. L. Bundy and J. L. Stephenson, Jr., *J. Proteome Res.*, 2004, **3**, 1082–1085.
42. P. Jagtap, J. Goslinga, J. A. Kooren, T. McGowan, M. S. Wroblewski, S. L. Seymour and T. J. Griffin, *Proteomics*, 2013, **13**, 1352–1357.
43. S. Woo, S. W. Cha, S. Na, C. Guest, T. Liu, R. D. Smith, K. D. Rodland, S. Payne and V. Bafna, *Proteomics*, 2014, **14**, 2719–2730.
44. S. Woo, S. W. Cha, S. Bonissone, S. Na, D. L. Tabb, P. A. Pevzner and V. Bafna, *J. Proteome Res.*, 2015, **14**, 3555–3567.
45. J. T. Robinson, H. Thorvaldsdottir, W. Winckler, M. Guttman, E. S. Lander, G. Getz and J. P. Mesirov, *Nat. Biotechnol.*, 2011, **29**, 24–26.
46. F. Guo, D. Wang, Z. Liu, L. Lu, W. Zhang, H. Sun, H. Zhang, J. Ma, S. Wu, N. Li, Y. Jiang, W. Zhu, J. Qin, P. Xu, D. Li and F. He, *J. Proteome Res.*, 2013, **12**, 179–186.
47. G. Menschaert, W. Van Criekinge, T. Notelaers, A. Koch, J. Crappe, K. Gevaert and P. Van Damme, *Mol. Cell. Proteomics*, 2013, **12**, 1780–1790.
48. J. Fan, S. Saha, G. Barker, K. J. Heesom, F. Ghali, A. R. Jones, D. A. Matthews and C. Bessant, *Mol. Cell. Proteomics*, 2015, **14**, 3087–3093.
49. P. Jagtap, T. McGowan, S. Bandhakavi, Z. J. Tu, S. Seymour, T. J. Griffin and J. D. Rudney, *Proteomics*, 2012, **12**, 992–1001.
50. J. D. Rudney, P. D. Jagtap, C. S. Reilly, R. Chen, T. W. Markowski, L. Higgins, J. E. Johnson and T. J. Griffin, *Microbiomes*, 2015, **3**, 69.
51. T. Muth, C. A. Kolmeder, J. Salojarvi, S. Keskitalo, M. Varjosalo, F. J. Verdam, S. S. Rensen, U. Reichl, W. M. de Vos, E. Rapp and L. Martens, *Proteomics*, 2015, **15**, 3439–3453.
52. B. Mesuere, G. Debyser, M. Aerts, B. Devreese, P. Vandamme and P. Dawyndt, *Proteomics*, 2015, **15**, 1437–1442.

53. B. Mesuere, B. Devreese, G. Debyser, M. Aerts, P. Vandamme and P. Dawyndt, *J. Proteome Res.*, 2012, **11**, 5773–5780.
54. T. Muth, A. Behne, R. Heyer, F. Kohrs, D. Benndorf, M. Hoffmann, M. Lehteva, U. Reichl, L. Martens and E. Rapp, *J. Proteome Res.*, 2015, **14**, 1557–1565.
55. J. Boekel, J. M. Chilton, I. R. Cooke, P. L. Horvatovich, P. D. Jagtap, L. Kall, J. Lehtio, P. Lukasse, P. D. Moerland and T. J. Griffin, *Nat. Biotechnol.*, 2015, **33**, 137–139.
56. C. N. Pang, A. P. Tay, C. Aya, N. A. Twine, L. Harkness, G. Hart-Smith, S. Z. Chia, Z. Chen, N. P. Deshpande, N. O. Kaakoush, H. M. Mitchell, M. Kassem and M. R. Wilkins, *J. Proteome Res.*, 2014, **13**, 84–98.
57. F. Giacomoni, G. Le Corguille, M. Monsoor, M. Landi, P. Pericard, M. Petera, C. Duperier, M. Tremblay-Franco, J. F. Martin, D. Jacob, S. Goulitquer, E. A. Thevenot and C. Caron, *Bioinformatics*, 2015, **31**, 1493–1495.
58. H. A. Ebhardt, A. Root, C. Sander and R. Aebersold, *Proteomics*, 2015, **15**, 3193–3208.
59. H. L. Rost, G. Rosenberger, P. Navarro, L. Gillet, S. M. Miladinovic, O. T. Schubert, W. Wolski, B. C. Collins, J. Malmstrom, L. Malmstrom and R. Aebersold, *Nat. Biotechnol.*, 2014, **32**, 219–223.
60. C. C. Tsou, D. Avtonomov, B. Larsen, M. Tucholska, H. Choi, A. C. Gingras and A. I. Nesvizhskii, *Nat. Methods*, 2015, **12**, 258–264, 257 pp. following 264.

CHAPTER 14

R for Proteomics

LISA M. BRECKELS[a], SEBASTIAN GIBB[b], VLADISLAV PETYUK[c] AND LAURENT GATTO*[a]

[a]Computational Proteomics Unit and Cambridge Centre for Proteomics, Cambridge Systems Biology Centre, University of Cambridge, Tennis Court Road, Cambridge, CB2 1GA, UK; [b]Department of Anesthesiology and Intensive Care, University Medicine Greifswald, Ferdinand-Sauerbruch-Straße, 17475 Greifswald, Germany; [c]Earth and Biological Sciences Directorate, Pacific Northwest National Laboratory, Richland, WA 99352, USA
*E-mail: lg390@cam.ac.uk

14.1 Introduction

R[1] is an open source environment and programming language for statistical computing. These features make it particularly well suited to address data intensive problems such as high-throughput biology. The Bioconductor project[2,3] is focused on the analysis and comprehension of high-throughput omics data; it provides over 1100 R packages and an active user and developer community of wet-lab biologists, computer scientists, computational biologists and statisticians. Mass spectrometry and proteomics are heavily dependent on the exploitation of advanced computing, visualisation and statistical technologies, and the Bioconductor project has, in recent years, benefited from numerous contributions from the mass spectrometry and proteomics community.[4]

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

The philosophy of flexible and robust data analysis is that the analyst controls all the steps of data processing and verifies their relevance to make informed decisions as to whether the final results can be trusted. When these multiple data analysis decisions have been tested and validated,[5] they can then be abstracted into a trusted monolithic pipeline that implements appropriate checks and visualisations to summarise key parameters. R and other similar environments enable opening up of the analysis and gives users control over their data, the crucial parameters of the data analysis and, ultimately, trust in the results. This flexibility also offers an invaluable environment to develop new tools and optimise pipelines for specific use cases by re-using and improving existing functionality.

In this chapter, we present an overview of some use cases and pipelines that are readily available to users and developers. Our aim is that the chapter should be accessible by general proteomics practitioners, although a basic understanding of R is expected to be able to make use of the featured packages (for example https://cran.r-project.org/doc/manuals/r-release/R-intro.html).

Interested readers might also want to consult previous introductory material,[6,7] that presents overviews of data processing and visualisation of mass spectrometry and proteomics data in R. The RforProteomics package, in particular, is an important reference. While we have included numerous code examples in this chapter, illustrating real-life executable application of the R language, RforProteomics provides general introductory material, information on how and where to find help and the complete, detailed and executable code to reproduce the examples described in this chapter as well as reproducible and colour versions of all the illustrations. In the following example, we load the RforProteomics package; its startup message provides links and commands to useful references:

```
library("RforProteomics")
##
## This is the 'RforProteomics' version 1.9.3.
##
##   To get started, visit
##     http://lgatto.github.com/RforProteomics/
##
##   or, in R, open package vignettes by typing
##   RforProteomics() # R/Bioc for proteomics overview
##   RProtVis()       # R/Bioc for proteomics visualisation
##
## For a full list of available documents:
##   vignette(package = 'RforProteomics')
```

Finally, every Bioconductor package provides dynamically compiled overview vignettes. These can be consulted online on the Bioconductor package pages or directly in R by calling the vignette function with the vignette's name.

14.2 Accessing Data

There are currently three different ways to access mass spectrometry and proteomics data directly from R. In the next section, we will describe how to read and handle these data; here, we want to focus on how to obtain such data programmatically, using existing R/Bioconductor infrastructure.

14.2.1 Data Packages

The Bioconductor project offers dedicated experiment–data packages (technically denoted ExperimentData packages) to disseminate specific datasets of interest. These data are typically associated with one or multiple publications or are used to demonstrate a data processing and analysis pipeline. They are typically relatively small or distributed in a processed form. One noteworthy example is the pRolocdata package, which accompanies the pRoloc software package, for the analysis and visualisation of spatial proteomics data using machine learning (see the section on statistics and machine learning for details). This data package distributes tens of published spatial proteomics and protein complexes datasets. These real-life data are used by the package developers to demonstrate their algorithm in the package documentation, and to systematically test existing and new algorithms on a wide range of diverse data. Users can easily obtain these data, including the results of the original publications, and compare these with their own data and analysis results. We demonstrate how to access the quantitative proteomics dataset from Christoforou *et al.*[8] as an MSnSet, a convenient and efficient data structure described in more details in the next section.

```
library("pRolocdata")
data(hyperLOPIT2015)
hyperLOPIT2015
## MSnSet (storageMode: lockedEnvironment)
## assayData: 5032 features, 20 samples
##   element names: exprs
## protocolData: none
## phenoData
##   sampleNames: X126.rep1 X127N.rep1 ... X131.rep2 (20 total)
##   varLabels: Replicate TMT.Reagent ... Fraction.No. (6 total)
##   varMetadata: labelDescription
## featureData
##   featureNames: Q9JHU4 Q9QXS1-3 ... Q9Z2R6 (5032 total)
##   fvarLabels: entry.name protein.description ...
##     cell.surface.proteins (24 total)
##   fvarMetadata: labelDescription
## experimentData: use 'experimentData(object)'
## Annotation:
## - - - Processing information - - -
## Loaded on Thu Nov 5 2015.
```

```
## Normalised to sum of intensities.
##   MSnbase version: 1.19.3
```

14.2.2 Data from the ProteomeXchange Repository

Specific datasets from the ProteomeXchange repository[9] can be queried and downloaded using the rpx package.[10] With a specific identifier (for example, in the code example shown, we use experiment PXD000001), it is possible to query an experiment for, among others, a citation reference, the data's ProteomeXchange URL, and the list of available files, which can be downloaded locally.

```
library("rpx")
px <- PXDataset("PXD000001")
strwrap(pxref(px)) ## reference
```

```
## [1] "Gatto L, Christoforou A. Using R and Bioconductor for
proteomics"
## [2] "data analysis. Biochim Biophys Acta. 2014 Jan;1844(1
Pt A):42-51."
## [3] "Review"
```

```
pxfiles(px) ## available files
```

```
## [1] "F063721.dat"
## [2] "F063721.dat-mztab.txt"
## [3] "PRIDE_Exp_Complete_Ac_22134.xml.gz"
## [4] "PRIDE_Exp_mzData_Ac_22134.xml.gz"
## [5] "PXD000001_mztab.txt"
## [6] "TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_
01-20141210.mzML"
## [7] "TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_01-
20141210.mzXML"
## [8] "TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_01.
mzXML"
## [9] "TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_01.
raw"
## [10] "erwinia_carotovora.fasta"
```

We can choose to download individual files, or all available files.

```
pxget(px,"TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_
01-20141210.mzML")
pxget(px, "all")
```

Programmatically downloading files is important when many files need to be processed. Furthermore, downloading files using scripts, as opposed to manually, is an efficient and reproducible way to assure provenance of the data. After running the downloading function pxget, the files are available on

Published on 15 November 2016 on http://pubs.rsc.org | doi:10.1039/9781782626732-00321

the user's file system, but not in the R environment yet. Fortunately, numerous developers have contributed infrastructure to import these data. The importing process, described in the next section, creates dedicated computational data structures, or objects, enabling the user to manipulate, process and analyse.

14.2.3 Cloud Infrastructure

Finally, some datasets are available through the Bioconductor AnnotationHub cloud infrastructure.[11] The advantages of this system are that it supports caching (*i.e.* that if a resource has already been accessed previously and is available locally, it will not be downloaded again), gives access to many different omics data (in addition to mass spectrometry and proteomics data) and provides users with appropriate data structures in R directly (rather than first downloading and then importing).

In the following example, we load the package, initialise the resource and query for the same experiment as in the rpx previous example. The raw data object (not the raw data file) is then downloaded and made directly available to the user.

```
library("AnnotationHub")
ah <- AnnotationHub()
query(ah, "PXD000001")

## AnnotationHub with 4 records
## # snapshotDate(): 2015-11-19
## # $dataprovider: PRIDE
## # $species: Erwinia carotovora
## # $rdataclass: AAStringSet, MSnSet, mzRident, mzRpwiz
## # additional mcols(): taxonomyid, genome, description, tags,
## #   sourceurl, sourcetype
## # retrieve records with, e.g., 'object[["AH49006"]]'
##
##           title
## AH49006 | PXD000001: Erwinia carotovora and spiked-in
   protein fasta file
## AH49007 | PXD000001: Peptide-level quantitation data
## AH49008 | PXD000001: raw mass spectrometry data
## AH49009 | PXD000001: MS-GF+ identiciation data

rw <- ah[["AH49008"]]
rw

## Mass Spectrometry file handle.
## Filename: 55314
## Number of scans: 7534
```

The drawback of the AnnotationHub infrastructure is that not all data are available, and need to be prepared and added individually. For more

details on the mass spectrometry and proteomics AnnotationHub infrastructure, and how to contribute data, see the ProteomicsAnnotationHubData vignette.[12]

14.3 Reading and Handling Mass Spectrometry and Proteomics Data

As mentioned in the previous section, there exist dedicated data structures–computational objects, enabling the efficient manipulation and processing of mass spectrometry and proteomics data in R. R provides standard structures such as vectors (of characters or numbers), matrices or data frames (spreadsheet-like tables); based on these, developers have defined more complex domain-specific data structures, which model omics data. Table 14.1 summarises some important data structures for mass spectrometry and proteomics, and gives the corresponding data they model and the R/Bioconductor packages they are defined in. Several of the supported file formats are PSI standards (see Chapter 11).

14.3.1 Raw Data

Raw mass spectrometry data, which comes as mzML[13] or mzXML[14] files (netCDF and mzData[15] are also supported) can be interrogated with the openMSfile function from the mzR package.[16] The import function openMSfile produces a data object of class mzRpwiz (using the ProteoWizard[16] back-end) or mzRramp (when using the older Ramp back-end, also part of the ProteoWizard code). The unique feature of this object is that it confers fast on disk access to the raw data, and enables to efficiently access MS1 and MS2 spectra and their corresponding annotation. This feature is relied on by many third-party packages in proteomics and metabolomics.

The following code chunk demonstrates how to import the data after downloading the files using the rpx package, as shown in the previous section.

Table 14.1 Mass spectrometry and proteomics data structure in R/Bioconductor.

Data type	File format	Data structure	Package
Raw	mzXML or mzML	mzRpwiz or mzRramp	mzR
Raw	mzXML or mzML	List of MassSpectrum objects	MALDIquant-Foreign
Raw	mzXML or mzML	MSnExp	MSnbase
Identification	mzIdentML	mzRident	mzR
Identification	mzIdentML	mzID	mzID
Quantitative	mzTab	MSnSet	MSnbase
Peak lists	Mgf	MSnExp	MSnbase
Imaging	imzML or Analyze 7.5	MSImageSet	Cardinal
Imaging	imzML or Analyze 7.5	List of MassSpectrum objects	MALDIquant-Foreign

```
f <- "TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_01-
20141210.mzML"
rw <- openMSfile(f)
rw
```

```
## Mass Spectrometry file handle.
## Filename: TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_
01-20141210.mzML
## Number of scans: 7534
```

The RforProteomics and mzR vignettes further detail how to access the data (with the peaks function) and metadata (with the header function) of raw data object.

14.3.2 Identification Data

Identification data in the mzIdentML format[17] can be parsed using the openIDfile or mzID functions from the mzID[18] and mzR packages. The former, which was the first package supporting the mzIdentML format, parses the XML file using the generic XML R package,[19] while the other uses the ProteoWizard[16] code base for fast on-disk access, which is particularly useful when many files need to be accessed and analysed. Each alternative offers annotation and identification result accessors, briefly illustrated as follows. More details are available in the respective package and in the RforProteomics vignettes.

```
idfile <-"http://psi-pi.googlecode.com/svn/trunk/examples/1_
1examples/55merge_tandem.mzid"
library("mzID")
id1 <- mzID(idfile)
```

```
## reading 55merge_tandem.mzid... DONE!
```

```
id1
## An mzID object
##
## Software used: X\!Tandem (version: x! tandem CYCLONE
(2010.06.01.5))
##
## Rawfile: D:/TestSpace/NeoTestMarch2011/55merge.mgf
##
## Database: D:/Software/Databases/Neospora_3rndTryp/Neo_
rndTryp_3times.fasta.pro
##
## Number of scans: 169
## Number of PSM's: 170
```

```
parameters(id1)
```

```
## $searchType
## [1] "ms-ms search"
```

```
##
## $threshold
##           name
## 1 no threshold
##
## $`parent mass type mono`
## [1] TRUE
##
## $`fragment mass type mono`
## [1] TRUE
##
## $enzymes
##         name
## 1 Trypsin
##
## $ParentTolerance
##   accession   cvRef unitCvRef unitName unitAccession value
## 1 MS:1001412 PSI-MS     UO     dalton    UO:0000221     1.5
## 2 MS:1001413 PSI-MS     UO     dalton    UO:0000221     1.5
##                       name
## 1 search tolerance plus value
## 2 search tolerance minus value
##
## $FragmentTolerance
##   accession   cvRef unitCvRef unitName unitAccession value
## 1 MS:1001412 PSI-MS     UO     dalton    UO:0000221     0.8
## 2 MS:1001413 PSI-MS     UO     dalton    UO:0000221     0.8
##                       name
## 1 search tolerance plus value
## 2 search tolerance minus value
##
## $ModificationRules
##  residues massDelta fixedMod name          Specificity
## 1       M 15.99492    FALSE   Oxidation          any
## 2       C 57.02147    TRUE   Carbamidomethyl    any
```

When using mzR, we first need to download the identification file and then open and query it.

```
download.file(idfile, basename(idfile))
id2 <- openIDfile(basename(idfile))
id2
```

```
## Identification file handle.
## Filename: 55merge_tandem.mzid
## Number of psms: 171
```

```
softwareInfo(id2)
## [1] "xtandem x! tandem CYCLONE (2010.06.01.5) "
```

```
## [2] "ProteoWizard MzIdentML 3.0.6239 ProteoWizard"
```

```
enzymes(id2)
```

```
## name nTermGain cTermGain minDistance missedCleavages
## 1 Trypsin H OH 0 1
```


14.3.3 Quantitative Data

The MSnSet class, defined in the MSnbase[20] package, is used to store quantitative data as well as feature and sample annotation in one coherent object. Figure 14.1 illustrates a simplified view of the MSnSet structure. The quantitative expression data are stored as a matrix of size *n* features along the rows (spectra, peptides of proteins) times *m* samples along the columns, and features and samples are annotated in their respective metadata tables. The features in the expression matrix and the feature metadata match exactly, and every row-wise re-ordering or sub-setting is automatically applied on the expression and metadata tables. The feature metadata can however be expanded along its columns (by addition of a new feature annotation, such as, for example, a *p*-value reflecting the difference of expression of the associated expression data, or the result of a classification analysis – see Section 14.8 *Machine Learning, Statistics and Applications*) without requiring any modification of the expression data. Conversely, sample metadata is stored in its own table, whose dimensions must match the number of columns in the expression data. MSnSet objects are used in a wide range of packages and use cases, ranging from spectral counting (see Section 14.5 *Analysis of Spectral Counting Data*), isobaric tagging (see Section 14.7 *Isobaric tagging and quantitative data processing*), general statistical analysis (for example MSstats[21]) and machine learning (see Section 14.8 *Machine Learning, Statistics and Applications*).

Such MSnSet objects can be created from mzTab files,[22] or any text-based spreadsheet files (such as comma- or tab-separated files) stemming from

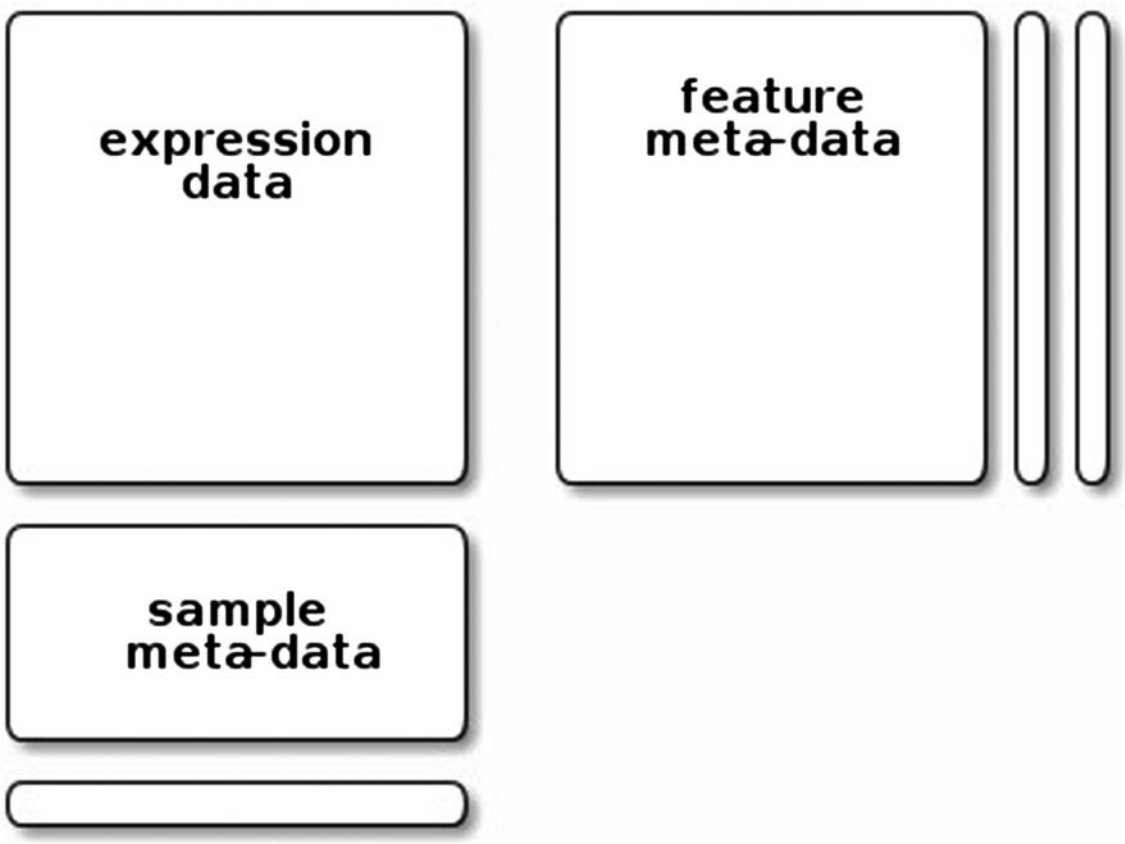

Figure 14.1 Simplified representation of the MSnSet data structure (reproduced with permission from the MSnbase vignette).

popular third-party applications such as, for example, MaxQuant[23] or Thermo Scientific's Proteome Discoverer. mzTab files can be read in with the readMzTabData function, while data in arbitrary spreadsheets can be imported with readMSnSet, or the more simple readMSnSet2.

The MSnbase package offers a class for raw data called MSnExp, that can be created from mzML or mzXML files (relying on mzR) or mgf peak lists. As opposed to the data structure provided by mzR, MSnExp are more flexible, but are created in-memory, making them only usable for small data or MS2 data only (see Section 14.7 *Isobaric tagging and quantitative data processing*).

14.3.4 Imaging Data

Cardinal and MALDIquantForeign support the import of imaging data in imzML or the Analyse 7.5 format. Similar to the methods provided by MSnbase, Cardinal has a readMSIData function that reads these files into an MSImageSet object. In addition to the imaging data the MALDIquantForeign package can import a lot of free and vendor-specific data formats, *e.g.* text-based spreadsheet files (such as comma- or tab-separated files), mzML, mzXML or Bruker Daltonics *flex format into MALDIquant specific MassSpectrum or MassPeaks objects. Mass spectrometry imaging is introduced and discussed in Section 14.6.3.

14.3.5 Conclusion

Every package that offers a specific data processing and visualisation functionality will rely on dedicated data structures, either by implementing their own or by depending on those provided by other packages. A strength of the open and collaborative development of the Bioconductor project is that different packages share some of these data structures to build more sophisticated data analysis pipelines, as will be demonstrated in following sections.

14.4 MSMS Identifications

14.4.1 Introduction

Identification of MS/MS spectra is traditionally done with highly specialised software tools like SEQUEST,[24] X!Tandem,[25] Mascot[26] and MS-GF+[27] to name a few (see Chapter 4). Typically it is a very computationally intensive process that is better handled by tools implemented in compiled languages such as C++ or Java. However, at the same time, the MS/MS identification is just a step in a more elaborate data analysis pipeline. It is tied with numerous downstream tasks such as tuning up filtering parameters, visualisation of the results at the level of spectra or protein sequences, proteogenomic inferences and quantitative analysis. R provides a versatile scripting environment

for stitching multiple steps in the pipeline all the way down to final inferences using a number of custom proteomics-related packages, Bioconductor's biological data analysis packages and general statistical data analysis packages. In this section we will specifically review proteomics-related packages covering MS/MS data searching (MSGFplus, MSGFgui and rTANDEM packages) and follow-up handling of identifications (MSnID package).

14.4.2 The MSGFplus Package

The automatic identification of peptides from LC-MS/MS experiments has become a widely used technique since the introduction of the SEQUEST algorithm in 1994,[24] but the process has constantly been refined and improved. Currently there exists a range of different algorithms for performing the identification task, all with strengths and weaknesses, and MS-GF+[27] is one of the latest, but an increasingly popular alternative. The MSGFplus package[28] makes it possible to set up MS/MS searches in R by defining and passing the parameters directly to the MS-GF+ executable followed by parsing of the results. Setting up parameters can be done using the msgfPar function and details about the parameters can be found on the MS-GF+ website.†

```
library("MSGFplus")
parameters <- msgfPar(
  database = system.file(package = 'MSGFplus', 'extdata',
'milk-proteins.fasta'),
  tolerance = '20 ppm',
  isotopeError = c(0, 2),
  enzyme = 'Trypsin',
  ntt = 0)
show(parameters)
```

```
## An msgfPar object
##
## Database: /home/lg390/R/x86_64-pc-linux-gnu-library/3.3/
MSGFplus/extdata/milk-proteins.fasta
## Tolerance:             20 ppm
## Isotope error range:   0-2
## Enzyme:                1: Trypsin
## No. tolerable termini: 0
```

All parameters, including expected post-translational modifications, can be accessed and modified using relevant setter and getter methods, as detailed in the MSGFplus and RforProteomics vignettes.

† http://proteomics.ucsd.edu/software-tools/ms-gf/

Another method is to read parameter data from a result file generated by MS-GF+. This makes it easy to quickly replicate the parameter used for a certain search in order to compare results.

```
parameters <- msgfParFromID('/path/to/results/file.mzid')
```

Parameters can also be set through a GUI interface using the gWidgets[29] package.

```
require(gWidgets)
parameters <- msgfParGUI()
```

Finally once the parameters are set appropriately, the MS-GF+ can be called by the runMSGF method. If multiple files are provided, these will be run in succession. By default result files are written besides the original raw files with an *.mzid extension instead of their original extension (silently overriding existing files). Alternatively a list of filenames, of the same length as the number of input raw files, can be provided to use as output file names.

```
idres <- runMSGF(par, 'your_rawfile.mzML')
```

The results are automatically re-imported into R as either an mzID or mzIDCollection object (see Section 14.3 *Reading and handling mass spectrometry and proteomics data*), depending on the number of raw files.

14.4.3 The MSGFgui Package

The MSGFgui package[30] is a graphical user interface for its sister package MSGFplus. It provides a GUI overlay, shown on Figure 14.2, for setting up MS/MS search parameters and a set of visualisations coded in JavaScript using D3.js.

```
library(MSGFgui)
MSGFgui()
```

The searches can be set-up through the interface or loaded directly in the form of mzIdentML files. The MS/MS identification results can be easily explored using visual tools along the sample → protein → peptide → MS/MS spectrum axis (Figure 14.3). After filtering the data using custom criteria, false discovery rate (FDR) calculations rely on the q-values calculated by MS-GF+. This means, however, that it is not updated after applying additional filters available through the GUI. Options are available to trim down the scans, either by only looking at specific samples, retention times, m/z values or charges. It is possible to choose only to look at peptides related to a subset of proteins or of a certain length, as well as having specific modifications. Once a filter is set, it is applied when another tab is selected.

It should be noted that the filtering is provided with the purpose of making it easier to find the information of interest. For instance if one is mostly

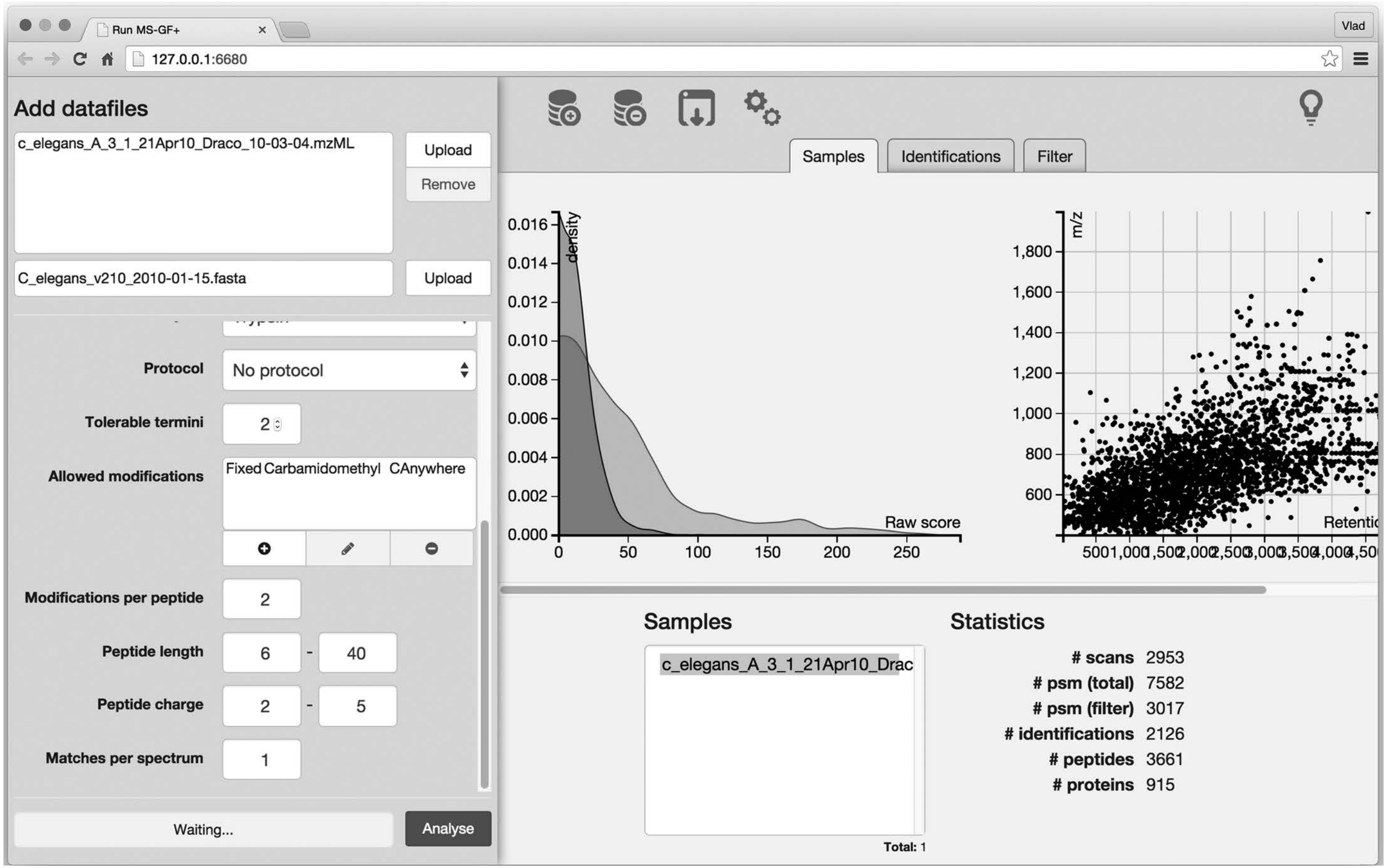

Figure 14.2 Main panel of MSGFgui showing the files and analysis parameters (left) and an overview of the search results.

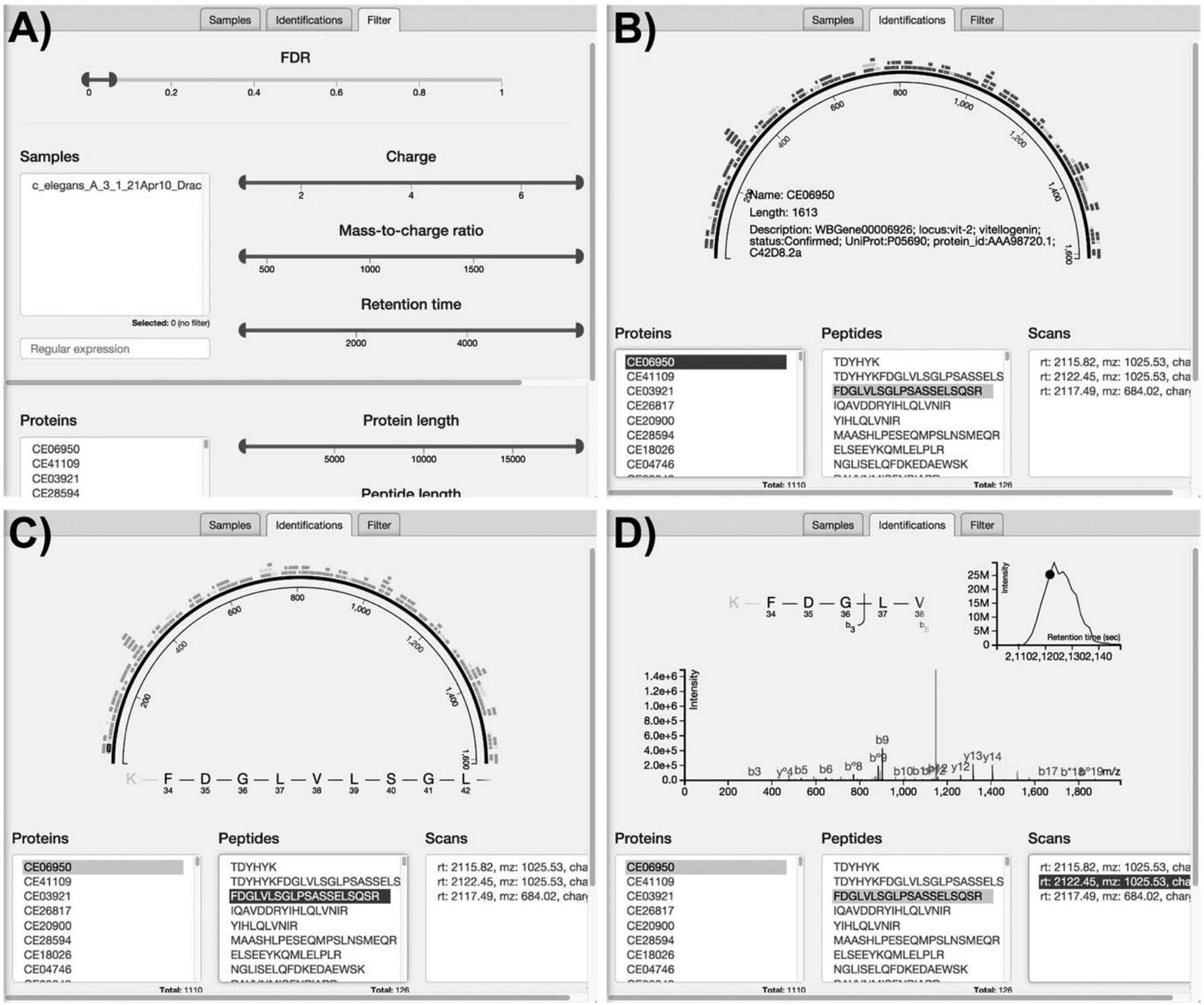

Figure 14.3 MSGFgui panels showing (A) filters, (B) protein, (C) peptide and (D) spectrum selection steps.

interested in looking at proteins with phosphorylation sites, selecting phosphorylation in the modification list will mean that only those proteins where phosphorylated peptides have been identified are visible. On the other hand it is not meant as a way to improve the quality of the results. The MSnID package, described in the next section, would be a better option for this task. The latter point also means that the filtering is not applied when exporting the results.

Further details on using MSGFgui are available in the package vignette.

14.4.4 The rTANDEM Package

X!Tandem is another popular MS/MS search engine[25] that is covered in Bioconductor with the rTANDEM package.[31] The rTANDEM implementation takes advantage of the R/C++ interface provided by the Rcpp package:[32] the X!Tandem C++ code is compiled during package installation and accessible from R. In rTANDEM, unlike in the MSGFplus implementation, the code of the MS/MS search engine is called directly by R as opposed to through the operating system utilities. rTANDEM's main tandem function takes as an

argument the path to an X!Tandem parameter file and returns the path to an X!Tandem output file. The package also offers functions to transform parameters or result files into R objects and *vice versa*, and to examine the results. See the rTANDEM and RforProteomics vignettes for a more detailed example.

```
library("rTANDEM")
## Setting parameters
taxonomy <- rTTaxo(taxon = "yeast",
                   format = "peptide",
                   URL = system.file("extdata/fasta/scd.fasta.
pro", package = "rTANDEM"))
## Running X!Tandem
param <- rTParam()
## Parsing results
result.path <- tandem(param)
results <- GetResultsFromXML(result.path)
proteins <- GetProteins(results, log.expect = −1.3, min.pep-
tides = 2)
peptides <- GetPeptides(protein.uid = "576", results)
```

As of rTANDEM version 1.10.0, X!Tandem yields results in its own original XML format. However, since the PILEDRIVER (2015.04.01) version of X!Tandem, it is possible to produce results in mzIdentML format by setting the "output, mzid" parameter to "yes". As soon as this option will become available in the rTANDEM package, it will be trivial to parse the results in mzIdentML format using the mzID or mzR packages, as described in Section 14.3 *Reading and handling mass spectrometry and proteomics data*.

14.4.5 The MSnID Package

The MSnID package[33] was developed with the idea of effective manipulation and filtering of the MS/MS identifications using R statistical and graphical capabilities. The core of the package is the MSnID data structure (same name as the package). The package provides utilities for constructing the object either by parsing a collection of mzIdentML or by providing the MS/MS identification results directly in a flat format (a data.frame). After collating the search results from multiple datasets, it assesses their identification quality and optimises filtering criteria to achieve the maximum number of identifications while not exceeding a specified FDR. The package also contains a number of utilities to explore the MS/MS results and assess missed and irregular enzymatic cleavages, mass measurement accuracy, *etc.* The following brief example outlines the key features of the package. More details can be found in the MSnID vignette.

The analysis starts with setting up a MSnID object by providing a path to the project directory. The main point of having a project directory is to

cache results (as provided by the R.cache package)[34] and avoid replicated operations.

```
library("MSnID")
msnid <- MSnID(workDir = ".")
```

MS/MS results are acquired by reading mzIdentML files (.mzid or .mzid.gz extensions). The read_mzIDs function uses the mzID package facilities. The example file c_elegans.mzid.gz, was produced by the MS-GF+ search engine.

```
msnid <- read_mzIDs(msnid, system.file("extdata", "c_elegans.
mzid.gz", package = "MSnID"))
```

Printing the MSnID object returns some basic information such as the working directory, the number of spectrum files used to generate the data, the number of peptide spectrum matches and corresponding FDR, the number of unique peptide sequences and corresponding FDR and the number of unique proteins or amino acid sequence accessions and corresponding FDR.

The FDR is defined here as the ratio of decoy accessions hits to the non-decoy (normal) accessions; in terms of forward and reverse protein sequences, this equates to the ratio of #reverse : #forward. While computing FDRs of PSMs and unique peptide sequences is trivial, the calculation of protein (accession) FDR is a subject of discussion in the field of proteomics. Here, protein (accession)-level FDR is computed the same way as in the IDPicker software[35] and simply constitutes a ratio of unique accessions from decoy component to non-decoy component of the sequence database.

```
show(msnid)

## MSnID object
## Working directory: "."
## #Spectrum Files: 1
## #PSMs: 19055 at 29% FDR
## #peptides: 9489 at 44% FDR
## #accessions: 7414 at 76% FDR
```

Particular properties of peptide sequences we are interested in are (1) irregular cleavages at the termini of the peptides and (2) missing cleavage sites within the peptide sequences. The RforProteomics vignette demonstrates how to apply and visualise these properties using the msnid object created previously in great detail.

The apply_filter function is used to filter the data. The second argument can be either (i) a string representing an expression that will be evaluated in the context of the MS/MS results or (ii) *n* dedicated MSnFilter object.

In this example we are going to retain only fully tryptic peptides (*i.e.* no irregular cleavage is allowed) and without any cleavages. Note, the reduction in FDR of PSM, peptide and protein identification.

```
msnid <- assess_termini(msnid, validCleavagePattern = "[KR]\\.
[^P]")
msnid <- assess_missed_cleavages(msnid, missedCleavagePattern
= "[KR](? = [^P$])")
msnid <- apply_filter(msnid, "numIrregCleavages = = 0")
msnid <- apply_filter(msnid, "numMissCleavages = = 0")
show(msnid)
```

```
## MSnID object
## Working directory: "."
## #Spectrum Files: 1
## #PSMs: 7573 at 8.1% FDR
## #peptides: 2978 at 15% FDR
## #accessions: 1996 at 33% FDR
```

Perhaps the key feature of the MSnID package is the flexibility for optimisation of the filtering criteria. Users can create specialised MSnIDFilter objects that can be used for storing, handling and optimisation filtering criteria. The object is initialised by a constructor function that takes an MSnID object as an argument so that the filter object is aware of all the parameters present in the data. Therefore all the criteria a user wishes to apply for data filtering (partitioning) must be present in the data. In practical terms this means that the parameters (*e.g.* mass measurement accuracy and scores) must be pre-transformed to the form that will be used in filter specification. For example, in the code chunk that follows, we create new score and ppm variables by taking −log10 of the MSGF+ *e*-value and the absolute value of the mass measurement error.

```
msnid$score <- -log10(msnid$`MS-GF:SpecEValue`)
msnid$ppm <- abs(mass_measurement_error(msnid))
```

Now we can initialise, specify, evaluate and, eventually, apply the filter object.

```
fltr <- MSnIDFilter(msnid)
fltr$score <- list(comparison = ">", threshold = 5.0)
fltr$ppm <- list(comparison = "<", threshold = 20.0)
show(fltr)
```

```
## MSnIDFilter object
## (score > 5) & (ppm < 20)
```

```
evaluate_filter(msnid, fltr)
```

```
## fdr n
## PSM 0.03236296 6667
## peptide 0.06622807 2431
## accession 0.16824034 1361
```

Filter parameters can be optimised using multiple methods. The objective of optimisations is to reach the maximum number of identifications while not exceeding an FDR at a specified level (PSM, peptide or accession).

Primarily they fall into two categories. The first is brute-force search (method = "Grid") of the combinations of the thresholds for all the parameters specified in the filter object.

```
fltr.grid <- optimize_filter(fltr, msnid, fdr.max = 0.01,
                             method = "Grid", level = "peptide",
                             n.iter = 1000)
```

A second category is based on Nelder-Mead (method = "Nelder-Mead") and simulated annealing (method = "SANN", as shown) approaches that need a starting value that will be optimised in terms of number of identifications.

```
fltr.sann <- optimize_filter(fltr, msnid, fdr.max = 0.01,
                             method = "SANN", level = "peptide",
                             n.iter = 1000)

show(fltr.sann)

## MSnIDFilter object
## (score > 7.6) & (ppm < 21)

evaluate_filter(msnid, fltr.sann)

## fdr n
## PSM       0.004363880 5984
## peptide   0.009694619 2083
## accession 0.024727992 1036
```

After applying data filters, the data can be accessed by psms, peptides, accessions or proteins methods depending on the follow-up step. In the case of spectral counting quantitative analysis the MSnID object can be converted to an MSnSet object (see Section 14.3 *Reading and handling mass spectrometry and proteomics data*). Spectral counting data can be explored and tested with the msmsEDA and msmsTests packages, reviewed in the next section.

```
msnid <- apply_filter(msnid, fltr.sann)
msnset <- as(msnid, "MSnSet")
```

Further details on package functionality and examples can be found at MSnID vignette.

14.4.6 Example

In the RforProteomics vignette, we provide a comprehensive example that ties some of these packages together. It demonstrates a real-world example, studying the effects of the daf-2 mutation, dietary restriction and age on the *C. elegans* proteome.[36,37] We start by downloading the raw data and FASTA files from ProteomeXchange using the rpx package (experiment PXD002161), proceed with peptide and protein identification using X!Tandem and rTANDEM and filter the MS/MS data with MSnID.

14.5 Analysis of Spectral Counting Data

14.5.1 Introduction

Spectral counting (see Chapter 8) is one the early quantitative methods used in bottom-up proteomics. Conceptually, it is similar to next-generation sequencing (NGS) read-counting methods such as RNA-Seq, ChIP-Seq, *etc.* The major distinction of LC-MS/MS proteomic spectral count data from NGS read count is the sampling rate or depth of the counts. A two order of magnitude lower (spectral) counts put some restrictions on the type of approaches that can be used for data analysis. A variety of statistical techniques have been tested and reviewed elsewhere.[38] In this section we will briefly review two related packages, namely msmsEDA[39] and msmsTests,[40] for exploratory data analysis and significance testing of spectral counting data.

14.5.2 Exploratory Data Analysis with msmsEDA

Exploratory Data Analysis (EDA) is used to identify the major components or factors explaining the variance in the data. Such factors may have real biological origins or could simply be nuisance-confounding factors artificially introduced during sample analysis. In EDA there is no preconceived notion about the origin of those factors. The package provides easy access to EDA methods such as PCA (Figure 14.4), hierarchical clustering (Figure 14.5) and

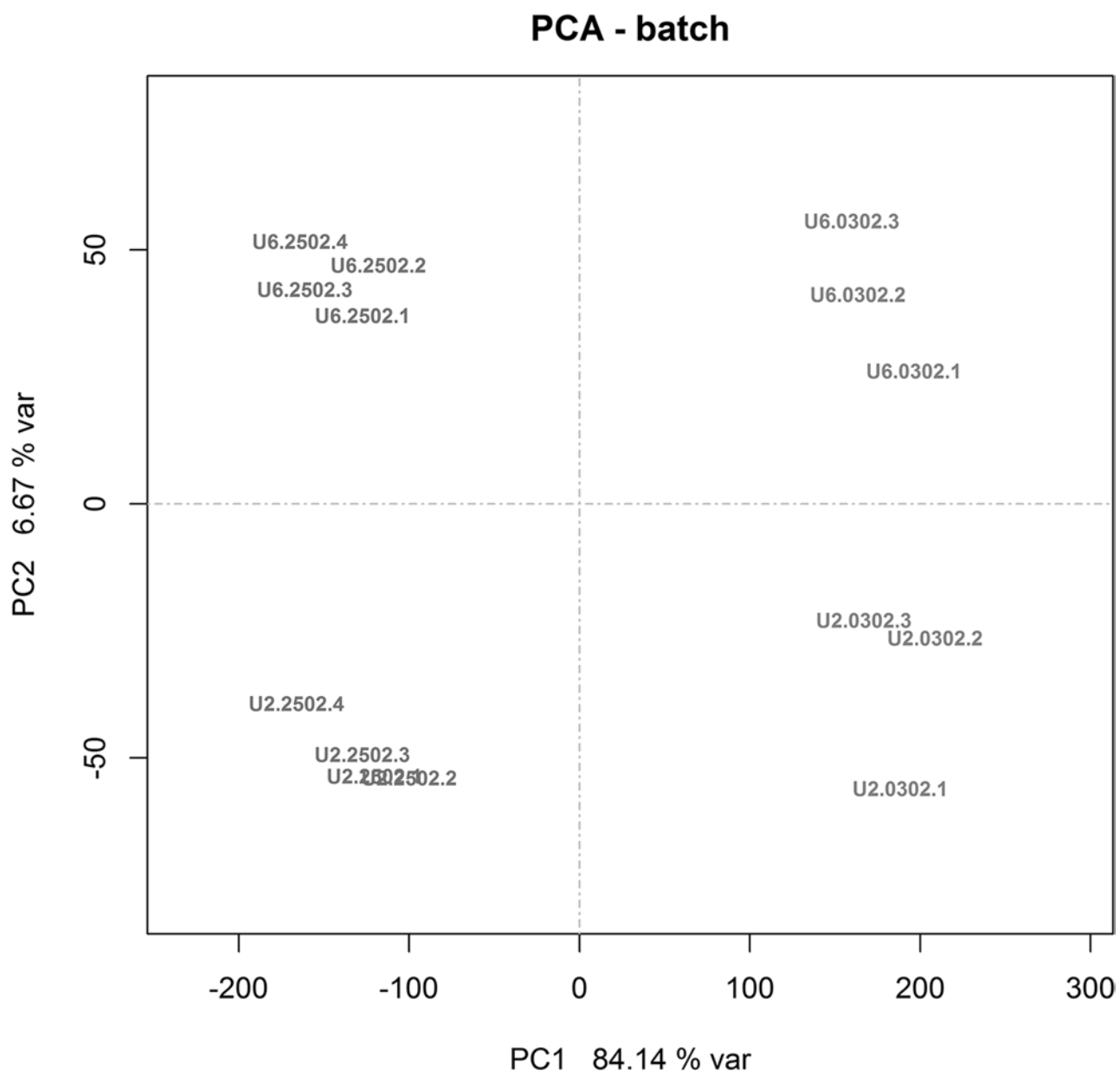

Figure 14.4 Demonstration of built-in PCA plot capability, illustrating the separation of the sample based on treatment (U2 *vs.* U6) and batch (2502 and 0302).

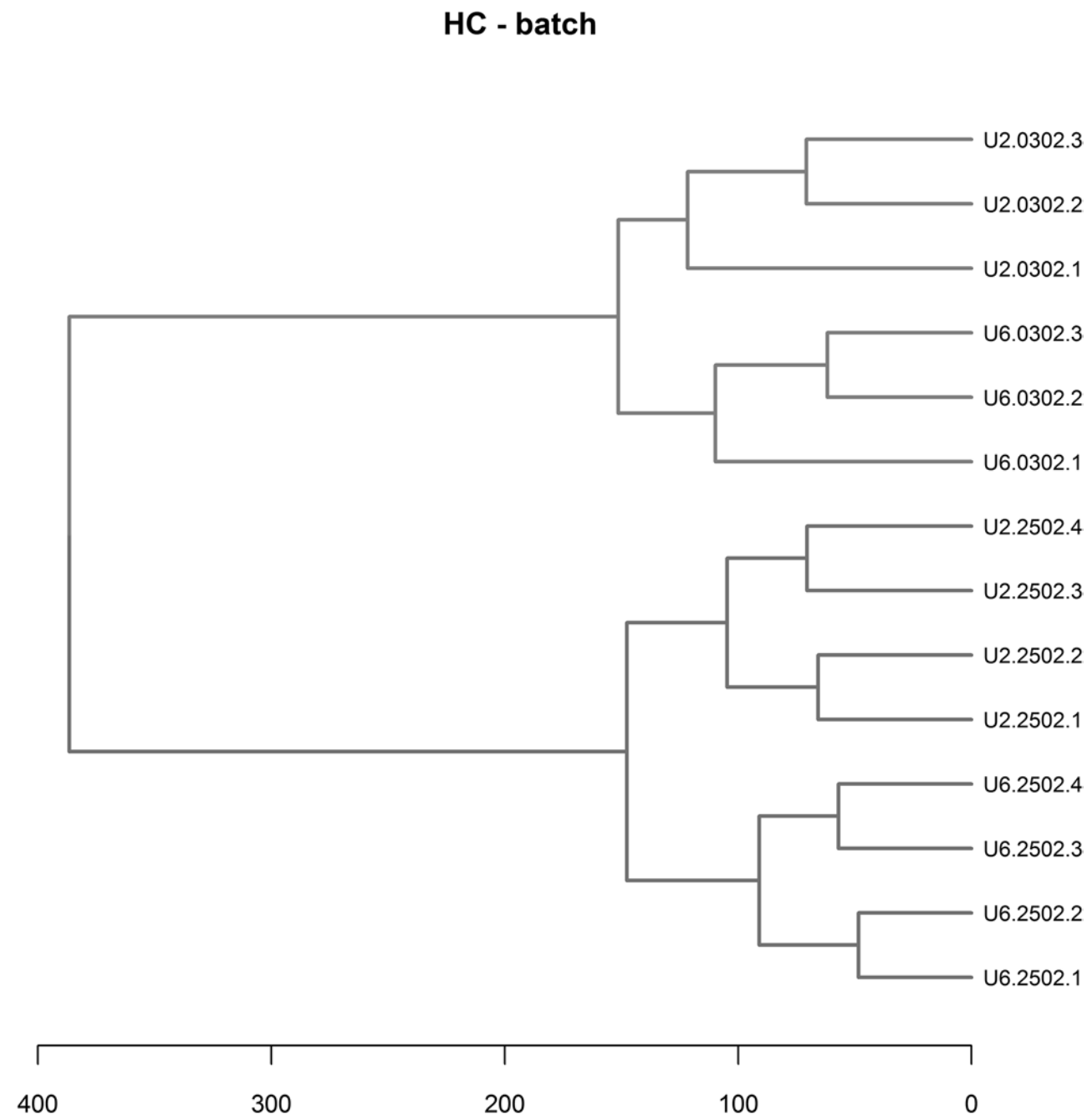

Figure 14.5 Visualisation of dendrograms is another capability of the msmsEDA package.

heatmap visualisation (Figure 14.6). To take advantage of the package, the data have to be pre-formatted into an MSnSet object, which is the central object type for quantitative proteomics-related packages in Bioconductor. Besides EDA tools the package has utility functions for generating summaries of spectral count statistics per sample.

```
library("msmsEDA")
data(msms.dataset) ## a test data
res <- counts.pca(msms.dataset,
       facs = pData(msms.dataset)[,"batch",drop = FALSE],
       snms = sampleNames(msms.dataset))
print(res$pc.vars[,1:4])
                             PC1       PC2       PC3       PC4
Standard deviation   163.82082 46.11839 32.27014 22.75523
Proportion of Variance 0.84135  0.06668  0.03265  0.01623
Cumulative Proportion  0.84135  0.90802  0.94067  0.95690
hcl <- counts.hc(msms.dataset, facs = pData(msms.dataset)
[,"batch",drop = FALSE])
msms.dataset <- pp.msms.data(msms.dataset)
counts.heatmap(msms.dataset, fac = pData(msms.dataset)[,"batch"])
```

Published on 15 November 2016 on http://pubs.rsc.org | doi:10.1039/9781782626732-00321

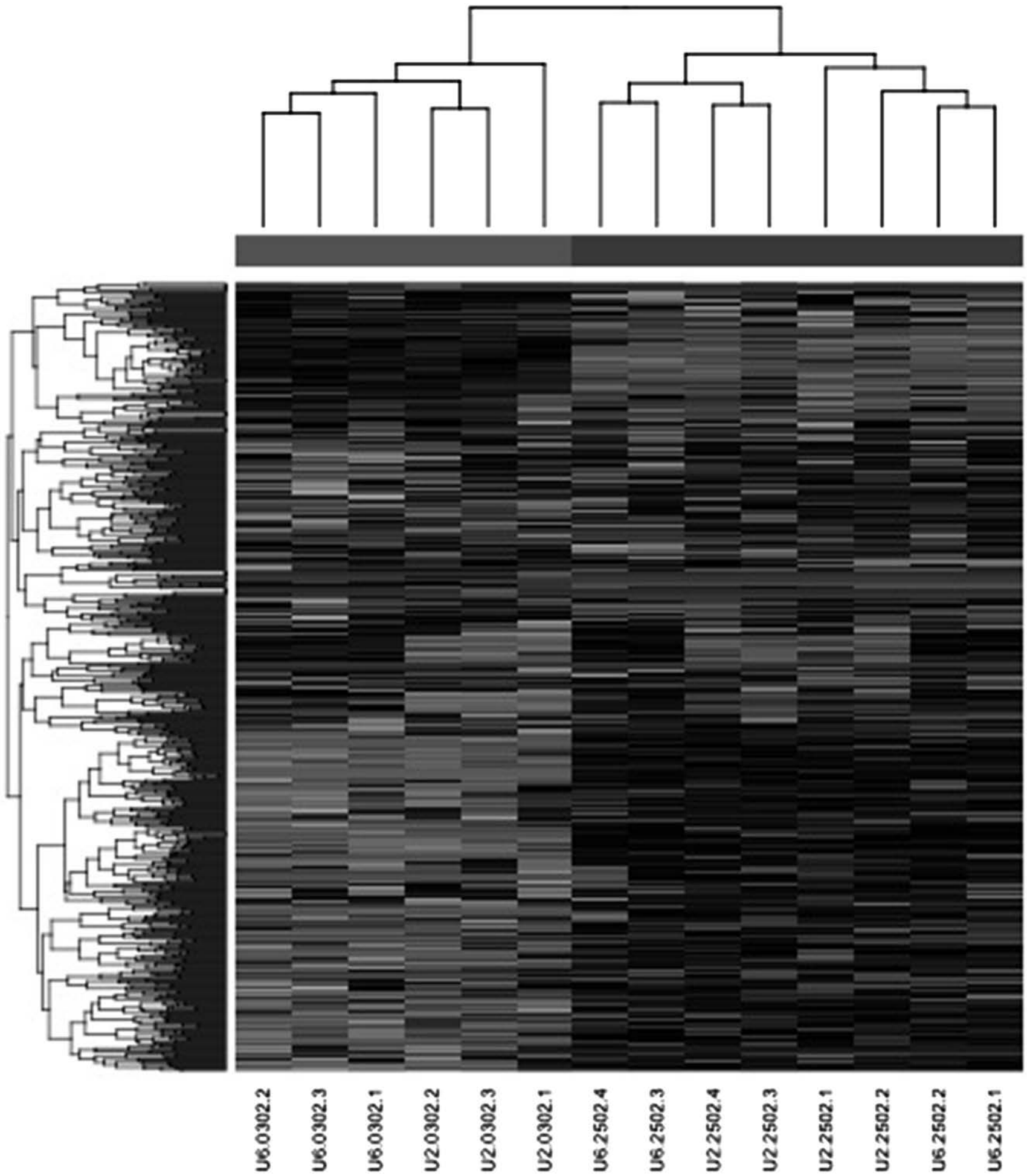

Figure 14.6 Pseudocoloring of relative protein abundances using heatmap capability. A coloured version of the figure is available in the Proteomics vignette.

14.5.3 Statistical Analyses with msmsTests

The msmsTests package contains a collection of statistical tests for label-free LC-MS/MS data by spectral counts, for the discovery of differentially expressed proteins between two biological conditions. The test is encoded with two models, one full model corresponding to the alternative hypothesis and the other, nested model, corresponding to the null hypothesis. Batch effects and other confounding factors can be conveniently accounted for in the model definitions. Three key tests are Poisson generalised linear models (GLM) regression, quasi-likelihood GLM regression and the negative binomial of the edgeR package, available through the msms.glm.pois, msms.glm.qlll and msms.edgeR functions.

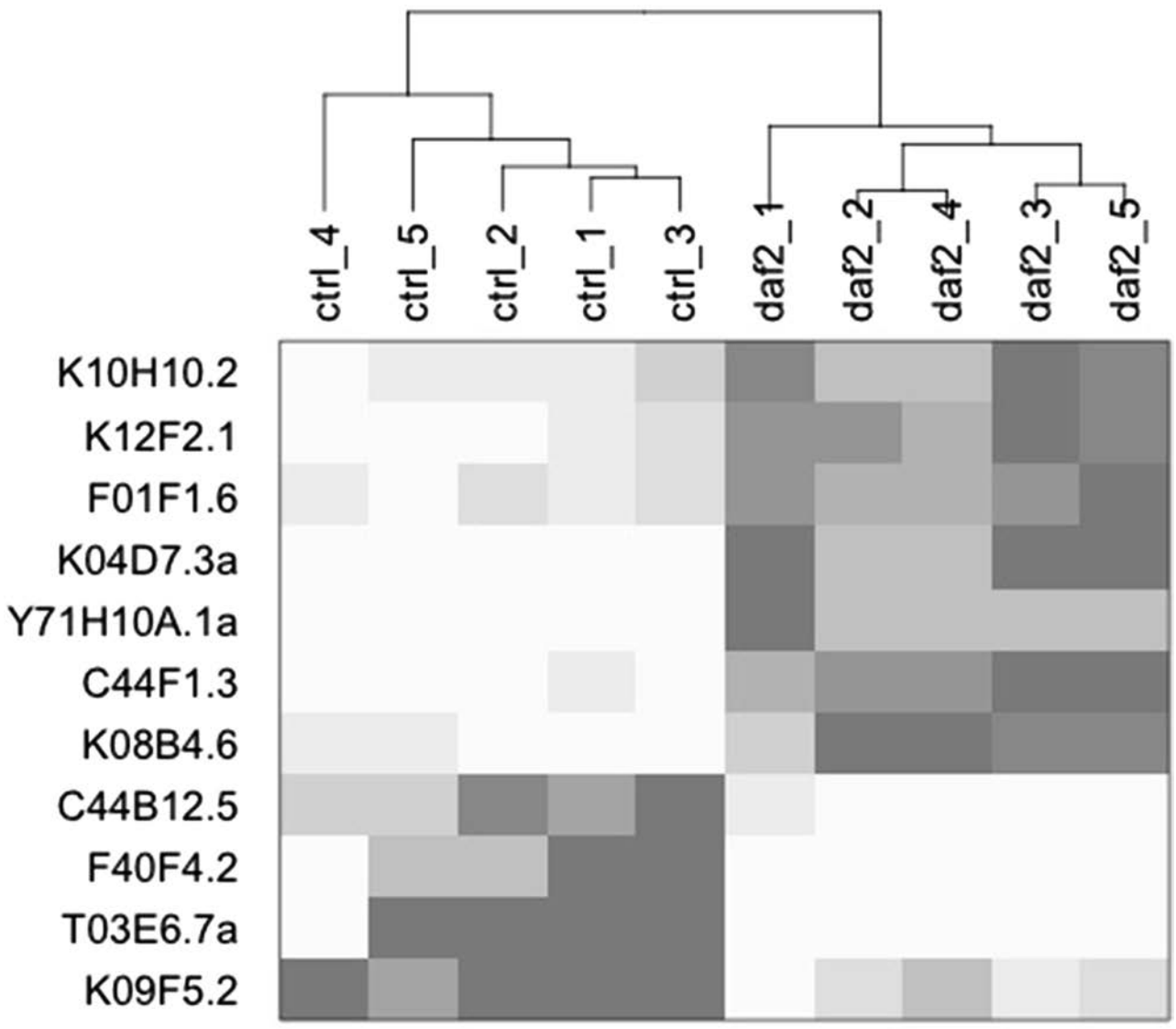

Figure 14.7 Heatmap of protein abundance changes for top-significant proteins. A coloured version of the figure is available in the Proteomics vignette.

```
ql.res <- msms.glm.qlll(msms.dataset,
                        form1 = "y ~ treat + batch",
                        form0 = "y ~ batch",
                        div = colSums(exprs(msms.dataset)))
ql.res$p.adj <- p.adjust(ql.res$p.value, method = "fdr")
sum(ql.res$p.adj < 0.05)
```

[1] 59

14.5.4 Example

In the RforProteomics vignette, we further explore and analyse the *C. elegans* data using the msmsEDA and msmsTests packages to identify statistically differentially expressed proteins. The heatmap in Figure 14.7 summarises the results for the proteins that have a fold change (up or down) of more than 2-fold and pass the 0.05 threshold for adjusted *p*-value.

14.6 MALDI and Mass Spectrometry Imaging

14.6.1 Introduction

The Matrix-assisted laser desorption–ionisation (MALDI) mass spectrometry is a soft ionisation technique resulting in mostly single or low-charged ions. As a result, it is widely used for the analysis of a wide range of biological

molecules and tissues and is very popular in microbiology and medicine to identify/classify bacterial species (fingerprinting), biomarkers or tissue composition (pattern recognition). Since it is an MS1-only technology, it is not suitable for peptide identification, and is generally followed by additional validation using ELISA or LC-MS/MS.

There are two main R packages designed for working with MALDI, data namely MALDIquant[41] and Cardinal.[42] While the first is a framework for the processing and analysis of MALDI-TOF and other 2D MS1-level data, the latter was written with mass spectrometry imaging (MSI) in mind.

In subsequent sections we will describe the preprocessing of MALDI spectra using MALDIquant and afterwards the analysis of MSI data with MALDIquant and Cardinal respectively.

14.6.2 MALDI Pre-Processing Using MALDIquant

MALDIquant provides a complete workflow for converting the raw MS1-level data into a matrix of feature intensities required for high-level analysis. The typical workflow is summarised on Figure 14.8 and some steps are detailed on Figure 14.9.

Each analysis with MALDIquant combines all or some of the following steps. First, we have to import the raw MS data into the R environment. MALDIquant-Foreign, an additional package to MALDIquant, offers multiple import functions for many vendor-specific and open file formats. Subsequently, the data are transformed for variance stabilisation and smoothed to remove high frequency noise. Next, a baseline correction is performed to remove the chemical background noise that is typical for MALDI data. Subsequently an intensity calibration step is necessary to allow comparison of intensity values across different spectra. Then, a peak detection algorithm is used to identify potential features and also to reduce the amount of data. As mass-to-charge ratios (*m*/*z*) differ across different spectra due to experimental settings, a peak alignment procedure is applied to adjust these differences accordingly. Finally, after peak binning, we obtain an intensity matrix that can be used as input for further statistical analysis, *e.g.* for variable selection or classification. In the following sections we will discuss each step in more detail.

14.6.2.1 Import Raw Data

The mass spectrometry community has to face a diverse list of vendor-specific and open file formats. MALDIquantForeign provides an easy way to import many raw MS data into MALDIquant objects (*e.g.* Bruker Daltonics

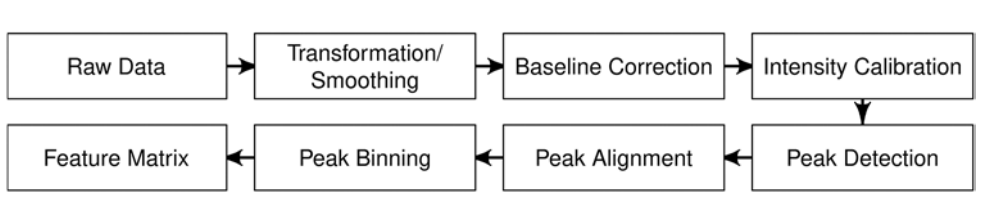

Figure 14.8 Typical MALDI workflow.

Published on 15 November 2016 on http://pubs.rsc.org | doi:10.1039/9781782626732-00321

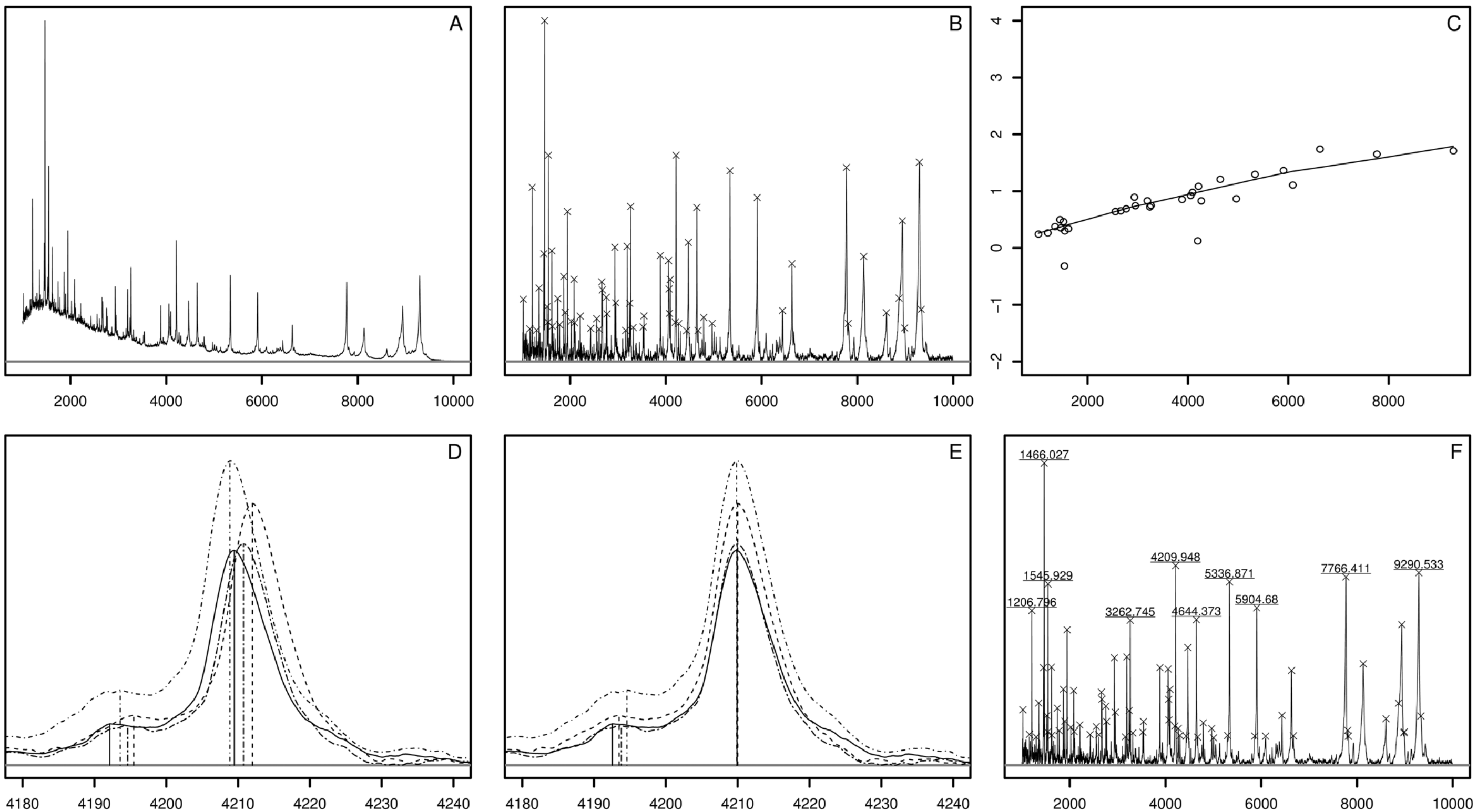

Figure 14.9 Illustration of the MALDIquant pipeline: raw MALDI spectrum (A); variance-stabilised, smoothed, baseline-corrected spectrum with detected peaks (B); fitted warping function for peak alignment (C); four unaligned peaks (D); four aligned peaks (E); merged spectrum with discovered and labelled peaks (F).

*flex Series files; see also Section 14.3 *Reading and handling mass spectrometry and proteomics data* for more information about the mzR[16] package). Among other features MALDIquantForeign reads whole directory trees and remote resources.

After importing the data, it is important to first run a quality control. MALDIquant provides functions to ensure all data have the same mass range, the same length, *etc.*, and offers multiple plotting functions to support visual quality checks.

```
library("MALDIquant")
library("MALDIquantForeign")

## load example data
data(fiedler2009subset)

## basic quality control
all(lengths(fiedler2009subset)==length(fiedler2009subset[[1]]))
all(sapply(fiedler2009subset, isRegular))
plot(fiedler2009subset[[14]]) # Figure 14.9(A)
```

14.6.2.2 Intensity Transformation and Smoothing

It is assumed that the intensity of MALDI mass spectrometry data follow approximately a Poisson distribution,[43] for which the variance depends on the mean. However, many statistical tests require a constant variance that is independent of the mean. Hence, we apply a square root transformation for variance-stabilisation and for an easier graphical visualisation. Other authors prefer stronger transformations, such as the logarithmic transformation,[44] which is also supported by MALDIquant.

To reduce small and high frequency variations, MS spectra need to be smoothed. MALDIquant offers the popular moving-average-smoother and the Savitzky-Golay-filter[45] methods. We favour the latter because it is based on polynomial regressions and, in contrast to the moving-average, it preserves the shape of the peaks.

```
spectra <- transformIntensity(fiedler2009subset, method = "sqrt")
spectra <- smoothIntensity(spectra, method = "SavitzkyGolay",
halfWindowSize = 10)
```

14.6.2.3 Baseline Correction

A typical MALDI spectrum is elevated by chemical noise such as matrix-effects and pollution. This so-called baseline influences the quantification of peak intensities and needs to be corrected. In recent years a lot of algorithms were developed for this but we focus on algorithms that preserve the peak shape and result in non-negative peak intensity values. MALDIquant provides three baseline correction methods:

1. The *convex hull* algorithm[46] doesn't require any tuning parameter but can't be applied to concave baselines that are often seen in MALDI spectra.
2. The *TopHat* algorithm is a combination of two morphological filters, namely moving-minimum and moving-maximum (erosion and dilaton)[47] steps. It has an additional parameter, the window size, controlling smoothness and accuracy of the baseline.
3. The *SNIP*[48] algorithm is the default baseline estimation algorithm in MALDIquant. It replaces the intensities in a window by the mean of the surrounding intensities, if the mean is smaller than the current intensity. The window size is decreasing iteratively starting from a user-defined limit.[49]

```
spectra <- removeBaseline(spectra, method = "SNIP", iterations
= 150)
plot(spectra[[14]]) # Figure 14.9(B)
```

While the algorithms shown are chosen for their favourable properties, MALDIquant implements the moving-median algorithm as well. It is commonly used in the community and the literature but may yield negative intensity values after the baseline correction.

14.6.2.4 Intensity Calibration

The intensity values in a MALDI mass spectrum are just a rough indicator of analyte abundance. Often the intensities are highly influenced by pre-analytical, analytical and environmental factors like sample collection, room temperature, crystallisation, operator, *etc.*[50] Because the systematic error could be stronger than the real biological effects it is important to minimise the former at the stage of data acquisition. However, it generally remains necessary to further calibrate the intensity values (often called normalisation) to compare them across different spectra.

MALDIquant provides two local and one global method for intensity calibration. The local methods, *Total Ion Current* (TIC) and *median* calibration, are applied to each spectrum individually. The third one, the *Probabilistic Quotient Normalization* (PQN),[51] is a global method that takes the information of all spectra into account. First all spectra are calibrated using the TIC calibration. Subsequently, a median reference spectrum is created and the intensities in all spectra are divided by the reference spectrum and a median calibration factor is calculated for each individual spectrum. This calibration factor is used to rescale the corresponding spectra.

As stated, the systematic errors are too strong to overcome by a simple recalibration of the intensities. Nevertheless it has been shown that applying intensity calibration is an essential step and that the *TIC* calibration is often the best choice.[52]

```
spectra <- calibrateIntensity(spectra, method = "TIC")
```

14.6.2.5 Peak Detection

The Peak Detection is used for two purposes. First it identifies relevant features and secondly it reduces the amount of data to be handled in further analysis steps. MALDIquant provides one of the most commonly used peak detection methods, based on local maxima.[53] A window is moved along the spectrum and local maxima are detected. If the local maxima are above the noise estimated by the Median-Absolute-Deviation (MAD) or Friedman's SuperSmoother,[55] they are considered as peaks. All maxima below the noise are discarded. Other authors prefer to use *wavelet*-based peak detection methods, which are already available in other R packages, namely MassSpecWavelet[56] and xcms.[54]

```
peaks <- detectPeaks(spectra, method = "MAD", SNR = 5, half-
WindowSize = 20)
plot(spectra[[14]])
points(peaks[[14]], pch = 4) # Figure 14.9(B)
```

14.6.2.6 Peak Alignment

Because of systematic errors like those described in the *Intensity Calibration* section, not only do the intensity values differ across spectra, but the *m*/*z* values do as well. To correct and equalise the *m*/*z* values in all spectra a recalibration, so-called alignment or warping, of all spectra is necessary.

MALDIquant uses a method that is known as peak-based *parametric time warping*.[57,58] It starts its alignment procedure by looking for stable peaks across all spectra, which are used as reference peak lists. Subsequently, MALDIquant looks for a locally weighted scatterplot smoothing (*LOWESS*) or polynomial-based function to warp the peaks of each spectrum against the reference peaks.

```
warpingFunctions <- determineWarpingFunctions(peaks)
peaks <- warpMassPeaks(peaks, warpingFunctions)
# Figure 14.9(C–E)
```

14.6.2.7 Peak Binning

After performing the warping, the peak positions are very similar but are not numerically identical yet, and are thus grouped into bins. MALDIquant sorts all *m*/*z* values in an ascending order and splits this list recursively at the largest gap until all *m*/*z* values in a bin are from different samples and their individual *m*/*z* values are in a small user-defined tolerance range around their mean. The latter becomes the new *m*/*z* value for all corresponding peaks in the bin. Finally MALDIquant generates an intensity matrix that can be used as input for further statistical analysis, *e.g.* for variable selection or classification.

```
peaks <- binPeaks(peaks)
intMatrix < - intensityMatrix(peaks)
```

14.6.2.8 Conclusion

MALDIquant is a versatile R package that provides a flexible analysis pipeline for MALDI-TOF and other 2D mass spectrometry data. We invite readers to consult the package web page‡ to find more features, examples and additional, detailed workflows.

14.6.3 Mass Spectrometry Imaging

Mass spectrometry imaging (MSI) combines mass spectra with their spatial information. A sample is divided in a coordinate grid and a mass spectrum is recorded for each point (x, y) enabling the visualisation and analysis of the spatial distribution of chemical compounds.

In general the preprocessing of each spectrum is very similar to the traditional mass spectrometry preprocessing described in the *MALDI preprocessing using MALDIquant* section. The spatial information is only used in the statistical analysis that follows the preprocessing.

14.6.3.1 Cardinal

Cardinal[42] is an R/Bioconductor package specifically designed for the analysis of MSI data. It offers a user-friendly interface to preprocessing and statistical methods for MSI. The following example is taken from the vignette *"Unsupervised analysis of MS images using Cardinal"*.[59]

```
library("CardinalWorkflows")
data(pig206, pig206_analyses)
image(pig206, mz = 256, col.regions = gradient.colors(100,
"black", "white")) ## Figure 14.10
```

The methods for baseline correction, peak detection and peak alignment implemented in Cardinal are different from those in MALDIquant. Nevertheless the workflow is similar. It starts with a *TIC* intensity calibration, followed by peak detection, peak alignment and a data reduction step:

```
pig206.norm <- normalize(pig206, method = "tic")
pig206.peaklist <- peakPick(pig206.norm, method = "simple",
SNR = 6)
pig206.peaklist <- peakAlign(pig206.peaklist, ref = pig206.
norm,
method = "diff", units = "ppm", diff.max = 200)
pig206.peaks <- reduceDimension(pig206.norm, ref = pig206.
peaklist, type = "height")
```

Subsequently one can investigate the spatial information using, for example, an unsupervised clustering method like spatial-aware k-means[60] (Figure 14.11):

‡http://strimmerlab.org/software/maldiquant/

```
pig206.skmg <- spatialKMeans(pig206.peaks, r = c(1, 2),
                             k = c(5, 10), method = "gaussian")
image(pig206.skmg, layout = c(2, 2),
      col = gradient.colors(10, "black", "white"))
```

Cardinal supports imzML[61] and the Analyze 7.5 format as input data. Beside the functions demonstrated, Cardinal implements further methods for unsupervised analysis such as principal component analysis, spatially-aware (SA) and spatially-aware structurally-adaptive (SASA) segmentation.[60] In addition

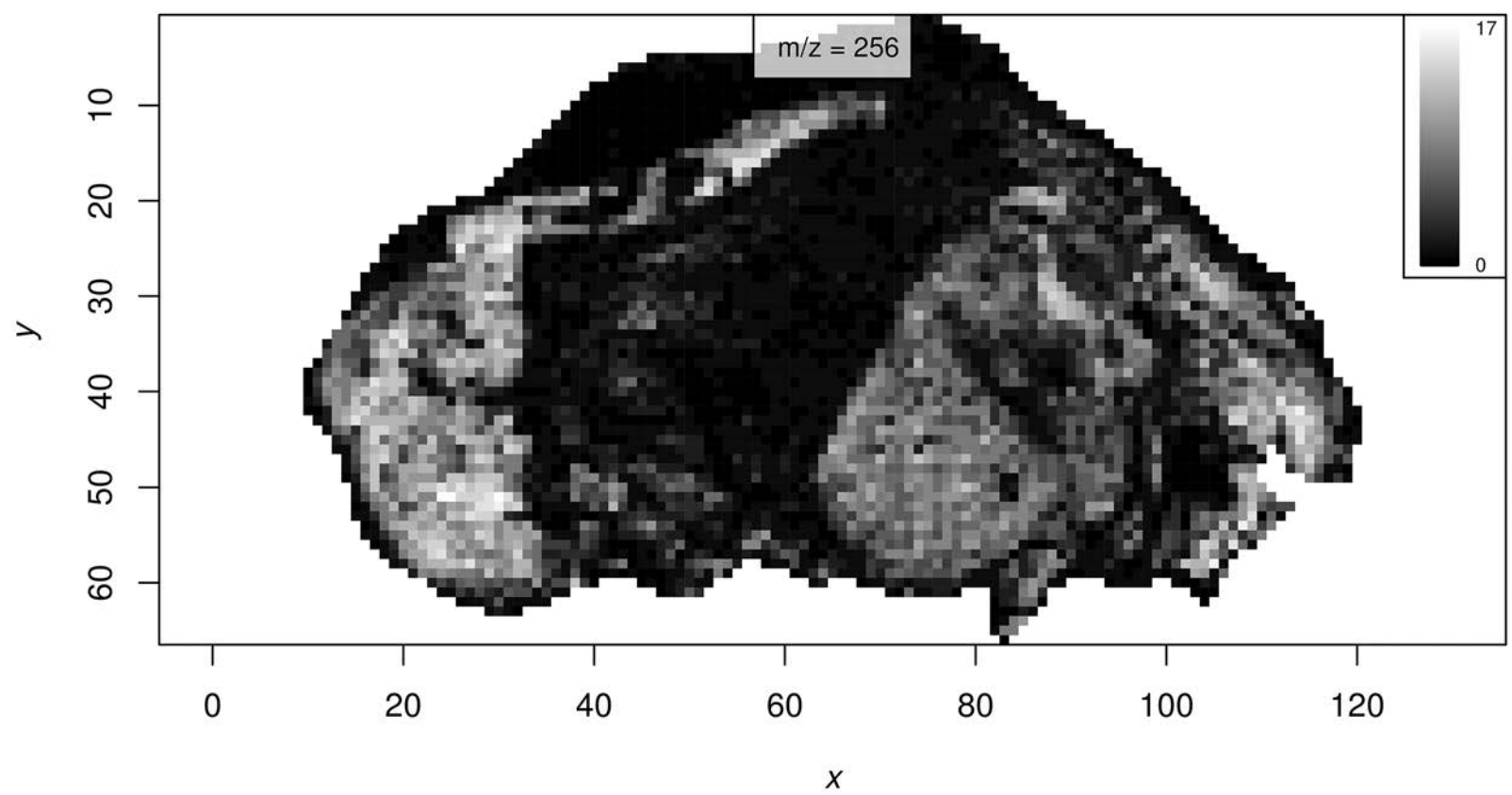

Figure 14.10 Ion image of a pig fetus at *m/z* 256 Da. A coloured version of the figure is available in the RforProteomics vignette.

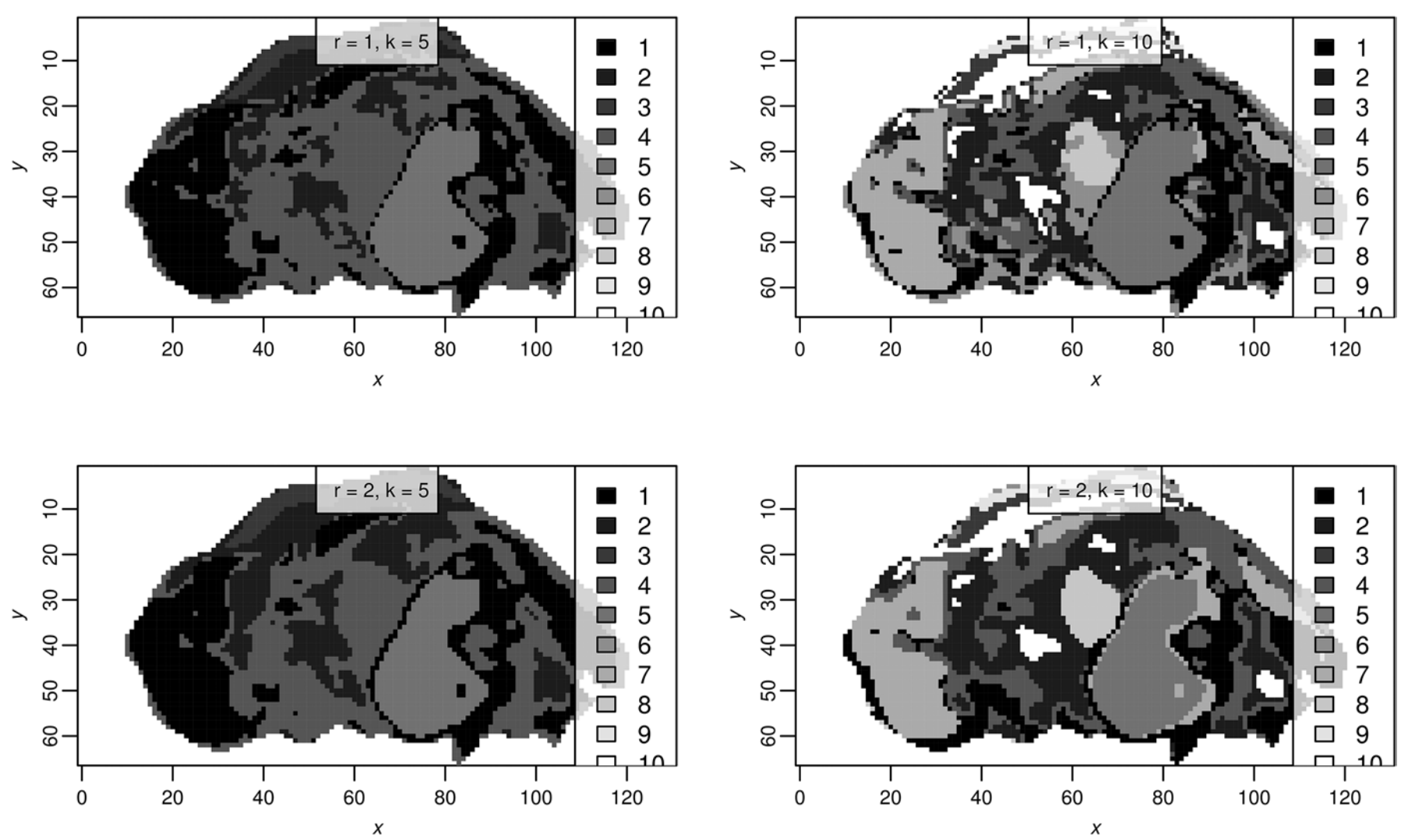

Figure 14.11 Segmental images for spatial-aware k-means using different smoothing radii ($r = c(1, 2)$) and number of segements ($k = c(5, 10)$). A coloured version is available in the RforProteomics vignette.

Cardinal comes with supervised algorithms like partial least squares discriminant analysis and orthogonal projections to latent structures discriminant analysis[62] and spatial shrunken centroids.[63] The package also has numerous plotting and visualisation functions. Interested readers are recommended to read the vignettes and workflows that are distributed with the package.

14.6.3.2 MALDIquant

While the focus of MALDIquant is on traditional mass spectrometry data analysis it also supports MSI: it can handle coordinates and divide the data into *m*/*z* slices to produce spatial visualisations.

```
## installed via biocLite("sgibb/MALDIquantExamples")
library("MALDIquantExamples")
spectra <- import(getPathNyakas2013())

spectra <- transformIntensity(spectra, method = "sqrt")
spectra <- smoothIntensity(spectra, method = "SavitzkyGolay",
halfWindowSize = 10)
spectra <- removeBaseline(spectra, method = "SNIP", iterations
= 10)
spectra <- calibrateIntensity(spectra, method = "TIC")

plotMsiSlice(spectra, center = 3364.079, tolerance = 0.5,
colRamp = colorRamp(c("black", "white")))
```

In contrast to Cardinal, MALDIquant lacks in more sophisticated spatial analysis methods.

14.6.3.3 Conclusion

Cardinal provides a powerful and user-friendly analysis pipeline for IMS data. Because it was carefully written and geared to R/Bioconductor programming philosophy it is easily extensible. More details and workflows can be found on the corresponding web page.§

14.7 Isobaric Tagging and Quantitative Data Processing

As explained in Chapter 8, isobaric tagging using iTRAQ[64] and TMT[65] isobaric tags is an efficient and widely used quantitative proteomics technique. It is well supported within the Bioconductor project. Two packages in particular can be used for such data. isobar[66] offers specific statistical modelling and can import processed data from csv spreadsheets and mgf files. MSnbase supports raw data quantitation and integration of identification data, relying

§http://cardinalmsi.org

on the infrastructure described in Section 14.3 *Reading and handling mass spectrometry and proteomics data*. Here, we describe different steps of a typical isobaric tagging experiment and quantitative proteomics data processing.

14.7.1 Quantification of Isobaric Data Experiments

The very first step consists of reading the raw data in mzML or mzXML files to create an MSnExp object (mx), which contains the MS2 spectra and their annotation. At this stage (step 2), individual or multiple spectra can be extracted and plotted. It is then possible to annotate each spectrum with the corresponding peptide-spectrum matches that have been generated from any third-party search engine and saved as an mzIdentML file (step 3). Note that this step could also be applied later in the pipeline, such as after quantitation of the isobaric tags. Quantitation (step 4) is performed by defining the *m*/*z* values of the ions of interest. The standard iTRAQ and TMT ions are predefined and are readily available, but users can define their own ReporterIons (a data structure that defines specific peaks of interest) that will be quantified. Quantification of the MSnExp object produces an MSnSet object, that we name qnt as follows.

1. Creation of a raw data MSnExp object containing all MSMS spectra:
   ```
   mx <- readMSdata("rawData.mzXML")
   ```
2. Extraction and plotting of a single spectrum:
   ```
   plot(mx[[100]], reporters = TMT10)
   ```
3. Addition of identification data:
   ```
   mx <- addIdentificationData(mx, "identData.mzid")
   ```
4. Quantification of TMT 10-plex reporter ions and creation of an MSnSet object:
   ```
   qnt <- quantify(mx, reporters = TMT10)
   ```

Note that as we have the flexibility to quantify any peaks of interest, we can quantify peaks that are characteristic of undissociated isobaric tags, to quantitatively measure incomplete dissociation and assess its impact on the quantitation accuracy. An example is provided in the MSnbase-demo vignette (available using vignette(“MSnbase-demo”)), which uses the iTRAQ5 ReporterIons to quantify the usual 114.1, 115.1, 115.1 and 117.1 peaks, as well as the undissociated 145.1 peak.

14.7.2 Processing Quantitative Proteomics Data

Proteomics datasets from data dependent analysis (DDA) workflows always contain a certain, sometimes substantial, proportion of missing values. Depending on the severity and nature of missing data, several options can be considered. In steps 5 and 6, we respectively filter out spectra that have more than 50% of missing values and impute the remainder missing values using nearest neighbour imputation. Missing data filtering and imputation are data dependent and require careful considerations, such as interesting

missing data patterns that could reflect protein presence or absence between experimental groups and the nature (random or non-random) of missing data. Finally, in steps 7 and 8, we combine spectra into protein (group) intensities using the *iPQF* method[67] and normalise the protein intensities using quantile normalisation.[68]

5. Removal of spectra containing more than 50% of missing values:
   `qnt <- filterNA(qnt, p = 0.5)`
6. Missing value imputation using nearest neighbours:
   `qnt <- impute(qnt, method = "knn")`
7. Quantitative data aggregation:
   `qprot <- combineFeatures(qnt, groupBy = "ProteinAccession", fun = "iPQF")`
8. Normalising protein-level quantitative data:
   `qprot <- normalise(qrot, method = "quantiles")`

All steps summarised in this section are thoroughly described and demonstrated in the MSnbase-demo vignette and the respective manual pages.

14.8 Machine Learning, Statistics and Applications

14.8.1 Introduction

R/Bioconductor provides an ideal environment for statistical computing, multivariate data analysis and machine learning. There is generic support for basic statistics, directly applicable to proteomics, but there also exists state-of-the-art biological data analysis packages designed specifically for the analysis of proteomics data, and many packages originally developed for genomics data analysis that can be directly applied to the field of proteomics. In this section we will discuss some of these current tools and packages in the frame of proteomics data analysis, including worked examples and use cases.

14.8.2 Statistics

To make confident inferences about biology, proteomics approaches must incorporate appropriate statistical measures of quantitative data. Over the last decade R has evolved to contain virtually every statistical method that the modern scientist would need. In the frame of statistical modelling and proteomics data analysis, Bioconductor offers a wide range of packages for the statistical analysis of biological data. Many of these packages are designed for the analysis of genomics data, and although there are many differences between the fields of genomics and proteomics, they also share many common statistical challenges and similar experimental designs. It is thus natural for one to use the sophisticated techniques that already are widely available in genomics and apply them in the field of proteomics to draw robust biological conclusions.

Quantitative proteomics experiments can be used to identify proteins that differ in abundance between treated and untreated cell populations, changes in disease state, *etc.* Labelling approaches such as stable isotope labelling by amino acids in cell culture (SILAC)[69] and 14N : 15N are popular and can be used to compare the ratios of observed peptides in two proteomes in a single LC-MS/MS) run. The evaluation of observed fold changes to draw conclusions about the underlying biological effects being measured can be challenging due to small sample sizes, complex experimental designs, normalisation issues, appropriate statistical analysis, and requirement for multiple testing adjustments. These issues are not unique to proteomics, in fact, many are common to the analysis of differential expression data from microarray experiments, and the limma (linear models for microarray data) package[70] that is widely used in genomics, provides functionality to address these problems. For normalisation there is lowess and quantile normalisation procedures, among others, and ratio *vs.* average plots (commonly called MA plots since their use became ubiquitous in microarray data analysis) can be useful for initially checking the distribution of ratios (see in particular *Visualization of proteomics data using R and Bioconductor*[7] for details about MA plots). limma provides a number of linear models that can accommodate more complex experimental designs. To statistically analyse relative abundance there is the empirical Bayes moderated *t*-test and other shrinkage methods appropriate for small sample sizes. Several packages such as limma, multtest[71] and *q*value[72] provide functionality for *p*-value adjustment to address the multiple-testing problem, including methods such as Bonferroni, Storey–Tibshirani and Benjamini–Hochberg.

For the analysis of spectral counting data, packages designed originally for count data from high-throughput sequencing assays contain analysis pipelines that are directly applicable. For example, DESeq,[73] DESeq2[74] and edgeR,[75] are all well documented packages where the Poisson and the negative binomial distributions, or the quasi-likelihood are considered. The msmsTests package (see Section 14.5 *Analysis of Spectral Counting Data*), which is based on edgeR provides functionality and protocols for analysing and statistically quantifying differential expression between two biological conditions. The tests available in msmsTests are based on a GLM model with offsets as normalising factors.

Other dedicated packages include MSstats,[21] applicable to a variety of proteomics workflows including label-free, SILAC and many types of fractionation, and also for a variety of data acquisition strategies *e.g.* LC-MS in data-dependent acquisition (DDA, or shotgun) mode, targeted selected reaction monitoring (SRM) and data-independent acquisition (DIA, or Sequential Windowed data independent Acquisition of the Total High-resolution Mass Spectra (SWATH-MS)). The package contains a wealth of statistical tools and approaches for the relative quantification of proteins and peptides. The MSstats pipeline consists of three mains steps: (1) data processing, including normalisation, visualisation and quality control, (2) statistical modelling and inference, including fitting an appropriate linear model, and

(3) statistical experimental design. Input data may also be in the form of an MSnSet, and therefore is interoperable with other packages that are specific to the analysis of proteomics data, such as MSnbase.[20] For isobaric tagging, such as iTRAQ and TMT, the isobar[66] package provides dedicated statistics for preprocessing, normalisation, and report generation. It also features a module for integrating and validating PTM-centric datasets (isobar-PTM). There is also the aLFQ[76] package in the CRAN repository for estimating absolute protein quantities from label-free LC-MS/MS.

14.8.3 Machine Learning

The field of machine learning (ML) is concerned with the design, development and application of data-driven algorithms that can learn and improve automatically through experience. To date, ML methods have been applied to address a broad range of areas within quantitative proteomics. Algorithms applied generally fall into the following broad categories: supervised, unsupervised and semi-supervised learning. All methods are concerned with the analysis of datasets containing multivariate observations. Supervised ML methods, also termed classification algorithms, aim to train classifiers to learn a mapping between a set of observed instances and a set of associated external attributes–class labels (This set of instances and labels is usually termed the training data.). The trained classifier can then be used to predict the class labels on data with unlabelled attributes. Unsupervised ML, also known as clustering, attempts to learn patterns and associations from a set of instances where there are no known class labels, and semi-supervised methods are algorithms that use a combination of both labelled and unlabelled instances.

There is a plethora of ML packages in R, and associated algorithms. We refer the reader to the *CRAN Task View: Machine Learning and Statistical Learning*¶ for a curated list and simple guide to all current packages and functions for ML in CRAN. Packages that are particularly relevant are meta-packages, for example MLInterfaces[77] in Bioconductor, and the mlR,[78] and caret[79] packages in CRAN, which provide direct, simple, unified interfaces to a breadth of ML methods. These meta-packages provide important ML infrastructure including functionality to re-sample your models, optimise hyperparameters, select features, cope with pre- and post-processing of your data and allow one to compare models in a statistically meaningful way.

In addition, there also exist specific packages, dedicated to the analysis of proteomics data using ML, namely the pRoloc package[80] and MLInterfaces. pRoloc was originally designed for the analysis of quantitative MS-based spatial proteomics data, however the algorithms that feature in the package can be applied to a wide range of ML problems, across a range of different proteomics pipelines. Both packages support the analysis of data stored in

¶https://cran.r-project.org/web/views/MachineLearning.html.

an MSnSet (see Section 14.3 *Reading and handling mass spectrometry and proteomics data*). They provide a complete infrastructure for unsupervised and supervised machine learning and data visualisation. pRoloc features a dedicated semi-supervised novelty detection algorithm for the identification of new clusters[81] and a transfer learning method for integrating data from heterogeneous sources.[82]

14.8.3.1 Supervised Machine Learning

Using the pRoloc pipeline for supervised ML, in the code chunk that follows, we demonstrate a typical classification analysis. Any supervised ML task can be broken down into a series of basic manageable steps; (i) acquiring data labels, (ii) training a model, (iii) evaluating model performance, and (iv) deploying your model for its intended task. In the examples that follow, we load data from a spatial proteomics experiment that was generated using the hyperLOPIT technology[8] on pluripotent mouse embryonic stem cells, and classify proteins with an unassigned spatial location to one of tens of sub-cellular niches, as described in detail in ref. 8. Briefly, cellular compartments, including organelles, vesicles and macromolecular complexes are separated along a continuous density gradient. A set of discrete fractions along the gradient is then sampled and their protein content identified and quantified. The protein quantitative profiles along the gradient reflect their original sub-cellular location. Given a set of sub-cellular markers, *i.e.* well-known residents, proteins of unknown or uncertain location can be matched to a sub-cellular niche based on the similarity of their profile to that of markers.

Step (i), acquiring class labels, requires one to find training examples. In the frame of predicting localisation using data generated from quantitative MS experiments (*i.e.* normalised ion intensities along a set of fractions for a set of proteins), class labels would be a set of known sub-cellular localisations for proteins in the data, *i.e.* well-known residents, termed marker proteins. An important factor to consider in one's choice of training examples, is how well they represent the multivariate data space over which the system performance will be measured, and a classifier will be learnt. pRoloc provides a convenience function, addMarkers, to directly add markers to an MSnSet object. These markers stem from a simple vector in R, a user-defined spreadsheet or, in case of sub-cellular localisations of proteins, a set of markers from previously published studies. Before one can generate a model on the training data and classify unknown residents, one has to take care of properly training the model parameters *i.e.* step (ii). It is widely known that wrongly-set parameters can have adverse effects on the classification performance and success of the learner to the same degree as using inappropriate training examples.

Parameter optimisation is fundamental to any ML application, not just a supervised schema, and can be conducted in a number of ways. A common approach is to optimise ones parameters using the convention of a *training set* (to model) and a *testing set* (to predict) which are subsets extracted

from the labelled training data. Using this schema, observed and expected classification results can be compared, and then used to assess how well a given model works by getting an estimate of the classifiers ability to achieve a good generalisation. A commonly used measure of classifier performance is the macro $F1$ score, $F1 = 2\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$, which is the harmonic mean of $\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}$ and $\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}$, such that tp = true positives and tn = true negatives. Using this protocol a grid search is often used to test a range of possible model parameters, and the best performing set of parameters is then used to construct a classifier using all labelled examples, which can be deployed to predict unlabelled instances. Estimation of the algorithmic performance, step (iii), can be assessed in many ways, such as *via* cross-validation. All packages mentioned previously provide functionality for cross-validation, in the example shown, the pRoloc package, is used to estimate algorithmic performance using stratified 20–80 partitioning, in conjunction with 5-fold cross-validation in order to optimise the free parameters of an Support Vector Machine (SVM) *via* a grid search. This procedure is usually repeated 100 times and then the best parameter(s) are selected upon investigation of associated macro $F1$ scores. A high macro $F1$ score indicates that the labelled instances *i.e.* the marker proteins, in the test dataset, are consistently correctly assigned by the algorithm. Often more than one parameter or set of parameters gives rise to the best generalisation accuracy (see Figure 14.12). As such, it is always important to investigate the model parameters and critically assess the best choice. The best choice may not be as simple as the parameter set that gives rise to the highest macro $F1$ score and one must be careful to avoid over-fitting and to choose parameters wisely. Once the best parameters have been selected they can then be used to build a classifier from the training data of organelle markers.

In the following example, we apply a weighted SVM classifier for protein classification. The labelled training data (Figure 14.12, left) were constructed from a manually curated marker set by experts in the field. Using the pRoloc package we employ a weighted SVM with a Gaussian kernel to learn a non-linear decision function on the training data to map proteins of unknown localisation to one of the known organelle classes. Class specific weights were used when creating the SVM model, which were set to be inversely proportional to the class frequencies to account for class imbalance. On the training data the two free SVM parameters, cost and sigma, were optimised over 100 rounds of stratified 5-fold cross-validation *via* a grid search and the best pair of parameters for the classifier were chosen from evaluation of the macro $F1$ scores (Figure 14.12, middle). The optimised SVM classifier was then used to predict protein localisation on the unlabelled data (Figure 14.12, right). The size of the points reflects the classification probabilities.

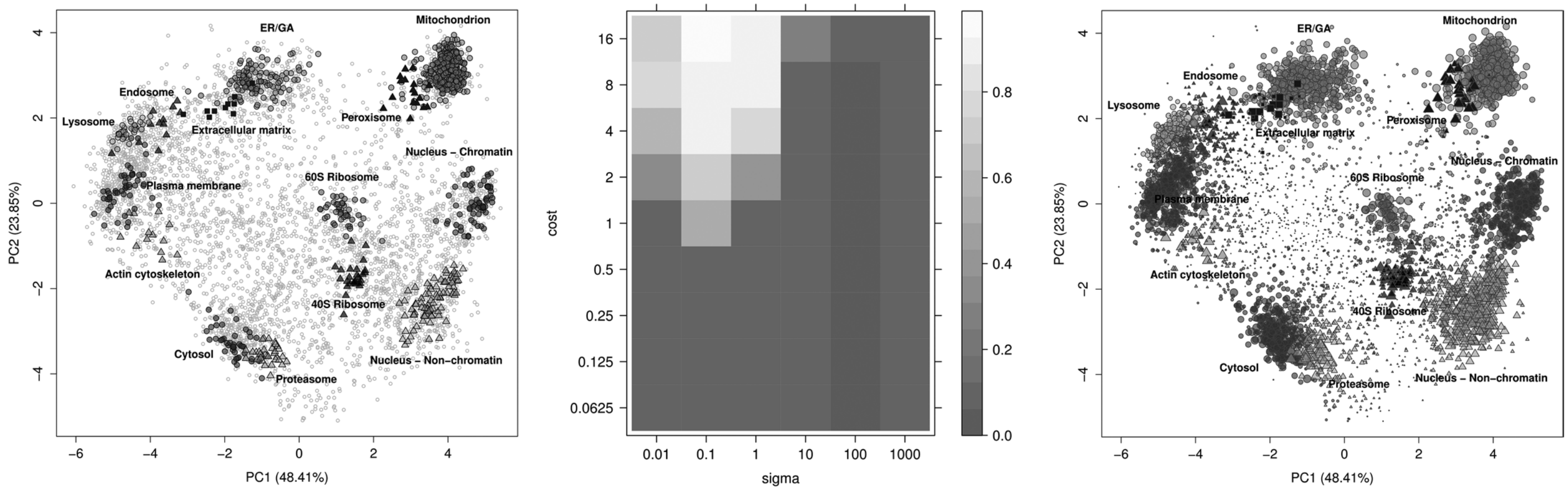

Figure 14.12 Application of a Support Vector Machine classifier (SVM) on hyperLOPIT data on mouse embryonic stem cells. Left: principal components analysis plot displaying the labelled input training data, one point represents one protein. Middle: grid search for the SVM parameters cost and sigma, highlighting optimal pairs of parameters. Right: application of a weighted SVM classifier. The size of the points reflects the classification probabilities.

```
library('pRoloc')
library('pRolocdata')

## (i) data, inlusing labelled and unlabelled instances
data("hyperLOPIT2015")

## class weights
w <- table(fData(hyperLOPIT2015)[, "markers"])
w <- 1/w[names(w) ! = "unknown"]
## (ii) training the SVM
params <- svmOptimisation(hyperLOPIT2015, fcol = "markers",
                          times = 100, xval = 5,
                          class.weights = w)
## (iii) evaluation of model parameters
levelPlot(params)
## (iv) classification
res <- svmClassification(hyperLOPIT2015, params)
## visualising classification results
ptsze <- exp(fData(res)$svm.scores) - 1
plot2D(res, fcol = "svm", cex = ptsze)
addLegend(res, where = "bottomleft", cex = .5, bty = "n")
```

14.8.3.2 Unsupervised Machine Learning

Unsupervised machine learning usually refers to clustering, *i.e.* finding structure in a quantitative, generally multi-dimensional dataset of unlabelled data. The prerequisites to performing unsupervised machine learning are (1) a dataset to cluster or a sub-selection of interesting features to cluster, (2) a choice of similarity metric for the comparison of samples, and (3) the choice of an algorithm to use. As mentioned, there are many clustering algorithms available in R/Bioconductor, and popular choices include the k-means, hierarchical and kernalised methods. More information can be found on the *CRAN Task View: Cluster Analysis & Finite Mixture Models* page[‖] and the respective package vignette.

14.8.4 Conclusion

R/Bioconductor provides several dedicated packages for the analysis of proteomics data, and a plethora of packages dedicated to the analysis of genomics data that are directly applicable to the field of proteomics in general. However, between and within different packages, algorithms and computational methods, the underlying theory and statistics are generally the same. Furthermore, one should remember that the choice of algorithm is not the most important consideration—the success of the learner is dependent on good data and appropriate training of your model is vital to obtaining robust results.

[‖]https://cran.r-project.org/web/views/Cluster.html.

14.9 Conclusions

In this chapter, we have presented some mature R and Bioconductor infrastructure for the analysis of mass spectrometry-based proteomics. There are however many topics that we have not addressed. We invite interested users to explore available software by browsing relevant categories (termed *biocViews*) using either the proteomicsPackages and massSpectrometryPackages functions from the RforProteomics package, or by directly browsing these categories on the Bioconductor software page**. There, readers will discover, among many others, packages such as proteoQC[83] and qcmetrics[84] for the quality control and assessment of mass spectrometry and proteomics data, synapter[85] for the analysis of MS^E data, PAA[86] for the analysis of protein arrays, specL[87] and SWATH2stats[88] for targeted and SWATH-MS data, TPP[89] for the analysis. of thermal proteome profiling experiments, various dedicated annotation packages such as rols,[90] an interface to the ontology look-up service, hpar,[91] to access data from the Human Protein Atlas[92] and generic pathway and Gene Ontology annotations or the analysis of mass spectrometry-based metabolomics data. Visualisation, including interactive visualisation, are other strengths of the R environment that we have not specifically addressed here but are reviewed in Gatto *et al.*[7] and in RforProteomics's *Visualisation of proteomics data using R and Bioconductor* vignette.

The wealth of software packages available from Bioconductor and other R repositories, the flexibility of the environment, the expressiveness of the programming language and R's well established infrastructure for reproducible research have made it a major player in data driven biology. While at times intimidating, R and Bioconductor benefit from a very active and helpful community, well crafted documentation, innumerable tutorials and reference material, as well as support forums and mailing lists to help new and seasoned users; all these resources are accessible from the Bioconductor web page†† and detailed in the RforProteomics vignette.

The R and Bioconductor ecosystem is of course far from providing a complete solution to every imaginable computational pipeline or question in mass spectrometry and proteomics. The goal of this community effort is of course not to replace some of the high quality software that are already available. The strengths of R and Bioconductor enable the community to build upon existing infrastructure and develop new software to address specific questions and do better data-driven and reproducible computational research and data analysis.

References

1. R Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2015.

**http://bioconductor.org/packages/release/BiocViews.html#___Software.
††http://bioconductor.org/.

2. R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang and J. Zhang, *Genome Biol.*, 2004, **5**, R80.
3. W. Huber, V. J. Carey, R. Gentleman, S. Anders, M. Carlson, B. S. Carvalho, H. C. Bravo, S. Davis, L. Gatto, T. Girke, R. Gottardo, F. Hahne, K. D. Hansen, R. A. Irizarry, M. Lawrence, M. I. Love, J. MacDonald, V. Obenchain, A. K. Ole's, H. Pagés, A. Reyes, P. Shannon, G. K. Smyth, D. Tenenbaum, L. Waldron and M. Morgan, *Nat. Methods*, 2015, **12**, 115–121.
4. L. Gatto, *Figshare*, 2015.
5. L. Gatto, K. D. Hansen, M. R. Hoopmann, H. Hermjakob, O. Kohlbacher and A. Beyer, *J. Proteome Res.*, 2016, 809–814.
6. L. Gatto and A. Christoforou, *Biochim. Biophys. Acta*, 2014, **1844**, 42–51.
7. L. Gatto, L. M. Breckels, T. Naake and S. Gibb, *Proteomics*, 2015, **15**, 1375–1389.
8. A. Christoforou, C. M. Mulvey, L. M. Breckels, A. Geladaki, T. Hurrell, P. C. Hayward, T. Naake, L. Gatto, R. Viner, A. Martinez Arias and K. S. Lilley, *Nat. Commun.*, 2016, **7**, 8992.
9. J. A. Vizcaíno, E. W. Deutsch, R. Wang, A. Csordas, F. Reisinger, D. Ríos, J. A. Dianes, Z. Sun, T. Farrah, N. Bandeira, P. A. Binz, I. Xenarios, M. Eisenacher, G. Mayer, L. Gatto, A. Campos, R. J. Chalkley, H. J. Kraus, J. P. Albar, S. Martinez-Bartolomé, R. Apweiler, G. S. Omenn, L. Martens, A. R. Jones and H. Hermjakob, *Nat. Biotechnol.*, 2014, **32**, 223–226.
10. L. Gatto, *rpx: R interface to the proteomeXchange repository*, 2015.
11. M. Morgan, M. Carlson, D. Tenenbaum and S. Arora, *AnnotationHub: Client to access annotationHub resources*, 2016.
12. L. Gatto and A. Sonali, *ProteomicsAnnotationHubData: Transform public proteomics data resources into Bioconductor data structures*, 2015.
13. L. Martens, M. Chambers, M. Sturm, D. Kessner, F. Levander, J. Shofstahl, W. H. Tang, A. Römpp, S. Neumann, A. D. Pizarro, L. Montecchi-Palazzi, N. Tasman, M. Coleman, F. Reisinger, P. Souda, H. Hermjakob, P.-A. Binz and E. W. Deutsch, *Mol. Cell. Proteomics*, 2011, **10**, R110.000133.
14. P. G. A. Pedrioli, J. K. Eng, R. Hubley, M. Vogelzang, E. W. Deutsch, B. Raught, B. Pratt, E. Nilsson, R. H. Angeletti, R. Apweiler, K. Cheung, C. E. Costello, H. Hermjakob, S. Huang, R. K. Julian, E. Kapp, M. E. McComb, S. G. Oliver, G. Omenn, N. W. Paton, R. Simpson, R. Smith, C. F. Taylor, W. Zhu and R. Aebersold, *Nat. Biotechnol.*, 2004, **22**, 1459–1466.
15. S. Orchard, L. Montechi-Palazzi, E. W. Deutsch, P.-A. Binz, A. R. Jones, N. Paton, A. Pizarro, D. M. Creasy, J. Wojcik and H. Hermjakob, *Proteomics*, 2007, **7**, 3436–3440.
16. M. C. Chambers, B. Maclean, R. Burke, D. Amodei, D. L. Ruderman, S. Neumann, L. Gatto, B. Fischer, B. Pratt and J. Egertson, *Nat. Biotechnol.*, 2012, **30**, 918–920.
17. A. R. Jones, M. Eisenacher, G. Mayer, O. Kohlbacher, J. Siepen, S. J. Hubbard, J. N. Selley, B. C. Searle, J. Shofstahl, S. L. Seymour, R. Julian, P. A.

Binz, E. W. Deutsch, H. Hermjakob, F. Reisinger, J. Griss, J. A. Vizcaíno, M. Chambers, A. Pizarro and D. Creasy, *Mol Cell Proteomics*, 2012, **11**, M111.014381.
18. T. L. Pedersen, V. A. Petyuk, L. Gatto and S. Gibb, *mzID: An mzIdentML parser for R*, 2015.
19. D. T. Lang and the CRAN Team, *XML: Tools for parsing and generating XML within R and S-Plus*, 2015.
20. L. Gatto and K. S. Lilley, *Bioinformatics*, 2012, **28**, 288–289.
21. M. Choi, C.-Y. Chang, T. Clough, D. Broudy, T. Killeen, B. MacLean and O. Vitek, *Bioinformatics*, 2014, **30**, 2524–2526.
22. J. Griss, A. R. Jones, T. Sachsenberg, M. Walzer, L. Gatto, J. Hartler, G. G. Thallinger, R. M. Salek, C. Steinbeck, N. Neuhauser, J. Cox, S. Neumann, J. Fan, F. Reisinger, Q. W. Xu, N. Del Toro, Y. Pérez-Riverol, F. Ghali, N. Bandeira, I. Xenarios, O. Kohlbacher, J. A. Vizcaíno and H. Hermjakob, *Mol Cell Proteomics*, 2014, **13**, 2765–2775.
23. J. Cox, I. Matic, M. Hilger, N. Nagaraj, M. Selbach, J. V. Olsen and M. Mann, *Nat. Protoc.*, 2009, **4**, 698–705.
24. J. K. Eng, A. L. McCormack and J. R. Yates, *J. Am. Soc. Mass Spectrom.*, 1994, **5**, 976–989.
25. R. Craig and R. C. Beavis, *Bioinformatics*, 2004, **20**, 1466–1467.
26. D. N. Perkins, D. J. Pappin, D. M. Creasy and J. S. Cottrell, *Electrophoresis*, 1999, **20**, 3551–3567.
27. S. Kim, N. Gupta and P. A. Pevzner, *J. Proteome Res.*, 2008, **7**, 3354–3363.
28. T. L. Pedersen, *MSGFplus: An interface between R and MS-GF+*, 2015.
29. J. Verzani, *gWidgets: gWidgets API for building toolkit-independent, interactive GUIs, R package version 0.0-54*, 2014.
30. T. L. Pedersen, *MSGFgui: A shiny GUI for MSGFplus*, 2015.
31. F. Fournier, C. J. Beauparlant, R. Paradis and A. Droit, *rTANDEM: Interfaces the tandem protein identification algorithm in R*, 2014.
32. D. Eddelbuettel and R. Francois, *J. Stat. Software*, 2011, **40**, 1–18.
33. V. Petyuk and L. Gatto, *MSnID: Utilities for exploration and assessment of confidence of LC-MSn proteomics identifications*, 2016.
34. H. Bengtsson, *R.cache: Fast and light-weight caching (memoization) of objects and results to speed up computations*, 2015.
35. B. Zhang, M. C. Chambers and D. L. Tabb, *J. Proteome Res.*, 2007, **6**, 3549–3557.
36. G. Depuydt, F. Xie, V. A. Petyuk, N. Shanmugam, A. Smolders, I. Dhondt, H. M. Brewer, D. G. Camp 2nd, R. D. Smith and B. P. Braeckman, *Mol Cell Proteomics*, 2013, **12**, 3624–3639.
37. G. Depuydt, F. Xie, V. A. Petyuk, A. Smolders, H. M. Brewer, D. G. Camp 2nd, R. D. Smith and B. P. Braeckman, *J. Proteome Res.*, 2014, **13**, 1938–1956.
38. M. Li, W. Gray, H. Zhang, C. H. Chung, D. Billheimer, W. G. Yarbrough, D. C. Liebler, Y. Shyr and R. J. C. Slebos, *J. Proteome Res.*, 2010, **9**, 4295–4305.
39. J. Gregori, A. Sanchez and J. Villanueva, *msmsEDA: Exploratory data analysis of LC-MS/MS data by spectral counts*, 2014.

40. J. Gregori, A. Sanchez and J. Villanueva, *msmsTestsP: LC-MS/MS differential expression tests*, 2013.
41. S. Gibb and K. Strimmer, *Bioinformatics*, 2012, **28**, 2270–2271.
42. K. D. Bemis, A. Harry, L. S. Eberlin, C. Ferreira, S. M. van de Ven, P. Mallick, M. Stolowitz and O. Vitek, *Bioinformatics*, 2015, **31**, 2418–2420.
43. M. Sköld, T. Rydén, V. Samuelsson, C. Bratt, L. Ekblad, H. Olsson and B. Baldetorp, *Bioinformatics*, 2007, **23**, 1401–1409.
44. R. Tibshirani, T. Hastie, B. Narasimhan, S. Soltys, G. Shi, A. Koong and Q.-T. Le, *Bioinformatics*, 2004, **20**, 3034–3044.
45. A. Savitzky and M. J. E. Golay, *Anal. Chem.*, 1964, **36**, 1627–1639.
46. M. A. Andrew, in *Information processing letters 9*, Elsevier, 1979, pp. 216–219.
47. M. van Herk, *Pattern Recognit. Lett.*, 1992, **13**, 517–521.
48. C. G. Ryan, E. Clayton, W. L. Griffin, S. H. Sie and D. R. Cousens, *Nucl. Instrum. Methods Phys. Res., Sect. B*, 1988, **34**, 396–402.
49. M. Morhác, *Nucl. Instrum. Methods Phys. Res., Sect. A*, 2009, **600**, 478–487.
50. K. A. Baggerly, J. S. Morris and K. R. Coombes, *Bioinformatics*, 2004, **20**, 777–785.
51. F. Dieterle, A. Ross, G. Schlotterbeck and H. Senn, *Anal. Chem.*, 2006, **78**, 4281–4290.
52. W. Meuleman, J. Y. Engwegen, M.-C. W. Gast, J. H. Beijnen, M. J. Reinders and L. F. Wessels, *BMC Bioinf.*, 2008, **9**, 88.
53. Y. Yasui, D. McLerran, B. Adam, M. Winget, M. Thornquist and Z. Feng, *J. Biomed. Biotechnol.*, 2003, **4**, 242–248.
54. C. A. Smith, E. J. Want, G. O'Maille, R. Abagyan and G. Siuzdak, *Anal. Chem.*, 2006, **78**, 779–787.
55. J. H. Friedman, *A variable span smoother*, DTIC Document, 1984.
56. P. Du, W. A. Kibbe and S. M. Lin, *Bioinformatics*, 2006, **22**, 2059–2065.
57. T. G. Bloemberg, J. Gerretzen, H. J. P. Wouters, J. Gloerich, M. van Dael, H. J. C. T. Wessels, L. P. van den Heuvel, P. H. C. Eilers, L. M. C. Buydens and R. Wehrens, *Chemom. Intell. Lab. Syst.*, 2010, **104**, 65–74.
58. R. Wehrens, T. Bloemberg and P. H. C. Eilers, *Bioinformatics*, 2015, **31**, 3063–3065.
59. K. D. Bemis and A. Harry, *Unsupervised analysis of MS images using Cardinal*, 2015.
60. T. Alexandrov, M. Becker, S.-O. Deininger, G. Ernst, L. Wehder, M. Grasmair, F. von Eggeling, H. Thiele and P. Maass, *J. Proteome Res.*, 2010, **9**, 6535–6546.
61. T. Schramm, A. Hester, I. Klinkert, J.-P. Both, R. M. A. Heeren, A. Brunelle, O. Laprévote, N. Desbenoit, M.-F. Robbe, M. Stoeckli, B. Spengler and A. Römpp, *J. Proteomics*, 2012, **75**, 5106–5110.
62. A. L. Dill, L. S. Eberlin, C. Zheng, A. B. Costa, D. R. Ifa, L. Cheng, T. A. Masterson, M. O. Koch, O. Vitek and R. G. Cooks, *Anal. Bioanal. Chem.*, 2010, **398**, 2969–2978.
63. R. Tibshirani, T. Hastie, B. Narasimhan and G. Chu, *Stat. Sci.*, 2003, **18**, 104–117.

64. P. L. Ross, Y. N. Huang, J. N. Marchese, B. Williamson, K. Parker, S. Hattan, N. Khainovski, S. Pillai, S. Dey, S. Daniels, S. Purkayastha, P. Juhasz, S. Martin, M. Bartlet-Jones, F. He, A. Jacobson and D. J. Pappin, *Mol Cell Proteomics*, 2004, **3**, 1154–1169.
65. A. Thompson, J. Schäfer, K. Kuhn, S. Kienle, J. Schwarz, G. Schmidt, T. Neumann, R. Johnstone, A. K. A. Mohammed and C. Hamon, *Anal. Chem.*, 2003, **75**, 1895–1904.
66. F. P. Breitwieser, A. Müller, L. Dayon, T. Köcher, A. Hainard, P. Pichler, U. Schmidt-Erfurth, G. Superti-Furga, J.-C. Sanchez, K. Mechtler, K. L. Bennett and J. Colinge, *J. Proteome Res.*, 2011, **10**, 2758–2766.
67. M. Fischer and B. Y. Renard, *Bioinformatics*, 2016, **32**, 1040–1047.
68. B. M. Bolstad, R. A. Irizarry, M. Astrand and T. P. Speed, *Bioinformatics*, 2003, **19**, 185–193.
69. S.-E. Ong, *Mol. Cell. Proteomics*, 2002, **1**, 376–386.
70. G. K. Smyth, *Stat. Appl. Genet. Mol. Biol.*, 2004, **3**, 3.
71. K. S. Pollard, S. Dudoit and M. J. van der Laan, Multiple testing procedures: R multtest package and applications to genomics, in *Bioinformatics and computational biology solutions using R and Bioconductor*, Springer, 2005.
72. J. D. Storey, *qvalue: Q-value estimation for false discovery rate control*, 2015.
73. S. Anders and W. Huber, *Genome Biol.*, 2010, **11**, R106.
74. M. I. Love, W. Huber and S. Anders, *Genome Biol.*, 2014, **15**, 550.
75. M. D. Robinson, D. J. McCarthy and G. K. Smyth, *Bioinformatics*, 2010, **26**, 139–140.
76. G. Rosenberger, C. Ludwig, H. L. Röst, R. Aebersold and L. Malmström, *Bioinformatics*, 2014, **30**, 2511–2513.
77. V. Carey, R. Gentleman and J. Mar, *MLInterfaces: Uniform interfaces to R machine learning procedures for data in Bioconductor containers*, 2016.
78. B. Bischl, M. Lang, J. Richter, J. Bossek, L. Judt, T. Kuehn, E. Studerus and L. Kotthoff, *mlr: Machine learning in R*, 2015.
79. M. Kuhn, *caret: Classification and regression training*, 2015.
80. L. Gatto, L. M. Breckels, S. Wieczorek, T. Burger and K. S. Lilley, *Bioinformatics*, 2014, **30**, 1322–1324.
81. L. M. Breckels, L. Gatto, A. Christoforou, A. J. Groen, K. S. Lilley and M. Trotter, *J. Proteomics*, 2013, **88**, 129–140.
82. L. M. Breckels, S. Holden, D. Wojnar, C. M. Mulvey, A. Christoforou, A. J. Groen, O. Kohlbacher, K. S. Lilley and L. Gatto, *Learning from heterogeneous data sources: an application in spatial proteomics*, Cold Spring Harbor Laboratory Press, 2015.
83. B. Wen and L. Gatto, *proteoQC: An R package for proteomics data quality control*, 2016.
84. L. Gatto, *qcmetrics: A framework for quality control*, 2016.
85. N. J. Bond, P. V. Shliaha, K. S. Lilley and L. Gatto, *J Proteome Res*, 2013, **12**, 2340–2353.
86. M. Turewicz, *ProteinArrayAnalyzer (PAA): a novel R/Bioconductor package for autoimmune biomarker discovery with protein microarrays*, 2016.

87. C. Panse, C. Trachsel, J. Grossmann and R. Schlapbach, *Bioinformatics*, 2015.
88. P. Blattmann, M. Heusel and R. Aebersold, *SWATH2stats: Transform and filter SWATH data for statistical packages*, 2015.
89. D. Childs, H. Franken, M. Savitski and W. Huber, *TPP: Analyze thermal proteome profiling (TPP) experiments*, 2016.
90. L. Gatto, *rols: An r interface to the ontology lookup service*, 2016.
91. L. Gatto, *hpar: Human protein atlas in R*, 2016.
92. M. Uhlen, P. Oksvold, L. Fagerberg, E. Lundberg, K. Jonasson, M. Forsberg, M. Zwahlen, C. Kampf, K. Wester, S. Hober, H. Wernerus, L. Björling and F. Ponten, *Nat. Biotechnol.*, 2010, **28**, 1248–1250.

Section IV

Integration of Proteomics and Other Data

CHAPTER 15

Proteogenomics: Proteomics for Genome Annotation

FAWAZ GHALI[a,b] AND ANDREW R. JONES*[a]

[a]Institute of Integrative Biology, University of Liverpool, Biosciences Building Crown Street, Liverpool, L69 7ZB, UK; [b]School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, Chester Street, Manchester, M1 5GD, UK
*E-mail: andrew.jones@liverpool.ac.uk

15.1 Introduction

The recent advances in sequencing technology, through the development of next-generation sequencing (NGS) platforms, have enabled the generation of genomes for a multitude of species. The volumes of data that can be generated by NGS technologies also allow for the generation of genome sequences for individuals within populations, rather than solely a single representative genome for the whole species. From these genomes, the generation of gene models, and the downstream protein sequence databases produced, underpin a vast array of biological and biomedical investigations, including most areas of the Life Sciences. As such, a critical step in genome biology is the generation of accurate gene models, understanding of gene splicing (including alternative splicing) and discovery of single nucleotide polymorphisms (SNPs) within individuals or populations.

The process of defining gene models: finding the start codon, stop codon, 5′ and 3′ untranslated regions (UTRs) and exon-intron boundaries, is often

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

called "genome annotation". The term genome annotation can also be used to describe the assignment of functions to genes, although in this chapter, we restrict our usage of the terminology to the former, narrower definition. In most genome annotation pipelines, bioinformatics software is run to predict genes, taking into account a variety of evidence. Evidence can include (i) the intrinsic structure of the genome itself, including GC content of exons *versus* non-coding DNA, motifs for promoters and UTRs and so on; (ii) comparative genomic data – using the predicted gene model structure or protein sequences from related species; (iii) experimental data, including sequencing of the mRNA pool by NGS methods (or Expressed Sequence Tags (ESTs)) in older methods. There is a growing appreciation that large scale LC-MS proteomics data sets can play an important role in genome annotation in so called *proteogenomics* approaches, which are the subject of this chapter.

Proteogenomics can be defined as the use of proteomics data from LC-MS to enhance genome annotation by providing supporting evidence for some or all of the following concepts, depending on the workflow:[1] (i) that a predicted gene model is transcribed and translated into protein sequence *i.e.* moving it from the level of "predicted transcript" to having confirmed protein-level evidence; (ii) that a predicted splice event occurs biologically, for example through the confident identification of peptides that map to the different exons that have been predicted to be spliced together; (iii) for the discovery of new genes, by mapping mass spectra directly against proteins that are not part of the official gene models, for example predicted by *de novo* gene prediction directly from the genome or by searching a six frame translation; (iv) for supporting the existence of alleles (different copies of the same gene within individuals or a sub-population), for example derived from SNPs, through the confirmation or discovery of a protein sequence with a changed amino acid content to the reference allele (and protein); (v) for the confirmation of the start codon of genes through the confident identification of the N-terminal peptide within the protein.

Proteogenomics techniques do not generally have the level of sensitivity of RNA Sequencing (RNA-Seq), which is now used very commonly in genome annotation pipelines. RNA-Seq techniques are able to provide mRNA evidence for all expressed genes, down to a few copies per cell. The data from RNA-Seq comprises a large pool of sequencing reads of length ~30–300 base pairs, depending on the platform and protocol. The mRNA reads can be mapped (aligned) back onto the genome, including using tools that are able to account for gaps in the alignment due to splicing, enabling *de novo* discovery of the exon-intron splicing that occurred in the cell. These techniques are undoubtedly powerful but they have not completely solved the annotation problem. First, RNA-Seq data tends to include the UTRs and so does not solve the problem of the identification of a start codon, since ATG (the typical start codon) can appear both upstream of the actual start codon or internal to the coding region. The correct stop codon is usually more straightforward to identify, as long as exons have been correctly predicted, since stop codons do not also code for amino acids. Second, RNA-Seq protocols can also extract non-coding RNA (RNA that does

not code for protein), and thus not all the data gives evidence that a gene is protein-coding, and in some cases could be misleading. Third, RNA-Seq data sets emerging appear to suggest that alternative splicing of genes may be far more common than previously expected, and mRNA can be mapped to many regions of the genome and combinations of exons not previously suspected to be protein-coding. It is currently unknown whether such extensive splicing is actually producing mRNAs that are translated into proteins or whether many of these entities are in fact some form of regulatory feedback loop. The inclusion of proteomics data into a genome annotation pipeline provide the ability to demonstrate that events predicted, based solely on *in silico* methods, or from RNA-Seq data, are indeed producing protein in the cell.

However, proteogenomics techniques are challenging to perform from an informatics point of view for some of the following reasons. First, as discussed in Chapter 3, the most popular method for identification is the sequence database search, using tandem MS (LC-MS/MS data), to produce sets of peptide-spectrum matches (PSMs). The search engine requires a protein sequence database, which in most cases is derived directly or indirectly from a set of "official gene models". Depending on the species being analysed, the official gene models are updated and released at intervals, in some cases yearly or more frequently. As such, proteogenomics techniques must be dynamic, as the "gold standard" (*i.e.* the considered best set of gene models at one point in time), is regularly changing. It is not straightforward to map proteomics data between different releases of gene models without repeating searches, which can be computationally intensive. Second, statistical approaches for validating peptide identifications (Chapter 4) and protein identifications (Chapter 5) have generally been designed under the assumption that the sequence databases searched do not include extensive redundancy, and that sequences shared between different database entries (proteins) are due to biological events (*e.g.* gene families giving rise to paralogues with shared sequences). In proteogenomics approaches, as will be described in this chapter, it is common to merge different possible sets of protein sequences, in some cases producing databases orders of magnitude larger than the official protein set for a given species. It is an open question as to whether standard methods for statistical significance, such as the use of target-decoy searches, are entirely appropriate and statistically sound in proteogenomics. Third, since proteogenomics approaches tend to require considerably larger search databases than regular proteomics studies, increased computing power and parallelisation is needed to process large volumes of input spectra. Fourth, it is common for RNA-Seq data sets (and other data types supporting genome annotation, such as orthologous sequences from other species) to be visualised in the context of a genome, for example *via* genome browser software. The file formats and visualisations have thus not generally been designed with proteomics data, and associated complexity (for example, around protein groups – see Chapter 5) in mind. Fifth, it can be a challenge relating protein sequence databases (and accessions or identifiers for proteins), back to gene model databases. Genome databases such as

Ensembl have a formal mapping between predicted proteins and transcripts, but such a process is not completely straightforward when linking external well-curated protein databases such as UniProt back to source genomes for different species.

This chapter first introduces the main methods used in proteogenomics from a theoretical point of view. We next discuss some software pipelines specifically designed for proteogenomics, and some options for visualisation of results. We briefly describe file formats and standards that support proteogenomics results, and end with a discussion of open challenges and future directions of research in this area.

15.2 Theoretical Underpinning

In this section we discuss the theoretical aspects of proteogenomics. The key aspect of all pipelines is to identify peptide sequences from tandem MS spectra, and demonstrate the position or positions in the genome from where the parent proteins of those peptides were likely transcribed and translated. As discussed in previous chapters there are several informatics methods available for identifying peptide sequences from fragment spectra. In theory, *de novo* sequencing strategies have a particular advantage for proteogenomics, in that these approaches do not rely on well annotated gene models (giving protein sequences to search against), and thus could be used for finding previously unannotated coding regions. However, in practice *de novo* sequencing algorithms are exploring a vast search space (all possible peptide sequences), and thus rarely have the sensitivity and accuracy of sequence database search methods. Spectral library searching is not generally considered an appropriate method for large-scale proteogenomics, as these methods rely upon having previously annotated library entries (spectra) with the identity of peptides, and thus they are not ideally suited for finding novel peptide sequences – necessary for improving gene annotations. Hybrid search methods that first employ a partial *de novo* sequencing stage to produce short sequence tags (tag search methods), followed by filtering a database of possible peptides to reduce the search space also present an option for proteogenomics, although tag-based search algorithms have generally not matched the popularity of standard sequence database search algorithms. As such, in most proteogenomics approaches, the sequence database search method is used, and thus the pipelines require a well-designed database to search against. The database used often contains not only the protein set derived from the official gene models, but is usually enriched with alternatives possibilities, for example produced by running gene finding software *de novo* or by obtaining sets of different gene predictions from those genome databases (*e.g.* Ensembl) that release both "official gene models" and "putative or predicted gene models" – considered currently to be lacking in evidence to be promoted to the official set. One of the key challenges in proteogenomics approaches is achieving the optimal database design, enabling the discovery of novel peptides where they exist in the sample, while also controlling the

database size. A larger database leads to lower statistical power and increased chance of finding false positives.

15.2.1 Gene Prediction

A common step in many proteogenomics approaches is to run gene finding software *de novo*, or to use predicted sets of gene finders generated by external groups or resources, such as providers of genome databases. Gene structure prediction by "gene finding" software is typically the first step of a genome annotation pipeline. This is can be done either by using mathematical techniques that use intrinsic evidence in the DNA sequences to find a gene structure or by using external evidence of various types to enhance the accuracy of gene prediction. In the following sections we discuss these two techniques: *ab initio* gene prediction and evidence-based gene prediction.

15.2.1.1 Ab initio *Gene Prediction*

Ab initio gene prediction methods use mathematical/computational techniques based on intrinsic evidence in the DNA sequences rather than external evidence to find a gene and its intron-exon structure. In *ab initio* prediction techniques, the genomic DNA sequence is searched for certain signals that are commonly observed in protein-coding genes but not in intergenic regions. Examples include the GC content, which tends to be higher in genes than intergenic regions, sequences motifs for splicing, motifs for start codons, UTRs and so on. One of the advantages of *ab initio* gene prediction is that it does not require external evidence to identify a gene and to determine the intron-exon structure. However *ab initio* gene predictors generally can at best output a ranked list of possible gene structures at a given locus (for eukaryotic genomes containing introns as the challenge is easier in prokaryotes lacking introns), and struggle to identify alternatively spliced isoforms with confidence.

Ab initio methods can have high sensitivity in that they are able to detect the presence of most genuine genes, but with a cost in terms of specificity, dependent upon the quality of training data.[2] It would be expected that correct prediction of intron-exon structure does not generally exceed 60–70% accuracy.

15.2.1.2 Evidence-Based Gene Prediction

In evidence-based gene prediction, external experimental evidence from expressed sequence tags (ESTs) sequencing (historically), or more recently, direct messenger RNA (mRNA) sequencing (RNA-Seq) data are used to enhance the accuracy of gene prediction. RNA-Seq data can be mapped back against the genome, using a variety of bioinformatics applications,[3] which can account for gaps in the alignment due to splicing, thus allowing *de novo* discovery of introns. Evidence-based gene predictions are generally more

accurate than *ab initio* gene predictions, but require additional cost in terms of data collection.

Moreover, it is also possible to combine and integrate *ab initio* gene prediction with evidence-based gene prediction to improve the accuracy of gene prediction, such as performed by Integrative gene Prediction (IPred)[4] or Maker.[5]

15.2.2 Protein and Peptide Identification

Here we discuss how protein and peptide identification approaches can be applied in the proteogenomics context. Usually peptides are identified by either: (1) using search engines against a protein sequence database, where the MS/MS spectra are compared against theoretical spectra for each peptide in a sequence database (see Chapter 3), or using (2) *de novo* sequencing methods (Chapter 2), where the peptide sequence (or partial sequence) can be extracted from the MS/MS spectra directly without using a protein sequence database, the extracted sequence can be used to search against a sequence database to identify the exact peptide, or by using (3) tag-based identification, where short sequence tags are extracted and used to search against a database, where the peptide list is limited to only ones that contain the extracted sequence tags.

When using the popular sequence database search method, or tag-based method, most approaches first design the search database to allow new events to be discovered, as well as attempting to maintain statistical power in the common identification of peptides matching the "official gene models" – discussed in the following sub-section.

15.2.3 Design of Protein Sequence Databases

The most common approach in proteogenomics is to create a sequence database to be searched, *via* combining and concatenating different input databases. Typically, the first database is the protein set from the official gene models (*i.e.* the current best set from the leading genome database for the species being analysed). Next, additional databases are added *e.g.* new sets of possible gene models from *de novo* running of gene finding software, possible transcripts from *de novo* assembly of RNASeq data (as described in Chapter 16), from a six frame translation of the genome (to give all possible "open reading frames") or "exon graphs". Exon graphs contain peptides that would be derived from all possible combinations of splicing events at a given locus *i.e.* connecting all predicted exons to all other predicted exons, and finding the sequences of all peptides that would overlap such regions.

There are various pros and cons to the selection and inclusion of these different types of databases. First, *ab initio* gene finders have variable quality, depending on the quality of training data available, and their suitability for the species being studied. Most gene finders will produce ranked lists of possible gene structures at each given locus, and thus a wide panel of

possibilities can be included. Such an approach can be advantageous if the gene finder is functioning well (*i.e.* well trained for the species in hand), but if it is poorly trained, this database type can add a lot of noise to the system. Improvements would usually be made if a gene finder is used that can incorporate evidence from other sources, such as other well annotated species.

Second, RNA-Seq data can be turned into a protein sequence database *via* several methods (Chapter 16). This approach would usually be advantageous over the native output of gene finding software, since it should contain only sequences that are actually transcribed. Noise will often be added though, since six frame translations may still have to be performed since the reading frame (and strand) often may not be deduced automatically.

Third, in "six frame translation" approaches, the chromosomal (or contigs if the genome is incompletely assembled) DNA sequences are translated into all possible protein sequences (three possible frames × two possible orientations/strands). The resulting sequences are turned into possible "open reading frames" (ORFs) by finding protein sequences in between each stop codon. These databases can be very large, and so in some cases the database may be filtered by including only protein sequences above a certain size, say 50 amino acids. Six frame translations have two major disadvantages: (i) the databases are generally much larger than other options (reducing statistical power) and (ii) they cannot find evidence for exon-intron structure; any peptides crossing splice junctions cannot be identified. Six frame translations of the genome can be a useful first step in a genome annotation process, if good quality gene annotations have not already been produced, but in annotation pipelines where moderate to good gene annotations exist, and RNA-Seq data are available, they are not commonly used.

Fourth, exon graph approaches create a specialised database intended to find "splice junction" peptides (Figure 15.1), by including the putative peptide sequences that would arise if any given splice event had occurred. The end result is in fact rather similar to using the output of a gene finding software set to produce a large number of ranked hits at a given locus. The advantage of a splice junction approach is that all possible combinations are considered, with only a moderate increase in the overall database size.

The result of the database design stage is typically a sequence database much larger than a database containing only the official protein set – potentially a few times larger, up to a database orders of magnitude larger (if using six frame translations and many different options for potential splice junctions). However, the overall file size of the search database can be misleading in terms of database search size. If the same peptide sequence is included many times over, for example across protein records containing different possible combinations of exons at a given locus, the search space is not actually increased. The search engine only searches against each peptide sequence in the database once, regardless of how many protein records it is contained within. More important in terms of search performance is the total size of the *peptide database* to be search against, and this is the critical measure for a proteogenomics database designer. For example, when producing a six

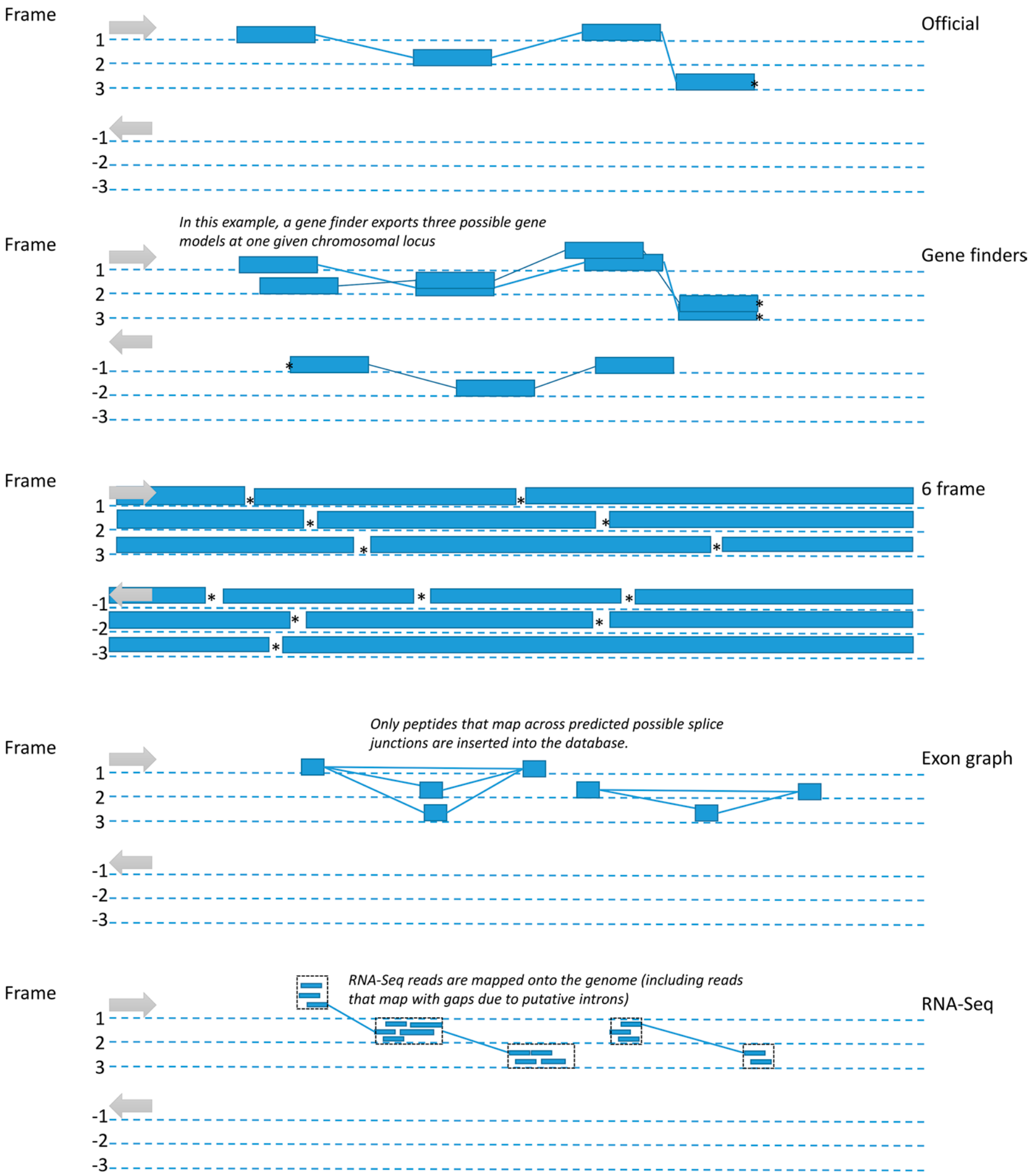

Figure 15.1 A schematic showing the different possible mechanism by which proteogenomics search databases can be assembled, and their relationship to chromosomal positions.

frame translation, the vast majority of the *peptide database* to be searched contains impossible or implausible sequences from a biological point of view. This can have a great impact on sensitivity of identification, compared with searching the same spectra against a database containing only moderate to well annotated sequences.[6]

In proteogenomics, one solution to improve the sensitivity of peptide identification is to use a multistage data analysis. In this strategy, the analysis uses a comparatively small protein sequence database (for example only the current set of official gene models) in stage one, and then another search with a much larger database in stage two.[7] The results from the first stage are used to refine the customised search in the second stage, where only those

spectra for which a confident identification cannot be made progress to being searched against a larger database. The advantage of such an approach is that (so long as reasonably good quality gene models exist), most "regular" peptides can be easily identified with maximum sensitivity in stage 1. In stage 2, a wider search space can be explored to find additional "novel peptides" of various types, depending on the design of the database. Such an approach has been implemented in ProteoAnnotator,[8] which combines multiple search databases generated by gene finding software or derived by assembly from RNA-Seq data, to be compared *versus* the official gene set. In the latest release of ProteoAnnotator, only the official gene models are searched in stage 1, and peptides are identified with strict FDR control. Any spectra for which a confident identification cannot be made progress to stage 2, and a second search is performed against RNA-Seq assemblies and large panels of possible gene models and sequence isoforms resulting from different alleles *etc.*, again controlled by FDR. The results of the two stage searches are combined in such a way to ensure that the resulting FDR is < 1% at the peptide and protein group level. There is still debate in the field as to whether the multi-stage search approach is statistically valid, given that two chances are given to some spectra to find an identification. As such, conservative FDR profiling should be applied on the results to ensure that only loci with strong evidence are further incorporated into the re-annotation processes.

15.2.4 Output of Proteogenomics Pipelines

It is possible to apply a classification to different types of peptides identified in proteogenomics, in terms of the role they can play in genome annotation. In the first category, are peptides that play a confirmatory role, when mapped against the gene model (Figure 15.2). Such peptides include those giving evidence towards the presence of a predicted exon (mapping uniquely to such an exon). Peptides can also give evidence towards the correctly predicted start codon, if a peptide is discovered where the preceding residue is not a site of digestion. In the case of trypsin, this means that the preceding residue is not K or R. However, there are four codons for R and two for K, and as such around 6/64 genuine start codons (assuming no biases in codons in the 5′ UTR), will be preceded by three bases (in the genuine 5′ UTR) that appear to be codons for R or K. Some groups have developed enrichment protocols for N-terminal peptides *e.g.* ref. 9, which can further assist in discovery of true start codons (Figure 15.3).

In a similar way, some peptides can give evidence towards the discovery of the correct C-terminal exon (and stop codon), if they are mapped immediately preceding a stop codon, and the final amino acid of (an assumed tryptic) peptide is not R or K. However, it is also possible (though rarely) for there to be further exons spliced downstream that could explain the same amino acids.

Given a suitable database design, it is also possible to discover evidence for improving gene models, through finding "novel peptides". These can provide evidence for the discovery of completely novel exons, or refinements

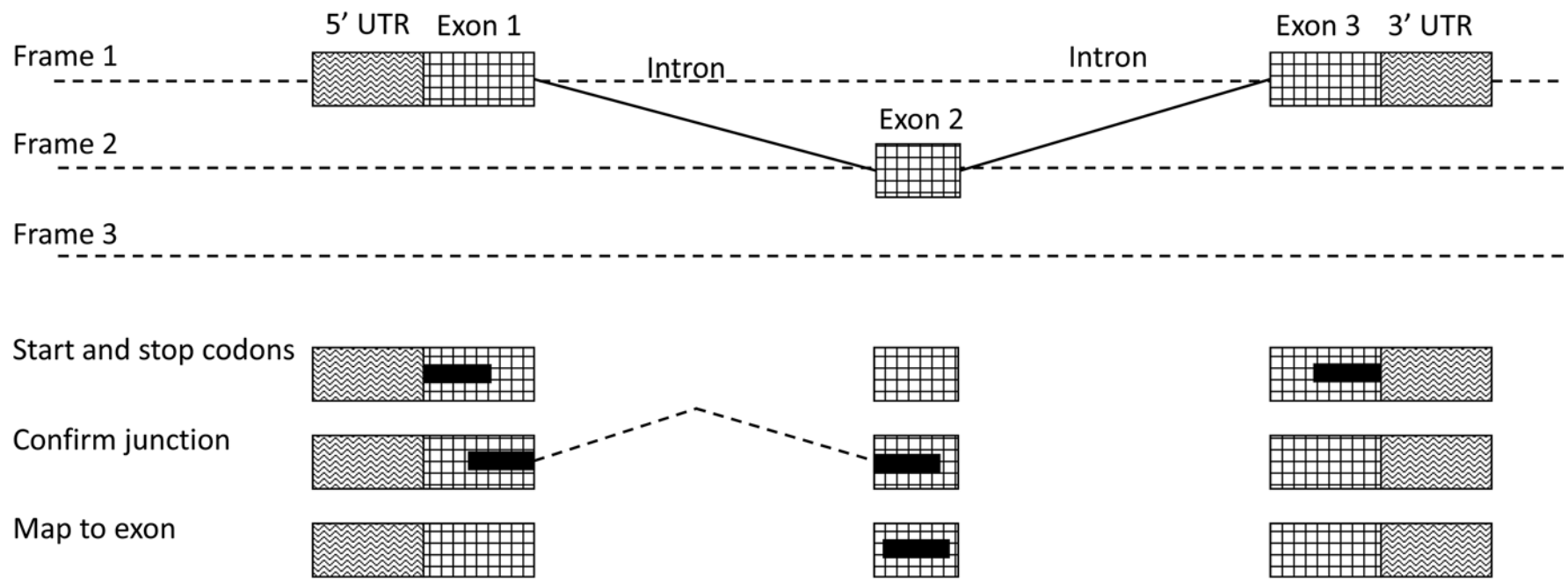

Figure 15.2 Peptides (black filled rectangles) mapped against exons from "official gene models" in large grid pattern, confirming start and stop codons, confirming splice junction or mapping to an exon; 5′ and 3′ UTRs are shown in zig zag pattern.

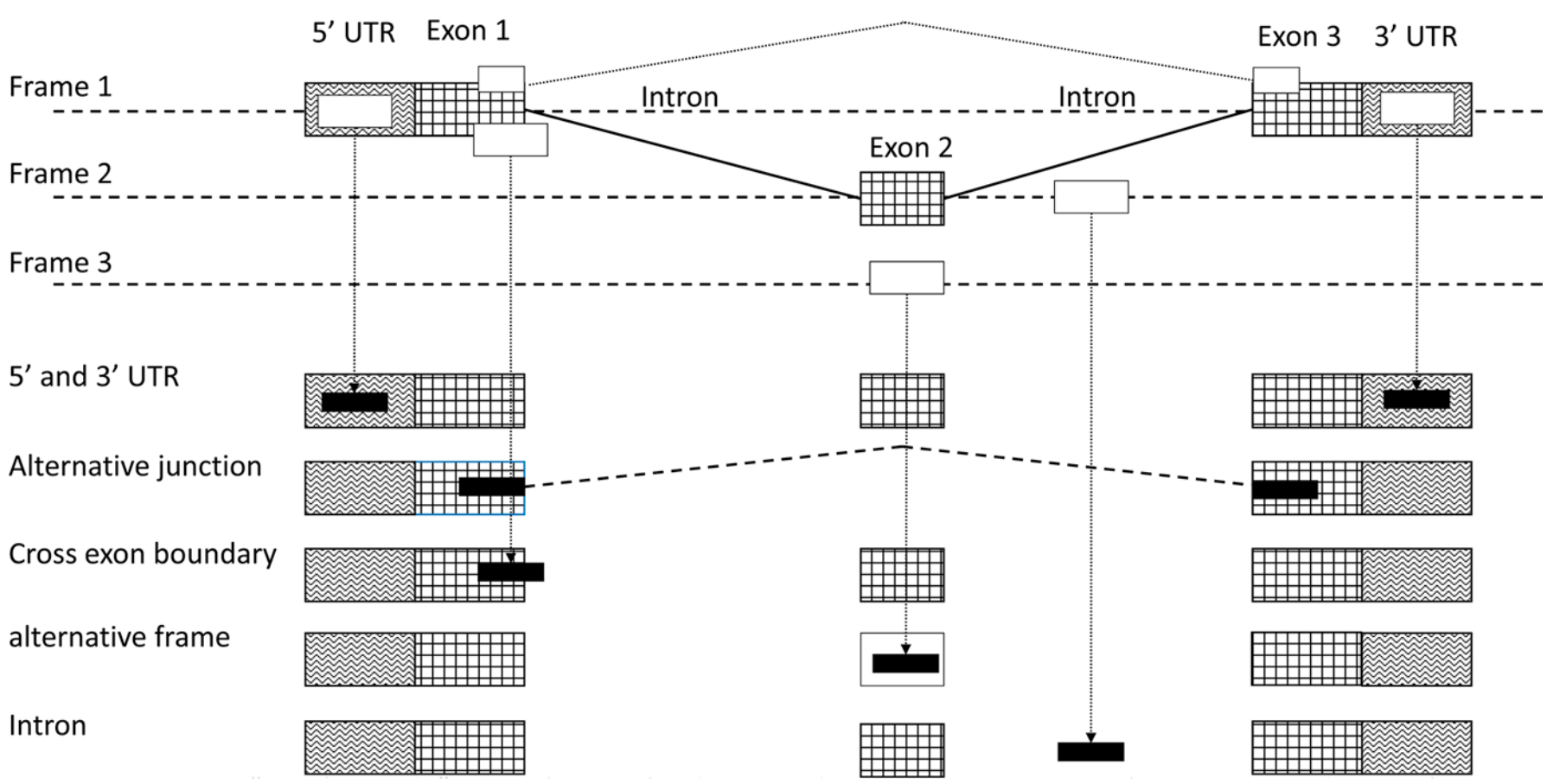

Figure 15.3 Some "novel peptide" types that can be discovered in proteogenomics applications. Peptides are shown in black filled rectangles; "alternative" predicted transcripts or reading frames against which peptides have been matched are shown in white filled rectangles; exons from the "official gene model" are shown in large grid pattern; 5′ and 3′ UTRs are shown in zig zag pattern.

at a given locus. At a given locus, novel peptides can be classified depending on the type of event, including peptides mapping to UTRs, introns, new splice junctions and so on.

15.2.4.1 Statistics and False Discovery Rate Calculation

Chapter 4 covered the topic of peptide-spectrum-matching scoring and validation and explained the statistical methods to rank and validate peptide spectrum matches (PSMs) such as false discovery rate (FDR) and *q*-values

calculations. In this section, we focus on the statistics and the calculation of false discovery rate in proteogenomics context. As discussed in the previous section, the multistage strategies provide one solution for increasing the sensitivity of peptide identifications. To estimate the false discovery rate (FDR) at each stage, an equal number of decoys should be appended to the protein sequence database.

Various groups have observed that the target-decoy approach for estimating FDR is accurate only so long as the decoys represent an appropriate model of what false positives look like within the target database. When querying very large databases containing mainly biologically implausible peptides, as in a six frame translation, biases in FDR estimates may be introduced.[10] More widely, Nesvizhskii has been advocating use of a "class specific" false discovery rate (FDR) calculations where the FDR computed separately for different types of novel peptides.[1] In practice, this means creating a specific decoy database for each type of novel peptide searched for – including splice junction peptides, peptides for confirming amino acid polymorphisms and so on.

A multi-stage search in which a different type of database is searched in each round, goes some way towards ensuring class-specific FDR, so long as FDR is estimated independently on each search. However there is another problem that target-decoy strategies do not accurately capture error rates for novel peptides that are highly homologous to reference peptides, for example they differ by 1–2 amino acids from the reference sequence.

15.3 Proteogenomics Platforms

In this section, we describe software packages available for different aspects of a proteogenomics workflow.

15.3.1 Gene Prediction Pipelines

AUGUSTUS[11] is a software package that can be used to predict genes in genomic sequences. It can predict the 5′UTR, 3′UTR and intron structure of genes. AUGUSTUS also has a protein profile extension that examines membership within protein families to improve predictions of exon–intron structure. Another gene prediction tool is mGene,[12] which uses machine learning and discriminative training techniques, such as support vector machines (SVMs) as well as hidden semi-Markov support vector machines (HSMSVMs). Other tools include *ab initio* gene finders such as Geneid[13] and SCANner (ALTSCAN).[14] A different type of tool is Maker,[5] which can annotate genomes and create genome databases, functioning first as to generate *ab initio* gene predictions, but it can also incorporate RNA-Seq data and proteomes from other species to improve the prediction. Maker is also trainable, allowing outputs from initial runs to be used to train the gene prediction algorithm and thus generate higher quality gene models. Trinity[15] is a method that can be used for full-length transcriptome assembly from RNA-Seq data

without a reference genome. Trinity also reconstructs alternatively spliced isoforms and transcripts from duplicated genes, as discussed in Chapter 16.

15.3.2 Proteogenomics Pipelines

The research community has developed various pipelines for proteogenomics. GAPP[16] was an automated software for the identification of human peptides from tandem mass spectra using the open source X!Tandem search engine to query against particular genome builds from a relational database. Another software tool is Peppy[17] which generates a peptide sequence database from a genome, tracks peptide loci, matches peptides to MS/MS spectra and performs false discovery rate (FDR) analysis. A recent automated proteogenomics pipeline is Integrated Transcriptomic-Proteomic (ITP)[18] which can be used for integrative analysis of transcriptomics and proteomics data. The ITP has two components; the first component uses open-source algorithms for a reference-based transcriptome assembly from RNA-Seq reads. The second component is EuGenoSuite, which is used for proteomic data analysis against this assembled transcriptome.

PPline[19] is a proteogenomics pipeline written in Python, which provides an automated single amino acid polymorphism (SAP) and alternative spliced variants discovery based on raw transcriptome and exome sequence data. Nucleotide EXon-graph Transcriptome Search (NextSearch)[20] is a proteogenomics pipeline, based on a nucleotide exon graph. It consists of building a compact nucleotide exon graph, which includes novel splice variations and a search tool that identifies peptides by directly searching the nucleotide exon graph against tandem mass spectra. Searching for peptide identifications is performed against this nucleotide exon graph, without converting it into a protein sequence in FASTA format, resulting in a reduction in the size of the sequence database storage. The results of NextSearch are stored in a general feature format (GFF) file.

ProteoAnnotator[8] is an automated open-source proteogenomics annotation tool, developed by the authors and colleagues. It is built on the top of the mzIdentML[21] standard and the mzIdentML Library,[22] where a set of routines are used for pre-processing and post-processing, and fully embeds multiple search engines *via* the SearchGUI interface.[23] It can export the results to various file formats such as mzIdentML, GFF3, ProBED and proBAM (see Section 15.3.5 Data formats and Standards), CSV which makes it easier to visualise the results. ProteoAnnotator can be run in two modes, a graphical user interface (GUI) mode or a command line mode. ProteoAnnotator was used in a large-scale proteogenomics study of apicomplexan pathogens.[24]

15.3.3 Proteomics Data Repositories for Proteogenomics

Here we discuss how to use proteomics data that are available publicly from proteomics data repositories. It is possible to reprocess proteomics datasets to improve genome annotation. Mining proteomics data repositories has

been discussed in ref. 25, where four different types of usage are defined: use, reuse, reprocess, and repurpose.

One of the benefits of proteomics data repositories is that they can be used to enhance genome annotations. For example, PeptideAtlas[26] and PRIDE (PRoteomics IDEntifications)[27] databases include large numbers of protein and peptide identifications and post-translational modifications, from a variety of species. These repositories can be mined directly, or serve as sources for data re-processing approaches to support genome annotation. In ref. 28 39 000 exons and 11 000 introns were validated at the level of translation—translation-level evidence was presented for novel or extended exons in 16 genes, 224 hypothetical proteins were confirmed, over 40 alternative splicing events were discovered or confirmed, and improved automated gene prediction by adding 800 correct exons. This study was done by searching 18.5 million tandem mass spectra from human proteomic samples. The data was downloaded from PeptideAtlas data repository, which consisted of spectra from the erythroleukemia K526 cell line, in addition to the data from the HUPO Plasma Proteome Project. In total, 1.8 million spectra in 621 MS runs were searched in this study. In another study[29] they have mapped peptides to over 35% of human proteins, including 150 genes expressed multiple alternative protein isoforms. They used proteomics data available from two large publicly available mass spectrometry repositories, PeptideAtlas with a dataset of 52 019 mzXML spectra files and the Global Proteome Machine (GPM)[30] with a dataset of 5809 mzXML spectra files.

15.3.4 Visualisation

The PG Nexus software[31] allows users to visualise peptides in the context of genomes, which is done in the Integrated Genome Viewer. PG Nexus tool is also integrated into the Galaxy cloud environment.[32] The Samifier tool is used to convert the results from MS/MS searches into a .SAM file format that can be visualised in the Integrative Genomics Viewer. The iPiG tool[33] integrates peptides from MS/MS spectra searches into existing genome browser visualisations. It also supports the mzIdentML standard as an input for the identified peptides. However, the tool does not perform post-processing, it relies on prior FDR estimation methods.

Visual Evaluation and Statistics to Promote Annotation (VESPA)[34] is another visualisation tool for proteogenomics, it is a Java-based desktop application that integrates proteomics data and transcriptomics into genomic context. The data are evaluated by using visual analysis on multiple levels of genomic resolution.

Another software is PGTools[35] which is an open source tool for analysing and visualising of proteogenomics data. It has an interactive HTML report after each run that summarises the main results of the run. PGTools can produce Venn diagrams to display unique and overlapping peptides, an interactive tree-map to show protein groups, and a chromosome distribution plot is used to show proteogenomics peptides based on genome coordinates.

ProBed (see the next section: 15.3.5 Data Formats and Standards) is a file format that can be used to visualise proteomics data at Ensembl.

15.3.5 Data Formats and Standards

Chapter 11 explained the topic of data formats and standards and covered the Proteomics Standards Initiative standards. Here we discuss how genome coordinates can be captured in the mzIdentML data format. Furthermore we discuss the developing adaptations to Browser Extensible Data (BED) and binary SAM (BAM) formats, as well as possible methods for annotating peptide data into General Feature Format (GFF).

The mzIdentML standard can capture PSMs and identified proteins (and groups of proteins with shared evidence), as well as the listing database peptides and database proteins. In mzIdentML 1.1 (stable version, see Chapter 11), from a sequence database search, each PSM is linked to all database proteins (*DBSequence* element in mzIdentML file) in which it can be located, *via* a mapping element called *PeptideEvidence*. *PeptideEvidence* contains the information on the position (start and end position) of the peptide within the parent protein. If the genome coordinates of the gene encoding the protein are known, the values in the start and end attributes can be used to map the peptide onto the genome. However, for some peptide types, such as those mapping across splice junctions, the start and end coordinates are not sufficient for genome visualisation and further analysis. Instead, data are needed demonstrating which parts of the peptide mapped to which exons. In mzIdentML 1.1 this can be achieved by adding extra parameters (*userParam* elements) with the exact chromosomal coordinates of each block of exon from which a peptide was derived. Such a formal encoding is planned to be added into the next major release version of mzIdentML (1.2).

A new framework, proBAMsuite,[36] was recently developed in which a new file format protein BAM (ProBAM) was created for organising peptide spectrum matches in the context of the genome. ProBAM is based on Sequence Alignment Map (SAM) and binary SAM (BAM) formats which are designed for encoding alignment information of sequencing reads to a genome. BAM has been extended for proteomics in ProBAM to include specific data such as PSM score, charge state and peptide level modifications. PSMs can be simply re-annotated using gene annotation schemes and assembled into both protein and gene identifications, thus providing the data integration between proteomics and proteogenomics. Also it is possible to visualise proBAM files in genome browsers, which has the advantage of bringing proteomics data analysis to the genomics community.

ProBed is another file format that is currently being developed which also works on integrating peptide and protein identifications from MS-based experiments within the genome. ProBed is built on top of BED file format that is used to describe genome coordinates. The BED file format contain 12 fields which have been extended with a further 11 fields in ProBed to include information about peptide spectrum matches, such as protein accession,

peptide sequence, PSM score, FDR, peptide modification, charge and PSM rank. Both proBAM and ProBED will be developed under The Human Proteome Organisation (HUPO) Proteomics Standards Initiative (PSI) umbrella for wider use.

15.4 Challenges and Future Research

While the field of proteogenomics made progress in the recent years, many challenges still need to be addressed and solved. There are technical and practical challenges in proteogenomics, due to the lack of compatibility and interoperability between different file formats used in different software tools. This brings extra work for scientists to convert the file format between different output files and to find software packages that fully integrate the range of different genomics and proteomics data types that exist.

Another challenge is calculating the false discover rate (FDR) in proteogenomics. While the procedure to calculate the false discovery rate in proteomics is straightforward, it is challenging in proteogenomics, especially when running a multi-stage search. Statistical research is still on-going to understand the behaviour of target-decoy methods in the proteogenomics context, and/or to develop alternative approaches that can calculate accurate statistics for novel peptides – giving evidence towards improvements to gene annotations.

MS-based proteomics can play an important role for post-genomic investigations. However as both proteomics and genomics datasets continue to evolve independently, dynamic software for integration is needed. Such efficient and up-to-date integration between proteomics and genomics information in public resources has only occurred to a limited extent so far. A solution to this challenge would be automatically re-mapping data or re-running searches as gene annotations change allowing dynamic linkages between proteomics and genomics.

15.5 Summary

MS-based proteomics methods allow the identification and characterisation of proteins, peptides, and post-translational protein modifications (PTMs), providing information about protein expression and functional states. These findings are not directly accessible with genomic sequencing methods. In proteogenomics, proteomic observations of specific peptides contribute to the definition of correct gene structures, alternative splicing, and discovery of new/support for previously weakly supported gene annotations. An effective integration of proteomics and genomics data is very challenging since there is usually a disconnect between the results of a proteomics analysis and the most recent version of the genome of the same organism. This occurs since search engines use a particular version of a protein sequence database, derived from a concrete genome build. Many research groups do not regularly update their local protein sequence databases and results are

only fully comparable when the same version of the same protein database is used. A dynamic integration between the current genome annotation and proteomics tools is important to ensure that results generated in the past, can still be interpreted as genomes, annotations and external software (such as pathway/network mapping) continue to evolve. Additionally, proteomics data can be improved from regular re-analysis in the context of updated genome and protein sequence databases, potentially increasing the numbers of proteins identified/quantified, and overall data quality as gene models improve.

References

1. A. I. Nesvizhskii, Proteogenomics: concepts, applications and computational strategies, *Nat. Methods*, 2014, **11**, 1114–1125.
2. S. J. Goodswen, P. J. Kennedy and J. T. Ellis, Evaluating high-throughput ab initio gene finders to discover proteins encoded in eukaryotic pathogen genomes missed by laboratory techniques, *PLoS ONE*, 2012, **7**, e50609.
3. R. Lindner and C. C. Friedel, A Comprehensive Evaluation of Alignment Algorithms in the Context of RNA-Seq, *PLoS One*, 2012, **7**, e52403.
4. F. Zickmann and B. Y. Renard, IPred-integrating ab initio and evidence based gene predictions to improve prediction accuracy, *BMC Genomics*, 2015, **16**, 134.
5. B. L. Cantarel, I. Korf, S. M. Robb, G. Parra, E. Ross, B. Moore, C. Holt, A. Sanchez Alvarado and M. Yandell, MAKER: an easy-to-use annotation pipeline designed for emerging model organism genomes, *Genome Res.*, 2008, **18**, 188–196.
6. A. I. Nesvizhskii, A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics, *J. Proteomics*, 2010, **73**, 2092–2123.
7. K. Ning, D. Fermin and A. I. Nesvizhskii, Computational analysis of unassigned high-quality MS/MS spectra in proteomic data sets, *Proteomics*, 2010, **10**, 2712–2718.
8. F. Ghali, R. Krishna, S. Perkins, A. Collins, D. Xia, J. Wastling and A. R. Jones, ProteoAnnotator–Open source proteogenomics annotation software supporting PSI standards, *Proteomics*, 2014, **14**, 2731–2741.
9. L. McDonald, D. H. L. Robertson, J. L. Hurst and R. J. Beynon, Positional proteomics: selective recovery and analysis of N-terminal proteolytic peptides, *Nat. Methods*, 2005, **2**, 955–957.
10. P. Blakeley, I. M. Overton and S. J. Hubbard, Addressing statistical biases in nucleotide-derived protein databases for proteogenomic search strategies, *J. Proteome Res.*, 2012, **11**, 5221–5234.
11. M. Stanke and S. Waack, Gene prediction with a hidden Markov model and a new intron submodel, *Bioinformatics*, 2003, **19**, ii215–ii225.
12. G. Schweikert, J. Behr, A. Zien, G. Zeller, C. S. Ong, S. Sonnenburg and G. Rätsch, mGene.web: a web service for accurate computational gene finding, *Nucleic Acids Res.*, 2009, **37**, W312–W316.

13. E. Blanco, G. Parra, and R. Guigó. Using geneid to identify genes. *Curr. Protoc. Bioinformatics*, 2007, 4.3.1–4.3.28.
14. Z. Hu, H. S. Scott, G. Qin, G. Zheng, X. Chu, L. Xie, D. L. Adelson, B. E. Oftedal, P. Venugopal and M. Babic, Revealing Missing Human Protein Isoforms Based on Ab Initio Prediction, RNA-seq and Proteomics, *Sci. Rep.*, 2015, 5.
15. M. G. Grabherr, B. J. Haas, M. Yassour, J. Z. Levin, D. A. Thompson, I. Amit, X. Adiconis, L. Fan, R. Raychowdhury, Q. Zeng, Z. Chen, E. Mauceli, N. Hacohen, A. Gnirke, N. Rhind, F. di Palma, B. W. Birren, C. Nusbaum, K. Lindblad-Toh, N. Friedman and A. Regev, Full-length transcriptome assembly from RNA-Seq data without a reference genome, *Nat. Biotechnol.*, 2011, **29**, 644–652.
16. I. Shadforth, W. Xu, D. Crowther and C. Bessant, GAPP: A Fully Automated Software for the Confident Identification of Human Peptides from Tandem Mass Spectra, *J. Proteome Res.*, 2006, **5**, 2849–2852.
17. B. A. Risk, W. J. Spitzer and M. C. Giddings, Peppy: proteogenomic search software, *J. Proteome Res.*, 2013, **12**, 3019–3025.
18. D. Kumar, A. K. Yadav, X. Jia, J. Mulvenna and D. Dash, Integrated transcriptomic-proteomic analysis using a proteogenomic workflow refines rat genome annotation, *Mol. Cell. Proteomics*, 2016, **15**, 329–339.
19. G. S. Krasnov, A. A. Dmitriev, A. V. Kudryavtseva, A. V. Shargunov, D. S. Karpov, L. A. Uroshlev, N. V. Melnikova, V. M. Blinov, E. V. Poverennaya and A. I. Archakov, PPLine: An Automated Pipeline for SNP, SAP, and Splice Variant Detection in the Context of Proteogenomics, *J. Proteome Res.*, 2015, **14**, 3729–3737.
20. H. Kim, H. Park and E. Paek, NextSearch: A search engine for mass spectrometry data against a compact nucleotide exon graph, *J. Proteome Res.*, 2015, **14**, 2784–2791.
21. A. R. Jones, M. Eisenacher, G. Mayer, O. Kohlbacher, J. Siepen, S. Hubbard, J. Selley, B. Searle, J. Shofstahl, S. Seymour, R. Julian, P.-A. Binz, E. W. Deutsch, H. Hermjakob, F. Reisinger, J. Griss, J. A. Vizcaino, M. Chambers, A. Pizarro and D. Creasy, The mzIdentML data standard for mass spectrometry-based proteomics results, *Mol. Cell. Proteomics*, 2012, **11**, M111.014381.
22. F. Ghali, R. Krishna, P. Lukasse, S. Martínez-Bartolomé, F. Reisinger, H. Hermjakob, J. A. Vizcaíno and A. R. Jones, Tools (Viewer, Library and Validator) that Facilitate Use of the Peptide and Protein Identification Standard Format, Termed mzIdentML, *Mol. Cell. Proteomics*, 2013, **12**, 3026–3035.
23. M. Vaudel, H. Barsnes, F. S. Berven, A. Sickmann and L. Martens, SearchGUI: An open-source graphical user interface for simultaneous OMSSA and X!Tandem searches, *Proteomics*, 2011, **11**, 996–999.
24. R. Krishna, D. Xia, S. Sanderson, A. Shanmugasundram, S. Vermont, A. Bernal, G. Daniel-Naguib, F. Ghali, B. P. Brunk and D. S. Roos, A large-scale proteogenomics study of apicomplexan pathogens—Toxoplasma gondii and Neospora caninum, *Proteomics*, 2015, **15**, 2618–2628.

25. M. Vaudel, K. Verheggen, A. Csordas, H. Ræder, F. S. Berven, L. Martens, J. A. Vizcaíno and H. Barsnes, Exploring the potential of public proteomics data, *Proteomics*, 2016, **16**, 214–225.
26. F. Desiere, E. W. Deutsch, N. L. King, A. I. Nesvizhskii, P. Mallick, J. Eng, S. Chen, J. Eddes, S. N. Loevenich and R. Aebersold, The PeptideAtlas project. Nucl, *Acids Res.*, 2006, **34**, D655–D658.
27. L. Martens, H. Hermjakob, P. Jones, M. Adamski, C. Taylor, D. States, K. Gevaert, J. Vandekerckhove and R. Apweiler, PRIDE: The proteomics identifications database, *Proteomics*, 2005, **5**, 3537–3545.
28. S. Tanner, Z. Shen, J. Ng, L. Florea, R. Guigo, S. P. Briggs and V. Bafna, Improving gene annotation using peptide mass spectrometry, *Genome Res.*, 2007, **17**, 231–239.
29. I. Ezkurdia, A. del Pozo, A. Frankish, J. M. Rodriguez, J. Harrow, K. Ashman, A. Valencia and M. L. Tress, Comparative proteomics reveals a significant bias toward alternative protein isoforms with conserved structure and function, *Mol. Biol. Evol.*, 2012, mss100.
30. R. Craig, J. P. Cortens and R. C. Beavis, Open Source System for Analyzing, Validating, and Storing Protein Identification Data, *J. Proteome Res.*, 2004, **3**, 1234–1242.
31. C. N. I. Pang, A. P. Tay, C. Aya, N. A. Twine, L. Harkness, G. Hart-Smith, S. Z. Chia, Z. Chen, N. P. Deshpande, N. O. Kaakoush, H. M. Mitchell, M. Kassem and M. R. Wilkins, Tools to Covisualize and Coanalyze Proteomic Data with Genomes and Transcriptomes: Validation of Genes and Alternative mRNA Splicing, *J. Proteome Res.*, 2013, **13**, 84–98.
32. J. Goecks, A. Nekrutenko, J. Taylor and T. The Galaxy, Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences, *Genome Biol.*, 2010, **11**, R86.
33. M. Kuhring and B. Y. Renard, iPiG: integrating peptide spectrum matches into genome browser visualizations, *PLoS One*, 2012, **7**, e50246.
34. E. S. Peterson, L. A. McCue, A. C. Schrimpe-Rutledge, J. L. Jensen, H. Walker, M. A. Kobold, S. R. Webb, S. H. Payne, C. Ansong and J. N. Adkins, VESPA: software to facilitate genomic annotation of prokaryotic organisms through integration of proteomic and transcriptomic data, *BMC Genomics*, 2012, **13**, 1.
35. S. H. Nagaraj, N. Waddell, A. K. Madugundu, S. Wood, A. Jones, R. A. Mandyam, K. Nones, J. V. Pearson and S. M. Grimmond, PGTools: a software suite For proteogenomic data analysis and visualization, *J. Proteome Res.*, 2015, **14**, 2255–2266.
36. X. Wang, R. J. Slebos, M. C. Chambers, D. L. Tabb, D. C. Liebler and B. Zhang, proBAMsuite, a bioinformatics framework for genome-based representation and analysis of proteomics data, *Mol. Cell. Proteomics*, 2015, M115.052860.

CHAPTER 16

Proteomics Informed by Transcriptomics

SHYAMASREE SAHA[a], DAVID MATTHEWS[b] AND CONRAD BESSANT[a]*

[a]School of Biological and Chemical Sciences, Queen Mary, University of London, E1 4NS, UK; [b]School of Cellular and Molecular Medicine, University of Bristol, University Walk, Bristol, BS8 1TD, UK
*E-mail: c.bessant@qmul.ac.uk

16.1 Introduction to PIT

Searching MS/MS spectra against a database of proteins that could be present in the sample, using the methods described in Chapter 3, remains the pre-eminent method for identifying proteins in liquid chromatography tandem mass spectrometry (LC-MS/MS) shotgun proteomics. The relevance and quality of this protein database clearly has a significant impact on the outcome of a proteomics study, because only proteins present in the database can be detected and increasing the database size reduces the number of significant protein identifications.[1] For well-studied species such as human it is tempting to assume suitable protein databases are available, and indeed a high quality complete proteome can be downloaded from UniProt[2] for human, and many other model organisms. However, even for these well-studied model organisms the true set of proteins that might be present is debatable and the proteome is not definitive. For example, large

New Developments in Mass Spectrometry No. 5
Proteome Informatics
Edited by Conrad Bessant

Published by the Royal Society of Chemistry, www.rsc.org

scale human proteome mapping projects have recently suggested that commonly used human protein databases include protein sequences that may never be expressed while simultaneously omitting proteins for which there is mass spectrometry evidence.[3,4] These studies have also confirmed that the expression of many proteins is tissue specific, suggesting that filtering a proteome prior to performing a database search could be a logical way of reducing search space if dealing with a sample from a specific tissue. However, this relies totally on the annotations of the proteins being correct. Alternative splicing[5] further complicates the situation. Protein isoforms derived from genome annotation are included in some protein databases, but the existence of every isoform may not have been confirmed by experimental evidence and including all possible isoforms increases search space, making peptide spectrum matching and protein grouping more difficult and resulting in fewer significant protein identifications.

For lesser studied organisms the situation is worse still, as available protein databases rely more on computational gene prediction, and in some species even this is not possible as a reference genome has not yet been assembled. This is a significant problem in fields such as virology, where some important disease vectors (*e.g.* lice, ticks, birds and bats) have poorly annotated genomes or no genome data at all. Similar challenges can be found in metaproteomics,[6] where individual samples contain proteins from multiple organisms, some of which may be unidentified or not previously sequenced.

To overcome these problems, a relatively new methodology called proteomics informed by transcriptomics (PIT)[7] has been developed, in which a sample-specific protein database is generated from transcripts that have been identified in the sample using RNA-seq.[8] The fundamental concept underpinning this approach (shown schematically in Figure 16.1) is that proteins are translated from mRNA, hence the majority of proteins present in a sample should be represented in the same sample by their corresponding transcripts. RNA-seq and LC-MS/MS proteomics are performed on extracts of the same sample. The transcripts can be assembled from raw RNA-seq data either by mapping to a genome or entirely *de novo* (*e.g.* if no suitable genome exists). These are then translated into a list of amino acid sequences by finding open reading frames (ORFs). This list is then used as a protein database for a standard database search, resulting in a typical list of identified polypeptides together with their scores and corresponding peptide evidence. At this point the identified polypeptides are characterised only by their amino acid sequences, lacking names or other annotation so further processing is needed to add biological meaning to these results—exactly how this is done depends on the experiment being undertaken and is covered later in the chapter.

Combining protein identifications derived from peptide mass spectra with transcriptomic data to annotate genomes has already been mentioned in the previous chapter, in the context of proteogenomics. Indeed, PIT can be used for genome annotation and this aspect of its functionality can certainly be considered within the realm of proteogenomics. However, PIT

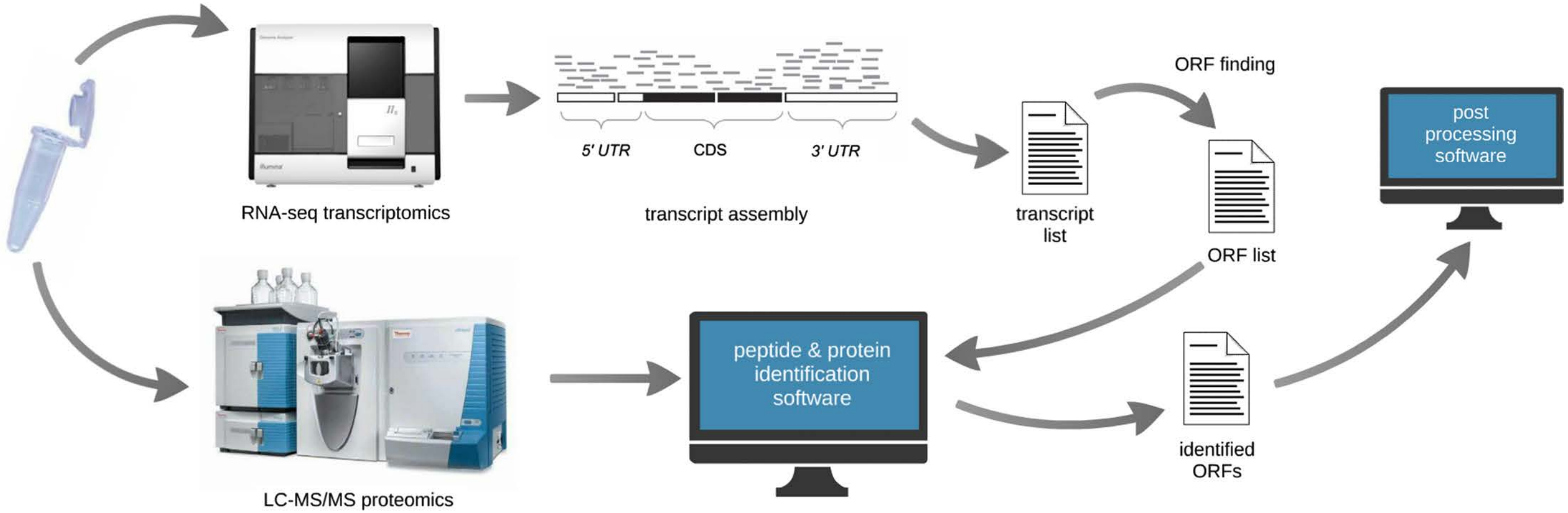

Figure 16.1 Schematic of PIT workflow. LC-MS/MS proteomics and RNA-seq transcriptomics analyses are carried out on the same sample. Transcripts are assembled from the RNA-seq data, and used to generate open reading frames (ORFs). There, ORFs are then used to create a protein database for traditional peptide spectrum matching. This results in a list of ORFs that have been identified as being present in the sample. Further application-specific downstream processing is needed to extract useful information from this list of ORFs.

goes beyond this, allowing proteomic analysis in the absence of a reference genome, as well as sensitive detection of sequence variation, and monitoring of dynamic processes such as isoform switching. The key facilitator of these additional capabilities is that, in PIT, RNA-seq data are collected from the same sample as the MS data—this is not necessarily the case in proteogenomics.

Although PIT can conceptually provide valuable new insights into the systems being studied, there are a number of technical and practical difficulties in integrating the RNA-seq and proteomic MS/MS data. As is evident from previous chapters, extracting protein identifications and abundances from MS data is a complex multi-step process which is time consuming because datasets are large. Analysis of RNA-seq transcriptomic data holds its own challenges, and data volumes are typically an order of magnitude greater than in proteomics. Combining data from proteomics and transcriptomics together is clearly more complicated than dealing with either one, and knowledge of tools and best practice from the two distinct communities is required as well as new approaches. The bulk of this chapter explains how each step in the PIT workflow can be tackled, starting with RNA-seq.

16.2 Creation of Protein Database from RNA-Seq Data

16.2.1 Introduction to RNA-Seq

RNA sequencing (RNA-seq) has revolutionised the study of transcriptomes, largely taking over from DNA microarrays in this field. RNA-seq uses next generation sequencing (NGS) to reveal the presence of transcripts and their quantity in a sample at a given time. RNA-seq is commonly used to investigate differential gene expression, infer gene interaction networks, monitor expression as a function of time and study biological events such as alternative gene splicing, gene fusion and post-transcriptional modification.[8] Unlike microarray technology, where specific probes are used to monitor specific genes, RNA-seq is an open technique, capable of sequencing almost any mRNA in the sample, thereby allowing the discovery of new transcripts. Importantly, RNA-seq (followed by appropriate sequence assembly) is able to generate full length transcripts, in contrast to shotgun proteomics where gaps in sequence are common due to undetectable peptides.

A detailed explanation of NGS is beyond the scope of this chapter, but a brief overview is needed to appreciate its capabilities and limitations, and the impact these have on PIT. Various NGS technologies are available but we focus here on Illumina sequencing as that has been most extensively applied in PIT experiments to date. The sequencing process starts with the preparation of a *library*, a collection of DNA from the sample (In the case of RNA-seq, the DNA is reverse transcribed from RNA.). DNA is cleaved into smaller fragments by sonication or enzyme digestion. Small

DNA sequences called adaptors are ligated to the ends of the fragments, to help with the amplification and sequencing of the fragments. The ligated fragments are size fractioned (often 200 to 300 bp is selected) and used as a template in a polymerase chain reaction (PCR). Illumina sequencing uses bridge PCR[9] to amplify the library by washing these ligated fragments across flow-cell channels covered in primers. The flow-cell is where the sequencing takes place. Before attachment to the flow cell, the ligated fragments are denatured, producing single-stranded copies of the fragments for sequencing. The adaptor constructs have two flow-cell binding sites, P5 and P7. The P5 and P7 binding sites of the single stranded fragments anneal to the complementary primers on the flow cell. The unattached flow cell oligonucleotides act as a primer and a strand complementary to the library fragment is synthesised by adding unlabelled nucleotides. The original fragments are washed away leaving the fragment copy bonded to the flow cell primers. At this stage the fragment makes a bridge shape before it is copied, hence the name bridge PCR. Thousands of copies of each fragment are generated and these create clusters of complementary fragments of the original sequence of interest. The flow-cell is then flooded with nucleotides and they get added to the fragments one base at a time at each cycle. The addition of each nucleotide releases a light or fluorescence.[10] These changes are detected and software used to determine the base added using a process commonly known as base calling.[11] One or both sides of the fragment might be sequenced, producing single or paired-end reads. Paired-end reads are the sequences of two ends of the same DNA molecule with a physical distance between two reads. One end of the molecule is sequenced and it is turned around to sequence the other end in paired-end sequencing.

As with most analytical techniques, NGS does not produce 100% error-free data. Sequencing errors can be introduced during sample handling and in the data acquisition process. Sample handling errors may happen at the sample preparation or at the amplification (PCR) step, and mutation artefacts have been reported due to oxidative DNA damage during sample preparation.[12] PCR should produce an exact copy of the library fragment and fragment number should double after each cycle, but in practice some artificial molecules are produced besides the original fragment molecule. Mutations or unwanted reactions between the template molecules result in these artificial molecules. In addition, amplification contracts during PCR due to reduction of repeat units.[13] Although, the amplification rate does not decrease equally for the entire library creating read coverage gap. Another explanation for the coverage variation is the formation of secondary structures (hairpins) in single-stranded DNA (ssDNA).[14] AT-rich repetitive sequence is also reported to have lower sequence coverage.[15] Besides these error causing factors, there are other biases due to imperfect chemistry, sensors and imaging technology such as phasing (where a strand fails to incorporate a nucleotide in a cycle) or pre-phasing (multiple bases getting incorporated in one cycle) causing erroneous fluorescence emission.

Fluorophore cross talk can also result in misinterpretation of signal. Additionally, strong correlation of A and C as well as G and T intensities because of similar emission spectra of the fluorescence introduce data acquisition errors.

Error rates of Illumina sequencing machines are reported to be of the order of $0.1–1 \times 10^{-2}$, varying according to sequence.[16] The accuracy of the sequence can be improved by increasing read coverage, essentially sequencing the same DNA multiple times, increasing the sequencing cost.[11] Sequencing errors and read coverage variation can be misinterpreted as polymorphisms, mutations, or copy number variations. When using RNA-seq data to generate a sequence database for peptide spectrum matching there is a clear danger that some of the RNA-level errors will translate into peptide-level errors, preventing a peptide from being identified. On the other hand, it is possible to envisage a strategy where proteomic data could be used to identify potential errors in the transcriptomic data.

Sequencing platforms typically produce sequencing data in a file format called FASTQ, which is often referred to as raw data in the sequencing community. The sequencer may produce tens of billions of bases, hence the data are often compressed. FASTQ is a text-based file format containing both the sequence of the reads and the sequencing quality. Figure 16.2 shows an example of the format. Four lines represent each read. The first line starts with @, usually followed by the title or the identifier of the sequence. The second line is the actual read sequence. The third line, beginning with +, is an optional line often containing the sequence title or identifier. The final line is the PHRED quality score of each base of the sequence encoded as ASCII printable characters (ASCII 33-126). PHRED score Q is a measure of the quality of the identification of bases during base calling which is logarithmically related to the sequencing error probability P, and calculated as follows:

$$Q = -10\log_{10}P$$

Sequencers commonly store paired-end sequences in two FASTQ files, one for the left side bases and the other for the right side bases. Depending on the sample preparation protocol, RNA sequencing is either stranded or un-stranded. Some of the RNA-seq assemblers or aligners require the library type information. There are four library types for stranded sequencing, as shown in Figure 16.3.

@Title/ID	@HISEQ2000-06:362:C3P0FACXX:3:1101:1571:2240 1:N:0:GCCTTCCT
Sequence	CTTTCTCTTCTTTAGGAATTTCTGTGACTTCTGCTGTAATTAATGGTGAAGCCACTCAAGCAGCATCTAATAAAGCAGTTCTCACATCCTTCGTTGGGAC
+Optional text	+
quality	1:?:=BAD?=DHB?FHIAAG?I>CH:,333+2<42211@?E:*111***?0000)*)9D88/)(=..B8=;=7)/7.=..7)==;=)=?;;36.6.('(5

Figure 16.2 FASTQ is a text file format to represent sequencing reads. Each read is represented as four lines, where first line is the ID of the sequence, second is the sequence. Third line is optional, beginning with +. The last line is the base quality of the read sequence.

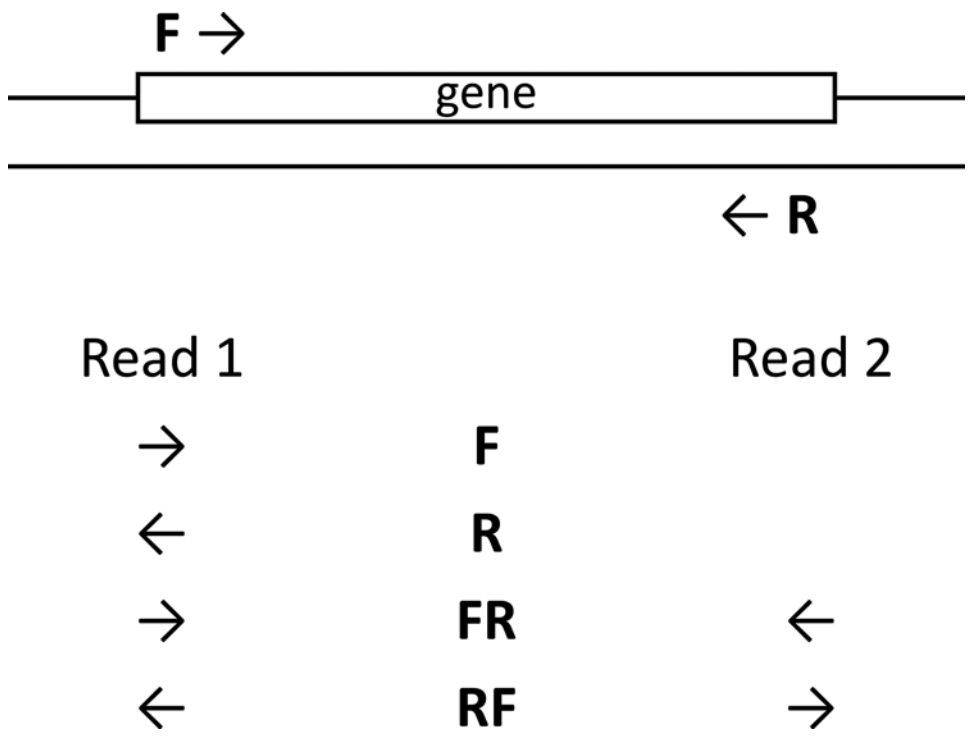

Figure 16.3 Orientation of single and paired-end reads. F and R represent the sense (forward) and antisense (reverse) orientation of single end respectively. Whereas FR means that, the first read of fragment pair is sequenced as sense and the second read as antisense. RF means the opposite of FR, *i.e.* the first read is anti-sense and the second as sense.

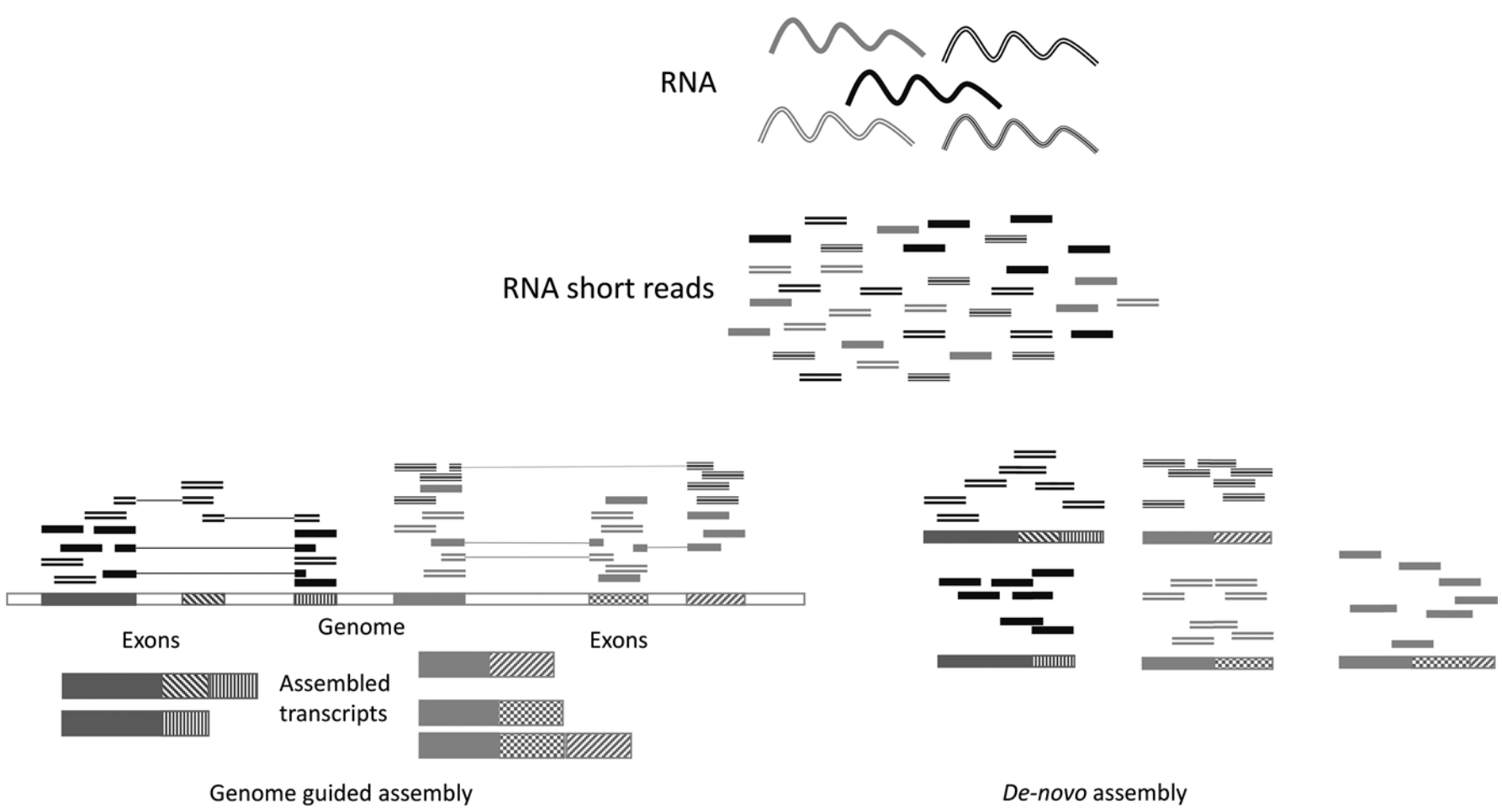

Figure 16.4 There are two ways of assembling RNA-seq data, (i) genome guided and (ii) *de novo* assembly. Short RNA-seq reads are mapped to a genome and mapped reads are assembled based on their genomic location. *De novo* assembly does not require a reference genome to assemble the short reads into a transcript.

16.2.2 Sequence Assembly

RNA-seq data analysis starts by assembling these reads into full-length transcripts. Two different methods, genome guided and *de novo*, are available for this purpose (Figure 16.4). The genome guided methods align the short raw reads to a reference genome and an assembler assembles the reads in order to reconstruct the original sequence. Bowtie,[17] Tophat,[18] BWA[19] are

examples among many specialised RNA-seq aligners. Genome-guided assemblers such as Cufflinks and Scripture build a transcript from the aligned reads. On the other hand, *de novo* transcriptome assemblers do not rely on a reference genome to reconstruct the transcript sequence. Trinity,[20] Velvet,[21] Oases[22] and *Trans*-ABySS[23] are commonly used *de novo* assemblers. It is the ability to assemble transcripts from short reads *de novo* that makes it possible to use PIT to perform proteomics on non-model organisms. The genome guided and the *de novo* assembled transcripts can be merged based on consensus using Program to Assemble Spliced Alignments (PASA)[24] to achieve better transcript coverage.

Though the instrument vendors do filter bad quality reads and correct errors at the base calling step, reads still tend to have low quality bases at the beginning and towards the end. Low PHRED scores suggest low reliability, and low quality reads will assemble into a low quality transcript. Erroneous bases can make their way through the assembly process, resulting in erroneous transcripts that can have a significant impact on the PIT analysis. Aside from obvious impact of non-synonymous errors on the amino acid sequence, an erroneous nucleotide call may result in early termination of an ORF, or a missing stop codon may extend the coding region. Additionally, low read coverage may result in a gap, *i.e.* an apparent deletion, which will cause incomplete and wrong transcript assembly leading to missed or erroneous protein identifications. Hence, prior to assembly, it is good practice to perform quality control to clean or trim reads using tools such as FASTQC,[25] FASTX (http://hannonlab.cshl.edu/fastx_toolkit/), Picard (http://broadinstitute.github.io/picard), HTSeq-QA,[26] NGS QC Toolkit.[27]

Following assembly, the raw reads are mapped back to the assembled transcripts to calculate the read coverage. Often transcripts with low read coverage are filtered out due to insufficient read evidence. Fragments per kilobase of transcript per million mapped reads (FPKM) and reads per kilobase of transcript per million mapped reads (RPKM) are often used as a proxy for gene expression level, isoform abundance and transcript assembly reliability. In PIT experiments, these can be compared with the expression levels of the associated proteins if a quantitative proteomics protocol has been employed.

Short read alignment software outputs the mapping results in Sequence Alignment/Map format (SAM) or binary SAM (BAM) format to reduce the size of the alignment file. SAM is a tab delimited text file with eleven mandatory fields.[28] Each line represents an alignment, with an optional header section prior to the alignment section in which every line starts with @. The assembled short reads reconstruct the transcripts, which can be described in a standard FASTA file.

16.2.3 ORF Finding

By this point, we have a FASTA file of RNA transcripts from the sample but for peptide spectrum matching we need to generate a FASTA file of corresponding amino acid sequences. This is achieved by predicting open

reading frames (ORFs) from the assembled transcripts, a task for which several software tools are available. All of these tools allow six frame translation. The getORF tool from EMBOSS[29] is one such tool where ORFs are defined by a region between two stop codons or between a start and a stop codon with a minimum length. Start and stop codons can be selected from a genetic code table suitable for the species. Transdecoder[30] is another option for ORF prediction which predicts coding regions within a transcript sequence. This tool predicts ORFs based on the minimum length of ORFs, computes log likelihood score in a similar way to the GeneID[31] software and ORFs from the first frame shift are given a higher score than the other five frame shifts. It outputs the gene structure in GFF3[32] and BED[33] file formats. All of these ORF prediction tools output the predicted ORFs in FASTA format, providing a rudimentary search database for peptide spectrum matching.

16.2.4 Finalising Protein Sequence Data for PIT Search

A core assumption of the PIT approach is that proteins present in a sample are accompanied by the RNA transcripts from which they were produced. However, one can envisage many proteins for which this is not the case. Firstly, the sample may contain contaminant proteins that are not endogenous to the sample under study. To account for these proteins the search database can be augmented with a list of common contaminants, such as the common Repository of Adventitious Proteins from the GPM.[34] Proteins with long half-lives constitute the second class of proteins for which a corresponding transcript may not be present, as the RNA precursors of such proteins may have degraded prior to the sample being analysed. To cover this eventuality, the RNA-seq database can be merged with a canonical proteome, although the optimal way to do this is an open question. A variation of this approach is to perform an initial search against the standard proteome and then use the database of ORFs in an attempt to identify unassigned spectra in a secondary search.[35] This approach carries a risk of wrongly identifying SAPs as PTMs. Researchers have used standard proteomes merged with amino acid sequences representing disease related SNPs and splice junction peptides to identify disease specific mutations and isoforms.[36,37] Finally, the transcript read coverage can be used to filter out ORFs based on low-quality transcripts to reduce erroneous peptide identification.

16.3 Interpretation of Identified ORFs

While the peptide and protein identification step is achieved using a standard database search, the results are not immediately useful as they consist only of a list of ORFs for which peptide evidence has been found. Extensive data analysis is needed to extract biologically useful information from these results, and exactly how this is done depends on the biological question being asked.

16.3.1 Identification of Proteins in the Absence of a Reference Genome

Although whole genome sequencing costs have plummeted and throughput vastly increased, there remains a large number of species for which a genome sequence is not currently available, or if a genome is available it is not well annotated (*i.e.* the protein coding regions are not well characterised, so a proteome is not available). At the time of writing, many important species fall into this category, including a number of disease vectors and food crops. Traditionally, proteins in such species have been identified by searching against the proteome of a closely related organism, but this is far from ideal as it biases results towards what is already known and limits the potential for new discoveries.

A published Galaxy workflow[38] supporting a PIT analysis to identify proteins in the absence of a reference genome is shown in Figure 16.5. The first step of the workflow produces a protein database comprising the longest open reading frames (ORFs) found within all six reading frames of each transcript. This database is then used in the peptide spectrum matching step, followed by post-processing to score PSMs (see Chapter 4) and group peptides to ORFs (see Chapter 5). In the final part of the workflow, all identified ORFs are BLASTed to find homologous proteins in selected species. The final result of the workflow is therefore a tabular file containing a list of ORFs for which peptide evidence has been found, together with an indication of the closest homologous protein in the selected species. This allows for the analysis of a sample from a species for which a reference genome is not available, by post-identification comparison with proteins in a closely related species. It also facilitates metaproteomics where proteins from two or more species (some of which may be unidentified) are present in the sample.

Figure 16.6 shows peptide and protein identifications from a PIT analysis compared with those from a similar search against UniProt using data from HeLa cells infected with adenovirus.[7] There is clearly a very significant overlap, with 87% of all identified protein groups being common to both analyses. The fact that only 247 UniProt proteins were missed by the PIT approach is very reassuring, particularly because the PIT database was purely derived from the RNA-seq data without any augmentation with canonical proteins. The majority of missed proteins either had no corresponding transcripts in the sample or were variants. Of the 158 proteins that were only found by PIT, the majority only had a single peptide hit in the UniProt search.[38] Note that, in this case, a BLAST *e*-value below 1×10^{-30} was used as the threshold for protein homology, so many of the PIT proteins matched to UniProt were in fact variants. Stricter sequence comparison allows detailed characterisation of individual sample-specific variation, as discussed in the next section.

16.3.2 Identification of Individual Sequence Variation

Protein level sequence variations, often called single amino acid polymorphisms (SAPs) or single amino acid variations (SAAVs), resulting from single nucleotide polymorphisms (SNPs) should clearly be identifiable in

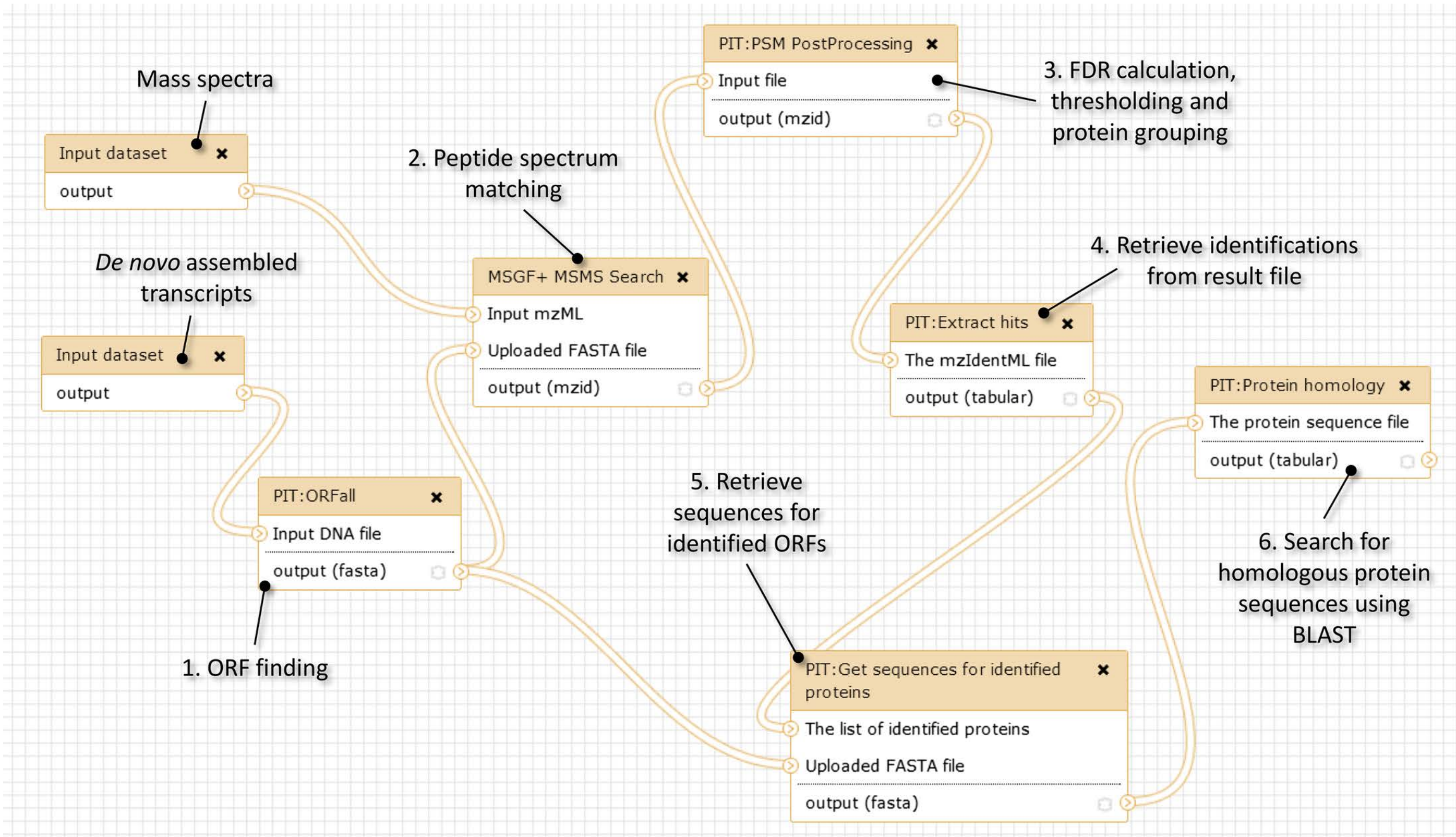

Figure 16.5 Galaxy workflow for conducting protein identification in the absence of a reference genome. The inputs to the workflow are a file containing raw spectral data and a list of transcripts assembled from RNA-seq data (*de novo* assembled in a separate workflow). Figure taken from ref. 38. © American Society of Biochemistry and Molecular Biology.

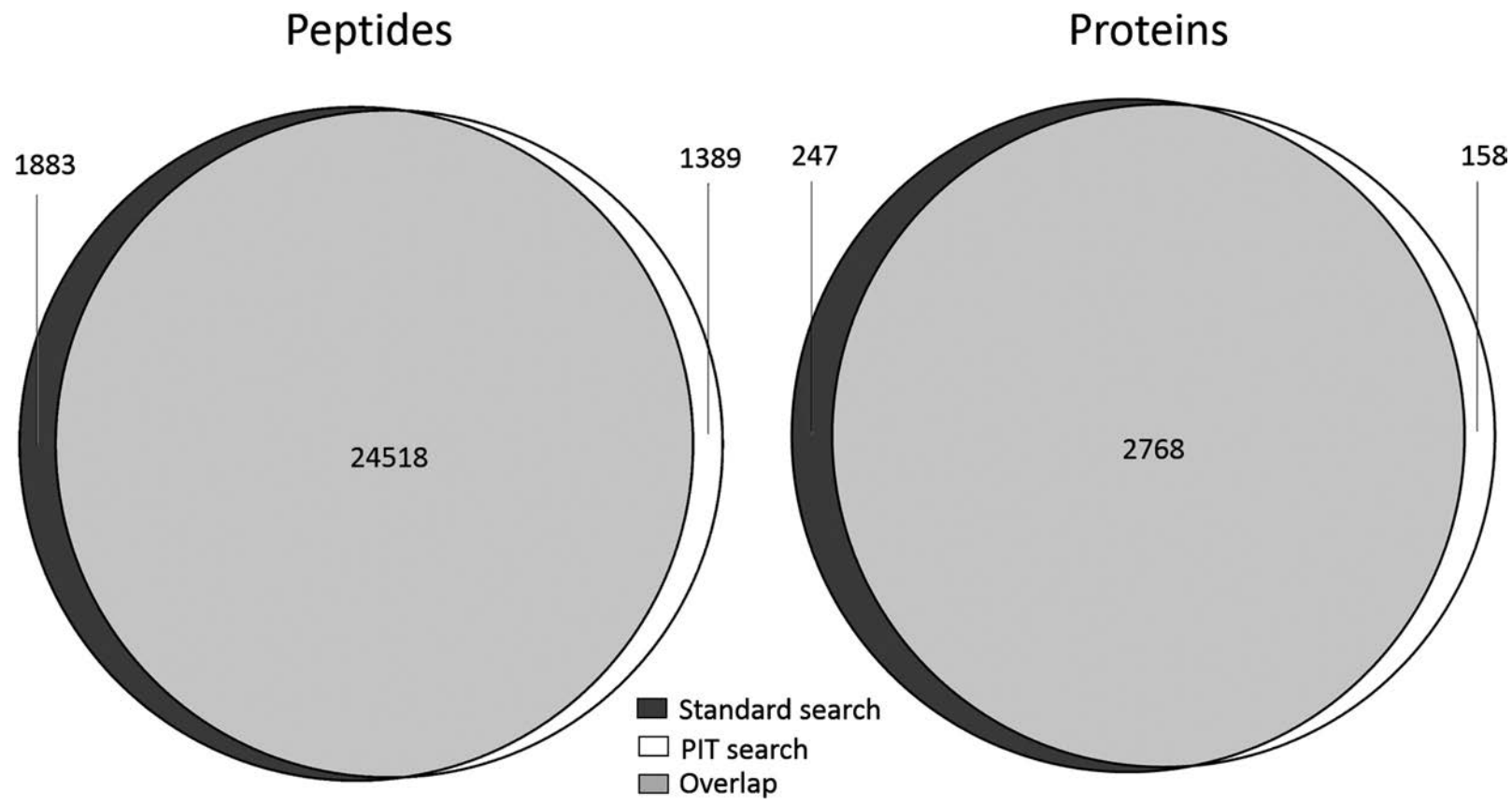

Figure 16.6 Comparison of peptides and protein groups identified by a standard protein identification workflow in which mass spectra were searched against UniProt and a PIT workflow without reference genome, for HeLa cells infected with adenovirus. Figure taken from ref. 38. © American Society of Biochemistry and Molecular Biology.

proteomics mass spectra. Previously proteomics researchers have considered such variations by incorporating known variations from existing variation databases[39] or by identifying non-synonymous variations at the RNA-seq level by employing variant calling tools[7] and introducing additional protein sequences with the variations along with the standard proteome.[36] In both cases, sequence variation identification is limited to model organisms which have a reference genome against which variations have been well characterised. Because it uses a sample-specific database, PIT provides a generically applicable alternative to these approaches, which is able to capture SAPs as well as insertions, deletions, and multiple amino acid alterations.[37]

To achieve this, after the identified ORF sequences are BLASTed against a proteome a variation identification algorithm can detect the variations from the BLAST mapping. One way of reporting these variations at the protein level would be to use a proteomic variant of the popular Variant Call Format (VCF) to report the peptide(s) supporting the variations. VCF is a text-based file format designed to describe genomic locations at which variations occur, and the nature of those variations. A VCF file contains metadata lines that start with ## and a header line (starting with #) followed by the data lines. The data lines have eight fixed fields. These fields are chromosome, position, ID, reference base, alternative base, PHRED scaled quality value, filter and information. We have proposed a proVCF format, based on the VCF file format but with the intention of describing protein variations. It also has all eight fields, but the position (POS) field gives position within a protein rather than a genomic location. The INFO column is designed to contain additional proteomics data. There are five fields in this column, separated by semicolons: subject id, query id and alignment from the BLAST result and type of

the variation (Type) and the position in respect to the ORF sequence (QPOS). These fields have a 'key = value' format. The alignment field is further divided into six categories, which includes query length (*i.e.* ORF length), query start (alignment start position in an ORF), query end (alignment end position in an ORF), subject length (reference protein length), subject start and end (*i.e.* alignment start and location in the reference protein). The QUAL (quality) field gives an average score of peptide identification q-value instead of PHRED scaled quality score, or a negative score if no peptide was found for the variation. An example of an early draft of the proVCF format is shown in Figure 16.7. Peptide evidence is identified for all variations by finding peptides that overlap with the variation boundary. Insertions, deletions and alterations may have partial peptide evidence, and this information too can be captured in the proVCF file. More work is needed to finalise the format, in particular to make it compatible with existing VCF tools, but the way in which such a format could be useful is evident.

16.3.3 Monitoring Isoform Switching

Protein isoforms are known to have a diverse range of functionalities[40] and isoform switching has been implicated in human diseases (*e.g.* cancer[41]). Previous research shows that a large portion of the disease causing mutations in human affect splicing rather than directly altering the coding sequence.[42] Isoforms can interact differently with other proteins, modifying their behaviour in pathways. Isoform usage is therefore an active area of study, but identification of protein isoforms is challenging due to the usual problem of finding a suitable search database. For model organisms, databases of protein isoform sequences derived from gene structure are available, but including these in a database search substantially increase search space. Furthermore, there is no guarantee that every isoform present in the database is actually seen in nature, or that the sample does not contain a novel isoform resulting from, for example, somatic mutation.

```
#CHROM  POS     ID        REF     ALT     QUAL    FILTER  INFO
11      100     78682.1   R       K       0       PASS
        SubjectId=P28347;QueryId=asmbl_10737;Alignment=[QueryLength=415:QueryStart=1:QueryEnd=415:SubjectLength=426:SubjectStart=16:SubjectEnd=426]
;Type:SSAP;QPOS:85;UniquePeptide:1;Sequence:AYHEQLSVAEITNACFEMVK
17      51      79006.1   EDLFQDLSHFQETWLAE       E       0.000004        PASS
        SubjectId=P43268;QueryId=asmbl_25708;Alignment=[QueryLength=461:QueryStart=1:QueryEnd=461:SubjectLength=484:SubjectStart=1:SubjectEnd=477];
Type:DEL;QPOS:51;UniquePeptide:0;Sequence:-
1       1530    79686.1   S       E       -1      FAILD
        SubjectId=P49454;QueryId=asmbl_5912;Alignment=[QueryLength=2482:QueryStart=801:QueryEnd=2482:SubjectLength=3210:SubjectStart=1529:SubjectE
nd=3210];Type:SAP;QPOS:802;UniquePeptide:1;Sequence:-
1       1611    79686.2   V       A       0       PASS
        SubjectId=P49454;QueryId=asmbl_5912;Alignment=[QueryLength=2482:QueryStart=801:QueryEnd=2482:SubjectLength=3210:SubjectStart=1529:SubjectE
nd=3210];Type:SAP;QPOS:883;UniquePeptide:1;Sequence:NPEISHLLANPDIMR
6       114     12735.1   E       D       0       PASS
        SubjectId=P08107;QueryId=asmbl_49775;Alignment=[QueryLength=641:QueryStart=1:QueryEnd=641:SubjectLength=641:SubjectStart=1:SubjectEnd=641];
Type:SSAP;QPOS:114;UniquePeptide:1;Sequence:NQDLALSNLESIPGGYNALR
12      287     310149.1  T       S       0.000007        PASS
        SubjectId=Q9BQE3;QueryId=asmbl_13329;Alignment=[QueryLength=451:QueryStart=1:QueryEnd=451:SubjectLength=449:SubjectStart=1:SubjectEnd=449]
;Type:SSAP;QPOS:287;UniquePeptide:0;Sequence:-
```

Figure 16.7 An example of the draft proVCF format for reporting protein variations identified by PIT. The format is a tab delimited text file with eight columns. The last column (INFO) is multicomponent string where each field has 'key = value' structure providing more information about the variation. In this figure the content of the INFO column had to be wrapped around to the following line.

The search space can be reduced significantly by using RNA-seq data instead of the genome, which also allows capture of alternative splicing events without requiring a genome annotation. A common approach to confirm alternative splicing events at the protein level is by identifying splice junction peptides,[43] or by identifying other peptides that uniquely map to the isoforms. However, these methods rely on high quality gene structure annotation from databases like Ensembl, restricting this technique to all but the most well annotated model organisms. In another isoform detection study,[44] an existing database has also been used to complete incomplete assemblies.

At the time of writing Swiss-prot has 20 195 reviewed canonical proteins sequences and 42 156 sequences including isoforms in the human proteome, so including reviewed isoforms in a proteomics experiment doubles the search space. Alternative splicing of the protein-coding regions results from several splicing events, of which there are five basic types[41,45] and two additional types. These are listed here and shown graphically in Figure 16.8.

(a) Exon skipping/retention: Exon skipping/retention is the most common splicing event where mRNAs may contain an exon under certain conditions or in particular tissue and not in others.

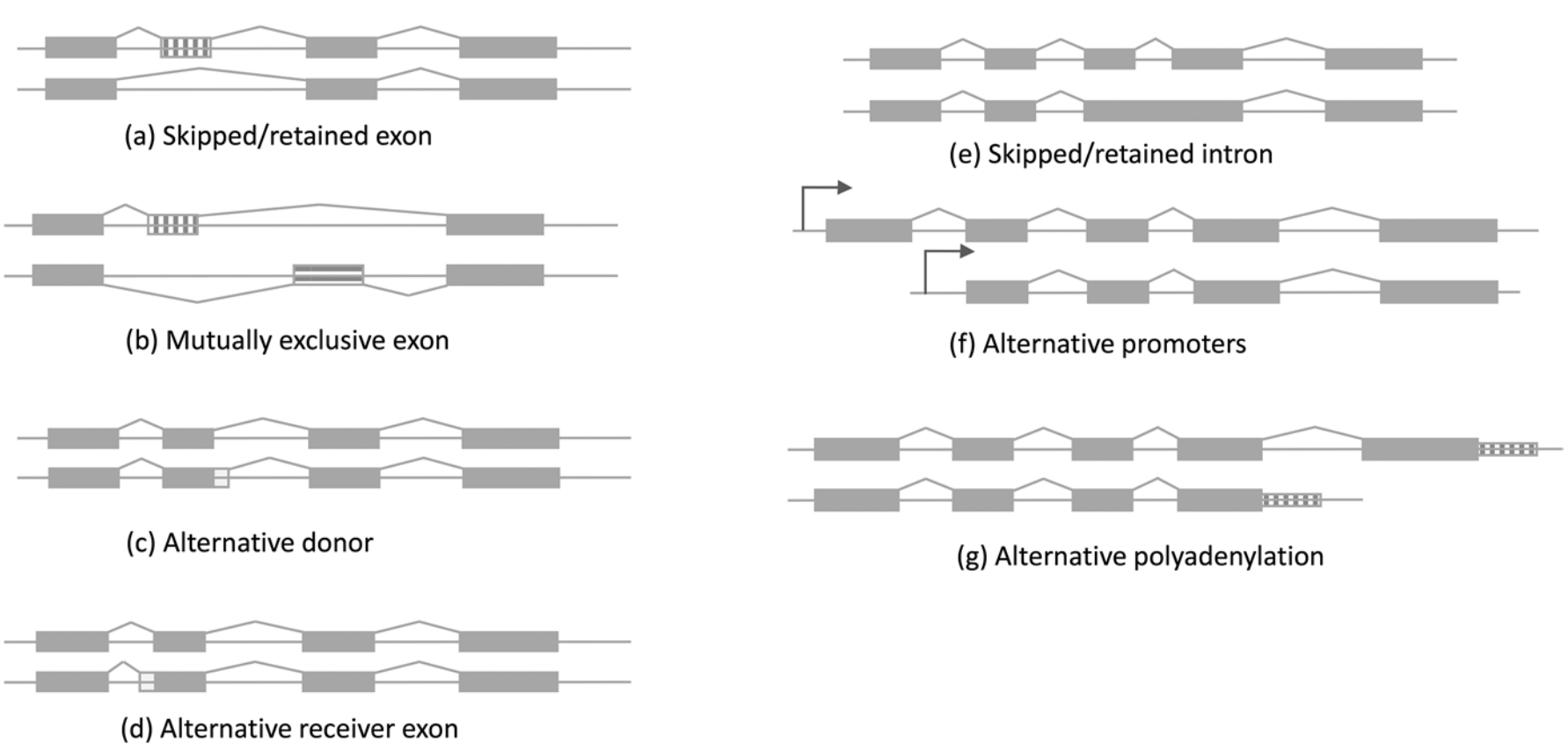

Figure 16.8 Alternative splicing events commonly observed. (a) Example of a skipped or retained exon. (b) Mutually exclusive exons are part of a group of exons where only one member can appear at a time in the mRNA. (c) An alternative donor site can lengthen or shorten exons and changes the 3-prime end of exons. (d) An alternative receiver is the opposite of alternative donor. It changes the 5-prime end of exons. (e) mRNA incorporates an intron when the splicing machinery fails to cleave the intron. (f) Multiple promoter regions can switch the 5-prime end of the mRNA and create isoforms with different starting exons. (g) Multiple polyadenylation sites switch the 3-prime ends of mRNA creating isoforms with different terminating exons.

(b) Mutually exclusive exon: Mutually exclusive exons occur when only one exon from a group of exons is included at a time in the mRNA.
(c) Alternative donor sites: Exons can lengthen or shorten due to alternative 5-prime splice sites. A single base mutation can change normal splicing sites and force into a longer exon. Alternative donor sites change the 3-prime boundary of exons.
(d) Alternative receiver sites: An alternative receiver site is similar to an alternative donor site, except it changes the 5-prime boundary of an exon.
(e) Intron retention: introns are removed during or shortly after transcription. A donor site (5-prime end of introns), branch site and an acceptor site (3-prime end of introns) are required for intron splicing.[46,47] mRNA incorporates an intron when the RNA splicing fails to splice both the members of the donor-acceptor altogether.
(f) Alternative promoters: Multiple promoter regions change the 5-prime end of mRNAs resulting in alternative star exons.
(g) Alternative polyadenylation: Alternative polyadenylation sites create mRNA isoform with different terminating exons.

In addition to these alternative splicing events, there are two other mechanisms by which multiple mRNAs may be produced from same gene. Multiple promoter and polyadenylation sites switch five prime and three prime ends of mRNAs.

According to Zhou *et al.* 8% of protein variants are generated from mRNA alternative splicing or SNPs, the remainder being mostly due to PTMs.[48] Research is ongoing to identify alternative splicing events by combining RNA-seq and LC-MS/MS proteomics. Protein level evidence for splice junctions have been identified using both public[44] and sample-specific RNA-seq data. So far, confirmation of isoforms at protein level has relied on either splice junction peptides or peptides uniquely mapping to an isoform. PIT allows a different approach to identify protein and mRNA isoforms, in which we identify protein isoforms based on ORF sequence and then find the cause from the RNA-seq data. ORFs are BLASTed against a standard protein database and isoforms are classified based on the BLAST alignment. We use multiple layers of filtering of alignments to separate novel proteins from the isoforms. If an ORF sequence is significantly smaller than the target sequence, or the alignment of below certain quality, the match is discarded. Transdecoder assigns a *type* to each predicted ORFs. If start, stop, or both codons are missing, the ORFs are labelled as five prime partial, three prime partial and internal respectively. Unlike previous approaches our method does not use an existing database such as Ensembl in an attempt to complete these incomplete ORFs or to identify translation frame. The ORF will be considered to be an isoform considering a combination of its length, completeness, transcript structure and peptide evidence. In contrast, an mRNA isoform is confirmed by identifying peptides from lengthened exons, or retained intron region. Incompleteness of ORFs is also recorded for further study. Identified ORFs with any of the splicing events are annotated accordingly and reported.

Crucially, unlike proteogenomics approaches covered in the previous chapter, this novel approach is able to identify and characterise isoforms without access to a well annotated reference genome.

16.3.4 Genome Annotation and Discovery of Novel Translated Genomic Elements

Thus far in this chapter we have concentrated on identification of proteins and their variants, as this is the focus of proteomics. However, studies have reported mass spectral evidence of peptides that map to the genome but are not considered to belong to what would normally be considered a protein. These include short ORFs (sORFs) typically coded for by genome sequences near protein coding genes, and peptides from various forms of so-called non-coding RNA (ncRNA). The veracity and function of these entities is the subject of much study, and disease relevance has been postulated in the literature.[49]

Various strategies can be used to seek proteomic evidence for these exotic products of the genome, the simplest being to add their sequences to a canonical protein database and use the combined database for peptide spectrum matching. This is possible, as repositories of short ORFs and ncRNAs do exist for several species, and these features might feasibly be computationally predicted from the genomes of others. However, PIT offers the opportunity to find such features from LC-MS/MS data regardless of whether they have been seen, or predicted, previously. There are, however, a number of open challenges here, particularly for sORFs that are, by definition, short and may only be supported by a single peptide identification. In these cases, diligent scoring of peptide spectrum matches, and experimental validation of findings, become even more important than in standard proteomics experiments.

16.4 Reporting and Storing PIT Results

Determining how best to share, report and store the results of a proteomics experiment has required a great deal of time invested by the Proteomics Standards Initiative and the developers of the various proteomics databases. The transcriptomics community has undertaken similar efforts for RNA-seq. Since PIT subsumes both these fields the reporting of results requires an understanding of practices in both communities, and a mechanism for their linkage.

At the very least, when reporting the results of a PIT experiment the RNA-derived sample-specific sequence database used should be provided alongside any protein–peptide identification results, with the same identifiers used in both. The most basic technical solution to achieve this is to provide the necessary sequences as a FASTA file, but a more elegant solution is inclusion of the sequences together with the identifications in a single mzIdentML file using the <DBSequence> tag (see Chapter 11). This would be compatible with

a full submission to public databases *via* ProteomeXchange.[50] This solution does not, however, capture the underlying transcriptomic data and associated information such as sequence quality. For that, it would be necessary to store the RNA-seq data, which could be in the form of a FASTA file of assembled transcripts or could also include the original FASTQ raw sequencing files, to facilitate full reanalysis. Ideally, in future it should be possible to make a joint submission in which, for example, proteomics data is submitted to PRIDE[51] and the associated RNA-seq data to ArrayExpress.[52]

Aside from simple deposition and sharing of data, it is also worth considering how PIT results could best be represented logically, *e.g.* in a relational database. The great majority of traditional proteomics experiments and databases are protein-centric, so databases are designed around protein identifications with additional information about the supporting peptide evidence. PIT differs in that it is capable of identifying a range of protein variants and other polypeptides that may not fit into the generally accepted definition of a protein. For PIT data we have developed an alternative database called PITDB (www.pitdb.org) designed around the more general concept of translated genomics element (TGEs). Each TGE is an entity from the search database that has been identified, and is assigned to a class determined by analysis of the TGE's amino acid sequence. For example, if it has been found to have very close homology to a known protein it would be classed as a known protein, but other analysis may reveal it to be a protein variant, or another type of TGE such as a sORF or translated ncRNA. Each TGE must be supported in the database by one or more TGE observations, which is similar in concept to a protein identification except that evidence underpinning a TGE observation consists of both peptide spectrum matches and the transcript evidence used to generate the TGE's amino acid sequence. Each type of evidence has its own associated confidence score. Using this approach, it is possible to build up a body of evidence for the existence of each individual TGE over the course of multiple experiments, with ready access to the information needed to judge the level of confidence in the existence of the TGE and any sequence variations within it.

16.5 Applications of PIT

Researchers have addressed a wide range of biological questions through the combination of RNA-seq and proteomics. Publicly available RNA data have previously been used with proteomics to help reduce database search space significantly.[53] However, that approach is unable to capture sample specific mutations, isoforms, and novel polypeptides. Moreover, publicly available RNA-seq data for non-model organisms are sparsely available making it unsuitable for non-model organisms. Use of sample specific transcriptome data in PIT and similar methods have allowed researchers to identify single nucleotide polymorphisms (SNPs) or single amino acid polymorphism (SAPs) in cancer.[54] The method has shown promising results for genome annotation of non-model organisms, *e.g.*[55] and to improve existing annotation.[56] PIT

has also identified novel human isoforms.[43] PIT and similar methods[36] have also been used to identify disease biomarkers,[57–59] and helped reveal how different hosts react differently to a common virus.[60]

16.6 Conclusions

Of all the methodologies covered in this book, PIT is the least mature so much remains to be done in terms of optimising workflows and reporting results. However, significant progress is being made and PIT has already helped to answer previously intractable biological questions. Although challenging, such integration of different omics techniques is crucial to the future of biological research, as it becomes clear that only a system-wide view can provide us with a full understanding of crucial biological phenomena. Proteomics is obviously an integral part of this future, and understanding the computational steps needed to convert mass spectrometry data into biologically important information is essential to ensuring high quality results and valid conclusions.

Acknowledgements

The authors would like to acknowledge funding from Queen Mary University of London, and BBSRC grants BB/M020118/1 and BB/M006174/1.

References

1. S. J. Hubbard, Computational approaches to peptide identification *via* tandem MS, *Methods Mol. Biol.*, 2010, **604**, 23–42.
2. The UniProt Consortium, UniProt: a hub for protein information, *Nucleic Acids Res.*, 2015, **43**(Database issue), D204–12.
3. M. S. Kim, *et al.*, A draft map of the human proteome, *Nature*, 2014, **509**(7502), 575–581.
4. M. Wilhelm, *et al.*, Mass-spectrometry-based draft of the human proteome, *Nature*, 2014, **509**(7502), 582–587.
5. D. L. Black, Mechanisms of alternative pre-messenger RNA splicing, *Annu. Rev. Biochem.*, 2003, **72**, 291–336.
6. R. L. Hettich, *et al.*, Metaproteomics: harnessing the power of high performance mass spectrometry to identify the suite of proteins that control metabolic activities in microbial communities, *Anal. Chem.*, 2013, **85**(9), 4203–4214.
7. V. C. Evans, *et al.*, *De novo* derivation of proteomes from transcriptomes for transcript and protein identification, *Nat. Methods*, 2012, **9**(12), 1207–1211.
8. Z. Wang, M. Gerstein and M. Snyder, RNA-Seq: a revolutionary tool for transcriptomics, *Nat. Rev. Genet.*, 2009, **10**(1), 57–63.
9. Illumina Inc., An Introduction to Next-Generation Sequencing Technology. Accessed on 24-05-2016.

10. D. R. Bentley, *et al.*, Accurate whole human genome sequencing using reversible terminator chemistry, *Nature*, 2008, **456**(7218), 53–59.
11. C. Ledergerber and C. Dessimoz, Base-calling for next-generation sequencing platforms, *Briefings Bioinf.*, 2011, 489–497.
12. M. Costello, *et al.*, Discovery and characterization of artifactual mutations in deep coverage targeted capture sequencing data due to oxidative DNA damage during sample preparation., *Nucleic Acids Res.,* 2013, **41**(6), e67.
13. L. A. Clarke, *et al.*, PCR amplification introduces errors into mononucleotide and dinucleotide repeat sequences, *Mol. Pathol.*, 2001, 351–353.
14. A. Stein, T. E. Takasuka and C. K. Collings, Are nucleosome positions *in vivo* primarily determined by histone-DNA sequence preferences?, *Nucleic Acids Res.*, 2010, **38**(3), 709–719.
15. O. Harismendy, *et al.*, Evaluation of next generation sequencing platforms for population targeted sequencing studies, *Genome Biol.*, 2009, **10**(3), R32.
16. D. I. Lou, *et al.*, High-throughput DNA sequencing errors are reduced by orders of magnitude using circle sequencing, *Proc. Natl. Acad. Sci. U. S. A.*, 2013, **110**(49), 19872–19877.
17. B. Langmead, *et al.*, Ultrafast and memory-efficient alignment of short DNA sequences to the human genome, *Genome Biol.*, 2009, **10**(3), 1.
18. C. Trapnell, L. Pachter and S. L. Salzberg, TopHat: discovering splice junctions with RNA-Seq, *Bioinformatics*, 2009, **25**(9), 1105–1111.
19. H. Li and R. Durbin, Fast and accurate short read alignment with Burrows-Wheeler transform, *Bioinformatics*, 2009, **25**(14), 1754–1760.
20. M. G. Grabherr, *et al.*, Full-length transcriptome assembly from RNA-Seq data without a reference genome, *Nat. Biotechnol.*, 2011, **29**(7), 644–652.
21. D. R. Zerbino and E. Birney, Velvet: algorithms for *de novo* short read assembly using de Bruijn graphs, *Genome Res.*, 2008, **18**(5), 821–829.
22. M. H. Schulz, *et al.*, Oases: robust *de novo* RNA-seq assembly across the dynamic range of expression levels, *Bioinformatics*, 2012, 1086–1092.
23. G. Robertson, *et al.*, *De novo* assembly and analysis of RNA-seq data, *Nat. Methods*, 2010, **7**, 909–912.
24. B. J. Haas, *et al.*, Improving the Arabidopsis genome annotation using maximal transcript alignment assemblies, *Nucleic Acids Res.*, 2003, 5654–5666.
25. *FastQC A Quality Control tool for High Throughput Sequence Data*, 2016, available from: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/.
26. S. Anders, P. T. Pyl and W. Huber, HTSeq—a Python framework to work with high-throughput sequencing data, *Bioinformatics,* 2015, **31**(2), 166–169.
27. R. K. Patel and M. Jain, NGS QC Toolkit: a toolkit for quality control of next generation sequencing data, *PLoS One*, 2012, **7**(2), e30619.
28. H. Li, *et al.*, The Sequence Alignment/Map format and SAMtools, *Bioinformatics*, 2009, **25**(16), 2078–2079.

29. P. Rice, I. Longden and A. Bleasby, EMBOSS: the European Molecular Biology Open Software Suite, *Trends Genet.*, 2000, **16**(6), 276–277.
30. B. J. Haas, *et al.*, *Transdecoder*, available from: https://transdecoder.github.io/.
31. GeneID. [cited 2016 24-05-2016]; available from: http://genome.crg.es/software/geneid/.
32. K. Eilbeck, *et al.*, The Sequence Ontology: a tool for the unification of genome annotations, *Genome Biol.*, 2005, **6**(5), R44.
33. BED format. Available from: https://genome.ucsc.edu/FAQ/FAQformat.html#format1. Accessed 25-05-2016.
34. R. Craig, J. P. Cortens and R. C. Beavis, Open source system for analyzing, validating, and storing protein identification data, *J. Proteome Res.*, 2004, **3**(6), 1234–1242.
35. A. I. Nesvizhskii, Proteogenomics: concepts, applications and computational strategies, *Nat. Methods*, 2014, **11**(11), 1114–1125.
36. X. Wang, *et al.*, Protein Identification Using Customized Protein Sequence Databases Derived from RNA-Seq Data., *J. Proteome Res.,* 2012, **11**(2), 1009–1017.
37. G. M. Sheynkman, *et al.*, Large-scale mass spectrometric detection of variant peptides resulting from non-synonymous nucleotide differences, *J. Proteome Res.*, 2014, **13**(1), 228–240.
38. J. Fan, *et al.*, Galaxy Integrated Omics: Web-based Standards-Compliant Workflows for Proteomics Informed by Transcriptomics, *Mol. Cell Proteomics*, 2015, **14**(11), 3087–3093.
39. M. K. Bunger, *et al.*, Detection and validation of non-synonymous coding SNPs from orthogonal analysis of shotgun proteomics data, *J. Proteome Res.*, 2007, **6**(6), 2331–2340.
40. X. Yang, *et al.*, Widespread Expansion of Protein Interaction Capabilities by Alternative Splicing, *Nat. Methods,* 2016, **13**(4), 291.
41. R. I. Skotheim and M. Nees, Alternative splicing in cancer: noise, functional, or systematic?, *Int. J. Biochem. Cell Biol.*, 2007, **39**(7–8), 1432–1449.
42. N. Lopez-Bigas, *et al.*, Are splicing mutations the most frequent cause of hereditary disease?, *FEBS Lett.*, 2005, **579**(9), 1900–1903.
43. G. M. Sheynkman, *et al.*, Discovery and mass spectrometric analysis of novel splice-junction peptides using RNA-Seq, *Mol. Cell. Proteomics*, 2013, **12**(8), 2341–2353.
44. A. P. Tay, *et al.*, Proteomic Validation of Transcript Isoforms, Including Those Assembled from RNA-Seq Data, *J. Proteome Res.*, 2015, **14**(9), 3541–3554.
45. M. Sammeth, S. Foissac and R. Guigó, A General Definition and Nomenclature for Alternative Splicing Events, *PLoS Comput. Biol.*, 2008, **4**(8), e1000147.
46. A. J. Taggart, *et al.*, Large-scale mapping of branchpoints in human pre-mRNA transcripts in vivo, *Nat. Struct. Mol. Biol.*, 2012, **19**(7), 719–721.
47. A. Corvelo, *et al.*, Genome-wide association between branch point properties and alternative splicing, *PLoS Comput. Biol.*, 2010, **6**(11), e1001016.

48. A. Zhou, F. Zhang and J. Y. Chen, PEPPI: a peptidomic database of human protein isoforms for proteomics experiments, *BMC Bioinf.*, 2010, **11**(suppl. 6), S7.
49. S. J. Andrews and J. A. Rothnagel, Emerging evidence for functional peptides encoded by short open reading frames, *Nat. Rev. Genet.*, 2014, **15**(3), 193–204.
50. J. A. Vizcaino, *et al.*, ProteomeXchange provides globally coordinated proteomics data submission and dissemination, *Nat. Biotechnol.*, 2014, 223–226.
51. J. A. Vizcaino, *et al.*, 2016 update of the PRIDE database and its related tools, *Nucleic Acids Res.*, 2016, **44**(D1), D447–56.
52. N. Kolesnikov, *et al.*, ArrayExpress update–simplifying data submissions, *Nucleic Acids Res.*, 2015, **43**(Database issue), D1113–6.
53. S. Woo, *et al.*, Proteogenomic database construction driven from large scale RNA-seq data, *J. Proteome Res.*, 2014, **13**(1), 21–28.
54. K. V. Ruggles, *et al.*, An Analysis of the Sensitivity of Proteogenomic Mapping of Somatic Mutations and Novel Splicing Events in Cancer, *Mol. Cell. Proteomics*, 2016, **15**(3), 1060–1071.
55. D. S. Kelkar, *et al.*, Annotation of the zebrafish genome through an integrated transcriptomic and proteomic analysis, *Mol. Cell. Proteomics*, 2014, **13**(11), 3184–3198.
56. S. Chocu, *et al.*, Forty-four novel protein-coding loci discovered using a proteomics informed by transcriptomics (PIT) approach in rat male germ cells, *Biol. Reprod.*, 2014, **91**(5), 123.
57. S. Woo, *et al.*, Proteogenomic strategies for identification of aberrant cancer peptides using large-scale next-generation sequencing data, *Proteomics*, 2014, **14**(23–24), 2719–2730.
58. C. H. Huang, *et al.*, Onco-proteogenomics identifies urinary S100A9 and GRN as potential combinatorial biomarkers for early diagnosis of hepatocellular carcinoma, *BBA Clin.*, 2015, **3**, 205–213.
59. J. C. Kim, *et al.*, Complex Behavior of ALDH1A1 and IGFBP1 in Liver Metastasis from a Colorectal Cancer, *PLoS One*, 2016, **11**(5), e0155160.
60. J. W. Wynne, *et al.*, Proteomics informed by transcriptomics reveals Hendra virus sensitizes bat cells to TRAIL-mediated apoptosis, *Genome Biol.*, 2014, **15**(11), 532.

Subject Index