Dissertation

submitted to the
Combined Faculties for Natural Sciences and for Mathematics
of the Ruperto-Carlo University of Heidelberg, Germany

for the Degree of
Doctor of Natural Sciences

presented by
Dipl.-Bioinf. Philip Alexander Kerner
born in Herdecke, Germany

Date of Disputation: May 18, 2015

MS_QBAT

A Software Suite for LC-MS Protein Quantification

Referees: Prof. Dr. Matthias Mayer

Prof. Dr. Lars Kaderali

Abstract

Accessing the relative changes in protein abundance is essential for a proper understanding of the various processes underlying disease progression and development. Nowadays, mass spectrometry-based proteomics allows for the identification of several thousand proteins in a single analysis. Unfortunately, mass spectrometry is inherently not quantitative, which is why additional techniques for protein quantification have to be developed.

To measure quantitative changes in protein abundance, biological samples need either to be labeled using stable isotopes or protein abundances have to be computed using so called label-free techniques.

Label-based quantification approaches are costly and the number of samples that can be quantified against each other is limited. Furthermore, depending on the sample, the introduction of the labels can be elaborate. Label-free quantification is not confronted with these limitations; principally, an unlimited number of samples can be quantified without the introduction of isotopes.

Yet these advantages have their price: The development of label-free quantification algorithms is not trivial and requires profound knowledge both in bioinformatics and mass spectrometry. Namely the design of systems flexible enough to quantify data deriving from different mass spectrometric systems and proteomic workflows require additional experience and time.

Existing software solutions for the quantitative analysis of data obtained by liquid chromatography followed by mass spectrometry largely lack two main properties that are important for their routine application: (i) usability by non-informaticians and (ii) flexibility with regard to instrumentation and/or workflow. Today, no software exists for the quantification of data from state-of-the-art LC-MALDI-MS systems. LC-MALDI-MS is a widely used system for routine applications in proteomics and as well in other fields of diagnostics. In contrast to LC-ESI-MS, the decoupling of liquid chromatographic sample separation and mass spectrometric sample analysis allows to separate MS^1 and MS^2 in time, thereby providing unique possibilities for advanced and dynamic data acquisition mostly unused today.

In order to quantify data acquired by LC-MALDI-MS and to take full advantage of the mentioned LC-MALDI-MS specific features, a novel software suite termed MS_QBAT was developed and evaluated. MS_QBAT is a platform independent software suite for MS^1-based, label-free protein quantification. In contrast to other software solutions, MS_QBAT is highly flexible and suited for

the quantification of mass spectrometric data from various instrumental setups and proteomic workflows, such as (Ge)LC-MALDI-MS and (Ge)LC-ESI-MS.

Quantification capabilities were evaluated using spike-in experiments analyzed using both different proteomic workflows and instruments. Human proteins were spiked in variable concentrations into a complex *E.coli* background proteome and processed using both an LC-MS and a GeLC-MS approach. Samples were chromatographically separated on a nanoACQUITY UPLC ® system using a 120 minutes gradient and subsequently analyzed by an AB SCIEX TOF/TOF™ 5800 system and an AB SCIEX QTRAP® 6500 system. Furthermore, a publicly available quantification benchmark data set [1] has been used to evaluate LC-ESI-MS quantification capabilities.

Obtained results show that MS_QBAT can be applied to quantify data deriving from both LC-/GeLC-MALDI-MS and LC-/GeLC-ESI-MS workflows with high accuracy. Therefore, this software suite has a range of application outperforming all currently available solutions.

Zusammenfassung

Das Wissen über relative Änderungen der zellulären Proteinmengen ist essentiell für unser Verständnis von zellbiologischen Prozessen, die beispielsweise für die Entstehung und Entwicklung diverser Krankheiten verantwortlich sind. Massenspektrometie basierte Proteomik erlaubt heutzutage die Identifikation tausender Proteine in einer einzigen Analyse. Unglücklicherweise ist die Massenspektrometrie grundsätzlich kein quantitatives Verfahren, weswegen zusätzliche Techniken zur Proteinquantifizierung erforderlich sind.

Um quantitative Änderungen in der Expression von Proteinen messen zu können, müssen biologische Proben entweder mit stabilen Isotopen markiert werden oder die Proteinabundanz muss durch markierungsfreie, computergestützte Methoden berechnet werden.

Isotopen-basierte Methoden zur Proteinquantifizierung sind kostspielig und die Anzahl der Proben, die gleichzeitig verglichen und somit quantifiziert werden können, ist limitiert. Weiterhin kann sich die Einführung der Markierung, je nach Probenmaterial, als sehr aufwendig bzw. unmöglich gestalten. Bei markierungsfreien Methoden treten diese Probleme nicht auf; prinzipiell kann eine unlimitierte Anzahl Proben analysiert und quantifiziert werden, ohne dass die Einführung von Isotopen nötig ist.

Diese Vorteile gegenüber markierungsbasierten Verfahren haben allerdings ihren Preis: Die Entwicklung von Algorithmen zur markierungsfreien Quantifizierung ist nicht trivial und erfordert detaillierte Kenntnisse sowohl der Bioinformatik als auch der Massenspektrometrie. Insbesondere das Entwerfen von Systemen, die flexibel genug sind um Daten verschiedener Systeme quantifizieren zu können, erfordert zusätzliche Erfahrung und ist sehr zeitaufwändig.

Existierenden Softwarelösungen zur quantitativen Auswertung von Daten, die durch Flüssigkeitschromatographie gefolgt von Massenspektrometrie gewonnen wurden, mangelt es heute fast ausnahmslos an zwei wichtigen Eigenschaften, die für eine routinemäßige Anwendung nötig sind: (i) Benutzerfreundlichkeit – speziell für Nicht-Informatiker – und (ii) die bereits erwähnte Flexibilität bezüglich Probenvorbereitung und/oder verwendeter Instrumente. So existiert heute beispielsweise keine Software zur quantitativen Auswertung von Daten, die durch neuste LC-MALDI-MS Geräte gewonnen wurden. LC-MALDI-MS ist ein weit verbreitetes System für proteomische Anwendungen und wird auch in anderen Bereichen der Diagnostik eingesetzt. Im Gegensatz zu LC-ESI-MS bietet das System durch die Entkoppelung von chromatographischer

Probenauftrennung und massenspektrometrischer Probenanalyse einzigartige Möglichkeiten der erweiterten und dynamischen Datenakquise, die bis heute größtenteils ungenutzt bleiben.

Um Daten, die durch LC-MALDI-MS gewonnen wurden, quantitativ auszuwerten und um von den genannten LC-MALDI-MS spezifischen Eigenschaften voll zu profitieren, wurde eine neue Softwarelösung names MS_QBAT entwickelt und evaluiert. MS_QBAT ist eine platformunabhängige Softwarelösung zur markierungsfreien, MS^1-basierten Proteinquantifizierung. Im Gegensatz zu existierenden Lösungen bietet MS_QBAT die Möglichkeit, proteomische Daten, die sowohl durch verschiedenste Probenvorbereitungsarten als auch durch unterschiedliche Analyseinstrumente gewonnen wurden, mit hoher Genauigkeit zu quantifizieren.

Die Möglichkeiten zur Proteinquantifizierung wurden durch Spike-in Experimente evaluiert, die mit Hilfe verschiedener Arbeitsabläufe und auf verschiedenen Geräten prozessiert wurden. Humane Proteine wurden in unterschiedlichen Mengen in ein komplexes *E.coli* Hintergrundproteom gemischt und anschließend in einem LC-MS beziehungsweise GeLC-MS Ansatz analysiert. Die Proben wurden auf einem nanoACQUITY UPLC ® System mit einem 120 Minuten Gradienten chromatografisch aufgetrennt und anschließend durch ein AB SCIEX TOF/TOF™ 5800- beziehungsweise ein AB SCIEX QTRAP® 6500 System analysiert. Darüber hinaus wurden öffentlich verfügbare Benchmark-Daten [1] zur Evaluierung der Kompatibilität mit LC-ESI-MS Daten herangezogen.

Die gewonnenen Ergebnisse zeigen, dass die in dieser Arbeit entwickelte Softwarelösung MS_QBAT LC-/GeLC-MALDI-MS sowie LC-/GeLC-ESI-MS Daten mit hoher Genauigkeit quantifizieren kann. Die neuentwickelte Software verfügt somit über ein Einsatzspektrum, dass über dasjenige aller zur Zeit veröffentlichter Lösungen hinausgeht.

Acknowledgements

First of all I would like to thank Christoph Rösli for his excellent supervision. Without his seemingly limitless knowledge and experience in mass spectrometry, proteomics and big data analysis this work would not have been possible. His enthusiasm and motivation have been contagious and helped to push the project through downs and accelerate it. His impressive ability for creative problem solving helped a lot especially in the final stage of the project. I also would like to thank him for extensive proof reading of this work.

Second, I would like to thank Andreas Trumpp for giving me the opportunity to work at the DKFZ and HI-STEM. I thank him for the trust he had in the project even though it was a very technical one.

I would like to thank my TAC members Jeroen Krijgsveld, Lars Kaderali and Martina Schnölzer for critically reviewing this project. They helped me to keep an objective point of view to the project and gave great advise for improvement.

I thank Matthias Mayer for assuming the chairmanship of my PhD defense at the last-minute and Lars Kaderali for being second assessor.

I would like to thank my colleagues from the lab: Philipp, for turning OMIX-ing marathons into sprints and giving great support to a female superior.

And Wiebke, for her amazing enthusiasm for this project and our collaboration. She is a great coworker and I cannot imagine any better partner for the projects we shared.

I also thank Sabrina, for getting everybody back to earth when necessary and helping me to cut down four days of sample chopping to two. I also thank her for a lot of beta-testing the software. It can be a very annoying thing to do and I appreciated her help very much!

And Amelie, for her support in the wet lab phase of the project. Without her protocols and helping hands this work would certainly have taken another two or three years.

I am also grateful to Laura, for decoding UPLC pressure profiles and for being a real inspiring personality with limitless happiness to share.

And Katharina Billian-Frey, who helped me to set up the first *E.coli* digest.

I thank Sascha Lippert for sleepless weekends of coding and Brother Firetribe. Without him and his superior programming skills MS_QBAT would today maybe miss a very important component. I hope he is not too disap-

pointed that the GeneticAlgorithms library will probably never serve its higher purpose but will be doomed forever to optimize alignment parameters.

I thank Pascal Gellert for being basically always available to help out with his excellent bioinformatic knowledge and off-topic chats to keep motivation up and things rolling.

I thank David Brocks and Paul Kaschutnig for our meat-eating contests which always reminded me that even bioinformatic geeks need more to live than mere air, love and the internet.

I thank Yassen Assenov for excellent help on advanced R topics and Erika Krückel for being the good soul of the lab and for always being available to help with non-scientific topics and German bureaucracy.

I am especially grateful to my family, Jürgen, Renate, Katharina, Johanna, and Kristina for always supporting and encouraging me to follow my path.

Ein großer Dank gilt meiner Familie, Jürgen, Renate, Katharina, Johanna, and Kristina, die mich immer unterstützt und ermutigt hat meinen Weg zu gehen.

To my parents.

“ Hofstadter’s Law: It always takes longer than you expect, even when you take into account Hofstadter’s Law. ”

Douglas Hofstadter

Contents

Abstract v
Zusammenfassung vi
Acknowledgements viii

Contents **xv**
List of Figures xix
List of Screenshots xxi
List of Tables xxiii
List of Listings xxv
List of Equations xxvii
List of Abbreviations xxix

I Introduction **1**

1 The Concept of Tumor Heterogeneity and Cancer Stem Cells **3**
1.1 Models of Tumor Progression 3
1.2 Therapy Failure and Targeted Therapy 10

2 Biomarkers **11**
2.1 Biomarker Discovery 12

3 LC-MS – a Tool for Biomarker Discovery **15**
3.1 LC-MS Workflow 15
3.2 Tandem Mass Spectrometry 24
3.3 Data-Independent and Targeted MS Techniques 26
3.4 The Nature of LC-MS Data 27
3.5 Low-level Data Processing 28

4 Peptide and Protein Quantification from LC-MS Data **31**
4.1 Label-based Peptide Quantification 32
4.2 Label-free Peptide Quantification 36
4.3 Assessment of Label-based and Label-free Quantification . . . 48
4.4 Protein Quantification 49

5 Genetic Algorithms as a Tool for Parameter Optimization **51**

6 Aim of This Work **57**

II Material and Methods **59**

7 Sample Preparation for LC-MS **61**
7.1 *E.coli* Cell Lysate 61
7.2 Preparation for GeLC-MS 62

8 Liquid Chromatography and Mass Spectrometry **65**
8.1 Liquid Chromatography for MALDI-MS 65
8.2 Liquid Chromatography for ESI-MS 67
8.3 Mass spectrometry 67

9 Preprocessing of Data **73**
9.1 LC-MALDI-MS 73
9.2 LC-ESI-MS 74
9.3 GeLC-MALDI-MS 74
9.4 GeLC-ESI-MS 75

10 Software Development **77**

III Results **79**

11 PepSir **81**

12 Cornerstones of Label-free Peptide and Protein Quantification **89**
12.1 Feature Annotation with Peptide Identification Information . . 89
12.2 Feature Boxing as a Novel Feature Extraction Algorithm . . . 91
12.3 Affinity-Driven Feature Alignment 95
12.4 Peptide and Protein Quantification 110
12.5 Annotation Propagation 112
12.6 Sample Groups 113

13 Additional Quantification Components **115**
13.1 Local Intensity Normalization via Spiked-in Internal Standard Peptides 115
13.2 Global Intensity Normalization 119
13.3 Dynamic Complexity Reduction 122
13.4 GA-based Alignment 122
13.5 Advanced Data Aquisition 123

14 Bridging Worlds **125**

15 MS$_Q$BAT **127**
15.1 Data Reading 127
15.2 Data Visualization 128
15.3 Details of Implementation 136

16 Software Validation by Spike-in Experiments **137**
16.1 Label-free Quantification of LC-MALDI-MS Data 137
16.2 Label-free Quantification of GeLC-MS Data 140
16.3 Label-free Quantification of LC-ESI-MS Data 143

IV Discussion **153**

17 Conclusions **155**
17.1 MS$_Q$BAT Allows for the Accurate, Label-free Quantification of LC-MALDI-MS Data 156
17.2 MS$_Q$BAT Allows for the Accurate, Label-free Quantification of GeLC-MALDI-MS and GeLC-ESI-MS Data 157
17.3 MS$_Q$BAT Allows for the Accurate, Label-free Quantification of LC-ESI-MS Data 159
17.4 MS$_Q$BAT Allows for the Quantification of a "Black-and-White" Situation 161

18 Applications of MS$_Q$BAT **163**
18.1 Novel Biotinylation Reagents 163
18.2 Vascular Accessible Markers in Kidney Cancer 164
18.3 PDAC Subtypes 165

19 Closing Remarks and Outlook **167**

Bibliography **170**

List of Figures

1.1 Tumor heterogeneity. 4
1.2 Clonal evolution as a model of tumor progression. 5
1.3 Hierarchical layout of a cell population. 7
1.4 Union of models of tumor progression. 8
1.5 Stemness versus differentiation. 9
1.6 Bidirectional transition into CSCs. 9
1.7 Cancer therapy and tumor relapse. 10

2.1 The -omics cascade. 12

3.1 Functional principle of mass spectrometry. 16
3.2 Tryptic digest of proteins. 16
3.3 Functional principle of liquid chromatography coupled to mass spectrometry. 18
3.4 LC-MS ion sources. 20
3.5 MALDI-TOF mass analyzer. 22
3.6 Tandem mass spectrometry. 25
3.7 ESI and MALDI Mass spectrum. 28

4.1 Fraction of quantified proteins in a sample. 33
4.2 Two label-free quantification strategies. 37
4.3 Feature extraction. 39
4.4 Retention time normalization. 42
4.5 Transitive alignment connections. 43
4.6 Strategies for an alignment of multiple samples. 45

8.1 Gradient configuration for LC-MALDI-MS. 66
8.2 Gradient configuration for LC-ESI-MS. 66

10.1 Overview of MS_QBAT's software architecture. 78

11.1 Context dependency of the term *proteotypic*. 86

12.1 Feature extraction and feature tightening. 94
12.2 Mapping and alignment. 97
12.3 Type hierarchy of six main types. 108
12.4 Peptide- and protein ratios. 111

13.1 Fraction wise intensity normalization. 117

16.1 Quantification results of LC-MALDI-MS data. 138
16.2 Quantification results of GeLC-MS data. 141
16.3 Measured regulations between UPS2 and UPS1. 144

18.1 Comparison of protein quantifications determined by MS_QBAT and by MRM. 166

List of Screenshots

11.1 Main window and preferences window of PepSir. 85

15.1 The “Sample List” view. 129
15.2 The “Sample Details” view. 129
15.3 The “Quant Table” view. 130
15.4 The “Quant Plot” view. 131
15.5 The “Alignment Path” view. 133
15.6 The “2D Peptide Map” view. 134
15.7 The “Quantification” perspective. 135
15.8 The “Details” perspective. 136

List of Tables

3.1 Different types of mass analyzers used in mass spectrometry. . 21

4.1 Alignment terminology. 42

5.1 Genetic algorithms terminology. 52

7.1 Spike-in scheme for GeLC-MS. 62
7.2 Fractionation scheme for gel slices. 63

8.1 Internal standard peptides. 66
8.2 Calibration peptides. 69

9.1 ProteinPilot™ settings. 75

11.1 Peptide numbers before and after filtering by PepSir. 87

16.1 LC-MALDI-MS: Regulation of background- and sample proteins. 146
16.2 GeLC-MALDI-MS: Regulation of background- and sample proteins. 146
16.3 GeLC-ESI-MS: Regulation of background- and sample proteins. 147
16.4 LC-ESI-MS: Regulation of background and sample proteins. . 147
16.5 LC-MALDI-MS: Parameters used for quantification. 148
16.6 GeLC-MALDI-MS: Parameters used for quantification. 149
16.7 GeLC-ESI-MS: Parameters used for quantification. 150
16.8 LC-ESI-MS: Parameters used for quantification. 151

List of Listings

5.1 Pseudo code of genetic algorithms. 55

12.1 Example of a `PeptideSummary.txt` file. 90
12.2 The `Mapping` interface. 96
12.3 The `Alignment` interface. 99
12.4 The `Comparator` interface. 102
12.5 The `ComparatorAlignment` class. 105
12.6 The `Feature` interface. 107
12.7 The `AlignmentPeak` interface. 107
12.8 The `AlignmentFeature` interface. 108

13.1 Example file for internal standard definition. 116
13.2 The `NormalizationStrategy` interface. 118
13.3 The `NormalizationStrategyOnStandards` interface. 119

List of Equations

5.1 The SMA filter. 52

12.1 "MALDI boxes." 93
12.2 Feature intensity. 95
12.3 Alignment score. 100
12.4 The *signum* function. 101
12.5 Peptide ratio. 110
12.6 Protein ratio. 111

13.1 Sample intensity. 120
13.2 The normalization factor. 120
13.3 Sample group intensity. 121
13.4 Normalization factor for sample groups. 121

List of Acronyms

2D-GE	two-dimensional gel electrophoresis . . . 17
ACN	acetonitrile . . . 65
ADC	analog-to-digital converter . . . 23
AIF	all-ion fragmentation . . . 25
API	application programming interface . . . 91
AUC	area under the curve . . . 37
BCA	bicinchoninic acid assay . . . 61
cc	clear cell . . . 164
CHCA	alpha-cyano-4-hydroxycinnamic acid . . . 65
CID	collision-induced dissociation . . . 24
CLI	command-line interface . . . 77
CPU	central processing unit
CSC	cancer stem cell . . . 6
DC	direct current . . . 23
DDA	data-dependent acquisition . . . 26
DIA	data-independent acquisition . . . 21
DP	dynamic programming . . . 40
DTW	dynamic time warping . . . 40
EMS	enhanced MS . . . 71
EMT	epithelial-mesenchymal transition . . . 6
EPI	enhanced product ion . . . 71
ER	enhanced resolution . . . 71
ESI	electrospray ionization . . . 15
FA	formic acid . . . 67
FDR	false discovery rate . . . 168
FP	false positive . . . 109
FT	Fourier transformation . . . 23
FWHM	full width at half maximum . . . 69
GA	genetic algorithm . . . 51
GeLC	SDS-PAGE followed by in-gel digestion followed by LC . . . 17
GUI	graphical user interface . . . 77
ICAT	isotope-coded affinity tag . . . 36
ICPL	isotope-coded protein label . . . 36

ICR	ion cyclotron resonance
IDE	integrated development environment 77
I/O	input/output 91
IR	infrared 19
iTRAQ	isobaric tags for relative and absolute quantitation 36
JVM	java virtual machine
LC	liquid chromatography 15
LIT	linear ion trap 22
MALDI	matrix-assisted laser desorption/ionization 15
MIC	metastasis initiating cell 8
MOPS	3-(N-morpholino)propansulfonic acid 62
MRM	multiple reaction monitoring 165
MS	mass spectrometry 15
m/z	mass-to-charge ratio 19
NSG	NOD scid gamma
ORF	open reading frame 12
OSGi	open service gateway initiative 77
PDAC	pancreatic ductal adenocarcinoma 165
pI	isoelectric point 17
ppm	parts per million 20
qPCR	quantitative polymerase chain reaction 164
QQQ	triple quadrupole 21
RCC	renal cell carcinoma 164
RCP	rich client platform 77
RT	real-time 164
SCM	source code management 77
SC	spectral counting 46
SDS-PAGE	sodium dodecyl sulfate polyacrylamide gel electrophoresis 17
SDS	sodium dodecyl sulfate 62
SILAC	stable isotope labeling by amino acids in cell culture 34
SIL	stable isotope labeling 35
SMA	simple moving average 29
S/N	signal-to-noise ratio 73
SRM	selected reaction monitoring 25
TCEP	tris(2-carboxyethyl)phosphine 62
TDB	trypsin digestion buffer 62

TFA trifluoroacetic acid 63
TMT tandem mass tag 36
TP true positive 109
UHPLC ultra-high performance liquid chromatography 17
UPS1 universal proteomics standard 143
UPS2 universal proteomics dynamic range standard 61
UV ultraviolet 19
XIC extracted ion chromatogram 38

Part I

Introduction

1

The Concept of Tumor Heterogeneity and Cancer Stem Cells

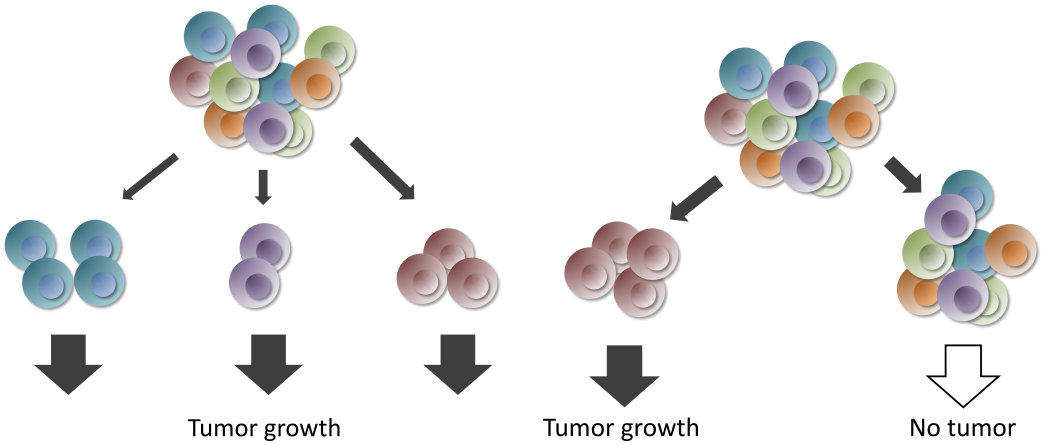
How do cancer cell populations change over time? Why does a tumor form metastases in distant organs and what is the reason for tumor relapse after treatment? These questions have tremendous impact on the general understanding of this disease and treatment strategies. In the last decade, different models have been developed addressing cancer cell heterogeneity and its development over time.

1.1 Models of Tumor Progression

Models describing tumor formation, progression and recurrence have been evolving over the years. Common to all of them is the underlying fact that cancer is a genetic disease and that tumor progression is driven by a sequence of genetic alterations, which eventually lead to uncontrolled and malignant cell proliferation [2, Chapter 2, Chapter 7], [3].

1.1.1 Stochastic model

The simplest and oldest concept is the *stochastic model*: All cells are equal and only chance determines which cells will be hit by the number of genetic and/or epigenetic changes required for malignant transformation (Figure 1.1a). These cells (re-)gain the ability for unlimited and rapid proliferation; the specific trait targeted by cytotoxic, anti-proliferative chemotherapeutics. The problem of frequent and severe tumor relapse has ever since highlighted the incompleteness of both the model and the available treatment procedures.



(a) All cells have the same tumor initiating capacity. Tumor cells themselves might be heterogeneous in many aspects, but all cells are equal in terms of tumor maintenance and metastasis initiation. According to this model, the genetic and epigenetic changes leading to tumor development and progression of malignancy are operative in all cells within the tumor. Existing therapeutics aiming at the tumor bulk are mainly based on this model.

(b) Only a subpopulation of the tumor bulk has tumor initiating capacity Tumor cells are heterogeneous, also with respect to tumor maintenance and metastasis initiation capacity. According to this model, certain cells are biologically and functionally distinct from the bulk of the tumor cells and must be specifically targeted by cancer treatments to achieve permanent cure.

Figure 1.1: Tumor heterogeneity. Adapted from [4].

1.1.2 Clonal evolution

The model of *clonal evolution* is a more advanced concept reflecting Darwin's ideas of evolution as a constant progress of adaptation and selection. While tumor initiation is still assumed to be a random event, tumor progression follows the rules of Darwinian evolution: A cell that gained a survival advantage by a random genetic/epigenetic change establishes a branch of dominant clones outperforming "normal", healthy cells in terms of proliferation. Over time, this branch accumulates more and more mutations increasing its abilities to survive and expand. This increasing "Darwinian fitness" can indicate tumor progression into a malignant cancer, which overcomes tissue barriers and eventually forms metastases at distant sites (Figure 1.2). Clonal evolution explains the observable steps of tumor progression, but fails to provide a thorough explanation for tumor relapse.

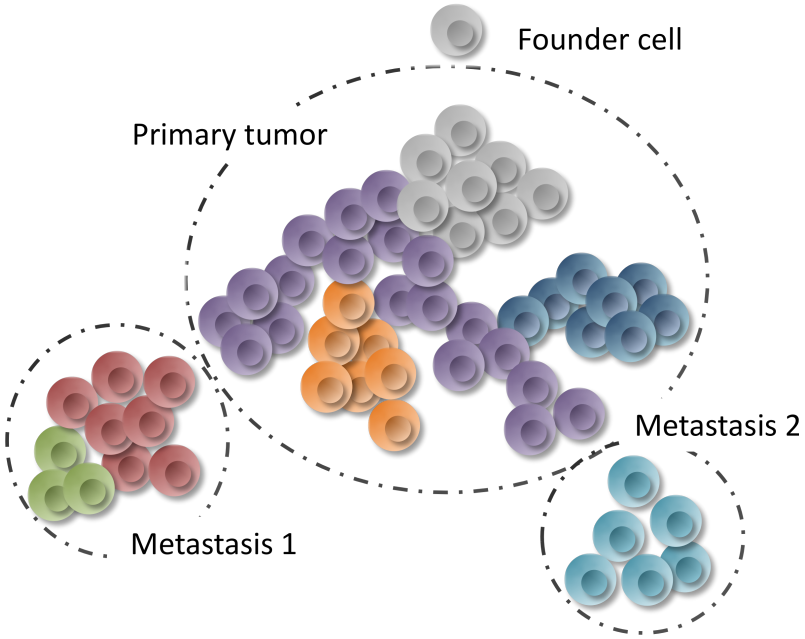


Figure 1.2: Clonal evolution as a model of tumor progression. Tumor initiation is a random event, but genetic and epigenetic changes establish clonal branches of tumor cells which outperform healthy cells in terms of proliferation. These branches follow the principles of Darwinian evolution and accumulate more and more "beneficial" mutations over time. Only cells from a dominant branch possess the ability to maintain the tumor and to establish metastases at distant sites. Adapted from [5].

1.1.3 Cancer stem cells

The term "cancer stem cell (CSC)" was first used in the early 1980s, when the idea of cancer as a disturbed function of stem cells arose [6, 7]. Since that time, the concept of CSCs has been developed further and was supported by the discovery of the heterogeneous nature of tumors. Within a certain tumor, many different kinds of tumor cells exist, and a tumor is therefore comprised of a highly heterogeneous and differentiated cell population both on the level of genetics and epigenetics [8–15] (see Figure 1.1). Initiated by research on the hematopoietic system and on leukemia, a concept established that a tumor is not only a collection of heterogeneous cells, but a cell population organized in a hierarchical manner [16–19]. Within this hierarchy, only very few cells have unlimited proliferative capacities and possess stem cell-like properties. These features are not only required for the maintenance and growth of the primary tumor cell population, but also for the initiation of metastases in distant sites. The vast majority of the tumor bulk, however, consists of differentiated cells that do not contribute to tumor maintenance or metastasis initiation. This model as well tightens the parallels of a tumor to a healthy organ, which is grown and maintained by a fine and granular regulation of self-renewal and differentiation[1] [4, 25, 47–52] (Figure 1.3).

1.1.4 Union of models

Since its emergence, the CSC model has evolved to cover a broader range of tumor characteristics, especially within the fields of epithelial-mesenchymal transition (EMT) and metastasis [42, 54, 55]. Step-by-step, all these models of tumor progression could be united into one concept:

[1] Leukemia was also the first cancer where CSCs could be successfully isolated [16, 20–23]. Characterization of CSCs in solid tumors followed later in the early 2000s, namely in cancer entities deriving from brain [24], breast [25], ovarian [26], prostate [27–29], colon [30–33], liver [34, 35], lung [36], pancreas [37], and others [38–40]. Until then, CSCs where solely defined as tumor cells that show an elevated rate of tumorgenesis after being transplanted into immunodeficient mice. The approach chosen for CSC validation led to the controversy whether the whole CSC hypothesis was only an artifact resulting from xenograft experiments [41–43]. Final proof was reached only after confirmation by complementary experiments using genetically modified mouse models and linage tracing of cancer cells [29, 44–46].

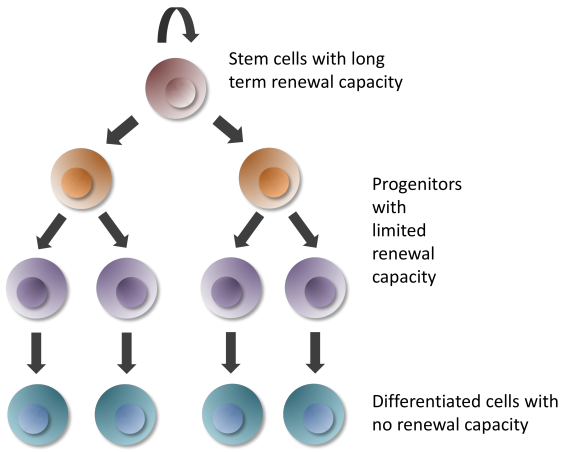
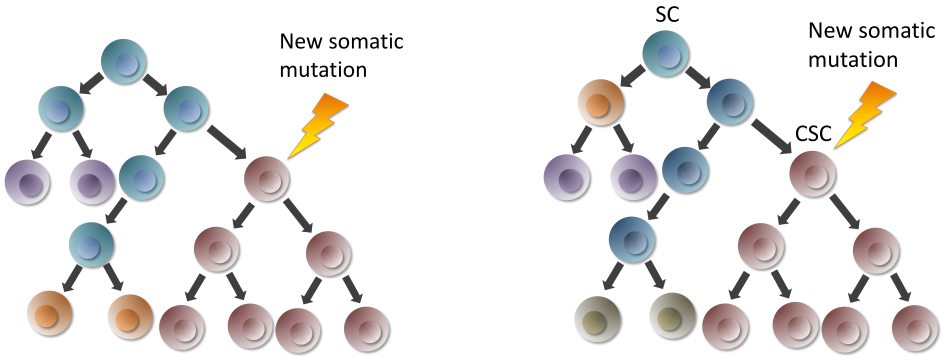


Figure 1.3: Hierarchical layout of a cell population. Cells are organized hierarchically, with the stem cells at the apex of the hierarchy. Stem cells possess long-term self-renewal capacity and are slowly proliferating. They give rise to progenitor cells, which proliferate more frequently and in turn give rise to fully differentiated tissue cells that might have only a limited life span. This hierarchy can be found in numerous healthy organs and has been proven to be present in many cancer tissues as well. Adapted from [53].

As the model of clonal evolution can be seen as an extension to the stochastic model, both can be united. Tumor initiation is a stochastic event hitting one single cell. Acquiring additional mutations is again a random event. Only if the initial mutation and additional changes hit the same cell, this cell may establish a dominant clone and start to form a tumor cell population. More and more random mutations in cells of this branch are needed to promote tumor progression. Importantly, multiple branches may exist at the same time and might be competing against each other. As new branches arise, some branches become extinct. All branches concurrently building the tumor cell population represent tumor heterogeneity and possess different levels of fitness against therapy and general survival not only in the current setting but also in new environments (Figure 1.4a). These concepts could also be integrated into the CSC model:

- Random mutations drive the clonal evolution of a tumor. Just like mutations can lead to an increase in function through the activation of specific genes, these mutations can lead to a gain in stem cell properties, e.g., by activating embryonic stem cell-associated genes like nanog, oct4, sox2 and c-myc [56]. A CSC does not necessarily derive unidirectionally from a somatic stem cell. Stem cell properties might be acquired by any progenitor or fully differentiated cell during tumor progression (tumorigenesis is a multi-step process [2, Chapter 11]). This theory stands in contrast to the idea that only stem cell (like) cells may be transformed into cancer stem cells and stand at the beginning of tumorigenesis [42] (see Figure 1.3 and Figure 1.4b).



(a) Union of stochastic model and clonal evolution. The stochastic model and the model of clonal evolution of tumor progression may be unified if each root of a dominant clonal branch is initiated by a random mutation event.

(b) Union of clonal evolution and the CSC model. The model of clonal evolution and the model of CSCs may be unified if each root of a dominant clonal branch is maintained by a biologically and functionally distinct population of tumor stem cells.

Figure 1.4: Union of models of tumor progression. Adapted from [53].

- Along with the initial transplantation assays, which were performed to prove the CSC model, the definition of metastasis initiating cells (MICs) was introduced and understood to be a subtype of "normal" CSCs. Importantly, both cell types are not necessarily mutually exclusive [57–59].
- The degree of differentiation versus stemness can be seen more as continuous and does not necessarily underlie a distinct and sharp separation of cell populations [60]. A tumor cell hierarchy might be more or less shallow and change over time (Figure 1.5).
- In close relation to EMT, stem cell properties can be acquired or lost. The transition from a "normal" tumor cell into a CSC is thought to be a bidirectional and reversible process and depends only on the plasticity of the cells [61–64]. This plasticity may be developed to different degrees and contribute strongly to both tumor heterogeneity and hierarchical layout (Figure 1.6).

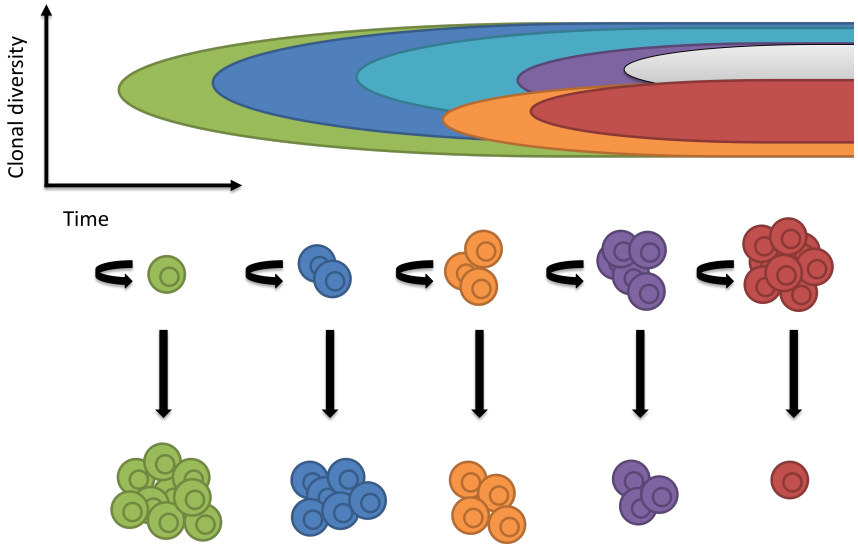


Figure 1.5: Stemness versus differentiation. The relative amount of CSCs within a tumor cell population is variable and might increase with tumor progression. The majority of cells from a late-state CML in blast crisis, for example, has acquired stemness properties and a clear hierarchy of cells is not present anymore [65]. Adapted from [55].

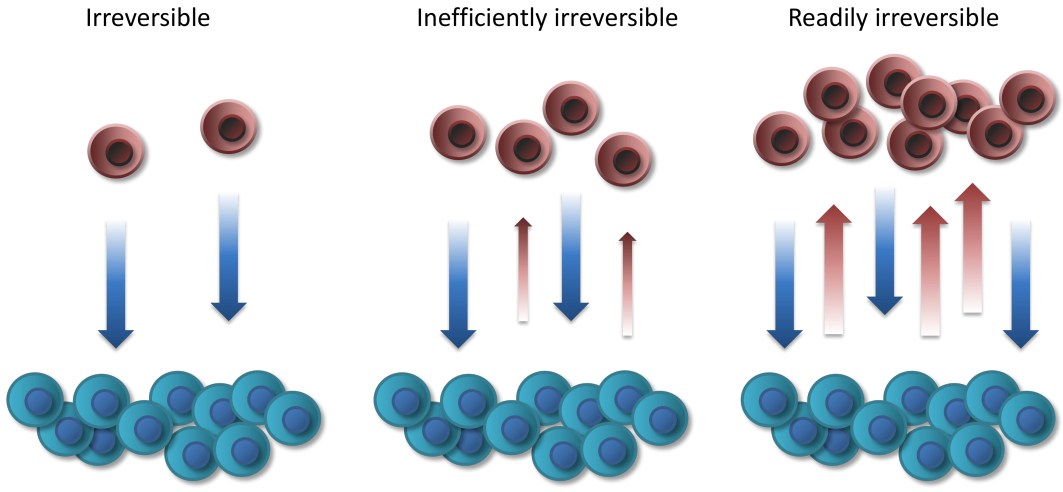


Figure 1.6: Bidirectional transition into CSCs. Cancer stemness is a reversible process and can be acquired also by progenitor or fully differentiated cells. The frequency of occurrence and reversion depends on cell plasticity. Adapted from [60, 62].

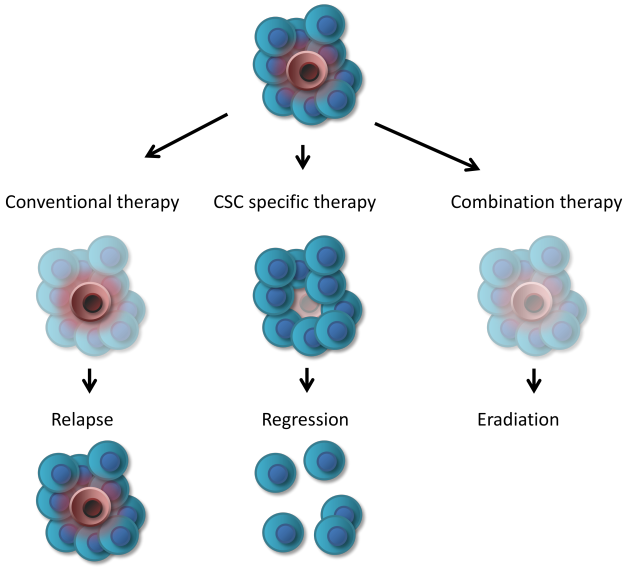


Figure 1.7: Cancer therapy and tumor relapse. A conventional cancer therapy might not be successful on the long run, since it fails to reach CSCs, which are responsible for tumor relapse. A targeted therapy approach, which selectively aims at the CSCs, might spare the majority of tumor cells but ultimately lead to tumor regression and prevent from a relapse due to the elimination of CSCs. Adapted from [66, 67].

1.2 Therapy Failure and Targeted Therapy

Just like normal adult stem cells, CSCs are defined by the typical functional stem cell properties, which are (i) long term self-renewal capacity, (ii) multipotency, and, most importantly in this context, (iii) the ability to reversibly enter a quiescent or dormant state [68].

This "hibernation" protects them efficiently against radiation and cytotoxic drugs used in almost all cancer therapies [61, 69–74]. Cytotoxic drugs aim at a property that is shared among all tumors – excessive cell proliferation. While these drugs can be successfully applied for tumor debulking and lead to impressive short term results, this approach fails to succeed on the long run and tumor relapse is often only a matter of time. Additionally, the patient not only faces the same illness again but often gets confronted with a much more aggressive form of the tumor showing a decreased responsiveness to treatment.

Failure of therapy and tumor relapse seem to be a direct consequence of dormant CSCs [56, 64, 75]. Consequently, first approaches to activate these cells to make them responsive to therapy have shown promising results and have laid the foundation to a more selective therapy, for which the term *targeted therapy* is used [76–81]. Therapies specifically directed against the "root of cancer" might result in much more durable responses and even a cure of metastatic cancer [74, 82] (see Figure 1.7).

2

Biomarkers

The identification of CSCs represents today's major challenge concerning the establishment of effective and widely applicable targeted therapies. Identifying CSC-specific properties allowing screening, sorting or otherwise targeting this subpopulation has proven to be exceedingly difficult [83]. These properties, or more generally speaking, any property, that can be used to distinguish CSCs from other cells is called a *biomarker*.

According to the National Institute of Health, a biomarker is defined as "a characteristic used to measure and evaluate objectively normal biological processes, pathogenic processes, or pharmacological responses to a therapeutic intervention". "Examples of biomarkers include everything from pulse and blood pressure through basic chemistries to more complex laboratory tests of blood and other tissues" [84–86]. In cancer research, mainly RNA, DNA or proteins serve as biomarkers.

A convenient biomarker needs to provide sufficient sensitivity and specificity to accurately distinguish its characteristics. Furthermore, it is necessary that the biomarker can be accessed and read easily in a non-invasive manner, for example in urine or blood. If the biomarker should not only be diagnostic but have a therapeutic value as well, for example via the application in targeted therapy, it needs to fulfill some additional requirements: In this case the biomarker needs to be a targetable molecule that is upregulated in the disease state and ideally not present in the healthy situation. Furthermore, it needs to be accessible via the vasculature [87–90].

Identifying biomarkers that do not only have diagnostic value but also provide accessible targets for therapy is a challenging but also promising task of today's cancer research.

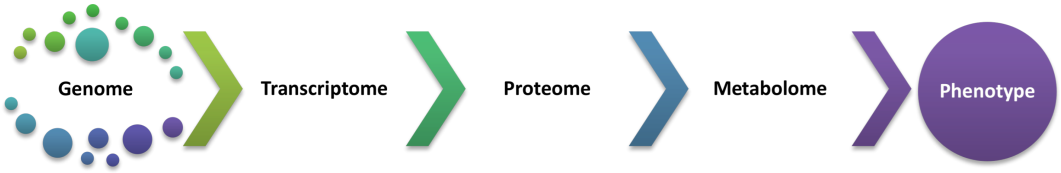


Figure 2.1: The -omics cascade. Four different entities mediate the expression of a phenotype: The genome provides the pool of all phenotypes that are possible to a cell. Only a small fraction of the genome is transcribed at a time. The trascriptome contains everything that is actually transcribed from genomic DNA into transcriptomic RNA. The Proteome is the collection of all proteins that are actually translated from mRNA. Proteins are the unions of action inside a cell. The metabolome represents the communication and cross talk within an organism. Metabolites are small chemical molecules that are produced by a machinery of proteins. More than all the other entities in the -omics cascade, the metabolome not only covers an isolated time point of action, but also connects the system in the dimension of time. Adapted from [91].

2.1 Biomarker Discovery

In the past decade, the rapid development of next-generation sequencing techniques has raised the search for biomarkers to another level. These sequencing techniques can be applied in a high-throughput fashion and their costs have decreased significantly in the last years. Nevertheless, the initial hope that large-scale whole genome and exome sequencing could complete our knowledge of the molecular mechanisms of cancer development and therefore help to discover cancer's weak points, has been frustrated over time.

It is obvious, that there is much more between an organism's genome and its phenotype. Differences on the genomic level are much smaller then we initially expected: The number of human genes is currently estimated to be approximately 23,000, "coming dangerously close to the numbers found for worms (20,000) and flies (14,000)" [92]. Furthermore, the process of translation is regulated to a variable degree and a linear correlation between the transcription of a gene and its occurrence as a protein is often not given [93–96]. This is due to intermediate processes such as post-transcriptional- and post-translational modifications that influence parameters such as RNA-stability or protein degradation. For instance, more than 60 % of the 23,000 putative open reading frames (ORFs) in the human genome encode more than one splice variant (often tens to hundreds), and these in turn are frequently subject to post-translational modification [97].

Whole genome and exome sequencing indeed allows for a complete view of copy number alterations, insertions and deletions. But these genomic changes are still implied to affect the proteome as well and indicate rather a potential than functional state of a cell or tissue [98, 99]. To be sure about the effects of a genetic transcriptomic alteration they need to be validated eventually on the proteomic level. Alternatively, biomarker discovery can be performed using proteomic techniques in the first place.

3

Liquid Chromatography Coupled to Mass Spectrometry – A Tool For Biomarker Discovery

The development of soft ionization techniques like matrix-assisted laser desorption/ionization (MALDI) and electrospray ionization (ESI) for biomolecules in the late 1980s, was the starting point for applying the technique of mass spectrometry (MS) to peptide and protein studies [100]. Since then, MS coupled to sample separation by liquid chromatography (LC) has developed to being the de-facto standard methodology for analyzing the proteome in general and cancer associated, proteomic changes in particular [100–103]. LC-MS is an especially suitable tool for biomarker discovery and is frequently used for this purpose in the field of proteomics.

3.1 LC-MS Workflow

Despite numerous technological variations of instruments and different vendor specific flavors of implementation, the general workflow, that is used in a typical LC-MS setup, remains the same and is descried in this chapter into more detail.

3.1.1 Sample preparation

In *bottom-up* or *shotgun* proteomics, proteins are first digested (usually enzymatically with trypsin) into peptides, which are then processed (see Figure 3.2). Information obtained from peptides need to be mapped back to the corresponding proteins in a second bioinformatics-based step.

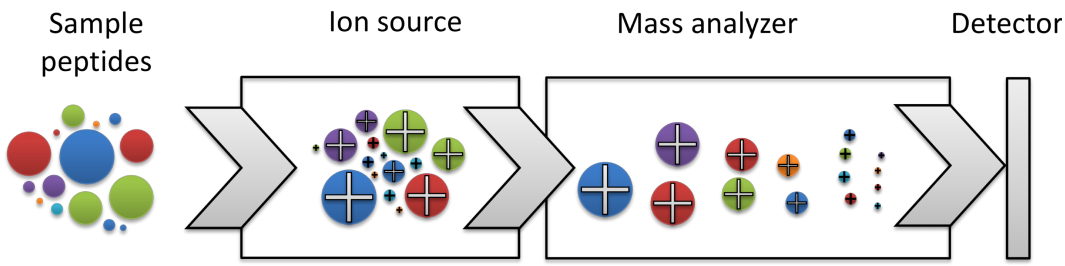
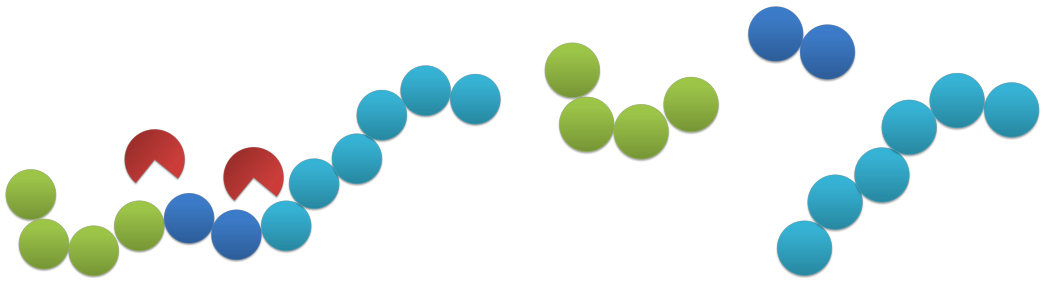


Figure 3.1: Functional principle of mass spectrometry. The functional principle of mass spectrometry is the separation of ions according to their molecular weight. All molecules to be analyzed must therefore be ionized first. Ion generation is performed by an ion source (see 3.1.3). After ionization, ions are separated according to their mass, more specifically, according to their mass-to-charge ratio (molecules may carry multiple charges) (see 3.1.4). Data acquisition is performed by the ion detector, which translates detected ions into two dimensional data points (arbitrary intensity value & mass-to-charge ratio). Adapted from [104].



(a) A(n intact) protein and cleavage enzymes.

(b) Peptides after digest by cleavage enzymes such as trypsin or caspase.

Figure 3.2: Tryptic digest of proteins. Proteins are composed of amino acids that chain together. Upon a tryptic digest, they are cleaved into separate peptide sequences (shorter amino acid chains) at specific cleavage sites. Amino acids are in green, blue and cyan, enzymes that cleave the protein in red (e.g. trypsin).

This is in contrast to *top-down* proteomics, where intact proteins are processed. Complete proteins often display a similar mass-to-charge ratio and are more difficult to handle than smaller peptides. Peptides, on the other hand, are more soluble, have similar physico-chemical properties, and can be more easily sequenced, which is why mainly the bottom-up approach is used in today's LC-MS workflows.

3.1.2 Chromatographic separation

Since even a relatively simple proteome (e.g. the one of *E.coli*) is still far too complex to be processed directly by MS, sample complexity is reduced by chromatographic separation prior to mass spectrometric analysis. Chromatographic separation is performed by ultra-high performance liquid chromatography (UHPLC) systems, which may be coupled directly (*online*) to the mass spectrometer or may operate as a separate unit of action (*offline* coupling) [105]. Whether the UHPLC system is coupled online or offline is specified by the ionization technique applied (see 3.1.3). In an online setup, the sample is directly and continuously introduced into the mass spectrometer and the ion source. An offline setup contains an intermediate sample fractionation step, where the sample is separated into different *fractions* and subsequently introduced into the ion source. LC uses a chromatographic column to separate peptides, based on one or more chemical or physical properties. In a state-of-the-art UHPLC system, a so-called reverse phase column is used to separate the peptide mix [106]: The column contains stationary, hydrophobic elements, which retard the migrating peptides differentially according to their hydrophobicity [107–110].

Before the application of LC, the identification of proteins from complex biological matrixes has been performed using two-dimensional gel electrophoresis (2D-GE) [111, 112]. 2D-GE separates proteins by both their isoelectric point (pI) and molecular weight. In this "divide-and-conquer" strategy, proteins are resolved into discrete spots that can then be selectively excised and analyzed [98, 112]. Today, the separation capacities of gels are sometimes still utilized as an extension to an LC-MS system. For example, sodium dodecyl sulfate polyacrylamide gel electrophoresis (SDS-PAGE), which separates proteins according to their electrophoretic mobility, can be used in combination with LC as an additional separation step [113]. In this scenario, a biological sample is first separated using SDS-PAGE; the resulting bands are then processed separately on the LC-MS workflow. A workflow, that consists of a separation by gel followed by a chromatographic separation followed by mass spectrometry is often abbreviated using *SDS-PAGE followed by in-gel digestion followed by LC (GeLC)-MS* [107–110].

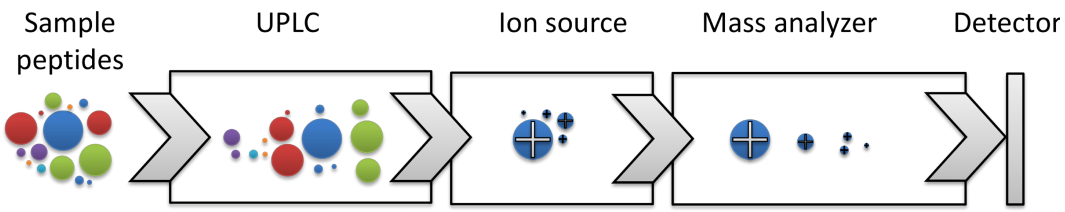


Figure 3.3: Functional principle of liquid chromatography coupled to mass spectrometry. Chromatographic separation is introduced as an additional sample separation step in order to reduce sample complexity. Peptides are separated based on one or more chemical or physical properties, such as hydrophobicity, polarity, affinity to other molecules, or size of the sample peptides. In case of an UHPLC system, peptides are separated according to their hydrophobicity (indicated by different colors). Adapted from [104].

3.1.3 Ionization

To date, there are two techniques available for soft ionization of biomolecules that are used in state-of-the-art LC-MS systems: electrospray ionization and matrix-assisted laser desorption/ionization.

3.1.3.1 Electrospray ionization

The sample is dissolved in an organic, aqueous solvent containing formic acid and is guided through a thin capillary. When applying a high voltage to this capillary, the eluting sample solvent forms an aerosol of charged droplets when exiting the capillary (the *Taylor Cone*) [114, 115]. The aerosol is *sprayed* into a vacuum chamber, where the liquid vaporizes, causing the droplets that contain the sample molecules to become smaller and smaller. Due to the *Rayleigh Limit*, at some point, the droplets "explode" into smaller ones [116, 117]. This process continues until the molecules themselves carry all the charge that was introduced before without any surrounding solvent (see Figure 3.4a). In the course of the ionization process, ions with multiple charge states are generated [118–120]. ESI sources require a constant sample flow, which is why they are coupled to the mass spectrometer online (see 3.1.2).[1]

[1]Summarized history of ESI: • 1882: Rayleigh calculates the maximum theoretical charge one liquid droplet could carry, the so-called *Rayleigh Limit* [114] • 1914-1964: Zeleny, Wilson and Taylor publish work on the electrospray *Taylor Cone*. [121–124] • 1968: First use of ESI as an ion source in a mass spectrometer [125] • 2002: John B. Fenn is awarded the Nobel Prize for chemistry for his work on ESI-mass spectrometers [119].

3.1.3.2 Matrix-assisted laser desorption/ionization

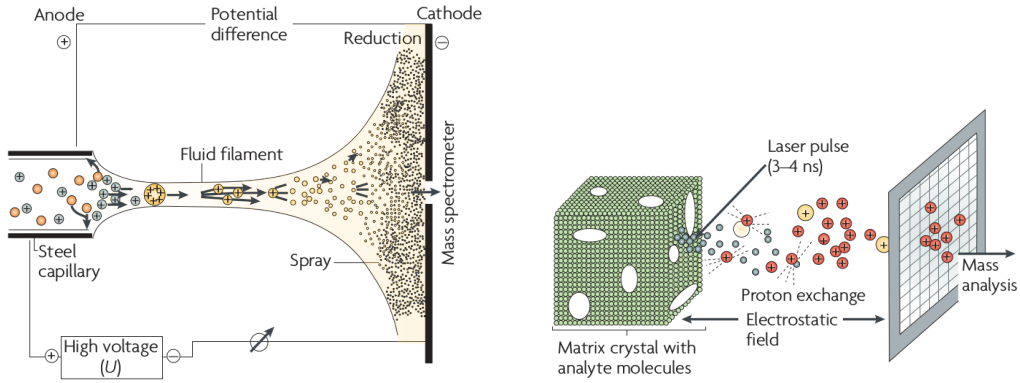
The sample is prepared by mixing it with an excess of *matrix* molecules. The resulting sample-matrix mixture is spotted on metal plates (depending on the instrument, 2,400 or more spots per plate can be applied routinely). After spotting, the buffer evaporates and the matrix molecules crystallize, forming a grid in which the sample molecules are embedded. The matrix molecules have a relatively low molecular weight and provide a strong optical absorption in either the ultraviolet (UV) or infrared (IR) range. These properties cause them to absorb optical energy when being shelled with a laser. Matrix molecules get ionized and burst out of the matrix-grid, together with co-crystallized sample molecules (see Figure 3.4b) [126]. The nature of charge propagation from matrix to sample molecules is still a matter of debate [127–131]. Ions resulting from MALDI are predominantly singly charged, which simplifies later data processing (see 3.5). MALDI ion sources are coupled offline to a mass spectrometer, since a sample-spotting step is needed between LC and MS. Furthermore, the matrix is in a solid state and the laser does not fire in a constant but rather in a pulsed mode.[2]

3.1.4 Ion sorting

The *mass analyzer* is the component of the mass spectrometer, which "sorts" the ions according to their mass-to-charge ratio (m/z). This sorting is necessary, since the ion-detector (see 3.1.5) cannot provide any information on the mass of the impacting ion. Its output is an arbitrary intensity value directly correlating with the abundance of the detected ion(s). Therefore, it is necessary to introduce only ions with the same m/z at a defined point into the ion-detector.

A mass analyzer's performance is typically measured using the following five different metrics:

[2]Summarized history of MALDI: • 1985-1987: Michael Karas and colleagues coin the term "MALDI" for matrix assisted soft ionization [132, 133] • 1988: Breakthrough of matrix assisted ionization due to the combination of a liquid matrix, a suitable laser wave length and a thin metal plate [134] • 1988-1990: Major advances of laser technology and optimization of wave lengths finally establish MALDI as an ion source of choice when applying mass spectrometric analysis of biomolecules [135–137]. • 2002: Tanaka is awarded the Nobel Prize for chemistry for this work, together with John B. Fenn, the pioneer of ESI. • Two things noteworthy: (1) The initial publication of idea and concept was not even five years apart from usage as an expedient for analysis of large biomolecules. (2) The principle of ionization and how exactly the three components analyte, matrix and metal plate are interacting, is still not clarified and still matter of debate.



(a) Electrospray ionization. Two main physical-chemical mechanisms enable the ionization: (i) The Taylor Cone, which is formed at the exit of a capillary when the sample buffer exits and a high voltage is applied to the capillary. (ii) The Rayleigh Limit, which causes the charge to be propagated from buffer to sample molecules in a soft manner (soft ionization).

(b) Matrix-assisted laser desorption/ ionization. MALDI uses a "helper matrix" to charge sample molecules softly: Matrix molecules take up optical energy when laser shelling is applied and get charged. Charge is propagated to sample molecules, which are embedded into a grid of crystallized matrix molecules, when high energy causes the matrix to break apart.

Figure 3.4: LC-MS ion sources. Reprinted with permission from [138].

Mass range: the minimal and maximal mass-to-charge ratios that can be measured.

Mass accuracy: the difference between actual m/z and measured m/z, typically expressed in parts per million (ppm).

Resolving power: the ability to separate peaks clearly from each other.

Transmission: the ratio between the number of ions reaching the detector and the number of ions entering the analyzer.

Scan speed: the time required to complete one m/z analysis.

Common to all mass analyzers is the use of electric and/-or magnetic fields to control ions in space and time. Mass analyzers can be grouped into discrete groups using different properties, such as continuous versus pulsed or ion beam versus ion trapping [105]. These characteristics may limit combinations of mass analyzers with ion sources.

	Time-of-flight	Time-of-flight (reflecton)	Quadrupole	Quadrupole ion trap
Principle of separation	Velocity (flight time)	Velocity (flight time)	m/z (trajectory stability)	m/z (resonance frequency)
Mass limit	1,000,000 Th	10,000 Th	4,000 Th	6,000 Th
Accuracy	200 ppm	10 ppm	100 ppm	100 ppm
Resolution	5,000	20,000	2,000	4,000
Ion sampling	Pulsed	Pulsed	Continuous	Pulsed

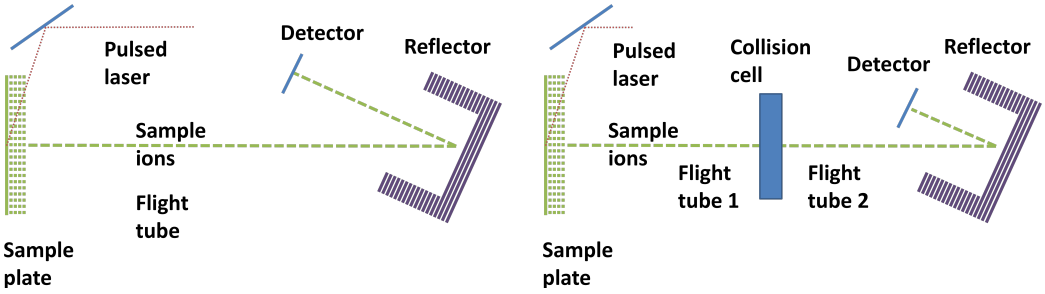
Table 3.1: Different types of mass analyzers used in mass spectrometry and some of their properties. Adapted from [105].

Hybrid instruments, that combine different mass analyzers in one instrument for better performance or advanced acquisition modes grow in popularity. The triple quadrupole (QQQ) ion trap-MS, for example, combines three quadruple analyzers, whereof the third can be switched into an ion trap, enabling data-independent acquisition (DIA) methods and new targeted quantification approaches (see 3.3).

Today's most prominent mass analyzers can be categorized into tree different types:

3.1.4.1 Time-of-flight

Assuming an equal mass and acceleration energy, all ions need the same time to fly through a defined field-free distance in space. The time an ion needs to travel is directly proportional to its mass-to-charge ratio. Heavier ions will travel more slowly than light ions and will take more time to complete the full distance. An equal acceleration energy and a defined start point for all ions are crucial requirements. More advanced instruments use an extended flight tube, which reflects ions at the end of the tube using a so-called *reflectron*. The reflectron compensates for variations in starting points and acceleration energy [139–142] (see Figure 3.5).



(a) Time-of-flight (TOF) mass analyzer.

(b) Time-of-flight-time-of-flight (TOF-TOF) mass analyzer.

Figure 3.5: Illustration of a TOF mass analyzer on the example of MALDI-TOF. Sample- and matrix molecules are spotted together on a steel plate, which is shelled by a pulsed laser. Ionized molecules traveling through a flight tube are reflected by a reflector/reflectron and hit the ion detector. In case of TOF-TOF, the flight tube is separated into two parts, divided by a collision cell for MS^2 analysis. Adapted from [103].

3.1.4.2 Quadrupole

A quadrupole consists of four cylindrical rods. The two opposing rods build pairing electrodes, which create an electric field through which ions travel. The field changes its potential frequently, causing ions to oscillate between the rods. For ions traveling on stable trajectories, kinetic energy is converted into electric potential and vice versa in line with changing potentials. When changing field frequency and/or applied voltages, only ions with a certain mass-to-charge ratio are able to maintain stable trajectories, whereas all other ions will be deflected from their course to the ion detector. [143–145].

3.1.4.3 Ion traps

3.1.4.3.1 Linear ion trap (2D ion trap) A linear ion trap (LIT) works in principle like a quadrupole. The main difference, nevertheless, is that ions are reflected at both ends of the rods by electric stopping potentials applied to end electrodes forcing the ions to travel forwards and backwards between these electrodes. Historically, LITs were developed later than Paul traps (see next paragraph) and possess some advantages, such as larger ion trapping capacity and higher trapping efficiency [144, 145].

3.1.4.3.2 Paul ion trap (3D ion trap) The Paul ion trap was the first mass analyzer that utilized the principle of quadrupole electrodes which establish electric fields for ion control. The working principle is similar to the quadrupole analyzer described above, with the main difference that the quadrupolar rods are bent to form circular electrodes. On top and bottom, additional cap-formed rods are placed in order to force electrons on 3D trajectories between the electric fields. The electric potentials of the electrodes can be adjusted to selectively expel ions with certain m/z values. This is in contrast to a LIT and a quadrupole, where the principle works the other way around, i.e, adjusted electric potentials are used to force ions on unstable trajectories between the rods, thereby filtering ions of a specific m/z [144, 146, 147].

3.1.4.3.3 Orbitrap An orbitrap mass analyzer is built from a total of three electrodes. The first one is spindle-shaped, surrounded by the other two electrodes, which together have the shape of a barrel, separated in the middle. Direct current (DC) voltages are applied forcing ions on trajectories around the spindle, oscillating back and forth the spindle axis. The ions' m/z values can be deducted from oscillation frequencies using Fourier transformation (FT) [148–150].

3.1.5 Ion detection

As briefly mentioned before, an ion-detector produces an analog, electrical signal directly correlating to the abundance of the detected ions (see 3.1.4). The signal is more or less continuous, depending on the scan speed of the detector. An analog-to-digital converter (ADC) is needed to convert the analog signal into a digital data stream, which in turn can be processed computationally. This is achieved by binning of the constant, analogous data stream. Every bin (usually in the range of nanoseconds) is converted into one digital data point [151–154].

3.2 Tandem Mass Spectrometry

Mass spectra as described before do not contain sufficient data to unambiguously identify the analyzed ions/peptides. Although the amino acid composition of a given peptide can be determined by certain high-resolution/highly accurate mass spectrometric methodologies (e.g., Fourier transformation-ion cyclotron resonance-MS [148, 155, 156]), it is impossible to determine the sequence order of the amino acids. Tandem mass spectrometry (MS/MS or MS^2) can be used to solve this problem. As the name suggests, tandem MS describes the application of multiple MS analyses separated either in time or space [143, 157, 158]. The latter involves two or more distinct mass analyzer and detectors that are coupled to each other. In contrast, tandem-in-time describes multiple MS analyses performed after each other in the same instrument. In the case of tandem-in-time and for certain instruments, an unlimited number of subsequent MS analyses can be performed (MS^n). Today, most proteomic workflows apply MS^2, since such spectra are sufficient for the identification of peptides. The advantage of multiple-, subsequent MS analyses can be compared to LC and MS: One or more preceding MS analyses can be utilized as another separation or selection step before a final MS scan is performed to identify ions of interest.

In the case of MS^2, the first MS analysis records a spectrum that contains all detected ions. The subsequent MS^2 analysis is performed for a selected species of ions (*parent ion* or *precursor ion*) to elucidate the species' identity, i.e., the corresponding peptide amino acid sequence. To record an MS^2 spectrum, a precursor ion is dissociated into smaller fragments by collision with inert gas molecules such as nitrogen or argon (collision-induced dissociation (CID)). MS^2 ion fragments are subsequently analyzed in the same way as the parent ions in MS^1 mode. As a result, peptides/peaks in an MS^1 spectrum obtain a MS^2 spectrum, which can be used to determine the peptide's specific amino acid sequence. For this purpose several algorithms have been implemented, such as Sequest [102], Mascot [159], X!Tandem [160], Paragon [161] or Andromeda [162, 163].

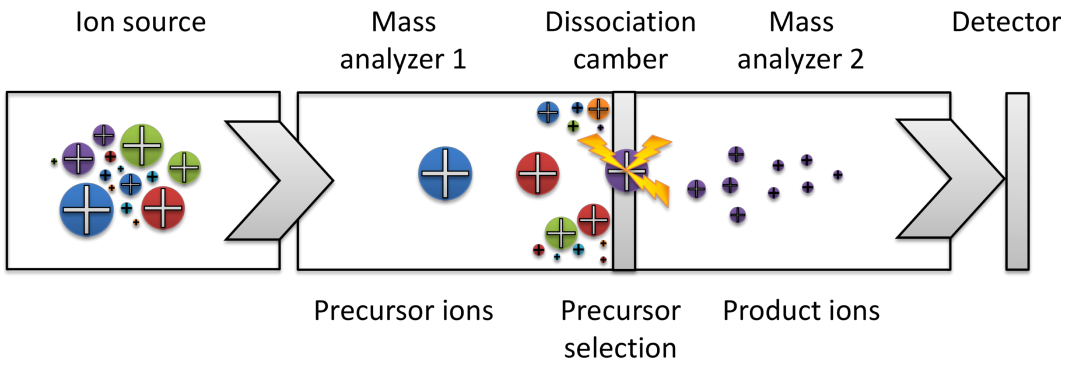


Figure 3.6: Tandem mass spectrometry. In tandem mass spectrometry (MS^2), ions with the mass-to-charge ratio of interest (parent or precursor ion) are selectively reacted to generate a mass spectrum of product ions.

Not every peak in a MS^1 spectrum is suited for subsequent MS^2 analysis. Typically, the sample amount (MALDI) or time (ESI) are limiting factors for MS^2 analyses, which is why only a subset of MS^1 species are selected for MS^2. Precursor selection is typically performed using either a *strongest first* or *weakest first* strategy: MS^1 peaks are ordered by their intensity value and one after another is selected for MS^2, starting with first element in the list or the last one, respectively. Nevertheless, advanced software tools and improvements in instruments' cycling time allow today for workflows omitting precursor selection (i.e., MS^2 analyses are performed on all ions). This approach is termed all-ion fragmentation (AIF) [164] and, since there is no precursor selection involved, it is counted among the DIA methods (see 3.3). The selection of a precursor ion for fragmentation and the subsequent MS analysis of the product ions is termed *product ion scan* [165]. The inverted scan mode is termed *precursor ion scan*, and describes the selection of product ions for the identification of precursor ions. In contrast to product ion scans, which find application in proteomics, this scan mode is only available in tandem-in-space instruments.

Another MS^2 scan mode that is becoming increasingly popular is termed selected reaction monitoring (SRM). SRM is a DIA method and describes a "scan" mode, which selects for certain pairs of precursor and product ions (so-called *transitions*). SRM finds application in the field of targeted proteomics and is realized in a QQQ instrument, where the first quadrupole selects for a certain precursor, the second one acts as a collision cell for fragmentation, and the third one selects for certain product ions. A scanning is not involved [165]. This setup provides outstanding sensitivity and allows for quantitative measurements.

3.3 Data-Independent and Targeted MS Techniques

The previously described MS^2 workflow is termed data-dependent acquisition (DDA), since MS^2 spectra are acquired based on previously generated (MS^1) data (see 3.2). Today's proteomic workflows mainly use a DDA approach, which faces some main issues rendering protein identification and quantification irreproducible: (i) under sampling [112, 166], (ii) stochastic precursor selection that is biased toward high-abundant peptides [164, 167, 168] and, for tandem-in-time, (iii) too long cycle times of the instruments [112, 164, 168].

DIA strategies try to overcome these limitations and can be divided into two main categories: (i) In a hypothesis-driven, targeted approach, the instrument is configured to fragment certain precursor ions independent of a prior MS^1 scan. (ii) Alternatively, generally all ions undergo fragmentation in parallel regardless of their intensities or any other characteristics. This results in a complete, unbiased data record with highly improved reproducibility and quantification capabilities [168, 169].

Precursor selection as applied in DDA approaches works by filtering all MS^1 ions by the precursor ion's m/z. Only ions matching the pre-defined m/z are selected for MS^2 analysis. In practice, this filtering utilizes certain m/z ranges, which are narrow enough to selectively filter for the desired precursor ion. In a DIA approach, these m/z ranges are broadened to cover a few or several dalton. The filtering is repeated using different m/z ranges until a "complete" m/z range (usually in the range of 400 – 1,200) is covered (PAcIFIC [170, 171] and SWATH-MS [172]). MS^E [173] and AIF [164] use a wide band pass filtering. Since wider m/z windows make subsequent peptide identification challenging or even impossible for modified forms of a peptide, these m/z windows can be first separated into smaller m/z windows, which are then multiplexed, analyzed, and demultiplexed. "Sub-windows" are chosen randomly from a set of 100 possible non-overlapping windows that cover the "complete" m/z range [169].

3.3.1 Selected reaction monitoring

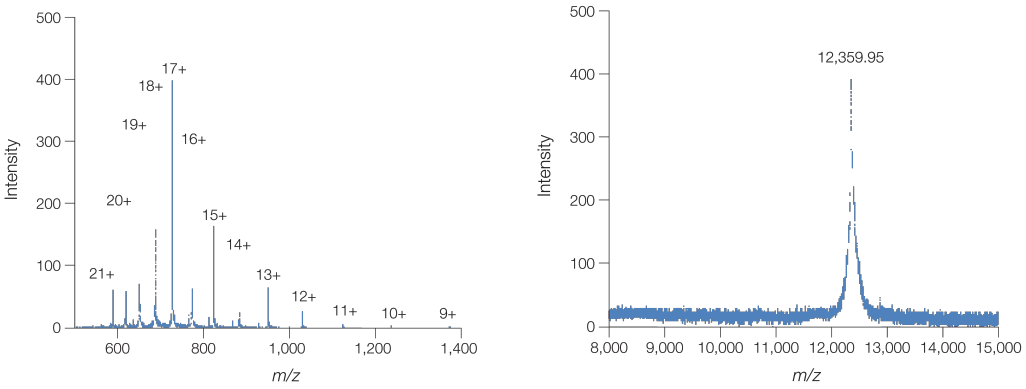
As briefly outlined before, one targeted DIA-MS technique is SRM [174, 175]. SRM is realized using QQQ instruments running with a specific configuration: The first and third quadrupole is configured to filter for a certain m/z, while the second quadrupole serves as a CID chamber. This way, the instrument is set up to detect only certain precursor-product ion combinations (transitions). It is a highly selective, hypothesis driven approach, but it provides excellent sensitivity since no scanning is involved. Furthermore, since all transitions are detected, the problem of undersampling is avoided and peptides can be quantified label-free by counting the number of occurring transitions (see also 4.2). For a relative quantification, the usage of stable isotope labeled (see also 4.1) internal standards [176] is often performed. SRM represents the most accurate and sensitive quantification method that is available today in the field of proteomics.

3.4 The Nature of LC-MS Data

Depending on the scan speed of the ion detector and the sampling frequency of the ADC, the amount of raw data that is produced by a standard MS analysis can be enormous. High-level software for the analysis of LC-MS data expects the data to be in the form of three-dimensional data points: (i) signal intensity, (ii) mass-to-charge ratio (m/z) and (iii) retention time. A considerable amount of preprocessing of the raw data is required in order to obtain this layout. As outlined before, the signal of the ion detector has only one dimension, namely the ion intensity (see 3.1.5). To obtain the corresponding mass(-to-charge ratio), information from the mass analyzer needs to be incorporated (see 3.1.4). The *retention time* is originated from the chromatographic separation by the LC and must not be confused with the just mentioned time-wise differences in signal detection.

To summarize:

1. Data obtained by the mass spectrometer:
   (a) Signal intensity: Directly originated from the ion-detector.
   (b) Mass-to-charge ratio (m/z): Computed by data both from the ion detector and the mass analyzer.
2. Data obtained by LC is the retention time, i.e., the time point of detection of an intensity-m/z pair or the elution time of an analyte molecule.



(a) ESI spectrum of cytochrome c. Multiple peaks are observed due to the different charge states.

(b) MALDI spectrum of cytochrome c. Only a single peak is observed for the analyte because ionization by MALDI generally produces singly charged ions.

Figure 3.7: A comparison of the mass spectra for cytochrome c generated using ESI and MALDI. Note the different m/z ranges in the spectra. Multiple charged ions (ESI) result in smaller m/z values compared to single charged ions (MALDI). Reprinted with permission from [143].

These three parameters join to a so-called *peak*. Displaying the intensity of a collection of peaks as a function of their m/z ratio results in the typical mass spectrum (see Figure 3.7).

3.5 Low-level Data Processing

Several pre-processing steps are performed to make the data manageable for high-level analysis software. These steps aim at a first "simplification" of the data by (i) extracting meaningful information (peak extraction), (ii) removing systematic noise and background (baseline subtraction) and by (iii) normalizing identified peaks to a monoisotopic and "mass-only" representation. The order of execution of the different processing steps may vary and, depending on the system that is used, additional steps might be required.

3.5.1 Baseline subtraction

Recorded raw spectra do not only contain the desired true signal resulting from the analyte ions but also a relatively constant baseline signal and a certain amount of noise. Noise in MS spectra can derive from sample preparation, the LC or some components of the mass spectrometer [177] (see Figure 3.7b). To separate true signals from noise and background, both basic and sophisticated filters are applied to the raw signal, for example, the Top-hat filter, Savitzky-Golay filter, or simple moving average (SMA) filters [178].

3.5.2 Peak extraction

Peak extraction (also referred to as peak finding, peak detection or peak picking) is the process during which the signals are separated from background and noise. Therefore, this processing step is strongly connected to baseline subtraction and noise filtering. Furthermore, peak extraction includes a certain level of data reduction, since only strong peak signatures are extracted from the raw signal (see Figure 3.7b). The extracted peaks assumed to represent the analyte ions of interest, which are, in the field of bottom-up proteomics, peptides resulting from a proteolytic digest of proteins [179–183].

3.5.3 Charge deconvolution

Spectra obtained by ESI-based MS often contain ions carrying multiple (different) charges (see 3.1.3 and Figure 3.7a). In general, the number of charges on a molecule ionized by electrospray depends both on its molecular weight [184, 185] and on the number of potential charge sites available [186]. Initially, when inspecting spectra, a peak's charge state is unknown and therefore also the mass of the corresponding molecule. Nevertheless, both can be obtained using computer-assisted charge deconvolution. Provided with key parameters, such as expected range of charge numbers and resolving power of the instrument, both the charge of an ion and therefore also its molecular weight can be calculated [185, 187–191].

3.5.4 Deisotoping

When inspecting naturally occurring molecules (such as tryptic peptides) on an atomic level, the presence of carbon isotopes and isotopes of other atoms will be observed. Approximately 1% of the naturally occurring carbon is found with an additional neutron (totaling seven instead of six) [182]. Since a typical peptide's mass is mostly made up of carbon, spectra derived from carbon isotopes are observed frequently. Although isotopes are chemically identical, the heavier isotope sister ion peaks exhibit greater apparent m/z than the predominant monoisotopic peak. Identifying these different isotope peaks and correcting their m/z ratio to a monoisotopic m/z ratio is called deisotoping or *peak centroiding*.

As described above for the two processing steps baseline subtraction and peak extraction, deisotoping and charge deconvolution are interconnected and algorithms performing multiple of these steps at once have been developed [189, 191, 192].

4

Peptide and Protein Quantification from LC-MS Data

For the identification of novel, vascular accessible biomarkers, quantitative comparisons between different biological states, such as healthy versus diseased, are a crucial requirement, since biomarkers suited as therapeutic targets need to be upregulated compared to the healthy state (see 2). A qualitative analysis of the proteome is therefore not sufficient but different proteomic states need to be compared quantitatively.

Even though an ion detector generates a signal within a certain dynamic range for which the intensity is directly proportional to the ion abundance [193] (see 3.1.5), MS is not inherently quantitative:

- Peak intensities deriving from different ions/peptides do not correlate to absolute ion/peptide abundance, since peptides exhibit a wide range of physiochemical properties such as size, charge, and hydrophobicity, leading to differences in the resulting signal intensity.
- Ion suppression effects frequently occur in complex peptide mixtures with a large dynamic range of ion abundance and hinder reproducibility of signal intensities leading to a underestimation of the respective protein abundance.
- As already outlined in 3.2, not all ions recorded in MS^1 can subsequently be passed on to identification by MS^2 and precursors are rather selected stochastically.

Nevertheless, a relative comparison is possible in principle, which is why MS is called a *semi-quantitative* technique. For biomarker characterization, usually absolute amounts are not needed and relative changes (e.g. compared to a healthy control) are sufficient.

In order to obtain quantitative data from LC-MS measurements, in principle, two approaches differing in many aspects, such as required workload, costs and limitation in sample number, are available. The first one makes use of artificial labels or tags, which are introduced into the samples at different points in the sample preparation workflow (*label-based* quantification), the other one does not require any modifications to the samples themselves but operates solely on the acquired LC-MS data (*label-free* quantification).

In general, it can be said that a label-based quantification requires additional processing during sample preparation, whereas label-free quantification takes an increased effort in bioinformatics.

4.1 Label-based Peptide Quantification

The idea behind label-based quantification is the introduction of artificial labels or tags into proteins and peptides during sample preparation. If appropriate labels are used, these molecular modifications (introduced into each sample) differ in their molecular weight but not in their physicochemical properties. The retention time of differentially labeled peptides therefore remains unaffected, while differences in their molecular weight are detectable by the mass spectrometer.

After successful label incorporation, samples are pooled and further processed together (*multiplexing*). Differentially regulated peptides will co-elute and will be recorded by the mass spectrometer at the same time. The resulting spectra contain signal intensities of all differently labeled peptides, which can be directly compared so that a relative quantification of the differentially labeled peptides can be performed.

Early introduction of the labels during the sample preparation process is essential to minimize possible sources of inter sample variation.

A significant number of different labels and labeling techniques have been introduced in the recent years and some of them have been extended over time to more advanced versions. Today, the incorporation of multiple labels gives the possibility to compare up to six [201], eight [202], and even twelve [203] different states in a single run. Labeling methods can be categorized using different characteristics:

- The way in which labels are introduced into peptides, which can happen (i) metabolically, (ii) chemically or (iii) enzymatically.

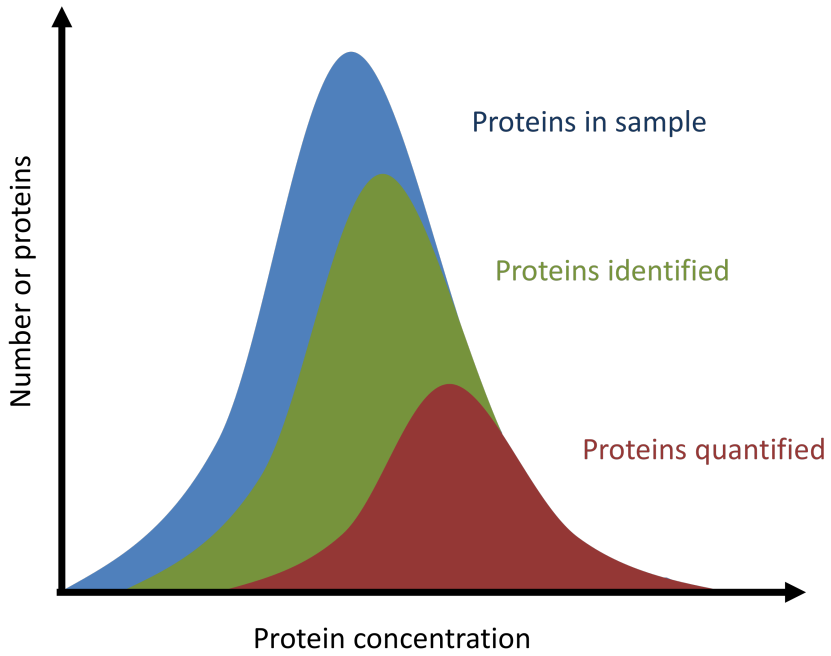


Figure 4.1: Fraction of quantified proteins in a sample. Due to mainly technical limitations, both caused by the LC-MS analysis itself (see main text) and by quantification methods, the number of proteins that can be quantified per single LC-MS analysis is limited and represents only a small fraction of the total proteome [194, 195]. Even though it has been shown that a nearly complete coverage of the proteome is possible for rather simple organisms, the required workload is tremendous and stands in no relation to the benefit [196–198]. For more complex proteomes, the rate of identified proteins usually lies below 10%. The number of proteins that are not only identified but also quantified is usually significantly smaller [112]. State-of-the-art LC-MS systems are capable of identifying and quantifying up to 10,000 proteins from a given proteome [199, 200]. Note that the number of proteins present in the proteome of an organism usually exceeds the number of genes by far, due to intermediate processes such as post-transcriptional- and post-translational modifications, which influence parameters such as RNA-stability or protein degradation (see 2). Adapted from [112].

- At what stage of MS data acquisition the label is recognized, which can be either (i) in the MS^1 or (ii) in the MS^2 (*isobaric* labeling) spectrum.

The most prominent types of labels used today are stable isotopes (e.g., ^{12}C vs. ^{13}C, ^{14}N vs. ^{15}N, ^{16}O vs. ^{18}O), which are ideal labels, since they do not alter the physiochemical behavior of modified peptides and produce a distinct mass shift in the MS^1 spectrum. Below, some of the most often used labeling techniques are described in detail.

4.1.1 Metabolic labeling

Metabolic labeling was first introduced for total labeling of bacteria using a ^{15}N-enriched cell culture medium [204]. Today, the most widely applied form of metabolic labeling is the stable isotope labeling by amino acids in cell culture (SILAC) described by Ong *et al.* in 2002 [205]. Cell cultures are grown in media containing either a light- or a heavy version of an amino acid, which is metabolically incorporated by the cells into all proteins. In principle, different isotopes can be used for the SILAC approach. Ong and his colleagues used deuterium as a label, while today's most commonly used implementations of the SILAC approach use ^{13}C and/-or ^{15}N isotopes for labeling. Usually, labeling is performed on the two amino acids arginine and lysine.

The metabolic incorporation of stable isotopes is not limited to single cells but was extended to bacteria [206], simple multicellular organisms [207] and even complex organisms like rodents [208]. Nevertheless, its main disadvantage, the limitation to organisms that can be fed exclusively by a stable isotope diet, remains. Therefore, this technology is not directly applicable to the analysis of human material like tissues or body fluids and the respective clinical applications [209]. Nevertheless, new SILAC-based approaches have been published recently, trying to extend its application also to human tissue, using either isotopically labeled standard peptides or a labeled (cell culture) peptide mixture that is used as a reference peptidome (super-SILAC) [210, 211]. Even though the super-SILAC and SILAC standards mix approaches provide some opportunities especially in terms of large-scale quantification of cellular and tissues samples, the main drawback remains, which is the need for a reference peptide-mix that does contain a labeled reference for (ideally) all proteolytic (human) peptides [212].

Just as with any other stable isotope labeling (SIL) method, the number of different samples that can be combined and quantified using SILAC is limited by the availability of differently labeled amino acids. In practice, usually not more than three samples are quantified. Higher sample numbers have been successfully quantified using deuterium as an additional label, however it has been shown that peptides labeled using deuterium can have significantly different retention times compared to their unlabeled counterparts (deuterium isotope effect [213]). Recently, the metabolic conversion of isotopically labeled peptides and the resulting addition of labels to unexpected amino acids have been reported [214].

The big advantage of SILAC is its high reproducibility due to the earliest possible time point of label introduction, which is during cell growth and division [215].

4.1.2 Enzymatic labeling

Enzymatic labeling utilizes the heavy isotope of oxygen ^{18}O, which is incorporated into peptides during tryptic digestion [216–219]. It replaces the normal, light variance ^{16}O at the C-terminus of peptides.

The major advantage of ^{18}O labeling is that the method is not limited to a specific subpopulation of peptides, which is the case for other labeling strategies (e.g., peptides containing a specific amino acid or post-translational modification) [220]. Furthermore, since peptides are enzymatically labeled, artifacts (i.e., side reactions) common to chemical labeling can be avoided [212]. Compared to SILAC, enzymatic labeling is not limited to cell culture but can be applied to principally any sample.

A disadvantage is the incorporation of labels, which is (i) commonly incomplete and (ii) happens at different incorporation rates depending on the peptide [212, 220–222].

4.1.3 Chemical labeling

Chemical labeling was introduced in the same year as metabolic labeling and represents another approach to use stable isotopes for label-based quantification [223]. In contrast to the metabolic incorporation of isotopes, labels are chemically introduced into peptides. The first application, termed isotope-coded affinity tags (ICATs), was published by Gygi *et al.* in 1999 [223]. ICAT introduces isotopic labels by derivatization of cysteine residues with a reagent containing either eight or zero deuterium atoms [112, 223]. Labeling is limited to cysteine, which is why the quantification using ICAT becomes problematic for proteins with no/only a few cysteine residues.

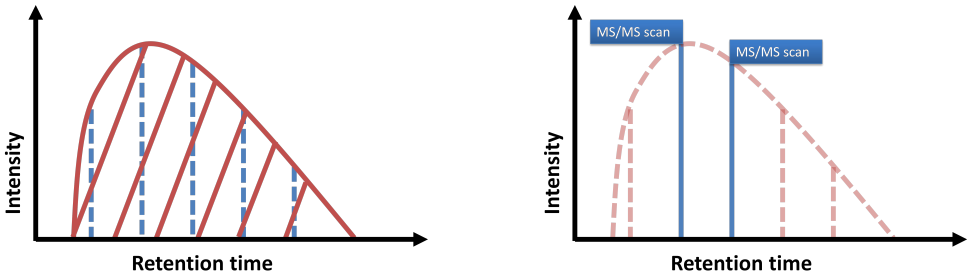
Other chemical labeling approaches are isotope-coded protein labels (ICPLs) [224], isobaric tags for relative and absolute quantitation (iTRAQ) [225], and tandem mass tags (TMTs) [201].

4.1.4 Isobaric labeling

iTRAQ and TMTs are chemical labeling strategies but can be further described as *isobaric labeling* techniques. In contrast to the isotope labeling strategies described before, isobarically labeled peptides are indistinguishable in MS^1 spectra and only become visible after fractionation in recorded MS^2 spectra. Up to ten different special ion fragments become visible, which can be used for quantification [202]. One big advantage of isobaric labels is the fact that labeled peptides are indistinguishable in MS^1 and sample complexity is not influenced. In contrast, isotopic labels described above increase sample complexity with every label that is incorporated.

4.2 Label-free Peptide Quantification

Label-based quantification is always associated with overhead in terms of additional sample preparation and costs. Furthermore, most labeling techniques are subject to limitations such as a limited number of samples that can be quantified, increased sample complexity due to introduced labels, and limited dynamic range.



(a) Area under the curve quantification. All peaks that belong to the same peptide are grouped together to a feature (feature extraction). Peptide abundance equals the "peak volume", which is the area under a feature's intensity curve.

(b) Quantification by spectral counting. The underlying assumption here is that the number of MS^2 scans that one peptide triggers directly correlates to its abundance.

Figure 4.2: Two label-free quantification strategies. Area under the curve/ intensity-based quantification and quantification by spectral counting. LC-MS peaks in blue, a feature in red.

To overcome these limitations, efforts have been made to perform peptide and protein quantification without utilizing additional labels and calculating ratios solely on MS^1 or MS^2 data. In this case sample multiplexing is not required and each sample is processed separately. Thereby, sample complexity is not altered and an unlimited number of samples can be compared.

4.2.1 Comparing ion intensities

As described before (see 3.1.3, 3.1.4 and 4), peptide ion intensities directly correlate to their abundance and therefore may be compared against each other quantitatively. Additionally, it is also possible to infer protein abundance from peptide abundances. Protein abundance is linearly correlated to the area under the curve (AUC) of peptide intensities with $r^2 = 0.9978$, in a range of 10 fmol to 1,000 fmol [226, 227]. Based on these findings, the comparison of peptide intensities was the first successfully applied strategy for label-free quantification [226, 228].

The steps required for label-free protein quantification based on the comparison of ion intensities are described into more detail below.

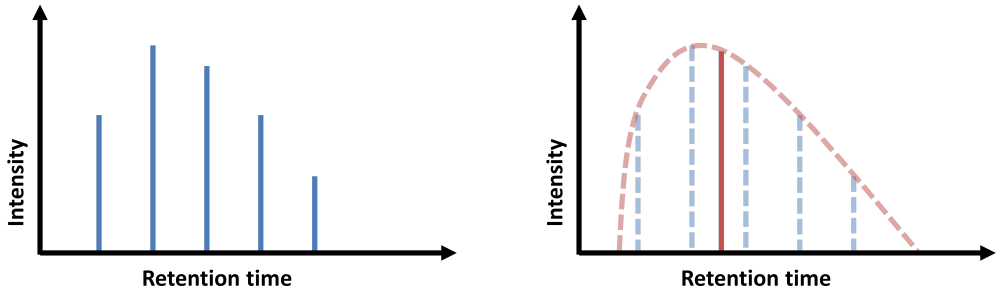
4.2.1.1 Feature extraction

To perform peptide ion intensity comparisons, the extraction of the respective intensity values from the MS[1] data is required. This data extraction step is a direct consequence of limitations deriving from the separation capacities of today's LC systems. Considering a perfect world scenario for chromatographic separation, every peptide would elute from the chromatographic column at one discrete time point. This in return would lead to perfectly sharp peaks reaching the mass spectrometer. Furthermore, each detected peak would represent exactly one single peptide.

However, in reality, the elution of most peptides is a continuous process, resulting in a continuous signal over a period of time. Since these signals/peaks all represent the same peptide, grouping these peaks and merging them into one item, which is called a *feature*, has to be performed [229]. All peaks making up a feature are called *member peaks*. Identifying member peaks and merging them into a single feature is called *feature extraction*. In literature, a different terminology may be used to describe a feature (e.g. 2D peak, 3D peak, extracted ion chromatogram (XIC), or form) and 2D/3D peak detection or form detection to describe the process of feature extraction [230, 231].

A feature shares all the properties of the contained peaks and can therefore be represented as a peak. In terms of object-oriented programming, feature *inherits* from peak. Merging many peaks into one peak (feature) requires certain data reduction strategies, since collections of m/z values, signal intensities, and retention times must be transformed into a single item.

In terms of mathematical analysis, the transformation of many peak intensities into one feature intensity can be performed by integrating the spectrum of all member peaks (AUC, as described in 3.4). This strategy is the typical approach to obtain total peptide intensity. The transformation of the remaining two peak properties, i.e., m/z values and retention times, is less crucial, since only the feature intensity is later used to perform the quantification. Nevertheless, these properties will be of importance in the later alignment process (see 4.2.1.2). For both the m/z values and the retention times, typically an average such as mean or median is calculated in order to obtain the corresponding feature property. Alternatively, data reduction can be performed by simply taking over the properties of the member peak that has the highest intensity (the so called *master-peak*).



(a) Peaks representing one eluting peptide.

(b) Feature representing one eluting peptide.

Figure 4.3: Feature extraction. During the process of feature extraction, all peaks representing the same eluting peptide are identified and merged into a feature. A feature is a *view* to the data in which each data point represents exactly one distinct peptide.

Importantly, the data processing of raw data deriving from a different LC-MS setup often requires distinct algorithms, since the patterns that need to be detected strongly depend on the instruments used. For example, the average feature length (retention time of last member peak minus retention time of first member peak) depends on the chromatographic capabilities of the LC system used and may vary significantly between different instruments. The same holds true for variations in m/z values, which depend on the mass accuracy of the mass spectrometer.

In summary, feature extraction is a process of data transformation. LC-MS data points are the input data while the output are features reflecting the biological context and each feature represents exactly one (tryptic) peptide.

4.2.1.2 Feature alignment

Following the identification of distinct peptides (i.e., feature extraction), a process that is performed for each sample independently, the resulting peptides need to be aligned across all samples to be compared. In other words: For peptide $\mathtt{p}_{\mathtt{s1}}$ from sample `s1`, its counterpart $\mathtt{p}_{\mathtt{s2}}$ in sample `s2` needs to be identified to be able to calculate an intensity ratio between the intensity of $\mathtt{p}_{\mathtt{s1}}$ and $\mathtt{p}_{\mathtt{s2}}$.

This step is connected to the chromatographic separation and the reproducibility of this process. The perfect world scenario outlined previously assumes a discrete elution of each peptide at a constant time point. Therefore, peptides could be matched just by identifying pairs of equivalent retention time and *m/z* values.

However, in reality, elution times of peptides are subject to more or less pronounced retention time shifts. The comparison of these retention time shifts and the identification of peptide pairs across samples is the task of the *feature alignment* (also termed retention time alignment, chromatographic alignment, peak matching or feature matching). From a technical point of view, this alignment can be seen as a normalization in the time dimension. That is why feature alignment is also called retention time normalization or dynamic time warping. Consequently, dynamic programming (DP)-based algorithms originating from the field of speech recognition (e.g., dynamic time warping (DTW)) are often applied [232].

Even though an alignment is a retention time normalization, one-dimensional alignment strategies (i.e., considering only the parameter retention time during the alignment process) can only be applied to low-resolution data with minimal data complexity. For this reason, all available feature properties (i.e., retention time, *m/z* ratio and, if available, feature/peptide identifications) are usually incorporated to find feature pairs across samples. These properties can be expected to be most similar for true peptide pairs.

Importantly, intensity values should not be considered, since they are the measure for peptide abundance and are expected to be significantly different in the case of differentially regulated proteins. Nevertheless, some algorithms also use intensity values to estimate the similarity between features; the reasoning is that most protein abundances can be expected to be not regulated and therefore their intensities should be similar.

Identifying feature pairs (alignments, see Table 4.1) can be achieved by searching a predefined search space. For example, possible matches (*mappings*, see Table 4.1) could be identified by selecting all features from samples `s1` and `s2` in a retention time range of three minutes and an *m/z* range of 0.05 Da. All these features present possible matches, and true pairings have to be identified. Algorithms used to identify true pairings face three main problems:

Shift problem: Since the retention time might be shifted, the use of a predefined retention time range to search for possible matches is an *a priori* source of error. An iterative approach, where the alignment is repeated (usually three times are sufficient) iteratively, using corrected retention times obtained by the prior run, can be a solution.

Transitive connections: A feature in sample `s1` likely has multiple possible matches in `s2`; however, these matched features in `s2` will not necessarily match back to the same feature in `s1` but might as well match to multiple other features in `s1`. This is problem known from graph theory, where graphs can have multiple edges (multi graph) or only single edges (simple graph). The conversion of a multi graph into a simple graph reflects, therefore, the same problem. It is usually solved by weighting the edges and keeping only the strongest ones (see Figure 4.5).

When performing feature alignments, dismissing edges affects not only directly connected notes/features but has a transitive effect on other edges at the same time; removing one edge can change the weight of other edges at the same time. This problem can be solved by DP.

Feature-alignment ratio: Considering a perfect alignment, the ratio of the total number of features in both input samples and the number of total alignments in the resulting aligned sample ($R_{fa} = \frac{count_f}{count_a}$) would be 2 (given that orphan alignments are not dismissed), since every feature in `s1` matches exactly 0 or 1 feature in `s2` and these matches are bidirectional (see Table 4.1). As the generation of a perfect alignment can often only be achieved by DP and is therefore very costly in terms of computational performance, heuristics can be used to avoid a complete DP approach. This in turn can result in features that are part in more than one alignment ($R_{fa} < 2$), especially when performing multiple alignments (see 4.2.1.2.1).

Peak	One LC-MS data point with the main properties m/z, intensity and retention time/fraction number.
Feature	A collection of peaks representing the same peptide in one sample.
Alignment	(context dependent) A pair of features representing the same peptide in two different samples. The process of aligning two (normal) samples. A sample that results from aligning two (normal) samples.
Mapping	A pair of features representing a possible alignment.
Sub alignment	An alignment that is the input for another alignment process.
Superalignment	(context dependent) An alignment process aligning two alignments. An alignment process aligning one sample and one alignment. A sample resulting from a superalignment process.
Orphan alignment	An alignment built from a feature to which no possible match in the other sample could be identified (see 4).
Complete alignment	An alignment that is built from one forward- (`m1`) and one backward mapping (`m2`), if both mappings contain the same pairing of features. A bidirectional matching of two features f_{s1} and f_{s2}.

Table 4.1: Alignment terminology.

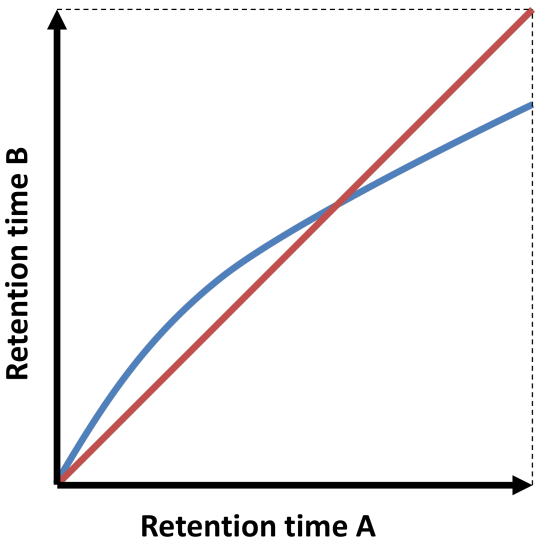
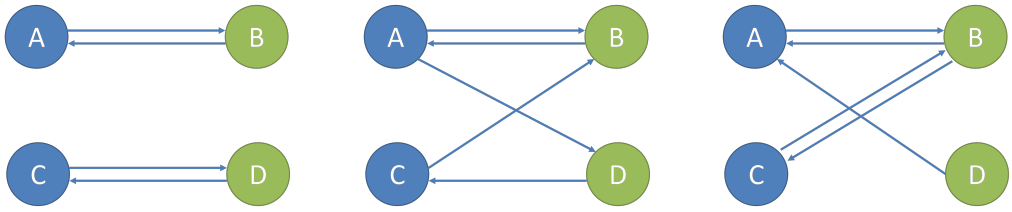


Figure 4.4: Retention time normalization. In a perfect world scenario for chromatographic separation, every peptide elutes from the chromatographic column at one discrete time point and this time point is constant across different runs of separation (illustrated in red). In the real world, chromatographic separation is subject to run-to-run variations resulting in relative retention time shifts (illustrated in blue). Retention time normalization aims at the transformation of the blue function into the red function; more precisely, to establish a normalization function that maps values from the blue function to values in the red function.



(a) Distinct alignment options. Since all possible matchings are distinct, building an alignment is obvious (A=B, C=D).

(b) Multiple but still distinct alignment options. Building an alignment is not as obvious as in (a), still all peaks may be assigned distinctly to one alignment if edge weighting is introduced. A simple approach would be to assign a stronger weight to bidirectional edges. In this case, A=B is the strongest edge, therefore A=D and B=C are dismissed. This subsequently leads to a distinct edge between C and D.

(c) Ambiguous alignment options. Again, edges are weighted since multiple options exist. If starting with peak A, first the alignment A=B is built, invalidating edges between B and C as well as edges between A and D. Therefore, C and D are left with no alignment (incomplete alignment). Considering these consequences, first B=C is built, invalidating the edge between A and B and resulting in another alignment A=D (complete alignment).

Figure 4.5: Transitive alignment connections and their effects on each other represented as graphs. Nodes A, B, C and D represent peaks or features. Blue nodes derive from sample `s1`, green nodes from sample `s2`. Possible alignment options are represented by edges between notes.

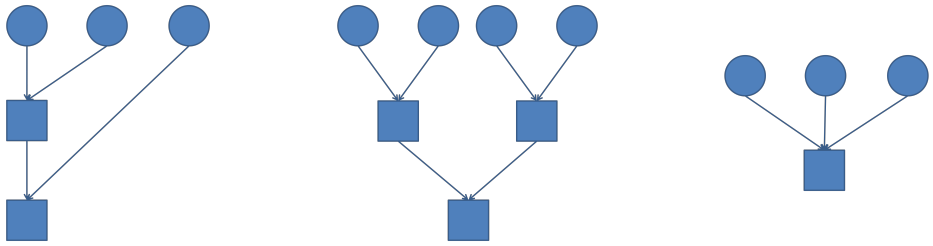
4.2.1.2.1 Multiple Alignment The described alignment process can be applied for aligning two samples against each other but is not suited for an alignment of more than two samples. This restriction appears natural, since an alignment is constructed in order to calculate an intensity ratio from two peptides ($r_p = \frac{Ip_{s1}}{Ip_{s2}}$). Nevertheless, it is often desirable to perform a multiple alignment; e.g., to compare more than two states against each other or to calculate statistics for which higher sample numbers are required. Strategies for a multiple alignment often work with so called *super-* or *master*-alignments/ samples. These terms describe an "alignment of an alignment". More precisely, a sample that results from the following workflow:

1. Creating an alignment `a1` from the two samples `s1` and `s2`
2. Creating an alignment `a2` from the two samples `s3` and `s4`
3. Creating an alignment `a3` from
   (a) the sample `s5` and the aligned sample `a1` or
   (b) from the two aligned samples `a1` and `a2`.

In this work, the terminology described in Table 4.1 is used. Different strategies to perform a multiple alignment exist; most of them make use of sub- and superalignments.

Linear alignment is an alignment strategy that creates a superalignment by first creating an initial (reference) alignment of two samples. This alignment is then aligned to the next sample, creating a new reference alignment, which is subsequently aligned to the next sample, and so forth. A linear process creates one superalignment by subsequently merging all samples. The quality of the final superalignment strongly depends on the alignment sequence. It is favorable to select an order that considers sample similarities, thereby aligning samples with a high similarity to each other first (see 12.3.6 and Figure 4.6a).

Hierarchical alignment is an alignment strategy somewhat similar to the linear alignment strategy. Subalignments are used to merge all samples subsequently into one superalignment. In contrast to the linear alignment, hierarchical alignment strategies do not make use of reference samples. First, subalignments are built from all samples. Then, these subalignments are aligned against each other until only one final superalignment is left (see Figure 4.6b).



(a) Linear multiple alignment. This alignment strategy uses always one sample as a reference. All samples are subsequently aligned to this reference.

(b) Hierarchical multiple alignment. In this alignment approach, samples are aligned in a hierarchical, tree-based manner. Most similar samples are aligned to supersamples, which in turn are aligned against each other.

(c) Iterative multiple alignment. This alignment strategy does not utilize super- and subsamples. All features from all samples to be aligned are processed at once.

Figure 4.6: Strategies for an alignment of multiple samples. Circles represent (normal) samples; squares represent supersamples.

Iterative alignment is an alignment strategy without subalignments. At first, an initial alignment containing only a subset of all features that need to be aligned is built. Iteratively, this alignment is extended by incorporating additional features until all features are merged in the initial alignment (see Figure 4.6c).

Complete pair wise alignment describes a strategy in which first all pair wise alignments are built from the collection of input samples. This results in $n\frac{n-1}{2}$ subalignments, which are subsequently merged together into one final superalignment/supersample. Merging is then performed by one of the strategies described above, omitting duplicate alignments.

4.2.1.2.2 Feature alignment based on peptide identifications

In principle, ion intensity-based protein quantification approaches can omit a feature alignment step. The finding of feature pairs for the comparison of intensities is trivial if both samples contain peptide identification information. By this approach, Bondarenko *et al.* quantified proteins in their first data set on label-free quantification in 2002 [226]. However, such a quantification strategy is limited to proteins identified in both samples with approximately the same number of peptides.

The advantage of using a feature alignment strategy consist in the fact that peptide identification information is required only for one of the two features and proteins can be quantified even if identification information is present in only one sample.

4.2.2 Counting MS^2 spectra

A label-free quantification approach introduced in 2004 assumes a direct correlation between triggered MS^2 events and peptide abundances [167]. Even though no direct link between the two parameters was presented, Liu could show label-free protein quantification based on the counting of MS^2 spectra. The spectral counting (SC) or spectral sampling approach quickly gained popularity due to its minimal computational requirements compared to the AUC approach (see 4.2.1). Since the quantification is solely based on MS^2 data, extensive feature extraction (see 4.2.1.1) and feature alignment (see 4.2.1.2) steps can be omitted.

Translating MS^2 scan events to protein abundances can be based on different strategies:

Sequence coverage: How much of a protein's sequence is covered by MS^2 scans.

Peptide count: How many peptides per protein have been identified.

Spectral count: How many MS^2 scans per protein have been triggered in total. The SC approach suffers from the main drawback that redundant identifications cannot be omitted. Blacklisting already identified precursor ions for repeated MS^2 analysis is a commonly used approach to save MS^2 resources. When identification counts are used for quantification, this dynamic exclusion of precursor ions cannot be applied, which amplifies the problems of DDA described before (see 3.2 and 4).

4.2.3 Assessment of MS^1 and MS^2 based quantification

Both MS^1- and MS^2-based approaches have their advantages and disadvantages, which will be discussed in this section.

- Advantages of MS^1-based protein quantification approaches:

  - In contrast to MS^2, peptide identification and quantification are two independently performed steps. Therefore, annotation propagation (see 12.5) as well as advanced data acquisition strategies are possible (see 13.5). For example, MS^1 quantification allows inverting the sequence of protein identification and quantification. An "anonymous" quantification, i.e., quantification without peptide assignment, can be performed at first. Subsequently, the exclusive identification of regulated features is conducted. Thereby, a significant amount of time is saved, as MS^2 analyses are only performed on a small subset of the data.
  - The commonly employed dynamic exclusion of ions already selected for fragmentation is impossible if MS^2 scans are used for the quantification of proteins. Without dynamic exclusion, a strong bias toward the quantification of highly abundant proteins has been observed. MS^1-based setups require only a single identification for each peptide to be quantified. Therefore, the application of exclusion lists does not influence the quantification results but allows for the identification (and quantification) of peptides with lower abundances.
  - The quantification of lower abundant proteins is more accurate in an MS^1 setup; proteins can be accurately quantified with as little as two peptides [212, 233]. In order to achieve the same accuracy in an MS^2-based approach, more peptides are required [212, 233]. Old *et al.* have shown that the detection of a threefold change in protein abundance requires at least four MS^2 spectra [233]. However, this number increases exponentially for smaller changes (approximately 15 spectra for twofold change in protein abundance). At the same time, saturation effects are observed at higher spectral counts and saturation levels are different for each protein [233].
  - The quantification of a "black-and-white" situation (i.e., expressing a ratio for proteins that are only present in one sample) is difficult to implement in a MS^2-based approach and therefore usually not performed in praxis (see also 17.4).

- Advantages of MS^2-based protein quantification approaches:
  - Quantification via MS^2 data is computationally much simpler compared to MS^1-based strategies, since feature extraction (see 4.2.1.1) and feature alignment (see 4.2.1.2),two non-trivial data processing steps, are not required.

4.3 Assessment of Label-based and Label-free Quantification

Both label-based and label-free quantification approaches have their advantages and disadvantages, which will be discussed in this section. Note that some of the mentioned points are specific to shotgun proteomics.

- Advantages of label-free quantification approaches:
  - No elaborate sample preparation.
  - No costly isotopes.
  - Unlimited number of samples to be compared.
  - Not limited to a specific sample type (e.g., SILAC, see 4.1.1).
  - Since samples are not multiplexed, the inherent dynamic range of the mass spectrometer is not influenced in a label-free setup. A decreased dynamic range is particularly problematic for the identification of peptide ions by MS^2, where only a limited number of peptide signals can be subject to CID fragmentation. The selection of precursor ions is biased toward high-intensity peptide signals and leads to a prominent CID undersampling of low-abundance peptides [233].
  - The quantification of a "black-and-white" situation (i.e., expressing a ratio for proteins that are only present in one sample) is difficult to implement in a label-free approach and therefore usually not performed in praxis (see also 17.4).
- Advantages of label-based quantification approaches:
  - Typically, samples are multiplexed and processed at once when using labels for quantification. This excludes all sources of quantification error introduced during both sample preparation (post combining the sample) and mass spectrometric procedures. Unlike in a label-free approach, systematic and non-systematic variations are thereby eliminated.
  - Label-based quantification approaches can be analyzed relatively easy by commercial software tools available for virtually any type of mass spectrometer.

4.4 Protein Quantification

In a *shotgun* proteomics approach (see 3.1.1), proteins are digested into peptides prior to LC-MS analysis. Therefore, quantification strategies described before yield peptide abundances instead of protein abundances. As described in 3.2, MS^2 spectra are used to assign peptide identification information to ion masses obtained by MS^1 acquisition. During this identification process, not only an amino acid sequence is matched to each fragmented ion, but also the protein of origin can be assigned. Nevertheless, a peptide sequence can sometimes be assigned to more than one protein, i.e., the peptide is not *proteotypic* (so-called *degenerated* or *shared* peptides). This problem is called *protein inference* [234–238] and three distinct approaches have been developed to solve it [231]:

1. Matching shared peptides to all possible proteins and quantifying all of these proteins.

2. Matching shared peptides to one or more protein(s), based on heuristic calculations (e.g., Occam's razor- or anti-Occam's razor principle).

3. Dismissing all shared peptides and performing the quantification on proteotypic peptides only.

Irrespectively of the applied approach, the protein abundance is calculated based on the sum of (all) peptide abundances that have been assigned to the protein (protein abundance calculation based on only a subset of peptides is possible as well, e.g., based on the top three abundant peptides). Calculating protein abundances indirectly through peptide abundances is common to all bottom-up proteomics quantification strategies.

5

Genetic Algorithms as a Tool for Parameter Optimization

The term "genetic algorithm (GA)" originates from the field of artificial intelligence and bioinformatics. It describes heuristic methods used to identify solutions to mathematical optimization problems [239–241]. While these solutions can be difficult to calculate, an exact result is often not required. If an approximation to the exact solution is sufficient, GAs are well suited, especially considering their reasonable use of resources in terms of time and computing effort.

Requirements for an application of GAs are:

1. Optimization problem, to which a "non-exact" solution is sufficient.
2. All possible solutions can be expressed as a function translating a solution into a numerical value, which can be compared to other values/solutions.

The concept of GAs is based on the idea to create two or more initial solutions to a problem and let them "evolve" in order to generate solutions that approximate the optimum. GAs are intended to mimic the natural process of evolution, natural selection and "survival of the fittest".

Parallelization plays an important role, since solutions can be calculated in parallel. As Moore's Law comes closer to losing its validity in the world of chip manufacturing, an increase in computing power can only be achieved by the use of more cores instead of faster ones [242].

$$ma(x) = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Where:

$ma(x)$	the moving average of a (time) series of data points,
n	the number of data points from which the average (mean) is calculated.

Equation 5.1: The SMA filter. The simple moving average filter can be used, for example, to separate noise from signal, like it is needed during the process of peak finding. The input function is transformed into an averaged output function. By averaging `n` data points, small variance (noise) is eliminated, only significant changes remain (signal).

Individual	One possible solution to a problem.
Gene	One atomic characteristic of an individual's ability to solve a problem.
Genome	The set of genes that together characterize an individual's ability to solve a problem.
Population	A collection of individuals.
Breeder	A "population-growing-function" that takes two individuals as parameter. The result of this function is one new individual. A breeder mimics the biological process of Meiosis. To build a genome, a breeder uses two genetic operators, which are crossover mutator and point mutator.
Crossover mutator	Mimics the biological process of chromosomal crossover: It takes one part of the first input genome and combines it with another part of the other input genome. Its frequency of action is usually 1.
Point mutator	Mimics the biological process of mutation: It randomly replaces one or more genes of one genome by any other gene. Its frequency of occurrence is variable, depending on the degree of randomness that should be involved in the whole optimization process. A starting point for a reasonable mutation rate is the natural mutation rate of eukaryotes and bacteria (app. 0.003).

Table 5.1: Genetic algorithms terminology.

Algorithms applied in low-level MS data processing (see 3.5), feature extraction (see 4.2.1.1), or feature alignment (see 4.2.1.2) often require a set of parameters. The SMA filter, illustrated in Equation 5.1, is a basic instance of such an algorithm. In this example, n is an algorithm parameter that needs to be defined. It is usually chosen empirically and strongly depends on the data. Importantly, it has a significant influence on the performance of the filter. In the case of baseline smoothing/subtraction and peak extraction (see 3.5.1), choosing a too small value for n results in false positive peaks, whereas a too large value results in false negative peaks.

In computer science, this type of parameters is often referred to as *magic numbers*, as they are empirically evaluated and a proper justification is usually missing. If a training set exists or if false positives/false negatives can be identified otherwise, the determination of these magic numbers is a tailor-made task for GAs. Numerous possible solutions can be calculated independent of each other using different parameters. Furthermore, solutions can be ordered according to their fitness by expressing the number of false positives/false negatives as a numerical value. The general structure of a GA is shown in Listing 5.1 and is further described in the following.

1. An initial set of possible solutions to a problem is created. Each solution is called *individual* and is characterized by a set of *genes* (termed *genome*). Each gene represents an atomic contribution to the solution (genome/individual) (see Table 5.1). Using the example of the SMA filter, only one parameter needs to be optimized (i.e., `n`). Therefore, every solution is represented by an individual, characterized by a genome that consists of exactly one gene. The gene is an integer value representing the number of data points from which the average should be calculated. Only this single gene defines the individual's capability to solve the problem.

   With respect to the example of the SMA filter, an initial set of solutions can be created randomly (e.g., create an initial population of individuals carrying genes that have been impressed with a random value for `n`.) or by choosing genes (values for `n`) empirically.

2. Subsequently, the population of individuals is enlarged. This is done by a "population-growing-function" termed *breeder*. A breeder is a core element of a GA and utilizes typically two operators (*crossover mutator* and *point mutator*, see Table 5.1) to generate new individuals from the existing population of individuals.

3. After new individuals have been generated, their *fitness* (e.g., the ability to solve the problem) is evaluated.

4. Finally, the "fittest" individuals within the population are chosen to breed *offspring*.

5. These steps are repeated until a stop condition is met, which can be for example a maximum population size, a maximum number of generations or a minimum fitness to reach.

A breeder operates typically on the basis of the crossover mutator and the point mutator. These *mutators* alter a genome at a specified frequency and principle of action. The point mutator mimics randomly occurring mutations that affect only one gene at a time. A point mutator introduces randomness, which is required to generate new solutions. Its frequency of action must not be chosen too large. Its frequency is typically based on the natural mutation rate (app. 0.003) or a little higher (up to 0.1).

The crossover mutator mimics the biological process of chromosomal crossover: It takes one part of the first input genome and combines it with a part of the other input genome.

A crossover typically takes place for each individual that is created, therefore its frequency of action is mostly `1`. The crossover mutator is a key element to "inherit" "fit" properties and combine them to build new, potentially even fitter genomes in a random manner.

```
createInitialPopulation()

while population.size() <= maxPopulationSize
  population.addAll(breedIndividuals())

solveProblem(population.getFittest())

function breedIndividuals()
begin
  Genome g1 = getFittest().getGenome()
  Genome g2 = getSecondFittest().getGenome()
  Genome g3 = MutatorCrossover.crossover(g1,g2)
  g3 = MutatorPoint.mutate(g3)
  return newIndividual(g3)
end
```

Listing 5.1: Pseudo code of a genetic algorithm. The population of individuals is grown until a defined maximum population size is reached (other stop criteria are possible, see main text). `breedIndividuals()` is a function to grow the population. It uses the first two individuals of the population in terms of fitness to generate one new individual. The new individual's genome is a combination of the two input genomes. It is, furthermore, randomly mutated with a defined frequency (not shown, see main text).

6

Aim of This Work

Mass spectrometry is a highly sensitive technique for the identification of peptides and proteins, but it is not inherently quantitative. To extract quantitative information from mass spectrometric data, additional sample preparation steps and/or software solutions are required. While several approaches are publicly available to perform protein quantification, almost all of them lack two important properties required for routine applications: user friendliness and flexibility. Furthermore, only very preliminary software solutions are available for the quantification of proteins using data obtained by LC-MALDI-MS. Today, LC-MALDI-MS is the work horse in several laboratories and has numerous advantages over LC-ESI-MS such as the decoupling of LC, MS^1, and MS^2.

The aim of this work was to

- extract quantitative data from LC-MALDI-MS data,
- utilize LC-MALDI-MS specific properties, and
- develop advanced data acquisition methodologies.

Furthermore, this work focused on the development of a software that overcomes the mentioned usability problems and which provides a graphical user interface allowing for easy and intuitive usage by non-experts. While the focus was clearly on the processing of LC-MALDI-MS data, the software should ideally be flexible enough to be extendable for the processing of LC-ESI-MS data.

Part II

Material and Methods

7

Sample Preparation for LC-MS

Methods and protocols described in this chapter have been established and performed in close collaboration with Wiebke Nadler.

7.1 *E.coli* Cell Lysate

10 ml 2xYT medium (MP Biomedicals molecular biology certified bacterial growth medium) were inoculated with a single colony of *E.coli* TG1 (electroporation-competent cells; K-12). Bacteria were grown in suspension culture at 37 °C overnight with continuous shaking. 500 ml 2xYT medium were inoculated in a ratio of 1:100 from the preculture and incubated at 37 °C with continuous shaking for 15 h. Bacteria were pelleted by centrifugation (4,000 x g, 4 °C, 15 min) and the pellet was washed twice with phosphate buffered saline. Bacteria where resuspended in 10 ml cracking buffer (10 mM Trizma® HCL (Sigma T1503), 5 mM EDTA (Sigma 03609), 1x cOmplete protease inhibitor cocktail (Roche Applied Sciences, pH 7.5) per gram wet weight. Lysates where sonified on ice at 35 % intensity for 2.5 min with 1 sec pulses using a Branson sonifier W250D and subsequently cleared by centrifugation (10,000 x g, 4 °C, 30 min). Protein lysates where stored at −20 °C for further experiments.

10.6 μg universal proteomics dynamic range standard (UPS2) was dissolved in 70.7 μl cracking buffer for a final protein concentration of 0.15 μg/μl. Different amounts of dissolved UPS2 were spiked into 50 μg *E.coli* lysate according to Table 7.1.

A five-fold excess of ice-cold acetone was added to 100 μl protein lysate. Samples were left at −20 °C for at least 12 h. The precipitated proteins were centrifuged (20,000 x g, 4 °C, 30 min) and the supernatant was discarded. The proteins were washed with 100 – 200 μl of 80 % ice-cold acetone, centrifuged (20,000 x g, 4 °C, 5 min) and dried in a SpeedVac.

To estimate protein content, a bicinchoninic acid assay (BCA) was performed using Pierce™ BCA Protein Assay Kit (Thermo) according to the manufacturer's instructions.

Sample	Ratio	Spike-in volume [μl]	Spike-in amount [μg]	*E.coli* lysate amount [μg]
A	0 %	0	0	50
B	0.0625 %	1	0.15	50
C	25.0 %	4	0.6	50
D	100.0 %	16	2.4	50

Table 7.1: Spike-in scheme for GeLC-MS. Human proteins have been spiked into *E.coli* cell lysate at four different concentrations, namely 16 μl UPS, 4 μl UPS, 1 μl UPS, and 0 μl UPS as a negative control (Sample A – D).

Per vial, 1 ml of a 2 mM Tris carboxyethyl phosphine hydrochloride (BioVision) solution in trypsin digestion buffer (TDB) was added and samples were incubated at 37 °C for 30 min at 1,200 rpm. Tris(2-carboxyethyl)phosphine (TCEP) solution was removed and samples were alkylated with 1 ml of 20 mM iodoacetamide (Sigma, freshly prepared) in TDB for 30 min at room temperature in the dark. The iodoacetamide solution was removed and gel slices were washed once with 1 ml water. 1 ml TDB was added and gel slices were incubated for 10 min for complete removal of the alkylating reagent.

The TDB buffer was removed and gel pieces were dehydrated in 80 % acetonitrile. The dehydration solution was removed and residual solvent was evaporated in a SpeedVac. Gel slices were resuspended with trypsin stock solution (20 ng/μl in TDB, Promega) until fully drenched and left for 15 minutes at room temperature. The gel slices were covered with 400 μl TDB and samples were incubated for 12 h at 37 °C under agitation.

Desalting was performed with a 96 well Protein/Peptide Desalting Lab-in-a-Plate Flow-Thru Plates (Glygen, FNSC18, Media bed volume 40 μl) according to manufacturer's instructions and dried in a SpeedVac.

7.2 Preparation for GeLC-MS

For each sample up to 50 μg of protein in the respective buffer were diluted with water to 80 % of the final sample loading volume. Sodium dodecyl sulfate (SDS) buffer was freshly prepared by adding mercaptoethanol in the ratio 1:100 to a buffer stock solution containing 208 μM tris (pH 6.8), 33 % glycine, 5 % SDS and 0.06 % bromophenol blue. The SDS reducing buffer was added and samples were incubated for 10 min at 70 °C before loading on a 4 – 12 % bis-tris gel (15 mm, 2-well) on a NuPage system. Separation was conducted using 3-(N-morpholino)propansulfonic acid (MOPS) buffer at 190 V constant for 1 h. The SDS-gel was stained with Simply Blue Safe Stain (Invitrogen) according to manufacturer's instructions.

Fraction	Number of gel slice	
1	1, 2, 3,	25, 26
2	4, 5, 6,	27, 28
3	7, 8, 9,	29, 30
4	10, 11, 12,	31, 32
5	13, 14, 15,	33, 34
6	16, 17, 18,	35, 36
7	19, 20, 21,	37, 38
8	22, 23, 24,	39, 40

Table 7.2: Fractionation scheme for gel slices. In order to compensate for different protein amounts in "early" and "late" slices, three "early" slices have been combiend with two "late" slices always.

Lanes were excised from the stained gel with a BioStep Lane Picker (48 bands 1.5 mm x 5 mm) or a similar tool build in-house and pooled into fractions starting from the bromophenol blue front (48 slices, pooled according to Table 7.2). The gel slices were destained in Eppendorf tubes for at least 2 h with 50 % v/v MeOH in TDB (50 mM Tris-HCL, pH 8.0, 1 mM $CaCl_2$) with overhead tumbling (Cole-Parmer Tube Rotator). Subsequently, gel slices were washed with TDB (50 mM Tris-HCL, 8.0 pH, 1 mM $CaCl_2$, BioVision).

Samples were sonicated in a water bath and the supernatant solution was transferred into a fresh LoBind Eppendorf tube. The remaining gel slices were overlaid with 800 μl extraction solution (50 % acetonitrile, 0.1 % trifluoroacetic acid (TFA)). After 10 minutes of incubation at room temperature under agitation, samples were sonified for 5 min and the supernatant solution was combined with the aquaeous peptide solution in the LoBind Eppendorf tube. The solvent was evaporated under vacuum and the resulting pellet was stored at −20 °C.

8

Liquid Chromatography and Mass Spectrometry

Samples have been analyzed using liquid chromatography coupled to mass spectrometry. Liquid chromatography was performed using two nanoACQUITY UPLC ® systems, subsequent mass spectrometric data acquisition was done using an AB SCIEX TOF/TOF™ 5800 system and an AB SCIEX QTRAP® 6500 system, respectively. The respective instrument configuration is described in this chapter.

8.1 Liquid Chromatography for MALDI-MS

Samples have been dissolved in 10.3 µl sample dissolving buffer (5 % acetonitrile (ACN), 0.1 % TFA) and subsequently vortexed and sonicated for 10 min. Injection volume was 7.3 µl.

Liquid chromatography was performed using a nanoACQUITY UPLC ® BEH130 C18 Column (1.7 µm, 75 µm x 250 mm) installed on a nanoACQUITY UPLC ® system. The UHPLC system was operated using MassLynx version 4.1 SCN779 (Waters).

Buffer A was 0.1 %(v/v) TFA water (UPLC grade), buffer B 0.1 %(v/v) TFA in ACN (UPLC grade). The column was equilibrated with 5% buffer B, followed by a trapping time of 40 min. Peptides were separated subsequently using a 110 min gradient with a constant flow rate of 350 nl/min (see Figure 8.1).

8.1.1 Preparation of matrix

Alpha-cyano-4-hydroxycinnamic acid (CHCA) (ProteoChem) was dissolved in 80 %(v/v) ACN, 0.1 %(v/v) TFA (ProteoChem) in water (UPLC grade) for a final concentration of 3 mg/ml. Peptides listed in Table 8.1 have been spiked into the matrix using a syrine-pump coupled to a SunChrom spotting robot that was used for spotting of fractions.

Time [min]	Buffer A [%]	Buffer B [%]
0.00	95.0	5.0
0.33	89.0	11.0
1.00	89.0	11.0
3.66	86.0	14.0
67.66	70.0	30.0
81.00	60.0	40.0
90.00	15.0	85.0
95.00	15.0	85.0
97.00	95.0	5.0

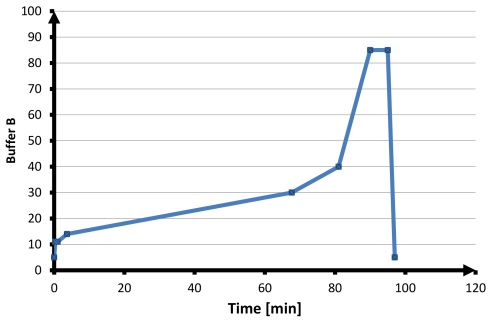


Figure 8.1: Gradient configuration LC-MALDI-MS. Total gradient length was 120 min. Percentage of buffer B was increased stepwise from 5 % to 85 % after 95 min.

Peptide sequence	Molecular weight [g/mol]
TVFDEAIR	951.0688
TGVFDEAIRTVGF	1,411.7221
CLEHMYHDLGLVRDF	1,846.8732
EEQPSTPAPKVEQQEEILC	2,155.0231

Table 8.1: Internal standard peptides spiked into each fraction of MALDI samples during spotting. Internal standard peptides that are spiked into each fraction of MALDI samples are used both for the automatic m/z calibration and a fraction-wise intensity normalization.

Time [min]	Buffer A [%]	Buffer B [%]
0.00	97.0	3.0
1.00	96.0	4.0
110.00	70.0	30.0
111.00	15.0	85.0
114.00	15.0	85.0
115.00	97.0	3.0
119.00	97.0	3.0

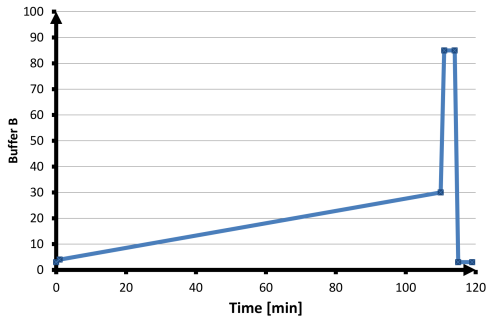


Figure 8.2: Gradient configuration LC-ESI-MS. Total gradient length was 121 min. Percentage of buffer B was increased stepwise from 3 % to 85 % after 114 min.

8.1.2 Sample fractionation

CHCA MALDI matrix was prepared in 80% ACN and 0.1 % TFA. The matrix was mixed with four internal standard peptides (see Table 8.1) and spotted following mixing with the sample on a stainless steel MALDI target plate using a spotting robot (Sunchrom micro fraction collector/MALDI spotter with SunCollect software version 1.7.26). The spotting program started the fractionation progress 10 minutes after injection at a matrix flow rate of 2 μl/min. The time for collecting a single fraction was 8 seconds for 600 fractions and 4 seconds for 1,200 fractions, respectively. After 80 minutes, the spotting was stopped.

8.2 Liquid Chromatography for ESI-MS

Samples have been dissolved in 14 μl ESI buffer (3 % ACN, 0.1 % formic acid (FA), 0.01 % TFA) and subsequently vortexed and sonicated for 10 min. Injection was done using full loop injection with an overfill factor of 1.2. Liquid chromatography was performed using an ACQUITY UPLC ® M-Class CSH™ C18 Column (1.7 μl, 300 μm x 150 mm) installed on a nanoACQUITY UPLC ® system. The UHPLC system was operated using Analyst version 1.6.1 (AB SCIEX). Buffer A was 0.1 %(v/v) FA and 0.01 %(v/v) TFA in water, buffer B was 0.1 %(v/v) FA and 0.01 %(v/v) TFA in ACN. The used gradient had a runtime of 121 min at a constant flow rate of 6 μl/min (see Figure 8.2).

8.3 Mass spectrometry

For mass spectrometric data acquisition an AB SCIEX TOF/TOF™ 5800 system and an AB SCIEX QTRAP® 6500 system have been used.

8.3.1 AB SCIEX TOF/TOF™ 5800

The AB SCIEX TOF/TOF™ 5800 system is shipped with a Microsoft® XP computer which runs the Series Explorer™ software (version 4.1.0.12, AB SCIEX) for data acquisition as well as an Oracle database (version 4.0.5) for raw data storage. Sample plate alignment, deflector offset (y2, x2) and laser intensity were manually optimized for every MS^1 run or at least once per day. Deflectors, timed ion selection and laser intensity were manually adjusted for every MS^2 run. Series Explorer™ has been configured to use the following settings to perform MS^1 and MS^2 spectra acquisition.

8.3.1.1 MS^1 spectra acquisition

MS^1 spectra acquisition was configured as follows:

8.3.1.1.1 Instrument MS^1 spectra were acquired at "Operating Mode" "MS Reflector Positive". CID Control was set to "CID off" and "Acquisition Control" was set to "automatic". "Mass Range" was set to 750 Da to 4,000 Da at a "Focus Mass" of 2,000 Da. "MALDI Matrix" was set to "alpha-cyano-4-hydroxycinnamic acid".

8.3.1.1.2 Spectrum "Acquisition Mode" was set to "Accumulate every n-shot sub-sepectrum that passes acceptance" with 250 shots per sub-spectrum, resulting in 2,000 total shots per spectrum. "Acceptance Criteria" was set to "accept every sub-spectrum". "Discard Shots" and "Single Shot Protection" were disabled.

8.3.1.1.3 Automatic Control "Sample Stage Mode" was set to "Continuous stage motion", "Search Pattern Parameters" was set to "Random" and "Uniform". "Stage Velocity" was set to 1,000 μm/sec, "Laser Intensity Mode" to "Fixed Laser Intensity".

8.3.1.1.4 Digitizer "Bin Size" was set to 0.5 ns, "Vertical Scale" to 0.5 V full scale and "Vertical Offset" to -0.5 % full scale. "Input Bandwidth" was set to 1,000 Mhz. "Laser Pulse Rate" was set to 400 Hz.

8.3.1.2 MS^1 spectra processing

MS^1 spectra processing was configured as follows:

8.3.1.2.1 Raw Spectrum Filtering/Peak Detection "Raw Spectrum Filtering" was disabled, spectrum smoothening was enabled, using a "FFT" filter with "Poisson Denoise". For peak detection, "Min SN" was set to 5, "Local Noise Window Width" to 250 m/z and "Min Peak Width at Full Width Half Max" was set to 1 bins. "Flag Monoisotopic Peaks" was enabled, as well as "Cluster Area SN Optimization" ("SN Threshold" 20). For monoisotopic peak detection, "Generic Formula" was set to "C6H5NO", with H as adduct.

8.3.1.2.2 Calibration Instrument calibration was performed using a calibration standard kit (Proteochem) including an additional peptide (see Table 8.2) spotted manually on calibration spots on the plates.

Name	Peptide sequence	Molecular weight [g/mol]
Gonadoliberin	PEHWSYGLRPG	1,182.581
Angiotensin I	DRVYIHPFHL	1,296.685
Neurotensin (-H_2O)	PELYENKPRRPYIL	1,690.928
ACTH (18-39)	RPVKVYPNGAEDESAEAFPLEF	2,465.199

Table 8.2: Peptides used for calibrating MALDI plates before each run. Four different peptides have been spotted manually on calibration spots on MALDI plates. Before each MS^1 run, the instrument was calibrated using these spots/ peptides.

In Series Explorer™, as "Calibration Type", "Internal" was selected. Parameters for "Peak Matching" where "Min" 10 S/N, "Mass Tolerance" ±0.3 mz, "Min Peaks to Match" 1 and "Max Outlier Error" 10 ppm. Use Monoisotopic Peaks Only was enabled. Weight Fit was set to Equal, Reference Masses where configured using internal standard peak masses (see Table 8.1). +1 was assumed for ion charge.

8.3.1.3 MS^2 spectra acquisition

MS^2 spectra acquisition was configured as follows:

8.3.1.3.1 Instrument MS^2 spectra were acquired at "Operating Mode" "MS-MS 1KV Positive". "CID ON" was selected for "CID Control", "Acquisition Control" was set to "Automatic". "Precursor Mass" was set to 2,465.199 Da, "Precursor Mass Window" to relative 200 resolution full width at half maximum (FWHM). "Metastable Suppressor ON" was enabled. "MALDI Matrix" was set to "alpha-cyano-4-hydroxycinnamic acid".

8.3.1.3.2 Spectrum "Acquisition Mode" was set to "Accumulate every n-shot sub-spectrum that passes acceptance" with 250 shots per sub-spectrum. The acceptance of 12 sub-spectra that pass acceptance was set as a stop condition resulting in "Total Shots/Spectrum" of 3,000. As an alternating stop condition, "After final spectrum reaches desired quality" was set to "high". "Acceptance Criteria" was set to "accept every sub-spectrum". "Discard Shots" and "Single Shot Protection" were disabled.

8.3.1.3.3 Automatic Control "Sample Stage Mode" was set to "Continous stage motion", "Search Pattern Parameters" was set to "Random" and "Uniform". "Stage Velocity" was set to 1,200 μm/sec, "Laser Intensity Mode" to "Fixed Laser Intensity".

8.3.1.3.4 Digitizer "Bin Size" was set to 1 ns, "Vertical Scale" to 0.5 V full scale and "Vertical Offset" to 0.1 % full scale. "Input Bandwidth" was set to 200 Mhz. "Laser Pulse Rate" was set to 1,000 Hz.

8.3.1.4 MS^2 spectra processing

MS^2 spectra processing was configured as follows:

8.3.1.4.1 Raw Spectrum Filtering/Peak Detection "Subtract Baseline" was disabled, spectrum smoothening was enabled and configured to perform a "Savitzky-Golay" filtering with "Points-Across Peak" at 5 and a "Polynomial Order" of 4. For peak detection, "Min SN" was set to 15, "Local Nose Window Width" to 250 and "Min Peak Width at Full Width Half Max" was set to 1.5 bins. "Flag Monoisotopic Peaks" was enabled as well as "Cluster Area SN Optimization" (SN Threshold 15). For monoisotopic peak detection, "Generic Formula" was set to "C6H5NO", with H as adduct.

8.3.1.4.2 Calibration "Calibration Type" was "Default".

8.3.1.5 MS^2 spectra interpretation

"Job-wide Precursor Selection/Methods" was configured as follows:

8.3.1.5.1 Monoisotopic precursor selection for MS^2 "Minimum SN filter" was set to 50, "Minimum Mass" to 750 Da, "Minimum Retention Time" to 0 min. "Maximum Mass" was set to 4,000 Da, "Maximum Retention Time" to 99,999 min. "Adduct Exclusion List" contained two entries, (i) 21.982 Da and (ii) 37.956 Da. "Adduct Tolerance" was set to $\pm$0.03 m/z and "Exclude precursors" was set to within 200 resolution. "Minimum Chromatogram Peak Width" was set to 1 fraction(s) and "Fraction-to-Fraction Precursor Mass Tolerance" to 200 ppm.

8.3.1.5.2 Precursor Final Selection Criteria "MS/MS Acquisition Order/Fraction" was set to "Weakest Precursor First", "First Precursor to Skip/Fraction" to 0, "Max Precursors/Fraction" to 35 and "Max Precursors/LC-Run" to 40,000.

8.3.2 AB SCIEX QTRAP® 6500

The AB SCIEX QTRAP® 6500 system is shipped with a Microsoft® 7 computer which runs the Analyst software (version 1.6.1 AB SCIEX) for both controlling the nanoACQUITY UPLC ® and the AB SCIEX QTRAP® 6500. MS data acquisition was performed using enhanced MS (EMS) at a scan rate of 10,000 Da/s and a positive polarity. Scan time was 120.008 min with 3,081 cycles (2.3371 seconds per cycle). Subsequent enhanced resolution (ER) scans were performed at a scan rate of 50 Da/s and a positive polarity. Selection of MS^2 was performed according to the following DIA criteria: Top three most intense precursors where selected when exceeding an intensity of 10,000 within a mass range of 450 m/z – 2,000 m/z and a charge state of two, three or unknown. Precursors where fragmented with rolling collision energy. After 15 occurrences target ions where excluded for 15 seconds within a mass tolerance of 250 mDa. Isotopes where excluded within a 4 Da window. Subsequently, charge state and/or isotope pattern were confirmed by enhanced product ion (EPI) scans at a scan rate of 10,000 Da/s and a positive polarity. Dynamic background substraction was applied.

9

Preprocessing of Data

The current version of MS_QBAT does not support LC-MS low-level data processing such as peak finding, peak deconvolution and peptide identification from MS^2 data. Therefore, these tasks have been performed by the third-party software DataExplorer, ProteinPilot™ and DeconTools respectively. An in-detail description of the application of third-party software is presented in this chapter.

9.1 LC-MALDI-MS

9.1.1 MS^1 data

Raw data was exported to `t2d`-files using the respective context menu in the Series Explorer™ software. `t2d`-files where subsequently loaded into the DataExplorer software, which was used to export peaks into tab-delimited text-files (table headers were `Centroid Mass`, `Height` and `S/N Ratio` for m/z, intensity and signal-to-noise ratio (S/N) values, respectively).

9.1.2 MS^2 data

Peptide identification was performed by the ProteinPilot™ software (version 4.5.1656, AB SCIEX, Paragon™ algorithm, version 4.5.0.0.1654). Raw data was directly imported from the database via ProteinPilot™'s "Add TOF/TOF data.." option and searched against a database containing the *E.coli* proteome as well as UPS2 proteins (database downloaded on August 1, 2013 from UniProt). Further ProteinPilot™ configuration is listed in Table 9.1. Peptide identifications were exported from ProteinPilot™ to `PeptideSummary.txt` files, which were further processed by PepSir (version 1.6.1, see 11); entries with a sequence confidence below 95 % have been dismissed.

9.2 LC-ESI-MS

`raw`-files, as well as available MaxQuant [196] result files were downloaded from ProteomeXchange [243]. Data was acquired by a QExactive™ system (Thermo Fisher) and originally analyzed with MaxQuant.

9.2.1 MS^1 data

`raw`-files where processed using DeconTools (version 1.0.5280) configured with the provided default parameter file. The generated output were `isos.csv` files, containing a monoisotopic peak list (table headers were `monoisotopic_mw`, `abundance` and `signal_noise`, for m/z, intensity and S/N values, respectively), separated by commas.

9.2.2 MS^2 data

Peptide identification was extracted from the `modificationSpecificPeptides.txt` file available from the ProteomeXchange website. This file provides the necessary information for feature annotation, which is a peptide sequence string, a monoisotopic mass and a MS^2 scan number representing the retention time. Nevertheless, the file summarizes peptide identification from all samples/`raw`-files in a non-redundant fashion which complicates subsequent processing with MS$_Q$BAT (see section 17.3).

In order to exclude non-proteotypic peptides, extracted peptide sequences have been further processed by PepSir (see 11, no filtering applied).

9.3 GeLC-MALDI-MS

9.3.1 MS^1 data

Data preprocessing was identical as described in 9.1.1.

9.3.2 MS^2 data

Data preprocessing was identical as described in 9.1.2.

MALDI/ESI	Parameter name	Parameter value
MALDI	Sample Type	`Identification`
MALDI	Cys Alkylation	`Iodoacetamide`
MALDI	Digestion	`Trypsin`
MALDI	Instrument	`5800`
MALDI	ID Focus	`Biological modifications`
		`Amino acid substitutions`
MALDI	Search Effort	`Thorough ID`
MALDI	Species	`None`
ESI	Sample Type	`Identification`
ESI	Cys Alkylation	`Iodoacetamide`
ESI	Digestion	`Trypsin`
ESI	Instrument	`6500 QTRAP ESI`
ESI	Special Factors	`Gel-based ID`
ESI	ID Focus	`Biological modifications`
		`Amino acid substitutions`
ESI	Search Effort	`Thorough ID`
ESI	Species	`None`

Table 9.1: ProteinPilot™ settings. Listed are ProteinPilot™ search settings used for analysis by LC-MALDI-MS (top) and by LC-ESI-MS (bottom).

9.4 GeLC-ESI-MS

9.4.1 MS^1 data

`wiff`-files where converted to `mzML` using MSConvert available from the ProteoWizard 3.0.7036 package. `mzML` files where further processed using DeconTools (version 1.0.5280) configured with the provided default parameter file. The generated output were `isos.csv` files, containing a monoisotopic peak list (table headers were `monoisotopic_mw`, `abundance` and `signal_noise`, for m/z, intensity and S/N values, respectively), separated by commas.

9.4.2 MS^2 data

Peptide identification was performed by the ProteinPilot™ software (version 4.5.1656, AB SCIEX, Paragon™ algorithm, version 4.5.0.0.1654). `wiff`-files where loaded and searched against a database containing the *E.coli* proteome as well as UPS2 proteins (database downloaded on August 1, 2013 from UniProt). Further ProteinPilot™ configuration are listed in Table 9.1. Peptide identifications where exported from ProteinPilot™ to `PeptideSummary.txt` files, which were further processed by PepSir (see 11); entries with a sequence confidence below 95 % have been dismissed.

10

Software Development

All algorithms as well as the application software MS_QBAT are written in Java 1.6. The Eclipse integrated development environment (IDE) (versions Helios, Indigo, Juno, and Kepler) was used for software development. MS_QBAT is a rich client application build upon the Eclipse rich client platform (RCP). RCP is a highly modular, service oriented plug-in system build upon the open service gateway initiative (OSGi) framework [244, 245]. The RCP framework is used as a basis for various software applications (open-source and commercial projects). Among these programs are news reader, file browser and the NASA JPL Maestro Space Mission Control software. Just as OSGi and RCP, MS_QBAT is a highly modular application and all main components are implemented as OSGi plug-ins. All OSGi plug-ins which are independent of RCP further support the incorporation as simple Java libraries into other software and are therefore independent from MS_QBAT, RCP or any graphical user interface (GUI). Other use cases and application environments are therefore possible, such as their integration in command-line interface (CLI)- or Spring-based applications [246]. Today MS_QBAT is composed of 41 core modules and 130 different helper modules (deriving from the Eclipse framework and other third-party libraries). Most of the libraries are build using Apache Maven (version 2.2.1), which is configured to create jar files that can be used either as simple Java libraries or OSGi plug-ins. Today Apache Maven still lacks proper support for building RCP components. This is why MS_QBAT plug-ins that require the RCP environment are build using Eclipse's export wizard (which internally uses *Apache Ant*). These RCP plug-ins are the GUI as well as few helper components. git (version 1.9.1) is used as source code management (SCM) system; branching and versioning mainly follows the git flow model.

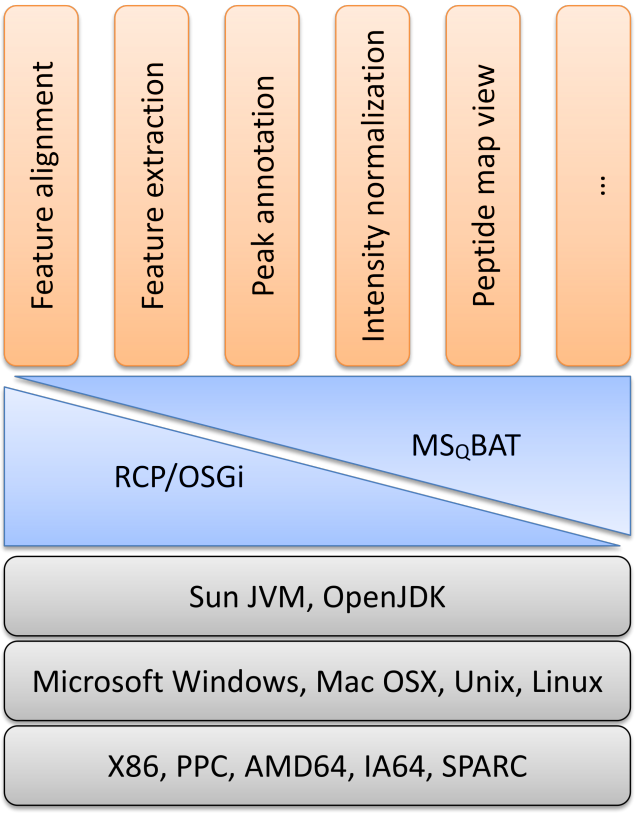


Figure 10.1: Overview of MS$_Q$BAT's software architecture. MS$_Q$BAT provides a modular and extensible software architecture. It consists of several plug-ins (illustrated in orange), such as the implemented algorithms for peptide and protein quantification (e.g. feature extraction, -alignment, etc.) as well as different components of the GUI (e.g. different views and perspectives). These plug-ins hock into the main MS$_Q$BAT component which is implemented as a RCP/OSGi plug-in (illustrated in blue). RCP is an extension to OSGi, which is a service-oriented plug-in framework. OSGi is pure Java and can be therefore executed on any system that has a JVM installed. JVMs are available for virtually any operating system and CPU architecture (illustrated in grey). Adapted from [247].

Part III

Results

11

PepSir

The identification of proteins in *bottom-up* proteomics experiments resembles a riddle comprised of a mixture of several thousand jigsaw puzzles each with a variable number of pieces [248]. Unfortunately, some of these pieces match several puzzles, thereby complicating their assembly. In bottom-up proteomics experiments, these pieces are termed *degenerated* peptides (see 4.4). By contrast, pieces matching only a single jigsaw puzzle (and therefore only a single protein sequence) are termed *proteotypic* peptides [249]. In quantitative proteomics experiments, proteotypic peptides carry the required quantification information, while degenerate peptides can water the quantification down. Therefore, the identification and utilization of proteotypic peptides is fundamental in bottom-up proteomics [250].

Importantly, there is no definitive proteotypic peptide. Whether a peptide is proteotypic or not depends on the context, i.e., on the sample composition. Figure 11.1 illustrates the correlation between the fraction of proteotypic peptides for each human protein represented in the manually annotated and reviewed Swiss-Prot database (one part of the protein knowledge base UniProtKB [251]) and the composition of the sample background. While virtually all tryptic peptides (no missed cleavage, mass range from 750 to 3,000 Da, repeatedly occurring peptides filtered) derived from the 20,012 human proteins listed in the Swiss-Prot database are proteotypic in an *E.coli* background, the fraction of degenerated peptides increases for an *H.sapiens* and an *H.sapiens*/*M.musculus* mixed background to approximately 5 % and 37 %, respectively. Although only 5 % of the peptides are degenerated with respect to an *H.sapiens* background, more than one third of the proteins contain at least one degenerated peptide. In the mixed human/mouse background, the percentage of proteins yielding degenerated peptides upon tryptic digestion is bigger than 80 %. Considering all 545,388 proteins contained in the Swiss-Prot database, only half of the tryptic peptides derived from the 20,012 human proteins are “truly” proteotypic and more than 10 % of all human proteins do not contain any proteotypic peptide.

In quantitative proteomic experiments, the occurrence of these degenerated peptides has to be taken into account. The selection of proteotypic peptides with regard to the sample background is essential for protein quantification and has to be performed carefully. Namely, proteomic experiments utilizing samples with mixed species background (e.g., deriving from xenograft mouse models) require a diligent evaluation of peptides employed for quantification (see Figure 11.1).

PepSir is a software tool for the identification of proteotypic peptides with respect to a user-defined background. PepSir is an easy-to-use Java - based software with graphical user interface and drag&drop capabilities (see Screenshot 11.1). The software requires solely two input files:

- A protein sequence database in the FASTA format, downloaded from UniProt representing the sample origin
- A list containing peptide sequences

All peptides contained in the sequence list are searched against the protein sequence database supplied. PepSir considers a peptide to be proteotypic if its sequence matches a single identifier. In order to cope with the existence of redundant database entries (such as protein isoforms, erroneous sequence entries, or multiple entries due to the presence of more than one species), PepSir allows the definition of this identifier. The software provides the user with four options: (i) accession number, (ii) protein name, (iii) gene name, and, (iv) a combination of protein/gene name. When searching non-redundant databases, the use of any of the four options results in the same list of proteotypic peptides. Because the same search against a redundant database would result in the identification of peptides wrongly assigned to be degenerated, a careful selection of the applied identifier is crucial. The option “combination of protein/gene name” might be helpful for the analysis of redundant databases, since peptides annotated to proteins with the same protein name and/or the same gene name are considered proteotypic. When performing searches with multi-species databases, PepSir allows the user to decide if species-specificity is required for proteotypic peptides.

Beside the removal of degenerated peptides, PepSir comprises the option to ignore and/or remove peptides with low confidence scores for additional data filtering. By indicating the location of the confidence information and the respective thresholds in the preferences menu, the software directly applies the additional filtering criteria. The resulting PepSir -output file is an extended version of the input file and includes information whether the peptide is proteotypic in the context of the utilized database/identifier as well as information on the database entries containing the peptide sequence.

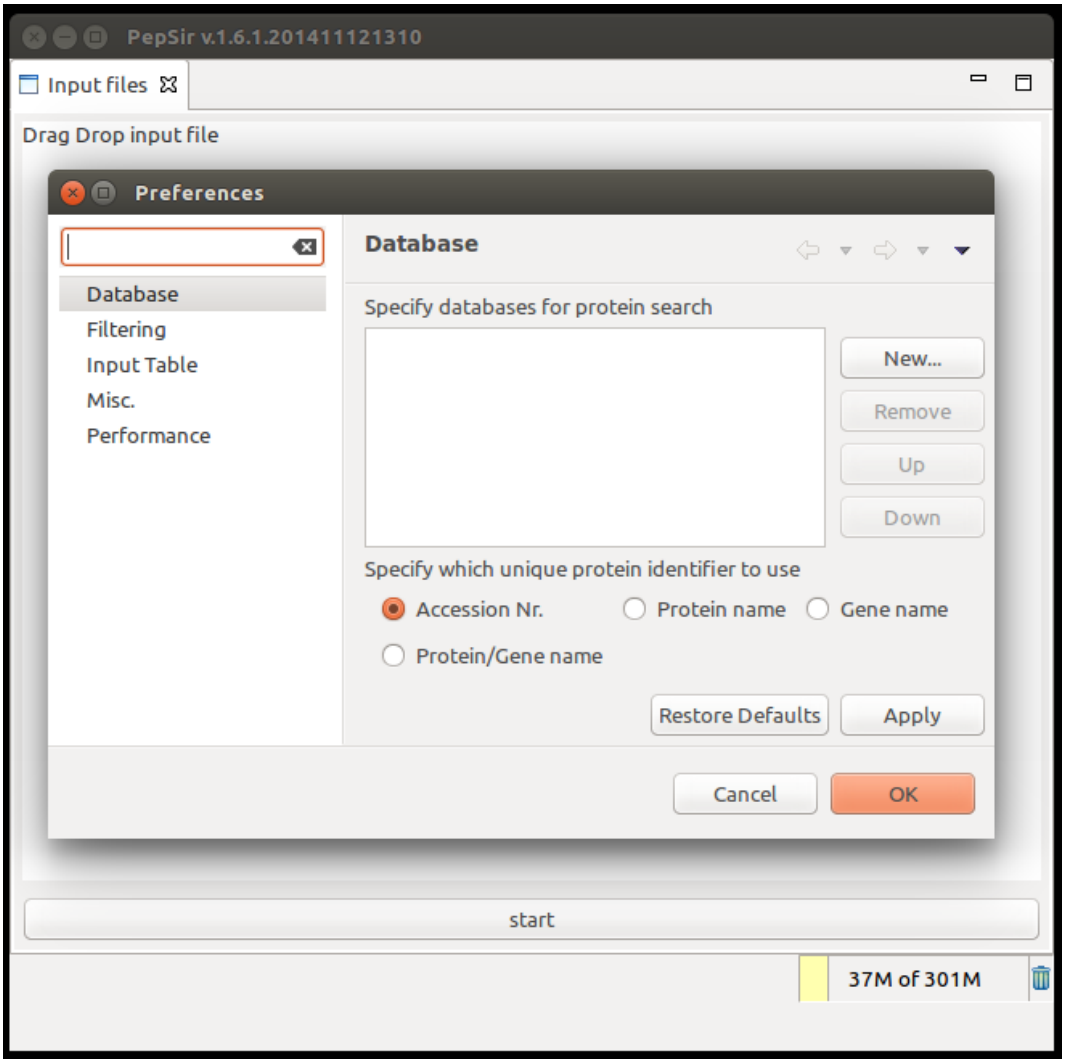
Alternatively, PepSir allows the generation of an output file, which is curated and filtered but structurally identical to the input file. Therefore, the file modified by PepSir can directly replace the original input file in any type of downstream data analysis.

PepSir can be executed on any standard PC or MAC with installed Java Runtime Environment without any particular hardware requirements. The processing of a file containing 10,000 peptide sequences takes 11 s, 97 s, and 177 s for a database containing 4,000, 20,000, and 37,000 protein entries, respectively. On multi-core systems, several input files can be processed in parallel. The software can easily be customized to the user's requirement. It allows the use of any type of sequence database in the FASTA format, including, e.g., modified and/or merged databases (as illustrated in Figure 11.1 with a combination of the two reference proteome sets for *H.sapiens* and *M.musculus* downloaded from the UniProt webpage). While the minimum input file consists of a simple list of peptide sequences, reports generated, e.g., by ProteinPilot™ [161] or Mascot [159] can be processed without further modification.

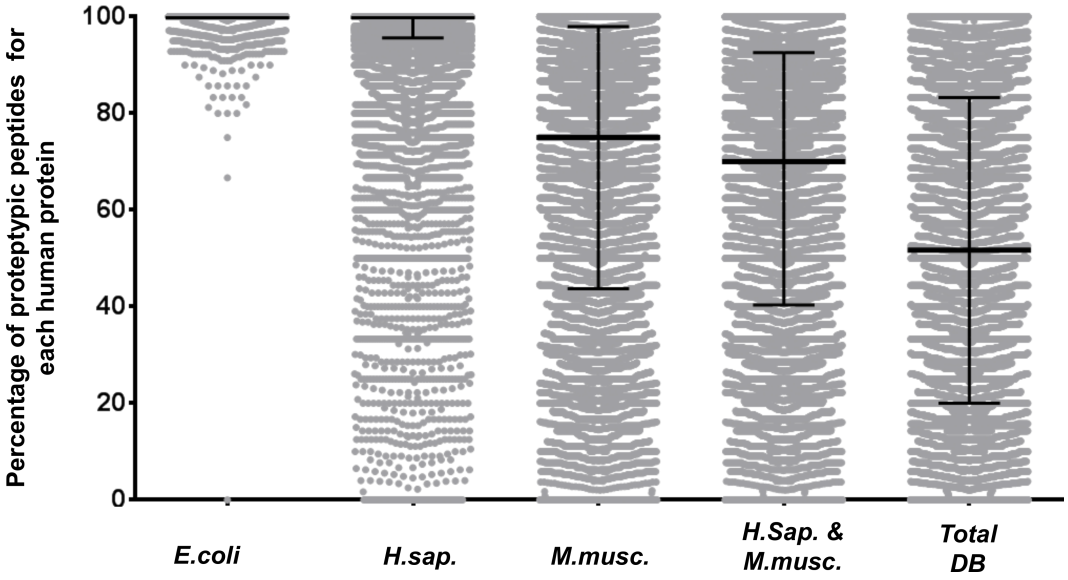
Proteomic data acquisition is often performed over an extended period of time. Since databases are usually updated several times per year, identification results might be partially outdated when the final analysis is performed. The assembly of protein identification data obtained from searches with different database versions can, in principle, result in inconsistent mapping of peptides to proteins. Heterogeneous annotations of peptides to proteins do not only influence protein identification and sequence coverage but might also result in distorted protein quantification. Similarly, the analysis of proteomic results derived from different labs can be hampered by the fact that not the same database was applied during the identification process.

PepSir allows for a simple, fast, and convenient updating of search results to the newest version of an input database. The software enables the automated take over of the most recent identifiers. The composition of this identifier can easily be adapted by the user to match the specific needs. Any combination of the primary accession number of the UniProtKB entry, protein name, gene name, and species separated by any string can be adopted as a new protein identifier.

In summary, PepSir allows to identify proteotypic peptides within user-defined backgrounds requiring only two input files: a list of identified peptides and a database in the FASTA format. Furthermore, the software allows filtering data and can be utilized to standardize datasets derived in different labs and/or with different database versions.



Screenshot 11.1: Main window and preferences window of PepSir. PepSir's GUI allows loading input files via drag&drop into the main window. In the preferences section, database(s) can be defined. Furthermore, the content of the output file can be configured according to the required protein identifier, the filtering of peptides identified with low sequence confidence and the structural format of the input/output file.



(a) Exemplary percentage of proteotypic peptides for each of the 20,012 human proteins as a function of the context, i.e., the background database. Median and interquartile range are indicated.

Parameter	Database	E.coli	H.sap.	M.musc.	H.sap. & M.musc.	total DB
Size of database (# of proteins)		4,431	20,012	16,669	36,681	545,388
Peptides	Proteotypic	469,890	448,894	305,422	297,243	243,378
	Degenerated	650	21,646	165,118	173,297	227,162
Proteins	Only proteotypic peptides	19,405	12,764	4,875	3,741	2,319
	At least one degenerated peptide	607	7,248	15,137	16,271	17,693
	Only degenerated peptides	1	142	614	750	2,088

(b) Listing of the database sizes, the number of proteotypic/degenerated peptides and the number of proteins identified either only with proteotypic peptides, with at least one degenerated peptide or only with degenerated peptides. A mixed species background, which is often encountered in biological research, necessitates a careful evaluation of the identified peptides with regard to their matching to multiple proteins.

Figure 11.1: Context dependency of the term *proteotypic*. All 20,012 reviewed proteins contained in the reference proteome set of *H.sapiens* from UniProtKB/Swiss-Prot were digested *in silico* with trypsin (no missed cleavage, mass range from 750 Da to 3,000 Da resulting in 470,540 peptides. Subsequently, these peptides were loaded into PepSir and searched against the five different reference proteome sets *E.coli*, *H.sapiens*, *M.musculus*, *H.sapiens* & *M.musculus*, and the total Swiss-Prot database.

Before	After	Δ	Δ [%]
10,335	7,397	2,938	28.4
9,146	6,771	2,375	26.0
7,646	5,441	2,205	28.8
6,713	4,855	1,858	27.7
10,789	7,931	2,858	26.5
8,992	6,748	2,244	25.0
10,763	7,934	2,829	26.3
6,397	4,476	1,921	30.0
9,034	6,311	2,723	30.1
9,972	7,454	2,518	25.3
5,344	3,466	1,878	35.1
3,947	2,614	1,333	33.8
average		**2,307**	**28.6**

(a) Peptides removed by the 95% confidence filter.

Before	After	Δ	Δ [%]
7,397	6,307	1,090	14.7
6,771	5,772	999	14.8
5,441	4,633	808	14.9
4,855	4,086	769	15.8
7,931	6,651	1,280	16.1
6,748	5,688	1,060	15.7
7,934	6,767	1,167	14.7
4,476	3,680	796	17.8
6,311	4,992	1,319	20.9
7,454	5,861	1,593	21.4
3,466	2,639	827	23.9
2,614	1,845	769	29.4
average		**1,040**	**18.3**

(b) Removal of degenerated peptides.

Before	After	Δ	Δ [%]
10,335	6,307	4,028	39.0
9,146	5,772	3,374	36.9
7,646	4,633	3,013	39.4
6,713	4,086	2,627	39.1
10,789	6,651	4,138	38.4
8,992	5,688	3,304	36.7
10,763	6,767	3,996	37.1
6,397	3,680	2,717	42.5
9,034	4,992	4,042	44.7
9,972	5,861	4,111	41.2
5,344	2,639	2,705	50.6
3,947	1,845	2,102	53.3
average		**3,346**	**41.6**

(c) Difference in peptide numbers between input and following double filtering (>95 % confidence and proteotypic).

Table 11.1: Peptide numbers before and after filtering by PepSir. Illustrated are peptide numbers before and after filtering peptide identifications. The data shown are exemplary extracts of a larger data set. Healthy NOD scid gamma mice have been perfused with different biotinylation reagents. Biotinylated proteins were captured, reduced, alkylated, delipidated, tryptically digested, and subsequently analyzed by LC-MALDI-MS.

12

Cornerstones of Label-free Peptide and Protein Quantification

For the reasons described before (see 6), algorithms required for a label-free peptide and protein quantification have been implemented from scratch focusing on the utilization of MALDI-specific properties. These algorithms build the basis for a label-free quantification and are described in detail in this chapter.

12.1 Feature Annotation with Peptide Identification Information

As described before (see chapter 9), MS^1 and MS^2 data are processed in different workflows. Peak lists, which build the initial input for MS_QBAT and its algorithms, are extracted by Series Explorer™ and DataExplorer from MS^1 raw data. MS^2 spectra are processed by ProteinPilot™ that performs the peptide/ protein identification.

Alternatively, peak lists as well as peptide information can be loaded using a generic table format, which needs to contain only elementary information such as m/z or molecular weight, intensity or peptide sequence.

```
N Unused  Total %Cov  %Cov(50)  %Cov(95)  Accessions  Names Used  Annotation  Contrib Conf
     Sequence  Modifications Cleavages dMass Prec MW Prec m/z  Theor MW  Theor m/z Theor z Sc
     Spectrum  Specific  Time  PrecursorSignal PrecursorElution
1 109.47  109.47  80.4700 76.6099 75.1100 sp|P69908|DCEA_ECOLI  Glutamate decarboxylase alpha
     OS=Escherichia coli (strain K12) GN=gadA PE=1 SV=1      1.2518  94.3700 LLTDFR      0.0197
       763.4426  764.4499  763.4228  764.4301  1 9 1.UPS2_2%_4.12.487.18 0 38.4  478048.7  38.4
       p sp|P69908|DCEA_ECOLI Glutamate decarboxylase alpha OS=Escherichia coli (strain K12) GN
     =gadA PE=1 SV=1
2 103.63  103.63  89.5500 80.3099 77.9100 P02768ups|ALBU_HUMAN_UPS  Serum albumin (Chain
     26-609) - Homo sapiens (Human)     2 99.0000 AAFTECCQAADK  Carbamidomethyl(C)@6; No
     Carbamidomethyl(C)@7   -0.0020 1313.5356 1314.543  1313.5379 1314.5452 1 15  1.UPS2_2%_4
     .12.241.7  1 22  1700.663  22  p P02768ups|ALBU_HUMAN_UPS Serum albumin (Chain 26-609) -
     Homo sapiens (Human)
2 103.63  103.63  89.5500 80.3099 77.9100 P02768ups|ALBU_HUMAN_UPS  Serum albumin (Chain
     26-609) - Homo sapiens (Human)     2 99.0000 ADDKETCFAEEGK Carbamidomethyl(C)@7  missed K-
     E@4  -0.0050 1498.6196 1499.627  1498.6246 1499.6318 1 20  1.UPS2_2%_4.12.235.5  1 21.6
     6861.537  21.6  p P02768ups|ALBU_HUMAN_UPS Serum albumin (Chain 26-609) - Homo sapiens (
     Human)
[...]
```

Listing 12.1: Example of a `PeptideSummary.txt` file. `PeptideSummary.txt` is a simple tab-delimted text file. It provides information on the amino acid sequence, m/z value, retention time and others. It is exported from ProteinPilot™ and used to extend MS^1-based peak information with MS^2-based peptide information.

12.1.1 Importing annotation information from ProteinPilot™

From ProteinPilot™, tab-delimited text files can be exported that contain precursor *m/z* values, fraction and the corresponding peptide identification such as the peptide name and sequence. The peptide identification files are read by the annotation library and are used to annotate peaks or features obtained by an MS^1 run. At this point, the annotation library is used to combine MS^1 and MS^2 data into one data set. An exemplary content of a tab-delimited protein identification file exported from ProteinPilot™ is shown in Listing 12.1. During the reading of the peptide identification files, peptide identifications can be filtered using different criteria such as sequence confidence or mass difference between calculated *m/z* and detected *m/z* to process only high quality identification information. High confidence annotations according to these criteria are subsequently assigned to peaks deriving from the previously loaded MS^1 data. The peak most closely maching the annotation with respect to *m/z* and retention time is selected and annotated.

12.1.1.1 Details of implementation

The annotation algorithm including all Java types is provided in the form of three OSGi plugins, which can easily be integrated both into simple command-line-only applications and into more complex, GUI enabled RCP applications. The first plug-in contains all types that represent the application programming interface (API) of the annotation library. The second plug-in provides all types that implement annotation related input/output (I/O) functionality required for reading annotation information from tab-delimited text files produced by an export from ProteinPilot™. The third plug-in contains all other types that implement the annotation API and provide the concrete annotation functionality.

12.2 Feature Boxing as a Novel Feature Extraction Algorithm

MS_QBAT provides a feature extraction algorithm developed from scratch in order to take full advantage of MALDI in general and of the nanoACQUITY UPLC ® coupled to the AB SCIEX TOF/TOF™ 5800 system in particular.

The algorithm was named *feature boxing*, because figuratively speaking, it works by "boxing" all peaks corresponding to the same peptide (*member peaks*, see 4.2.1.1). The resulting boxes represent identified features and are therefore termed *feature boxes*. In principle, the algorithm can be divided into two components:

Box initiator represents a strategy that initiates feature boxing by grouping member peaks and building preliminary features.

Box tightener represents a strategy that "tightens" existing feature boxes. More precisely, a box tightener evaluates if the properties of existing features can be improved by splitting boxes into smaller ones. Importantly, tighteners cannot be used to "enlarge" an existing box. It is therefore crucial for the box initiator to choose the initial feature boxes large enough.

In order to take full advantage of MALDI, a box initiator *MALDI boxes* has been implemented that makes use of the MALDI specific ionization pattern. MALDI boxes assumes that peak masses can only occur within a specific mass range since their charge state is predominantly 1. Assuming a detectable mass range of 499.75 Da to 4,500.7495 Da and a mass interval of 1.0005 Da, all possible feature boxes are represented by a sequence of masses (see Equation 12.1) Each mass represents the "center" of the box and each box has a range of ±0.50025 Da.

After the initial feature selection, preliminary features are further refined using several box tightener:

Mass tightener introduces a maximum, inter-feature mass variance (see Figure 12.1a). This tightener iterates over a feature's member peaks and calculates the *m/z* difference between the current peak and the next peak. If this *m/z* difference is larger than a defined threshold (configurable, 50 ppm is used per default) the feature is split at this position. All peaks of a certain feature inspected before the splitting position stay in the current feature, while all others will are moved to a new feature.

Gaps tightener introduces a maximum number of "missing" fractions within a feature (see Figure 12.1b). When iterating over member peaks, the difference in the fraction number of the current peak and the next peak must not be above this defined limit. Assuming an eluting peptide generating a continuous signal over the complete time of detection, features can be tightened based on signal gaps. A feature is split if a gap of at least x fractions occurs. The maximum gap length can be defined both as an absolute number and as a dynamic value, depending on a feature's length.

Annotation tightener ensures that a feature is only annotated with one peptide identification (see Figure 12.1c). As peaks are annotated (see 12.1) before the extraction of features, features can in principle contain more than one annotated peak. Such features do not represent one distinct peptide and are therefore split. Thereby, two or more features are created and each feature contains only one distinct peptide identification.

$$\{a, a+i, \ldots, s-i, s\}$$

Where:

- a is the lower bound of the sequence,
- s is the upper bound of the sequence,
- i is the interval of the sequence.

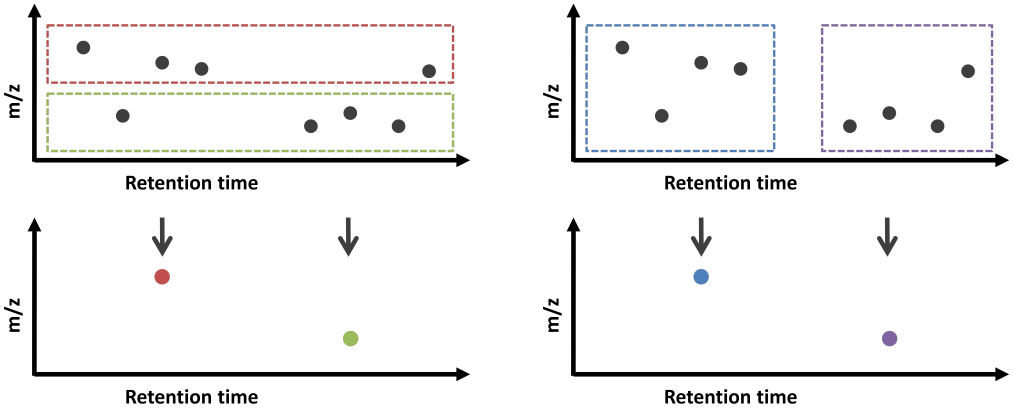
Equation 12.1: "MALDI boxes." Assuming a charge state of 1 and the generic formula $(C_6H_5NO)_n$ for the atomar composition of a peptide, a peptide-ion's fractional mass is approx. 0.5 per 1,000 Da. Valid peptide-mass windows can therefore be calculated by the formula depicted above, using an interval of 1 Da. Lower bound and upper bound are more or less arbitary but usually in the range of 500 Da – 5,000 Da.

It has to be mentioned, that the order of application of tighteners might have an impact on the final result even in the case that all tighteners operate at unchanged settings. The *m/z* -tightener for example operates solely on the *m/z* -dimension. All peaks that stay in the defined *m/z* range are valid member peaks regardless of their retention time. The same is true for the tightening by fraction gaps. This algorithm operates solely with the retention time dimension. Only the results of the annotation tightener are independent of the order of execution. However, this algorithm can only be applied to annotated features which represent rarely more than 10% of all features.

The selected member peaks are finally assembled into a feature. Its intensity is the sum of all member peak intensities (see Equation 12.2). All other properties such as *m/z* or retention time are propagated from the master peak.

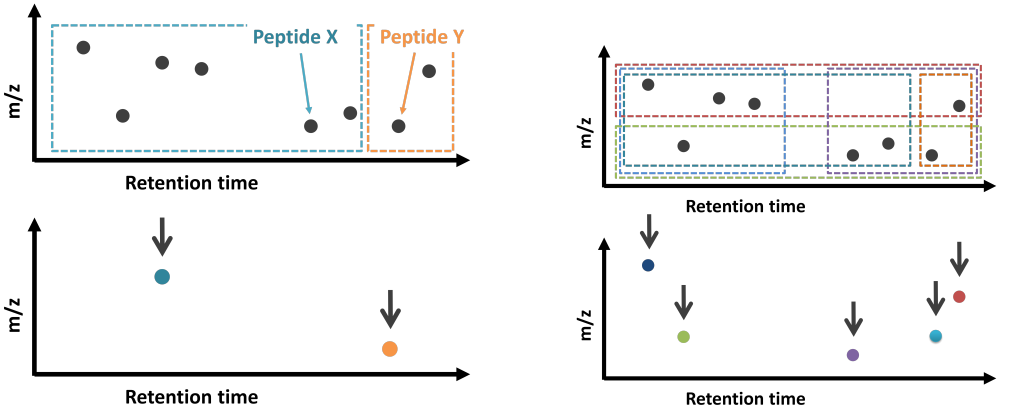
12.2.1 Details of implementation

Feature tightening is implemented following the *Strategy* or *Expert* pattern [252]. It is possible to register an unlimited number of feature tighteners to the feature extractor instance that will perform a feature tightening using the output from the previous tightener as an input to the next one, as described in the *Pipes and Filters* design pattern [252].



(a) Feature tightening by mass delta. If a feature contains a mass shift bigger than any given threshold the feature is split.

(b) Feature tightening by fraction gaps. If a feature contains "gaps", which is one or more fraction without a peak, the feature is split.



(c) Feature tightening by annotations. If a feature contains two or more peaks that contain different annotations, the feature is split so that all resulting features contain at most one unique annotation.

(d) Result of feature extraction. Combination of three feature tightening strategies result in the final features/feature boxes.

Figure 12.1: Feature extraction and feature tightening. Feature tightening (see main text) is performed using three independent strategies which refine features according to different criteria such as m/z (a), retention time (b), and annotations (c).

$$I_f = \sum_{n=0}^{N} mp_n$$

Where:

I_f	is a feature's intensity,
mp	is a feature's member peak.

Equation 12.2: Feature intensity. A feature's intensity equals the sum of all member peak's intensity values.

Importantly, a feature may contain in principle as many different annotations as it contains peaks, as the annotation process is performed prior to feature extraction (see 12.1). To avoid these problems, the annotation process is either performed *after* feature extraction or an annotation-based feature tightening is applied.

12.3 Affinity-Driven Feature Alignment

Besides feature extraction (see 4.2.1.1 and 12.2), feature alignment (see 4.2.1.2) is the second main component of MS_QBAT and of this work. Details of the implemented alignment algorithm are described below.

12.3.1 Mapping

As most feature alignment strategies, and as already described in 4.2.1.2, feature alignment is initiated by building all possible alignments for a certain peak (see Table 4.1) in a defined range of *m/z* and retention time. Both the described process and possible alignment matches identified during the process are called *mapping*. A mapping represents an unidirectional relationship between a single feature from sample `s1` and 0 to n features from sample `s2` (see Figure 12.2a).

A feature to which matches are searched is called *mapping key*, a feature that was found to be a possible match to a mapping key is called *mapping value*. When aligning `s1` against `s2`, a mapping from `s1` to `s2` is called *forward mapping*, a mapping from `s2` to `s1` *reverse mapping*.

```
public interface Mapping<K, V, A extends MappingKey<?
    extends K>, B extends MappingValue<? extends V>> {

    void addValue(B value);

    A getKey();

    List<B> getValues();

    void setValues(List<? extends B> values);

}
```

Listing 12.2: The `Mapping` interface. A mapping represents an unidirectional relationship between a single feature from sample `s1` and 0 to n features from sample `s2`.

To ensure that every feature from both `s1` and `s2` is contained in at least one mapping (identify *orphan* alignments (see Table 4.1)), the mapping process is typically performed twice. A forward mapping, i.e., a mapping from `s1` to `s2`, where all features from `s1` are mapping keys and all features from `s2` are possible mapping values is performed at first. Then a reverse mapping, i.e., a mapping from `s2` to `s1`, is performed where mapping keys and mapping values are inverted.

The result of this two-step mapping process (forward- and backward-mapping) is a collection of mappings that represent possible alignments. The number of mappings always equals the size (e.g., the number of peaks) of `s1` plus the size of `s2`.

12.3.2 Alignment path as a retention time normalization function

To correct for retention time shifts, a so-called *alignment path* is build. An alignment path represents a function to recalculate retention times according to the local retention time shifts present between two samples. Therefore, an alignment path maps retention times from sample `s1` to retention times in sample `s2` and vice versa. Applying this alignment path to all retention times in `s1` or `s2` results in retention time normalization and allows to compare retention times of two samples.

(a) Mapping. A mapping represents an unidirectional relationship between a feature from `s1` (left) and multiple features from `s2` (right). A forward mapping reflects all possible alignments for a feature from `s1`, a reverse mapping all possible alignments for a feature from `s2`.

(b) Alignment. An alignment represents a bidirectional relationship between a feature from `s1` (left) and a feature from `s2` (right).

Figure 12.2: Mapping and alignment. All alignments are build from mappings by selecting from all possible matches the most fitting one.

In order to directly calculate retention time shifts and to avoid a computationally expensive iterative approach (see 4.2.1.2), only unambiguous mappings are selected from the initial two step mapping process performed beforehand. An unambiguous mapping contains exactly one mapping value and has therefore a high probability of representing a correct pairing. Based on these unambiguous mappings, a retention time shift can be calculated with high accuracy for every retention time value.

Building the alignment path purely based on annotated features is an alternative approach not implemented so far. While this approach offers a high accuracy (finding matching feature pairs is trivial if annotations are present in both samples), it is limited by the requirement for annotations to be present in both samples.

12.3.2.1 Alignment path polishing

After building an initial set of high-confidence mappings, a certain degree of post-processing of the resulting *raw path* is required.

12.3.2.1.1 Interpolation of missing values A raw path usually does not contain mapping information for each discrete retention time value (fraction index) and therefore needs to be complemented. For every fraction index in sample `s1` an according index in sample `s2` has to be present and vice versa. In order to complement a path, a linear interpolation approach [253], which is used most prominently in the fields of computer graphics and image processing [254–258], is applied. Briefly, missing values between two successive retention time pairs are calculated assuming a linear connection between the two pairs.

As described in 4.2.1.2 and Figure 4.4, plotting retention times of two perfectly aligned samples results in a line crossing the origin and holding a slope of 1. Therefore, only an interpolation describing a line equation with a slope of approximately 1 is valid. If an interpolation between two retention time pairs results in a slope differing from 1 more than ±20 %, this interpolation is considered to be invalid. As a result, the second retention time pair is removed from the alignment path and the interpolation process is restarted.

Importantly, the path completion/interpolation process must not start at the beginning of the path (i.e., with the retention time pair having the lowest index), since (especially at the beginning and at the end of an alignment) retention time shifts might significantly differ from those in between. Based on own experiences, the path between two retention time pairs in the central part of the LC gradient is mostly constant and expressing a line equation with a slope of approximately 1. Therefore, the interpolation process is started at a randomly chosen retention time pair in the central part of the path (e.g., a retention time pair having index values between 30 – 70 % of the highest index value).

12.3.2.1.2 Path smoothening After a complete path has been built, a *smoothening* step is applied to correct for path entries considered to be outliers. Therefore, path smoothening is an outlier-removal step, invalidating path entries that express a line equation to their neighbors having a slope that differs by ±20 % from a of slope of 1.

Importantly, these outliers are present in the raw path and are therefore not affected by the interpolation criteria described above, which enforce an underlying smooth linear equation having a slope of approximately `1`.

```
public interface Alignment<A, B> extends Pair<A, B> {

    @Override
    public A getFirst();

    Mapping getMapping();

    @Override
    public B getSecond();

    AlignmentType getType();

}
```

Listing 12.3: The `Alignment` interface. An alignment represents an bidirectional relationship between two peaks/features that represent the same peptide in two aligned samples. Since its bidirectional character, the `Alignment` interface extends `Pair`.

Even when building an alignment path from annotated features only, outliers, which do not properly fit in the linear equation model describing a perfect alignment, may be introduced. This is due to a misconception that is the basis for virtually all alignment algorithms. The theory would predict that peptides will always elute from an UHPLC column in the same order. However, only if this theory is assumed to be correct, retention times can be mapped using a linear model.

Mitra *et al.* [259] showed that this assumption can be violated and that a linear order of peptide elution times cannot be always taken as granted. Data points not fitting the linear elution model are therefore rather the rule than the exception, necessitating outlier removal.

Gaps in the alignment path introduced by outlier removal are subsequently filled by the interpolation strategy descried above.

12.3.3 Building alignments

Following the identification of all possible alignments (*mappings*, see Table 4.1 and 12.3.1) the transformation of mappings into alignments is performed. An alignment is a pair of features which represent the same peptide in two different samples (see Table 4.1). Both features share a bidirectional, undirected relationship. On the contrary, a mapping is a unidirectional relationship between 1 feature and 0 to n features from another sample.

$$AS = (|f1_{m/z} - f2_{m/z}| * 1000) + (|f1_{fracnr} - f2_{fracnr}| * 2)$$

Where:

AS	the alignment score,
$f1$	first feature,
$f2$	second feature.

Equation 12.3: Alignment score. The Alignment score represents a numerical value to express an alignment's quality. It is expressed as a positive value. In case of a perfect match, the alignment score is 0. The lower the score, the more likely the alignment is correct. The affinity score is the product of the differences of two features in m/z and fraction number/retention time.

As described in 4.2.1.2, the transformation from mapping to alignment can be compared with a transformation of a multigraph into a simple graph. During the construction of alignments, the correct mapping of feature 1 from `s1` to feature x to `s2` has to be identified in order to form a proper alignment.

12.3.3.1 Ranking mapping values

In analogy to the weighting edges in a multigraph, a "mapping value weighting" is introduced in order to choose the right/best mapping value. More precisely, for each mapping value an *affinity score* is calculated that expresses a mapping value's "affinity" to its mapping key as a numerical value. The affinity score is the product of the differences of two features in m/z and fraction number (retention time) (see Equation 12.3). Fraction numbers are corrected for retention time shifts before calculating the affinity score. For this the previously calculated alignment path (see 12.3.2) is used. The alignment algorithm can optionally be configured to take available annotations into account. If this option is chosen, the score is calculated as shown in Equation 12.3, but peptide sequence, sequence modifications, and missed cleavages are evaluated as well. If one of these annotation properties does not match, an *annotation penalty* is added to the previously calculated score (1,000 per default).

The mapping value with the lowest affinity score is selected to build the alignment in combination with the mapping key.

$$sgn(x) \begin{cases} +1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

Equation 12.4: The *signum* function. In mathematics, the *signum* function is a function that extracts the sign of a real number. In computer science, it is often used to numerically compare objects. An object is "less" than another one, if the *signum* function returns less than 0, "more" than another object, if the *signum* function returns more than 0, and equal, if the *signum* function returns 0.

12.3.3.1.1 Details of implementation When sorting elements in a list, the ordering can be calculated using the *signum* function (see Equation 12.4). All elements in the list are compared to each other, where the result of this comparison is either -1, 0 or $+1$. If the result equals -1, the left hand element is "smaller" than the right hand element and should therefore have a lower position index in the list than the right hand element. Vice versa in the case of $+1$. If the result equals 0, no resorting of these two elements is needed.

An abstract *signum* function is defined by the `java.util.Comparator` interface (see Listing 12.4). Providing an implementation of `java.util.Comparator` and a list of matching types, the static method `java.util.Collections.sort(List l, Comparator c)` can be used to sort the elements in a given list ascending.

Finding the right mapping value makes use of this function. Mapped values are stored in a list. Upon alignment building, this list is sorted using an implementation of `java.util.Comparator` that compares affinity scores. Since an optimal affinity score is 0 and the ordering is ascending, the mapping value with the lowest affinity score will be at list position 0 after sorting.

12.3.3.2 Alignment types

Different types of alignments can result from the alignment building process. Usually one mapping is transformed into one alignment and different alignment types may result depending on how many mapping values were found for a mapping key.

```
public interface Comparator<T> {

    int compare(T o1, T o2);
}
```

Listing 12.4: The `Comparator` interface. The `Comparator` make types logically comparable. Its `compare(T o1, T o2)` function takes two objects of type `T` as parameters. The result is an integer. If the result is 0, the two objects are equal. If the result is less than 0, the first object is *less* than the second one. If the result is greater than 0, the first object is *greater* than the second one. The Comparator represents the Sign function and can be used to implement any comparing strategy. When sorting lists (e.g. by using `java.util.Collections.sort()`), comparison of elements is delegated to a Comparator.

Single alignments: A single alignment is constructed from a mapping with has one mapping value only. This means that only a single matching feature could be found in the mapping process in the m/z and retention time range defined.

Orphan alignments: An orphan alignment results from a mapping that contains no mapping values, i.e., no matching feature could be found in the mapping process in the m/z and retention time range defined. An orphan alignment is the result of a presents-vs-absence/"black-and-white" situation as described in 4.4.

Complete alignments: In contrast to the other alignment types a complete alignment results from two mappings. If two mappings exist for which the following holds true:
$m1_k = m2_{v1}$ and $m2_k = m1_{v1}$
where

- `m1` is the first mapping;
- `m2` is the second mapping;
- `k` is a mapping key;
- `v1` is a mapping value at list position 0 after ranking mapping values (see 12.3.3.1);

then a complete alignment is built from these two mappings.

12.3.3.3 Dummy seeding

Relative protein quantification relies on the existence of at least two features/ peptides in order to calculate a ratio of abundances. That is why no ratio can be calculated for feature `f1` without matching feature `f2`. Since such a presents-vs-absence situation can principally indicate a strong "regulation", the calculation of a specific ratio is difficult.

In order to be able to calculate protein abundance ratios for a "black-and-white" situation, so called *(mapping) dummies* are "seeded" into the sample. A dummy is a proxy to a missing feature and acts as a placeholder if no matching feature could be found. During the process of dummy seeding, all orphan alignments are "filled up" with a mapping dummy to ensure the calculation of intensity ratios for all alignments, including orphan alignments.

As described in 3.5.2 and 3.5.1, MS data contains noise or background signal. Dummy seeding allows the quantification against the background intensity and replaces non-existent values by background intensity values. Thereby, all feature intensities have a positive value and the calculation of a ratio for each alignment is possible.

12.3.4 Cleaning alignments

The building of alignments is purely based on the ranking of mapping values. Once values have been ranked, the best fitting value will is taken to build the final alignment. The ranking is mapping-local, which is why transitive effects are not considered (see 4.2.1.2). The approach taken in MS_QBAT has one major advantage compared to a complete solution, which can be obtained e.g. by a dynamic programming approach – it is extremely performant. In principle, the ranking can take place in all mappings concurrently, which makes this approach highly susceptible for parallelization. Building thousands of alignments is therefore only a matter of milliseconds.

Nevertheless, this strategy is heavily simplified and is therefore associated with one major drawback: transitive alignment connections (see 4.2.1.2) are not resolved. As a consequence, features are frequently member in more than one alignment. This is why the post-processing of alignments before their use for protein quantification is inevitable.

Therefore, after alignments have been build from mappings, each alignment is checked for peaks that are member in multiple alignments. These peaks and all corresponding alignments are collected. Since a peak must only be member in a single alignment, all but one alignment sharing the respective peak are removed from the final alignment list. All possible alignments are compared against each other in terms of alignment quality and only the highest ranking alignment is kept.

Importantly, by removing alignments, peaks can principally get "lost". If one peak is member in multiple alignments and another peak is only member of one of these alignments, the second peak is lost if the respective alignment is removed. While the first peak is not present in multiple alignments anymore, the second peak is removed from the alignment list and will be therefore not be considered in a subsequent quantification.

12.3.4.1 Details of implementation

To find the best alignment, alignments are ranked according to their quality in a similar way the mapping ranking is performed (see 12.3.3.1). `ComparatorAlignment` (see 12.5) ranks alignments according to their type (see 12.3.3.2). If the type is the same, comparison is delegated to comparison by affinity score (see 12.3). Complete alignments are of highest quality, followed by orphan alignments and single alignments.

Using `ComparatorAlignment`, a list of alignments can be sorted according to alignment quality. Post sorting, only the alignment at list index `0` is kept, all other alignments are dismissed.

```
public class ComparatorAlignment implements Comparator<
    AlignmentPeak> {

     public static int delegateToAffi(final AlignmentPeak
    o1, final AlignmentPeak o2) {
          final double affiScore1 = o1.getAffinityScore();
          final double affiScore2 = o2.getAffinityScore();
          final int compAffi = Double.compare(affiScore1,
    affiScore2);
          return compAffi;
     }

     @Override
     public int compare(final AlignmentPeak o1, final
    AlignmentPeak o2) {
          if (o1.getType().equals(o2.getType())) {
               return delegateToAffi(o1, o2);
          }
          if (o1.getType().isComplete()) {
               return -1;
          }
          if (o2.getType().isComplete()) {
               return 1;
          }
          if (o1.getType().isOrphan()) {
               return -1;
          }
          if (o2.getType().isOrphan()) {
               return 1;
          }
          if (o1.getType().isSingle()) {
               return -1;
          }
          if (o2.getType().isSingle()) {
               return 1;
          }
          if (o1.getType().equals(AlignmentType.INVALID)) {
               return -1;
          }
          if (o2.getType().equals(AlignmentType.INVALID)) {
               return 1;
          }
          throw new RuntimeException("Could not compare " +
    UtilString.NEW_LINE_STRING + o1
                    + UtilString.NEW_LINE_STRING + o2);
     }
}
```

Listing 12.5: The `ComparatorAlignment` class. Similar to evaluating alignment affinity by the alignment score, `ComparatorAlignment` is used to compare multiple alignments. Alignment quality is evaluated by alignment type, or, if alignment type is the same, by alignment score.

12.3.5 Multiple alignment

The alignment library contains all required functionality to perform superalignment-based multiple alignments such as linear alignment (see Figure 4.6a) or hierarchical alignment (see Figure 4.6b) as described in 4.2.1.2.1. While hierarchical alignments require manual selection of samples to be combined into supersamples, linear alignments can be performed automated. As described in 4.2.1.2.1, a linear alignment is achieved by subsequently aligning all input samples to a selected reference sample. The overall alignment quality depends strongly both on the alignment order and on the sample similarity. To obtain an optimal alignment order for multiple input samples, a sample similarity score for each sample to be aligned is calculated (see 12.3.6). To improve the overall performance of the process, only a fraction of the total data of each sample is used to perform these alignments. Strategies for the selection of these data parts, which have still to be representative for the original sample, have to be carefully evaluated. To estimate sample similarity for cross alignment, dismissing low intensity features from the original sample has been identified as a valuable strategy. In order to preserve data for the whole retention time range, a defined percentage of peaks per fraction is kept. Importantly, annotated features are never dismissed. From the resulting matrix of similarity scores (r^2) an alignment order is derived and samples with a high similarity are aligned first.

12.3.5.1 Details of implementation

In order to provide highly flexible and integrated support for superalignments, `AlignmentFeature` extends `Feature`. Thereby, an aligned feature can be used in all classes and methods which work with the `Feature` type. The alignment algorithm for example expects two collections of features (`sample1` and `sample2`) as input. Since `AlignmentFeauture` extends `Feature`, it is possible to provide collections of features as well as collections of alignments (see Figure 12.3).

```
public interface Feature extends ComposableElement<Peak>,
    Peak, Iterable<Peak> {

    @Override
    Feature clone();

    int getIndexCenter();

    int getIndexFirst();

    int getIndexLast();
}
```

Listing 12.6: The `Feature` interface. `Feature` inherits from `ComposableElement`, since it represents a *composition* of several peaks (member peaks). It furthermore inherits from `Peak`, since it shares all properties of a peak. The only extension to `Peak` are three different "retention times". center index equals the fraction index of the master peak (see main text). First- and last index equal the lowest- and the highest fraction index of all member peaks, respectively.

```
public interface AlignmentPeak extends AlignmentSame<Peak
    >, Peak {

    void addDummy();

    @Override
    public Peak getElement();

    MappingContext getContext();

    @Override
    MappingPeak getMapping();

}
```

Listing 12.7: The `AlignmentPeak` interface. Just like `Feature`, `AlignmentPeak` inherits from `Peak`, since it shares all properties of a peak. `AlignmentPeak` extends `Peak` by providing a method for dummy seeding (see main text). It provides further methods to obtain a peak-representation (may return `this`) that can be used for caching a peak instance and simple getters for the context of the mapping and the mapping itself.

```
public interface AlignmentFeature extends AlignmentPeak,
    Feature {

     @Override
     public Feature getElement();

     @Override
     public Feature getFirst();

     @Override
     public Feature getSecond();

}
```

Listing 12.8: The `AlignmentFeature` interface. `AlignmentFeature` represents an alignment of features. Consistently, it inherits both from `Alignment`/`Alignment-Peak` and `Feature`. `AlignmentFeature` does not provide any new methods but only overrides supertype methods to return the correct type (`Feature` instead of `Peak`).

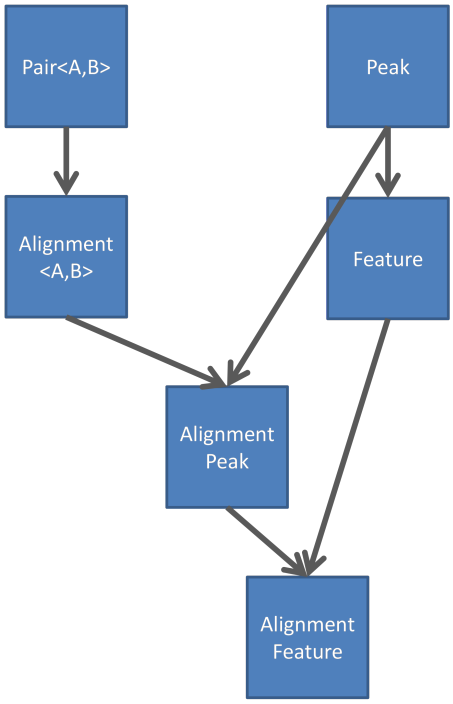


Figure 12.3: Type hierarchy of six main types. Since `Feature` extends `Peak`, all methods that require parameter(s) of type `Peak` can be provided also with objects of type `Feature`. Since `AlignmentFeature` extends `Feature`, all methods that require parameter(s) of type `Feature` or `Peak` can be provided also with objects of type `AlignmentFeature`. This way functionality to build superalignment-based multiple alignments is provided without the need for special types or methods. Instead the *polymorphism* provision in object-oriented programming enables it *a priori*.

12.3.6 Sample similarity and alignment quality

As described briefly in 12.3.5, the alignment order plays an important role for the quality of the final superalignment when performing a linear multiple alignment. Alignment quality refers to the number of correct alignments versus the number of false alignments (true positive (TP) alignments versus false positive (FP) alignments), whereas sample similarity indicates the reproducibility of the LC-MS system during data acquisition of the respective samples.

While a mostly linear shift in retention time can easily be compensated by the retention time shift correction described in 12.3.2, variances in *m/z* accuracy can significantly decrease alignment quality.

In general, high sample similarity directly translates into good alignment quality, since samples with small shifts in retention time and *m/z* are more likely to result in correct alignments than samples with larger shifts. A linear multiple alignment should therefore be initiated with the most similar sample pair and integrating samples with lower similarity at the end (see 12.3.5). To allow for the ranking of samples with respect to their similarity, the sample similarity needs to be expressed in a numerical and comparable manner. This numerical value can subsequently used to determine an optimal alignment order.

Within MS_QBAT, sample similarity is expressed by the coefficient of determination (r^2). r^2 is calculated using a common least squares regression model that is used for a linear approximation of retention times to an optimal alignment path (see Figure 4.4 and 12.3.2). As for establishment of the alignment path (see 12.3.2), only retention time pairs from unambiguous mappings are used as input values for the model. A perfect similarity gives an r^2 of 1; the higher the r^2 is, the more similar two samples are.

The assessment of the alignment quality is relatively easy performed if both samples contain annotations. For this purpose, all alignments containing two annotated features are checked with respect to matches in peptide sequence, cleavage sites, and modifications. If all three parameters match, the alignment is marked as correct; if not, it is identified as a false alignment. The number of correct alignments minus the number of false alignments allows to establish a good numerical representation of alignment quality.

$$r_{pep} = \frac{I_{fl}}{I_{fr}}$$

Where:

r_{pep}	is the intensity ratio for a peptide,
I_{fl}	is the intensity of alignment's left feature,
I_{fr}	is the intensity of alignment's right feature.

Equation 12.5: Peptide ratio. A peptide ratio is calculated by dividing an alignment's left feature intensity by its right feature intensity.

However, this approach is only practicable if both aligned samples contain annotated features. Additionally, no qualitative assessment of alignments is possible with samples containing only non-overlapping annotated features. However, sample similarity can always be assessed and allows for a good estimation of alignment quality since similar samples are likely to exhibit a high alignment quality (see above).

12.4 Peptide and Protein Quantification

The input data for the quantification algorithm implemented within MS_QBAT are aligned samples (i.e., a collection of alignments (see 12.3.3)). A peptide ratio is calculated from each annotated alignment by dividing the intensity of the alignment's first feature by the intensity of the alignment's second feature (see Equation 12.5 and Figure 12.4). To calculate a quantification ratio for a protein, all peptides belonging to the same protein are collected. Subsequently, the summed intensities of all left features from all alignments are divided by the sum of the intensities of all right features from all alignments (see Equation 12.6 and Figure 12.4).

To calculate a peptide ratio and assign it to a certain protein, all alignments are considered that contain at least one annotation. As described before (see 4.2.1.2, 12.3 and 12.5), if only one feature of an alignment is annotated, this annotation is considered to be valid for the other feature as well.

If both features contain annotation information, but these annotations do not match, the regarding alignment is incorrect and therefore not considered for quantification.

$$R_{prot} = \frac{\sum_{n=0}^{N} I_{fl_n}}{\sum_{n=0}^{N} I_{fr_n}}$$

Where:

R_{prot}	is the intensity ratio for a protein,
I_{fl}	is the intensity of alignment's left feature,
I_{fr}	is the intensity of alignment's right feature.

Equation 12.6: Protein ratio. A protein ratio is calculated by dividing the sum of all left peptides/alignments intensity by the sum of all right peptides/alignments intensity.

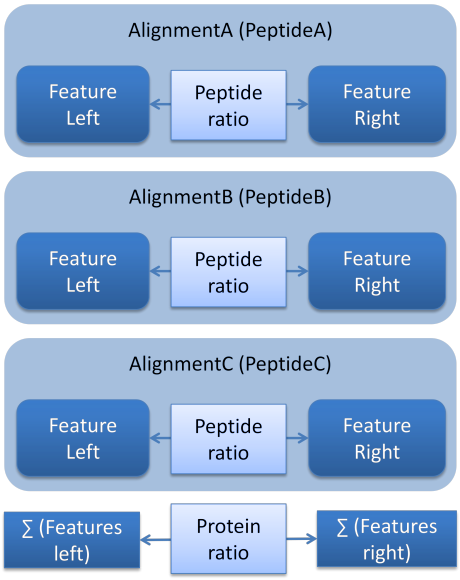


Figure 12.4: Peptide- and protein ratios. Graphical illustration of the ratio calculation for peptides and proteins. As each alignment represents one peptide, a peptide ratio is calculated by dividing an alignments left feature intensity by its right feature intensity. A protein intensity is the sum of all regarding peptide intensities. Consistently, a protein ratio is calculated by dividing the sum of all left feature intensities by the sum of all right feature intensities.

12.5 Annotation Propagation

Annotation propagation might be the biggest advantage of MS^1 -based quantification strategies. Within an alignment of two features, the identification information attached to one of these features can be propagated to the aligned feature without annotation information. This implies that upon quantification the annotation of the first feature is also valid for the second feature in the alignment. Furthermore, if several samples are aligned to each other, a peptide identification information present as an annotation of one of the aligned features is also valid for all other features of this alignment.

MS_QBAT provides the possibility to explicitly propagate annotations, allowing for the propagation of annotations within an alignment of two or more samples. Thereby, peptide identifications can be transferred from one or more samples to one or more other samples.

Additionally, annotation propagation can be utilized in more advanced alignment scenarios. If the total number of annotations is unbalanced between samples, it might be desirable to transfer annotations between these samples via annotation propagation prior to final alignment and quantification. Parameters for the alignment performed for annotation propagation must be chosen very tight, since false positive alignments would lead to false annotations, which in turn would negatively affect subsequent alignment and quantification. Choosing very tight alignment parameters minimizes false positive alignments at the cost of increased false negative alignments. Unlike false positive alignments, false negative alignments do neither affect subsequent alignment nor quantification. The only drawback of an increase in false negative alignments is the reduced possibility for (correct) annotation propagation.

12.6 Sample Groups

In a LC-MS workflow, every run generates one sample that is a coherent entity technically independent from other samples. In a GeLC-MS workflow, an additional layer of complexity is introduced. Samples are initially separated by SDS-PAGE and the gel is subsequently cut into multiple gel slices (see 3.1.2). Each of these slices is separately digested and processed by LC-MS. Since each slice represents only a subset of the initial, biological sample, each LC-MS sample in return is biologically meaningless without the context of the corresponding other samples. In contrast to LC-MS samples, GeLC-MS samples share an additional *additive* relationship to each other. This relationship needs to be considered while processing the respective data.

MS_QBAT introduces *sample groups* in order to account for this relationship. Different samples may be grouped together in order to reflect an additive relationship between samples. Sample groups can be processed just like "normal" samples and be ungrouped at any time.

Importantly, sample groups are quantifiable. The quantification algorithm accounts for the additive nature of sample groups. Thereby, the quantification of GeLC-MS data is enabled in a very intuitive way.

13

Additional Quantification Components

Besides the described cornerstones of label-free peptide and protein quantification, this work includes some additional components that are not essential to a label-free quantification but help to improve it significantly. Together with the cornerstones of label-free quantification, these components define MS_QBAT's range of features and are described in detail in this chapter.

13.1 Local Intensity Normalization via Spiked-in Internal Standard Peptides

The offline coupling of LC and MS in a typical LC-MALDI-MS setup enables the spike-in of internal standard peptides into each of the sample fractions. Beside the application of these spike-in peptides for real-time *m/z* calibration of each MS spectrum, they can be used for fraction-wise (local) intensity normalization [230]. Local intensity normalization is beneficial since LC-MS data is subject to diverse local and systematic variations. Local variations can result for example from ion-suppression effects (peak intensity is lowered or peaks are missing), detector saturation effects or intensity values approaching background levels [229]. In order to perform fraction wise, local intensity normalization, an algorithm was implemented performing the following steps for all fractions of a sample.

13.1.1 Defining standard peptides

Depending on the standard peptides used, corresponding *m/z* values are loaded (e.g. from a xml- or plain text file). For each *m/z* value, an intensity value needs to be provided. Intensities can be freely chosen, as they only function as a reference value for the subsequent normalization process. Listing 13.1 shows a configuration file that was created to use the four standard peptides specified in Table 8.1.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM
   "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>Standard peptides used for fraction-wise intensity
   normalization</comment>
<entry key="intst-0-height">10000</entry>
<entry key="intst-0-mass">1411.7220</entry>
<entry key="intst-1-mass">1846.8730</entry>
<entry key="intst-1-height">2500</entry>
<entry key="intst-2-height">500</entry>
<entry key="intst-2-mass">2155.0230</entry>
<entry key="intst-3-height">8000</entry>
<entry key="intst-3-mass">950.4950</entry>
</properties>
```

Listing 13.1: Example file for internal standard definition. Shown is a configuration that uses three standard peptides. 1 to n standard peptides may be defined. A standard peak is defined by a m/z value that is defined by the calculated mass of the used standard peptide and a user-chosen intensity value.

13.1.2 Identifying standard peptides

In each sample fraction peaks representing the predefined standard peptides have to be identified. For each m/z value of the standard peptides, an m/z -filter is applied to all peaks selecting for those peaks matching the user-defined range of tolerance. With regard to the standard settigns of MS_QBAT, this tolerance was set to ± 0.25 Da of the *in silico* determined peptide m/z value. Intensity values for identified standard peptide peaks are overridden by the corresponding user-defined intensity values.

For the remaining part of this chapter, the term *standard* will be used to describe a peak representing an internal standard peptide, whereas *peak* will be used to describe all other peaks representing all other signals in a spectrum.

13.1.3 Finding a normalization factor

The specific way how the normalization factor is calculated depends on the number of standards identified before:

No standards: if no standard could be identified no normalization is performed. This could be the case if no standards could be extracted due to ion-suppression effects (see 3.5.2).

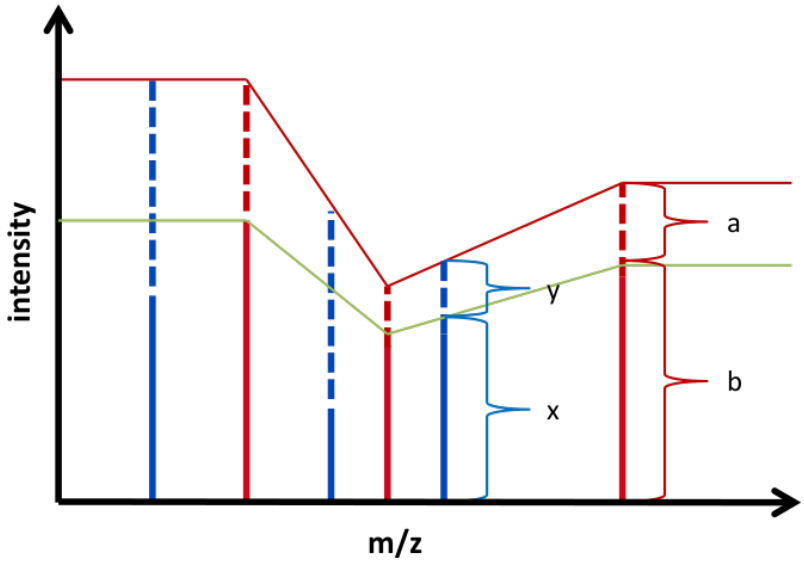


Figure 13.1: Fraction wise intensity normalization. During the spotting process, in each fraction a set of standard peptides is spiked-in. All peak intensities in this fraction are normalized to the standard intensities in this fraction. Standard intensities are set to a user-defined value. Blue: fraction peaks, red: fraction standards. $nf = \frac{a}{b} = \frac{y}{x}$, where nf is normalization factor, a is standard's set intensity, b is standard's measured intensity, x is peak's measured intensity, y peak's normalized intensity.

One standard: in the special case that only one standard could be found in a fraction, the normalization factor is the same for all peaks in this fraction. The normalization factor nf is calculated as the ratio between a and b ($nf = \frac{a}{b}$), where a is the standard's user-defined intensity, and b is the standard's measured intensity.

Two and more standards: if more than one standard could be found, a and b are calculated for each *m/z* value separately using a theoretical standard. The *m/z* value of this theoretical standard corresponds to the *m/z* value of the peak whose intensity has to be normalized. In order to obtain the respective intensity values, the distance between the intensity of the adjacent standard with a smaller *m/z* value compared to the peak and the intensity of the adjacent standard with a higher *m/z* value compared to the peak is calculated. If the *m/z* values of all standards are smaller or larger than the peak's *m/z* value, the one standard strategy is applied, using the standard with the most similar *m/z* value.

```
public interface NormalizationStrategy {

    public static enum TYPE {
        INTENSITY, INTENSITY_TO_NOISE
    }

    TYPE getType();

    Sample normalize(Sample sample);

    void setType(TYPE type);
}
```

Listing 13.2: The `NormalizationStrategy` interface. Shown is the interface needed to be implemented by any normalization strategy. It provides the main method `normalize(Sample sample)` as well as getter and setter for the type of normalization (intensity- or singal-to-noise can be used for a normalization).

13.1.4 Applying the normalization factor

The normalization itself is the last step to complete the fraction-wise intensity normalization. The normalization consists of a recalculation of all peak's intensity values using the respective normalization factor. Therefore, all fractions can be normalized by multiplying all peak intensities with the calculated normalization factor (see 13.1.3).

13.1.5 Details of implementation

The normalization algorithm including all Java types is provided in the form of two OSGi plugins, which can be easily integrated both into simple command-line-only applications and into more complex, GUI enabled RCP applications. The first plug-in contains all types that represent the API of the normalization library. The second plug-in provides all classes that implement the normalization API and provide the concrete normalization functionality. Main types are the `NormalizationStrategy` interface (Listing 13.2) and the `NormalizationStrategyOnStandards` interface (Listing 13.3). Intensity values or S/N values may be used for normalization. Handling fractions in which no standards could be found is delegated to the interface `NoStandardsFoundStrategy`, which itself implements `NormalizationStrategyOnStandards`. Thereby, different strategies can be implemented to handle fractions without standards.

```
public interface NormalizationStrategyOnStandards extends
    NormalizationStrategy {

    NoStandardsFoundStrategy getNoStandardsFoundStrategy();

    List<Standard> getStandards();

    Fraction normalize(Fraction fraction);

    List<Fraction> normalize(List<? extends Fraction>
    fractions);

    void setStandards(List<? extends Standard> standards);
}
```

Listing 13.3: The `NormalizationStrategyOnStandards` interface. The definition of `NormalizationStrategy` is very generic and applies both for local- as well as global normalization strategies. `NormalizationStrategyOnStandards` extends `NormalizationStrategy` by characteristics that are specific to a local-, fraction-wise normalization, i.e., getters and setters for standard definitions and a reference to a `NoStandardsFoundStrategy`, to which normalization is delegated in case no standards could be identified.

13.2 Global Intensity Normalization

In a label-free quantification approach samples are not multiplexed but processed separately. This has some major advantages as described in 4.3 but on the other hand has one disadvantage as well. The data is subject to global, inter-sample variations that affect all three dimensions of a peak; (i) m/z, (ii) retention time and (iii) signal intensity.

Variations in m/z are corrected by calibrating the instrument to internal standard peptides as described briefly in 13.1.

Correcting for retention time shifts is the topic of feature alignment as described in 4.2.1.2 and 12.3.

To correct for these variations in terms of signal intensity, sample intensities are normalized globally to a fixed value. In order to do so, as a first step, a normalization factor is calculated (see Equation 13.2). Applying the normalization is simply done by multiplying every peak's intensity with this normalization factor.

$$I_s = \sum_{n=0}^{N} I_{p_n}$$

Where:

I_s	is sample intensity,
I_{p_n}	is peak intensity.

Equation 13.1: Sample intensity. Sample intensity is the sum of all peaks/features in this sample.

$$nf = \frac{I_s}{I_{fixed}}$$

Where:

nf	is the normalization factor,
I_s	is the actual sample intensity,
I_{fixed}	is a fixed, arbitrary intensity value (normalization intensity).

Equation 13.2: The normalization factor. In a very simple case of global normalization, the normalization factor is a factor describing the difference between actual sample intensity and desired sample intensity. Applying this factor to all peaks/features in the sample will result in a total sample intensity of I_{fixed}.

When quantifying a collection of samples against each other, it is reasonable to normalize all samples against the same normalization intensity. This way their intensity values become comparable. When omitting this normalization step, quantification can be expected to be shifted; more precisely, calculated ratios are biased in one direction or the other and the average of all ratios is different from 1.

$$I_g = \sum_{n=0}^{N} I_{s_n}$$

Where:

I_g	is the actual sample group intensity,
I_s	is the actual sample intensity.

Equation 13.3: Sample group intensity. Intensity value of a sample group is simply the sum of all sample intensities in this group.

$$nf = \frac{I_g}{I_{fixed}}$$

Where:

nf	is the normalization factor,
I_{fixed}	is a fixed, arbitrary intensity value (normalization intensity).

Equation 13.4: Normalization factor for sample groups. Calculation of a normalization factor for a sample group is not different from the calculation of a normalization factor of a "normal" sample, since this factor as well can be applied directly to all peaks/features in a sample group (i.e., all peaks/features from all samples in this group).

13.2.1 Normalizing sample groups

If multiple samples do no represent independent units of data but rather sub-units of a larger entity, as it is the case for GeLC-MS data, it is not desirable to normalize all samples independently from each other but to perform a group-wise normalization instead (see 3.1.2 and 12.6). In this case, the normalization factor is not calculated using I_s (see Equation 13.2) but rather using the sample group's intensity (see Equation 13.3).

13.3 Dynamic Complexity Reduction

Sample complexity plays an important role in (label-free) protein quantification. It has a direct influence on both the computational requirements as well as the time that is needed to perform a quantification. More important, it can also affect quantification accuracy since with higher sample complexity also the chance for false positive alignments increases. It is therefore reasonable to reduce sample size and complexity as far as possible without altering a sample's special characteristics.

MS_QBAT is able to reduce a samples complexity without losing its characteristics that are crucial for a quantification. MS_QBAT can be configured to dismiss low intensity features or features that span only over a small number of fractions. Requirement for this is the existence of annotations. Annotations are used to define a threshold below which features might dismissed without negative effects on quantification. More precisely, thresholds on feature intensity and -length are decreased iteratively as far as no feature annotations are lost. This way MS^2 data can be used to separate true signal from noise in MS^1 data.

13.4 GA-based Alignment

Affinity-driven feature alignment (see 12.3) is characterized mainly by two parameters: (i) an initial search space defined by a retention time range and a m/z range (see 12.3.1) and (ii) a maximum difference between actual and corrected retention time of an alignment (deviation from corrected retention time, see 12.3.2) Since these parameters can only be evaluated empirically they are a classical example for *magic numbers*. As described before (see 5), in such a case optimal values can be evaluated using optimization heuristics, for example by GAs. A requirement for the application of GAs is the possibility to evaluate the quality of a solution numerically. If an alignment was build from two samples that both contain annotated features, the quality of this alignment can be easily defined by the number of correct alignments in terms of matching annotations (see 12.3.6). It is therefore possible to optimize alignment parameters mentioned before by a GA.

The default build of MS_QBAT contains a plug-in which can be used to automatically find optimal alignment parameters by a GA. The plug-in uses the GeneticAlgorithms library to provide general GA functionality.

13.5 Advanced Data Aquisition

The offline coupling of LC and MS within LC-MALDI-MS based workflows offer unique unique possibility for advanced data acquisition strategies not possible with a typical ESI workflow. MS_QBAT provides two basic tools that enables the user to take advantage of thereof.

13.5.1 Exclusion list generator

Avoiding repeated peptide identifications can be reasonable for different reasons when processing multiple samples deriving from the same starting material or from a similar cohort. First, in a typical LC-MALDI-MS workflow, the acquisition time required for MS^2 data acquisition is the limiting factor. Second, sample material is restricted, which is why the number of MS^2 scans per sample fraction is limited too.

Since MS^2 peptide identifications can be propagated from one sample to another (see 12.5), acquisition time and sample material can be saved by excluding peptides identified in a previously analyzed sample from MS^2 analysis of samples that are subsequently analyzed.

The AB SCIEX TOF/TOF™ 5800 can be configured to exclude certain precursor masses when selecting parent ions for MS^2 analysis. For this purpose, a tab-delimited list of precursor masses and retention times can be imported to Series Explorer™. MS_QBAT can generate these lists from any sample. For this, the m/z and retention time from every annotated feature in a sample is copied to the exclusion list. The retention time is calculated from a fraction-wise system (i.e., a system with binned fractions) to seconds if necessary (in this case, requesting a conversion factor from the user (e.g. one fraction equals four or eight seconds)).

An exclusion list in combination with annotation propagation can be useful to increase the number of identified peptides for a cohort of samples. Instead of repeated identification of high abundant proteins, MS^2 capacities are used to identify lower abundant-less intense peptides-ions. Instead of identifying high abundant proteins over and over again, MS^2 capacities are used instead to identify also lower abundant-less intense peptides-ions.

13.5.2 Inclusion list generator

Beside the above described exclusion lists, the AB SCIEX TOF/TOF™ 5800 system supports inclusion lists for MS^2 data acquisition. In this case, only precursor ions are selected for MS^2 analysis matching masses and retention times provided in the inclusion list.

As described before, in a LC-MALDI-MS workflow MS^1 and MS^2 data acquisition are independent from each other and samples are not lost during the analysis. Sample material is consumed but sample spots usually provide enough material to reanalize. After an initial quantification, it is therefore possible to explicitly search for additional peptides from certain proteins of interest in a second MS^2 run.

MS_QBAT generates inclusion lists for such proteins by performing the following steps: (i) the software retrieves a protein's amino acid sequence from a FASTA database. (ii) this sequence is digested *in silico* to obtain all putative tryptic peptides. (iii) peptide masses are subsequently written to the inclusion list. (iv) The corresponding retention times are either defined to cover the full range of the chromatographic separation or are predicted using tools such as SSRCalc [260] or others [261–265].

Inclusion lists can help to improve protein quantification with regard to both significance and accuracy.

The combination of annotation propagation and inclusion-/exclusion lists allows for a targeted data acquisition using MALDI-MS. This characteristic is unique to MALDI-MS and not applicable in an ESI-MS workflow.

14

Bridging Worlds

Besides the quantification of LC-/GeLC-/MALDI-MS data, MS_QBAT has also been successfully used to quantify data deriving from ESI-MS, which is one of the features that are unique to MS_QBAT.

To allow the software to analyze ESI-derived data, the two main differences that distinguish ESI-MS data from MALDI-MS data had to be compensated for. Firstly, MALDI-MS retention times are expressed as integer values, since samples are fractionated after LC into discrete spots on a MALDI target plate (see 3.1.2). By contrast, ESI retention times are represented by floating point values, accounting for the continuous nature of ESI retention times. Secondly, MALDI results mainly in single-charged peptide ions, whereas ESI produces predominantly doubly- and triply-charged peptide ions (see 3.1.3).

Different charge states can easily be compensated for by respective software such as DeconTools [266]. DeconTools produces output files containing peak lists similar to the ones exported from DataExplorer.

When processing MALDI-MS data, in principle both monoisotopic molecular weight as well as m/z can be used as a qualifier describing a peak's mass. For ESI-MS, only the monoisotopic molecular weight can be used, since ions may carry multiple charges. Importantly, this difference has to be taken into account when matching annotation information from MS^2 data (see 12.1).

In principle, two different approaches can be considered for the adaptation of ESI-MS retention time values. Continuous retention time values can be binned to fit existing algorithms that expect integer retention time values as in the case of MALDI-MS data. The respective bin size can be chosen arbitrarily; an ESI-like, effectively continuous retention time distribution can be achieved by choosing small bin sizes.

Alternatively, spectra numbers, which indicate the spectrum where a peak was detected in, can be used for the adaptation of the retention time. Importantly, MS^1 and MS^2 spectra are numbered consecutively, complicating retention time qualification by a spectrum number in mainly two ways:

1. When mapping MS^2 retention times to MS^1 retention times during the process of feature annotation (see 12.1), a calculated precursor ion must be matched to a measured precursor ion. Thereby, the experimentally measured precursor's spectrum number is smaller than the calculated precursor ion's spectrum number. Importantly, the difference between MS^2- and MS^1-scan number is not necessarily equal to 1.

2. Feature alignment operates on MS^1 data and expects retention times/ fraction numbers to be consecutive. As described before, ESI derived MS^1 data does not match this requirement, since spectra numbers are not consecutive when processing MS^1 data only.

MS_QBAT is able to adapt to these ESI-MS specific characteristics.

1. MS_QBAT's annotation algorithm matches calculated- and measured precursors by similarity in terms of m/z and retention time/fraction number, since a direct mapping is often not realizable. The search space used for mapping can be configured very precisely; it is possible to define a search space that includes only a range of preceding values in terms of retention time. When processing ESI-MS data and utilizing spectra numbers as a retention time qualifier, the search space can be configured accordingly.

2. MS_QBAT provides the possibility to "normalize" non-consecutive retention time values and convert them into a consecutive layout. After applying this fraction normalization to ESI-MS data, the data can subsequently be processed and aligned just like MALDI-MS data.

15

MS_QBAT

As outlined before, the described algorithms for label-free protein quantification (i.e., (i) fraction-wise intensity normalization, (ii) feature extraction, (iii) feature alignment, and (iv) protein quantification) have been implemented as independent libraries usable in any Java application.

Since usability of bioinformatic algorithms and software plays an essential role with regard to their daily application (see 6), the quantification libraries as well as the additionally developed libraries have been incorporated into one software package called *MS_QBAT*. MS_QBAT not only provides the essential functionality needed to perform a label-free peptide and protein quantification experiment, but also is equipped with all the "helper" functionality required to provide a convenient and easy usage.

MS_QBAT's main features, which go beyond the basic requirements of protein quantification, are described in more detail in this chapter.

15.1 Data Reading

MS_QBAT supports different files types from which data can be read:

- A tab-delimited plain text format to read peak lists.
- A tab-delimited plain text format to read peptide identifications.
- The mzml format [267] to read peak lists.
- The own qbt format, which is used for high performance data reading and writing as well as export, for example, to Microsoft® Excel.

15.1.1 Details of implementation

Data reading and writing is handled by a separate RCP plug-in, which itself delegates data processing to registered converters. In doing so, the implementation of additional converters supporting further file types can be achieved easily. Per default, converters for the above mentioned file types are already installed.

Following the visitor pattern [252], the I/O plug-in hands over an input stream to each of the registered converters. If one converter fails to read the file (indicated by throwing a `FileTypeException` or `FileFormatException`), the stream reader position is reset and the file stream is handed over to the next converter. If the last converter also throws an exception while trying to read the file, a final `IOException` is thrown by the I/O plug-in, indicating that no converter has been registered that is able to read the specified file.

15.2 Data Visualization

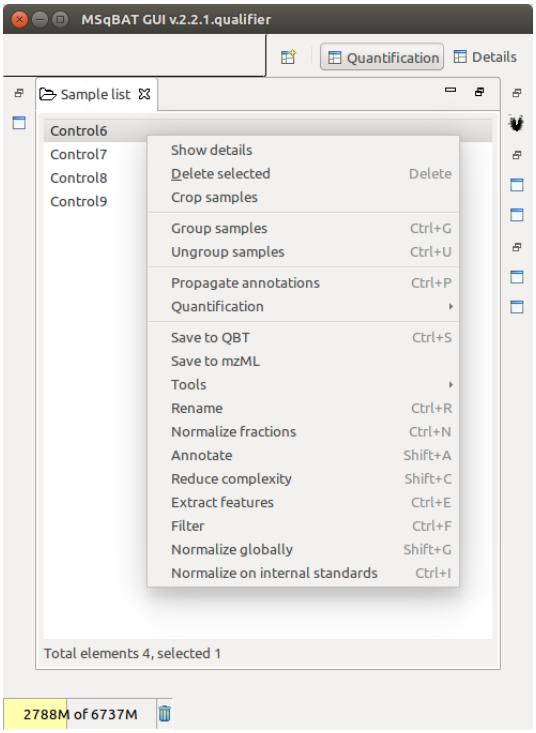
RCP supports so-called *perspectives* and *views*. Multiple views can be grouped into one perspective. Perspectives are usually context-dependent unions of views. MS_QBAT provides two perspectives; first the "Quantification" perspective, which provides several views enabling the user to perform a quantification as well as to visualize the results. Furthermore, the "Details" perspective has been implemented. This perspective can be used to further inspect samples, e.g., to visualize extracted features and their corresponding member peaks.

15.2.1 Views

MS_QBAT provides in total eight different views, which are per default grouped into two different perspectives.

15.2.1.1 "Sample List" view

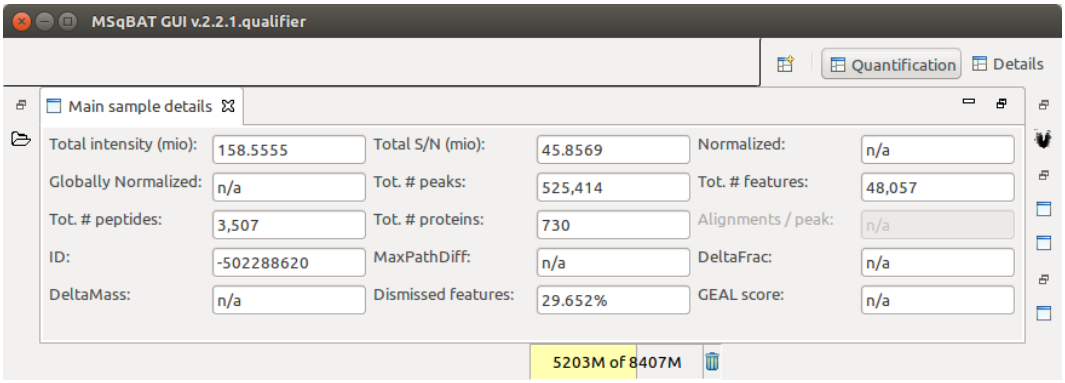
The "Samples List" view is a simple list displaying all currently loaded samples. It provides a right-click context menu containing all the commands available. These commands (e.g., feature extraction, normalization or alignment) can be applied to one or more samples. Furthermore, the "Sample List" view provides drag&drop functionality to load samples by dragging folders or files into the "Samples List" view (see Screenshot 15.1).



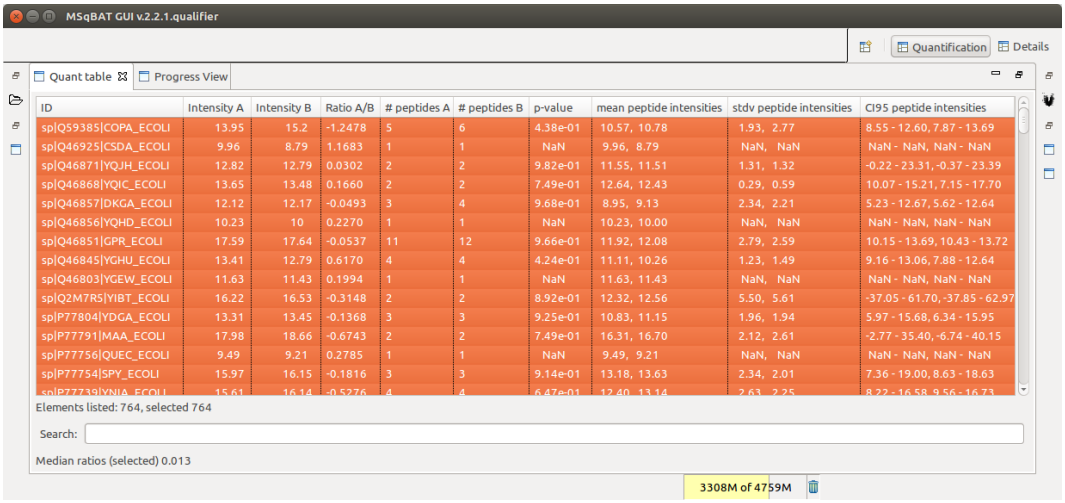
Screenshot 15.1: The "Sample List" view. The "Sample List" view provides drag&drop functionality for easy sample loading. Loaded samples are listed. A right-click-context-menu provides easy access to sample processing commands such as feature extraction, alignment or quantification.

15.2.1.2 "Sample Details" view

The "Sample Details" view displays key properties of the currently selected sample(s), for example total number of peaks, features, proteins, and peptides. Additionally, it displays a total sample intensity, a signal-to-noise ratio of selected samples, and information whether a normalization has been performed or not (see Screenshot 15.2).



Screenshot 15.2: The "Sample Details" view. The "Sample Details" view displays information on key properties of the selected sample(s) such as number of peaks, features, annotated proteins, and peptides.



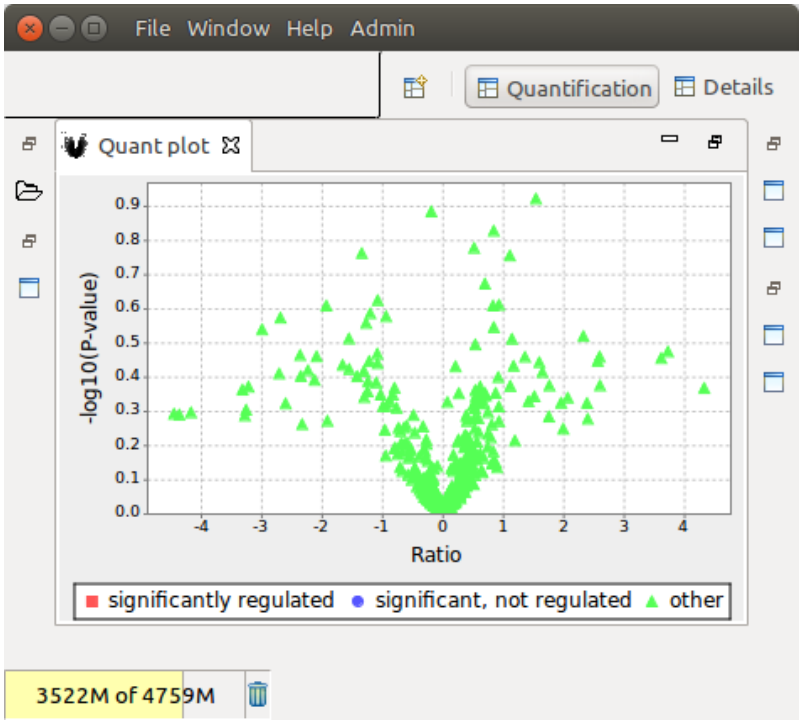
Screenshot 15.3: The "Quant Table" view. The "Quant Table" view is a tabular representation of a quantification result. It displays, for example, information on protein identifier, intensity values, quantification ratio, and p-value. Furthermore, the median ratio of all selected entries is displayed in the lower left corner. This can help to get a first impression of quantification trends. Full-text search and table sorting (by clicking on column headers) is supported.

15.2.1.3 "Quant Table" view

The "Quant Table" view displays a tabular representation of the quantification results. Columns are protein identifier of sample A and sample B, calculated ratio, number of peptides per protein, a p-value, and several additional properties (see Screenshot 15.3).

15.2.1.4 "Quant Plot" view

The "Quant Plot" view is a graphical representation of a quantification experiment. It is a scatter plot and each dot represents one quantified protein. On the x-axis, the calculated ratio is plotted (log-scale), whereas on the y-axis, the $-\log_{10}$ of the calculated p-value is displayed (see Screenshot 15.4). This type of graph is commonly referred to as *volcano plot*.



Screenshot 15.4: The "Quant Plot" view. The "Quant Plot" view is a graphical representation of a quantification result. Every dot represents one quantified protein. Ratios are plotted on the x-axis, $-log_{10}$ of the calculated p-values on the y-axis (volcano plot). Dots are color-coded: red (regulation smaller -2 fold or larger 2 fold and regulation significance smaller than 0.01), blue (regulation between -2 fold and 2 fold and regulation significance smaller than 0.01), and green (anything else).

15.2.1.5 "Alignment Path" view

The "Alignment path" view is a graphical representation of an alignment. It is a scatter plot and each dot represents one alignment. On the x-axis, the fraction index of the alignment's first feature is plotted, whereas on the y-axis, the fraction index of the alignment's second feature is displayed. Additionally, the plot contains all retention time pairs of the calculated alignment path (see Screenshot 15.5).

15.2.1.6 "2D Peptide Map" view

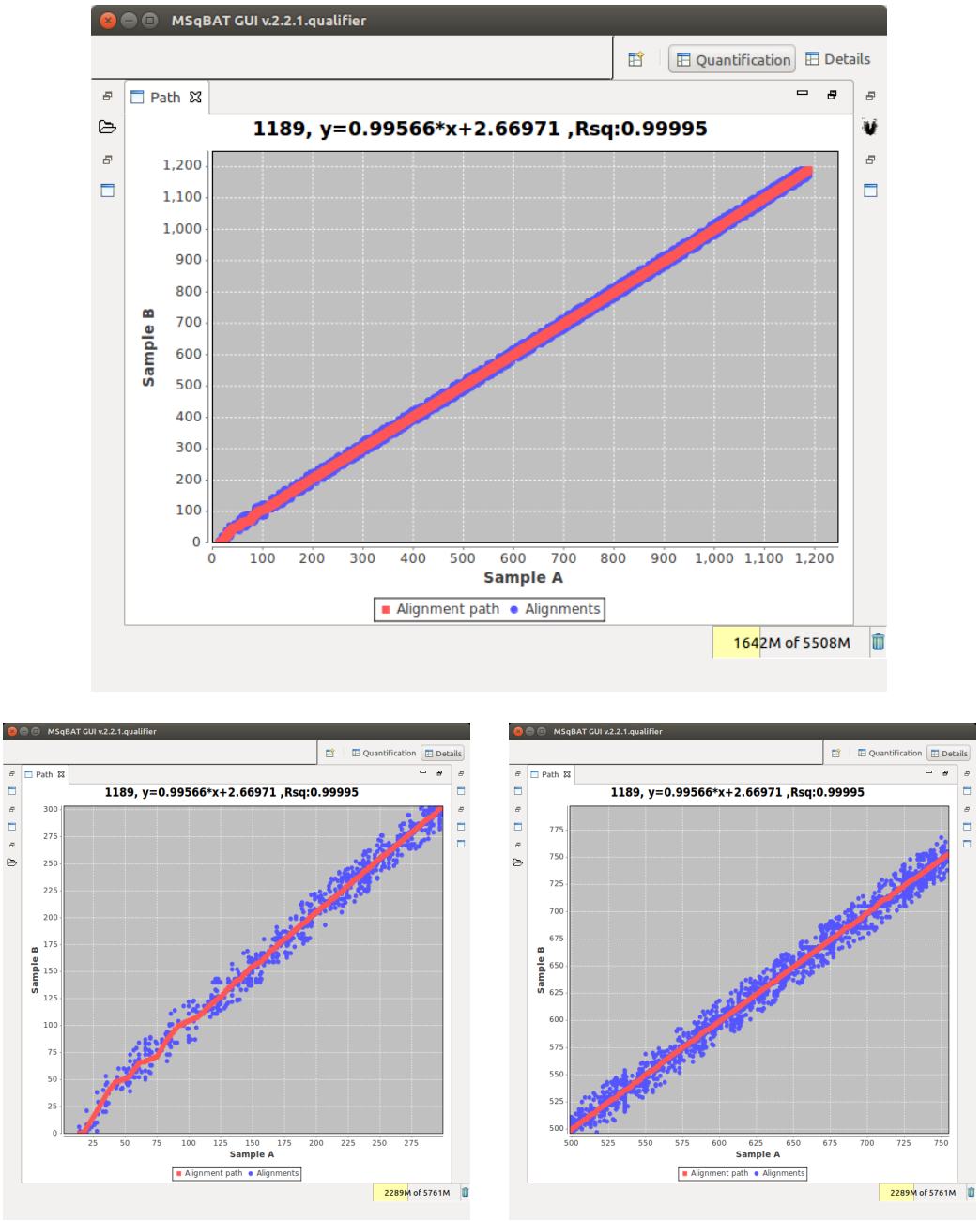
The "2D Peptide Map" view is a graphical representation of a sample and its peaks and/or features. It is a scatter plot and each dot represents one peak or feature. Peaks and features are plotted using different colors, so that all features and all contained peaks can be displayed at the same time. On the x-axis of the plot, a peak's or feature's retention time is shown; on the y-axis, its *m/z* value is displayed (see Screenshot 15.6). Like any other plot in MS_QBAT, the "2D Peptide Map" is zoomable and scrollable. In addition, the "2D Peptide Map" is coupled to the "Peak Table" view and the "2D Intensities Map" view. Therefore, zooming or scrolling is reflected in the other two views as well (e.g., only peaks that are displayed in the "2D Peptide Map" are visible in the two other views).

15.2.1.7 "2D Intensities Map" view

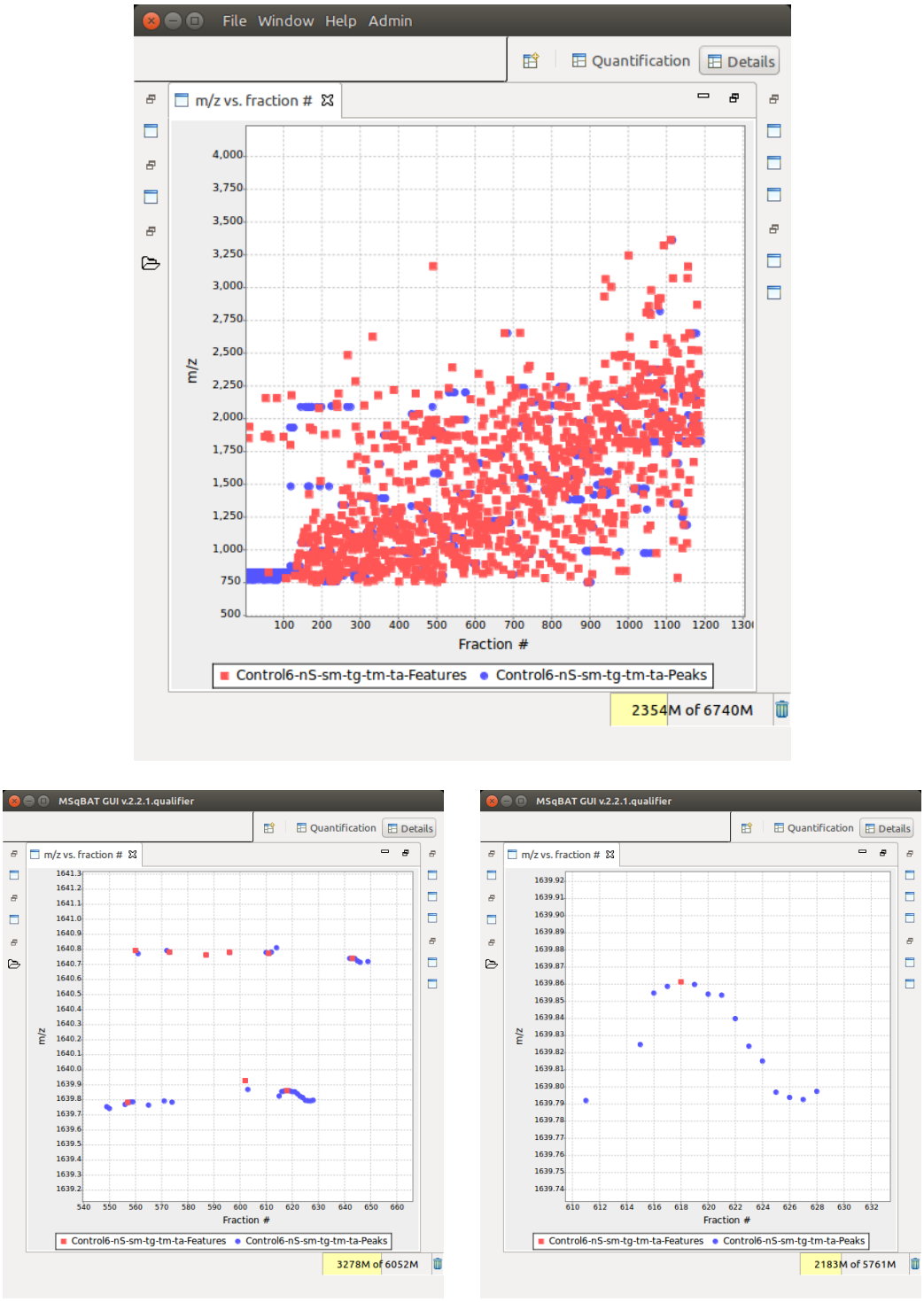
The "2D Intensities Map" view is a graphical representation of a sample and its peak and/or feature intensities. It is a scatter plot and each dot represents one peak or feature. Peaks and features are plotted using different colors; therefore, all features and all contained peaks can be displayed at the same time. On the x-axis of the plot, a peak's or feature's retention time is plotted; on the y-axis, its intensity value is displayed.

15.2.1.8 "Peak Table" view

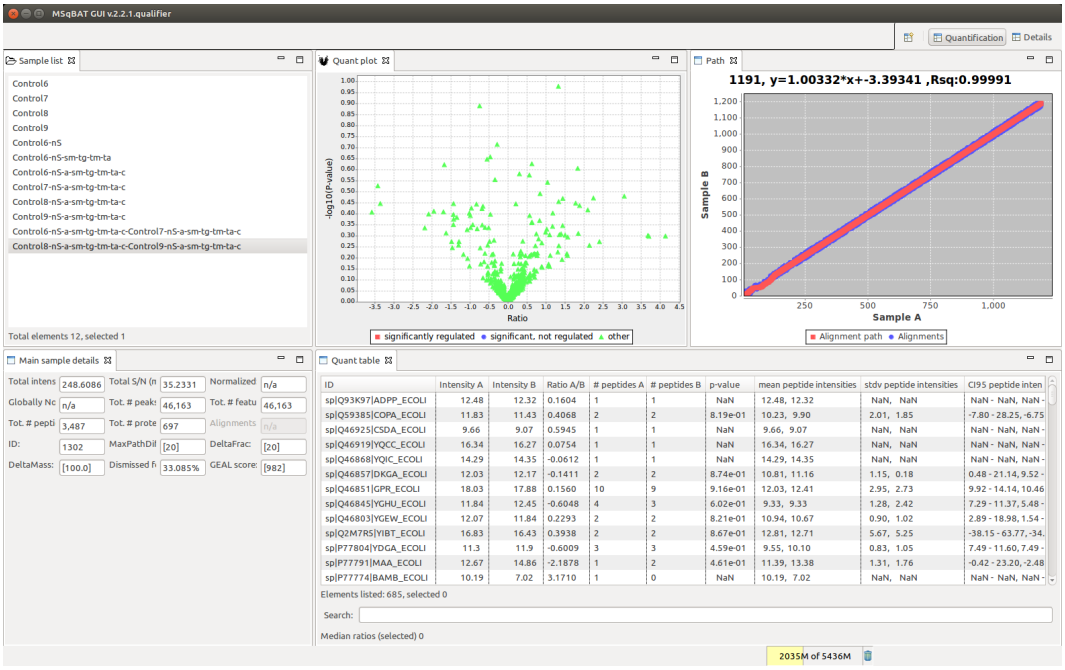
The "Peak Table" view is a tabular representation of a sample and its peaks or features. Columns provide various information about a peak's/feature's properties, such as retention time, *m/z*, and intensity values, as well as annotation information if available.



Screenshot 15.5: The "Alignment Path" view. The "Alignment Path" view is a graphical representation of an alignment. Sample A's fraction indices are plotted on the x-axis, sample B's fraction indices on the y-axis. Each blue dot represents an alignment. Red squares indicate the alignment path.
The lower two screenshots show a zoom-in of the graph on top. These two screenshots nicely illustrate LC-MS reproducibility, which usually varies most at the very beginning of sample separation by LC.



Screenshot 15.6: The "2D Peptide Map" view. The 2D Peptide Map view is a graphical representation of a sample's peaks and features. Fraction indices are plotted on the x-axis, m/z values on the y-axis. Blue dots indicate peaks; red dots show features. The map is zoomable and scrollable by mouse; the lower two screenshots show a zoom-in of the graph on top at different zoom scales. They illustrate eluting peptide patterns and the union of different peaks into single features.



Screenshot 15.7: The "Quantification" perspective. The "Quantification" perspective groups different views together. Per default, the "Sample List" view, the "Sample Details" view, the "Quant Plot" view, the "Quant Table" view and the "Alignment Path" view are shown in this perspective.

15.2.2 Perspectives

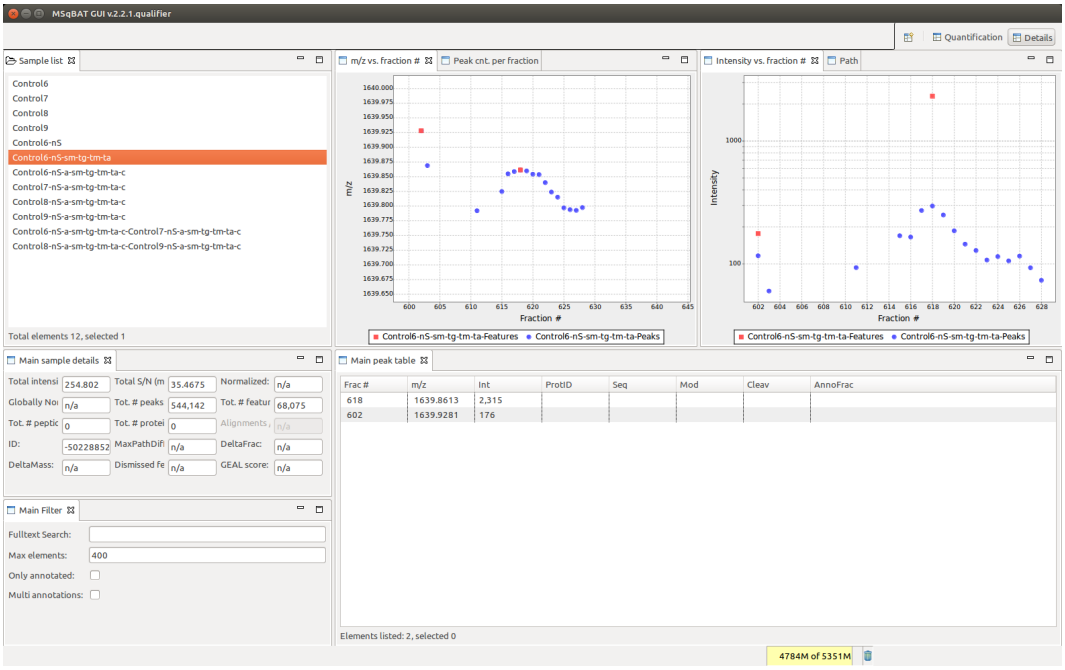
MS$_Q$BAT provides currently two different perspectives, which group the eight views according to their primary usage.

15.2.2.1 "Quantification" perspective

Per default, the "Quantification" perspective groups together the "Sample List" view, the "Sample Details" view, the "Quant Plot" view, and the Quant Table view (see Screenshot 15.7).

15.2.2.2 "Details" perspective

Per default, the "Details" perspective groups together the "Sample List" view, the "Sample Details" view, the "2D Peptide Map" view, the "2D Intensities Map" view and the "Peak Table" view (see Screenshot 15.8).



Screenshot 15.8: The "Details" perspective. The "Details" perspective arranges different views together. Per default, the "Sample List" view, the "Sample Details" view, the "2D Peptide Map" view, the "2D Intensities Map" view, the "Peak Table" view and a data-filtering view are shown in this perspective.

15.3 Details of Implementation

Code was submitted during the development of MS_QBAT to the following independent but topic-related libraries:

- *Kerner Utilities*
- *Kerner Utilities - I/O*
- *Kerner Utilities - Collections*
- *Kerner Utilities - RCP*
- *JRanges*
- *JTables*
- *JFASTA*
- *JMGF*
- *BioUtils - Proteomics*
- GeneticAlgorithms

Direct dependencies, to which no code was contributed during the development of MS_QBAT, are the following:

- *Apache Commons Math™*
- *Apache Commons Lang™*
- *XStream*
- *jmzML* [268]
- JFreeChart

16

Software Validation by Spike-in Experiments

Quantification capabilities of MS_QBAT have been validated for different instrumental setups using different spike-in experiments as well as one publicly available LC-ESI-MS data set [1, 269]. The specific spike-in set up is common to all validation experiments and consists of an *E.coli* background proteome in which human proteins have been spiked at different concentrations.

Four scenarios have been tested:

1. Quantification of LC-MALDI-MS data,
2. Quantification of GeLC-MALDI-MS data,
3. Quantification of GeLC-ESI-MS data,
4. Quantification of LC-ESI-MS data.

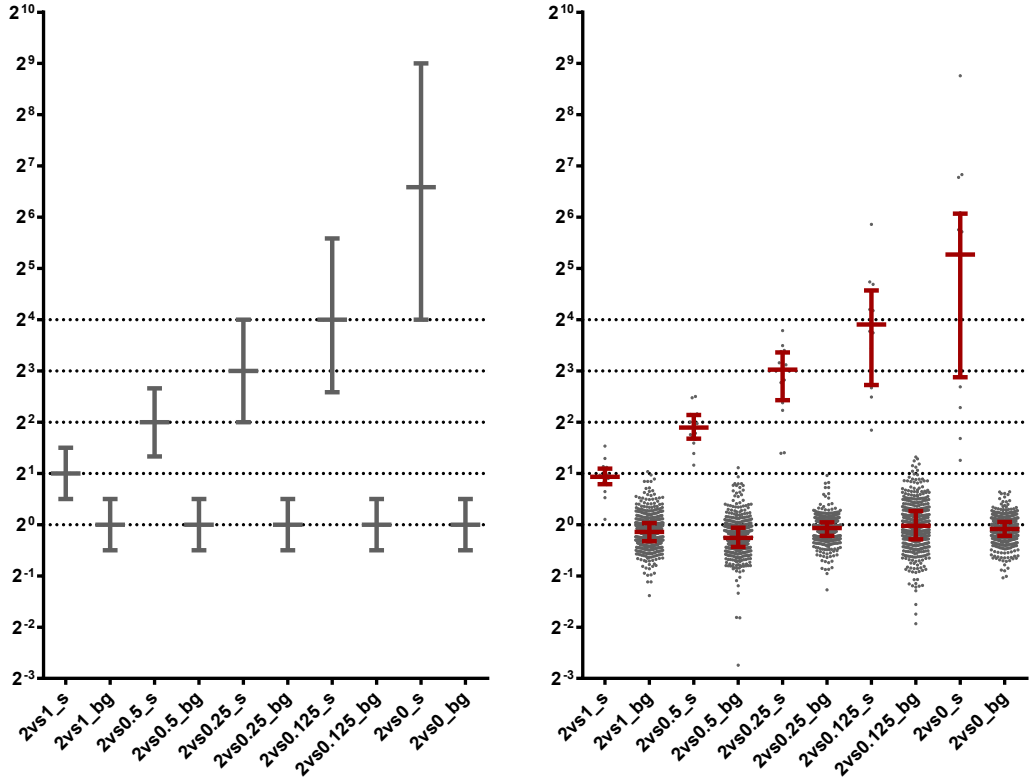
The results of each of the four test scenarios will be described in detail in this chapter.

16.1 Label-free Quantification of LC-MALDI-MS Data

To validate quantification capabilities for data resulting from a LC-MALDI-MS workflow, UPS2 has been spiked into an *E.coli* background proteome at different concentrations, resulting in six different samples (see 7).

At least four technical replicates of each sample where analyzed.

1. 2% UPS
2. 1% UPS
3. 0.5% UPS
4. 0.25% UPS
5. 0.125% UPS
6. 0% UPS.



(a) Expected regulations between UPS2% and all other samples. Illustrated are the expected regulations resulting from a quantification of UPS2% against all other samples. Human proteins (sample proteins) should show regulations of 2, 4, 8, and 16 fold (left to right). The quantification against the negative control (2vs0_s) is expected to show some regulation that is clearly distinct from the "real" comparisons. A certain degree of variance of the measured regulations with respect to the expected regulation can be expected and is illustrated by the ranges. The variance should increase as protein concentrations decrease due to interpolation of missing values ("background" intensities; see 12.3.3.3 and 17.4).

(b) Experimentally determined regulations between UPS2% and all other samples. Every dot represents one quantified protein. The different comparisons are shown on the x-axis. Spike-in proteins (*H.sapiens*/sample) and background proteins (*E.coli*) are displayed separately. Indicated in red are median regulations and interquartile ranges. As expected, the *E.coli* protein regulations cluster around $2^0(1)$ for every comparison representing no regulation. Human spike-in proteins show median regulations closely matching the expected ratios (see (a)).

Figure 16.1: Quantification results of LC-MALDI-MS data. Human proteins (UPS2) have been spiked into an *E.coli* background proteome at six different concentrations (2%, 1%, 0.5%, 0.25%, 0.125%, and 0%). The sample with the highest concentration of human proteins (named UPS2%) has been compared against all other samples. Illustrated are the expected regulations (a) and the experimentally determined regulations (b).

To cover different regulations and dynamic ranges (i.e., different ratios between the spiked proteins), sample UPS2% has been compared against all other samples, resulting in five different comparisons:

1. UPS2% vs. UPS1%
2. UPS2% vs. UPS0.5%
3. UPS2% vs. UPS0.25%
4. UPS2% vs. UPS0.125%
5. UPS2% vs. UPS0%.

The following steps have been performed for all comparisons:

1. Loading of MS^1 data from peak files by dragging the experiment folder into MS_QBAT (see 15.1).
2. Normalization to internal standards (see 13.1).
3. Annotation of MS^2 data (i.e. peptide and protein identifications; see 12.1).
4. Extraction of features (see 12.2).
5. Complexity reduction (see 13.3).
6. Global intensity normalization (see 13.2).
7. Cross alignment of technical replicates and subsequent annotation propagation between replicates (see 12.5).
8. GEAL alignment (see 13.4) of at least four technical replicates into one supersample representing the specific spike-in experiment (see above).
9. GEAL alignment of supersamples for subsequent quantification, resulting in five alignments of different spike-in samples (see above).
10. Quantification (see 12.4).

All these steps are available in MS_QBAT as a one-click command from a sample's context menu. For the configuration of the different algorithms, parameters listed in Table 16.5 have been used. A comparison between the expected results and experimentally determined results are displayed in Figure 16.1a and Figure 16.1b, respectively. Expected ratios as well as measured ratios are furthermore displayed in Table 16.1.

16.2 Label-free Quantification of GeLC-MS Data

The quantification of GeLC-MS data is a non-trivial extension to the processing of LC-MS data described in 16.1.

In order to validate quantification capabilities for GeLC-MS, UPS2 has been spiked into an *E.coli* background proteome at different concentrations, resulting in four different samples (see 7).

1. Sample D (16 μg UPS2)
2. Sample C (4 μg UPS2)
3. Sample B (1 μg UPS2)
4. Sample A (0 μg UPS2)

Samples were separated on a SDS-PAGE gel, and in total eight separate bands were cut out and tryptically digested in gel. The results were 4×8 distinct samples, which were subsequently analyzed using an GeLC-MALDI-MS and GeLC-ESI-MS workflow.

To cover different regulations and dynamic ranges (i.e., different ratios between the spiked proteins), sample D has been compared against all other samples, resulting in three different comparisons:

1. D vs. C
2. D vs. B
3. D vs. A

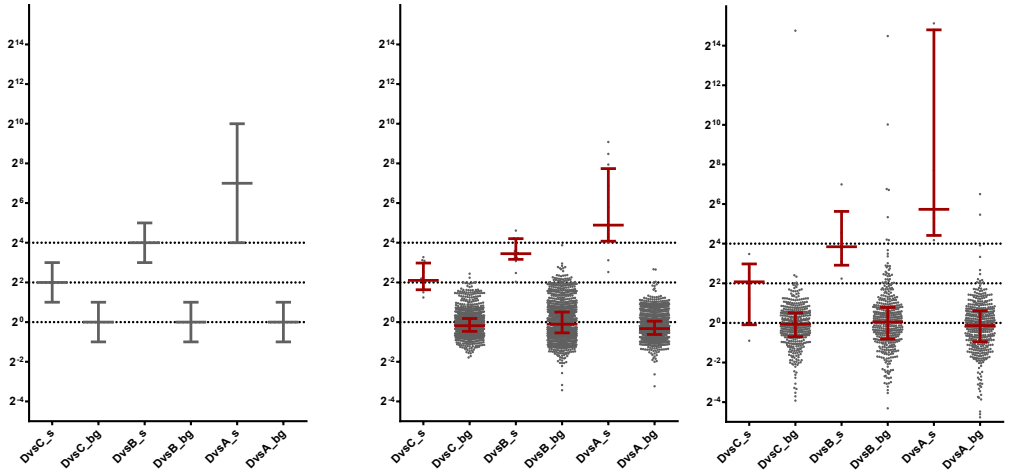
The expected regulations for these quantifications are illustrated in Figure 16.2a.

16.2.1 Label-free quantification of GeLC-MALDI-MS data

To test MS_QBAT's quantification capacities for data deriving from a GeLC-MALDI-MS workflow, the following steps have been performed for all comparisons:

1. Loading of MS^1 data from peak files by dragging the experiment folder into MS_QBAT (see 15.1).

2. Normalization to internal standards (see 13.1).



(a) Expected regulations between sample D versus all other samples. Illustrated are the expected regulations resulting from quantifying sample D against all other samples. Human proteins (sample proteins) should show regulations of 2 and 8 (left to right); the quantification against the negative control (sample A) is expected to show some regulation that is clearly distinct from the "real" comparisons. A certain degree of variance of the measured regulations with respect to the expected regulation can be expected and is illustrated by the ranges. The variance should increase as protein concentrations decrease due to interpolation of missing values ("background" intensities; see 12.3.3.3 and 17.4).

(i) GeLC-MALDI-MS. **(ii) GeLC-ESI-MS.**

(b) Comparison of protein abundances between sample D versus all other samples. Every dot represents one quantified protein. The x-axis displays the different comparisons. Spike-in proteins (human/sample) and background proteins (*E.coli*) are displayed separately. Indicated in red are median regulation and interquartile range. *E.coli* regulations cluster around $2^0(1)$ for every comparison, which represents no regulation. Human spike-in proteins show regulations depending on the experiment of 4 and 8 (from left to right), which is in agreement with the expected ratios (see left).

Figure 16.2: Quantification results of GeLC-MS data. Human proteins (UPS2) have been spiked into an *E.coli* background proteome at four different concentrations (16 μg, 4 μg, 1 μg, 0 μg). The sample with the highest concentration of human proteins (named sample D) has been compared against all other samples. Illustrated are the expected regulations (a) and the experimentally determined regulations (b).

3. Annotation from MS^2 data (i.e., peptide and protein identifications; see 12.1).
4. Extraction of features (see 12.2).
5. Complexity reduction (see 13.3).
6. Grouping of (sub-)samples (see 12.6).
7. Global intensity normalization of sample groups (see 13.2).
8. Ungroup samples.
9. GEAL alignment (see 13.4) of samples for subsequent quantification, resulting in three alignments of different spike-in samples (see above).
10. Grouping of samples.
11. Quantification (see 12.4).

All these steps are available in MS$_Q$BAT as a one-click command from a sample's context menu. For the configuration of the different algorithms, parameters listed in Table 16.6 have been used. A comparison between the expected results and experimentally determined results are displayed in Figure 16.2a and Figure 16.2i, respectively. Expected ratios as well as measured ratios are furthermore displayed in Table 16.2.

16.2.2 Label-free quantification of GeLC-ESI-MS data

To test MS$_Q$BAT's quantification capacities for data deriving from a GeLC-ESI-MS workflow, the following steps have been performed for all comparisons:

1. Loading of MS^1 data from `isos.csv` files by dragging them into MS$_Q$BAT (see 15.1).
2. Normalization of fraction indices (see 14).
3. Annotation from MS^2 data (i.e., peptide and protein identifications; see 12.1).
4. Extraction of features (see 12.2).
5. Grouping samples (12.6).
6. Global intensity normalization of sample groups (13.2).

7. Ungrouping of samples.
8. GEAL alignment (13.4) of samples for subsequent quantification, resulting in three alignments of different spike-in samples (see above).
9. Grouping of samples.
10. Quantification (12.4).

All these steps are available in MS_QBAT as a one-click command from a sample's context menu and were configured using the parameters listed in Table 16.7. Results of these quantifications are illustrated in Figure 16.2ii.

16.3 Label-free Quantification of LC-ESI-MS Data

To validate quantification capabilities for data resulting from a state-of-the-art, high resolution LC-ESI-MS workflow, a publicly available quantification benchmark dataset was downloaded and processed [1, 269]. Human proteins (universal proteomics standard (UPS1) and UPS2) have been spiked into an *E.coli* background proteome at fixed (UPS1) and dynamic (UPS2) concentrations. Comparing protein abundances between these two samples results in a total of six differently regulated groups of human proteins.

In total, eight `raw`-files have been downloaded [269]: two different samples (UPS1 and UPS2) with four technical replicates each.

Monoisotopic peak lists have been extracted using DeconTools (see 9.2.1). Peptide identification information was extracted from the `modificationSpecificPeptides.txt` file (available from the ProteomeXchange website (see 9.2.2)). Importantly, this file does not contain the full set of peptide identifications but only a non-redundant subset. Parameter optimization using GEAL was therefore not available for this dataset. The original quantification results obtained by MaxQuant were taken from the `proteinGroups.txt` file (also available from the ProteomeXchange website) and are compared to the quantification results obtained by MS_QBAT (see Figure 16.3).

The following steps have been performed for all comparisons:

1. Loading of MS^1 data from peak files by dragging the `raw` files into MS_QBAT (see 15.1).

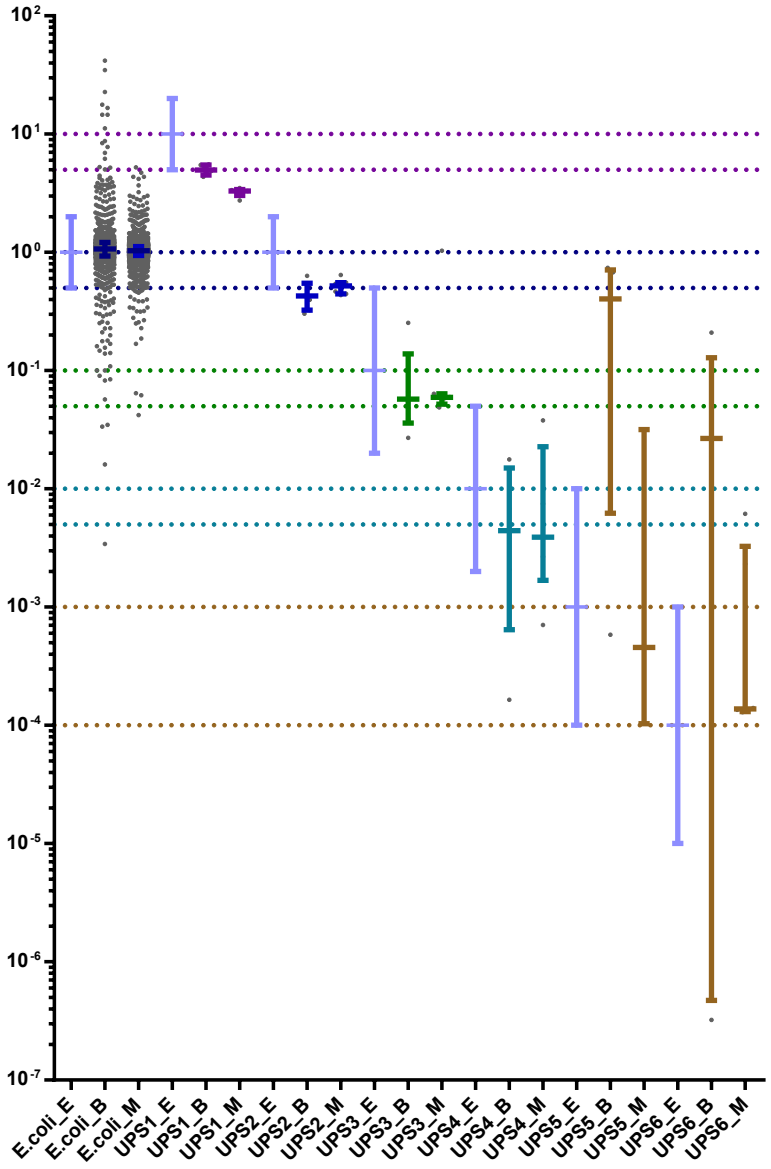


Figure 16.3: Measured regulations between UPS2 and UPS1. Every dot represents one quantified protein. In total, three different data sets are shown: expected quantification results (E.coli_E and UPS1_E – UPS6_E), experimentally determined ratios by MS_QBAT (E.coli_B and UPS1_B – UPS6_B), and experimentally determined ratios by MaxQuant (E.coli_M and UPS1_M – UPS6_M). The x-axis displays the different comparisons, the y-axis the calculated and expected ratios (log_{10}). Background- (*E.coli*) and sample proteins (*H.sapiens*) are displayed separately (E.coli and UPS1 – UPS6). Expected regulations are indicated (E.coli_E: 1, UPS1_E: 10, UPS2_E: 1, UPS3_E: 0.1, UPS4_E: 0.01, UPS5_E: 0.001 and UPS6_E: 0.0001). Since regulations seem to be systematically underestimated, an additional line is displayed on the y-axis, representing the expected regulation shifted by −50 %. Regulations are expected to show some variance that increases with decreasing protein abundances (indicated by increasing ranges). Determined ratios by MS_QBAT and MaxQuant for the different sample protein groups are color-coded, using the same color respectively. For the last two sample protein groups, the same color-code is used, since they do not separate clearly from each other and do not match the expected ratios.

2. Annotation from MS^2 data (see 12.1).
3. Normalization of retention times/fraction numbers (see 14).
4. Filtering by m/z (600 – 4,500) and fraction number (2,000 – 11,000).
5. Extraction of features (see 12.2).
6. Global normalization (see 13.2).
7. Cross alignment of the respective sample replicates (see 12.5). Sample pairs that showed the best alignment (evaluated by r^2 and visually by inspecting the resulting quant-plot (see 15.2.1.4)) were selected to build a representative superalignment for the biological sample (UPS1 and UPS2). For both samples, superalignments were built using fixed parameters from two replicates each.
8. GEAL alignment (see 13.4) of supersamples representing the two biological samples (UPS1 and UPS2).
9. Quantification (see 12.4) of the final alignment.

All these steps are available in MS_QBAT as a one-click command from a sample's context menu and were configured using the parameters listed in Table 16.8. Expected results and experimentally determined results are displayed in Figure 16.3. Table 16.4 shows some numerical metrics for the experimentally determined results by MS_QBAT. Since experimentally obtained regulations seem to be systematically sifted, two lines on the y-axis indicate the expected ratio as well as the expected ratio considering a shift of $-50\,\%$.

Background/ Sample	Quantification	Expected	Actual	Std.dev.	25^{th} percentile	75^{th} percentile	Number of proteins
E.coli	2% vs. 1%	0.00	−0.14	0.32	−0.32	0.03	561
E.coli	2% vs. 0.5%	0.00	−0.26	0.36	−0.43	−0.06	528
E.coli	2% vs. 0.25%	0.00	−0.07	0.25	−0.22	0.05	511
E.coli	2% vs. 0.125%	0.00	−0.02	0.49	−0.29	0.27	504
E.coli	2% vs. 0%	0.00	−0.08	0.24	−0.22	0.05	546
H.sapiens	2% vs. 1%	1.00	0.94	0.32	0.79	1.09	16
H.sapiens	2% vs. 0.5%	2.00	1.89	0.36	1.68	2.14	16
H.sapiens	2% vs. 0.25%	3.00	3.02	0.69	2.43	3.36	16
H.sapiens	2% vs. 0.125%	4.00	3.90	1.07	2.73	4.57	14
H.sapiens	2% vs. 0%	$\gg$4.00	5.28	2.08	2.85	6.07	16

Table 16.1: Expected and experimentally determined median regulations of background- and sample proteins for LC-MALDI-MS. Five different comparisons are shown, always including UPS 2% (comparisons to 1%, 0.5%, 0.25%, 0.125%, and 0%). Depicted are expected regulation (log_2), actual regulation (log_2), standard deviation, and interquartile range.

Background/ Sample	Quantification	Expected	Actual	Std.dev.	25^{th} percentile	75^{th} percentile	Number of proteins
E.coli	D vs. C	0.00	−0.19	0.59	−0.47	0.18	1316
E.coli	D vs. B	0.00	−0.11	0.91	−0.55	0.50	1342
E.coli	D vs. A	0.00	−0.33	0.64	−0.63	0.05	1422
H.sapiens	D vs. C	2.00	2.10	0.68	1.63	2.97	13
H.sapiens	D vs. B	4.00	3.45	0.73	3.16	4.20	13
H.sapiens	D vs. A	$\gg$4.00	4.87	2.13	4.07	7.68	12

Table 16.2: Expected and experimentally determined median regulations of background- and sample proteins for GeLC-MALDI-MS. A total of three different comparisons are shown, always including sample D (comparisons to sample C, B and A). Depicted are expected regulation (log_2), actual regulation (log_2), standard deviation and interquartile range.

Background/ Sample	Quantification	Expected	Actual	Std.dev.	25^{th} percentile	75^{th} percentile	Number of proteins
E.coli	D vs. C	0.00	0.17	1.29	−0.49	0.72	429
E.coli	D vs. B	0.00	0.04	1.85	−0.81	0.78	443
E.coli	D vs. A	0.00	−0.14	1.70	−0.95	0.61	442
H.sapiens	D vs. C	2.00	1.89	1.48	0.09	2.94	6
H.sapiens	D vs. B	4.00	3.84	1.63	2.87	5.16	6
H.sapiens	D vs. A	$\gg$4.00	5.73	4.77	4.42	14.80	7

Table 16.3: Expected and actual median regulation of background- and sample proteins for GeLC-ESI-MS. Shown are three comparisons, sample D against all other concentrations (sample C, sample B, and sample A). Depicted are expected regulation (log_2), actual median regulation (log_2), standard deviation and interquartile range.

Background/ Sample	Quantification	Expected	Actual	Std.dev.	25^{th} percentile	75^{th} percentile	Number of proteins
E.coli	UPS2 vs. UPS1	10^0	1.065	1,387	0.93	1.21	1,250
UPS1	UPS2 vs. UPS1	10	$5 \cdot 10^1$	$48 \cdot 10^{-2}$	$45 \cdot 10^{-1}$	$55 \cdot 10^1$	6
UPS2	UPS2 vs. UPS1	1	$43 \cdot 10^{-2}$	$12 \cdot 10^{-2}$	$32 \cdot 10^{-2}$	$55 \cdot 10^{-2}$	7
UPS3	UPS2 vs. UPS1	10^{-1}	$57 \cdot 10^{-3}$	$8 \cdot 10^{-2}$	$4 \cdot 10^{-2}$	$14 \cdot 10^{-2}$	6
UPS4	UPS2 vs. UPS1	10^{-2}	$44 \cdot 10^{-4}$	$1 \cdot 10^{-2}$	$2 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	4
UPS5	UPS2 vs. UPS1	10^{-3}	$4 \cdot 10^{-2}$	$35 \cdot 10^{-2}$	$1 \cdot 10^{-2}$	$71 \cdot 10^{-2}$	5
UPS6	UPS2 vs. UPS1	10^{-4}	$27 \cdot 10^{-3}$	$8 \cdot 10^{-2}$	$5 \cdot 10^{-7}$	$13 \cdot 10^{-2}$	6

Table 16.4: Expected and experimentally determined median regulation of background and sample proteins for LC-ESI-MS. Seven comparisons between UPS2 and UPS1 are shown (*E.coli* proteins and *H.sapiens* proteins at six different concentrations). Depicted are expected regulation, experimentally determined regulation, standard deviation, and interquartile range.

Processing Step	Parameter	Value
Normalization on internal standards	Mass delta (+/-)	25
Normalization on internal standards	Mass delta is ppm	`true`
Annotation	Fraction delta (-)	1
Annotation	Fraction delta (+)	1
Annotation	Mass delta (+/-)	25
Annotation	Mass delta is ppm	`true`
Annotation	Minimum sequence confidence	95 %
Annotation	Maximum mass shift	0.3 Da
Feature extraction	Fraction gaps tightener (+)	1
Feature extraction	Mass delta tightener	50
Feature extraction	Mass delta tightener is ppm	`true`
Feature extraction	Use annotations tightener	`true`
Complexity reduction	Minimum percentage peptide identifications to keep	95 %
Complexity reduction	S/N low	0
Complexity reduction	S/N high	200
Complexity reduction	S/N interval	10
Complexity reduction	Feature length low	0
Complexity reduction	Feature length high	4
Complexity reduction	Feature length interval	1
Global intensity normalization	Reference intensity	250 mio
Alignment for annotation propagation	Fraction shift	10
Alignment for annotation propagation	Mass shift	5
Alignment for annotation propagation	Mass shift is ppm	`true`
Alignment for annotation propagation	Maximum path diff	2
Alignment for annotation propagation	Use annotations to find best alignment	`true`
GEAL alignment	Number of iterations	400
GEAL alignment	Number of peaks to use	10
GEAL alignment	Fraction delta low	5
GEAL alignment	Fraction delta high	100
GEAL alignment	Fraction delta interval	5
GEAL alignment	Mass delta low	5
GEAL alignment	Mass delta high	50
GEAL alignment	Mass delta interval	5
GEAL alignment	Path low	2
GEAL alignment	Path low	50
GEAL alignment	Path low	1
Quantification	Minimum peptide number	3

Table 16.5: LC-MALDI-MS: Parameters used for quantification.

Processing Step	Parameter	Value
Normalization on internal standards	Mass delta (+/-)	25
Normalization on internal standards	Mass delta is ppm	`true`
Annotation	Fraction delta (-)	1
Annotation	Fraction delta (+)	1
Annotation	Mass delta (+/-)	25
Annotation	Mass delta is ppm	`true`
Annotation	Minimum sequence confidence	95 %
Annotation	Maximum mass shift	0.3 Da
Feature extraction	Fraction gaps tightener (+)	1
Feature extraction	Mass delta tightener	50
Feature extraction	Mass delta tightener is ppm	`true`
Feature extraction	Use annotations tightener	`true`
Complexity reduction	Minimum percentage peptide identifications to keep	95 %
Complexity reduction	S/N low	0
Complexity reduction	S/N high	200
Complexity reduction	S/N interval	10
Complexity reduction	Feature length low	0
Complexity reduction	Feature length high	4
Complexity reduction	Feature length interval	1
Global intensity normalization	Reference intensity	3,000 mio
GEAL alignment	Number of iterations	400
GEAL alignment	Number of peaks to use	10
GEAL alignment	Fraction delta low	5
GEAL alignment	Fraction delta high	100
GEAL alignment	Fraction delta interval	5
GEAL alignment	Mass delta low	5
GEAL alignment	Mass delta high	50
GEAL alignment	Mass delta interval	5
GEAL alignment	Path low	2
GEAL alignment	Path low	50
GEAL alignment	Path low	1
Quantification	Minimum peptide number	3

Table 16.6: GeLC-MALDI-MS: Parameters used for quantification.

Processing Step	Parameter	Value
I/O generic	Column identifier m/z	monoisotopic_mw
I/O generic	Column identifier intensity	abundance
I/O generic	Column identifier S/N	signal_noise
I/O generic	Column delimiter	“Comma”
Annotation	Fraction delta (-)	100
Annotation	Fraction delta (+)	100
Annotation	m/z delta (+/-)	800
Annotation	m/z delta is ppm	`true`
Annotation	Scale time values to fit fractions	`true`
Annotation	Reader to use	“ProteinPilot”
Annotation filter	Max mass delta [Da]	1,000
Annotation filter	Min sequence confidence	95
Annotation ProteinPilot	Column identifier m/z	“Theor MW”
Annotation ProteinPilot	Column identifier sequence	“Sequence”
Annotation ProteinPilot	Column identifier protein ID	“Accessions”
Annotation ProteinPilot	Fraction index	“Use time directly”
Feature extraction	Use fraction gaps tightener	`true`
Feature extraction	Gap size [fractions] (+)	7
Feature extraction	Use m/z tightener	`true`
Feature extraction	Mass shift for m/z tightening (abs)	800
Feature extraction	Use ppm mass shift for m/z tightening	`true`
Feature extraction	Use annotations tightener	`true`
Global intensity normalization	Reference intensity	160,000
Alignment	Max fraction delta during mapping (+/-)	40
Alignment	Max m/z delta during mapping (+/-)	800
Alignment	Max m/z delta during mapping is ppm	`true`
Alignment	Orphanize alignments with path diff greater than (+/-)	40
Alignment	Also use annotations to find best alignment	`true`
GEAL alignment	Number of iterations	400
GEAL alignment	Number of peaks to use	10
GEAL alignment	Fraction delta low	10
GEAL alignment	Fraction delta high	100
GEAL alignment	Fraction delta interval	10
GEAL alignment	Mass delta low	400
GEAL alignment	Mass delta high	1,200
GEAL alignment	Mass delta interval	50
GEAL alignment	Path low	10
GEAL alignment	Path high	80
GEAL alignment	Path step	2
Quantification	Min number peptides	3

Table 16.7: GeLC-ESI-MS: Parameters used for quantification.

Processing Step	Parameter	Value
I/O generic	Column identifier m/z	monoisotopic_mw
I/O generic	Column identifier intensity	abundance
I/O generic	Column identifier S/N	signal_noise
I/O generic	Column delimiter	“Comma”
Annotation	Fraction delta (-)	100
Annotation	Fraction delta (+)	1
Annotation	Mass delta (+/-)	25
Annotation	Mass delta is ppm	`true`
Annotation	Reader to use	“Generic”
Annotation	Minimum sequence confidence	95 %
Annotation	Maximum mass shift	1,000 Da
Annotation Generic	Column identifier m/z	Mass
Annotation Generic	Column identifier fraction	MS/MS Scan Number
Annotation Generic	Column identifier sequence	Sequence
Annotation Generic	Column identifier protein ID	Proteins
Feature extraction	Fraction gaps tightener (+)	50
Feature extraction	Mass delta tightener	50
Feature extraction	Mass delta tightener is ppm	`true`
Feature extraction	Use annotations tightener	`true`
Global intensity normalization	Reference intensity	16,000,000 mio
Alignment	Fraction shift	1,000
Alignment	Mass shift	30
Alignment	Mass shift is ppm	`true`
Alignment	Maximum path diff	30
Alignment	Use annotations to find best alignment	`true`
GEAL alignment	Number of iterations	200
GEAL alignment	Number of peaks to use	10
GEAL alignment	Fraction delta low	200
GEAL alignment	Fraction delta high	4,000
GEAL alignment	Fraction delta interval	100
GEAL alignment	Mass delta low	10
GEAL alignment	Mass delta high	100
GEAL alignment	Mass delta interval	10
GEAL alignment	Path low	10
GEAL alignment	Path high	100
GEAL alignment	Path interval	5
Quantification	Minimum peptide number	3

Table 16.8: LC-ESI-MS: Parameters used for quantification.

Part IV

Discussion

17

Conclusions

The aim of this work was the development and application of a software tool for label-free quantification of LC-MALDI-MS data. Currently, no software solution exists that is able to perform a quantitative analysis of LC-MALDI-MS data acquired by state-of-the-art LC-MALDI-MS systems. For this purpose, a software suite should be built that provides this functionality and presents it to non-expert users in an intuitive and user-friendly fashion. Ideally, the software should be flexible enough to be extendable for the processing of GeLC-MALDI-MS- and LC-/GeLC-ESI-MS data.

MS_QBAT was built with the intention to satisfy all these requirements. MS_QBAT is a platform-independent software package enabling the user to perform large-scale label-free protein quantification using both LC-/GeLC-MALDI-MS and LC-/GeLC-ESI-MS data. The software was written from scratch to guarantee the desired flexibility and user-friendliness. It implements all algorithms required for a relative, label-free peptide and protein quantification based on ion intensities, namely (i) feature extraction, (ii) feature annotation, (iii) feature alignment, and (iv) peptide/protein quantification. Furthermore, MS_QBAT contains additional libraries for (i) local and (ii) global intensity normalization, (iii) dynamic and automatic sample complexity reduction, (iv) automatic parameter optimization by genetic algorithms, and (v) a tool set for advanced data acquisition.

To validate MS_QBAT's quantification capabilities, several spike-in experiments have been performed. Furthermore, publicly available benchmark data was used for the processing and quantification of LC-ESI-MS data.

17.1 MS_QBAT Allows for the Accurate, Label-free Quantification of LC-MALDI-MS Data

To validate MS_QBAT's quantification capacities of LC-MALDI-MS data, a spike-in test was set up. Human proteins (UPS2) were spiked into an *E.coli* background proteome at six different concentrations, namely 2%, 1%, 0.5%, 0.25%, 0.125%, and 0%. The resulting samples were subsequently analyzed using a nanoACQUITY UPLC ® LC system and an AB SCIEX TOF/TOF™ 5800 MS system.

At least four technical replicates have been aligned into one supersample, representing the corresponding UPS concentration:

1. 2% UPS
2. 1% UPS
3. 0.5% UPS
4. 0.25% UPS
5. 0.125% UPS
6. 0% UPS

To cover different regulations and dynamic ranges (i.e., different ratios between the spiked proteins), sample UPS2% was compared against all other samples, resulting in a total of five comparisons:

1. UPS 2% vs. UPS 1%
2. UPS 2% vs. UPS 0.5%
3. UPS 2% vs. UPS 0.25%
4. UPS 2% vs. UPS 0.125%
5. UPS 2% vs. UPS 0%

The experimentally determined protein quantifications closely matched the expected regulations. Background proteins (*E.coli*) did not show any regulation and therefore clustered around a regulation of 0 (log2), while the spiked human proteins showed the expected regulations.

The variance of the measured regulations is very small. Quantified background proteins constantly show very small variance (standard deviation between 0.24 and 0.49), which enables the clear separation from background protein regulations down to ratios as low as four fold. A separation of regulated human proteins from background *E.coli* proteins is also observable for two fold regulations; however, this separation is not as clear as the one for higher ratios.

Importantly, in terms of biomarker discovery, the identification of lower fold changes (i.e., <10-fold) is only of limited benefit, since therapeutically usable biomarkers usually express strong regulations of tenfold and more (see 2). Very small variances, which allow for a clear separation of regulated proteins in the range of two- to tenfold, are therefore not necessarily needed.

Nevertheless, the identification of lower fold changes with high accuracy can be achieved even with higher variances, if statistical significance is taken into account. MS_QBAT calculates for each regulation simple statistics to be able to estimate a regulation's significance. Calculated p-values are not shown in Figure 16.1b, but help to distinguish between significant regulations and noise even for very low regulations.

In summary, it can be concluded that the implemented algorithms allow for a highly accurate LC-MALDI-MS protein quantification with low variances of the calculated ratios. Increased variances of calculated protein ratios can only be found for regulations >16-fold for low abundant proteins (see also 17.4). However, in contrast to the variance, the median protein regulation remains highly accurate.

17.2 MS_QBAT Allows for the Accurate, Label-free Quantification of GeLC-MALDI-MS and GeLC-ESI-MS Data

The ability to quantify GeLC-MS-data was an important requirement and extension to MS_QBAT's quantification capabilities. To validate GeLC-MS quantification capabilities, a spike-in test very similar to the one discussed in 17.1 was set up. Human proteins (UPS2) were spiked into an *E.coli* background proteome at four different concentrations, namely 16 μg, 4 μg, 1 μg, and 0 μg. Samples were separated on a SDS-PAGE gel and eight separate bands were cut out and tryptically digested in-gel. The result were 4×8 distinct samples, which were subsequently analyzed using a nanoACQUITY UPLC ® LC system and both an AB SCIEX TOF/TOF™ 5800 MS system and an AB SCIEX QTRAP® 6500 MS system.

1. D (16 μg UPS2)
2. C (4 μg UPS2)
3. B (1 μg UPS2)
4. A (0 μg UPS2)

To cover different regulations and dynamic ranges (i.e., different ratios between the spiked proteins), sample D was compared against all other samples, resulting in three comparisons:

1. D vs. C
2. D vs. B
3. D vs. A

The experimentally determined protein quantification closely matched the expected regulations for both the MALDI-based approach and the ESI-based workflow. Background proteins (*E.coli*) did not show any regulation and therefore clustered around a regulation of 0 (log2), while the spiked human proteins showed the expected regulations.

The variances of the quantification results were higher compared to data obtained in the LC-MALDI-MS workflow discussed above. The increased variance of experimentally determined regulations has mainly two reasons: First, in the GeLC-MS-based approach, sample complexity is significantly increased due to the additional separation step. In this case, each sample (A, B, C, D) consists of eight separate subsamples, which had to be merged computationally. More importantly, samples are not covered by technical replicates as it is the case for the LC-MALDI-MS workflow discussed in 17.1. The different comparisons have been performed with one sample per spike-in group only. Technical replicates increase quantification accuracy, since the final ratio is an average of multiple (replicate-) regulations.

Importantly, in contrast to the spike-in setup used for the evaluation of LC-MALDI-MS protein quantification performance, variance of calculated ratios does not increase when quantifying higher ratios. This is because in the GeLC-MALDI-MS setup, higher total protein amounts have been used and therefore a quantification "into background signal" is not given (see also 17.4). An increased variance of calculated ratios is also observable for the data obtained by GeLC-ESI-MS compared to GeLC-MALDI-MS. The main reason for this difference is the number of identified and quantified proteins [270]. In the GeLC-MALDI-MS workflow, approximately three times more *E.coli* proteins and more than two times more *H.sapiens* proteins could be identified (see Table 16.2 and Table 16.3).

In summary, it can be concluded that the implemented quantification algorithms allow for a highly accurate quantification of GeLC-MS data with acceptable variances both for data deriving from a MALDI-based- as well as data deriving from an ESI-based workflow. The variances of the quantification results could be improved by the application of technical replicates.

17.3 MS_QBAT Allows for the Accurate, Label-free Quantification of LC-ESI-MS Data

The quantification of LC-ESI-MS data was not initially planned when MS_QBAT was implemented. Nevertheless, a maximum of flexibility of the algorithms with regard to input data has been an important requirement; i.e., an adaptation to ESI data should be relatively easy later on. During the final stage of the software development, its flexibility has been successfully tested with LC-ESI-MS data. LC-ESI-MS is the most prominent flavor of LC-MS today and respective data is available online via the ProteomeXChange repository [243]. To evaluate MS_QBAT's capabilities for the quantification of this data type, raw data files (`.raw` files, i.e., the generic data format from Thermo Fisher) containing all MS^1 and MS^2 information were downloaded [269]. As MS_QBAT is not capable of performing a peptide identification by MS^2, the respective data generated by MaxQuant was used. The file `modificationSpecificPeptides.txt`, which is available on ProteomeXChange as well, was used to extract respective peptide identification information.

Importantly, this file does not contain redundant peptide identification information but only the "best" identifications (chosen from all runs). These non-redundant peptide identifications are sufficient for protein quantification. Nevertheless, a GA-based alignment is not feasible if samples do not share common peptide identifications. Therefore, the respective alignment parameters had to be chosen manually resulting in a non-optimal alignment and hence a non-optimal quantification result.

The experimentally determined quantification results have been compared to the expected ratios, as well as to the results obtained by Cox and colleagues in the original paper [1]. Compared to the expected ratios, experimentally determined quantification results by MS_QBAT systematically underestimate the real ratios by approximately 50 %. Interestingly, ratios calculated by MaxQuant show this shift as well. MS_QBAT and MaxQuant both show this shift for the calculated ratios of human proteins. In contrast, quantified *E.coli* proteins do not show this shift, but the calculated ratio by both MS_QBAT and MaxQuant is close to 1 (1.07 and 1.03). Since only the spike-in proteins show a systematic shift, whereas quantified *E.coli* proteins show expected ratios, it can be hypothesized that the samples reflect a real difference in theoretical ratio and actual ratio (e.g., caused by discrepancies in sample preparation).

Comparing experimentally determined protein ratios by MS_QBAT to the ratios determined by MaxQuant, quantified *E.coli* proteins as well as the first four differentially regulated human protein groups (expected regulations of 10^0 to 10^{-2}) show a high conformity with each other. Groups five and six (expected regulations of 10^{-2} and 10^{-3}) could not be accurately quantified and show a large variance for quantifications calculated by both MS_QBAT and MaxQuant. By tendency, MaxQuant is able to state protein regulations for these two groups with lower variance.

Calculated ratios by MS_QBAT and MaxQuant show a variance that is low for the first three differentially regulated protein groups and higher for protein group four. Since the variance of ratios increases when quantifying proteins in group four, and is large for groups five and six, it can be assumed that human proteins in group five and six could be identified only at the detection limit of the instrument. As discussed briefly before and in detail in 17.4, a quantification of proteins at abundances near background signal leads to decreased quantification accuracy and increased variance of calculated ratios.

In summary, it can be concluded that the implemented algorithms are not only applicable for GeLC-/LC-MS data but are suitable for protein quantification of LC-ESI-MS data as well. Experimentally determined ratios closely match those obtained by MaxQuant, a widely used software for protein quantification exclusively for LC-ESI-MS data. Importantly, alignment parameters, which are a crucial component of the quantification process, could not be optimized using the implemented GA-based optimization routines; therefore, an improved quantification can be expected if redundant peptide identification information for each run is available.

17.4 MS_QBAT Allows for the Quantification of a "Black-and-White" Situation

As already briefly mentioned in 4, 4.3, and 4.2.3, the availability of exactly two protein/peptide abundances/intensities is, in principle, a prerequisite for the calculation of protein abundances for quantification. This requirement limits protein quantification to both samples containing the protein of interest. A *"black-and-white"* situation, which is given if a protein is present in one sample only, and which represents a very strong "regulation", cannot be covered. Therefore, it has always been a challenge to cover also such protein regulations. For label-based approaches, the accounting for presence-absence situations is very difficult and, therefore, usually not performed in praxis [1]. The same holds true for MS^2-based quantifications, as spectra numbers are compared. This approach faces the same problems as a label-based quantification. "Zero spectra" and "zero label" really mean zero. By contrast, a missing value in an MS^1-based label-free approach can relatively easily be interpolated by a "zero-value", which is a value representing a background signal.

However, interpolating missing values comes at a price. While proteins present only in one out of two samples to be compared can be "quantified", this "quantification" is not fully accurate and cannot be compared to a readout of an actual ratio. Nevertheless, especially in biomarker discovery, a general ratio trend is more important than an accurately calculated ratio, since very strong regulations representing protein abundances at or close to the detection limit of an instrument occur frequently. MS_QBAT implements the approach of interpolating missing values by background intensity values.

Figure 16.1b clearly shows the effects of the introduction of artificial background values: Sample proteins (*H.sapiens*) show an increasing variance when quantifying larger dynamic ranges. The first two comparisons, namely UPS2% vs. UPS1% and UPS2% vs. UPS0.5%, show a very small variance (standard deviation between 0.32 and 0.36), highly similar to the variance of quantified background proteins. As the protein amounts in the second sample (UPS0.25% and UPS0.125%) decrease, the observed variances continuously increase. A "black-and-white" comparison (UPS2% vs. UPS0%) finally results in a very large variance compared to the other quantifications.

The calculation of background intensity presents a rough estimation of the actual protein regulation ratio, which is why interpolated values may vary significantly. The lower the number of identified peptides (i.e., the more "background peptides" have to be considered), the higher the observed variance in calculated ratios. These findings are illustrated by increasing interquartile ranges (see Figure 16.1b).

In summary, it can be concluded that MS_QBAT is capable to detect "black-and-white" situations in terms of protein abundances. Protein ratios resulting from such quantifications only indicate a strong ratio trend instead of an accurately calculated ratio. The respective algorithms have been implemented in such a way that these ratios generally underestimate the real situation. This is in contrast to existing solutions, which tend to overestimate such ratios [1].

18

Applications of MS_QBAT

In the last two years, preliminary versions of the MS_QBAT software have been successfully applied in a number of studies with very diverse biological and/or chemical issues.

18.1 Novel Biotinylation Reagents

MS_QBAT has been used to evaluate the performance both of different commercially available biotinylation reagents and of two novel in-house developed biotinylation reagents [271].

The biotinylation approach is used for the selective enrichment of vascular accessible proteins such as cell surface proteins and extracellular matrix proteins [272]. The general mode of action of these reagents is the modification of primary amino groups of proteins with membrane-impermeable, water-soluble chemical derivatives of biotin [273–275]. After cell lysis, the sample is enriched by purifying biotinylated proteins on streptavidin-coated resin [276, 277]. Several biotinylation reagents are commercially available, such as Sulfo-NHS-LC-Biotin and NHS-PEG_{12}-Biotin. These reagents show different characteristics in terms of membrane-impermeability and enrichment of cell surface- and extracellular matrix proteins.

The biotinylation performance was evaluated by *in vivo* biotinylation of mice using the two commercially available reagents Sulfo-NHS-LC-Biotin and NHS-PEG_{12}-Biotin, and two novel in-house synthesized biotinylation reagents. The enriched proteome fraction from kidney tissue was analyzed by LC-MALDI-MS and quantified using MS_QBAT. The resulting protein quantification data confirm the increased selectivity of the novel biotinylation reagents for cell-membrane proteins compared to the two commercially available reagents Sulfo-NHS-LC-Biotin and NHS-PEG_{12}-Biotin.

18.2 Vascular Accessible Markers in Kidney Cancer

MS_QBAT has been used to identify vascular accessible biomarkers for clear cell (cc) renal cell carcinoma (RCC), and resulting lung metastases [278].

Seven different patient-derived xenograft mouse models have been developed and used for the generation of primary tumors and lung metastases. Commercially available primary renal cell lines have been used as healthy controls. To enrich for cell surface proteins, the aforementioned *in vivo* biotinylation approach (see 18.1) was applied. Proteomic analysis was performed by LC-MALDI-MS. The chromatographic separation of tryptic peptides was performed using a nanoACQUITY UPLC ® system. Subsequent mass spectrometric analysis was performed using an AB SCIEX TOF/TOF™ 5800 system and quantification of the acquired LC-MS data was achieved by MS_QBAT.

As xenograft mouse models have been used, human as well as murine proteins have been identified in the LC-MS analysis. An experimental setup with xenograft models additionally complicates the protein inference problem described in 11. To allow for the distinction between human and murine proteins, identified peptides have been processed by PepSir (see 11) prior to quantification by MS_QBAT. PepSir was configured to check proteotypicity in a two-species (*H.sapiens* & *M.musculus*) context. Only peptides proteotypic in this context have been used for the subsequent quantification by MS_QBAT. Hierarchical cluster analysis was performed based on the determined intensity values of identified and quantified proteins. The data clustered in three main groups, clearly separating proteins regulated in healthy controls, metastases, and primary tumor. Furthermore, MS_QBAT could be used to identify several putative markers, which subsequently have been validated by immunofluorescence and real-time (RT)-quantitative polymerase chain reaction (qPCR).

18.3 PDAC Subtypes

MS$_Q$BAT has been used to characterize pancreatic ductal adenocarcinoma (PDAC) subtypes on the proteomic level and to identify subtype specific markers as well as novel pan PDAC markers [279]. In 2011, Collisson *et al.* identified three different PDAC subtypes on the genomic level, which show different responses to therapy [280]. Twelve PDAC-derived primary patient-matched cell lines grown under serum-free conditions as well as two control cell lines have been analyzed using a GeLC-MALDI-MS approach. Proteins separated by SDS-PAGE have been divided into twelve subsamples (i.e., gel slices). The separation of resulting tryptic peptides was performed using a nanoACQUITY UPLC ® system; the mass spectrometric analysis of each LC-fraction was conducted using an AB SCIEX TOF/TOF™ 5800 system. This setup led to a total of 168 samples to be processed. Due to the extensive volume of this study, MS data have been acquired over more than 18 months. Principally, such an extended time period complicates label-free protein quantification and puts any quantification software on the test. The resulting data confirmed the three subtypes characterized by Collisson *et al.* on the proteomic level. In addition, subtype-specific biomarker candidates could be identified. MS$_Q$BAT identified several regulated and highly specific biomarker candidates, whereof the most promising ones have been selected for validation by multiple reaction monitoring (MRM) (see Figure 18.1). The MRM measurements indicate a strong correspondence to ratios calculated by MS$_Q$BAT and confirm MS$_Q$BAT's protein quantification accuracy even for highly complex proteomes, which have been processed and analyzed over several months and in subsections of more than 160 samples.

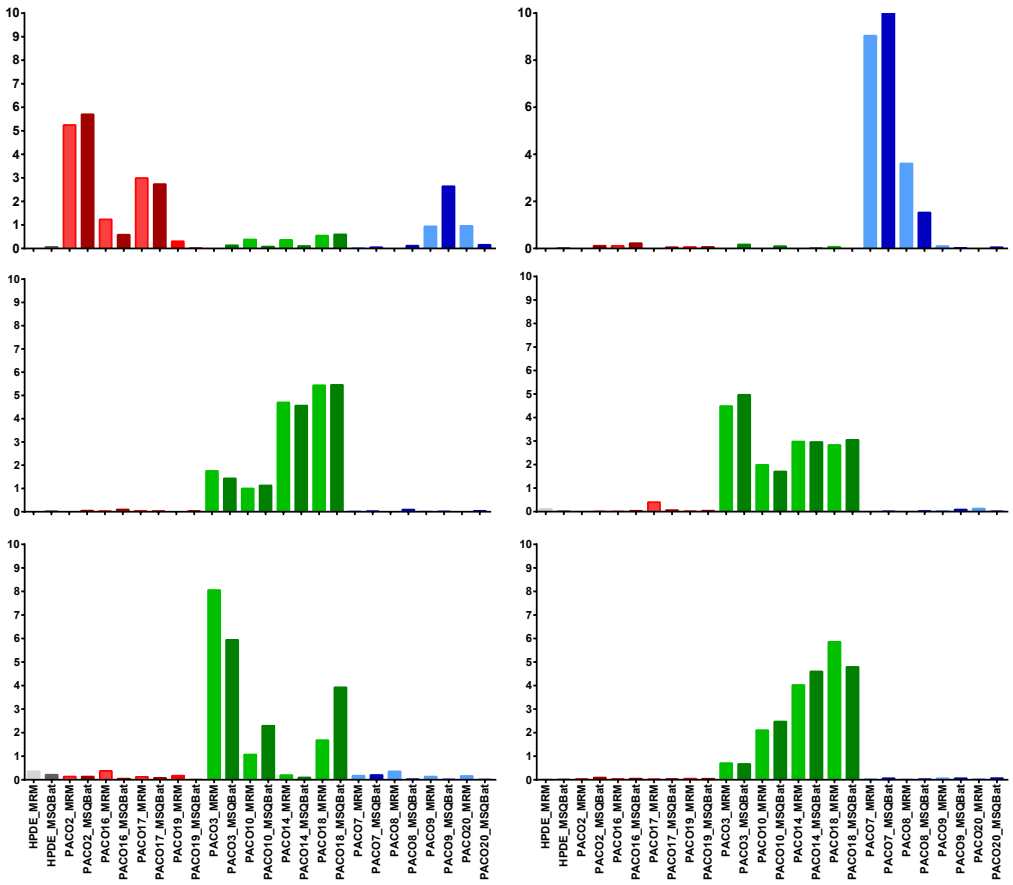


Figure 18.1: Comparison of protein quantifications determined by MS_QBAT and by MRM. Displayed are MS_QBAT and MRM quantification results for six different subtype-specific markers. Different samples (cell lines) are plotted on the x-axis. Red, green, and blue indicate the three different subtypes. Light colors represent MRM measurements; dark colors MS_QBAT protein quantification results. Fold changes are plotted on the y-axis (linear scale). Fold changes have been normalized to the average of all measured regulations. Printed with permission from [279].

19

Closing Remarks and Outlook

MS_QBAT was developed to extract quantitative information from LC-/GeLC-MALDI-MS data. The software package has been validated by using a range of different spike-in experiments and was successfully applied in "real-world" applications and biomarker discovery. Protein regulations of identified biomarkers experimentally determined using MS_QBAT, were confirmed by MRM, qPCR, and immunofluorescence experiments. Furthermore, MS_QBAT was tested for the handling of GeLC- and LC-ESI-MS data, acknowledging MS_QBAT's high flexibility and uniqueness in terms of applicability in different proteomic workflows.

MS_QBAT was programmed from scratch, implementing all basic algorithms necessary to perform ion intensity-based label-free peptide and protein quantification. Additionally, the software package comprises a number of extensions to this basic protein quantification such as an exclusion- and inclusion list generator, dynamic and automatic sample complexity reduction as well as an automatic parameter configuration based on genetic algorithms.

While MS_QBAT has primary been developed for the quantification of LC-MALDI-MS data, the software offers the potential for further extensions, especially in the direction of (LC-)ESI data processing. Processing of data from various LC-MS workflows is possible with the current version of the software. However, these workflows can only be implemented with additional knowledge of the underlying data, and the preconfigured default values only apply for an LC-MALDI-MS setup. The implementation of configurations (i.e., sets of specific parameters) for other setups available to users by applying a simple drop-down selection is a desirable extension for the near future. Along this line, the extension to the automatic parameter-finding algorithm with genetic algorithms is a promising future project. Currently, its application is limited to the alignment of samples. Nevertheless, automatic parameter finding could as well be utilized in the context of feature extraction. The feature extraction algorithm requires parameters defining the expected mass-accuracy as well as the feature gap size characterizing separation capacities of the LC system used.

To obtain a software package "fit for all purposes", the implementation of label-based quantification approaches could be considered, even though their popularity is dwindling. MS$_Q$BAT provides an extensive framework for computational proteomics, from which large parts have been translocated to the independent library *BioUtils - Proteomics*. This library provides all functionalities required for the implementation of label-based quantification approaches. This is why the realization of such a feature would be comparably easy and would be achievable in little time.

Low-level data processing, namely peak finding and background subtraction, deisotoping and charge deconvolution, is another candidate for a "fit for all purposes" software package. The current version of MS$_Q$BAT does not support low-level data processing and this initial data processing step needs to be performed by third-party software such as DataExplorer or DeconTools.

Statistical evaluation of experimentally determined protein regulations and other steps typically following protein quantification have so far been implemented only partly. Per default, MS$_Q$BAT calculates p-values for each regulated protein. Outlier removal, p-value correction, and calculation of a false discovery rate (FDR) are typical examples for procedures following protein quantification and have not been implemented so far.

MS$_Q$BAT was programmed in a highly modular manner, providing extensive possibility for modification/extension, and the implementation of the above-mentioned features is therefore not particularly difficult. The modular design of MS$_Q$BAT provides enough encapsulation of the different functionalities involved in a complete label-free quantification workflow, so that the development of a novel plug-in does not require knowledge of the whole system.

MS$_Q$BAT not only provides great potential for extending the software and/or the implemented quantification algorithms. It is conceivable to use MS$_Q$BAT for studies related to technical aspects of LC-MS data acquisition and the used instruments. MS$_Q$BAT was used to process highly diverse LC-MS data obtained from a MALDI TOF-TOF-, QTRAP-, and an LTQ Orbitrap MS instrument. These instruments show significant differences in resolution, which results, for example, in highly differing numbers of detected peaks. Interestingly, applying the developed feature extraction algorithm to samples containing peak data from these MS instruments resulted in similar feature counts (LC-MALDI-MS: peaks: 520,255, features: 166,571; LC-ESI-MS: peaks: 2,687,146, features: 219,304). While the number of peaks is more than five times higher in the latter instrument setup, the detected number of features is just 1.3 times increased.

Based on these preliminary data, it could be hypothesized that in terms of protein quantification, (very) high resolution instruments are beneficial not primary because of higher feature counts but because of higher peak counts representing the same feature. As a result, the feature AUC is much better covered, which subsequently leads to a more accurate peptide and protein quantification. The current build of MS_QBAT already provides excellent possibility to inspect LC-MS data visually. Characterizing LC-MS data in more detail can certainly be a topic of further, more technical studies.

MS_QBAT, PepSir, and other software developed in the context of this work, as well as the respective source code, are freely available at http://proteomicstools.org.

Bibliography

[1] Cox, J. *et al. MaxLFQ allows accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction.* Molecular & Cellular Proteomics, pp. M113.031591–, 2014.

[2] Weinberg, R. *The biology of cancer.* Garland Science, 2007.

[3] Weinberg, R. *One Renegade Cell: How Cancer Begins: The Quest for the Origins of Cancer.* Basic Books, 1998.

[4] Wang, J.C.Y. *et al. Cancer stem cells: Lessons from leukemia.* Trends in Cell Biology, vol. 15(9):494–501, 2005.

[5] Caldas, C. *Cancer sequencing unravels clonal evolution.* Nature Biotechnology, vol. 30(5):408–410, 2012.

[6] Carney, D.N. *et al. Demonstration of the stem cell nature of clonogenic tumor cells from lung cancer patients.* Stem cells, vol. 1:149–164, 1982.

[7] Pierce, G.B. *et al. Tumors as caricatures of the process of tissue renewal: Prospects for therapy by directing differentiation.* Cancer Research, vol. 48(8):1996–2004, 1988.

[8] KLEIN, G. *et al. Conversion of solid neoplasms into ascites tumors.* Annals of the New York Academy of Sciences, vol. 63:640–661, 1956.

[9] Makino, S. *Further evidence favoring the concept of the stem cell in ascites tumors of rats.* Annals of the New York Academy of Sciences, vol. 63:818–830, 1956.

[10] Henderson, J.S. *et al. The plating of tumor components on the subcutaneous expanses of young mice. Findings with benign and malignant epidermal growths and with mammary carcinomas.* The Journal of experimental medicine, vol. 115:1211–1230, 1962.

[11] Gray, J.M. *et al. Relationship Between Growth Rate and Differentiation of Melanoma in Vivo.* Journal of the National Cancer Institute, vol. 32(6):1201–1210, 1964.

[12] Prehn, R.T. *Analysis of antigenic heterogeneity within individual 3-methylcholanthrene-induced mouse sarcomas.* Journal of the National Cancer Institute, vol. 45:1039–1045, 1970.

[13] Fidler, I.J. *Tumor Heterogeneity and the Biology of Cancer Invasion and Metastasis.* Cancer Research, vol. 38(September):2651–2660, 1978.

[14] Shipitsin, M. *et al. Molecular Definition of Breast Tumor Heterogeneity.* Cancer Cell, vol. 11(March):259–273, 2007.

[15] Brocks, D. *et al. Intratumor DNA Methylation Heterogeneity Reflects Clonal Evolution in Aggressive Prostate Cancer.* Cell Reports, pp. 798–806, 2014.

[16] Bonnet, D. *et al. Human acute myeloid leukemia is organized as a hierarchy that originates from a primitive hematopoietic cell.* Nature medicine, vol. 3:730–737, 1997.

[17] Patrawala, L. *et al. Hierarchical organization of prostate cancer cells in xenograft tumors: the CD44+alpha2beta1+ cell population is enriched in tumor-initiating cells.* Cancer research, vol. 67(22):6796–6805, 2007.

[18] Chen, R. *et al. A Hierarchy of Self-Renewing Tumor-Initiating Cell Types in Glioblastoma.* Cancer Cell, vol. 17(4):362–375, 2010.

[19] Melchor, L. *et al. Single-cell genetic analysis reveals the composition of initiating clones and phylogenetic patterns of branching and parallel evolution in myeloma.* Leukemia, vol. 28(8):1705–15, 2014.

[20] Bruce, W.R. *et al. a Quantitative Assay for the Number of Murine Lymphoma Cells Capable of Proliferation in Vivo.* Nature, vol. 199:79–80, 1963.

[21] Sabbath, K.D. *et al. Heterogeneity of clonogenic cells in acute myeloblastic leukemia.* Journal of Clinical Investigation, vol. 75:746–753, 1985.

[22] Lapidot, T. *et al. A cell initiating human acute myeloid leukaemia after transplantation into SCID mice.* Nature, vol. 367:645–648, 1994.

[23] Ailles, L.E. *et al. Growth characteristics of acute myelogenous leukemia progenitors that initiate malignant hematopoiesis in nonobese diabetic/severe combined immunodeficient mice.* Blood, vol. 94:1761–1772, 1999.

[24] Singh, S.K. *et al. Identification of a cancer stem cell in human brain tumors.* Cancer Research, vol. 63:5821–5828, 2003.

[25] Al-Hajj, M. *et al. Prospective identification of tumorigenic breast cancer cells.* Proceedings of the National Academy of Sciences of the United States of America, vol. 100:3983–3988, 2003.

[26] Bapat, S.a. *et al. Stem and progenitor-like cells contribute to the aggressive behavior of human epithelial ovarian cancer.* Cancer Research, vol. 65:3025–3029, 2005.

[27] Collins, A.T. *et al. Prospective identification of tumorigenic prostate cancer stem cells.* Cancer Research, vol. 65:10946–10951, 2005.

[28] Collins, A.T. *et al. Prostate cancer stem cells.* European Journal of Cancer, vol. 42(9):1213–1218, 2006.

[29] Wang, J. *et al. Symmetrical and asymmetrical division analysis provides evidence for a hierarchy of prostate epithelial cell lineages.* Nature communications, vol. 5(May):4758, 2014.

[30] O'Brien, C.a. *et al. A human colon cancer cell capable of initiating tumour growth in immunodeficient mice.* Nature, vol. 445:106–110, 2007.

[31] Ricci-Vitiani, L. *et al. Identification and expansion of human colon-cancer-initiating cells.* Nature, vol. 445:111–115, 2007.

[32] Dalerba, P. *et al. Phenotypic characterization of human colorectal cancer stem cells.* Proceedings of the National Academy of Sciences of the United States of America, vol. 104:10158–10163, 2007.

[33] Barker, N. *et al. Identification of stem cells in small intestine and colon by marker gene Lgr5.* Nature, vol. 449:1003–1007, 2007.

[34] Yang, Z.F. *et al. Significance of CD90+ Cancer Stem Cells in Human Liver Cancer.* Cancer Cell, vol. 13:153–166, 2008.

[35] Sell, S. *et al. Liver cancer stem cells.* Journal of Clinical Oncology, vol. 26(17):2800–2805, 2008.

[36] Eramo, a. *et al. Identification and expansion of the tumorigenic lung cancer stem cell population.* Cell death and differentiation, vol. 15:504–514, 2008.

[37] Hermann, P.C. *et al. Distinct Populations of Cancer Stem Cells Determine Tumor Growth and Metastatic Activity in Human Pancreatic Cancer.* Cell Stem Cell, vol. 1:313–323, 2007.

[38] Prince, M.E.P. *et al. Cancer stem cells in head and neck squamous cell cancer.* Journal of Clinical Oncology, vol. 26(17):2871–2875, 2008.

[39] Takaishi, S. *et al. Gastric cancer stem cells.* Journal of Clinical Oncology, vol. 26(17):2876–2882, 2008.

[40] Suvà, M.L. *et al. Identification of cancer stem cells in Ewing's sarcoma.* Cancer research, vol. 69(5):1776–1781, 2009.

[41] Kelly, P.N. *et al. Supporting Online Material for Tumor Growth Need Not Be Driven by Rare Cancer Stem Cells.* Methods, vol. 337(5836):337, 2007.

[42] Kennedy, J.a. *et al. Supporting Online Material.* Methods, vol. 318(5857):1722; author reply 1722, 2007.

[43] Kennedy, J.a. *et al. Comment on "Tumor growth need not be driven by rare cancer stem cells".* Science (New York, N.Y.), vol. 318(5857):1722; author reply 1722, 2007.

[44] Chen, J. *et al. A restricted cell population propagates glioblastoma growth after chemotherapy*, 2012.

[45] Driessens, G. *et al. Defining the mode of tumour growth by clonal analysis*, 2012.

[46] Schepers, a.G. *et al. Lineage Tracing Reveals Lgr5+ Stem Cell Activity in Mouse Intestinal Adenomas.* Science, vol. 337(6095):730–735, 2012.

[47] Fialkow, P.J. *et al. Clonal origin of chronic myelocytic leukemia in man.* Proceedings of the National Academy of Sciences of the United States of America, vol. 58:1468–1471, 1967.

[48] Spangrude, G.J. *et al. Purification and characterization of mouse hematopoietic stem cells.* Science (New York, N.Y.), vol. 241:58–62, 1988.

[49] Morrison, S.J. *et al. The long-term repopulating subset of hematopoietic stem cells is deterministic and isolatable by phenotype.* Immunity, vol. 1:661–673, 1994.

[50] Baum, C.M. *et al. Isolation of a candidate human hematopoietic stem-cell population.* Proceedings of the National Academy of Sciences of the United States of America, vol. 89(April):2804–2808, 1992.

[51] Osawa, M. *et al. Long-term lymphohematopoietic reconstitution by a single CD34-low/negative hematopoietic stem cell.* Blood, vol. 273(1995):3–6, 1996.

[52] Singh, S.K. *et al. Cancer stem cells in nervous system tumors.* Oncogene, vol. 23(43):7267–7273, 2004.

[53] Beck, B. *et al. Unravelling cancer stem cell potential.* Nature reviews. Cancer, vol. 13(10):727–38, 2013.

[54] Valent, P. *et al. Cancer stem cell definitions and terminology: the devil is in the details.* Nature reviews. Cancer, vol. 12(11):767–75, 2012.

[55] Kreso, A. *et al. Evolution of the cancer stem cell model.* Cell Stem Cell, vol. 14(3):275–291, 2014.

[56] Lyle, S. *et al. Quiescent, slow-cycling stem cell populations in cancer: A review of the evidence and discussion of significance.* Journal of Oncology, vol. 2011, 2011.

[57] Fidler, I.J. *et al. Metastasis results from preexisting variant cells within a malignant tumor.* Science (New York, N.Y.), vol. 197(7):893–895, 1977.

[58] Fidler, I.J. *et al. Biological diversity in metastatic neoplasms: origins and implications.* Science (New York, N.Y.), vol. 217(September):998–1003, 1982.

[59] Baccelli, I. *et al. Identification of a population of blood circulating tumor cells from breast cancer patients that initiates metastasis in a xenograft assay.* Nature biotechnology, vol. 31(6):539–44, 2013.

[60] Magee, J.a. *et al. Cancer Stem Cells: Impact, Heterogeneity, and Uncertainty.* Cancer Cell, vol. 21(3):283–296, 2012.

[61] Wilson, A. *et al. Hematopoietic Stem Cells Reversibly Switch from Dormancy to Self-Renewal during Homeostasis and Repair.* Cell, vol. 135(6):1118–1129, 2008.

[62] Meacham, C.E. *et al. Tumour heterogeneity and cancer cell plasticity.* Nature, vol. 501(7467):328–37, 2013.

[63] Oskarsson, T. *et al. Metastatic stem cells: Sources, niches, and vital pathways.* Cell Stem Cell, vol. 14(3):306–321, 2014.

[64] Singh, a. *et al. EMT, cancer stem cells and drug resistance: an emerging axis of evil in the war on cancer.* Oncogene, vol. 29:4741–4751, 2010.

[65] Jamieson, C.H.M. *et al. Granulocyte-macrophage progenitors as candidate leukemic stem cells in blast-crisis CML.* The New England journal of medicine, vol. 351(7):657–667, 2004.

[66] Reya, T. *et al. Stem cells, cancer, and cancer stem cells.* Nature, vol. 414(6859):105–111, 2001.

[67] Kaiser, B.J. *et al. The cancer stem cell gamble.* Science, vol. 347, 2015.

[68] Wilson, A. *et al. Balancing dormant and self-renewing hematopoietic stem cells.* Current Opinion in Genetics and Development, vol. 19(5):461–468, 2009.

[69] Domen, J. *et al. Systemic overexpression of BCL-2 in the hematopoietic system protects transgenic mice from the consequences of lethal irradiation.* Blood, vol. 91(7):2272–2282, 1998.

[70] Peters, R. *et al. Apoptotic regulation in primitive hematopoietic precursors.* Blood, vol. 92:2041–2052, 1998.

[71] Zhou, S. *et al. The ABC transporter Bcrp1/ABCG2 is expressed in a wide variety of stem cells and is a molecular determinant of the side-population phenotype.* Nature medicine, vol. 7(9):1028–1034, 2001.

[72] Hill, R.P. *et al. Cancer Stem Cells, Hypoxia and Metastasis.* Seminars in Radiation Oncology, vol. 19(2):106–111, 2009.

[73] Diehn, M. *et al. Therapeutic Implications of the Cancer Stem Cell Hypothesis.* Seminars in Radiation Oncology, vol. 19(2):78–86, 2009.

[74] Kreso, A. *et al. Self-renewal as a therapeutic target in human colorectal cancer.* Nature medicine, vol. 20:29–36, 2014.

[75] Schillert, A. *et al. Label retaining cells in cancer - The dormant root of evil?* Cancer Letters, vol. 341(1):73–79, 2013.

[76] Harrison, D.E. *et al. Most primitive hematopoietic stem cells are stimulated to cycle rapidly after treatment with 5-fluorouracil.* Blood, vol. 78:1237–1240, 1991.

[77] Deininger, M. *et al. The development of imatinib as a therapeutic agent for chronic myeloid leukemia Review in translational hematology The development of imatinib as a therapeutic agent for chronic myeloid leukemia.* Development, vol. 105(7):2640–2653, 2012.

[78] Tallman, M.S. *et al. Drug therapy for acute myeloid leukemia*, 2005.

[79] Krause, D.S. *et al. Right on target: eradicating leukemic stem cells.* Trends in Molecular Medicine, vol. 13(11):470–481, 2007.

[80] Milas, L. *et al. Cancer Stem Cells and Tumor Response to Therapy: Current Problems and Future Prospects.* Seminars in Radiation Oncology, vol. 19(2):96–105, 2009.

[81] Essers, M.A.G. *et al. Targeting leukemic stem cells by breaking their dormancy.* Molecular Oncology, vol. 4(5):443–450, 2010.

[82] Gupta, P.B. *et al. Identification of Selective Inhibitors of Cancer Stem Cells by High-Throughput Screening.* Cell, vol. 138(4):645–659, 2009.

[83] Pardal, R. *et al. Applying the principles of stem-cell biology to cancer.* Nature reviews. Cancer, vol. 3(December):895–902, 2003.

[84] *National institute of health.* http://www.niehs.nih.gov/health/topics/science/biomarkers/. Accessed: 2014-12-30.

[85] Strimbu, K. *et al. NIH Public Access.* Curr Opin HIV AIDS, vol. 5(6):463–466, 2011.

[86] Frantzi, M. *et al. Clinical proteomic biomarkers: relevant issues on study design & technical considerations in biomarker development.* Clinical and translational medicine, vol. 3:7, 2014.

[87] Diamandis, E.P. *Point: Proteomic patterns in biological fluids: Do they represent the future of cancer diagnostics?* Clinical Chemistry, vol. 49(8):1272–1275, 2003.

[88] Petricoin, E. *et al. Counterpoint: The vision for a new diagnostic paradigm*, 2003.

[89] Diamandis, E.P. *Re: Serum proteomic patterns for detection of prostate cancer.* Journal of the National Cancer Institute, vol. 95(6):489–490; author reply 490–491, 2003.

[90] Petricoin, E.F. *et al. Use of proteomic patterns in serum to identify ovarian cancer.* Lancet, vol. 359(9306):572–577, 2002.

[91] Dettmer, K. *et al. Mass spectrometry-based metabolomics.* Mass Spectrometry Reviews, vol. 26(1):51–78, 2007.

[92] Werner, T. *Next generation sequencing in functional genomics.* Briefings in Bioinformatics, vol. 11(5):499–511, 2010.

[93] Gygi, S.P. *et al. Correlation between protein and mRNA abundance in yeast.* Molecular and cellular biology, vol. 19(3):1720–1730, 1999.

[94] Greenbaum, D. *et al. Comparing protein abundance and mRNA expression levels on a genomic scale.* Genome biology, vol. 4(9):117, 2003.

[95] Schwanhäusser, B. *et al. Global quantification of mammalian gene expression control.* Nature, vol. 473:337–342, 2011.

[96] Ning, K. *et al. Comparative analysis of different label-free mass spectrometry based protein abundance estimates and their correlation with RNA-Seq gene expression data.* Journal of Proteome Research, vol. 11(4):2261–2271, 2012.

[97] Cartegni, L. *et al. Listening to silence and understanding nonsense: exonic mutations that affect splicing.* Nature reviews. Genetics, vol. 3(4):285–298, 2002.

[98] Zhang, X. *et al. Multi-dimensional liquid chromatography in proteomics-A review.* Analytica Chimica Acta, vol. 664(2):101–113, 2010.

[99] Cabezas-Wallscheid, N. *et al. Identification of Regulatory Networks in HSCs and Their Immediate Progeny via Integrated Proteome, Transcriptome, and DNA Methylome Analysis.* Cell Stem Cell, vol. 4:1–16, 2014.

[100] Russell, H. *et al. A Mass Spec Timeline.* Today'S Chemist AT Work, pp. 47–49, 2003.

[101] Hunt, D.F. *et al. Characterization of peptides bound to the class I MHC molecule HLA-A2.1 by mass spectrometry.* Science, vol. 255(5049):1261–1263, 1992.

[102] Eng, J.K. *et al. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database.* Journal of the American Society for Mass Spectrometry, vol. 5(11):976–989, 1994.

[103] Aebersold, R. *et al. Mass spectrometry-based proteomics.* Nature, vol. 422(6928):198–207, 2003.

[104] Gehlenborg, N. *et al. Diplomarbeit Bioinformatik Visualization and Integration of Liquid Chromatography / Mass Spectrometry Proteomics Data.* Diplomarbeit, Eberhard Karls Universität Tübingen, 2006.

[105] de Hoffmann E. *et al. Mass Spectrometry - Principles and Application.* Wiley, third edition, 2007.

[106] Lottspeich F., Engels J.W., Z.L.S. *Bioanalytik.* Spektrum Akademischer Verlag, third edition, 2012.

[107] Gelpí, E. *Interfaces for coupled liquid-phase separation/mass spectrometry techniques. An update on recent developments.* Journal of Mass Spectrometry, vol. 37(February):241–253, 2002.

[108] Klampfl, C.W. *Review coupling of capillary electrochromatography to mass spectrometry.* Journal of Chromatography A, vol. 1044(1-2):131–144, 2004.

[109] Zhao, Y.Y. *et al. UPLC-MSE application in disease biomarker discovery: The discoveries in proteomics to metabolomics.* Chemico-Biological Interactions, vol. 215:7–16, 2014.

[110] Zhao, Y.Y. *et al. Ultra-performance liquid chromatography-mass spectrometry as a sensitive and powerful technology in lipidomic applications.* Chemico-Biological Interactions, vol. 220:181–192, 2014.

[111] Buchanan, T.S. *et al. Neuromusculoskeletal modeling: Estimation of muscle forces and joint moments and movements from measurements of neural command.* Journal of Applied Biomechanics, vol. 20(10):367–395, 2004.

[112] Bantscheff, M. *et al. Quantitative mass spectrometry in proteomics: A critical review.* Analytical and Bioanalytical Chemistry, vol. 389(4):1017–1031, 2007.

[113] Schirle, M. *et al. Profiling core proteomes of human cell lines by one-dimensional PAGE and liquid chromatography-tandem mass spectrometry.* Molecular & cellular proteomics : MCP, vol. 2(12):1297–1305, 2003.

[114] Rayleigh, L. *XX. On the equilibrium of liquid conducting masses charged with electricity.* Philosophical Magazine Series 5, vol. 14:184–186, 1882.

[115] Wilm, M.S. *et al. Electrospray and Taylor-Cone theory, Dole's beam of macromolecules at last?* International Journal of Mass Spectrometry and Ion Processes, vol. 136(2-3):167–180, 1994.

[116] Bruins, a.P. *et al. Ion Spray Interface for Combined Liquid Chromatography/Atmospheric Pressure Ionization Mass Spectrometry.* Anal.Chem., vol. 59:2642–2646, 1987.

[117] Meesters, G. *et al. Generation of micron-sized droplets from the Taylor cone.* Journal of Aerosol Science, vol. 23(l):37–49, 1992.

[118] Mann, M. *et al. Interpreting Mass Spectra of Multiply Charged Ions.* Analytical Chemistry, vol. 61(15):1702–1708, 1989.

[119] Fenn, J.B. *et al. Electrospray ionization for mass spectrometry of large biomolecules.* Science (New York, N.Y.), vol. 246(6):64–71, 1989.

[120] Ho, C.S. *et al. Electrospray ionisation mass spectrometry: principles and clinical applications.* The Clinical biochemist. Reviews / Australian Association of Clinical Biochemists, vol. 24(February):3–12, 2003.

[121] Zeleny, J. *The electrical discharge from liquid points, and a hydrostatic method of measuring the electric intensity at their surfaces.* Physical Review, vol. 3(2):69–91, 1914.

[122] Wilson, C.T.R. *et al. The bursting of soap-bubbles in a uniform electric field.* Mathematical Proceedings of the Cambridge Philosophical Society, vol. 22:728–730, 1925.

[123] Macky, W.a. *Some Investigations on the Deformation and Breaking of Water Drops in Strong Electric Fields.* Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 133(822):565–587, 1931.

[124] Taylor, G. *Disintegration of water drops in an electric field.* Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, vol. 280(1382):383–397, 1964.

[125] Dole, M. *et al. Molecular Beams of Macroions.* The Journal of Chemical Physics, vol. 52(1968):4977, 1970.

[126] Dreisewerd, K. *The desorption process in MALDI*, 2003.

[127] Zenobi, R. *et al. Ion formation in maldi mass spectrometry.* Mass Spectrom. Rev., vol. 17(5):337–366, 1998.

[128] Knochenmuss, R. *et al. MALDI ionization: The role of in-plume processes.* Chemical Reviews, vol. 103(2):441–452, 2003.

[129] Knochenmuss, R. *A quantitative model of ultraviolet matrix-assisted laser desorption/ionization including analyte ion generation.* Analytical Chemistry, vol. 75(10):2199–2207, 2003.

[130] Karas, M. *et al. Ion formation in MALDI: The cluster ionization mechanism.* Chemical Reviews, vol. 103(2):427–439, 2003.

[131] Dreisewerd, K. *et al. Fundamentals of matrix-assisted laser desorption/ionization mass spectrometry with pulsed infrared lasers.* International Journal of Mass Spectrometry, vol. 226:189–209, 2003.

[132] Karas, M. *et al. Influence of the wavelength in high-irradiance ultraviolet laser desorption mass spectrometry of organic molecules.* Analytical Chemistry, vol. 57:2935–2939, 1985.

[133] Karas, M. *et al. Matrix-assisted ultraviolet laser desorption of non-volatile compounds.* International Journal of Mass Spectrometry and Ion Processes, vol. 78:53–68, 1987.

[134] Tanaka, K. *et al. Protein and Polymer Analyses up to mlz 100 000 by Laser Ionization Time-of-flight Mass Spectrometry.* Rapid Communications in Mass Spectrometry, vol. 2(8):151–153, 1988.

[135] Karas, M. *et al. Laser desorption ionization of proteins with molecular masses exceeding 10,000 daltons.* Analytical chemistry, vol. 60(29):2299–2301, 1988.

[136] Beavis, R.C. *et al. Matrix-assisted laser-desorption mass spectrometry using 355 nm radiation.* Rapid communications in mass spectrometry : RCM, vol. 3:436–439, 1989.

[137] Karas, M. *et al. Laser desorption ionization mass spectrometty of large biomolecules.* Science, vol. 9:321–325, 1990.

[138] Sauer, S. *et al. Mass spectrometry tools for the classification and identification of bacteria.* Nature reviews. Microbiology, vol. 8(1):74–82, 2010.

[139] Weickhardt, C. *et al. Time-of-flight mass spectrometry: State-of the-art in chemical analysis and molecular science.* Mass Spectrometry Reviews, vol. 15(October):139–162, 1996.

[140] Wiley, W.C. *et al. Time-of-flight mass spectrometer with improved resolution.* Review of Scientific Instruments, vol. 26(1955):1150–1157, 1955.

[141] Guilhaus, M. *Principles and instrumentation in time-of-flight mass spectrometry: Physical and instrumental concepts.* Journal of Mass Spectrometry, vol. 30(11):1519–1532, 1995.

[142] Mann, M. *et al. Developments in matrix-assisted laser desorption/ionization peptide mass spectrometry.* Current Opinion in Biotechnology, vol. 7:11–19, 1996.

[143] Glish, G.L. *et al. The basics of mass spectrometry in the twenty-first century.* Nature reviews. Drug discovery, vol. 2(2):140–150, 2003.

[144] March, R. *An Introduction to Quadrupole Ion Trap Mass Spectrometry.* J. Mass Spectrom., vol. 32(February):351–369, 1997.

[145] Douglas, D.J. *et al. Linear ion traps in mass spectrometry.* Mass Spectrometry Reviews, vol. 24(1):1–29, 2005.

[146] Todd, J.F.J. *Ion trap mass spectrometer—past, present, and future (?).* Mass Spectrometry Reviews, vol. 10(1):3–52, 1991.

[147] Stafford, G.C. *et al. Recent improvements in and analytical applications of advanced ion trap technology.* International Journal of Mass Spectrometry and Ion Processes, vol. 60(1):85–98, 1984.

[148] Scigelova, M. *et al. Fourier transform mass spectrometry.* Molecular & cellular proteomics : MCP, vol. 10(7):M111.009431, 2011.

[149] Hu, Q. *et al. The Orbitrap: A new mass spectrometer.* Journal of Mass Spectrometry, vol. 40(4):430–443, 2005.

[150] Makarov, A. *Electrostatic axially harmonic orbital trapping: A high-performance technique of mass analysis.* Analytical Chemistry, vol. 72(6):1156–1162, 2000.

[151] Coates, P.B. *Analytical corrections for dead time effects in the measurement of time-interval distributions.* Review of Scientific Instruments, vol. 63(1992):2084–2088, 1992.

[152] Schultz, J.C. *et al. Polymerase chain reaction products analyzed by charge detection mass spectrometry.* Rapid Communications in Mass Spectrometry, vol. 13(1):15–20, 1999.

[153] Hilton, G.C. *et al. Impact energy measurement in time-of-flight mass spectrometry with cryogenic microcalorimeters.* Nature, vol. 391(6668):672–675, 1998.

[154] Koppenaal, D.W. *et al. MS detectors.* Analytical chemistry, vol. 77(21):418A–427A, 2005.

[155] Page, J.S. *et al. FTICR mass spectrometry for qualitative and quantitative bioanalyses.* Current Opinion in Biotechnology, vol. 15(1):3–11, 2004.

[156] Schrader, W. *et al. Liquid chromatography/Fourier transform ion cyclotron resonance mass spectrometry (LC-FTICR MS): An early overview.* Analytical and Bioanalytical Chemistry, vol. 379(7-8):1013–1024, 2004.

[157] McLafferty, F.W. *Tandem mass spectrometry.* Science, vol. 214(October):280–287, 1981.

[158] De Hoffmann, E. *Tandem mass spectrometry: A primer.* Journal of Mass Spectrometry, vol. 31:129–137, 1996.

[159] Perkins, D.N. *et al. Probability-based protein identification by searching sequence databases using mass spectrometry data.* Electrophoresis, vol. 20(18):3551–3567, 1999.

[160] Craig, R. *et al. A method for reducing the time required to match protein sequences with tandem mass spectra.* Rapid communications in mass spectrometry : RCM, vol. 17(20):2310–2316, 2003.

[161] Shilov, I.V. *et al. The Paragon Algorithm, a next generation search engine that uses sequence temperature values and feature probabilities to identify peptides from tandem mass spectra.* Molecular & cellular proteomics : MCP, vol. 6(9):1638–1655, 2007.

[162] Cox, J. *et al. Andromeda: A peptide search engine integrated into the MaxQuant environment.* Journal of Proteome Research, vol. 10(4):1794–1805, 2011.

[163] Sandin, M. *et al. Data processing methods and quality control strategies for label-free LC-MS protein quantification.* Biochimica et biophysica acta, vol. 1844(1 Pt A):29–41, 2014.

[164] Geiger, T. *et al. Proteomics on an Orbitrap benchtop mass spectrometer using all-ion fragmentation.* Molecular & cellular proteomics : MCP, vol. 9(10):2252–2261, 2010.

[165] Domon, B. *et al. Mass spectrometry and protein analysis.* Science (New York, N.Y.), vol. 312(5771):212–217, 2006.

[166] Michalski, A. *et al. More than 100,000 detectable peptide species elute in single shotgun proteomics runs but the majority is inaccessible to data-dependent LC-MS/MS.* Journal of Proteome Research, vol. 10(4):1785–1793, 2011.

[167] Liu, H. *et al. A model for random sampling and estimation of relative protein abundance in shotgun proteomics.* Analytical Chemistry, vol. 76(14):4193–4201, 2004.

[168] Geromanos, S.J. *et al. The detection, correlation, and comparison of peptide precursor and product ions from data independent LC-MS with data dependant LC-MS/MS.* Proteomics, vol. 9(6):1683–1695, 2009.

[169] Egertson, J.D. *et al. Multiplexed MS/MS for improved data-independent acquisition.* Nature methods, vol. 10(8):744–6, 2013.

[170] Panchaud, A. *et al. Precursor acquisition independent from ion count: How to dive deeper into the proteomics ocean.* Analytical Chemistry, vol. 81(15):6481–6488, 2009.

[171] Panchaud, A. *et al. Faster, quantitative, and accurate precursor acquisition independent from ion count.* Analytical Chemistry, vol. 83(6):2250–2257, 2011.

[172] Gillet, L.C. *et al. Targeted Data Extraction of the MS/MS Spectra Generated by Data-independent Acquisition: A New Concept for Consistent and Accurate Proteome Analysis.* Molecular & Cellular Proteomics, vol. 11(6):O111.016717–O111.016717, 2012.

[173] Silva, J.C. *et al. Absolute quantification of proteins by LCMSE: a virtue of parallel MS acquisition.* Molecular & cellular proteomics : MCP, vol. 5(1):144–156, 2006.

[174] Anderson, L. *et al. Quantitative mass spectrometric multiple reaction monitoring assays for major plasma proteins.* Molecular & cellular proteomics : MCP, vol. 5(4):573–588, 2006.

[175] Kitteringham, N.R. *et al. Multiple reaction monitoring for quantitative biomarker analysis in proteomics and metabolomics.* Journal of Chromatography B: Analytical Technologies in the Biomedical and Life Sciences, vol. 877(13):1229–1239, 2009.

[176] Gillette, M.a. *et al. Quantitative analysis of peptides and proteins in biomedicine by targeted mass spectrometry.* Nature methods, vol. 10(1):28–34, 2013.

[177] Krutchinsky, A.N. *et al. On the nature of the chemical noise in MALDI mass spectra.* Journal of the American Society for Mass Spectrometry, vol. 13:129–134, 2002.

[178] Yang, C. *et al. Comparison of public peak detection algorithms for MALDI mass spectrometry data analysis.* BMC bioinformatics, vol. 10:4, 2009.

[179] Coombes, K.R. *et al. Quality control and peak finding for proteomics data collected from nipple aspirate fluid by surface-enhanced laser desorption and ionization.* Clinical Chemistry, vol. 49(10):1615–1623, 2003.

[180] Hilario, M. *et al. Machine learning approaches to lung cancer prediction from mass spectra.* Proteomics, vol. 3(9):1716–1719, 2003.

[181] Coombes, K.R. *et al. Improved peak detection and quantification of mass spectrometry data acquired from surface-enhanced laser desorption and ionization by denoising spectra with the undecimated discrete wavelet transform.* Proteomics, vol. 5(16):4107–4117, 2005.

[182] Listgarten, J. *et al. Statistical and computational methods for comparative proteomic profiling using liquid chromatography-tandem mass spectrometry.* Molecular & cellular proteomics : MCP, vol. 4(4):419–434, 2005.

[183] Du, P. *et al. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching.* Bioinformatics, vol. 22(17):2059–2065, 2006.

[184] Chapman, J.R. *et al. Advantages of High-Resolution and High-Mass Range Magnetic-Sector Mass Spectrometry for Electrospray Ionization.* Organic Mass Spectrometry, vol. 27(3):195–203, 1992.

[185] Felitsyn, N. *et al. Origin and number of charges observed on multiply-protonated native proteins produced by ESI.* International Journal of Mass Spectrometry, vol. 219(1):39–62, 2002.

[186] Gross, J.H. *Mass Spectrometry: A Textbook.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[187] Labowsky, M. *et al. Three-dimensional deconvolution of multiply charged spectra.* Rapid Commun. Mass Spectrom, vol. 7(1):71–84, 1993.

[188] Amad, M.H. *et al. Importance of gas-phase proton affinities in determining the electrospray ionization response for analytes and solvents.* Journal of Mass Spectrometry, vol. 35(January):784–789, 2000.

[189] Wehofsky, M. *et al. Automated deconvolution and deisotoping of electrospray mass spectra.* Journal of Mass Spectrometry, vol. 37(2):223–229, 2002.

[190] Maleknia, S.D. *et al. Charge ratio analysis method: Approach for the deconvolution of electrospray mass spectra.* Analytical Chemistry, vol. 77(1):111–119, 2005.

[191] Yuan, Z. *et al. Features-based deisotoping method for tandem mass spectra.* Advances in Bioinformatics, vol. 2011:210805, 2011.

[192] Urban, J. *et al. Fundamental definitions and confusions in mass spectrometry about mass assignment, centroiding and resolution.* TrAC - Trends in Analytical Chemistry, vol. 53:126–136, 2014.

[193] Neubert, H. *et al. Label-Free detection of differential protein expression by LC/MALDI mass spectrometry.* Journal of Proteome Research, vol. 7(6):2270–2279, 2008.

[194] Aebersold, R. *Constellations in a cellular universe.* Nature, vol. 422(6928):115–116, 2003.

[195] Mann, M. *et al. The Coming Age of Complete, Accurate, and Ubiquitous Proteomes.* Molecular Cell, vol. 49(4):583–590, 2013.

[196] Cox, J. *et al. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification.* Nature biotechnology, vol. 26(12):1367–1372, 2008.

[197] de Godoy, L.M.F. *et al. Comprehensive mass-spectrometry-based proteome quantification of haploid versus diploid yeast.* Nature, vol. 455(October):1251–1254, 2008.

[198] Nagaraj, N. *et al. System-wide Perturbation Analysis with Nearly Complete Coverage of the Yeast Proteome by Single-shot Ultra HPLC Runs on a Bench Top Orbitrap.* Molecular & Cellular Proteomics, vol. 11(3):M111.013722–M111.013722, 2012.

[199] Nagaraj, N. *et al. Deep proteome and transcriptome mapping of a human cancer cell line.* Molecular Systems Biology, vol. 7(548):1–8, 2011.

[200] Beck, M. *et al. The quantitative proteome of a human cell line.* Molecular Systems Biology, vol. 7(549):549, 2011.

[201] Thompson, A. *et al. Tandem mass tags: a novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS.* Analytical chemistry, vol. 75(8):1895–1904, 2003.

[202] Choe, L. *et al. 8-Plex quantitation of changes in cerebrospinal fluid protein expression in subjects undergoing intravenous immunoglobulin treatment for Alzheimer's disease.* Proteomics, vol. 7(20):3651–3660, 2007.

[203] Zeng, D. *et al. Improved CILAT reagents for quantitative proteomics.* Bioorganic and Medicinal Chemistry Letters, vol. 19(7):2059–2061, 2009.

[204] Oda, Y. *et al. Accurate quantitation of protein expression and site-specific phosphorylation.* Proceedings of the National Academy of Sciences of the United States of America, vol. 96(12):6591–6596, 1999.

[205] Ong, S.E. *et al. Stable isotope labeling by amino acids in cell culture, SILAC, as a simple and accurate approach to expression proteomics.* Molecular & cellular proteomics : MCP, vol. 1(5):376–386, 2002.

[206] Neher, S.B. *et al. Proteomic Profiling of ClpXP Substrates after DNA Damage Reveals Extensive Instability within SOS Regulon.* Molecular Cell, vol. 22(2):193–204, 2006.

[207] Krijgsveld, J. *et al. Metabolic labeling of C. elegans and D. melanogaster for quantitative proteomics.* Nature biotechnology, vol. 21(8):927–931, 2003.

[208] Krüger, M. *et al. SILAC Mouse for Quantitative Proteomics Uncovers Kindlin-3 as an Essential Factor for Red Blood Cell Function.* Cell, vol. 134(2):353–364, 2008.

[209] Wu, C.C. *et al. Metabolic labeling of mammalian organisms with stable isotopes for quantitative proteomic analysis.* Analytical Chemistry, vol. 76(17):4951–4959, 2004.

[210] Geiger, T. *et al. Use of stable isotope labeling by amino acids in cell culture as a spike-in standard in quantitative proteomics.* Nature protocols, vol. 6(2):147–157, 2011.

[211] Geiger, T. *et al. Super-SILAC mix for quantitative proteomics of human tumor tissue.* Nature methods, vol. 7(5):383–385, 2010.

[212] Bantscheff, M. *et al. Quantitative mass spectrometry in proteomics: Critical review update from 2007 to the present.* Analytical and Bioanalytical Chemistry, vol. 404(4):939–965, 2012.

[213] Zhang, R. *et al. Fractionation of isotopically labeled peptides in quantitative proteomics.* Analytical Chemistry, vol. 73(21):5142–5149, 2001.

[214] Van Hoof, D. *et al. An experimental correction for arginine-to-proline conversion artifacts in SILAC-based quantitative proteomics.* Nature methods, vol. 4(9):677–678, 2007.

[215] Yeoun, J.K. *et al. Reproducibility assessment of relative quantitation strategies for LC-MS based proteomics.* Analytical Chemistry, vol. 79(15):5651–5658, 2007.

[216] Yao, X. *et al. Proteolytic 18O labeling for comparative proteomics: Model studies with two serotypes of adenovirus.* Analytical Chemistry, vol. 73(13):2836–2842, 2001.

[217] Stewart, I.I. *et al. 18O labeling: a tool for proteomics.* Rapid Communications in Mass Spectrometry, vol. 15(24):2456–2465, 2001.

[218] Reynolds, K.J. *et al. Proteolytic 18O labeling for comparative proteomics: Evaluation of endoprotease Glu-C as the catalytic agent.* Journal of Proteome Research, vol. 1(1):27–33, 2002.

[219] Miyagi, M. *et al. Proteolytic 18O-labeling strategies for quantitative proteomics.* Mass Spectrometry Reviews, vol. 26(1):121–136, 2007.

[220] Mueller, L.N. *et al. An assessment of software solutions for the analysis of mass spectrometry based quantitative proteomics data.* Journal of proteome research, vol. 7(1):51–61, 2008.

[221] Johnson, K.L. *et al. A method for calculating 16O/18O peptide ion ratios for the relative quantification of proteomes.* Journal of the American Society for Mass Spectrometry, vol. 15(4):437–445, 2004.

[222] Ramos-Fernández, A. *et al. Improved method for differential expression proteomics using trypsin-catalyzed 18O labeling with a correction for labeling efficiency.* Molecular & cellular proteomics : MCP, vol. 6(7):1274–1286, 2007.

[223] Gygi, S.P. *et al. Quantitative analysis of complex protein mixtures using isotope-coded affinity tags.* Nature biotechnology, vol. 17(10):994–999, 1999.

[224] Schmidt, A. *et al. A novel strategy for quantitative proteomics using isotope-coded protein labels.* Proteomics, vol. 5(1):4–15, 2005.

[225] Ross, P.L. *et al. Multiplexed protein quantitation in Saccharomyces cerevisiae using amine-reactive isobaric tagging reagents.* Molecular & cellular proteomics : MCP, vol. 3(12):1154–1169, 2004.

[226] Bondarenko, P.V. *et al. Identification and relative quantitation of protein mixtures by enzymatic digestion followed by capillary reversed-phase liquid chromatography - Tandem mass spectrometry.* Analytical Chemistry, vol. 74(18):4741–4749, 2002.

[227] Chelius, D. *et al. Quantitative profiling of proteins in complex mixtures using liquid chromatography and mass spectrometry.* Journal of Proteome Research, vol. 1(4):317–323, 2002.

[228] Wang, W. *et al. Quantification of proteins and metabolites by mass spectrometry without isotopic labeling or spiked standards.* Analytical Chemistry, vol. 75(18):4818–4826, 2003.

[229] Benk, A.S. *et al. Label-free quantification using MALDI mass spectrometry: Considerations and perspectives.* Analytical and Bioanalytical Chemistry, vol. 404(4):1039–1056, 2012.

[230] Fugmann, T. *et al. DeepQuanTR: MALDI-MS-based label-free quantification of proteins in complex biological samples.* Proteomics, vol. 10(14):2631–2643, 2010.

[231] Eidhammer, I. *et al. Computational and statistical methods for protein quantification by mass spectrometry.* Wiley Online Library, 2013.

[232] Sakoe, H. *et al. Dynamic programming algorithm optimization for spoken word recognition.* IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 26(1):43–49, 1978.

[233] Old, W.M. *et al. Comparison of label-free methods for quantifying human proteins by shotgun proteomics.* Molecular & cellular proteomics : MCP, vol. 4(10):1487–1502, 2005.

[234] Nesvizhskii, A.I. *et al. Interpretation of shotgun proteomic data: the protein inference problem.* Molecular & cellular proteomics : MCP, vol. 4:1419–1440, 2005.

[235] Alves, P. *et al. Advancement in protein inference from shotgun proteomics using peptide detectability.* Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, vol. 420:409–420, 2007.

[236] Qeli, E. *et al. PeptideClassifier for protein inference and targeted quantitative proteomics.* Nature biotechnology, vol. 28(7):647–650, 2010.

[237] Li, Y.Y.F. *et al. A Bayesian Approach to Protein Inference Problem.* Journal of Computational Biology, vol. 16(8):1183–1193, 2009.

[238] Huang, T. *et al. Protein inference: A review.* Briefings in Bioinformatics, vol. 13(5):586–614, 2012.

[239] Davis, L. *et al. Handbook of genetic algorithms*, vol. 115. Van Nostrand Reinhold New York, 1991.

[240] Mitchell, M. *An introduction to genetic algorithms.* MIT press, 1998.

[241] Michalewicz, Z. *Genetic algorithms+ data structures= evolution programs.* springer, 1996.

[242] Dubash, M. *Moore's law is dead, says gordon moore.* http://news.techworld.com/operating-systems/3576581/moores-law-is-dead-says-gordon-moore/. Accessed: 2014-10-20.

[243] Vizcaíno, J. *et al. ProteomeXchange provides globally coordinated proteomics data submission and dissemination.* Nature . . . , vol. 32(3):223–226, 2014.

[244] Wütherich, G. *et al. Die OSGi Service Platform.* Dpunkt. verlag GmbH, 2008.

[245] McAffer, J. *et al. Eclipse rich client platform.* Addison-Wesley Professional, 2010.

[246] Johnson, R. *Expert one-on-one J2EE design and development.* John Wiley & Sons, 2004.

[247] Wenig, P. *et al. OpenChrom: a cross-platform open source software for the mass spectrometric analysis of chromatographic data.* BMC bioinformatics, vol. 11:405, 2010.

[248] Zhang, Y. *et al. Protein analysis by shotgun/bottom-up proteomics.* Chemical Reviews, vol. 113:2343–2394, 2013.

[249] Kuster, B. *et al. Scoring proteomes with proteotypic peptide probes.* Nature reviews. Molecular cell biology, vol. 6(7):577–583, 2005.

[250] Anderson, N.L. *et al. A human proteome detection and quantitation project.* Molecular & cellular proteomics : MCP, vol. 8:883–886, 2009.

[251] Apweiler, R. *et al. Activities at the Universal Protein Resource (UniProt).* Nucleic Acids Research, vol. 42(November 2013):191–198, 2014.

[252] Gamma, E. *et al. Design patterns: elements of reusable object-oriented software.* Pearson Education, 1994.

[253] Blu, T. *et al. Linear interpolation revitalized.* IEEE Transactions on Image Processing, vol. 13:710–719, 2004.

[254] Malvar, H. *et al. High-quality linear interpolation for demosaicing of Bayer-patterned color images.* 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, 2004.

[255] Brudnyi, Y.A. *et al. Interpolation of linear operators.* Journal of Soviet Mathematics, vol. 42:2009–2113, 1988.

[256] Thévenaz, P. *et al. Image interpolation and resampling.* In *Handbook of Medical Image Processing and Analysis*, pp. 465–493. Academic Press Inc., 2nd edition, 2009.

[257] Li, X. *et al. New edge-directed interpolation.* IEEE Transactions on Image Processing, vol. 10:1521–1527, 2001.

[258] Hsu, B.J. *Generalized linear interpolation of language models*, 2007.

[259] Mitra, V. *et al. Inversion of peak elution order prevents uniform time alignment of complex liquid-chromatography coupled to mass spectrometry datasets.* Journal of Chromatography A, vol. 1373:61–72, 2014.

[260] Spicer, V. *et al. Sequence-specific retention calculator. A family of peptide retention time prediction algorithms in reversed-phase HPLC: Applicability to various chromatographic conditions and columns.* Analytical Chemistry, vol. 79(22):8762–8768, 2007.

[261] Klammer, A.a. *et al. Improving tandem mass spectrum identification using peptide retention time prediction across diverse chromatography conditions.* Analytical Chemistry, vol. 79(16):6111–6118, 2007.

[262] Palmblad, M. *et al. Protein identification by liquid chromatography-mass spectrometry using retention time prediction.* Journal of Chromatography B: Analytical Technologies in the Biomedical and Life Sciences, vol. 803:131–135, 2004.

[263] Krokhin, O.V. *et al. Use of peptide retention time prediction for protein identification by off-line reversed-phase HPLC-MALDI MS/MS.* Analytical Chemistry, vol. 78(17):6265–6269, 2006.

[264] Krokhin, O.V. *et al. An improved model for prediction of retention times of tryptic peptides in ion pair reversed-phase HPLC: its application to protein peptide mapping by off-line HPLC-MALDI MS.* Molecular & cellular proteomics : MCP, vol. 3:908–919, 2004.

[265] Petritis, K. *et al. Use of artificial neural networks for the accurate prediction of peptide liquid chromatography elution times in proteome analyses.* Analytical Chemistry, vol. 75(5):1039–1048, 2003.

[266] Jaitly, N. *et al. Decon2LS: An open-source software package for automated processing and visualization of high resolution mass spectrometry data.* BMC bioinformatics, vol. 10:87, 2009.

[267] Deutsch, E. *mzML: A single, unifying data format for mass spectrometer output.* Proteomics, vol. 8:2776–2777, 2008.

[268] Côté, R.G. *et al. JmzML, an open-source Java API for mzML, the PSI standard for MS data.* Proteomics, vol. 10(7):1332–1335, 2010.

[269] *Dynamic range benchmark dataset.* http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PXD000279. Accessed: 2015-01-25.

[270] Motulsky, H. *Intuitive biostatistics: a nonmathematical guide to statistical thinking.* Oxford University Press, 2013.

[271] Hanke, S. *unpublished.*

[272] Elia, G. *Biotinylation reagents for the study of cell surface proteins.* Proteomics, vol. 8:4012–4024, 2008.

[273] Brandli, a.W. *et al. Transcytosis in MDCK cells: Identification of glycoproteins transported bidirectionally between both plasma membrane domains.* Journal of Cell Biology, vol. 111(6):2909–2921, 1990.

[274] Zhang, W. *et al. Affinity enrichment of plasma membrane for proteomics analysis.* Electrophoresis, vol. 24:2855–2863, 2003.

[275] Zhao, Y. *et al. Proteomic Analysis of Integral Plasma Membrane Proteins.* Analytical Chemistry, vol. 76:1817–1823, 2004.

[276] Scheurer, S.B. *et al. Identification and relative quantification of membrane proteins by surface biotinylation and two-dimensional peptide mapping.* Proteomics, vol. 5(11):2718–2728, 2005.

[277] Strassberger, V. *et al. Chemical proteomic and bioinformatic strategies for the identification and quantification of vascular antigens in cancer.* Journal of Proteomics, vol. 73(10):1954–1973, 2010.

[278] Benk, A. *Proteomics-based discovery of novel vascular accessible markers in kidney cancer.* Dissertation, Ruperto-Carola University of Heidelberg, 2014.

[279] Nadler, W. *unpublished.*

[280] Collisson, E.a. *et al. Subtypes of pancreatic ductal adenocarcinoma and their differing responses to therapy.* Nature medicine, vol. 17(4):500–503, 2011.