Development and Application of Flexible Algorithms for the Protein Inference Problem in Bottom-up Mass Spectrometry

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

Dipl.-Informatiker (Bioinformatik) Julian Uszkoreit

aus Hamburg

Tübingen
2016

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	7. März 2017
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Oliver Kohlbacher
2. Berichterstatter:	PD Dr. Christian Stephan

Abstract

Liquid chromatography coupled to mass spectrometry (LC-MS) has become the most commonly used method for proteomics in recent years. This is mainly due to its relative affordability in comparison to gel-based methods combined with its fast and straight forward usage. The possibility to identify thousands of proteins by tandem mass spectrometry (MS/MS) in a few hours let LC-MS/MS become a widely used high-throughput method in the life sciences. The current state-of-the-art mass spectrometers though makes it necessary to digest proteins into peptides: too large and too highly charged molecules cannot be measured with sufficiently high resolution in high throughput. Peptides on the other hand can be detected and identified, most often employing database search engines. This bottom-up approach comes with the drawback that identified peptides have to be re-assembled to protein sequences. This step is called protein inference and is not trivial: due to peptide ambiguities a correct solution of the inference cannot be given in most cases. A peptide that was found in the original database used for the identification, can sometimes be assigned to more than one protein. The reason for this can have multiple causes, for example homologous proteins and protein domains, isoforms or simply redundant sequences originating from multiple entries for the same protein or sequence fragment. These shared peptides lead to a set of proteins, which are built-up of the same sets or sub-sets of sequences. This problem is known as the "protein ambiguity" and without further assumptions or additional knowledge it remains uncertain, which protein of such a set was actually present in the measured sample, unless a unique peptide, which belongs to only one protein, was detected.

The work presented herein addresses open problems in protein inference. At first, the problem and its causes are addressed in detail. Additionally some of the basic algorithms for peptide identification as well as possibilities to merge their results are introduced. During this work the tool "PIA - Protein Inference Algorithms" was developed. PIA was compared to four other inference methods in an in-depth assessment. In this analysis the differences, but also the similarities of these tools and their reports were highlighted. During the development of PIA special care was taken, that no single proteins, but protein groups are reported. Furthermore, it allows a user to chose from multiple algorithms for the inference and set a multitude of different filters, as well as to merge the results of multiple search engines. The community standard

file formats mzIdentML and mzTab can be used for data import and export, thus giving the opportunity to easily include PIA into bigger proteomics pipelines. PIA can be executed in the workflow environment KNIME or directly via the command line. Additionally, it provides a user friendly web interface, which can be accessed with any current browser. Besides comprehensive and easily browsable result lists, PIA offers an intuitive visualisation of the relations between the MS spectra, peptides and proteins, which are contained in a generated protein result list.

Zusammenfassung

Flüssigkeitschromatographie gekoppelt mit Massenspektrometrie (LC-MS) ist in den letzten Jahren zu der am meisten verbreiteten Methode der Proteomik geworden. Dies ist besonders auf die relative Kostengünstigkeit gegenüber gelbasierten Methoden, sowie der schnellen und einfachen Handhabung zurückzuführen. Die Möglichkeit, tausende Proteine innerhalb weniger Stunden mittels Tandemmassenspektrometrie (MS/MS) zu identifizieren, macht die LC-MS/MS zu einer weit verbreiteten Hochdurchsatzmethode in den Lebenswissenschaften. Der technische Stand der Massenspektrometer macht es jedoch nötig, dass Proteine zu Peptiden verdaut werden, da zu große Moleküle und solche mit zu hohem Ladungszustand nicht mit genügend hoher Auflösung im Hochdurchsatzverfahren vermessen werden können. Die Peptide hingegen können detektiert und meist mittels Datenbank Suchmaschinen identifiziert werden. Dieser bottom-up Ansatz hat jedoch den Nachteil, dass die identifizierten Peptide wieder zu Proteinsequenzen zusammengesetzt werden müssen. Dieser Schritt wird als Proteininferenz bezeichnet und ist nicht trivial: aufgrund von Peptidambiguitäten kann es oftmals keine genaue Lösung der Inferenz geben. Es kann vorkommen, dass ein Peptid in der zugrundeliegenden Datenbank, welche zur Identifikation benutzt wurde, mehreren Proteinen zugewiesen wird. Dies kann mehrere Gründe haben, beispielsweise homologe Proteine und Proteindomänen, Isoformen oder einfach redundante Sequenzen (mehrere Einträge für dasselbe Protein oder Sequenzfragmente). Diese gemeinschaftlichen Peptide führen zu einer Menge von Proteinen, welche aus denselben Mengen oder Untermengen von Sequenzen aufgebaut sind. Dieses Problem ist bekannt als die „Proteinambiguität“(„protein ambiguity“) und ohne weitere Annahmen oder zusätzliches Wissen kann nicht klar entschieden werden, welches Protein einer solchen Proteinmenge in der gemessenen Probe vorhanden war. Es sei denn, ein Peptid, welches nur einem Protein zugewiesen werden kann, wurde ebenfalls detektiert.

Die vorliegende Arbeit befasst sich mit dem Problem der Proteininferenz. Zunächst wird das Problem und seine Ursachen genau vorgestellt. Außerdem wird auf einige der Grundlegenden Algorithmen zur Peptididentifikation sowie Möglichkeiten um deren Ergebnisse zu vereinheitlichen eingegangen. Im Laufe dieser Arbeit wurde das Tool „PIA - Protein Inference Algorithms“ entwickelt. Dieses wird alleine und zusammen mit vier weiteren Proteininferenzmethoden in einer ausführlichen Begutachtung analysiert. In dieser Untersuchung werden

die Unterschiede, aber auch Gemeinsamkeiten, der Tools und deren Ergebnislisten herausgearbeitet. PIA ist speziell darauf ausgelegt, dass es keine einzelnen Proteine, sondern immer Proteingruppen als Ergebnisse liefert. Außerdem gibt es dem Benutzer die Entscheidung aus mehreren Algorithmen für die Inferenz zu wählen und eine Vielzahl an Filtern zu setzen, sowie die Ergebnisse mehrerer Suchmaschinen zu vereinen. Es beherrscht sowohl für den Import als auch den Export die Community-Standarddateiformate mzIdentML und mzTab und bietet dadurch einen einfachen Einbau in größere Proteomik-Pipelines. PIA kann sowohl in der Workflowumgebung KNIME als auch über die Kommandozeile ausgeführt werden. Zusätzlich bietet es ein benutzerfreundliches Web-Frontend, welches über jeden aktuellen Browser aufgerufen werden kann. Neben ausführlichen und leicht inspizierbaren Ergebnislisten bietet PIA auch eine intuitive Visualisierung der Verhältnisse zwischen MS-Spektren, Peptiden und Proteinen, welche zu der Erstellung einer Ergebnislisten geführt haben.

Acknowledgements

I am very grateful to Christian Stephan and later on Martin Eisenacher for introducing me into the field of proteomics and being great work group leaders. Many thanks also to my colleagues in the bioinformatics unit of the MPC, especially Maike, Micha and Micha. Even if we always complain about everything, our job cannot be that bad. With this, I also thank Prof. Katrin Marcus and the complete Medizinisches Proteom-Center for the nice working climate. Without all your discussions, testings and support, PIA would never have existed.

Furthermore, my deepest thanks to Oliver Kohlbacher, who as my supervisor had always good ideas to promote my work. I also want to thank all my collaborators, who published articles and prepared workshops with me.

Special thanks go to Dominik, Sebastian and Jo for reading through all this, even though each had only the knowledge of a very small portion of it! And also for everything else!

I like to thank my scout group "Salomon Idler Univiertel" and all my friends around it for their support, trust and motivation through all these years. Be prepared and remember: *A Scout smiles and whistles under all circumstances.*

Last but not least: thanks to my parents, siblings and family! You made me very early in my life aware of my affection to natural sciences and computers and – most important – let both prosper. Finally I thank Elke for everything and dedicate the thesis to her, who is not yet born!

General Remarks

- In accordance with the standard scientific protocol, I will use the personal pronoun ***we*** to indicate the reader and the writer, or my scientific collaborators and myself.
- Some contents of this theses were published before in scientific journals, partly literally.

  Parts of Chapter 5 (mainly Section 5.2) and parts of Chapter 6 were published before in the article: "PIA: An Intuitive Protein Inference Engine with a Web-Based User Interface.", Uszkoreit et al. *J Proteome Res*. 2015 Jul 2;14(7):2988-97.[1]

  Other parts of Chapter 5 (mainly Section 5.3) were published before in the article: "In-depth analysis of protein inference algorithms using multiple search engines and well-defined metrics.", Audain and Uszkoreit et al. *J Proteomics*. 2016 Aug 4;150:170-182.[2]

Contents

Chapter 1

Introduction

1.1 Motivation

Since the discovery of the deoxyribonucleic acid (DNA) in the late 19th century and the discovery of its potential to encode the genetic information of all known living organisms, it shaped the history of biology during the 20th century. A climax was reached when Watson and Crick described the double-helix structure of DNA[3] and roughly 50 years later when the Human Genome Project[4] published the final sequencing results of the human DNA in 2004, after more than 13 years of work. Only about four years later, the so-called Next Generation Sequencing (NGS)[5] technologies were introduced. These highly parallelized methods were much more advanced than the traditional Sanger sequencing and it allows nowadays the sequencing of a single person's genome in almost a day's time[6]. All this led to the start of the "1000 Genomes Project" in 2008, aiming at the complete sequencing of at least thousand humans from different ethnic backgrounds and finding variations in the genetic code. These efforts resulted in a much more comprehensive overview of the genetic variation inside one species alone[7,8].

All these studies of the genome brought many valuable new insights, but could by far not solve all biological problems. Additionally, several paradigms, which were considered to be true, had to be re-inspected. For example it was no longer possible to assume a conclusion of an organism's complexity from the number of protein-coding genes. The publication of the human genome revealed that man has about 20,000 – 25,000 protein coding genes[9], more recent estimations place the numbers at just 19,000 genes[10], which is even less than the genome of the nematode *C. elegans* (about 20,000)[11]. On the other hand, the genome alone cannot explain the complete phenotype. One of the most obvious example for this is the metamorphosis of a caterpillar into a butterfly. Both have exactly the same gene sequence, but their physiology could not be more different. Furthermore, it is known today that one gene codes for more than one protein, for example through alternative splicing[12]. Apart from the pure genome sequence,

it seems more important nowadays to know which genes are activated and thus translated into proteins or influence the organism in any other way, like being coding templates for miRNAs.

To address challenges in these new directions, several new disciplines emerged within the life sciences. Most of them are so called "omics", which address the collective characterisation and quantification of all biological molecules of a respective "ome". An "ome", on the other hand, describes the complete set of a specific biological molecule in a species or sample. Thus, for example the genome describes the complete set of a species' genes and "genomics" is the scientific field addressing it. Besides "transcriptomics" and "metabolomics", the field of "proteomics" was developed. The name proteome was coined 1994 by Marc Wilkins on a conference and one year later in a publication[13].

1.2 LC-MS/MS Proteomics

Mass spectrometry allows the characterisation of molecules by their mass to charge ratio. In proteomics this is used to either detect and identify whole proteins (top-down) or, more often, peptides of digested proteins (bottom-up) in a sample. To measure a biological component with a mass spectrometer, it has to be ionised first which is for example done by electrospray ionisation. The knowledge of the mass however does not suffice to fully identify a component, as there may be several entities in a searched database with nearly the same mass[14]. For this reason, the ions are fragmented and the fragment spectra (MS/MS or MS2) together with their parent ion's mass are used for identification. This is feasible, as the parent ions fragment preferentially at the peptide bond. Thus, the MS/MS spectra can either be used for identification by so-called database search engines like SEQUEST[15], Mascot[16] or X!Tandem[17], which match measured spectra against theoretical spectra calculated from protein sequence database entries, or for *de novo* peptide or protein identifications[18,19].

The current state-of-the-art mass spectrometers still cannot nearly create complete MS/MS spectra of complete proteins in high-throughput, i.e., spectra which contain ions to explain the complete amino acid sequence. As this though is feasible for peptides, the bottom-up or also called shotgun technique[20] has become the method of choice for high-throughput protein identification in recent years. There the proteins of a sample of interest are enzymatically digested to peptides. For the digestion mainly enzymes which cut at specific positions within a protein and thus produce predictable peptides are used, to minimise the search space used by database search engines. The complex peptide mixture is often separated by liquid chromatography (LC)[21] prior to measurement. This approach is referred to as "liquid chromatography coupled to tandem mass spectrometry", or short LC-MS/MS. It is performed, because the mass spectrometer can detect and fragment only a limited number of ions per time. Also, with this additional temporal separation due to physico-chemical features the signal inference for the measured molecules is decreased.

1.3 The Protein Inference Problem

One drawback of bottom-up proteomics is, that the database search engines as well as *de novo* approaches identify only peptides through the calculation of peptide spectrum matches (PSMs). As researchers are more often interested in the actual proteins rather than the peptides, which are created only to obtain detectable molecules for the mass spectrometers, it is necessary to generate protein lists, which contain database accessions, from the identified PSMs. The step from PSMs to proteins is called "protein inference"[22]. This step is not trivial, because a significant number of tryptic peptides in a database search is not unique for one protein entry but shared by multiple entries. This holds especially true for higher organisms, due to homologous proteins and protein domains or isoforms contained in databases. These shared (sometimes also called "degenerated") peptides lead to sets of proteins, which are built up of the same set or subset of peptides. This problem is known as the *protein ambiguity* and without further assumptions or knowledge it cannot be decided which of the proteins of a set are in the sample, unless a unique peptide was found. Often for each such protein ambiguity group only a representative accession number is reported in the result list and the other proteins are reported as "similar proteins" or "group members".

For a more complete result list, all these potential proteins (according to the inference algorithm) should be reported, as was already suggested by Nesvizhskii et al.[22] in 2005. The set of PSMs selected for the protein inference, the logic and algorithm of the inference and the selection of reported representatives vary significantly between inference algorithms. For some methods - usually the commercial ones, but also some freely available - the details of their algorithms are scarcely documented, so that results cannot be explained or it cannot be judged whether they are reasonable. Though additionally to the search engines' inherent inference algorithms there are quite a lot of stand-alone programs for protein inference from PSMs (e.g., ProteinProphet[23], Scaffold[24] and IDPicker[25]), some of them support only specific search engines and most are limited in their settings for inference parameters.

Merging the results from multiple search engines is also desirable to either increase the number of identified spectra passing an FDR threshold and thus the number of corresponding proteins, or to amplify the evidence of peptides detected in the analysed sample[26]. This poses a major problem, because each search engine's algorithm generates its own value for the quality of a PSM, generally a score or probability value (throughout this thesis and in this context score always means the score or probability, if not further specified). These scores are usually not directly comparable but need the calculation of another comparable score[27–29].

1.4 Contributions of this Thesis

This work addresses and discusses some challenges caused by protein ambiguity. Furthermore it introduces a tool suite, which allows to import data from virtually all available search engines, due to support of current standard formats, lets a user inspect and assess the identified PSMs and peptides, gives the user full control of the protein inference, and visualises the dependencies between PSMs, peptides and proteins.

In Chapter 2 the experimental background for a proteomics experiment is given. It is described, how the biological sample is processed and the mass spectrometrical data is created. Though this gives the basis of all proteomics studies, only a short overview of the most important aspects for the rest of this thesis can be provided. This includes a general overview of liquid chromatography, followed by two of the mainly used ionisation methods. The basic concepts of mass spectrometers are explained, as well as some of the current experimental issues.

The further processing after the data collection and the computational background is described in Chapter 3. We outline how the generated spectra are automatically interpreted by *de novo*, spectral library and database search engines. Like most high-throughput methods, also the MS proteomics suffers from the identification of false positives. Therefore, strategies to maintain a good quality of results are given. To highlight the theoretical and practical dimensions of the peptide and protein ambiguity problem some of the most commonly used protein databases are *in silico* digested and analysed. This shows that most peptides in manually curated databases are unique for one protein, though in databases containing isoforms, the fraction of shared peptides is much higher. These numbers are further compared with results of two MS/MS datasets, one containing mouse and the other containing human sample data. This analysis confirms, that in real life experiments more shared peptides are identified than the *in silico* digestion would suggest.

Chapter 5 is separated into two different assessments of protein inference algorithms. In the first part, the performance of PIA alone is analysed. The implementation of PIA is tested with different settings on three datasets, a real-life mouse dataset created at the Medizinisches Proteom-Center (MPC) and two ground truth datasets, one containing yeast and the other also mouse samples. For the ground truth datasets, the proteins contained in the samples are claimed to be known. The analysis of PIA's performance on the real-life dataset showed, that it performs well on reporting protein identifications and a merge of peptide identifications from multiple search engines can boost the results from single search engines, also on protein level. Using the yeast ground truth dataset it is possible to show, that the actually reported proteins are also contained in the sample, according to the provided reference set of accessions. Finally, an assessment of the results on the second ground truth dataset could verify, that PIA does also perform well on identifying the expected number of protein isoforms in a sample.

The second part of Chapter 5 compares the results of PIA to four more methods for protein inference. Therein, a workflow is created which allows unbiased interpretation of the inference results based on several well-defined metrics. This in-depth assessment gives no final conclusion on which method performs better, but highlights several considerations for choosing an appropriate inference method. For example, the complexity of the database used for peptide identification has great influence on some of the methods. Furthermore, some methods also report protein sub-groups, which might not be desired by the user and must be taken into account when selecting the tool used for a study. On the assessed metrics, PIA outperforms the other methods slightly, which highlights, that it creates high quality protein reports.

Furthermore, PIA comprises currently to the best of my knowledge the most comprehensive set of inference methods and respective settings and filters. Together with an intuitive visualisation of the complex relations between PSMs, peptides and proteins in an MS-based proteomics analysis and the report of these in easily browsable interfaces, it allows in-depth analysis of the data as well as the reliable creation of protein lists. As PIA is relatively robust when using large datasets and protein databases, it facilitates the analysis of common single species analyses as well as metaproteomics datasets.

The actual concepts and goals of the tool suite "PIA - Protein Inference Algorithms" is highlighted in Chapter 6. This chapter describes the main principles as well as the implementation of the methods. Furthermore, all implemented algorithms are explained and the different ways to execute a PIA analysis are highlighted. Here it is important to notice, it cannot only be called by the command line and thus integrated into any pipeline, but there are also more user-friendly methods like the integration into the KNIME workflow environment as well as an intuitive web frontend.

In the last chapter, the conducted work is concluded and an outlook for further studies based on the work of this thesis is given.

Chapter 2

Experimental Background

In the last decades, many different mass spectrometers and protocols for sample preparation and analysis were established for use in proteomics. This chapter gives an overview of some widely used methods and instrument types, ordered by their chronological appearance in a general mass spectrometry based proteomics workflow. As each of these steps is a large scientific field in itself and beyond scope of this work, only the basic principles will be discussed herein. Likewise, only the relevant steps for bottom-up mass spectrometrical proteomics experiments are covered, which are necessary for the further understanding of the work in this thesis. The expression "bottom-up" or synonymously also "shotgun" proteomics derives from the identically termed genomic sequencing counterparts. Here, the whole genome, respectively chromosomes, are broken down into smaller fragments which could easily be sequenced at once. Similarly, in bottom-up proteomics, the proteins are cut or, more accurately, digested into peptides of smaller size. The reason behind this and the fact, that at the time of writing mainly bottom-up MS proteomics is performed instead of top-down, is due to the inability of currently used mass spectrometers to measure whole proteins, at least in complex samples and high-throughput. At present, the lengths of a peptide should be from 5-45 amino acids to be measurable on most machines.

2.1 Sample Preparation

At some time before starting an MS analysis, a sample must be prepared. There are internal standards which may consist of some well defined peptides or proteins, which are processed with buffers etc. and directly given to the LC-MS/MS. These kind of samples are mainly run for reasons of quality control, though. Most of the real-life proteomics experiments consist of either samples collected from patients, test animals or plants (e.g., tissues, body fluids, post-mortem samples) or harvested from cell cultures (e.g., from samples which were treated and untreated by a drug or genetically modified in any way). The exact starting point of the sample

preparation varies depending on the kind of sample and optionally the applied quantification method. For experiments which use labels to differentiate between states of treatments, like the "Stable Isotope Labeling by Amino Acids in Cell Culture" (SILAC,[30]), the sample preparation already starts with growing the cells: they are incorporating isotopic labelled amino acids from the growth medium into their proteins. For other samples, for example label free proteomics of patient data, the sample preparation starts after collecting the samples. In Figure 2.1 an overview of a protein identification workflow is shown, which shows the collection and any pre-processing as the first two steps.

After harvesting respectively collecting the samples, the cells are usually broken up and the proteins are extracted. Either only some sub-cellular compartments, e.g. mitochondria or nuclei, or the whole contents of the cell are used for further analysis steps. Sometimes also a intra-cellular separation is performed, either on the organelles or e.g. by cleaving outer membrane proteins from the cell. Other protocols are using body fluids, extracellular matrices or a secretion for further analysis instead of the cellular contents.

For a bottom-up proteomics experiment, the samples' proteins need to be digested into peptides. The actual digestion into peptides can either be performed in-gel[31] or in-solution[32], both alternatives having disadvantages and advantages: While in-gel digestion is more robust against impurities, which are interfering with the digestion, the peptide extraction cannot be automated as easily. In-solution methods on the other hand are easily automated, but the proteome may be incomplete solubilized and the digestion impeded by contaminating substances[33]. Trypsin is the mainly used enzyme for the digestion, because it tends to produce peptides of a suitable length for mass spectrometry analysis. Furthermore, due to biotechnological improvements, the nowadays commercially available trypsin is well suited for laboratory usage (overnight digestions and room temperature) and the cleavage sites are very strict and thus predictable[34]: it cleaves after each lysine (Lys, K) and arginine (Arg, R), if they are not followed by a proline (Pro, P) or, which happens less often, masked by a modification. Other widely used enzymes are for example chymotrypsin and Lys-C, both also with well predictable cleavage sites but leading to less suitable average peptide lengths and consequently laboratory protocols. Pepsin is used less frequently, since the cleavage sites are hard to predict and thus the data analyses typically turns out to be more error prone and cumbersome.

If the sample is too complex and the LC separation (explained in Section 2.2) not sufficient, another fractionation procedure like isoelectric focusing (IEF) or "Polyacrylamide gel electrophoresis" (PAGE) can be performed. After the separation of the proteins or peptides, a resulting lane can be sliced into several bands, which are processed and measured separately, in successive MS runs. This pre-fractionation furthermore increases the separation capacity and leads to possibly more and better identifications (and also quantifications) of peptides and thus proteins.

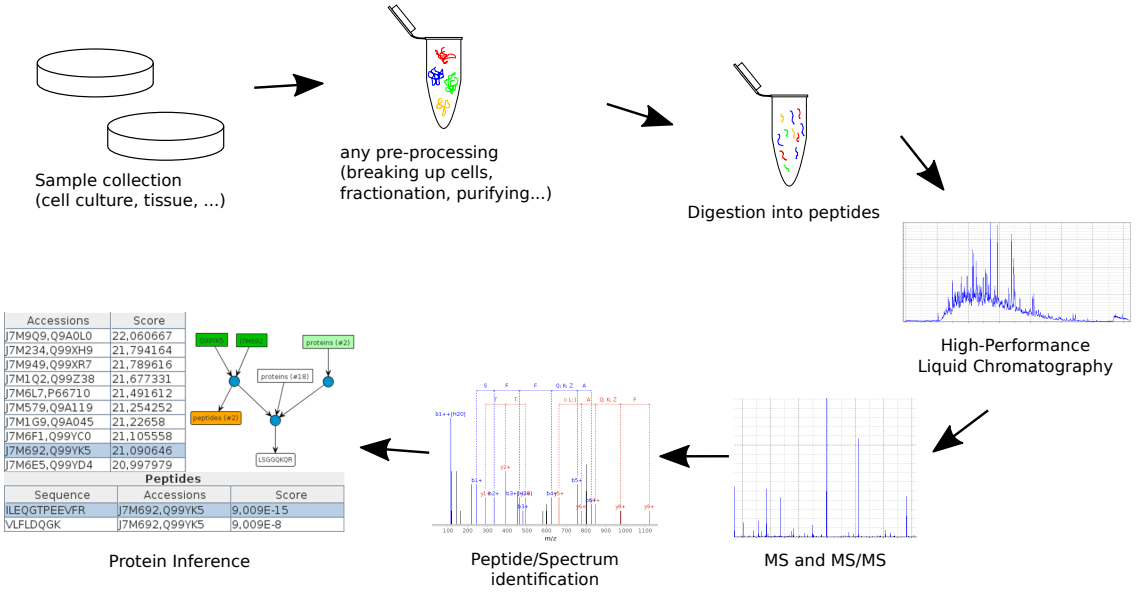


Figure 2.1: Simplified overview of a bottom-up mass spectrometry protein identification workflow. Usually such a workflow starts with the collection of samples and any required pre-processing steps. To perform a high-throughput protein identification with current technology, the proteins are digested into peptides and given onto a HPLC, which is coupled to a mass spectrometer. For the peptide identification, MS/MS spectra are identified either by database search engines or *de novo*. Based on these peptides the protein inference reports a list of proteins, which have evidence to be in the sample.

2.2 Liquid Chromatography

A mass spectrometer can only measure a limited number of compounds per time. Therefore, complex samples like full cell lysates, which contain several thousand proteins or peptides, are commonly separated by liquid chromatography (LC) before injection into the mass spectrometer[20]. The mainly used LC methods are commonly abbreviated HPLC, which stands for high-performance LC, but may also stand for high-pressure LC, due to the high pressure flow through the columns.

The basic principle of the LC used for MS proteomics is the adsorption of proteins or peptides dissolved in an appropriate solvent (mobile phase) by the packing material (stationary phase) of a column[21]. Though various other techniques like ion-exchange or affinity chromatography are also used, the currently most widely used method is the reversed-phase chromatography (RPC or RP-HPLC). In contrast to a hydrophilic stationary phase in the normal phase, a hydrophobic stationary phase (column) is used for RPC. After loading the column with analytes of a sample, a gradient of solvent mixtures with different polarities in order to increase the separation is used as the mobile phase. The RPC and the gradient cause the less hydrophobic particles to elute before the more hydrophobic analytes. Thus, the complexity of the sample is spread over the time of the gradient and ideally each analyte elutes at a well defined retention time only.

As a rule of thumb, the slower the gradient and the longer the used column, the better is the separation. For MS-based proteomics, gradients between one and three hours and column lengths between 25–50 cm are commonly used at the time of writing.

To further enhance the performance of the separation process, different LC techniques can be coupled. For example, in order to analyse phosphorylated proteins respectively peptides, electrostatic repulsion-hydrophilic interaction chromatography (ERLIC,[35]) is frequently used to enrich phosphopeptides before using a default LC-MS approach for identification and quantification.

2.3 Ionisation Methods

Before it is possible to measure a biological molecule like a peptide or protein with a mass spectrometer, these molecules have to be ionised. For the ionisation mainly two methods are used in proteomics: "Matrix Assisted Laser Desorption/Ionisation" (MALDI) and "Electro-Spray Ionisation" (ESI).

2.3.1 Matrix-Assisted Laser Desorption/Ionisation

When using matrix-assisted laser desorption/ionisation (MALDI[36,37]) to ionise analytes, the samples have to be spotted onto a target plate and mixed with a matrix. These treated samples are pulsed by a laser in such a way, that mainly singly charged ions of the sample are generated and measured by the MS. To run a complex sample with MALDI, it is common to perform a prior separation of the sample via two-dimensional differential gel electrophoresis (2D-DIGE,[38]), followed by the MALDI measurement of single spots on the gel. These contain in the ideal case only one peptide respectively protein species per spot. Usually only the most interesting (e.g. differently expressed) spots of a DIGE experiment are spotted on a MALDI plate and measured in this way.

Though it is possible to further fragment the ions created by MALDI before identification (see Section 2.4.2), more often peptide mass fingerprints are identified using database search engines[39]. For this, the peaks of a measured spectrum are matched against the peptides of proteins in a database, assuming that each peak originates from one peptide and that multiple peptides of the same protein can be found in one spectrum. The need to separate complex samples with a time-consuming and expensive method like DIGE limits the number of identified compounds, but MALDI allows also to make spatial profiles of tissue samples. For this technique, called MALDI imaging, a whole tissue slice is fixed an a mounting plate, if desired treated with an enzyme to digest the proteins to peptides, and coated with matrix. Afterwards, the tissue will be rastered and pulsed by a laser to measure an MS spectrum for each raster point. With

this method, identifications are hardly possible, but the samples can be classified spatially by hierarchical clustering of the spectra or other machine learning approaches.

MALDI is not a high-throughput method for protein identification, at least not any more compared with advances in other technologies. Its application for proteomics though was acknowledged with the Nobel prize for chemistry in 2002[40].

2.3.2 Electrospray Ionisation

Electrospray ionisation (ESI) is, at the time of writing, the most widely used method when performing MS-based proteomics experiment. When using this technique, the capillary of an LC is coupled to an ESI source. In this source, a high voltage electrostatic field is generated between the outlet of the solvent (usually referred to as the needle) and the cathode, which leads through an orifice into the mass spectrometer. The applied flow of the LC lets the solvent emerge from the needle's tip in a stream of droplets. Simultaneously, the solvent's surface gets charged by the applied electric field[21]. These charged droplets are attracted by the electrode and shrink in size while travelling through the field, due to the evaporation of the solvent. Therefore, the droplets get higher charged respectively to their size, until the Rayleigh limit[41] is reached: the droplets explode, finally creating a stream of ionised single molecules, in the case of proteomics protein ions, respectively peptide ions, which are transferred into the mass spectrometer for further analysis, as outlined in Figure 2.2.

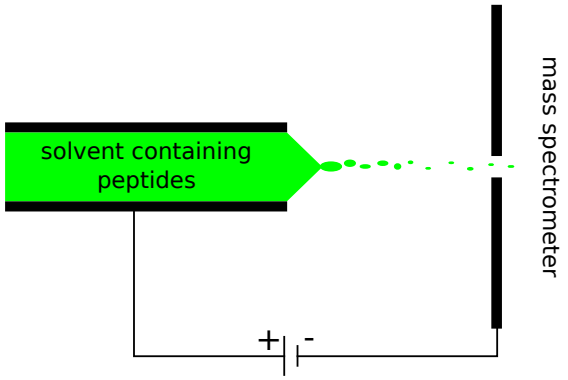


Figure 2.2: Schematic overview of an electrospray ionisation (ESI) ion source. Under high voltage, charged droplets containing the solved analytes fly through an electrostatic field. While travelling to the cathode, the droplets shrink due to evaporation of the solvent and become higher charged respectively to their size. When the charge-to-volume ratio reaches a certain limit, the droplets explode and finally ionised single molecules are generated which are transferred into the mass spectrometer.

2.4 Mass Spectrometers

In the mass spectrometer, the signal intensity of a certain mass-to-charge (m/z) ratio associated with a specific molecule is recorded. Currently, there are several types of mass spectrometers in use, which originate from different methods to differentiate between ions. In time-of-flight (TOF) machines, the time needed by an ion to travel through an electric field of specific strength and length is used to derive its m/z. A quadrupole mass analyser (mostly abbreviated as quadrupole) consists of four parallel metal rods. If specific alternating voltage fields are applied to these rods, only ions of a specific m/z ratio can pass the quadrupole, while other ions have an unstable trajectory and are thus filtered out[42]. The quadrupole is used in combination or as a filter with other types of MS. If used as sole technique, most often a triple-quadrupole (QQQ) is used. A quadrupole can also act as an ion trap. There are different kinds of ion traps, but all are used to hold ions (of a specified m/z) until a sufficient amount of them coming from the ion source is stored to be detected[43]. After the ions are separated by their m/z ratios, their intensity is subsequently measured by a detector.

Another, currently very popular kind of ion trap is the Orbitrap, which is often coupled to a linear trap quatrupole (LTQ) as marketed by Thermo Fisher Scientific. An Orbitrap consists of a cylindrical outer and an axial inner electrode[44]. Injected ions orbit around the inner electrode and ions of same m/z ratios are packed into bands. These perform harmonic oscillations depending on their m/z ratios. The ions are detected all at once by their induced current on the outer electrode and a Fourier transformation is used to extract a mass spectrum.

Considerable advancements have been made concerning the m/z and temporal resolutions of mass spectrometers, which now can create some tens of spectra per second and Orbitraps can differentiate between ions with a resolution of up to 240,000[45,46] at 200 m/z. The resolution or "resolving power" R of a mass spectrometer is defined by its ability to distinguish between two neighbouring peaks. The formula for the resolution is

$$R = \frac{M}{\Delta M},$$

where M is the mass of a peak and ΔM is nowadays most often the *full width at half maximum* (FWHM) of a peak[47]. This can be surpassed by "Fourier transform ion cyclotron resonance" (FTICR) mass spectrometers, which have resolutions of up to 1,000,000, but are much more expensive and not applicable for high-throughput analyses.

2.4.1 Liquid Chromatography coupled to Mass Spectrometry

As explained before, the current method of choice for MS based proteomics is the (high-performance) liquid chromatography coupled to mass spectrometry (LC-MS). Here, the peptides are passed to a mass spectrometer after separation by LC in such a way, that molecules

are eluted and measured on the MS at a specific retention time. This separation is usually performed directly before the measurement and thus called online, whereas techniques, where the separation is performed and the fractions are collected for later measurement are called off-line separation. The LC improves the resolution by separating different analytes in the sample. Furthermore, it is necessary to allow the identification of peptides using tandem mass spectrometry.

To visualise successively recorded MS spectra, retention time (RT) m/z maps are generated (Figure 2.3). These maps are the basis for quantitative methods working with ion or ion trace quantification, like many label-free techniques[48]. As the quantification using MS is a very broad field itself, it will not be discussed in further detail in this work.

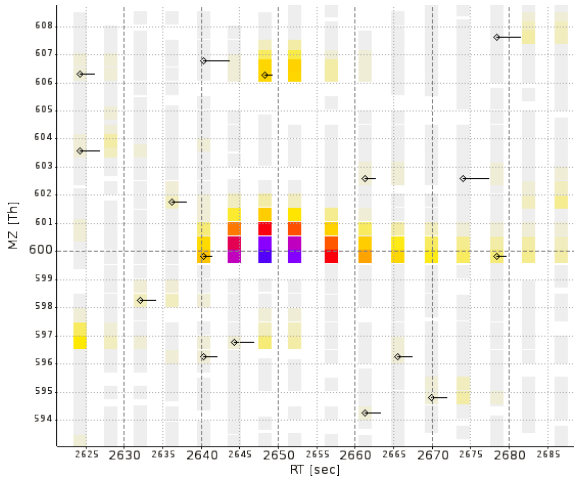


Figure 2.3: Detail of a RT-m/z map created by the OpenMS module TOPPView. The intensity of detected ions is colour-coded, going from white/light-grey (least intense) over yellow to violet (most intense). In the white "gaps" between the MS scans, MS/MS spectra were recorded: black circles indicate the triggering parent m/z and its RT, the end of the adjacent lines the actual RT of the recorded MS/MS. In the depicted map ion traces of several features (presumably peptides) are visible, one larger and more intense in the centre. On a higher zoom levels, the m/z of individual isotope levels would be visible. These features can be used for quantification approaches using mass spectrometry.

2.4.2 Tandem MS

The identification of peptides based on a mass spectrum alone poses big challenges when using database approaches. The main reason is, that many peptides with identical masses within an instrument-specific tolerance window exist in current protein databases. In order to solve ambiguities on MS level one identifications, tandem MS (MS/MS or MS2) spectra are created. It is assumed that ions of a certain m/z ratio at a single retention time belong to a specific peptide in the sample. In a data dependent acquisition (DDA), the ions with the highest intensity after a MS level one scan over the complete m/z range are subsequently

fragmented (compare Figure 2.3). For this, the selected parent ions of the specified m/z ratio and inside a certain tolerance window are collected and some kind of energy is applied to break the peptide ions. For most experiments considered in this thesis, the fragmentation used is collision induced dissociation (CID): the peptide ions are accelerated by an electrical potential and collided with a collision gas, which induces a fragmentation[49]. It is important to know, that the backbone of peptides breaks for specific fragmentation methods preferentially on well defined positions[50] (e.g. mainly b and y ions are created by CID), which will be explained in the next section. Thus, it is feasible to match these peptide fragment spectra to the original peptide sequences, as described in Section 3.1.

Structure of Peptides

Proteins and peptides are biochemically amino acids concatenated by peptide bonds. The peptide bond is a special chemical bond, which connects the α-carboxyl group of one amino acid with the α-amino group under the loss of a water molecule[51]. Therefore, the formation of a peptide bond is a condensation. Most important is the fact, that an arbitrary number of amino acids can be concatenated by peptide bonds. These polypeptides contain all a backbone with repetitive elements and different side-chains for the respective amino acids, see Figure 2.4. The end of the peptide backbone, which has a free amino group, is called the N-terminus, the other end, with a free carboxyl group, is the the C-terminus. The peptide or protein sequence is usually read from the C- to the N-terminus. All the proteins of the known living organisms are composed of 22 proteinogenic amino acids in varying sequences. Besides the peptide bonds, there can be further static bonds between the side-chains of amino acids, like disulfide bridges, as well as transient connections like hydrogen bonds, which form secondary and tertiary structure elements. These though will be no further discussed in this work.

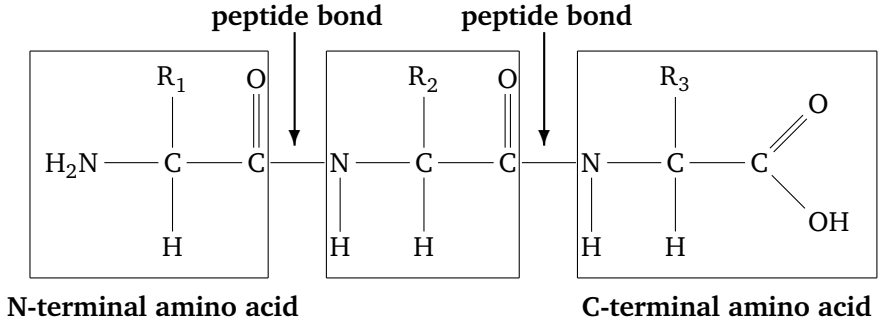


Figure 2.4: A peptide consisting of three amino acids. The sketch shows three amino acids connected by peptide bonds between the carboxyl and amino groups of two neighbouring amino acids. The side-chains are abbreviated with R_1, R_2 and R_3 respectively. Each peptide consists of this basic structure with a variable number of inner amino acids.

Peptide Fragmentation and Creation of Ion Series

In MS-based proteomics, the fact that peptides break preferentially at the backbone is used to create tandem MS spectra, which can be matched to amino acid sequences. Furthermore, the preferred fragmentation site depends on how the energy to induce it was applied. When applying collision induced dissociation (CID), the peptides tend to break between the C and N atom of a peptide bond and thus creating mainly b- and y-ions[52]. Fragments produced by electron transfer dissociation (ETD) on the other hand produce mainly c- and z-type ions[53]. The nomenclature of the produced ions depends on the position of the backbone break: a, b and c ions contain the N-terminus of a peptide, while the corresponding x, y and z ions contain the C-terminus. An index at the ion, like b_2, indicates the number of contained amino acid side chains in the ion, see also Figure 2.5.

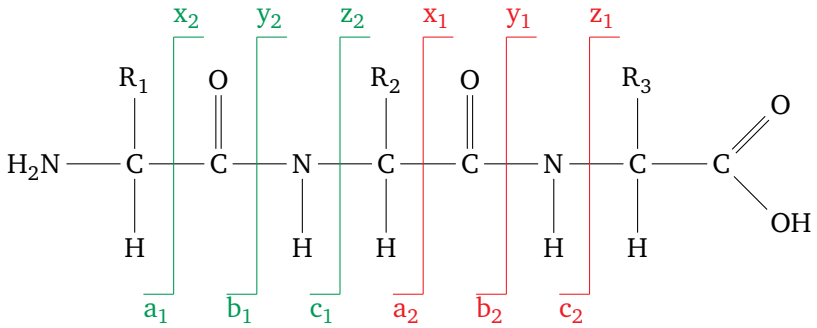


Figure 2.5: Nomenclature of ion series. This sketch highlights the breaking positions for the creation of ions and their nomenclature on a small peptide, in green the positions between the first and second amino acid, in red between the second and third. The a, b and c ions contain the free amino group of a peptide, while the x, y and z ions contain the C-terminus. The indices indicate the number of amino acid side chains in the respective fragment.

Whether an ion is detected or formed at all depends on many factors, like the type of applied fragmentation energy, but also the amino acid sequence and possible modifications. Also, only charged ions can be detected at all by the mass spectrometer, therefore the created fragment has to contain at least one charge and thus form an ion. All measured fragment ions of a precursor form an *ion series* or *ion ladder*. For the identification of the original peptide, it is important to create and measure a complete as possible ion series of the analytes, as explained later.

Annotation of Tandem Mass Spectra

With the knowledge of the preferred fragmentation site, it is possible to match a tandem MS spectrum to an amino acid sequence. For this, it is assumed that the most intense peaks of a spectrum correspond to the ion types created by breaking of the peptide at these fragmentation sites. Using this, the weight differences for each amino acid in a fragment and preferably some

more information about a tandem MS spectrum, like the precursor mass and charge, the amino acid sequence of a the originally analysed peptide can be recovered as explained in Section 3.1. After (or for de novo methods also during) matching a spectrum to an amino acid sequence, the spectrum can be annotated by the sequence as shown in Figure 2.6. Here the matching of the b- and y-ion series for the peptide DEVDGGPAGPPGGAAK on a spectrum is depicted. In Table 2.1 the masses of the respective ion series and the annotated fragments are shown. In the example it can be seen that the singly charged y-ion series can be annotated completely from y_4 up to y_14, and the b ions from b_2 to b_12. With the usage of doubly charged ions it is possible to annotate the complete peptide sequence in this case. Annotations like this make a manual inspection of interesting results in a high-throughput identification possible.

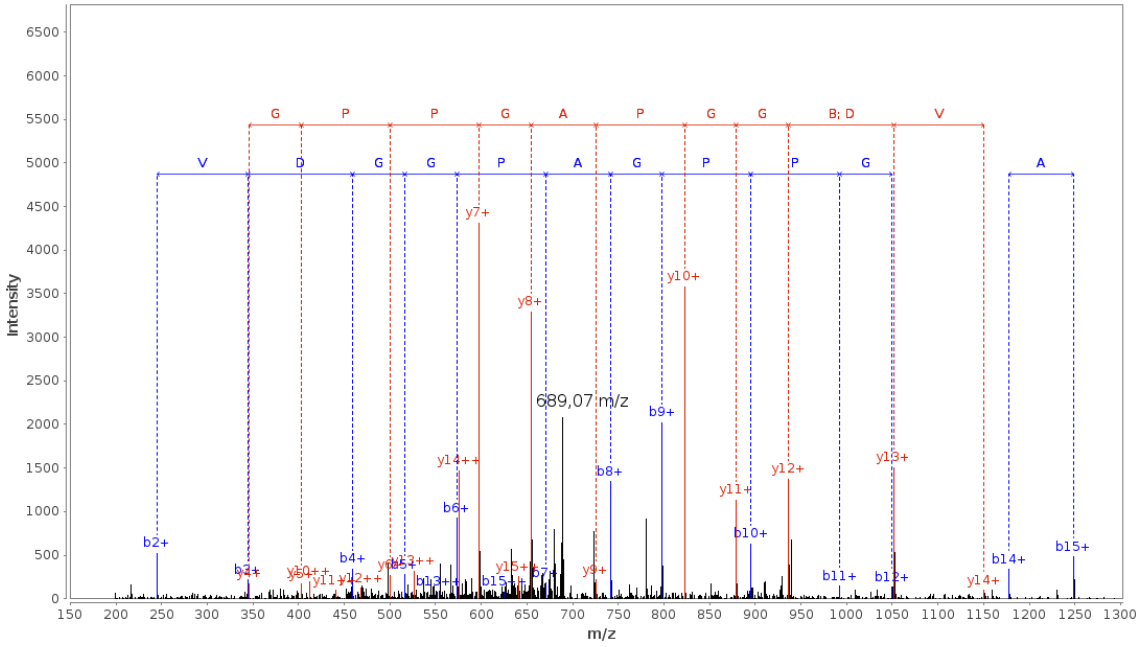


Figure 2.6: Annotated MS/MS spectrum for the peptide DEVDGGPAGPPGGAAK. The figure shows an annotated MS/MS spectrum for the peptide with the sequence DEVDGGPAGPPGGAAK from a 2+ charged precursor ion. In red the ion series for the y-ions is annotated and in blue for the b-ions. The distances between two peaks are characteristic for specific amino acids. Both series are almost complete (compare Table 2.1) and most high abundant peaks are explained by the sequence, which lead to a relatively high Mascot Ion Score in this case (68.86). But there are also some non-annotated peaks with high intensity, for example around 689.07 m/z. This specific fragment could be a neutral loss of water (688.83 m/z) or ammonium (689.32 m/z), which both are in the tolerance range for a spectrum of the used LTQ Orbitrap Elite (0.4 Da). As in almost all spectra, there are also several low abundant peaks, of which some are belonging to contaminants or also noise. The annotated spectrum was generated with the PRIDE Inspector 2.5.

Table 2.1: The ion series for y- and b-ions of the 2+ charged peptide DEVDGGPAGPPG-GAAK identified by Mascot. This table shows the m/z values for all possible y- and b-ions of the given sequence. The highlighted masses correspond to the annotated peaks in Figure 2.6. These theoretical values are often used by database search engines to match a spectrum to a peptide, but can also be generated by *de novo* sequencing strategies.

y ion series			**seq**	**b ion series**		
pos	**y**	**y++**		**b**	**b++**	**pos**
15	1279.628	**640.318**	D			
14	**1150.586**	**575.797**	E	116.035	58.521	1
13	**1051.517**	**526.263**	V	**245.077**	123.043	2
12	**936.490**	**468.749**	D	**344.146**	172.577	3
11	**879.469**	**440.238**	G	**459.173**	230.090	4
10	**822.447**	**411.728**	G	**516.194**	258.601	5
9	**725.395**	363.201	P	**573.216**	287.112	6
8	**654.357**	327.683	A	**670.268**	335.638	7
7	**597.336**	299.172	G	**741.305**	371.157	8
6	**500.283**	250.646	P	**798.327**	399.667	9
5	**403.230**	202.119	P	**895.380**	448.194	10
4	**346.209**	173.608	G	**992.432**	496.720	11
3	289.188	145.098	G	**1049.454**	525.231	12
2	218.150	109.579	A	1106.475	**553.742**	13
1	147.113	74.061	A	**1177.513**	589.260	14
			K	**1248.550**	**624.770**	15

Chapter 3

Computational Background

After the measurement of a biological sample by a mass spectrometer, the data needs to be interpreted. A contemporary MS run generates tens of thousands of spectra to be analysed and interpreted, which obviously cannot be done by hand. This chapter gives an overview of the computational background required for analyses of data obtained from bottom-up MS proteomics experiments. First, the interpretation and identification of LC-MS/MS data is explained in Section 3.1, which in this work is confined to the peptide-spectrum matching. After briefly discussing the concepts of several widely used search engines, some common protein databases are explained. In Section 3.2 approaches to estimate and maintain the quality of peptide identifications are highlighted and in Section 4 an examination of some protein databases regarding shared peptides is given. This directly leads to the necessity of defining some terms to model the characteristics and relations of inferred proteins, which is discussed in Section 3.4. The Human Proteome Organization (HUPO) and the efforts of its Proteomics Standards Initiative (PSI) for computational mass spectrometry are finally described in Section 3.5. The definition of standards is important to let bioinformaticians and developers focus on the tasks of analysing data or creating tools and not how to extract information from vendor data.

3.1 LC-MS/MS Data Interpretation

After the measurement in an mass spectrometer, a scientist is provided with the spectral data of the samples. In the currently most widely used MS based proteomics method, the bottom-up or shotgun approach, the actual proteins were digested into peptides, as described in the prior chapter. Thus, in theory the data of any MS/MS spectrum contains only the fragment ions of a single peptide ion. As the induced breakage of the peptide ions is well defined, it is possible to identify the sample peptide from an MS/MS spectrum. In the early years of MS proteomics, the spectra were only few per run and could be inspected and annotated "by hand" by the scientists.

In a modern high-throughput setting, where usually some ten-thousand MS/MS spectra are generated in each run, this is no longer possible. To overcome this, there are currently three computational strategies for data analysis: *de novo* sequencing, database searching and spectral library searching. An implementation of any of these techniques in proteomics is called a "search engine" (SE), sometimes more precisely a "peptide search engine" or (less precisely) "protein search engine", which obviously is not to be confused by web search engines. A SE usually calculates a SE specific probability or score value for a peptide spectrum match (PSM), i.e. a value for how well a peptide matches the given spectrum. As it is tedious to always exactly differentiate between scores and probabilities, the more commonly used term "score" is used throughout this work as a designator for both, unless stated otherwise.

3.1.1 Peptide Search Engines

In a ***de novo* sequence analysis**[19], the software does the same as a scientist would do by hand: by inspecting the peaks of a MS/MS spectrum for amino acid specific distances, respectively their ion series (for example mainly b- and y-ions for CID spectra), the original peptide sequence can be recovered. The more complete an ion series can be restored, the better the spectrum might be scored. As this most naive approach is rather time consuming, faster strategies were implemented recently[18]. All these *de novo* approaches obviously lack the link from peptides to proteins. A mapping of the identified peptides to a protein database can be performed afterwards to recover this knowledge. The mapping should probably be error tolerant, to allow for example amino acid mutations, which hinder a database search approach. For unsequenced species, i.e. species which genome is unknown and therefore also no protein database is available, *de novo* approaches are still widely used.

For sequenced species, **database searches** are the current standard for peptide identifications in MS-based proteomics. Therefore it will be discussed in more detail in the following section.

Spectral library searches build on *de novo* and/or database searches. In these approaches, the spectra of well matching PSMs from prior searches are used and matched against newly measured spectra. These approaches have the advantage of circumventing any theoretically generated spectra. Furthermore, for the quality of the match also the intensity of the ions can be used. The identification of peptides, which were not at least once identified before, is impossible, though. Also, the spectral libraries are dependent on the generating mass spectrometer, and partially also on the used setup. For samples measured by the SWATH[54] method, adjusted implementations of spectral library identifications are currently the most promising attempt. This is due to the fact, that SWATH fragment spectra often contain intensity peaks of multiple peptides. The identification of chimeric spectra though is a big problem for most database search engines.

3.1.2 Database Search Engines

The idea behind protein database searches in MS-based proteomics is to start with a protein database, which contains entries for the proteins of the analysed sample, and to match the measured spectra against theoretical *in silico* spectra or spectrum models, generated from peptides. To create *in silico* peptides from the databases' proteins, mostly regular expressions describing the cleavage sites of enzymes are used. During the creation also missed cleavages are allowed, which occur regularly in real world samples. As the mass of a measured precursor ion is known, the number of peptides from the database, against which a measured spectrum has to be matched can be greatly reduced. The theoretical MS/MS spectra or spectrum models are created from the *in silico* peptides taking into account the preferred breaking of bonds in the peptide backbone during the MS/MS fragmentation. Depending on the algorithm, either arbitrary scores or e-values are returned to measure the quality of a peptide match for a spectrum. Most algorithms return more than one of these as quality measures. Eventually, a spectrum can only be identified correctly, if its original peptide is present in the database. Therefore, the choice of the underlying sequence database is one of the biggest issues for all database searches. During the last two decades a plethora of different methods and implementations for database searches emerged. Some of the search engines, which are most common and were also used in other parts of this work, are summarised in the paragraphs after the following issues.

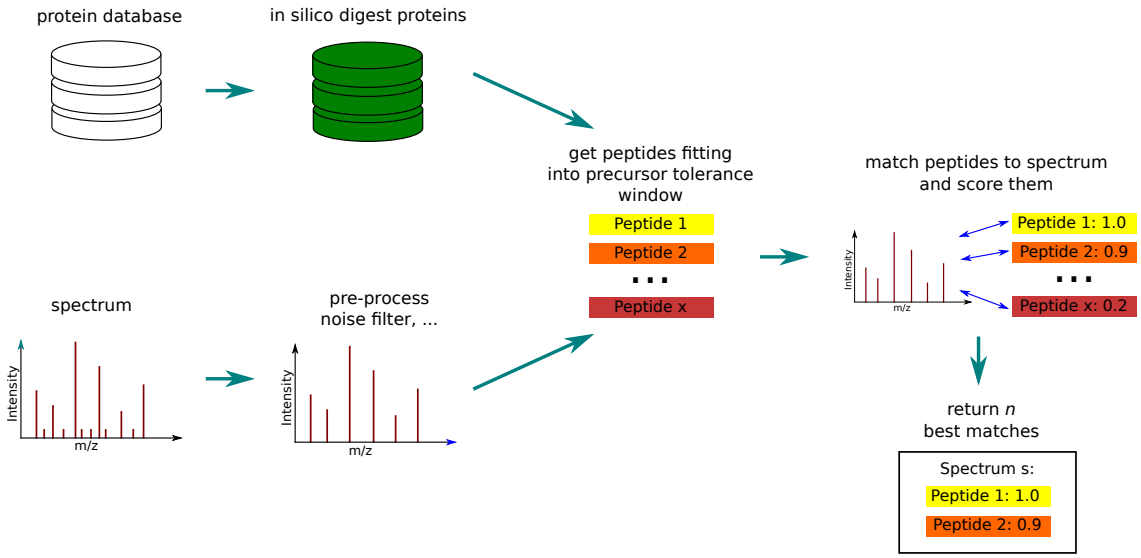


Figure 3.1: Schematic overview of a protein database search. The proteins in a database are *in silico* digested using the specified cleavage sites. Each spectrum is optionally pre-processed (e.g. using a noise filter) and all peptides fitting into the specific precursor window are selected from the database. These peptides are matched against the spectrum and scores are calculated. Finally, either all possible or only a selected number of the best matches are reported together with the scores.

Common Issues of Peptide Search Engines

A parameter, that greatly influences the outcome of a database search, is the choice of the allowed modifications and number of missed cleavage sites. Static modifications, which usually originate from the sample preparation like the common carbamidomethylation of cystein, do not increase the search space, but are rather a mass shift for the affected amino acids. Any variable modification, though, increases the search space and thus usually the runtime and a search with and without the modification may lead to significantly different results. Some of these modifications are very common, like the oxidation of methionine, which is used as a default setting in most searches. While the modifications are used to more accurately explain MS/MS spectra, the usage of inaccurately set variable modifications may lead to an increase in false positives. Some search engines (e.g., Mascot and X!Tandem) give the opportunity to perform a "second round" or refinement search to restrict the increased runtime. With this approach, the spectra are matched in a first round with only some modifications and in a second step, more modifications are allowed, but only matching against peptides belonging to proteins, which were already identified by some peptides in the first round. Some of the above mentioned search engines also have a more elaborate scoring scheme to handle modifications, which will not be further explained in this work.

One big issue in almost all high-throughput methods is the estimation of false positives. The scores and probabilities, which are calculated by the SEs, may give a hint, whether a specific match is good or bad. But with most implementations it is possible to match any sequence against any spectrum, yielding presumably a relatively bad score. Some approaches to overcome this problem are given in Section 3.2. Furthermore, the SEs report, which peptide of a given set of peptides best matches a given spectrum and score this. While the peptide scores are thus comparable for matches of a specific spectrum, the scores are not per se comparable between different spectra[55]. This bias can be adjusted by calibrating the scores, though this is not a wide spread method yet.

SEQUEST

SEQUEST[15] was one of the first protein database search algorithm for high-throughput mass spectrometry. Several consecutive calculations of different scoring types are performed. The first score, the S_p, takes into account the theoretical fragments, the actually measured ions matching these, the continuity of matched ion series (which series, e.g. b- and y-ions, must be given) and also the presence of immonium ions for several amino acids of the matched sequence. Based on this score, the top ranked peptide matches for a spectrum (500 matches in the original implementation) are further analysed performing a cross-correlation. For this, the theoretical and the measured spectra are aligned and shifted along the m/z axis, computing the correlation for each shift. For a good match of a peptide to a spectrum, the correlation

should be significantly better at one shift position than on all other. This should be at a zero-shift for a well calibrated mass spectrometer. This analysis is reflected in the $XCorr$ score, which is in this and most other works used as the main score of SEQUEST. Additionally the difference to the next best match of the same spectrum is reported as ΔC_n, primarily to distinguish correct matches from false positives. The original implementation is rather slow, when compared to more recent algorithms. Nevertheless, SEQUEST is still widely used, though mainly using faster, parallelised implementations now. For a long time, the original algorithms were patented and distributed exclusively by Thermo Finnigan. Meanwhile several open source implementations exist, most of them further decreasing the necessary runtime. Among the more recent implementations are for example Tide[56] and Comet[57].

Mascot

One very widely used commercial software is Mascot[16], developed by Matrix Science, which is based on MOWSE[58] and handles besides MS/MS ion searches also peptide mass fingerprints. Though the actual used algorithm is not explained in detail and its implementation is not open source, some of the basics are known. Mascot's scoring algorithm is probability based, i.e. it reports the probability that a peptide matches randomly in the given database. This approach has benefits, like an easily interpretable score, but also a dependency on the underlying database content as drawback. For the actual scoring, the most intense peaks, which lead to the lowest randomly matching probability score on the given ion series, are used. The capability to run Mascot on a server environment, using multiple nodes, makes it a good choice if short runtimes for many spectra are required.

X!Tandem

X!Tandem, which was originally called TANDEM[17] and released in 2003, claims in its original publication to be the first non-propriety and open-source implementation of a database search engine. The scoring system described in[59] calculates in a first step the dot-product of a measured MS/MS spectrum and the theoretical spectra of the database's peptides, which fall in a specified precursor mass tolerance. The dot-products are further refined to the so called HyperScore, multiplying it with the faculties of the numbers of matching b-, y- and possibly other ion series which were selected for scoring. The scores are assumed to be distributed under an extreme value (or Gumbel) distribution. The logarithmic values of the counts can be interpreted as results of a survival function and can thus be seen as e-values. An E-value (or e-value for expectation value) is a score, that describes for the experiment how many random hits in the database are expected to have the same or a better score. Logarithmic counts plotted against the original score allow a linear interpolation for the high scoring portion of the plot. Thus, the e-value of the peptide with the best HyperScore is calculated and reported.

X!Tandem is a very fast and commonly used search engine, which gets regular updates and supports community standard file formats for import and export.

OMSSA

OMSSA[60] stands for Open Mass Spectrometry Search Algorithm and is another open source search engine, developed by the NCBI and released only shortly after X!Tandem. The first step of the search algorithm of OMSSA is a noise filter, filtering out the low intensity peaks and allowing for charge 1+ spectra only one peak in a 27 Da window, for higher charged peaks two peaks in a 14 Da window. These values are chosen, as they are smaller than the residue mass of the smallest amino acid, which is glycine with an immonium ion mass of 30 MW. These two peaks are allowed, to allow peaks of two different ion series in one bin. These filtered peaks are than matched against the theoretical spectra of peptides within the precursor mass tolerance. The base score for OMSSA is the number of matching peaks for a PSM. It is assumed, that the distribution of these scores for all matched peptides follows a Poisson distribution. This assumption is finally used to calculate an e-value for each PSM. Though OMSSA was, and to a certain degree still is, used in many workflows, it is no longer maintained and thus its usage will probably slowly cease.

MS-GF+

A more recent search engine, which gained much popularity during the last years is MS-GF+[61]. A unique feature of this algorithm is the usage of the "generating function approach"[62,63], which is not used in any other database search algorithm. In this approach, not only all peptides that fall within the mass tolerance in the database are scored against a respective spectrum, but all possible amino acid sequences. With this the generating function estimates an e-value given the best score for a spectrum obtainable by the peptides in the database and the best score for all possible peptides falling into the respective mass window. With this approach it is claimed that no further estimation of false positives is necessary, as all possible peptides are tested for each spectrum. MS-GF+ fully integrates community standard formats and is open source, programmed in Java, which makes it very portable. The runtime and memory imprint of MS-GF+ is, compared to other search engines, relatively high, but the developers claim in its publication, that it is much more sensitive than other search engines, i.e. it reports more high quality PSMs than others, and is readily available for all types of MS/MS experiments.

3.2 Estimating the Quality of Identifications

As with all high-throughput technologies, mass spectrometry suffers from the identification of false positives due to multiple testing: every single feature (in this case PSMs) may be

identified with a relatively low error probability. But the fact, that there are ten-thousands of these features makes the probability that none is a false positive small. The score of most search engines describes in some way the probability of a PSM to be a random hit, as described above. But most of these estimates are dependent on the sample, the instrument and/or the database. Additionally, the distribution of the scores is not known. To overcome this problem, one of the most widely used strategy in MS proteomics is the target-decoy-approach (TDA)[64,65]. This approach allows the estimation of a false discovery rate (FDR)[66] and thus controlling the allowed number of false positives in a list of reported identifications.

The original idea behind the TDA is, that the search engines are presented with sequences, which are not part of the original (target) database, but are decoys, which when matched represent false identifications. This approach is based on the assumption that a search engine may match a target sequence with the same likelihood as a decoy. For this, it is necessary that roughly the same number of target and decoy peptide sequences fall into the precursor tolerance for any given spectrum. This was shown to be true in the original manuscript[64], but with increased accuracy of modern MS instruments tends to hold only for large enough databases, more accurately large enough decoy parts of databases.

3.2.1 Creation of Decoy Databases

For the creation of decoy databases several tools exist, for example the DecoyDatabase utility in OpenMS or the DecoyDatabaseBuilder[67]. Three conceptually different strategies to create decoys exist: the creation of random protein sequences, the reversing and the shuffling of existing proteins. The creation of totally random protein sequences is the least used of these strategies, as usually some biological aspects, like the frequency of amino acid usage in proteins and protein lengths, are desired to match between decoy and target databases. Using reversed sequences as decoys simply reverses the amino acid sequences, while when using shuffling, the original amino acids of a protein are permutated to create the decoy sequence. There has always been some debate in the literature, whether it is better to use reversed or shuffled proteins as decoys[67,68], until now without any conclusive result and both strategies are used next to each other. An argument for reversing the targets is that the average length of peptides are identical for targets and decoys. Shuffling, on the other hand, creates more seldom palindromic sequences. Also, there was always some debate, whether a combined target-decoy database should be used for identification or both databases should be used separately. In this work, when not stated otherwise, combined target-decoy databases with shuffled decoys were used.

3.2.2 Estimation of the False Discovery Rate

The goal of the FDR estimation is to estimate the amount of false positives (FP) and thus to limit the ratio of FPs in the reported identifications. As it cannot be known after a database

search, whether a target identification is FP or true positive (TP), the decoy identifications are used as placeholders for FPs. The idea behind this is, that under the assumption that decoy and target matches are equally likely on noisy spectra, each decoy identification (which is a true negative, TN) in a ordered list of PSMs stands for another FP with a similar score in the neighbourhood. This allows to calculate a local FDR in a list of PSMs, that is ordered by the scores. The original formula to calculate the FDR for any given rank i in the ordered list would be

$$FDR_i = \frac{FP_i}{TP_i + FP_i},$$

where FP_i respectively TP_i are the number of FP and TP identifications up to the rank i. The actual values for TP_i and FP_i are not known, as explained before. But the sum of them is the number of target identification $\#targets_i$ up to rank i in the list. For FP_i it is assumed that it is the same as $\#decoys_i$, the number of decoys at the given rank. Therefore, the used formula throughout this work is

$$FDR_i = \frac{\#decoys_i}{\#targets_i}.$$

This is one of the proposed and widely used formulas for the FDR calculation using the target decoy approach[69]. There is also a slightly different formula, which is not used throughout this manuscript[64].

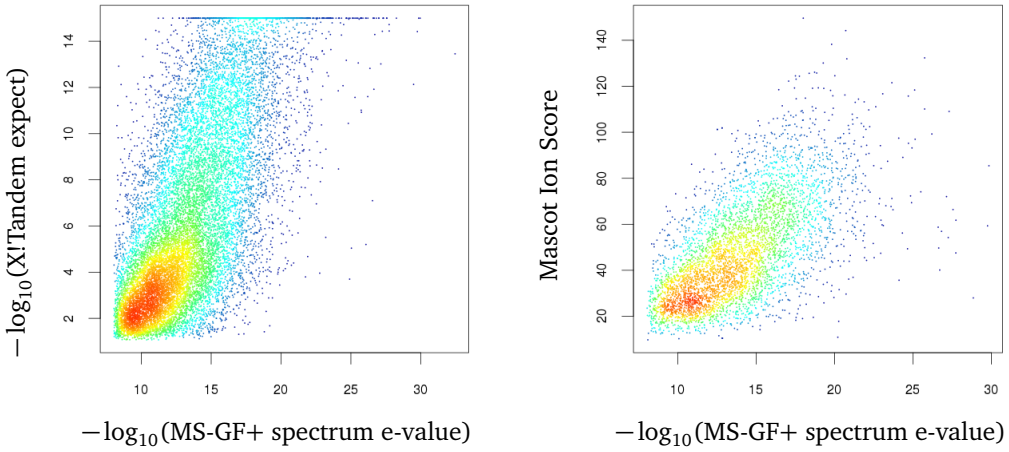
Often the q-value instead of the local FDR is used as a meta-score in MS proteomics. The value for a local FDR increases with every found decoy in the ordered list and decreases with subsequent targets. The q-value at a given position is the smallest possible FDR value at this or any following position in the list. Therefore, the q-value is monotonically increasing with steps at each decoy in the list. This allows setting an FDR threshold and filtering for a specific false discovery rate. The most commonly used threshold at the time of writing is 0.01 or 1%, which effectively allows 1% false identifications to be in the final list of reported PSMs. Provided the used FDR estimation is correct. Though the calculation was only explained for PSMs here, it can be similarly performed on the peptide and protein level.

3.2.3 Smoothing the FDR and Combining Search Results

The stepwise increases in the q-value can be smoothed as shown by Jones et al. in[27]. This smoothing is performed by interpolating the q-values between two decoy identifications based on the scores, which were originally used for sorting. The resulting meta-score was named the *FDR Score*.

As each SE has its own algorithms and assumptions for the score calculation, combining the results of different SEs is not straight forward. Also, in general the scores cannot be linearly scaled to convert one score into another, as is depicted in Figure 3.2. Jones et al. therefore proposed a recalculation of the *FDR Score* to reflect, how good the agreement over multiple

search engines is for the reported PSMs. Basically, the PSMs are sorted into groups reflecting the combinations of search engines and for each set the *average FDR Score* is calculated for the PSMs, respectively PSM sets. A PSM set includes the identical PSMs for a spectrum identified by different search engines. For these, the *average FDR Score* is a geometric mean of the PSMs' single *FDR Scores*. For the groups of PSMs, which were identified by only a single search engine, the *average FDR Score* is set to the same value as the previously calculated *FDR Score*. Finally, in each group of PSM (sets) for each SE combination, the PSM (sets) are ordered by their *average FDR Scores* and a *Combined FDR Score* is calculated in the same way, as the *FDR scores* were calculated from the original SE scores.



(a) MS-GF+ against X!Tandem

(b) MS-GF+ against Mascot

Figure 3.2: Plotting the scores of PSMs identified by different search engines. Shown are the plots of 1% FDR valid PSMs found by both of the respective search engines. The density of the plotted scores is color-coded, ranging from dark blue (less dense) to red (densest). These plots highlight, that there is no linear dependency between the scores, thus a direct comparison of results originating from different search engines is not possible.

Using the *FDR Scores* for the combination of identifications obtained from different search engines is included in PIA and therefore explained here. There are several other commonly used methods, like the calculation of the *Consensus Score*[28] or employing iProphet[70]. Merging the results from multiple search engines is desirable to either increase the number of identified spectra passing an FDR threshold and thus hopefully also the number of corresponding proteins or to solidify the evidence of peptides detected in the analysed sample[26].

3.3 Protein Databases

While in *de novo* approaches a reported peptide depends on the used algorithm, in database searches, only peptides contained in the underlying database can be reported. Therefore the choice of the database is a crucial point. *UniProtKB*[71] and the protein part of the *NCBI databases*[72] are probably the two most commonly used database resources for MS proteomics at present. Besides these, there are many other databases like the species specific *WormBase* (for *C. elegans*) and the *Saccharomyces Genome Database* (SGD, for *S. cerevisiae*). Mainly only the simple amino acid sequences in the (relatively loosely defined) FASTA format are used to match spectra by a search engine. Though most databases offer much more information about the containing proteins, like the genes, gene ontology terms, secondary structure elements and, if known, also clinical knowledge of the respective protein.

Most protein entries of the databases originate from genome annotations. Correct genome annotation, especially the identification of translation start- and stop-sites and in higher organisms also the annotation of exon-intron-structures is one of the oldest fields in bioinformatics. But it is still error prone and *in silico* annotated proteins usually need validation on multiple layers. For this, the UniProtKB gives five levels of *protein existence* (PE), in decreasing order of reliability these are:

PE=1 Experimental evidence at protein level
PE=2 Experimental evidence at transcript level
PE=3 Protein inferred from homology
PE=4 Protein predicted
PE=5 Protein uncertain

For the numbers of proteins with the respective PE, see Table 3.1.

The UniProtKB also includes the annotation of protein isoforms, which have alternative sequences of a gene product compared to the canonical sequences. These may be splice variants, alternative promoter usage, alternative initiation or ribosomal frameshifting[73]. Often, only the canonical sequence is used for spectrum identification, though this may not be the actual expressed protein in an analysed sample. The entries in UniProtKB are further structured into the curated Swiss-Prot and the automatically annotated TrEMBL part. The entries in Swiss-Prot are non-redundant, manually annotated and most have a high level for the *protein existence* (see Table 3.1). For the human and mouse portions of the Swiss-Prot, the numbers of entries are almost equal to the the numbers of annotated protein-coding genes. The entries in TrEMBL are computational analysed transcript data, enriched with automatic annotation.

It is advisable to perform the peptide searches on a database, which covers all the proteins of the species contained in the analysed sample[74]. For UniProtKB these are the "complete proteomes" (or only "proteomes"), which are available for most common species used in analyses.

Table 3.1: Table showing the numbers of proteins with the respective protein existence level. The numbers are shown for the human part of the Swiss-Prot and the human complete proteome of the UniProt release 2015_11. For both databases, most entries have experimental evidence at protein level (PE=1), but for the complete proteome, a large proportion consists of predicted proteins only.

Database	PE	proteins (%)	
Swiss-Prot (*H. sapiens*)	1	14,696	(72.77)
	2	4,136	(20.48)
	3	651	(3.22)
	4	123	(0.60)
	5	588	(2.91)
Proteome (*H. sapiens*)	1	51,298	(73.20)
	2	4,595	(6.56)
	3	1,206	(1.72)
	4	12,388	(17.68)
	5	588	(0.84)

The complete proteomes not only contain the complete Swiss-Prot part of the species but also the TrEMBL entries, which can be mapped to annotated proteins on the genome. Whether additional sequences for isoforms or known single nucleotide polymorphisms (SNPs) should be included depends on the goal of the respective study.

To match as many good spectra as possible, it is also good practice to use a database containing common laboratory contaminants, like amylase and keratin. The number of entries in a database used for spectrum identification influences the quality of the results. While, when using a too small database, it might be impossible to match many good quality spectra and an estimation of false positive matches is impeded, a too big database will lead to an overestimation of false positives.

3.4 Specifications of the Terminology for Protein Ambiguity Groups, Sub-Groups and Clusters

In the beginning of the MS/MS proteomics, it was popular to report "long lists" and often it was considered, that "the longer the list, the better the analysis". At this time grouping of proteins was sometimes neglected and proteins were reported independently, even if proteins were identified only by shared peptides and there was no actual evidence which could distinguish one protein from another. Nesvizhskii et al. highlighted already in 2005[22], that it is necessary to report groups of proteins together, if there is no way to decide, which protein has more evidence than any other with the same (sub-)set of peptide identifications. The necessity for a protein inference due to the shared peptides will be further highlighted in Chapter 4. In this section some terms describing the relationship of reported proteins will be explained. In

this work, mostly the same terms are used as in the mzIdentML[75] documentation and in "A standardized framing for reporting protein identifications in mzIdentML 1.2"[76].

Groups of accessions, which are inferred from exactly the same PSMs or spectra, are called a "protein ambiguity group" (PAG), or shorter only group. As a matter of concept, a PAG can also consist of only one accession and its PSMs. Groups, which are inferred by a sub-set of the PSMs of another group, are termed sub-groups. Depending on the used protein inference and its reporting method, these sub-groups are reported or not. Furthermore, the identical sub-group can be assigned to multiple PAGs, if each contains the same sub-set of peptides. A cluster is a set of groups, which for some reason belong together, for example depending on the respective data or additional biological knowledge. PIA for example puts groups deriving from the same connected component in its intermediate data into one cluster.

To explain the importance and difficulties of reporting ambiguity groups, sub-groups and clusters correctly, it will be explained in more detail with the aid of Figures 3.3, 3.4 and 3.5. These figures are generated based on identifications of the iPRG 2008 benchmark dataset, which will be further explained in Chapter 5.

Throughout this work, the inference from identified spectra to corresponding proteins will be performed on three levels: peptide spectrum matches (PSMs), peptides (respectively amino acid sequences) and proteins (respectively database accessions representing protein sequences). One way to depict the relationship between these three layers is by drawing a directed acyclic graph with arrows describing a "belongs to"- or "contains"- connection, as further explained in Section 6.2.5 and shown in Figures 3.3, 3.4a and 3.5a. These graphs show three clusters with increasing complexity between the different layers.

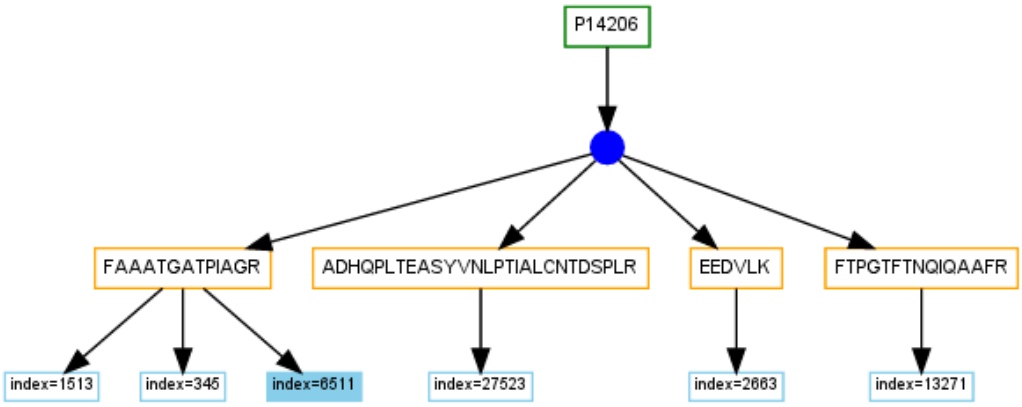
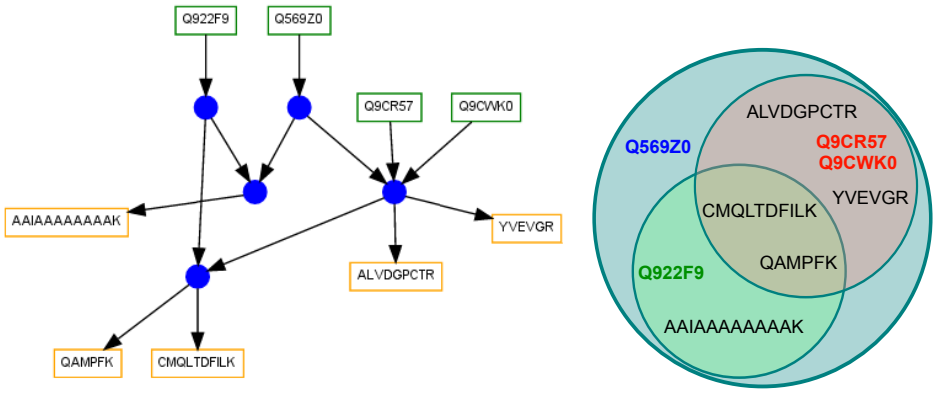


Figure 3.3: Sample of the connections between a single accession/protein, peptides and spectra. For the given sample, the accession P14206 (green) has evidence by four peptides (orange). Three spectra (lightblue) were identified for the left peptide, the other three peptides have one identified spectrum each. The blue circles are used to connect all nodes correctly, the reasoning behind this is explained in detail in Section 6.4.2.



(a) **(b)**

Figure 3.4: Sample showing the connections between accessions, peptides and PSMs for a more complex cluster. In a) the same visualisation method as in Figure 3.3 is used. The accessions Q9CR57 and Q9CWK0 belong to the same connection group and therefore form a PAG. The accession Q569Z0 was identified by one more peptide than the PAG with Q9CR57 and Q9CWK0, which makes this PAG a sub-group of the group containing Q569Z0 only. The group of Q922F9 forms another sub-group of Q569Z0's group, which does not contain the two peptides of the PAG with Q9CR57 and Q9CWK0. b) shows the same relations by drawing overlapping peptide sets with their corresponding accessions.

The graph in Figure 3.3 shows the relatively straight-forward relations of exactly one protein, its peptides and PSMs. The protein (in the green box) was identified by four peptides (orange). Of these peptides, only one was detected by three PSMs (light-blue), the remaining three by one PSM each. To ascertain the graph to be acyclic and thus allow from each node to any other node exactly one path along the graph, connection nodes (blue circles) are added into the graph. The algorithm for the creation of the graph is discussed in Section 6.4.2. Each accession is constructed in a way, that it belongs to a connection node. Thus it is easy to see, when a set of accessions are identified by the same set of peptides and PSMs: these accessions are attached to the same node. In Figure 3.4a this is the case for the two accessions Q9CR57 and Q9CWK0, which therefore form a PAG. The accession Q569Z0 was identified by one more peptide than the PAG with Q9CR57 and Q9CWK0, which makes this PAG a sub-group of the group containing Q569Z0 only. The group of Q922F9 forms another sub-group of Q569Z0's group. The graph in Figure 3.5a shows a rather complex example, which additionally contains sub-groups without any common peptides.

Another way to visualise the relationships between proteins, peptides and PSMs is by drawing overlapping peptide sets with their corresponding accessions, as shown in Figure 3.4b and Figure 3.5b. These overlapping sets are easier to interpret for a human, but much more complex to be created by an algorithm. In the less complex example (Figure 3.4b), one can directly see, that there is one group which contains all the peptides of the two subgroups. In the complex example, the actual relations are still hard to comprehend. But it is easy to see, that there is one big group (Q3U8U8), which contains almost all other sub-groups. Only the

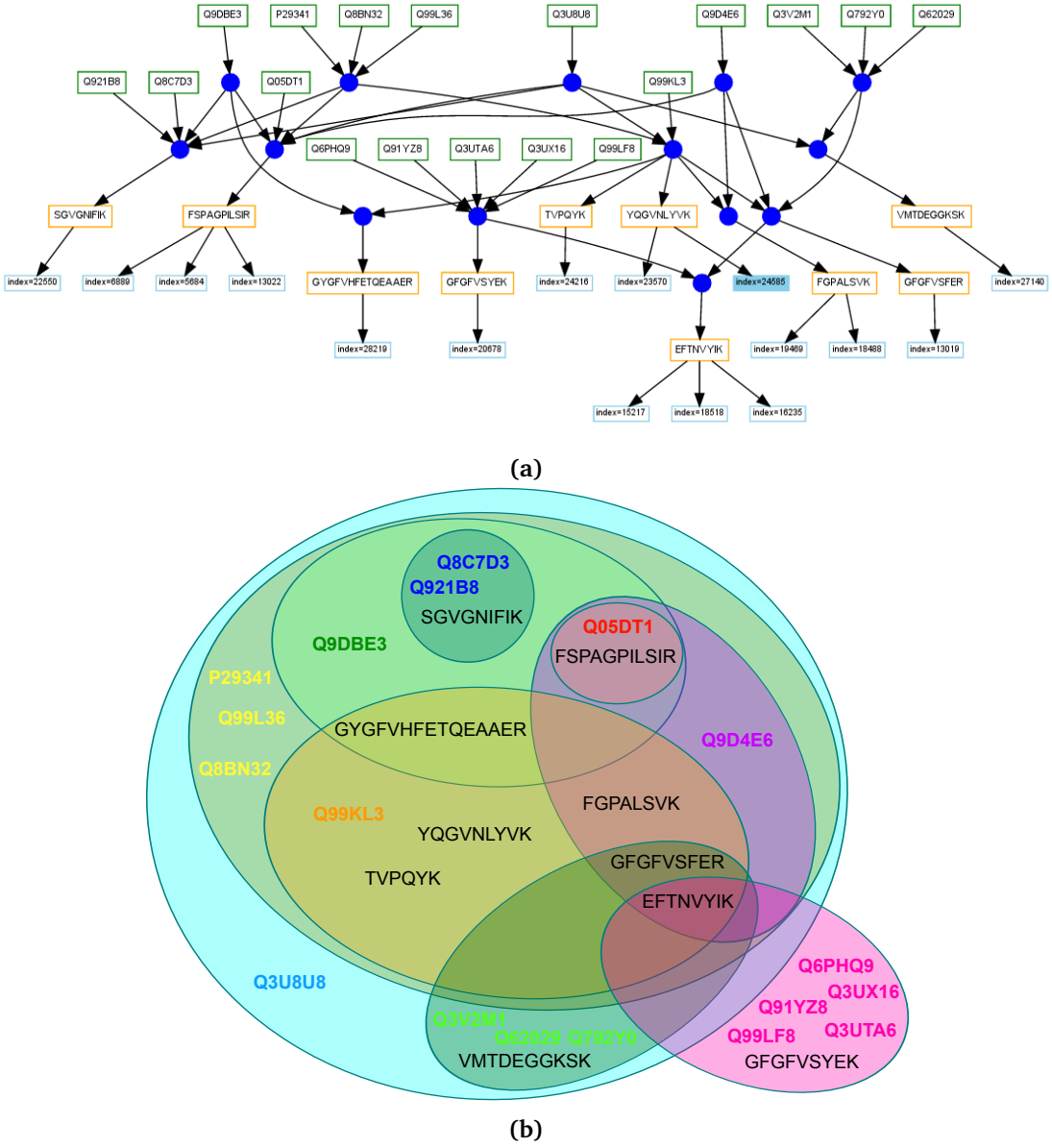


Figure 3.5: Sample showing the connections for a rather complex sample. additionally to Figures 3.3 and 3.4, this example contains sub-groups having no common peptides with any other group. While the visualisation in a) is getting hard to interpret, the sets in b) can still be interpreted by a human: one group containing Q3U8U8 encircles almost all peptides, except the one identified for the pink group containing five accessions.

group visualised in pink with five accessions has one additional peptide as well as only one shared with the big group.

3.5 The Proteomics Standards Initiative of the Human Proteome Organization

The Human Proteome Organization (HUPO) is an international scientific organisation, which acts according to its following mission statement (cited from `www.hupo.org`):
"*To define and promote proteomics through international cooperation and collaborations by fostering the development of new technologies, techniques and training to better understand human diseases.*"

The HUPO was launched in 2001 and is since actively involved in the development of proteomics. To achieve the goals of its statement, among other things the HUPO organises the "Annual HUPO World Congress", one of the biggest proteomics conferences worldwide, and furthermore has several initiatives, which work on various proteomics associated topics. For the work in this thesis, the activities of the HUPO "Proteomic Standards Initiative" (HUPO-PSI) are of special interest, as these address common bioinformatics challenges and provide community standard file formats, which allow scientists to focus on analysing data instead of writing parsers and converters.

3.5.1 Goals of the HUPO-PSI

The HUPO-PSI was founded in 2002[77] with the goal to define community standards for data representation in proteomics and thus facilitate data comparison, exchange and verification. The initiative is open for scientists but also industry members and journal representatives and everyone interested in defining and improving standard formats and guidelines may join and contribute to their development. There are three kinds of output generated by the HUPO-PSI: guidelines, standard formats and controlled vocabularies (CVs).

The "minimum information about a proteomics experiment" (MIAPE) guidelines[78] contain specifications about the minimal needed information to report complete results of specific techniques or experiments. If all points in the appropriate document are described in a publication, it should be possible to fully understand and even reproduce the experiment. The standard data formats facilitate the exchange between software packages. They aim not at replacing the vendor formats, but allow easier development of tools by using single application programming interfaces (APIs) instead of needing importers for every vendor's format. Though it is not a necessary part of the standard development, for most of the formats the participating developers or the community created a Java API alongside. Furthermore, the formats are at least in part human readable (mostly XML) and thus allow to extract information even without an API. Many fields in the standard formats are filled with values from controlled vocabularies (CVs), like the name of a specific analysis software, a physical unit or the name of a mass spectrometer

used for the analysis. The separated curation of the CVs from the standards allows a faster and easier update of the contained terms, than the complex update of a standard's schema[79].

If a new standard format is developed or changes outside the CVs are needed, the HUPO-PSI follows a self-created "document process". This includes public review and peer-reviewing of the specifications and, if necessary, revisions of the proposal. In this way the HUPO-PSI makes sure, that in the end outputs are developed, which suffice a community consensus.

3.5.2 Standard Formats for Peptide and Protein Identification

The HUPO standard for raw MS data is the mzML[80], which uses the best aspects of the two prior standards for raw data: mzData and mzXML. The new standard was developed in cooperation by the original developers of its predecessors, the HUPO-PSI and the Seattle Proteome Center at the Institute for Systems Biology. The mzML format has reached a relatively stable and mature status at the time of writing and converters for many vendor formats exist. Also, many search engines have native support to use mzML files as input.

As the FASTA format for sequence databases is neither standardised, nor holds much information besides the sequence, accession and sometimes a short description, the HUPO-PSI is currently developing a format which can be used by sequence search programs and associated tools or repositories. The proposed name for this format is PEFF, which stands for "PSI Extended Fasta Format". PEFF files will be text-based and backwards compatible with FASTA files, but will contain additional information like isoforms, splice variants, known mutations and post translational modification sites. The current schematics propose to encode these additional information into the sequence headers. At the time of writing this thesis, the PEFF format though was still under development without any publication or wide usage.

To communicate and store peptide and protein identification data, there are currently two formats with different scopes: mzIdentML[75] and mzTab[81]. MzIdentML aims to capture all available data which might be generated by a search engine, including all analysis settings, protein database sequences and even the product ions used for the identification. While this format is suited to provide a software tool with information, it is not practical to exchange and review identifications between scientists. For this, the tab separated mzTab format was developed, which by far is not complete in terms of information and does not suffice the MIAPE guidelines, but can be viewed by most spreadsheet software. While reporting of protein ambiguities was considered for both formats, mzTab has no support for the report of sub-proteins respectively sub-groups, same-sets and other complex protein relations. For mzIdentML, a guideline on how to report these cases was recently published and is further explained in the following paragraphs.

Reporting Protein Identifications in mzIdentML 1.2

The recommended ways of reporting protein identifications in mzIdentML 1.2, which are highlighted in the following paragraphs, have been published in[76]. This framing describes methods to clearly model the identification and inference of single proteins, groups and clusters described before in Section 3.4.

A single protein should in mzIdentML be encoded as a ProteinDetectionHypothesis (PDH). Each PDH is assigned to (at least) one ProteinAmbiguityGroup (PAG), for which the creating software has enough evidence to report it as an independent entity. A PAG as at least one score and a value, whether it passed a certain threshold for reporting. The groups may also have further scores like e.g. FDR or p-values. Each PDH must furthermore be annotated as a "leading protein" or "non-leading protein", to give some proteins more evidence. The "non-leading protein" can be used to flag members of sub-groups. Also, one member that is flagged as "leading protein" may be flagged as "group representative". Otherwise it is assumed, each "leading protein" could be used as representative. Each PDH can contain its own scores and additional information to define it as a sequence or spectrum subset or otherwise subsumable of another PDH. The cluster affiliation should be added to a PAG, if the exporting software supports this.

This framework enables the representation of quite complex relations between identified and inferred proteins. Unfortunately, the version 1.2 of the mzIdentML standard was at the time of writing this thesis not yet finalised and published. Most of the recommendations though can also be encoded into mzIdentML 1.1 and are already incorporated by PIA.

Chapter 4

Analysis of the Uniqueness of Peptides and Proteins

One of the biggest drawbacks of bottom-up MS proteomics is the fact, that peptides are detected, although the actual molecules of interest are most often proteins. This is due to the fact, that even modern MS instruments are not optimized for the analysis of complete proteins in high-throughput, but are specialized on measuring smaller molecules, as already discussed in Chapter 2. From this arises the need to infer proteins from the identified peptides.

The easiest way to infer proteins from identified peptides would be to simply argue, if the peptide was identified with a probability surpassing a certain threshold, the corresponding protein was in the sample. Unfortunately, it cannot be decided which exact protein was present in a sample with peptides, which originate from multiple proteins in the database used for the spectrum identification. Approaches for the protein inference will be discussed in the following chapters, here the uniqueness of peptides and accessions is highlighted first. As already mentioned before, the currently most widely used approach for spectrum identification is the database search. Therefore, it is dependent on the used databases, whether a peptide belongs to only one accession in the searched database (this will be called *unique* in the following) or is shared by several accessions respectively proteins. A peptide, which is unique in one database (for example the Swiss-Prot portion of UniProtKB) may be shared in a database containing more entries (for example a complete proteome of UniProtKB). Sometimes a shared peptide is also called *degenerated* in analogy to the so-called degeneracy of the genetic code, which could in principle encode 64 amino acids but does so for only 21 amino acids in most organisms.

Some of the most common reasons for the origin of shared peptides are shown in Figure 4.1. For a peptide of six amino acids, the probability for two sequences being equal is roughly $\left(\frac{1}{20}\right)^6 = 1.5625 \cdot 10^{-8}$ assuming only 20 amino acids and a uniform distribution of amino acids. Thus, though a peptide can be shared just by coincidentally having the same amino

acid sequence, mostly a biological background can be found. If no species specific database was used, there are often homologous proteins contained. These differ only by short stretches or even single amino acids and have for the biggest part identical sequences. Also truncated sequences or protein fragments, which are completely overlapped by another protein, are often found in bigger databases, like the reference proteomes of UniProtKB. For higher organisms, alternative splicing leads to proteins, which have at least partially the same peptides as related splice variants.

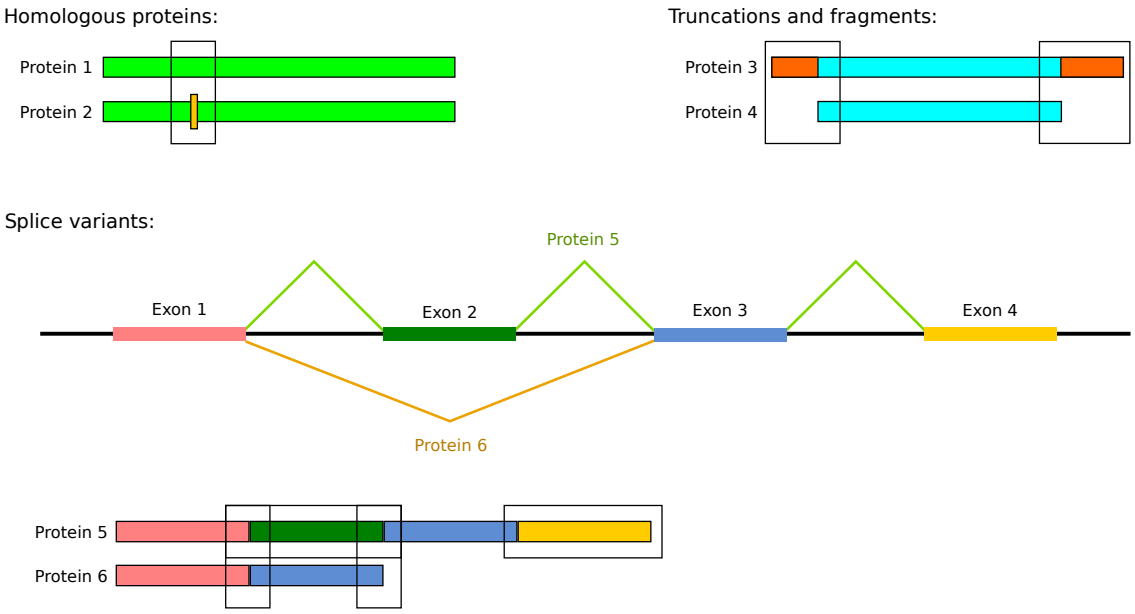


Figure 4.1: Some of the most common reasons for the origin of shared peptides. Regions with identical amino acid sequences have the same colour, those possibly containing unique peptides are highlighted with boxes. The first depicted reason is the occurrence of homologous proteins in the database, for example from different species. These usually are very similar and differ only in short sequences or single amino acids. The next example shows truncations or fragments of proteins, which are completely overlapped by another protein entry in the database. More complex are splice variants, which occur in higher species only. Here, alternative splicing of the mRNA can lead to proteins, which are concatenated of translations originating from different exons on the DNA. Under specific conditions during the maturation of the messenger RNA exons are spliced out and thus the resulting proteins do not contain the respective amino acid sequences. In the depicted example, there are four exons and *protein 5* contains all of the translated sequences, while *protein 6* contains only the sequences from exons 1 and 3.

4.1 *In Silico* Digestion of UniProtKB Databases

To get an overview on the number of unique peptides and of accessions, which contain at least one unique peptide sequence, the tryptic peptides of some UniProtKB databases are analysed here. For this, all protein sequences in the Swiss-Prot part of UniProtKB for *H. sapiens*

and *M. musculus* were *in silico* digested, using the regular expression [RK]|{P} as the default enzymatic cleavage site. This means, a cleavage is expected after each arginine (R) and lysine (K), except either is followed by a proline (P), which is the expected case when peptides are digested by Trypsin and the amino acid sequence is read in the usual way from the N- to C-terminus. As the digestion might not be perfect, the entries were digested allowing zero, one and two missed cleavages. All sequences, which were shorter than 6 AAs, which are usually neglected by SEs, as they are too small to be detected by the MS machines and are mostly shared, or longer than 45 AAs, which are too big for most MS measurements, were discarded. The same digestions were performed to the complete proteome sets of the two species and to the whole Swiss-Prot portion without any taxonomy restriction. For all these studies the UniProtKB release 2015_11 was used. A similar approach was performed in[74], though with older databases and without considering the missed cleavage sites. The results of the *in silico* digestion are shown in Table D.1. The table shows the number of accessions in the used database. Furthermore, the number of accessions with at least one unique peptide, the total number of peptides and the number of unique peptides are given for either zero, one and two allowed missed cleavages.

The analysis shows, that the number of accessions, peptides and their uniquenesses varies greatly, depending on the underlying database. The first obvious point is the varying number of total entries in the databases. The Ensembl information pages (accessed 21.12.2015) lists 20,313 protein-coding genes for human and 22,533 for mouse for the genome releases GRCh38 (human) respectively GRCm38 (mouse). The human numbers correlate roughly with the number of entries in the human Swiss-Prot, whereas for the mouse the evidence in this manually curated database is still missing for approximately 6,000 proteins. The complete proteome datasets, which additionally contain non-manually curated entries, is 2.9 times bigger for the mouse and 3.4 times bigger for the human database, which both is much bigger than the annotated numbers of genes. The complete proteome sets, though, are the recommended datasets, if a proteome wide analysis should be performed, as these contain entries for all proteins which are thought to be expressed in the respective organisms, including alternative variants and fragments. The complete Swiss-Prot with 549,832 accessions of 13,251 taxonomies (including different species' strains) should be rarely used, unless for pre-searches such as conducted in metaproteomic projects[82]. Nevertheless, the data is given in the table for comparison.

In the Swiss-Prot databases, between 96% and 98% of the peptides are unique, whereas only 54% – 55% of the peptides in the complete proteomes are unique. This decreased uniqueness is expected and can be explained due to the alternative sequences and fragments of protein entries in these databases, which generally share a mutual part of their sequences with the respective canonical entries but also have some unique parts. The uniqueness is generally increased by allowing missed cleavages, though this effect is with 0.5% – 2.5% very small. Interestingly, each analysed database has about 2.6 times more peptides in overall when allowing one missed

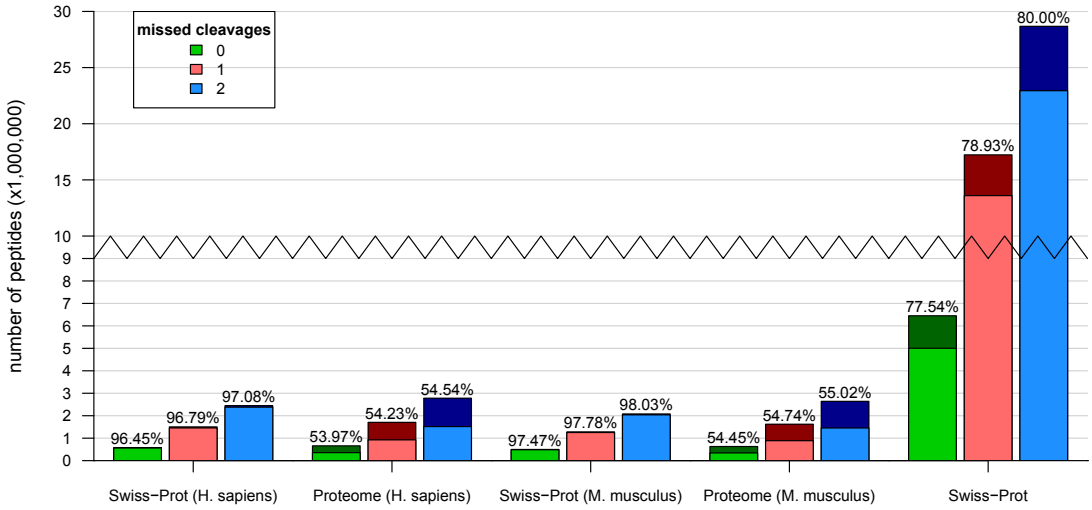


Figure 4.2: Fraction of unique tryptic peptides in common databases. This figure shows the results of an *in silico* digestion of often used databases from the UniProtKB. Shown are the results of human and mouse entries of Swiss-Prot, the human and mouse proteomes and the complete Swiss-Prot of the UniProtKB release 2015_11. The protein sequences of each database were digested using the tryptic regular expression [RK]|{P}, allowing 0, 1 and 2 missed cleavages and discarding peptides shorter than 6 or longer than 45 amino acids. The fraction of unique peptides is depicted in a light colour, the shared peptides in a dark colour. The percentage of unique peptides is given above the bars. The reference proteome databases contain a significantly higher fraction of shared peptides than the respective taxonomical Swiss-Prot databases. For both species, these fractions are similar, while for the digest of the complete Swiss-Prot database the fraction is between the taxonomic Swiss-Prot and reference proteomes. The actual numbers are given in the appendix in Table D.1.

cleavage and 4.2 times more when allowing two missed cleavages than the strict enzymatic rule.

The number and percentage of protein entries, which contain at least one unique peptide, varies not only greatly between the databases but is also greatly influenced by the allowed missed cleavages. In the taxonomically filtered Swiss-Prot databases almost each accession has a unique tryptical peptide (98.67% – 99.73%), independent of the number of allowed missing restriction sites. In the proteome sets, the percentage of entries with unique peptides is significantly increased, when increasing the number of allowed miscleavage sites. The largest increase is already achieved by allowing one missed site with 10% in the human and 8% in the mouse data, whereas allowing two missed sites increases the uniqueness by an additional approx. 1.5% in both species to 87.6% for the human and 80.6% for the mouse database. Though these percentages are much less than the almost 100% for the Swiss-Prot databases,

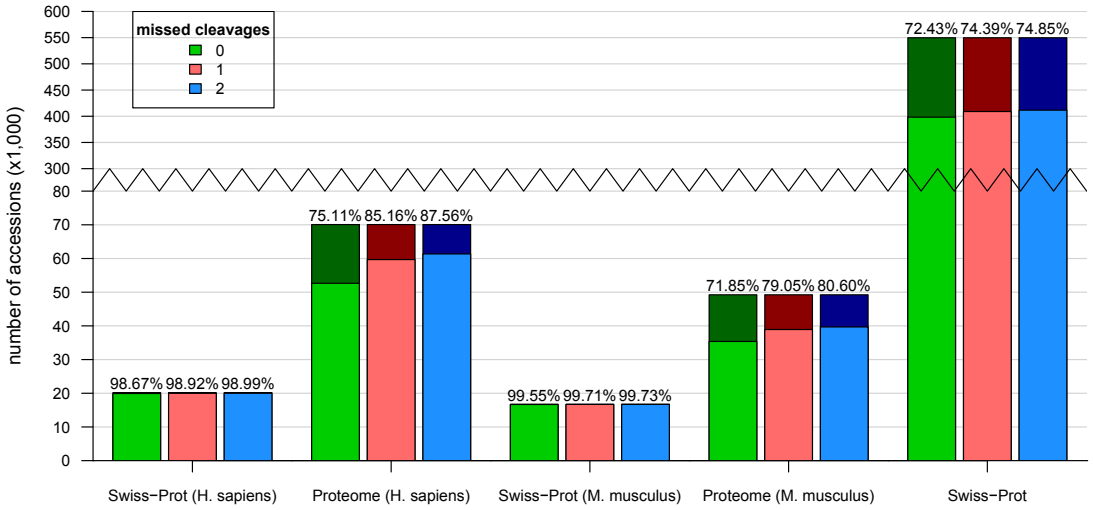


Figure 4.3: Fraction of accessions in common databases containing at least one unique peptide. In this figure, the fractions of accessions with at least one unique peptide (light colour) and accessions with shared peptides only (dark colour) are given. For this analysis the same databases and *in silico* digestion as in Figure 4.2 was used. The percentage of accessions containing at least one unique peptide is given above the bars. For the taxonomic databases the fraction of accessions without unique peptides is decreased by half in comparison to the fraction of shared peptides in Figure 4.2, though this is only 1%-2% in the Swiss-Prot databases. Interestingly, for the complete Swiss-Prot database, the fraction of accessions with unique peptides is decreased in comparison to the relative fractions on peptide level.

it is interesting to notice, that the uniqueness for the complete proteomes is on accession level 20%-30% higher than on peptide level. For the complete Swiss-Prot, the uniqueness on accession level is relatively constant with 72.4% – 74.9% and interestingly even slightly lower than on the respectively peptide level.

4.2 Peptide and Protein Uniqueness in Example Datasets

To compare these theoretical values with real data, two datasets from exemplary human and mouse samples were compared. The human samples consist of a tryptic digestion of lysed A549 cells, a human cancer cell line. These samples are measured in the MPC regularly as an in-house standard on all machines. The mouse samples originate from an immortalised murine myoblasts cell line, for which the publication is pending (for the sample preperation see Section 5.1.1). Both samples were measured on an LTQ Orbitrap Elite (Thermo Fisher Scientific), coupled to an UltiMate 3000 RSLC nano LC system (Dionex). The samples were measured

on a 2 h gradient and generated approx. 26,000 MS/MS spectra per run. These experiments can be considered as standard measurements, i.e. the results should reflect usual values for MS/MS experiments. For the human and mouse samples 20 MS/MS runs were analysed and the spectra identified using X!Tandem. To maintain a PSM FDR of 1%, concatenated target-decoy databases were created of the databases, which were analysed for the *in silico* digestion. For all samples, two missed cleavages were allowed.

Table 4.1: Table showing the uniqueness of identifications on an exemplary human dataset. The original data was generated by an Orbitrap LTQ Elite (Thermo Fisher Scientific), coupled to an UltiMate 3000 RSLC nano LC system (Dionex). Samples are from lysed A549 cells, a human cancer cell line. Identification was performed by X!Tandem against the respective target-decoy database of the given type (compare to Table D.1 and text) and the identifications are filtered on a 1% PSM FDR threshold. Up to two missed cleavages where allowed for the spectrum identification. Peptides were inferred directly from the sequences of PSMs not considering modifications or charge states. Proteins reflect the number of different accessions linked to the identified spectra, "unique" on protein level reflects the number of accessions, which have at least one identified peptide not shared by any other accession. The numbers are the total identifications, in brackets are the unique percentages of these identifications.

Database	**proteins (with unique peptide)**	**peptides (unique)**	**PSMs (unique)**
Swiss-Prot (*H. sapiens*)	3,244 (79.96%)	11,396 (91.43%)	15,079 (88.05%)
Proteome (*H. sapiens*)	9,045 (14.84%)	10,886 (38.70%)	14,299 (37.33%)
Swiss-Prot	18,287 (5.78%)	9,504 (24.82%)	12,810 (21.82%)

Table 4.2: Table showing the uniqueness of identifications on an exemplary mouse dataset. The samples in this table originate from an immortalised murine myoblasts cell line. The setting were the same as for the data in Table 4.1. The numbers are the total identifications, in brackets are the unique percentages of these identifications.

Database	**proteins (with unique peptide)**	**peptides (unique)**	**PSMs (unique)**
Swiss-Prot (*M. musculus*)	2,326 (85.91%)	11,056 (93.05%)	14,697 (90.31%)
Proteome (*M. musculus*)	5,261 (20.31%)	11,150 (39.94%)	14,703 (38.43%)
Swiss-Prot	16,040 (6.51%)	9,953 (27.19%)	13,617 (24.00%)

The average values regarding the uniqueness of identifications on PSM, peptide and protein level for the 20 runs of each sample are collected in the Tables 4.1 and 4.2 respectively. Comparing these data with the theoretical values in Table D.1 gives some important insights. The first point to consider is the fact, that PSMs are measured, and not peptides directly. As several PSMs may stand for the same peptide, there are generally less peptides than PSMs in a

sample. For the peptide inference in this analysis, only the sequence is considered, no modifications or charge states. The uniqueness on the PSM level is generally slightly (1.4% – 3%) lower than on peptide level. On the peptide level the differences to the theoretical digestions become apparent. For the taxonomical Swiss-Prot databases, the uniqueness is decreased by approximately 5% for both species, on the complete proteomes by approximately 16%. When searched against the complete Swiss-Prot, the uniqueness is decreased by more than 50%. This indicates, that many peptides with homologous entries in different species are actually identified by MS experiments. Also the missed cleavages must be considered: while increasing the number of allowed missed cleavages, also the percentage and number of theoretical unique peptides is increased, the biggest part of identified spectra have no missing cleavage site. The percentage of identified accessions, which have at least one uniquely identified peptide, is even further decreased from the theoretical values than the theoretical values of the peptides.

4.2.1 Conclusion and discussion

These analyses highlight the need for a protein inference on the collected data. Well curated databases like taxonomical parts of Swiss-Prot contain a high percentage of unique peptides. The actually identified peptides in real-life samples though do not reflect this percentage, but the number of identified unique peptides are usually lower than the theoretical. When searching for novel biomarkers in a sample, also the less curated databases, like the reference proteome sets, are favourable, which contain even less percentages of unique peptides and therefore increase the need for a protein inference, if the proteins are the molecules of interest in an LC-MS/MS experiment.

The reason for the decreased uniqueness in real life databases is not obvious and can only be guessed. The occurrence of too small (less than 6 AAs) or too big (more than 45 AAs) peptides in the databases can be ruled out, as these were filtered out in the analyses. One explanation could be, that for many of the shared identified peptides actually more than one protein was in the sample. If this was the case, than these peptides have a higher abundance than other peptides, which are not shared, and therefore are more often fragmented and recorded by MS/MS spectra. Also, MS/MS spectra originating from high abundant precursors get better identifications.

Chapter 5

Assessment of Protein Inference Methods

After explaining the necessity to perform a protein inference in a LC-MS/MS experiment in the prior chapters, this chapter presents different protein inference approaches and benchmarks their results. This includes widely used implementations as well as the methods implemented in PIA, which will be described in detail in the next chapter. The first part of this chapter contains a description of the subsequently analysed datasets. In Section 5.2, PIA and its application to complex datasets is assessed, before in Section 5.3 it is compared with other inference algorithms and an in-depth assessment of the results obtained from multiple datasets of varying complexity is performed.

The contents of this chapter are, partly literally, published in the articles "PIA: An Intuitive Protein Inference Engine with a Web-Based User Interface.", Uszkoreit et al. *J Proteome Res*. 2015 Jul 2;14(7):2988-97.[1] (mainly in Section 5.2) and "In-depth analysis of protein inference algorithms using multiple search engines and well-defined metrics.", Audain and Uszkoreit et al. *J Proteomics*. 2016 Aug 4;150:170-182.[2] (mainly in Section 5.3)
The figures depicted in this chapter are created after the original figures used in the stated publications.

5.1 Description of the Benchmark Datasets

To ideally benchmark a protein inference algorithm a dataset should be used, which contents at least on the protein level are completely known. Such a dataset is often called a "ground truth" or "gold standard" dataset. Furthermore, the dataset should be complex enough, that useful

analyses of different inference algorithms can be conducted. Creating and measuring such a dataset with LC-MS/MS methods though is cumbersome and expensive. The commonly used ISB 18 dataset[83] contains 18 highly purified proteins, of which the sequences are perfectly known. But even this dataset contains 93 identified contaminants like keratins and other proteins of human origin and several more proteins originating from nutrient solutions used to feed the hosts, which were used to express the proteins of interest. However, the number of 111 known proteins in the dataset proved to be too small to perform any useful assessment of protein inference: all proteins were identified by almost all tested algorithms, without returning any significant amount of false positives (data not shown here). The sample has furthermore no information about any protein ambiguities, which should be detected by any protein inference, like the existence of different isoforms.

For peptide level analyses there exist several complex datasets, for example a dataset containing one synthetic peptide for almost each human gene[84]. Also for quantitative analyses there are several spike-in datasets available as shown in[85]. Still, at the time of writing only two public accessible complex datasets claiming gold-standard-status could be found: the dataset created for the 2008 Proteome Informatics Research Group (iPRG) study (abbreviated as iPRG 2008 dataset) and the "Gold Standard of Protein Expression in Yeast" dataset described by Ramakrishnan et al.[86] ("yeast gold standard"). Besides these gold standards, the benchmark results of two further complex datasets will be discussed: one murine myoblast cell line sample prepared at the MPC ("mouse dataset") and one human lung cancer dataset downloaded from ProteomeXchange (PXD000603). All datasets will be further described in the following paragraphs.

5.1.1 Mouse Dataset

For the creation of the mouse dataset, cultured cells of a murine myoblast cell line were used. A complete description of these cells is not possible in this thesis, as the publication by a colleague is still pending. The cells were harvested and centrifuged for 5 min at 800 g. The cell pellet was resuspended in lysis buffer (3 mM Tris-HCl, 7 M urea, 2 M thiourea, pH 8.5), homogenised, and lysed via sonification (six times for 10 s, on ice). After centrifugation (15 min, 16,000 g), the supernatant was collected, and protein content was determined by Bradford protein assay. For the following tryptic in-solution digestion, 20 μg of sample was diluted in 50 mM ammoniumbicarbonate (pH 7.8) to a final volume of 100 μL, reduced by adding DTT, and alkylated with iodacetamide as described in[87]. After digestion, the peptide concentration was determined by amino acid analysis, and 200 ng of the peptide sample was subsequently analysed by a label-free mass spectrometry approach using an UltiMate 3000 RSLC nano LC system directly coupled to an LTQ Orbitrap Elite mass spectrometer (both Thermo Fisher Scientific, Dreieich, Germany).

For spectrum identification, an mzML file was created from a Thermo RAW file using the msConvertGUI of ProteoWizard[88], which was further converted by OpenMS into an MGF file (Mascot Generic Format, the de-facto standard for text based spectrum files used by almost all search engines). This MGF was searched against a decoy database of the mouse complete proteome set downloaded from UniProtKB on 26.11.2014 (44,467 entries). A shuffled decoy database was created using the DecoyDatabaseBuilder[67]. The used search engines were set to allow a parent mass tolerance of 5 ppm and fragment mass tolerance of 0.4 Da as well as one missed cleavage. Oxidation of M, acetylation of the protein N-terminus, Glu to pyro-Glu, and Gln to pyro-Glu were used as variable modifications, the latter three corresponding to the default and recommended settings for X!Tandem. Additionally, carbamidomethylation of C was used as a fixed modification, due to the sample preparation.

5.1.2 Gold Standard of Protein Expression in Yeast

The RAW data files for the yeast dataset were downloaded from `http://www.marcottelab.org/MSdata/Data_02`. The measured samples contain proteins of wild-type yeast, grown in rich medium and harvested in log phase. This dataset was published and analysed by Ramakrishnan et al[86]. Of the original 32 RAW files (eight different mass spectrometer settings with four SCX salt steps each) available, the four runs of the mass spectrometer with the highest number of spectra were used (070119-zlmudpit07-1). For these runs, the RAW files were converted to mzML using msConvertGUI and further processed to MGF files using OpenMS. The expressed proteins contained in the sample were identified by MS- and non-MS-based methods and are available as a reference set. This original reference set contains 4,265 protein entries, which needed refinement due to merged accessions in newer databases, as explained later. The complete search parameters will be given in the respective assessment description.

5.1.3 iPRG 2008 Dataset

The used MGF files and the provided concatenated target-decoy database, containing reversed sequences as decoys, were downloaded from the homepage of the Proteome Informatics Research Group (iPRG,[89]). These data were also provided for the Association of Biomolecular Resource Facilities (ABRF) iPRG 2008 study, which aimed to "assess the quality and consistency of protein reporting on a common data set", as stated on the study's slides. More details on the actual study design will be described in Section 5.2.3. For this study, mouse samples were trypsin-digested, and peptides were labelled by four-plex iTRAQ and fractionated via strong cation exchange (SCX) chromatography. The fractions were measured by LC-MS/MS on a 3200 QTrap (some fractions were measured multiple times with different exclusion lists), which resulted into 29 files. These data were analysed by members of the iPRG by a variety of search engines and protein inference tools. The results were used to create a list of protein

clusters that are detectable in the data. A protein cluster contains multiple database accessions, which share some peptide information (compare Section 3.5.2). For each cluster, the number of expected identifications was identified using the iPRG's members analyses. The expected number of reported protein groups in a cluster was set to the number returned by at least three of the six iPRG group members. Furthermore, the clusters were assigned to five different classes, of which only the first three classes were graded in the further assessment. Class 1 (16 clusters) contains "consensus multiple identifications", i.e. at least three members of the iPRG identified a minimum of two protein groups per cluster and each cluster was identified by at least one protein group by each iPRG member. Class 2 (11 clusters) contains "debatable multiple identifications", meaning most iPRG members identified at least one protein group for each cluster and for some clusters the expected number is set to two after discussion. Finally, class 3 (182 clusters) contains a "consensus single identification" per cluster: each cluster was identified by one protein group by each iPRG member. More information on this can be found on the iPRG's homepage.

For the peptide identification, a mass spectrometer specific precursor and fragment tolerance of 0.45 Da was set and one missed cleavage was allowed. For the fixed modifications, four-plex iTRAQ on K and N-termini as well as methylthio on C were used, due to sample preparation, and for the variable modifications oxidation of M was used.

5.1.4 Human PXD000603 Dataset

The PRIDE submission PXD000603 contains the data of a lung cancer study, analysed with an LTQ Orbitrap. Of this dataset, the RAW files of the lung cancer samples (LC1-LC12) were converted into the mzML format using ProteoWizard. For the sample preparation and measurement, please refer to the information in the PRIDE archive and the respective publication[90].

5.1.5 Human PXD001118 Dataset

This PRIDE submission contains HCD (higher energy collisional dissociation) fragmentation spectra, in contrast to all other datasets which have CID fragmentations. For the analyses only the MGF file containing this fragmentation type was evaluated. The measured sample originates from an enriched histone study. Thus, these samples are not as complex as the other datasets. For the sample preparation and measurement, please refer to the information in the PRIDE archive and the respective publication[91].

5.2 Assessment of PIA

To evaluate the reliability and to describe the behaviour of PIA, it was assessed on one real-life in-house dataset, of which the precise protein contents are not known, and on two public

datasets with knowledge of the protein contents. The in-house dataset is a label-free mass spectrometry analysis of a murine cell culture sample (see Section 5.1.1). The first dataset with known protein content is part of the Gold Standard of Protein Expression in Yeast (Section 5.1.2). The other dataset that also contains known proteins was produced for the iPRG 2008 study of the ABRF (Section 5.1.3). All three datasets were used to measure and compare the performances of the PIA algorithms using the common search engines Mascot (version 2.4.1), MS-GF+ (v9949), and X!Tandem (Sledgehammer, 2013.09.01.1). PIA intermediate XML files, containing compilations of the original SE's outputs (compare 6.2.1), were generated with various search engine's result files per dataset and used to generate protein group results with different PIA settings and filters.

All benchmarking datasets, including the plotted search results, and used KNIME workflows have been deposited to the ProteomeXchange Consortium via the PRIDE partner repository with the dataset identifiers PXD000790, PXD000792, and PXD000793.

5.2.1 Assessment of the Mouse Dataset

With this assessment the application of PIA on a current dataset was analysed. For this, the searches performed by Mascot, MS-GF+, and X!Tandem were first analysed separately and then through a combination of all searches. The numbers of identified protein groups using Spectrum Extractor (Section 6.2.4) as inference method are plotted against the protein FDR q-value in Figure 5.1. For Mascot searches, three protein inferences were performed using allowed PSM *FDR Score* values below 0.30, below 0.10, and below 0.01 (the latter value is recommended by us), respectively. Although decreasing the allowed FDR level also decreases the total number of reported protein groups, the number of target protein groups in the low-FDR range is increased, i.e., the very beginning of the list contains fewer false positives. This effect can be seen when comparing the Mascot plots with FDR ≤ 0.30 and FDR ≤ 0.10 on the q-value of 0.01: the latter reports more groups up to this threshold. This increase of reported high-quality proteins is observable only until a certain FDR level is reached, below which the number of reported proteins rapidly decreases, as can be seen in the plot of Figure 5.1 when allowing only PSMs up to an FDR below 0.01. Additionally, the number of groups identified when using only MS-GF+, X!Tandem, and PSMs with an FDR Score below 0.01 are plotted, which show equal trends even though there are different numbers of reported protein groups at given q-values. Finally, the number of reported groups when using the combination of all search engines and keeping the PSM FDR level (using the *Combined FDR Score*) at 0.01 exceeds the number of reported proteins for each single search engine at every q-value. This indicates that a combination of search engine results with PIA improves the number of true identifications in a list of protein groups.

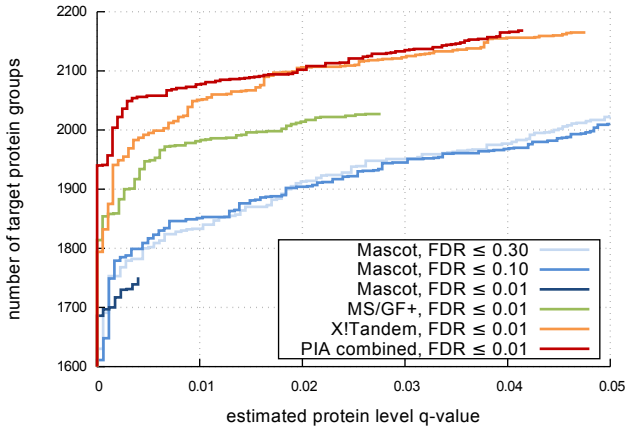


Figure 5.1: Performance of PIA on the mouse dataset. Plotted is a pseudo ROC curve of the number of target (in contrast to decoy) protein groups against the protein FDR q-values for protein inferences using the PSMs from three different search engines and Spectrum Extractor. The number of protein groups determined after a combination of search engine results with PIA exceeds the number of protein groups identified when using results of a single search engine at every q-value while using the same PSM FDR threshold. Although decreasing the allowed FDR level also decreases the total number of reported proteins, the number of protein groups in the low-FDR range is increased, i.e., the beginning of the protein list contains fewer false positives. This increase of reported high-quality proteins is observable only until a certain FDR level is reached, below which the number of reported proteins rapidly decreases, as plotted for the Mascot data (blue curves).

5.2.2 Assessment of the Yeast Gold Standard Dataset

For the performance measurement in this assessment, a shuffled decoy database created with the DecoyDatabaseBuilder of the protein database from the Saccharomyces Genome Database (SGD, `www.yeastgenome.org`, downloaded on 28.05.2014, 6,717 entries) was used for protein identification. As some of the entries in the reference set are no longer in the SGD database due to newer protein annotations, the reference set of proteins known to be in the sample was adjusted and finally contains 4258 accessions, 7 entries fewer than the original reference set. For the peptide identification a mass spectrometer specific precursor tolerance of 25 ppm, a fragment tolerance of 0.5 Da, one missed cleavage and the variable modifications for oxidation of M and protein N-terminal acetylation were allowed.

The performance of PIA using the Spectrum Extractor and Occam's Razor inference, with the need for one unique peptide per reported protein group, was assessed for each search engine and the combination of search engines on this dataset. As for this dataset the proteins contained in the sample are known, the local FDR and q-value of the ranked protein results can be calculated using the proteins contained in the reference set as true positive identifications and all other identifications as false positives. With these values, a pseudo ROC curve plotting the number of true positives against the corresponding q-values depicts the quality of the

results. In Figure 5.2, the curves for the combination by PIA and the X!Tandem results alone with at least one unique peptide per protein group are shown.

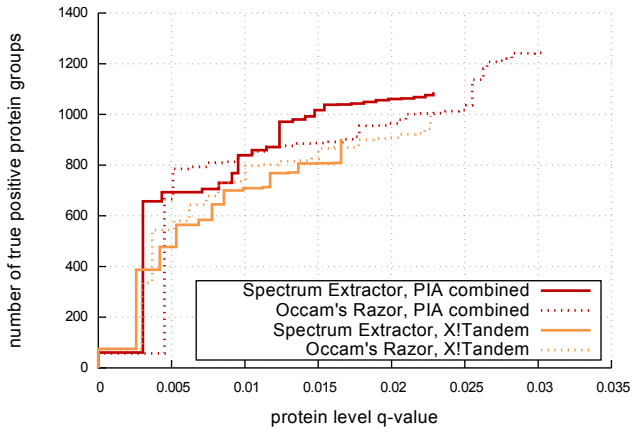


Figure 5.2: Performance of PIA on the yeast gold standard dataset. For this dataset, the expected identifications are known, which allows the number of true positive identifications to be plotted against the q-value in a pseudo ROC curve. Plots are shown for protein inferences run with Spectrum Extractor and Occam's Razor for a combination of search engines and the usage of X!Tandem results only. Generally, Spectrum Extractor outperforms Occam's Razor in the very high confidence regions, but it also tends to report fewer protein groups in the overall perspective, if no protein level FDR threshold is set.

Although the general behaviour is similar, it is interesting to note that, for the assessed dataset, Spectrum Extractor usually yields better performance in the very low q-value regions but the overall number of reported proteins is higher with Occam's Razor, if no protein level FDR threshold is set. Although these observations are dataset dependent, the data show overall good results for the analysed inference algorithms and do not make many false reports, as the plotted curves all stop before a q-value of 0.035. For all analysed settings, the protein group containing the accessions YLR227W-B and YPR158C-D was identified at around rank 60, although it is not in the reference set. The quality of the identification, though, indicates that it is a false negative, missing in the reference set. Usually, it can be said that Spectrum Extractor reports fewer proteins because it uses a spectrum only for exactly one peptide, if the search engine reports more than one PSM per spectrum. Again, the combination of search results by PIA yields more highly evident protein groups, like in the assessment of the mouse dataset.

5.2.3 Assessment of the iPRG 2008 Dataset

In this dataset, the expected number of identified protein groups per cluster was calculated by the ABRF group members. For classes 1, 2, and 3, these numbers are assessed and result in a total maximum of 258 true positive (TP) identifications. A false positive (FP) identification

has too many identifications per cluster, whereas a false negative (FN) identification has too few identifications.

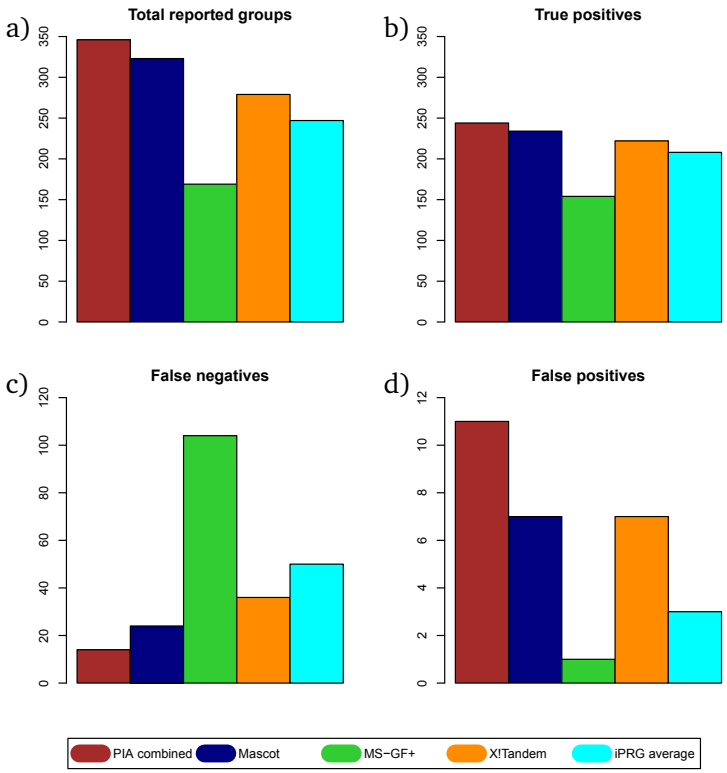


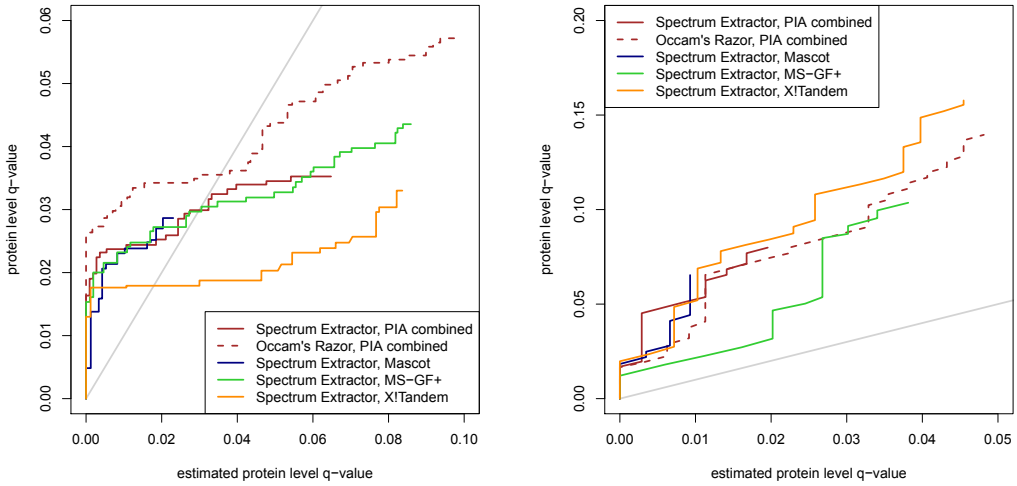
Figure 5.3: Performance of PIA on the iPRG 2008 dataset. (note that the y axes differ). Number of (a) total reported proteins, (b) true positives, (c) false negatives, and (d) false positives for the inferred proteins generated by PIA for either a combination of the search engine results or each search engine alone as well as the average result of the iPRG 2008 participants. For the PIA analysis, Spectrum Extractor with a FDR threshold of 0.01 was used on the PSM and protein levels. It can be seen that a combination by PIA outperforms the average iPRG results except when using the MS-GF+ results alone.

In Figure 5.3, the results of the (a) totally reported, (b) TP, (c) FN, and (d) FP identifications are shown for protein inferences conducted by PIA in comparison to the average outcome of the iPRG 2008 study. With PIA, the PSMs with an FDR below 0.01 for each search engine alone and in combination were inferred to a protein group list using Spectrum Extractor; for the comparison, only protein groups with an FDR below 0.01 were used. The combination of search results yields the highest number of reported proteins and also outperforms most of the iPRG study participants; only 4 of the 23 participants reported more proteins. More interesting is that the number of true positives is much higher in the report from the combination than the single search engines, which is surpassed by only six iPRG participants. Also the false negative rates with the assessed PIA settings are better than the average iPRG participant's results. An

exception is the MS-GF+ search, which reports the fewest total protein groups and therefore also has the highest number of false negatives, whereas the PIA combination is outperformed by only six of the iPRG participants. The relatively high number of false positives in all runs except the MS-GF+ run corresponds mainly to clusters, which are also stated as dubious in the slides of the ABRF study. For these, many ABRF group members and study participants found more than the expected number of distinguishable detectable isoforms. The number of false positives can be decreased by stricter inference parameters, such as the need to have at least one unique peptide per protein, although stricter settings also decrease the total number of reported proteins and thus true positives.

5.2.4 Comparison of Estimated and True Protein Level FDR

For the datasets with known content, plots of the target decoy estimated protein level q-values against the (claimed) true protein level q-values are shown in Figure 5.4. These plots show significant differences between the datasets. For the iPRG 2008 dataset, the estimated error is consistently much lower than the actual value, although the ratio goes down with the number of reported proteins. For the yeast gold standard dataset, the actual values are underestimated on the top of the protein list and overestimated after a certain value (for the combination of all search engines at an FDR of 0.03). These differences are presumably due to the underlying ways of how the actual protein content was measured. For the iPRG 2008 dataset, prior search results of the same actual MS data were used; thus, identifying more proteins than those that are claimed to be valid is more probable with different search engines. For determination of the yeast dataset's content, other technologies were also used, which allows to create a more complete compilation of the contained proteins. For an in-depth analysis of the estimation between true and estimated protein q-values of protein inference algorithms, more datasets of complex protein mixtures having exactly known content would be needed, which are not available at the time of writing this thesis.



(a) Yeast Gold Dataset **(b)** iPRG 2008 Dataset

Figure 5.4: The "true" protein level q-value plotted against the estimated q-value. For the datasets with known content, the true FDR and q-value can be calculated allowing only the proteins of the reference sets to be true positives. These values are plotted against the estimated q-value using the target decoy approach. For the yeast dataset the estimated q-values are too low at the top of the list until about 0.02, after which the estimation is always too high. For the iPRG 2008 dataset, the values of the target decoy approach are always underestimating the true values.

5.3 Assessment of Protein Inference Algorithms using a Workflow Framework and Well-Defined Metrics

In the following sections, we evaluate and benchmark five common tools for protein inference: ProteinProphet[23], MSBayesPro[92,93], ProteinLP[94], Fido[95] and PIA. To achieve this, the three search engines Mascot, X!Tandem, MS-GF+ and their combinations were used with every protein inference tool. We implemented a workflow in the highly customisable KNIME[96] workflow environment using a series of OpenMS[97] nodes and several new workflow nodes (available at `https://github.com/KNIME-OMICS`) to study all combinations of these search algorithms and inference algorithms. We provide different metrics to benchmark the algorithms under study. Amongst others, the numbers of reported proteins, peptides per protein and uniquely reported proteins per inference method are used to assess the performance of each inference method. Four datasets of different complexities and from different species were employed to evaluate the performance of protein inference algorithms: the yeast gold standard dataset (Section 5.1.2), the iPRG 2008 dataset (Section 5.1.3) and the two PRIDE datasets PXD000603 (Section 5.1.4) and PXD001118 (Section 5.1.5).

5.3.1 Description of the Analysed Inference Methods

PIA - Protein Inference Algorithms

PIA's algorithms and background information will be discussed in detail in Chapter 6. Therefore, only the applied settings will be explained here. In this analysis, we employed the Spectrum Extractor for protein inference. In short, this algorithm assigns to each spectrum only the peptide, which increases the total probability or score of the according protein. This is only meaningful, if a spectrum was assigned to more than one peptide by the search engines. If this was not the case, it basically behaves like Occam's Razor and reports the smallest set of protein groups, which explain all PSMs, and therefore can be classified as a parsimonious approach.

Fido

Fido performs a fast Bayesian inference to report protein groups, given peptide probabilities. Its speed is achieved by three graph-transforming procedures: partitioning, clustering and pruning for computing the posterior probabilities of proteins. For each connected component of peptides and accessions Fido collapses protein nodes that are connected to identical sets of peptides and prunes spectral nodes (with user specified parameters), which results in splitting of the connected components[95,98]. Fido and the underlying generative (Bayesian) model relies on reasonable probabilities for the observed peptides, which are besides the three model parameters (gamma, alpha and beta) the only input to the algorithm. A parameter estimation, employing and expectation-maximization (EM) algorithm, can be conducted to yield the best parameter setting. This, though, was not performed in this study but the recommended (0.5, 0.1 and 0.01 for gamma, alpha and beta respectively) parameters were used. This is due to the case, that Fido did not perform well on the EM algorithm, as explained later.

ProteinProphet

ProteinProphet is one of the most widely used protein inference algorithms in proteomics. This is mainly due to its implementation into the Trans-Proteomic Pipeline (TPP,[99]), which allows easy processing of MS/MS data from RAW files, identification with different search engines and peptide validation/combination - using iProphet[70] - and finally the generation of protein lists with ProteinProphet. ProteinProphet takes a pepXML file as input that contains peptides with associated probability scores. Different peptide identifications corresponding to the same protein are combined together to estimate the probability that their corresponding protein is present in the sample. Finally, a protein grouping is employed adjusting the individual peptide probabilities, making the approach more discriminative.

ProteinLP

The ProteinLP model works with the joint probability that both a protein and its constituent peptides are present in the sample. To obtain a linear model, a mathematical transformation of this joint probability is used. The marginal probability of a peptide being present is expressed as a formula in terms of the linear combination of these variables. The model assumes that the marginal probability of each identified peptide being present is known, then the protein inference problem is formulated: the algorithm tries to find a minimal set of proteins while peptide probabilities should be as close to its known value as possible.

MSBayesPro

MSBayesPro is a Bayesian protein inference algorithm for LC-MS/MS proteomics experiments. Besides peptide probabilities derived from the spectrum scoring it also incorporates "peptide detectabilities" in its probabilistic model. A non-parametric model for estimating protein probabilities is used. To estimate posterior protein probabilities, MSBayesPro requires peptide identification probabilities and a set of peptide detectabilities, which makes the detectability an important distinguishing feature of MSBayesPro.

5.3.2 Benchmark Workflow

The presented protein inference comparison workflow is based on KNIME[96] and OpenMS[97]. We made use of the existing OpenMS nodes, but we also implemented additional nodes for some of the analysed inference tools. The developed workflow can be split into seven different steps (Figure 5.5). The first step (A) configures basic variables like the regular expression to identify decoys in the FASTA protein database and the allowed FDR q-value threshold. Also, if a gold-standard dataset is analysed, the reference protein list is loaded, which contains the set of accessions known to be in the sample. Step (B) performs conversion to mzML, optional peak centroiding for spectra recorded in profile mode, and removal of MS1 spectra. The remaining tandem spectra are searched in step (C) using three different search engines (X!Tandem, Mascot, and MS-GF+), employing the adapter nodes provided by OpenMS. Furthermore, the results are filtered for peptides with a minimum length of seven amino acids and exported to idXML files, OpenMS's internal storage format, for further processing. In step (D) all possible combinations for the results of the three search engines are created. Peptide posterior error probabilities (PEPs) are calculated with the `IDPosteriorErrorProbability` tool, which is a standalone OpenMS node used for estimating the probability of peptide hits to be incorrectly assigned[100]. For the assessment of combined search engine results, identifications are combined using the `Consensus ID`[28] incorporated in OpenMS with the PEPMatrix algorithm. After calculating the PSM FDR using the target decoy information, all peptides with FDR q-

value > 0.01 are filtered out and no longer considered in the analyses. To evaluate the FDR on the protein level later on, the target and decoy PSMs below the 0.01 FDR q-value threshold are passed together to all protein inferences.

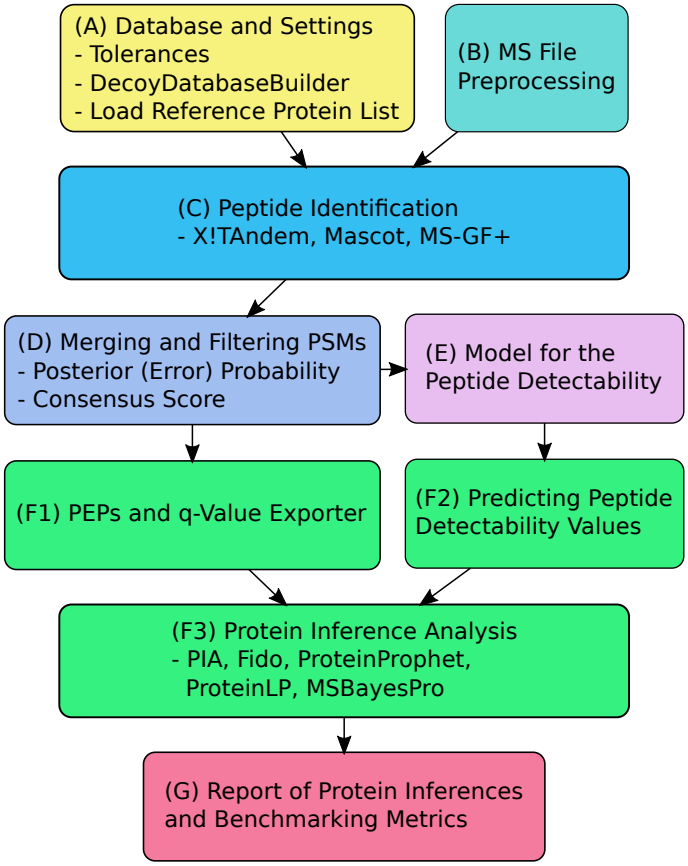


Figure 5.5: Simplified representation of the workflow used for the peptide identification and protein inference in KNIME. As input of the workflow, raw MS data in mzML format is used; the output consists of graphs and tables, as well as a complete report of the analysed protein inferences. This workflow can be split into seven different stages, A-G. (A) Settings and database, import of protein knowledge for ground truth datasets, (B) spectrum file pre-processing, (C) peptide identification, (D) merging of PSMs (E) model creation for peptide detectability, (F) protein inferences, (G) calculating tables and graphs of the inferences.

Since MSBayesPro requires a peptide detectability additional to the probability during its inference process, we compute a detectability model of all results in step (E) using the OpenMS node `PTModel` [101]. The `IDFilter` node was used to get the high scoring identifications (500 distinct peptides or at least one fourth of all available peptides) to train a model with `PTModel`. Additionally, we provide a subset of the PSMs for low-confidence peptides (i.e. those 500 peptides with lowest identification scores/probabilities or the lowest scoring fourth of all available peptides) as training input to the model. In the next step (F) of the workflow the final list of peptides with the corresponding probability and detectability values are passed

to the PIA, Fido, ProteinProphet, ProteinLP and MSBayesPro nodes to generate protein lists. PIA used the SpectrumExtractor algorithm with the recommended settings (using the best PSM FDR Score per peptide as basis for protein score with multiplicative protein scoring). For Fido the recommended parameters for gamma, alpha and beta (0.5, 0.1 and 0.01) were used for each run. ProteinProphet takes a pepXML file as input that contains peptides with associated probability scores. The pepXML files were refined using ProteinProphet's xinteract to correct the decoy annotations and FASTA file connection, which was lost in the idXML format. Afterwards, ProteinProphet was executed without any parameters except MINPROB0.05 to include only peptides with probabilities of at least 5% into the inference. For this, it is important to remember, that the PSMs were filtered on an FDR of 0.01 beforehand, such that for most runs of ProteinProphet all PSMs have probabilities above 5%. MSBayesPro incorporates besides peptide probabilities derived from the spectrum scoring also the peptide detectability from the PTModel node in the probabilistic model, while ProteinLP does not need any further parameters except the peptide probabilities.

The FDR q-values on protein level were calculated, based on the target-decoy approach, to control the number of false identifications[102]. Finally, in (G) the inference reports are generated, including both numbers and graphs explained in Section 5.3.3. For each search engine combination the number of FDR filtered PSMs is reported to give an overview of the identification step. For all protein level metrics besides the pseudo-ROC curves, the analyses were restricted to the high confidence proteins with a q-value below 0.01, or equivalently 1%.

5.3.3 Benchmark Metrics

Benchmarking requires both a high-quality dataset and defined metrics to evaluate the improvements and potential pitfalls for the benchmarked tools[103]. We used a set of metrics based in previous studies to benchmark protein inference algorithms and tools[104,105], and added some additional metrics. An overview of the metrics assessed in this analysis is given in Table .

The number of FDR filtered protein groups represents the first intuitive metric for a quick overview of the inference performance (see Figure 5.11). A protein (ambiguity) group is an indistinguishable entity reported by an algorithms (compare Section 3.5.2). In such groups, the sets of peptides overlap perfectly in the set of proteins which are reported for the respective group. In addition, we studied the overlap of protein groups between all inference algorithms since the number of protein groups reported may identical and yet their identities may be different. The proportions of mutually reported groups were calculated to gain deeper insight into the consensus of the reported groups (see Figure 5.10 and E.3). We additionally created Venn diagrams to visualise the overlap of reported protein groups in a widely known and intuitive way (see Figure E.2).

We studied the behaviour of the number of reported protein groups along FDR q-values (0-5%) on protein level using pseudo-ROC curves (Figure E.1). We also highlighted the true positive protein groups for the assessed ground truth datasets, the yeast and the iPRG 2008 dataset. Furthermore, we reported the number of identified peptides per protein groups (see Figure E.4). For all the current metrics we use the protein groups and protein sub-groups if the inference algorithm reports them by default, which is the case for Fido, ProteinLP and MS-BayesPro. A protein sub-group in this analysis is a protein group whose peptides are completely explained by another protein group, like defined in the prior chapter.

Some of these metrics cannot be taken as absolutely distinguishing numeric values for comparison and using either higher or lower values as better results between datasets. This is obviously true for the plots, which need human interpretation. But also a higher number of reported protein groups is not per se a better quality feature, if these numbers are increased by false positives, as discussed in Section 5.3.7. These metrics were rather chosen to highlight the differences and similarities between inference algorithms on a range of datasets and using varying protein databases.

Table 5.1: Assessed metrics of this analysis and a short explanation.

Metric	**Explanation**
Number of target protein groups	The number of target protein groups under a given FDR. This can give only a first hint, whether an inference method was comparable to the other methods, unless used with true-positives of a ground truth dataset.
Pseudo-ROC curves: number of target groups against FDR q-value	Can be used to visual inspect the trends of inferences and compare, which inference yields most under a given FDR. Can also be used with true positives only on a ground truth dataset.
Area under the curves (AUC) of pseudo-ROCs	The AUC indicates, not only whether high number of reported (true positive) targets are reached, but also whether this happens at low FDR values. Normalising this on the total amount of proteins allows inter-dataset comparison.
Numbers of TP, FP, TN and FN	Only applicable for ground truth datasets. This allows also the comparison of precision and recall between the methods.
Overlap of reported groups	Shows the consensus of analysed methods. We calculated also the fractions of groups, that are identified by all, 4, 3, 2 and 1 more method(s) or by the given method uniquely.
Number of peptides per protein group	Gives a quality measure for the reported protein groups. This can either be given in a heatmap-like plot or as single value giving the number or fraction of groups with more than X peptides.

5.3.4 Benchmarked Databases and Search Parameters

The tandem mass spectra of the four datasets were searched against appropriate protein sequence databases using the target decoy approach (TDA, compare Section 3.2) with three different search engines: X!Tandem (version Sledgehammer 2013.09.01.1), MS-GF+ (version beta v10089) and Mascot (version 2.5). The first two tools are not the most recent versions, but the ones shipped with the version 2.0 of OpenMS, which was the most current stable release at the time of writing and used for all employed OpenMS tools and utilities.

Table 5.2: Databases used in the study, downloaded from UniProtKB on 10.02.2016 (release version 2016_01). For the Yeast Gold Standard dataset, the used databases are almost equally complex, the Swiss-Prot and proteome being identical, while the mouse and human databases differ significantly.

Database	Dataset	Species	Number of target proteins
iPRG2008 provided	iPRG 2008	*M. musculus*	53,883
Mouse Swiss-Prot			16,761
Mouse UniProt Proteome			50,189
Mouse Proteome with isoforms			58,239
Yeast Swiss-Prot	Yeast Gold Standard	*S. cerevisiae* (strain ATCC 204508 / S288c)	6,721
Yeast UniProt Proteome			6,721
Yeast Proteome with isoforms			6,743
Human Swiss-Prot	PXD000603 and PXD001118	*H. sapiens*	20,187
Human UniProt Proteome			69,986
Human Proteome with isoforms			91,923

To analyse the influence of database complexities in protein inference, each dataset was searched against three different taxonomy databases: (i) UniProtKB/Swiss-Prot, (ii) UniProt reference proteome, and (iii) UniProt reference proteome containing known isoforms for each gene, in contrast to the first two, which contain only the longest isoform for each gene (Table 5.2). Only two databases were analysed for the yeast dataset, because the Swiss-Prot and the reference proteome sets are identical in this case. The iPRG 2008 dataset was additionally identified using the provided mouse database. The decoy databases were created with the DecoyDatabaseBuilder by shuffling the protein sequences and appending them to the target database creating a concatenated target-decoy database. An exception was the provided iPRG 2008 database, where the provided target-decoy sequences with reversed decoys were used. We used the same search parameters wherever possible for each search engine, for the individual settings of each dataset see Table 5.3. For the digestion of proteins to peptides a fully tryptic digestion was selected, except for the histone dataset, where the cleavage at lysine was masked by a fixed modification and therefore neglected. The workflows, search engine results and all of the final results are available via ProteomeXchange and GitHub.

Table 5.3: The datasets and search engine settings used for the comparison of inference algorithms. All settings were chosen according to the description of the dataset in the respective publication or repository.

Dataset	Instrument and Fragmentation	Peptide / Fragment Tolerances	Modifications and Enzyme's cleavage regular expression
iPRG 2008	3200 QTRAP CID	0.45 Da 0.45 Da	**Fixed:** iTRAQ 4-plex (K, N), Methylation (C) **Variable:** Oxidation (M) **Cleavage:** [KR]|{P}
Yeast Gold Dataset	LTQ Orbitrap CID	25 ppm 0.5 Da	**Variable:** Oxidation (M) **Cleavage:** [KR]|{P}
PXD000603	LTQ Orbitrap XL CID	10 ppm 0.8 Da	**Fixed:** Carbamidomethyl (C) **Variable:** Oxidation (M) **Cleavage:** [KR]|{P}
PXD001118	LTQ Orbitrap Velos HCD	10 ppm 0.02 Da	**Fixed:** Propionyl (N-Term and K) **Cleavage:** [R]|{P} (K is blocked by modification)

5.3.5 Combination of Search Engine Results on PSM Level

Running the aforementioned workflow, we analysed 420 protein lists due to the combination of the three different search engines, five inference tools and four datasets using ten different databases. We analysed the number of FDR filtered PSMs for each single search engine and their combinations before performing any protein inference evaluation. The benefit of combining search engine results for spectrum identification has already been shown extensively in other publications ([106], [27], and Section 3.2.3). It is generally accepted that search engines in combination yield more valid PSMs, especially in low-resolution fragment ion measurements.

Figure 5.6 shows exemplary the numbers of PSMs for the PXD000603 dataset matched against the human proteome database with isoforms. X!Tandem and MS-GF+ identified more PSMs than Mascot in almost all combination; the only exception was in the Swiss-Prot PXD000603 run, where X!Tandem was slightly outperformed by Mascot, and the UniProt proteome PXD001118 run. In the latter MS-GF+ alone was performing surprisingly suboptimal (X!Tandem reported 3.4, Mascot 2.7 times as many PSMs), due to relatively high-ranking decoy PSMs. The second biggest discrepancy between two single search engines was when the PXD000603 dataset was searched against the proteome database with isoforms, where MS-GF+ reported 1.55 times as many PSMs as Mascot. The average ratio between the lowest and highest single search engine was 1.62 (1.44 excluding the two prior mentioned outliers). Each combination of two search engines returned more than the respective single SEs. The combination of all three engines yielded the most PSMs for each dataset, increasing the report of the best single result by 90% on average and ranging from 17% (iPRG 2008 dataset with proteome database) to 173% (UniProt proteome database on the PXD001118 dataset). For

all analyses in this work it must be considered that we inspected only the *Consensus ID* with the PEPMatrix algorithm provided by OpenMS for the combination of PSM results and the posterior error probabilities (PEPs) calculated by *IDPosteriorErrorProbability*.

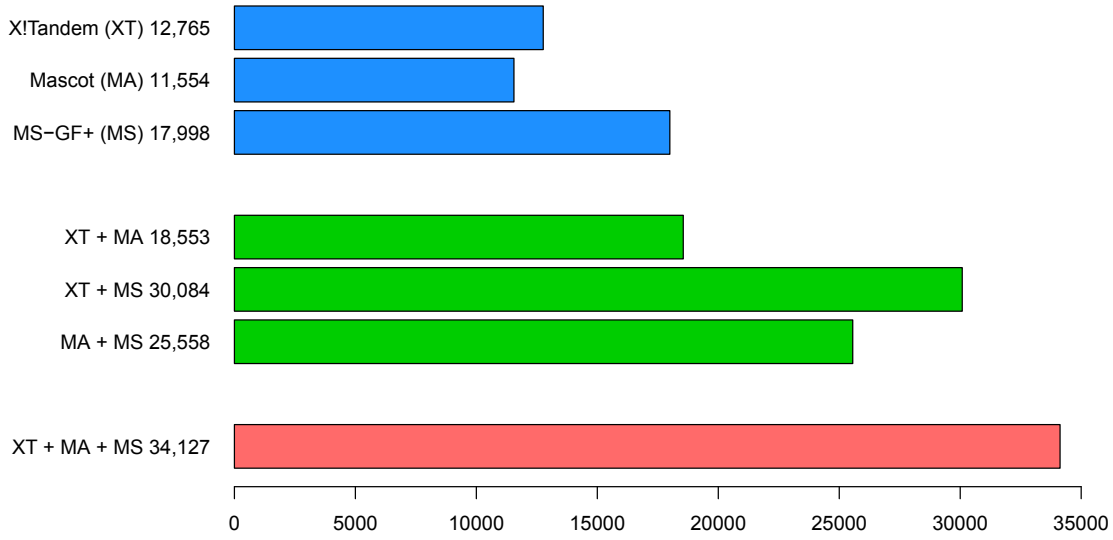


Figure 5.6: Number of PSMs reported by the single search engines and their combinations for the PXD000603 dataset. The spectra were matched using the specified search engine against the human proteome database with isoforms and combined using *Consensus ID*. As expected, each combination of search engines yields more FDR 1% valid PSMs than the respective search engines alone, also the combination of all three together yields more than each combination of two search engines.

5.3.6 General Assessment of the Protein Inference Algorithms

Figure 5.7 shows pseudo-ROC curves for the number of reported target protein groups against the local protein FDR q-values for the PXD000603 dataset using the combination of all three search-engines and the respective Swiss-Prot database (similar plots for all datasets are shown in Figure E.1). The overall number of reported protein groups under a certain q-value varies slightly between the different inference algorithms, while most algorithms follow the same general trend.

When looking at the actual numbers, Fido outperforms the other inference algorithms at 1% FDR q-value for the more complex datasets yeast and PXD000603 with 5.8% respectively 0.2% more protein groups. However, all other approaches except MSBayesPro outperform Fido significantly on the iPRG 2008 dataset. The main reason in this particular case is the highly unbalanced composition of target (34,127) versus decoy (332) PSMs. This resulted in much larger groups for target proteins reported by Fido, leading to reduced posterior probabilities of them, eventually boosting the ranks and therefore the q-values of decoys. The MSBayesPro

results show significantly fewer reported protein groups than all other approaches on each q-value. Additionally to the normal Fido results, the protein Groups were processed to not contain any sub-groups of other reported groups. These results are labelled as "Fido w/o Subgroups" of "FidoNoSubs" in the plots. These plots show a small decrease in the number of reported groups, though the general trend is the same as for the unprocessed results.

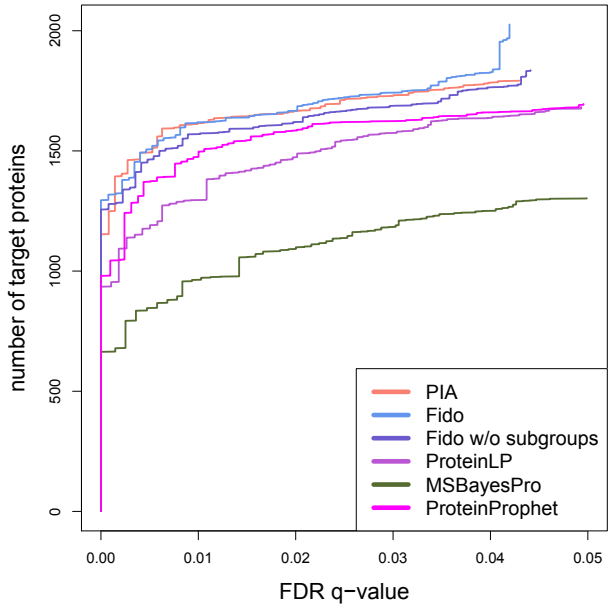
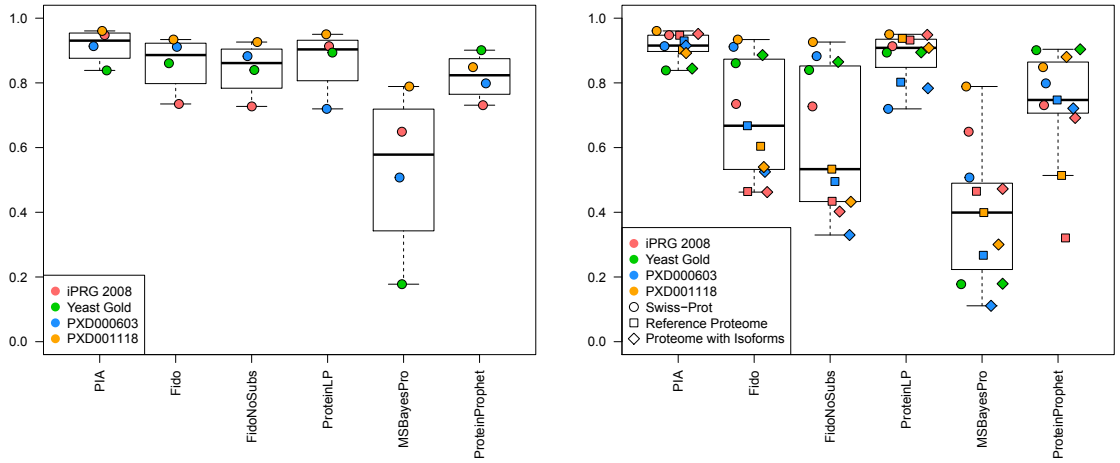


Figure 5.7: Pseudo-ROC curves showing the number of reported protein groups against the FDR q-value for the PXD000603 datasets using the Swiss-Prot database and the combination of all three search engines. The plots indicate that the main trend is similar for all inference algorithms. On this dataset, ProteinLP performs a little bit worse than the other algorithms, while MSBayesPro performs significantly poorer than all other methods on all datasets (compare Figure E.1).

The pseudo-ROC curves highlight additionally to the raw numbers of reported protein groups also, whether a high value is reached early on the q-value scale. To get a value for the comparison of this between algorithms, that does not need a human inspection, the area under the curve (AUC) of the pseudo-ROC curves was calculated. For this, only the FDR q-values up to 1% were analysed, as this is usually used as the cutoff in current analyses. To make a comparison between the datasets possible, the number of protein groups were normalised on the highest reported value for the given set at the 1% FDR threshold. Thus, the calculated AUC for each run and inference is a value between 0 and 1. Boxplots for the AUCs from the merge of all search engines are given in Figure 5.8, Figure 5.8a shows the data for the Swiss-Prot databases only, while Figure 5.8b incorporates all analysed databases.

The plot of the Swiss-Prot databases further confirms the findings of the pseudo-ROC curves alone: most algorithms perform similar, while MSBayesPro performs poorer on all datasets,



(a) Swiss-Prot only **(b)** all databases

Figure 5.8: Boxplots of the area under the curve (AUC) values for all datasets matched against (a) the Swiss-Prot and (b) all databases. The AUCs were calculated up to 1% FDR on protein group level and normalised to the maximal number of reported groups per dataset and database combination. The plots show, that PIA is the most robust algorithm when changing the complexity of the used database. Fido, on the other hand, is very dependent on the database complexity. Furthermore, MSBAyesPro performed very poorly, independent of the dataset or database.

while the actual difference to the other methods varies greatly. Furthermore it becomes obvious, that PIA performs best on all datasets, except for the yeast dataset. When comparing all analysed datasets and databases (Figure 5.8b), another trend becomes apparent. For the least complex yeast dataset, the variance is relatively small for all methods and as the analysed databases for this dataset are differing by only 22 protein entries, the compared results are relatively equal. On the other datasets, where the databases differ up to 4.5 times in the number of entries, not only the number of reported 1% FDR valid protein groups differ, but also the ratios of these between the analysed algorithms. Especially Fido's results are strongly dependent on the database complexity, but also the variance of ProteinProphet over all databases is increased, as compared to Swiss-Prot alone. On the other hand, the values for PIA are over all combinations the most stable. This indicates, that PIA constantly reports a large fraction of possible protein groups up to the 1% hreshold earlier than the other algorithms, even if some might overall report more groups up to this level.

5.3.7 Analysis of the Ground Truth Datasets

The number of reported protein groups for a given threshold is a basic metric to evaluate the performance of a given inference algorithm. However, it should be complemented with other metrics to label a protein inference superior to any other. In fact, it is more relevant to see

whether the true protein groups are reported. There are only few publicly available datasets containing ground truth data for peptides[84] and only relatively small protein datasets[83]. Of both analysed ground truth datasets, only the yeast gold standard can be considered as a complex mixture for current high throughput MS, identifying several thousand proteins per MS run. In the iPRG 2008, on the other hand, only several hundred proteins can be detected.

We used three Venn diagrams for the reference set using Swiss-Prot database to examine the content of correctly identified proteins in the yeast dataset (Figure E.2). We consider a protein group as true positive if it contains at least one accession of the reference set of accessions, which are known to be in the sample. Figure E.2a shows all the proteins identified in the yeast dataset by every inference algorithm without discrimination of true and false positives regarding the reference set. The yeast reference set contains 4,253 protein entries that are known to be in the sample (also validated by 2D-DIGE).

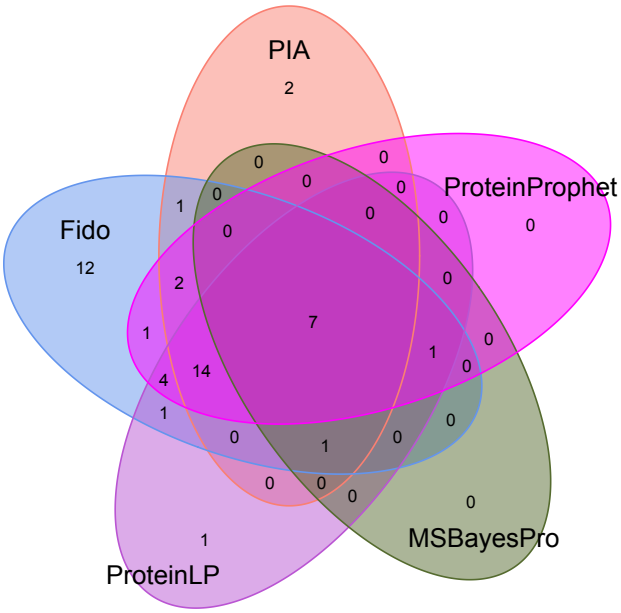


Figure 5.9: Venn diagrams showing the number of reported protein for each inference algorithm at 1% FDR for the ground truth yeast dataset using the Swiss-Prot database Number of reported proteins that are not in the reference set and thus false positives. For the overlap of all reported proteins and the false negatives, see Figure E.2.While Fido reports many false positives uniquely, it can be seen that the overlap of all algorithms excluding MSBayesPro is the largest fraction, and also the total overlap is the third largest fraction. This might indicate, that the identification of the real true positives for this dataset might be incomplete, which is a big problem for all ground truth datasets.

The number of reference proteins reported by PIA, Fido, ProteinLP, MSBayesPro, and ProteinProphet were 1,095, 1,193, 1,149, 215 and 1,152 respectively (Figure E.2b). If more than one entry of the reference set was matched to one protein group, these were counted as often as the reference set matched. Therefore, e.g. the intersection of all methods except MSBayesPro

has a higher number of true positives than for all reported groups. Fido identified 98, 44, and 41 more reference proteins than PIA, ProteinLP and ProteinProphet, respectively. Most of the proteins uniquely identified by Fido (78%) are included in the reference set. However, the inference algorithms are also reporting proteins that are not included in the reference set (Figure 5.9). The number of proteins uniquely reported by PIA, Fido, ProteinLP, MSBayesPro and ProteinProphet, that are not included in the reference set were 2, 12, 1, 0 and 0, respectively. This shows, that Fido uniquely reports the highest number of protein groups (12 groups) that are not in the reference set (FPs), but also the highest number of uniquely reported TPs.

Even more important is the fact, that 14 proteins were reported in consensus of all methods except MSBayesPro and 7 more in consensus of all methods. This raises the question, whether these proteins are rather true positives, which for some reason were not detected or validated by other methods at the time of creating the ground truth dataset. This is a general problem for ground truth datasets and could also be observed with other numbers for the iPRG 2008 dataset (not shown).

Also, we studied the protein clusters for the iPRG 2008 dataset. A cluster is a set of proteins with partially shared peptides and in the iPRG 2008 study a certain number of these protein groups should be reported as true positives (see Section 5.2.3 and the the original study's homepage[89]). This allows the calculation of numbers for false positives (i.e. too many reported protein groups in a cluster), false negatives (too few reported groups in a cluster) and true positives (exact number of reported group in a cluster). With these values the precision ($precision = \frac{TP}{TP+FP}$), recall ($recall = \frac{TP}{TP+FN}$) and the F1 score ($F1 = 2 \cdot \frac{precision \cdot recall}{precision+recall}$) as a combined metric of the precision and recall can be calculated, see Table 5.4 for the combination of all search engines and Table D.2 for all possible combinations, ordered by decreasing F1 score, as this allows an unbiased analyses of all the available values for each combination of algorithm and search engine results. It must be highlighted, that TP, FP, FN and TN in this context is relative to the number of correct protein groups per cluster, and not whether a protein was reported and is in the reference set, as in the preceding discussion for the yeast dataset.

The tables show, that PIA has a very good precision on all combinations, and also the highest F1 scores, except for the Mascot and MS-GF+ searches alone. ProteinLP yielded the same recall rates as PIA, but the precision was decreased. ProteinProphet had always a very high precision, which means it reported almost no false positives. This, though, was achieved by a relatively small overall size of the report and therefore low values for true positives and high false negatives. Fido on the other hand had high values for the overall reported groups, but with this also many false positives, which lead to the worst precision of all analysed algorithms. MSBayesPro suffered on the fewest reported groups and thus the smallest values for the recall. Even with a a perfect precision these values lead to the poorest F1 scores.

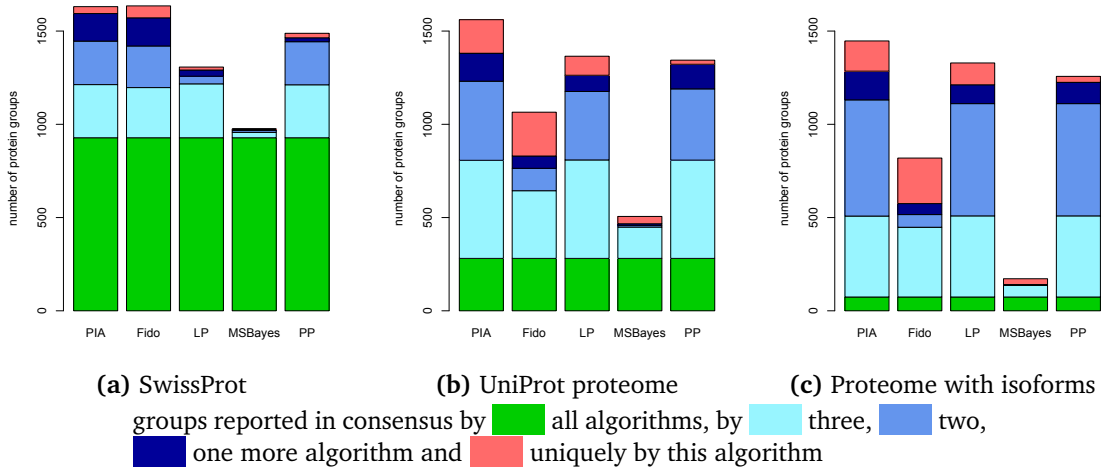
Table 5.4: The numbers of reported true positive, false negative, false positive, the resulting precision, recall and F1 score and the total number of protein groups for the iPRG 2008 dataset using the Swiss-Prot database. The table shows the respective numbers for each inference algorithm and the combination of all search engines ordered by decreasing F1 score. For discussion of the table, see the text. Results of all search engine merges are given in Table D.2.

Run	TP	FN	FP	precision	recall	F1 score	total number of groups
PIA	231	27	5	0.98	0.90	0.94	304
ProteinLP	231	27	20	0.92	0.90	0.91	319
ProteinProphet	192	66	1	0.99	0.74	0.85	199
Fido	99	159	120	0.45	0.38	0.42	236
MSBayesPro	57	201	0	1.00	0.22	0.36	77

5.3.8 Evaluation of the Overlap between Inference Algorithms

The next inspected metric is the number of reported protein groups per algorithm as well as the fraction of protein groups reported by other inference algorithms (Figure 5.10). The plots represent the numbers of protein groups reported for the PXD000603 datasets for the combination of the three search engines. First, we analysed the impact of the complexity of the underlying database used for the identification by comparing results with Swiss-Prot (Figure 5.10a), UniProt proteome (Figure 5.10b) and UniProt proteome with isoforms (Figure 5.10c). The overlap of protein groups reported by all inferences (green) is bigger when using the least complex database (Swiss-Prot) for identification. The same can be seen in the plots for all analysed datasets, given in the appendix (Figure E.3). Here it is important to remember, that the yeast Swiss-Prot and UniProt proteome database are identical.

Figure 5.10 shows that Fido increases more than any other algorithm the number of uniquely reported proteins when more complex databases are used (especially Figure 5.10b and 5.10c). This is mainly because Fido reports sub-protein groups (groups whose peptides are contained in another group). In contrast, PIA, ProteinLP and ProteinProphet seemed more robust against changes in the database complexity. PIA and ProteinLP tended to report the most groups on more complex databases (e.g. on PXD000603 PIA reports 16% more groups than ProteinProphet for the UniProt proteome without isoforms and 15% more for the proteome with isoforms). On the iPRG 2008 dataset, PIA and ProteinLP reported on average 42% and 40% more than ProteinProphet, respectively. Here, Fido and ProteinProphet reported similar numbers of protein groups for the Swiss-Prot dataset. However, on the other two databases (more complex ones) Fido reported 33% fewer than ProteinProphet. In less complex databases Fido performs better than the other inference algorithms (e.g. an average of 5% more protein groups than ProteinProphet on the yeast dataset). These analyses and the prior discussed Venn diagrams show that even if the actual numbers of reported protein groups may be similar be-



(a) SwissProt **(b)** UniProt proteome **(c)** Proteome with isoforms

Figure 5.10: Number of protein groups reported and their overlap for the PXD000603 datasets using different inference algorithms and databases. The plots show the number of protein groups under a 1% FDR q-value for the PXD000603 dataset with the corresponding (a) Swiss-Prot, (b) UniProt proteome and (c) UniProt proteome with isoforms databases. The bars colour-codes represent the overlap: protein groups reported by all inferences are in green (bottom), groups reported by 2, 3 and 4 groups in blue with increasing darkness. Unique groups are red. It can be seen, that with increasing complexity of the database, the reports' consensus decreases. While Fido's results are decreasing, the number of uniquely reported groups increases more than in the other groups. This can be explained due to reported sub-proteins. PIA, ProteinLP and ProteinProphet seem to be relatively robust against the changes in database complexity.

tween the inference algorithms, the actually reported groups and their overlaps differ between the algorithms.

5.3.9 Impact of Multiple Search Engines on the Protein Level

Not only the PSMs, but also the number of reported protein groups is often increased when more search engines are combined compared with the results of single search engines (Figure 5.11). Interestingly, in the given example though, the numbers from MS-GF+ alone yields the most FDR 1% valid results for all inference algorithms, while for all other combinations the combination of more search engines leads to more protein groups. Furthermore, for each single dataset a pattern for the ratios between the inference algorithms and search engine combination can be observed. For example Figure 5.11 shows that Fido reports the largest number of protein groups, followed by PIA and ProteinProphet, then ProteinLP and MSBayesPro always reporting significantly fewer groups. However, the combinations of search engine and inference algorithm should be selected carefully, as some seemed not to produce optimal results. For example in the given plot, when X!Tandem alone is used the number of reported groups is fewer with respect to all other combinations.

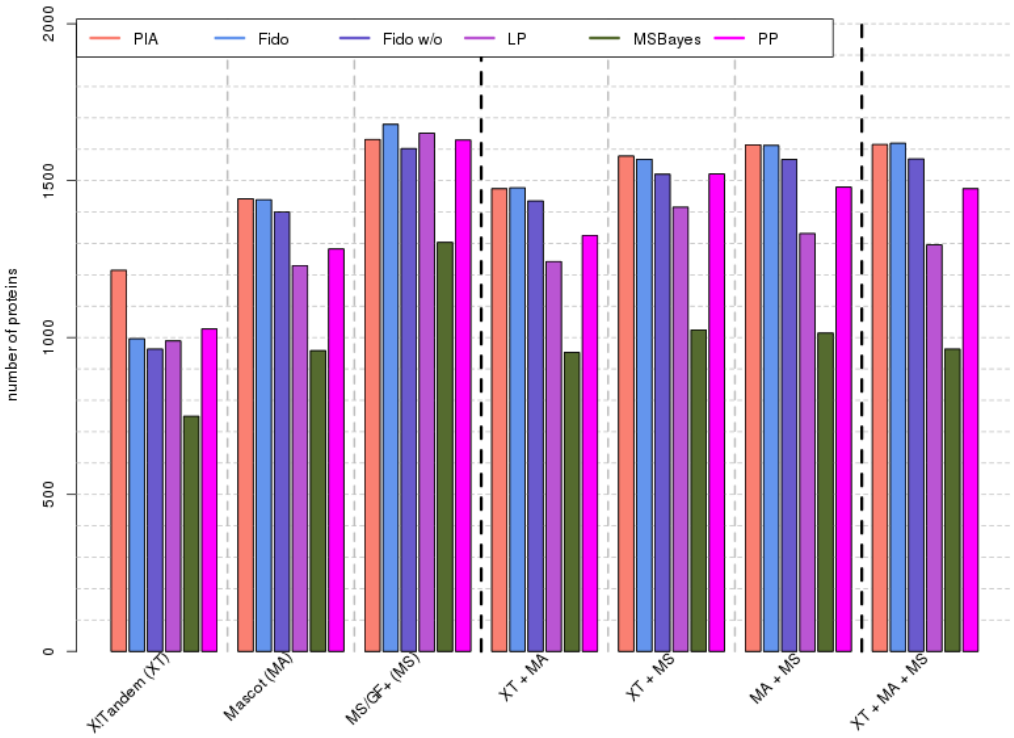


Figure 5.11: Number of protein groups reported using different inference algorithms and search engine results on the Swiss-Prot database for the PXD000603 dataset. The bars represent the number of FDR 1% valid protein groups reported for all analysed inference algorithms and combinations of search engine identifications. For most combinations the same pattern for a ratio between the inference algorithms can be seen, as well as an increase in the number of reported protein groups when combining search engines.

As a further assessed quality metric for reliable identifications, we analysed the number of peptides per protein groups for each protein inference algorithm, as recommended by Omenn recently[107]. The numbers of peptides per protein group were plotted for the results of the PXD000603 dataset with the Swiss-Prot database in a heatmap-like way in Figure E.4 and also given in Table 5.5. Independently of the inference algorithm, most protein groups are reported with few peptides and only a small fraction is represented by ten or more peptides. In the shown dataset, the number of inferred protein groups with ten or more peptides from the single search engines' results with PIA, Fido and ProteinProphet are on average 5.7% (ranging from 5.1% - 6.7%) of all reported groups. Using the results from multiple search engines increases these groups in average to 6.5%, though for the X!Tandem-Fido combination the percentage is decreased by 0.2%. The actual numerical values are always increased by at least five protein groups with ten or more peptides in the merged results than in the runs using single search engines' results. The choice of at least ten peptides per protein was not based on specific observations, but the table shows that for this dataset almost 95% of the groups

have less peptides. Therefore, protein groups with at least ten peptides are of reasonable good identification quality.

Table 5.5: This table shows the total numbers of reported protein groups and the number of groups with at least 10 peptides per group for the given search engines and inference algorithms. Given are the results for each single search engine and the merge of all three search engines for the PXD000603 dataset and the Swiss-Prot database. The first column represents the total number of 1% FDR valid groups and the second the respective number and percentage of groups with at least 10 peptides per group. The table shows, that the merge of search engines always increases the number of groups, that have more than 10 peptides. Also in most cases the fraction of these groups is increased, except for Fido, which has the biggest fraction of groups with at least 10 peptides using the X!Tandem results alone.

	X!Tandem		**Mascot**		**MS-GF+**		**Merged**	
PIA	1214	64 (5.3%)	1442	74 (5.1%)	1631	93 (5.7%)	1615	**101 (6.2%)**
Fido	996	67 **(6.7%)**	1439	80 (5.6%)	1679	96 (5.7%)	1619	**105** (6.5%)
ProteinLP	989	64 (6.5%)	1229	77 (2.3%)	1651	93 (5.6%)	1295	**104 (8.0%)**
MSBayesPro	749	24 (3.2%)	958	26 (2.7%)	1303	31 (2.4%)	963	**36 (3.7%)**
ProteinProphet	1027	64 (6.2%)	1282	73 (5.7%)	1629	91 (5.6%)	1474	**99 (6.7%)**

5.3.10 Difference between Protein Level FDR and q-value

Interestingly, the inference of Fido in some runs that include results from Mascot alone is significantly increased regarding numbers of protein groups compared to all other inferences. Additionally, this can be seen on the combination of Mascot and MS-GF+ results in the iPRG 2008 dataset with the provided database (both not shown here). This effect can be explained by the fact that the local FDR and the FDR q-value on protein level differ under certain circumstances (Figure 5.12). Under specific conditions the local FDR (and therefore the q-value of all preceding elements in a sorted report) returns to a low value, after increasing rapidly due to several reported decoys. If during this increase and decrease of the local FDR many targets are reported the respective pseudo-ROC shows a step or a peak, if this occurs at the end of the list. This effect could be observed in several of the created pseudo-ROC curves for Fido and ProteinLP in this analysis. Though except for only a few combinations this effect occurs on q-values exceeding the threshold of 0.01 (i.e. somewhere between 0.01 – 0.05, in the given Figure 5.12 at 0.022). For all analyses discussed here we used the q-value, as it is currently a widely accepted method. This behaviour though shows that a method controlling both FDR q-value and local FDR might be more applicable in general.

5.3.11 Ranks of Uniquely Reported Protein Groups

Though it cannot be used as a metric, but only as an observation, an evaluation of the ranks of the uniquely reported protein groups sorted by probability/score revealed some unintuitive

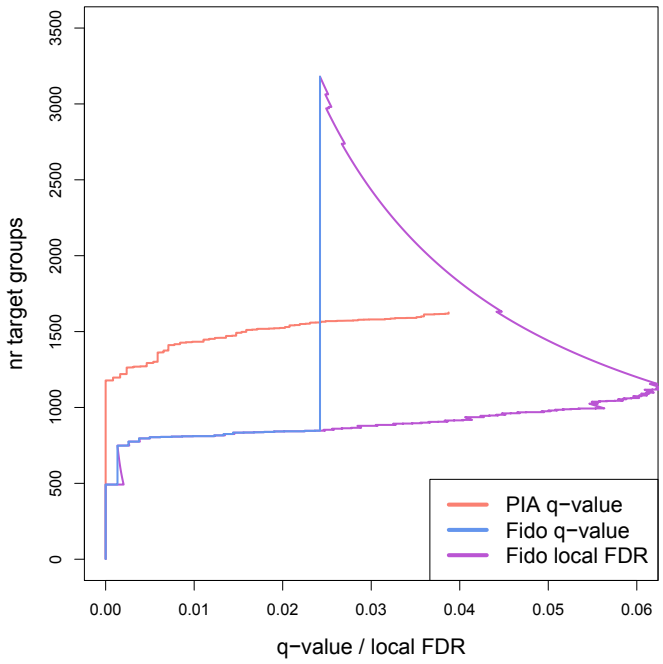


Figure 5.12: Pseudo-ROC plots of the protein groups reported for the PXD000603 datasets using the merged results of all search engines on the proteome database with isoforms and either the FDR q-values or the local FDR. This plot shows for the Fido results, that under certain circumstances the q-value can differ significantly from the local FDR. If this effect emerges below the given q-value threshold (usually 1%), the affected method generates more reports than expected. Larger differences between the local FDR and q-values can be seen at two ranges: one at q-values of 0.001 and a much more significant one for q-values of 0.022. The respective plot for the PIA q-values is given as a reference: here no larger discrepancies could be detected and therefore the PIA local FDR values were not plotted.

distributions (Figure 5.13). If we assume that the top ranking protein groups are the most valid, an intuitive distribution should represent protein groups that are not reported in consensus (unique) at the end of the reported lists with low scores. For PIA almost all uniquely reported groups are at the end of the list, as expected. On the other hand, Fido and ProteinLP distributed the unique groups over the complete range of indices, only with a tendency to the end of the list. The most extreme case is ProteinProphet which reports its unique groups at the very beginning. This reveals that the intuitive assumption that the majority of the uniquely reported groups are located at the end of the report is not correct for most of the analysed algorithms.

5.3.12 Discussion

We have evaluated in detail the performance of different inference algorithms using four different datasets and a set of well-define metrics. MSBayesPro needs detectability predictions for each peptide as an input of the inference algorithm. These values can only be calculated

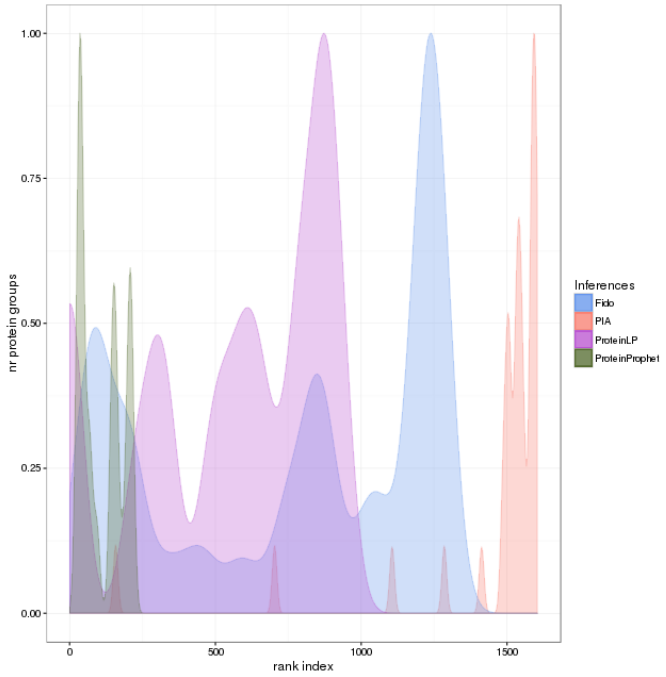


Figure 5.13: Distribution of the ranks of uniquely reported protein groups. This plot shows for the analysed inference methods, on which ranks in the reported list of protein groups uniquely reported groups occur. Depicted is the data from the merge of PSMs from Mascot, X!Tandem and MS-GF+ for the PXD000603 dataset using the Swiss-Prot database. For PIA it can be seen, that almost all uniquely reported groups are at the end of the list. Fido and ProteinLP, on the other hand, distribute the unique groups over the complete range of indices, though with a tendency to the list's end. The most extreme case is ProteinProphet which reports its unique groups at the beginning. This reveals that an intuitive assumption, that the relatively high consensus of reported groups is found in the top of the report, is not correct for all algorithms.

using the results of preceding experiments or estimated using algorithms like the *PTModel*. Both modelling approaches have drawbacks when experimenting with analytical methods (e.g., enrichment, different fractionation methods) for which there are no preceding reference results. In these cases, these inference algorithms depending on detectabilities will not perform well. Prediction of detectability increases the running time and the predicted model (MSBayesPro) is not available making the integration into bioinformatics pipelines difficult.

A uniqueness of the Fido implementation in OpenMS is that it requires a decoy database and can perform an estimation maximization to find the best values of the parameters (α, β, and γ) by combining a ROC optimization (in a supervised manner) with FDR estimation. Fido is a very fast implementation with a small memory footprint. With its integration into OpenMS it can easily be used in bigger workflows. In most of the analyses Fido reports more protein groups than the other algorithms. The underlying generative (Bayesian) model relies on reasonable probabilities for the observed peptides, which are besides the three model parameters the

only input to the algorithm. Although being relatively robust for multiple types and shapes of distributions of these input probabilities, even with parameter estimation, it cannot correct for heavily ill-shaped distributions. Therefore for Fido, one should be careful with inputs of extreme probabilities for the peptides, such as 0.0 or 1.0. In a Bayesian model these extrema strictly exclude every combination not using this peptide although other information suggests differently (which is especially a problem when assigning a probability of 1.0 to so many peptides as was observed for X!Tandem in several cases). A second problem is the lack of discriminative power between equal scores. Since the parameter estimation of Fido tries to create well-calibrated and well-discriminating results at the same time, this creates an issue. Extreme values for peptide probabilities as inputs are likely to generate extreme probabilities for the proteins. If more than a minor percentage of the proteins are assigned probabilities of 1.0 and these include decoy proteins, the first q-value-cutoff considered in a corresponding receiver operating curve (ROC) results in an uninformative straight line in the upper part of the curve covering all proteins of probability 1.0. This, in turn, limits the usefulness of the parameter estimation at all. These factors make the results of Fido less constant than the other algorithms and demand more benchmark and tuning of the pipeline. This also explains the significant decrease of reported protein groups for the more complex databases when using Fido.

ProteinLP and Fido have as a main concept not the parsimony of peptides or spectra but the probability of proteins' occurrences given the PSM or peptide probabilities. By design, they report sub-proteins if the respective probabilities are sufficiently high. This difference to the parsimonious approaches such as PIA or ProteinProphet should be considered when choosing an inference algorithm. If many sub-protein groups were reported (like in the data given in Figures 5.10b and 5.10c, which show many unique groups for Fido), the FDR q-value did increase due to reported decoy sub-groups as well, and thus the total number of reported protein groups decreased. In some combinations of databases, datasets and search engines the number of reported groups rises significantly above the reports of other inference algorithms. This is due to an effect of the protein FDR q-value and the local protein level FDR values (Figure 5.12). During this effect, the local FDR may exceed a given threshold significantly and drop below it after reporting many target proteins towards the end of the report. This leads to distinct steps in a corresponding pseudo-ROC curve and suggests to employ more advanced methods than the q-value or local FDR alone, either combining these two approaches or employing algorithms like Mayu[108].

ProteinProphet, which is one of the oldest approaches, has a very low memory imprint and thus scales well to process big datasets. It is more conservative in reporting protein groups than other approaches, but also reports less false-positives in the reference datasets. It can be observed in the Venn diagrams and the iPRG 2008 cluster analyses that it consistently reports a low amount of unique proteins reducing the possibility of false positive identifications (Figure

E.2). One of its main strength is the integration into the Trans-Proteomic Pipeline, which incorporates multiple search engines like X!Tandem, SEQUEST, Comet or Mascot.

ProteinLP consumes the highest amount of memory and time. Although, it performs well for most of the datasets in all the analysed metrics it is outperformed by other inference options.

Among parsimonious approaches, PIA mostly reports more target protein groups than ProteinProphet in the studied datasets. PIA consumes a relatively large amount of memory analysing a not-FDR filtered or very big dataset. It reports high numbers of confident protein groups and like other parsimonious approaches it is relatively fast. However, ProteinProphet yields less false positive identification when it is used to analyse the ground truth datasets (section 5.3.7). For both of these datasets Fido reports more proteins than other algorithms but also more possible false positives (proteins that are not labelled in the reference set). Still, there is one big issue with the ground truth datasets: many proteins could be identified in consensus by all inference algorithms, which were not in the reference set. Thus, these were labelled as false positives, though they might also be true positives. For iPRG 2008 this would definitely have been the case, if the dataset was analysed by other search engines and inference algorithms at the time of the creation of the reference set. For this dataset, the references are the identifications of the initiators of the study. Also for the yeast dataset, a re-analysis or validation by additional proteomics methods might change the reference set of accessions.

A feature that should be considered when choosing an inference algorithm is the robustness when using complex databases for spectrum identification. While PIA, ProteinLP and ProteinProphet were only slightly affected by this, Fido and MSBayesPro reported significantly fewer valid protein groups at 1% FDR q-value when using more complex databases. Table 5.5 and Figure E.4 show the number of peptides per protein group under 1% FDR. These present a small drawback for parsimonious approaches, which reported slightly fewer groups with many peptides than Fido and ProteinLP. The latter reported also more single peptide proteins, but ProteinProphet had also on some combinations the highest percentage of groups with more than ten peptides. The "two peptides"-rule, i.e. at least two peptides per protein are required, which is applied quite often in proteomics to control protein false discoveries[107], can affect and change the results of the experiment depending of the inference algorithm used and further increase the quality of the results, though at the cost of sensitivity.

Also, the interoperability and ease-of-use of an inference algorithm will influence its application by a user. All analysed algorithms except PIA need special non-standard input formats. It would be very beneficial for users, if standard formats like mzIdentML or even the search engines' default result files could be used as input. PIA also has the advantage that it works with spectrum identifications coming from various file formats, search engines and bioinformatics workflows. It is the only implementation that works natively with standard file formats such as mzIdentML and mzTab. ProteinProphet uses the pepXML format that does have converters for many search engine results and is well known in the proteomics community[109]. PIA and

Fido are the only algorithms at the moment which can be fully integrated into an OpenMS workflow and thus inside KNIME.

5.3.13 Conclusion of the Assessment

We introduced a workflow that uses three search engines and five open-source, generally applicable protein inference algorithms for a fair and in-depth comparison. The workflow and inference methods were tested on four datasets with different complexities of protein databases. While there is no explicit best inference algorithm, different considerations for choosing a tool can be given.

The analysis of identifications using protein databases with varying complexity shows some algorithm specific results. Due to the occurrence of more decoys, all inference algorithms report fewer groups when more complex databases are used. The numbers of reported groups by PIA, ProteinProphet and ProteinLP, though, are much less dependent on the database complexity than Fido and MSBayesPro are. If the detection of specific isoforms is important in the scientific context, this stability could compensate for slightly less reported protein groups, though. Furthermore, the increasing demand for analysing metaproteomics datasets, i.e. datasets containing the proteomic analyses of a multitude of species, like gut or soil samples, needs inferences, which are robust to very bg protein databases. Therefore, depending on the required analysis, it might be better to use an inference that is a bit slower and more memory consuming like PIA, but more robust against size variations.

Depending on the underlying report, the FDR q-value may be not sufficient to filter for good identifications. This is especially the case if the local FDR exceeds a given threshold for a big part of the report, but finally drops below the threshold again. To improve on this problem, other strategies should be developed. Another very interesting comparison of protein inference algorithms and the fundamental search engines would be an analysis of the reported isoforms or splice variants matched on a gold standard dataset, containing the knowledge of isoforms and splice variants. This could not be tested thoroughly, due to the lack of publicly available datasets at the time of writing. We also expect that more complex "gold standard" datasets will lead to a fairer comparison of protein inference algorithms.

ProteinProphet has a more conservative approach and reports less false positives in all the analyses. This also fits to the fact, that it has often the highest fraction of proteins with more than ten peptides on a percentage basis. The parsimonious approaches are less dependent on the search engine scores distribution than Fido. Although being relatively robust for multiple types and shapes of distributions of the input probabilities, even with parameter estimation, Fido cannot correct for heavily ill-shaped distributions like some results from X!Tandem in the discussed analyses.

Furthermore, the created workflow could easily be adjusted to benchmark different protein inference algorithms in the future and thus gives a fair framework for testing of protein inference algorithms in general. Overall, one possibility for future improvements to the inference methods could be the use of additional information during the inference process, especially, since data from the MS1 level, like deviation from predicted retention time or intensities, is readily available in almost all experiments. Another source of information could come from technical replicates, where agreeing identifications may boost the confidence of its correctness.

Many of these aspects might not impact a daily analysis, but should be taken into consideration when choosing a protein inference. For example many analyses are run using only a respective taxonomical Swiss-Prot portion of UniProt. On these databases, most algorithms perform relatively equal. It should be considered, though, whether one would use a parsimonious approach or not. If the goal is to look for protein isoforms as well, or even analyse a metaproteomic sample, either PIA or ProteinProphet should be selected, based on the given assessments. PIA actually shows the best overall performance: it reports many qualitatively good proteins early under 1% FDR (Figure 5.8), has the best combination of precision and recall when using the F1 score as comparative metric and is robust against issues of big protein databases for spectrum identification. Furthermore, it is platform independent and can be used in different ways (command line, KNIME and web frontend), which makes it a good overall choice.

5.4 Advanced Application Example: Protein Isoform Detection

5.4.1 Motivation

For many entries in the UniProt databases, there are isoforms additionally to the canonical sequences. Most MS-based proteomics studies were conducted using the canonical sequences only in the past. Lately, also the reference proteome sets are used, which contain some isoforms (compare also Chapter 4 for nomenclature and number of entries).

Generally, it is not only of interest to identify the canonical sequence, but also decide, whether a specific isoform appears in a studied sample. Several current studies are aiming at the detection, identification and quantification of protein isoforms and their functional analyses[110–112]. This is a not trivial task, as much of the canonical protein sequence and its isoforms are identical, due to the concept. Furthermore, identifying peptides against an database containing all isoforms can lead to problems with the inference algorithms, as explained in Section 5. PIA provides a valuable framework for the isoform analysis, as it is not deeply impacted by the usage of increased databases and gives good visualisation and browsing opportunities to inspect isoforms in-depth. In this section, a publicly accessible dataset (PXD002279[112]) was analysed and some of the observed isoforms are explained in detail.

5.4.2 Analysed Dataset

The analysed dataset was created to gain insight into the protein isoforms encoded by the human SLC12A3 gene. This gene translates into three isoforms, which are summarized under the accession P55017 in UniProt. For the MS-based proteomics analysis, urine samples were measured on a Q-Exactive using HCD after digestion using trypsin or Lys-C. For more information on the sample preparation, please refer to the original manuscript[112] or the PRIDE download page.

As an application example for isoform identification using PIA, the RAW file of the second sample of the tryptic digest (QE1_150213_OPL4021_TRAS_SLC12A3_T2.raw) was downloaded and converted into mzML using msconvert. For the identification of spectra Mascot, MS-GF+ and X!Tandem were used with the UniProt reference proteome for *H. sapiens* including all isoforms (version 2016_01, see Table 5.2). Carbamidomethylation of cystein was set as fixed modification, oxidation of methionine, Gln-> pyroglutamate of N-terminal glutamine and N-terminal acetylation as variable modifications. Two missed tryptic cleavages, a precursor tolerance of 10 ppm and a fragment tolerance of 20 mmu were allowed, as stated in the publication.

The identifications of Mascot, MS-GF+ and X!Tandem were combined with PIA, using the *FDR Score*. For the protein inference, the *Spectrum Extractor* was used with all PSMs fulfilling

an *FDR Score* threshold of 0.01. Overall, 1.326 protein groups were identified on an FDR threshold of 1%.

5.4.3 Detected Isoforms

In the analysis, only the isoform entries, which also have an additional canonical entry in the database and are identified by the accession suffix "-X", were assessed. Alternatively there are entries, which have the term "isoform" in the protein description. Of the identified 1.326 protein groups, 27 (2%) were matched to isoforms only, i.e. they had no canonical accession in the group. More often though, the groups contain canonical and isoforms (564 groups, 43%), while the remaining 735 groups (55%) contained only canonical accessions. For six of the groups with isoforms only, also a group with the canonical accession was detected.

Groups containing canonical accessions and the respective isoforms can be easily explained: for them only common enzymatic sequences are identified. In some cases, several isoforms become sub-sets of the reported group, as their sequences and more are completely explained by the peptides of the reported group. As explained before, the existence of the sub-groups' proteins in the sample cannot be excluded based on the identifications only. Sometimes for isoform only groups, no unique peptide for the reported isoforms are identified. Then the specific isoforms are reported, because they are the intersecting set of accessions of the respective peptides - and other isoforms are in sub-groups.

In Figure 5.14 an example of the analysed dataset is given. The protein group with the accessions [P01133, P01133-3] and [P01133-2] are both reported. The isoform P01133-2 misses a stretch if amino acids in the middle of the sequence and is otherwise identical to the canonical sequence, while P01133-3 misses another stretch toward the C-terminal of the protein. There are 7 peptides identified, which are in the sequence missing in P01133-2, but none in the missing stretch of P01133-3. Also, no peptide spanning the characteristic gap created by the sequence lost of P01133-3 is identified, therefore the group [P01133, P01133-3] with common sequences is reported. For P01133-2 a unique sequence was detected, which spans the characteristic gap between amino acid 314–355 in the canonical sequence and is thus reported as well.

Another interesting example is the report of isoform O00159-2. This isoform is a truncation of the canonical form, missing the first 35 amino acids. The N-terminal peptide of this isoform was identified, while there is no tryptic cleavage site at position 35 in the canonical sequence. Hence, the report of the isoform group, but also the canonical group, as a peptide inside the first 35 amino acids was detected as well. It is arguable though, whether there might be a semi-tryptic cleavage which created the N-terminal peptide of the isoform. This though is not verifiable with the given methods.

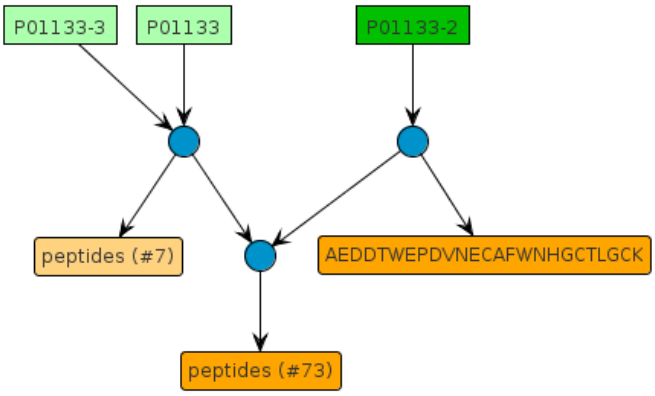


Figure 5.14

The original aim of the study, for which this dataset was created, was the analysis of the proteins given by the accessions P55017, P55017-2 and P55017-3. P55017-2 has the addition of the amino acid sequence GARPSVSGAL at amino acid 807, while P55017-3 has this additional sequence as well as the missing of one amino acid at position 95. Figure 5.15 shows the cluster containing the P55017 isoforms. It can be seen, that the specific additional peptide GARPSVSGALDPK, which specify P55017-2 and P55017-3, is identified in the sample, shifting the favour for the report towards these isoforms. Furthermore, the peptide with the sequence KVRPTLADLHSFLKQEGR was detected, which contains the position, that is missing in isoform P55017-3. Additionally, 40 common peptides are identified. Thus, only a group for [P55017-2] is reported, as there is no further unique peptide. Still, also the other two peptides may be present in the sample.

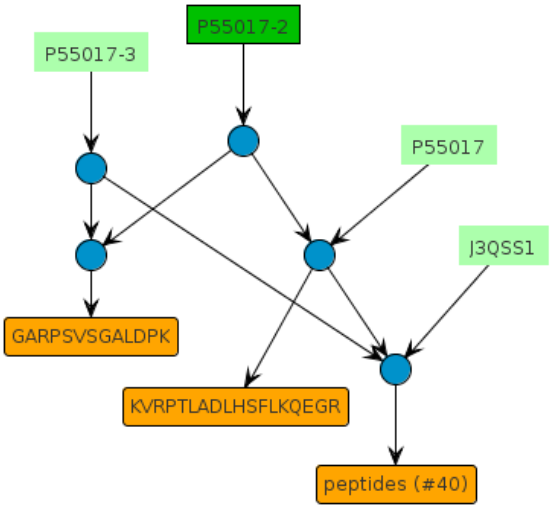


Figure 5.15

5.4.4 Conclusion

For analyses of the kind described in this section, the web-interface and also the current KNIME nodes of PIA are ideally. Here, the cluster combinations and all peptides and PSMs leading to the reported protein groups can be inspected in-depth. The visualisations and the concept to always report protein groups instead of representatives helps in inspecting the relationships, which can become quite complex.

Chapter 6

PIA - Protein Inference Algorithms

In this chapter the set of algorithms and tools called "PIA - Protein Inference Algorithms" will be described in detail. PIA is a flexible software suite for combining PSMs from different search engine runs and turning these into consistent results. PIA can be integrated into proteomics data analysis workflows in several ways, using KNIME or the command line, as described in Section 6.3). Additionally, a user-friendly graphical web interface can be run either locally or (e.g., for larger core facilities) from a central server.

The contents of this chapter are in parts published in:
"PIA: An Intuitive Protein Inference Engine with a Web-Based User Interface.", Uszkoreit et al. *J Proteome Res*. 2015 Jul 2;14(7):2988-97.[1]

6.1 Design Goals

As explained in Chapter 5, several protein inference algorithms exist already. Most of them, though, suffer from one of the following points: limited support to report details of protein ambiguity groups, inflexibility of settings, restrictions regarding the import of identified spectra and export of protein results, no visualisation of the PSM, peptide and protein relations. Furthermore, the actual algorithms of the protein inference are not always open, especially in commercial products.

One aspect which makes a protein inference necessary for bottom-up MS proteomics is the existence of shared peptides, for which the originating protein can not be detected unambiguously (see Chapter 4). This is especially true for eukaryotic organisms due to homologous proteins or domains and multiple protein isoforms. These shared peptides lead to sets of proteins, the protein ambiguity groups, which are built up of the same (sub-)set of peptides and it cannot be decided which of the proteins were actually present in the sample unless discriminating (unique) peptides are found or any other knowledge about the proteins' existence in

a sample can be used for the inference. Often for each such group only one representative protein accession is reported in the result list and the other proteins are - if at all - reported as "similar proteins" or "group members". For a complete result list, all these possible proteins (according to the inference algorithm) should be reported.

The set of PSMs selected for protein inference, the inference algorithm, and the selection of reported representatives vary significantly between inference methods as shown in[105]. For some, mainly the algorithms included in commercial search engines and tools, but also some freely available algorithms, the details are scarcely described, so that results cannot be completely explained or it cannot be judged whether they are reasonable for a specific question. Additionally to the search engine inherent inference algorithms, there are also stand-alone programs for protein inference from PSMs, some of which were highlighted in the preceding chapter. Many of these implementations support only specific search engines and most are limited in their settings for inference parameters.

Merging the results from multiple search engines is desirable to either increase the number of identified spectra passing an FDR threshold and thus hopefully also the number of corresponding proteins, or to solidify the evidence of peptides detected in the analysed sample, as discussed in Section 3.2.3. This poses a major problem, because each search engine's algorithm generates its own value for the quality of a PSM, generally a score or probability value. These scores are usually not directly comparable. They thus need to be translated to a directly comparable, search-engine independent score[27–29] prior to combining different search results.

The implementation of "PIA - Protein Inference Algorithms" addresses several of these concerns. It is based on this concept and designed to only work on protein groups, never the single accessions alone. It also provides concepts to report all sub-groups and creates comprehensive exports into standard formats, which support these information. PIA reports consistent and comparable protein ambiguity groups as result of one of the implemented flexible protein inference methods. The implementation allows the choice of several protein inference and scoring methods and direct access to all required parameters. Essential analyses like the calculation of false discovery rates (FDR) on the PSM level and the protein level are directly included. For import and export, PIA supports the standard formats mzIdentML[75] and mzTab[81] for protein identifications developed by the HUPO Proteomics Standards Initiative (PSI) and thus virtually all search engine results. Additionally, importers for the most commonly used search engines are provided as well as algorithms for the combination of PSMs obtained from different data sets and/or search engines.

To give the user easy access, PIA is fully integrated into KNIME[96], providing nodes to connect the protein inference to OpenMS[97] workflows, including quantitative analyses. This also encompasses KNIME nodes to visualise the relationship between PSMs, peptides and proteins as well as the logic used for the inference. This same functionality is also included in an intuitive web-based graphical user interface (written for JavaServer Faces, JSF). This

interface can either be used in local installation or via a public web server. For scripting and automation, all functionalities can also be called via the command line. PIA is open-source software and completely written in Java.

6.2 Basic Concepts

The implementation of PIA is based on some assumptions and principles, which will be discussed in the following paragraphs. The concepts mainly encompass the separation into two steps, compilation of search engine results and a following analysis, the three conceptual layers consisting of the PSMs, peptides and proteins, the implemented protein inference and scoring methods and finally the visualisation of the protein inference.

6.2.1 Compilation and Analysis Steps

Before doing any analysis, the imported search engine results are structured into a tree like graph containing PSMs, peptides, proteins and additional group nodes. This graph can be stored in an intermediate XML file together with additional search engine settings and identification information. This information is especially useful to later perform exports into fully annotated standard formats like mzIdentML. The structured data enables the algorithms of PIA to quickly access the hierarchical information connecting all PSMs to peptides and proteins and vice versa and provides an intuitive visualisation of these connections. The methods, how this graph is generated, is further discussed in Section 6.4.2 and Figure 6.8. If the compilation is stored, it must be performed only once per set of search engine results.

It is important to note, that the information stored in PSMs and proteins, respectively their relations, are not checked or updated by PIA. The importer relies on the information given by the search engines or any preceding steps. This must be especially noted, if a file which might be filtered in any way or for which PSMs are connected to only few proteins, like PRIDE-XML, is imported. It is thus recommended that before importing from these files, peptide indexing to a suitable database or a similar preprocessing is performed.

6.2.2 The Three Layers of Data in PIA

For the analysis it is important to know, that PIA structures the data into three layers:

- **PSMs**: A peptide spectrum match (PSM) refers to a match from an MS/MS spectrum to an amino acid sequence with charge state and identified modifications, which derives from one search engine run and contains the search engine's scores.
- **Peptides** A peptide in contrast refers to an amino acid sequence without charge state, either regarding modifications or not, depending on user settings used for the inference.

- **Proteins** A protein refers to an entry in a database (the raw amino acid sequence without any post-translational modifications), mandatorily containing an accession and, if available, a complete amino acid sequence and further descriptions.

On the PSM level results from multiple search engines for the same LC-MS run are assembled into PSM sets, which combine the identical identifications originating from different search engines. To assemble these sets, all basic PSM information (m/z, retention time, source ID, spectrum title, sequence, modifications and charge) available from all input files is used. If all input files have information for the source ID, which refers to the actual identified spectrum for each PSM, the source ID, sequence and modifications (with their correct locations) are sufficient to construct the PSM sets. If the assembly of PSM sets is not needed, then it can be turned off, e.g., in case a compilation of successive LC-MS/MS runs is intended.

To evaluate the quality of the identification data and to calculate the FDR, a search against a target-decoy database is always recommended (compare e.g. Elias et al[64], Käll et al.[69] and Section 3.2). If such a search was conducted, then not only can a regular expression to distinguish decoy accessions from targets be set but also decoys generated by an internal target-decoy search can be used (e.g., used by Mascot and ProteomeDiscoverer). FDR, q-value, and *FDR Score* for each PSM are then calculated from this data. For PSM sets, the *Combined FDR Score* is computed as a comparable quality value for results from different search engines (compare Section 3.2.3).

For an inspection of the data on the peptide level, all PSMs and PSM sets with the same amino acid sequence are grouped into peptides. Additionally, it can be specified as to whether modifications should be considered in order to distinguish peptides. This peptide step can be used to review the peptides and associated PSMs of proteins of interest or to obtain a general overview of the identified peptides. If spectral counting as a fast, though not very reliable, method for peptide and protein quantification should be performed, this can be easily done on the peptide level, as shown in[87,113,114].

Before accessing the protein layer, a protein inference must be performed. The protein inference in PIA depends on the choice of the method for the protein scoring, the inference algorithm, and selected filters. The implemented methods for these will be discussed in the following sections. PIA is based on the principle to always report protein ambiguity groups and no single accessions. Therefore, on the protein level groups are reported. These, though, might contain only a single accession. Furthermore, a group may have *sub-groups*, which are groups that consist of a sub-set of the PSMs and peptides of the respective protein group, see Section 3.4.

The data on each level can be filtered by a multitude of parameters, for example sequence, m/z value, mass deviation, retention times and scores etc. Independent of the used frontend,

export into mzIdentML, mzTab, idXML and CSV formats are possible for any further analysis or submission into repositories like PRIDE[115].

6.2.3 Protein Scoring Methods

Depending on the type of PSM scores (arbitrary search engine score, p-values, or e-values), different rules for the protein scoring are applicable and can be chosen. Currently three protein scoring algorithms are implemented:

- **Additive scoring**
  For this scoring, the PSM scores are simply summed up to calculate the final protein score. This score should only be used, when PSM scores are used, for which higher scores represent better values, like the *Mascot Ion Score*. This scoring schema is very common and also used by Mascot and ProteomeDiscoverer.

- **Multiplicative scoring**
  This scoring method multiplies the PSM scores and should therefore be applied, if a probability or p-value like score is used as basis. To avoid numerical problems with very small numbers, the negative logarithmic values are added for the actual protein score. Thus, the protein score tends to grow with the number of assigned peptides.

- **Geometric mean scoring**
  The geometric mean is the n-th root of the product of n numbers. This scoring can be used for both kinds of PSM scores, either with higher or lower score better. The resulting protein score gives a tendency towards the mean of the original PSM scores, it will lay in between the highest and lowest used PSM score. This value of the protein score therefore does not reflect the amount of assigned peptides. Also for this score, logarithmic values are used if PSM scores with lower score better are used.

Besides the better numerical correctness, the usage of logarithmic values in the multiplicative scorings has additionally the benefit, that for the resulting protein score a higher numerical value is always better. As this is true for all of the three implemented scoring methods, no special care regarding this must be taken in any surrounding workflows.

Additionally to selecting the method, one of the available PSM scores must be selected. All search engine scores (like Mascot Ion Score, X!Tandem Hyperscore and X!Tandem expectation value etc.) but also the *FDR Score* and the *Combined FDR Score* can be used for the basis of the protein scoring. Furthermore it must be decided, whether all PSMs or only the best scoring PSM per peptide should be considered for the calculation of protein scores.

6.2.4 Protein Inference Methods

Currently PIA supports three different inference methods. Each of these can additionally be customized by several filters on the PSM, peptide and protein level (compare Section 6.4.2). These filters can also be seen as additional settings for quality standards, as they allow to set score thresholds or to require at least a certain number of spectra or peptides per protein. As discussed before, PIA is based on the principle to always report protein ambiguity groups instead of single protein accessions. Also, no representative is chosen, as it is in general not possible to decide for any accession to be superior in a protein group.

All currently implemented inference methods are purely deterministic, in contrast to probabilistic algorithms like Fido[95] or MSBayesPro[92]. Though in theory the framework of PIA would also allow the implementation of probabilistic algorithms.

The following paragraphs explain the implemented "Report All", "Occam's Razor" and "Spectrum Extractor" in more detail.

Report All

This is the simplest possible inference method, returning any possible protein group in the compilation of search results. Taking the PIA intermediate structure, the reported proteins are very rapidly calculated, as only one protein group for each group in the graph containing at least one protein node needs to be created. The advantage of this method is its short runtime, with the disadvantage of calculating no sub proteins. This method does not report protein lists that would be accepted in current publications, but it can be used to obtain a quick overview of the PSM and peptide data for a protein, which is actually not reported by any other method.

Occam's Razor

Here, the goal is to use the principle of maximum parsimony to report a minimal set of protein groups, which explains the occurrence of all the identified peptides that pass the selected filters. Given the example in Figure 6.1 (and assuming no further filters), the protein groups with single proteins *Protein 3*, *Protein 4* and *Protein 5* would be reported (marked with red boxes). This method also reports subgroups; in the example, the group containing *Protein 1* and *Protein 2* would be a subgroup of *Protein 3*.

As the data is already structured into the intermediate format, each single cluster or connected component of accessions, peptides and PSMs can be processed concurrently. To speed up the calculation of reported protein groups, parallelization is implemented and uses as many threads as are available or allowed by the user.

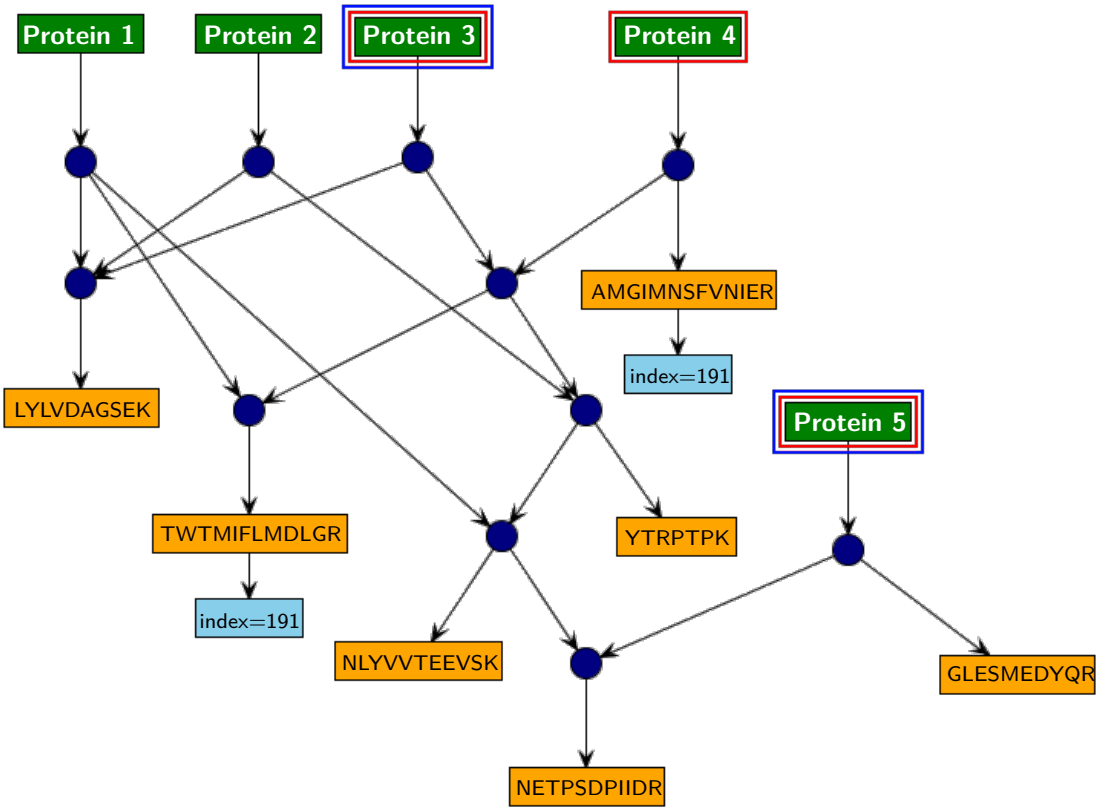


Figure 6.1: This figure highlights differences of the three inference methods in PIA. The connections between several accessions and peptides are shown. The spectra are not shown, except the one spectrum "index=191", which has identifications for two peptide sequences, which are found in seperate accessions. For simplicity, there is only one accession per group depicted.
Assuming no further filters, the "Report All" algorithm would report all five protein groups, each one with all its possible peptides and spectra. "Occam's Razor", acting as a parsimonious approach on the peptide level, would report the three groups marked by the red boxes. The remaining two groups containing *Protein 1* and *Protein 2* are both reported as sub-groups of the group containing *Protein 3*. Assuming that *Protein 3*'s group would get a better score with the peptide identified for the spectrum *index=191* than the group of *Protein 4*, the blue marked groups of *Protein 3* and *Protein 5* would be reported when using the "Spectrum Extractor".

Spectrum Extractor

The "Spectrum Extractor" is a spectrum-centric algorithm, in contrast to the two other implementations, which are peptide-centric. The major difference in this concept is that a spectrum, which gets assigned to a peptide once, never gets assigned to another peptide. This concept is closer to reality, as, in most cases, one MS/MS spectrum contains the fragment ion data of only one peptide, although this may not always receive the highest score by search algorithms. This inference method is very similar, although not equal, to the inference method called Protein

Extractor[116] implemented in the LIMS ProteinScape (Bruker, Bremen, Germany). If, instead of a score for a single PSM, a PSM set score (e.g., the *Combined FDR Score*) was selected as the base score for the inference, then the combined PSM sets from multiple search engine runs are used for the inference.

If only one peptide per spectrum is assigned, the algorithm's results are identical to the Occam's Razor. It should be considered, though, that also search engines like X!Tandem, which report only the best peptide match per spectrum, yield multiple peptides if the best score can be assigned to several sequences. As the occurrence of two peptides with similar scores in one spectrum reflect most probably not the reality, it is always recommended to use the Spectrum Extractor to get the most reliable protein results, unless the data was preprocessed in a way which deconvolutes multiple peptides per spectra.

The first step of the Spectrum Extractor is the creation of a protein group for each group in the PIA intermediate structure containing any accession. These groups contain every possible peptide of the respective group, regarding the selected filters. Afterwards, the following steps are performed:

1. For every protein group that has not yet been reported, examine each peptide. If a peptide is already reported, then allow it to be reported in this protein group with the prior set PSMs and score. Otherwise, construct the peptide with all still available PSMs fulfilling the given inference filters.

   If a spectrum is present in more than one peptide in a protein group, then use it for protein scoring only in the peptide where it has the best score.

   Should there be more than one peptide in a protein group for which the spectrum has the best score, collect all spectra that may account for the affected peptides. If there are peptides that are in all of the affected spectra, then one of these peptides is used with all of the spectra while scoring, and all other peptides are not considered during scoring. If the affected spectra are distributed over several peptides, calculate the score of these peptides without the questionable spectra. For this, it is important whether it was selected to score the peptide by each PSM or the best peptide only. The peptide with the best score gets all of its spectra assigned. If there are peptides with the same score and spectra, then all of their spectra are assigned, but only one is considered for protein scoring. Repeat these last steps until all spectra are assigned to peptides.

2. Calculate the score for each protein group, and select the group with the best score. Check whether this protein group is a subgroup of any already reported protein group regarding peptides or PSMs. If it is a same set (i.e., the protein groups contain the same PSMs and peptides) or sub protein group, then assign it to the respective group appropriately. If it is not, then add the protein group and all of its peptides and PSMs to the set of reported items and report this protein group.

3. Repeat steps 1 and 2 until there are no further protein groups to be reported.

Also this inference algorithm can be run parallelized. To do this, it is important to prior sort the clusters in the PIA intermediate structure in such a way, that clusters with not disjoint sets of spectra are binned together. These new bins can then be processed in parallel.

6.2.5 Visualisation of the Inference

PIA allows an intuitive visualisation of the relationship between proteins, peptides and PSMs of an analysed dataset. For this, a representation of the PIA intermediate structure (Sections 6.2.1 and 6.4.2) is used. Figure 6.1 shows such a representation of one cluster, i.e. one connected component in the intermediate structure. In this way the proteins, respectively their accessions, are shown on the top (green boxes) and connected via group nodes (blue circles) to peptides (orange boxes) and spectra (light blue boxes). An arrow towards another node represents a "belongs to" relationship. For example to *Protein 1* in Figure 6.1 belongs first a group node, to this three more group nodes, to each one of them belonging one peptide. To one of the groups belongs one more group node etc. To a peptide belongs always at least one spectrum node, to an accession always exactly one group node. A group node is necessary only to uphold the correct connections.

This representation of connections between the three layers gives an easy to comprehend overview, as long as the visualised connected component is not too complex. It is implemented into three of the available frontends of PIA: the KNIME nodes, web-frontend and the PRIDE Inspector implementation. In the web-frontend, only a static image for a given cluster is rendered, while in the other two implementations the JUNG2[117] (Java Universal Network/Graph Framework) framework was used, which allows manipulation of the visualisation. This includes zooming and panning as well as collapsing of spectrum, peptide and accession nodes to yield a more accessible overview. Regardless of the implementation, filtered out spectra and peptides are greyed out in the representation and, if any component was selected, the connected components in the current report are highlighted, showing e.g. exactly why a certain protein was reported and the reasons behind another not being reported, which for example can be due to filters or being a sub-protein.

6.3 Frontends for PIA

PIA is fully developed in Java, and all of its components can be used directly from the command line; thus, it can be integrated into any scripted identification pipeline. However, there are currently three more user friendly ways to conduct an analysis using PIA: a web frontend, KNIME nodes and the PRIDE Inspector.

6.3.1 The PIA Web Frontend

The web interface is written for JavaServer Faces (JSF), which requires a running installation of a JavaServer Pages web server (e.g., Apache Tomcat[118] or GlassFish Server[119]). The interface may then be accessed via any current browser either locally, via a network, or via the Internet from any modern computer. If the server is equipped with enough memory and hard drive space, this way of performing a very large PIA analysis is the most suitable. In any way, it is the best way to easily learn all concepts of PIA while conducting an analysis.

Project Management

As explained in the prior paragraphs, before performing an actual analysis, the data must be structured. The web frontend provides a special section for this, the project manager. In this context, a project refers to a compilation of search engine results into the PIA intermediate format. If a new project should be created, the user is presented with an interface where search engine result files can be uploaded and the project can be assigned with a descriptive name (Figure 6.2). If a Mascot server is accessible from the server's site, searches can be directly imported without the need to copy them on the local computer first. Usually the format of an imported file is assigned automatically, but can be adjusted if the detected format is not correct. Finally, the user can start the compilation.

A list of all finished and still running compilations can be shown. From this list all projects on the server can be accessed and further analysed.

Wizard Mode

The default and most intuitive procedure of an analysis using the web interface is the wizard (Figure 6.3). after selecting a project, the wizard assists a user through the default steps of an analysis by performing an FDR calculation on the PSM level, choosing a protein inference and scoring method, and performing the inference. For each step, the settings are suggested based on the imported data and preceding steps. Additionally, after each step, some descriptive statistics are shown. On these a basic quality check can be performed. The wizard can be aborted at any time, which directs the user to a more advanced interface.

Advanced Mode

In the advanced mode, all settings can be adjusted to the user's demands. The interface allows also for an in-depth inspection of the identification results, the results of the combination from different search runs, and the inferred peptides from the (combined) PSMs. For this, the listed results on each level can be expanded to show the results of the underlying level, as shown in Figure 6.4. For filtering and exporting the PSM, peptide, and protein lists, the user

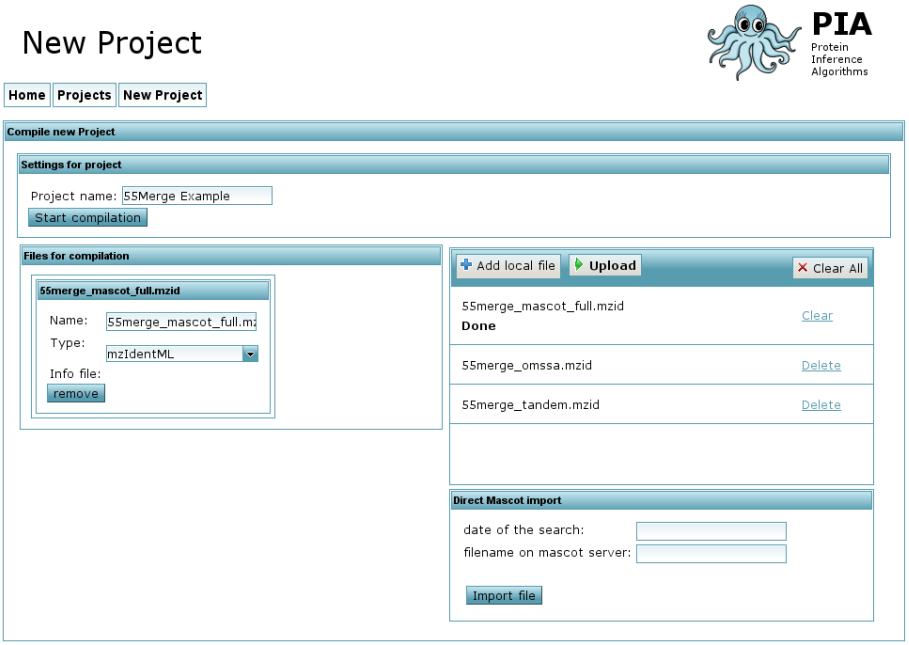


Figure 6.2: Screenshot showing the "New Project" screen of the web frontend. In this interface a user can easily upload search engine result files for compilation into the PIA intermediate format.

can select a variety of variables and thus filter for score, mass deviation, sequence, or other attributes. Furthermore, the visualisation described in Section 6.2.5 can be accessed on each level, highlighting the connections of the respective PSM, peptide or protein.

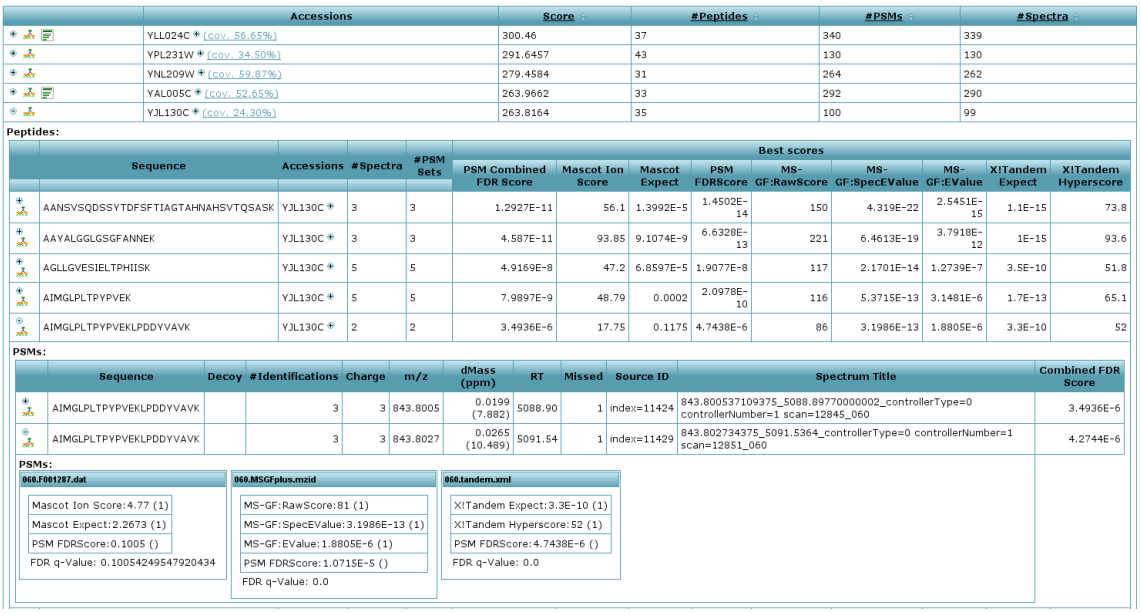
6.3.2 PIA KNIME Nodes

KNIME, the KoNstanz Information MinEr[96], is an open source workflow and data analysis environment. To integrate PIA into new or existing KNIME workflows, special nodes are developed and can be downloaded using the KNIME repositories. Besides PIA, there are several other nodes for bioinformatics tools. To integrate PIA into already existing proteomics workflows or facilitate the creation of new workflows, the PIA nodes are compatible with the respective OpenMS nodes for identification and quantification. Using an environment like KNIME has several benefits like the reusage of approved workflows, but also the reproducibility of conducted experiments.

Using the "Generic KNIME Nodes"(GKN[120,121]) it is possible to quickly create nodes which execute a command line tool and return the results as files. A first implementation for PIA into KNIME was created in this way. Treating PIA as command line tool only has several drawbacks, though. One of the major problems was the selection of the single analysis steps like FDR calculation, PSM filtering etc. These had to be concatenated, one node for each analysis step.

Figure 6.3: Screenshots of the web interface's wizard, the most convenient way to perform a PIA analysis. The wizard is part of the web interface and guides the user through the analysis while suggesting default values for most of the used parameters. The wizard starts with the calculation of the PSM FDR values for each search engine run (step 1) and shows statistics on these values such as the number of target and decoy PSMs as well as the distribution of mass deviations (step 2). In the third step, the PSMs are inferred to peptides, and an overview of the number of identifications per peptide is shown. In step four, the protein inference method is selected and processed. The final step 5, shows a short overview of the inferred proteins.

To prevent this and make a more user-friendly experience, a new set of PIA KNIME nodes was developed (Figure 6.5). All KNIME extensions are Java classes, which greatly facilitated an implementation of PIA. The new nodes consist of a "PIA Compiler" which performs the task of structuring search engine results into the intermediate format. This node takes a list of URLs to the file system, pointing to the search engine result files. The compiled XML file is directly stored as an compressed XML object inside the workflow. This file can be written to the file system for permanent storage or backup, either compressed or uncompressed. For a subsequent analysis using PIA, it can simply be passed to the "PIA Analysis" node.



	Accessions	Score	#Peptides	#PSMs	#Spectra
	YLL024C (cov. 56.65%)	300.46	37	340	339
	YPL231W (cov. 34.50%)	291.6457	43	130	130
	YNL209W (cov. 59.87%)	279.4584	31	264	262
	YAL005C (cov. 52.65%)	263.9662	33	292	290
	YJL130C (cov. 24.30%)	263.8164	35	100	99

Peptides:

Sequence	Accessions	#Spectra	#PSM Sets	Best scores: PSM Combined FDR Score	Mascot Ion Score	Mascot Expect	PSM FDRScore	MS-GF:RawScore	MS-GF:SpecEValue	MS-GF:EValue	X!Tandem Expect	X!Tandem Hyperscore
AANSVSQDSSYTDFSFTIAGTAHNAHSVTQSASK	YJL130C	3	3	1.2927E-11	56.1	1.3992E-5	1.4502E-14	150	4.319E-22	2.5451E-15	1.1E-15	73.8
AAYALGGLGSGFANNEK	YJL130C	3	3	4.587E-11	93.85	9.1074E-9	6.6328E-13	221	6.4613E-19	3.7918E-12	1E-15	93.6
AGLLGVESIELTPHIISK	YJL130C	5	5	4.9169E-8	47.2	6.8597E-5	1.9077E-8	117	2.1701E-14	1.2739E-7	3.5E-10	51.8
AIMGLPLTPYPVEK	YJL130C	5	5	7.9897E-9	48.79	0.0002	2.0978E-10	116	5.3715E-13	3.1481E-6	1.7E-13	65.1
AIMGLPLTPYPVEKLPDDYVAVK	YJL130C	2	2	3.4936E-6	17.75	0.1175	4.7438E-6	86	3.1986E-13	1.8805E-6	3.3E-10	52

PSMs:

Sequence	Decoy	#Identifications	Charge	m/z	dMass (ppm)	RT	Missed	Source ID	Spectrum Title	Combined FDR Score
AIMGLPLTPYPVEKLPDDYVAVK		3	3	843.8005	0.0199 (7.882)	5088.90	1	index=11424	843.800537109375_5088.89770000002_controllerType=0 controllerNumber=1 scan=12845_060	3.4936E-6
AIMGLPLTPYPVEKLPDDYVAVK		3	3	843.8027	0.0265 (10.489)	5091.54	1	index=11429	843.802734375_5091.5364_controllerType=0 controllerNumber=1 scan=12851_060	4.2744E-6

PSMs:

060.F001287.dat	060.MSGFplus.mzid	060.tandem.xml
Mascot Ion Score:4.77 (1)	MS-GF:RawScore:81 (1)	X!Tandem Expect:3.3E-10 (1)
Mascot Expect:2.2673 (1)	MS-GF:SpecEValue:3.1986E-13 (1)	X!Tandem Hyperscore:52 (1)
PSM FDRScore:0.1005 ()	MS-GF:EValue:1.8805E-6 (1)	PSM FDRScore:4.7438E-6 ()
FDR q-Value: 0.10054249547920434	PSM FDRScore:1.0715E-5 ()	FDR q-Value: 0.0
	FDR q-Value: 0.0	

Figure 6.4: Screenshot showing the inferred protein groups in the advanced mode. For each protein group, the accessions (with sequence coverage), the score, and the number of peptides, PSMs, and spectra are listed. Additionally, information of the underlying peptides as well as the (combined) PSMs can be shown by expanding the respective rows in the list.

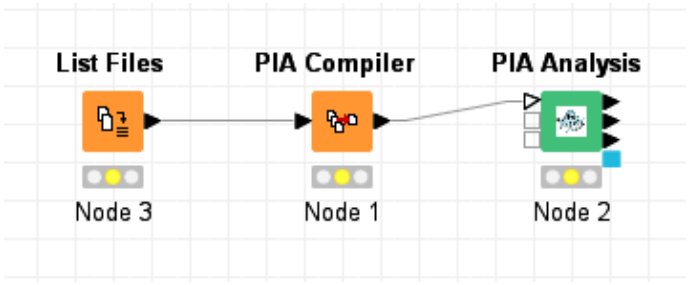


Figure 6.5: Screenshot showing the smallest possible KNIME workflow for a PIA analysis. *List Files* selects some search engine result files from the file system. These are merged by the *PIA Compiler* into the PIA intermediate format, compressed and stored inside the workflow. This compressed file can directly be passed to the *PIA Analysis*, which conducts the complete analysis.

The "PIA Analysis" node allows the user to adjust all possible PIA settings and filters in dialogs before running the analysis, like the wizard in the web frontend. The input file can either be passed directly from the "PIA Compiler" node or from an "Input File" node, which points to an intermediate XML file on the file system. After running the analysis three tables are generated, one for each of the PSM, peptide and protein levels. These results can directly be used by subsequent KNIME nodes. This way allows exporting the tables to CSV or XLS files using the respective KNIME nodes. Furthermore, the "PIA Analysis" node allows an export

to one of the file formats supported by PIA, including mzIdentML, mzTab and idXML, which facilitates further analysis outside the KNIME environment. An export of the results to an idXML file, which is OpenMS's intermediate format, enables the usage of the inferred proteins in quantitative workflows. The "PIA Analysis" node additionally contains a view, which allows an in-depth inspection of the results and the visualisation mentioned in Section 6.2.5.

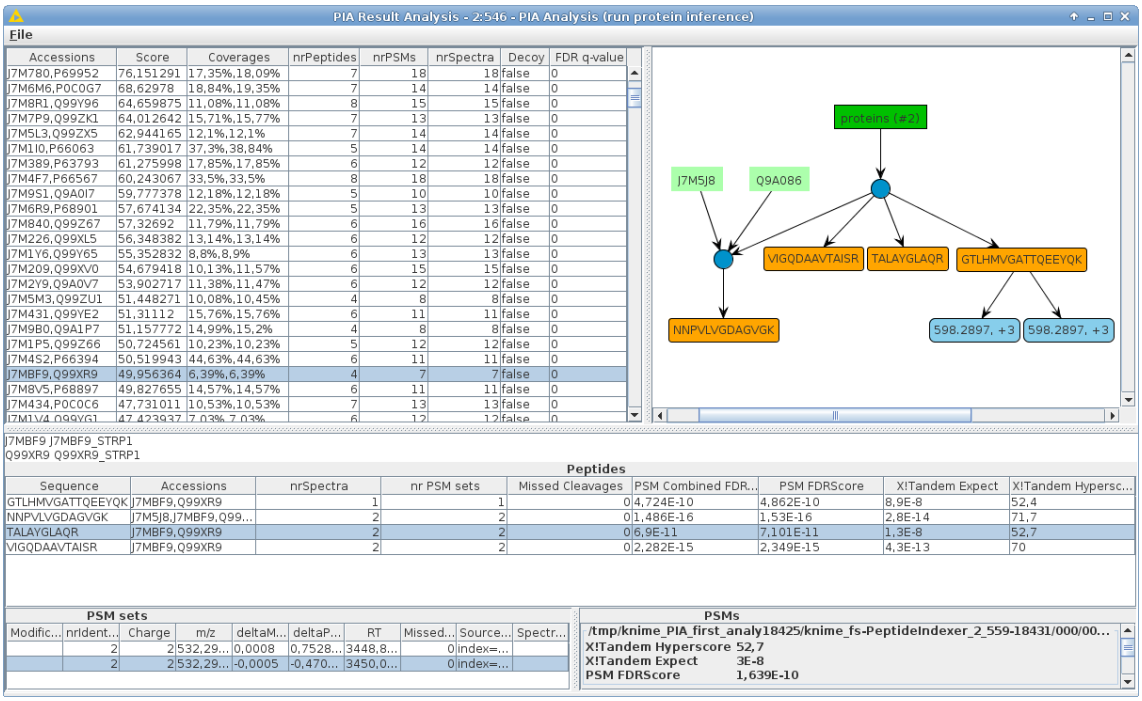


Figure 6.6: Screenshot of the "PIA Analysis" view in KNIME. This view allows an in-depth inspection of the results on the protein, peptide and PSM level as well as the visualisation mentioned in Section 6.2.5.

6.3.3 Implementation of PIA in PRIDE-Inspector

The PRIDE Archive (PRoteomics IDEntifications) is a centralised, standards compliant repository for MS proteomics experiment results[122]. This includes identification as well as quantification analyses. The ProteomeXchange project had as a goal the creation of one central point for MS proteomics submissions and thus the facilitation of the submission process for scientists[123]. Deposition of scientific data in a central repository has several benefits: firstly it allows the inspection of data connected to a publication and thus verification of the results by independent scientists. Currently this is required for an increasing number of journals. Secondly, it allows dissemination of the data for re-analysis with different questions or e.g. newer databases for identification. Additionally, the PRIDE repository allows the inspection of annotated spectra if a so called "complete submission" of a dataset was conducted. This eliminates the submission

of these annotations included into a manuscript. This was required by some journals and was a very tedious task in the past.

For the easy inspection of submitted data and to conduct some basic quality control tasks, the "PRIDE Inspector" was developed[124,125]. In the recent release of version 2.5, some parts of PIA were included to allow a basic inference for datasets containing peptide identifications only. This implementation is based on the ms-data-core-api[126], a Java library for all kinds of MS proteomics data and also the basis for most of PRIDE Inspector's functionality. Currently, only "Occam's Razor" is supported as inference algorithm, which uses all peptides in an opened file, regardless of the score or probability. Furthermore, the already mentioned visualisation for the relations between PMS, peptides and proteins is included.

6.3.4 Command Line Execution

The command line execution is meant for pipelining outside the KNIME environment. Basically, two Java classes of PIA are executable: the PIACompiler and the PIAModeller. The compiler is used to create the intermediate XML file, the modeller for every other aspect and is therefore the class which is executed by default when starting PIA's executable JAR file.

The **PIACompiler** has the following three arguments:

- **outfile** Path to the created PIA XML file
- **name** (optional) Name of the PIA compilation, used mainly internally
- **infile** Path to a search engine result file, this can be given multiple times to compile more than one file into the PIA-XML.

The following command line snippet shows, how a PIA compilation can be called on the command line:

Listing 6.1: Calling PIACompiler on the command line.

```
java -cp pia.jar de.mpc.pia.intermediate.compiler.PIACompiler \
        -infile /path/to/searchengine/result \
        [-infile /path/to/searchengine/result2... \]
        -name "name of compilation" \
        -outfile /path/to/compilation.pia.xml
```

Running a PIA analysis via the command line requires the creation of a parameter file, that describes the consecutive analysis steps. This file has to be in an XML format, loosely following the Common Tools Description (CTD) schema used by the OpenMS tools: for each analysis step there is an NODE element. The action is described by the name tag of the NODE. If the respective step needs further arguments, they are given by ITEM elements enclosed by the

respective NODE. Though such a pipeline described by an XML document could be created manually, it is recommended to use the respective KNIME nodes created by GKN and only perform small changes manually. Listing F.1 shows an example of such an pipeline XML file. This example conducts a default analysis: first the creation of PSM sets is activated and two scores, Mascot Ion score and X!Tandem expect, will be used for FDR estimation. The next setting instructs PIA to only use the top identification per spectrum for the FDR estimation, using "s.*" as the decoy-pattern. Then, the FDR for all input files will be calculated followed by the *Combined FDR Score*. For the peptide level there is only the setting whether modifications are considered for the peptide inference, which is turned off in the given sample. On the protein level, one inference filter is added, to use PSMs with an *Combined FDR Score* ≤ 0.01 (LEQ = "less or equal"). Finally, the protein inference using the *Spectrum Extractor* with multiplicative scoring using only the best PSM per peptide and the *Combined FDR Score* as base score is performed.

The PIAModeller class, which is called to perform an analysis on the command line, needs at least two parameters (infile and paramFile) and to actually write out any results it requires at least one export parameter. The recommended parameters for usage are:

- **infile** Path to the used PIA-XML file, which was created by the PIACompiler
- **paramFile** Path to the parameter XML file, which should be executed. This can also be created or extended, but manual usage of this is discouraged and should be performed by the GKN KNIME nodes only.
- **execute** (optional) Execute the parameter file given by paramFile. This is the default behaviour and does not need to be stated.
- **psmExport** `outfile format [fileID spectralCount]` (optional)
  Exports the analysis on the PSM level. The results on the PSM level file will be written to outfile in the format specified by format (for supported formats see Section 6.4.2). The fileID specifies, whether the overview/merge (0, default) or the PSMs of only one input file should be exported. If a CSV export is performed, with spectralCount set to "yes" the export will be in a spectral count friendly format, i.e. each accession of a PSM will be exported into a separate line.
- **peptideExport** `outfile format [fileID exportPSMs exportPSMSets oneAccessionPerLine]` (optional)
  Exports the analysis on the peptide level to the given file path specified by format (see Section 6.4.2). The fileID specifies, whether the overview/merge (0, default) or the peptides of only one input file should be exported. Setting exportPSMs or exportPSMSets to "yes" (default "no") allows to also export the PSMs and PSMSets of the peptides, if this

is optional for the selected format. Setting oneAccessionPerLine to "yes" writes for each accession the whole information into one row (for CSV only).

- **proteinExport** `outfile format [exportPSMs exportPSMSets exportPeptides oneAccessionPerLine]` (optional)
  Exports the analysis on the protein level to the file path given by outfile in the format specified by format (see Section 6.4.2). Setting exportPSMs, exportPSMSets or exportPeptides to "yes" (default "no") allows the export of the PSM, PSMSet or peptide information of the proteins, if this is optional for the selected export format. Setting oneAccessionPerLine to "yes" writes for the CSV export for each accession the complete exported information into one line.

6.4 Technical Details of the Implementation

The implementation of PIA is divided into three parts: (1) the core with all algorithms, visualisation and the command line execution as the base project and based on it (2) the KNIME nodes and (3) the web frontend using JavaServer Faces. For an overview of the architecture, see Figure 6.7. All code of PIA is open source and licensed under a Three-Clause-BSD-License. At the time of writing this theses, all of the source code was deposited at GitHub. The following paragraphs highlight some technical and programmatic details of the respective implementations.

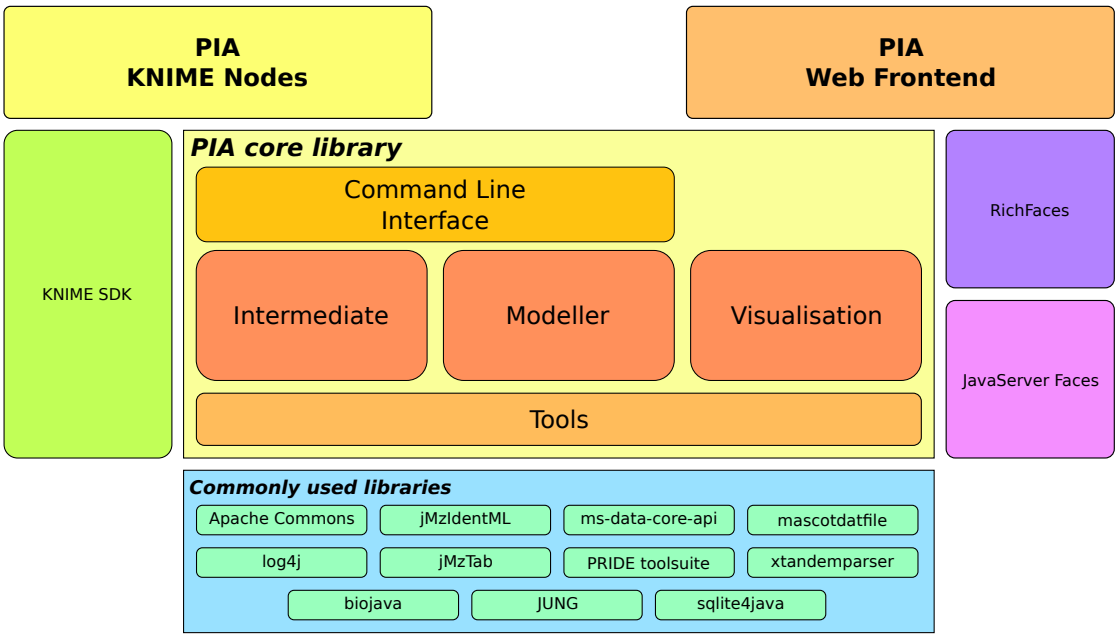


Figure 6.7: Architecture of PIA.

6.4.1 Commonly used Libraries

For many commonly used tasks, PIA used available open source libraries. Two widely used libraries used throughout the code are not restricted to biology or proteomics tasks: The *Apache Commons* project provides a variety of libraries for many reusable Java components. These are for example used for general String manipulations, parsing of command line arguments and escaping of XML tags. As an easy to implement and configurable logging interface, *log4j* is used. All other external libraries will be further discussed in their context in the following paragraphs.

6.4.2 The PIA Core Library

PIA's core is completely written in Java and all of its functionality is compatible with the Java 1.6 API. The main project is hosted at `https://github.com/mpc-bioinformatics/pia`. It uses Apache Maven [127] for the management of dependencies and the building process. Maven allows the developer to define the required dependencies of a project and their exact versions. These dependencies are usually stored on Maven servers and can be accessed during the build process. Thus, shipping of all required dependencies with the source code is not needed. Nor must a developer take care of downloading them manually before building the project.

The source of the core library is structured into four main Java packages: `intermediate`, `modeller`, `tools` and `visualisation` as shown in Figure 6.7. The `intermediate` package contains all functionality to create, save and read the PIA intermediate structure. Furthermore, all importers are stored in this package. In the `modeller` package, almost all the remaining logic required for a PIA analysis is stored, like the handling of the three layers, the scoring, filtering and the actual protein inference. Routines to correctly parse controlled vocabularies (CVs), PRIDE and OpenMS files, as well as special logic for standard formats and constants are defined in the `tools` package. Finally, all classes concerned with any kind of visualisation are located in the `visualization` package.

Importers

Before an actual analysis can be conducted, the results of one or more search engine runs must be imported. PIA's main source of information are the peptide spectrum matches and the associated accessions. To create a PSM its charge, experimental m/z value, the peptide sequence, any modifications and scores and the respective set of accessions are essential. Furthermore, the delta between measured and theoretical mass, retention time, number of missed cleavages, titles of the matched spectra and their ID in the original spectrum file as well as the information, whether the PSM matches a decoy, can be extracted from the search results

and used for the analysis. For an accession, only the actual ID is mandatory, but a description and the protein sequence is also imported, if available.

Currently there are several different importers for native and processed search results, supporting the following file formats:

- **Mascot DAT**
  For the import of Mascot results the MascotDatfile[128] library is used. The DAT format used by Mascot is plain text, which can be easily accessed and contains all the information, which PIA is able to process.

- **X!Tandem XML**
  The XML files created by a X!Tandem search are an extension of the the specifications of the Biopolymer Markup Language (BIOML) and the General Analytical Markup Language (GAML). These files also contain all possible information PIA can use. The TandemParser[129] is used to parse these information.

- **Thermo MSF**
  Thermo's ProteomeDiscoverer uses so called "Magellan Storage Files" (MSF) to save the search results, which actually are SQlite databases. To extract the PSM information of these files, a not-published library, developed at the MPC, is used. This library can handle files created by ProteomeDiscoverer 1.2 - 1.4 and was tested for searches running Mascot, SEQUEST and MS Amanda.

- **Tide TXT**
  Tide[56] is a newer implementation of the original SEQUEST algorithm, which can process database searches much faster than the original implementation. One of the supported export formats is a plain text format, containing only the mandatory information for peptide spectrum matches.

- **PRIDE XML**
  PRIDE XML was the original format for all uploads submitted to the PRIDE repository before the mzIdentML standard was developed. Unfortunately it has some shortcomings concerning the grouping of proteins. Only the final results given by a converter are saved and therefore many PSMs are filtered out, as are accessions contained in sub-groups of proteins. But if no protein inference was conducted before storing a PRIDE XML file, most information is still contained in the file. The pride-jaxb library developed by the EBI is used to parse PRIDE XML files.

- **idXML**
  OpenMS uses idXML as an internal format. All search engine adapters and identification preprocessing tools developed by the OpenMS team export to this format. To facilitate

seamless integration of PIA into OpenMS workflwos an importer and exporter using Java's JAXB architecture was developed. There are plans to extract this functionality in the future into a dedicated library and give a Java parser for idXML files to the community.

- **mzIdentML**
  MzIdentML is the HUPO-PSI format for all identification results produced by MS proteomics. With the possibility to import this format, virtually all search engines are supported by PIA. For parsing, the jMzIdentML library[130] was used.

To permit a PSI standard compliant export of a PIA analysis, not only the search results are imported, but also some information about how a search was conducted. This includes all settings provided by the respective format like the processed protein database, allowed missed cleavages and modifications, the enzyme used for digestion etc.

Compilation of Search Engine Results

To allow fast access to the data used in an analysis, it is structured into a tree-like graph by the `PIACompiler` class. To achieve this, the PSMs are assigned to their peptides, defined by their amino acid sequence, after collecting data from all search engine runs. While doing so, a map from the peptides to the proteins' accessions is built to accelerate subsequent evaluations (Figure 6.8a). Next, all PSMs and peptides in the map are structured into clusters, which form maximal connected sets with their mapped proteins/accessions, i.e., all data in one cluster has no connection to any other cluster (Figure 6.8b). These sets can be subsequently processed in parallel to consecutively insert each peptide into its corresponding acyclic graph compartment along with its protein accessions. The graph is constructed in a straightforward way and consists of nodes for proteins, peptides with their PSMs, and additional group nodes (Figure 6.8c). The group nodes connect the protein and peptide nodes such that the following rules are valid:

1. Each peptide and each protein belongs exactly to one group,
2. a group can have other groups as children,
3. there are no circles in the graph, even with respect to the (undirected) group-group relations,
4. there is exactly one path from each protein to its peptides (with PSMs) and vice versa, which allows the relations between proteins and peptides/PSMs to be retrieved rapidly.

Each disjoint cluster forms a separate tree in the created graph and is the basis for PIA's visualization (see Section 6.2.5). After the compilation is finished, the graph data is stored in an XML file.

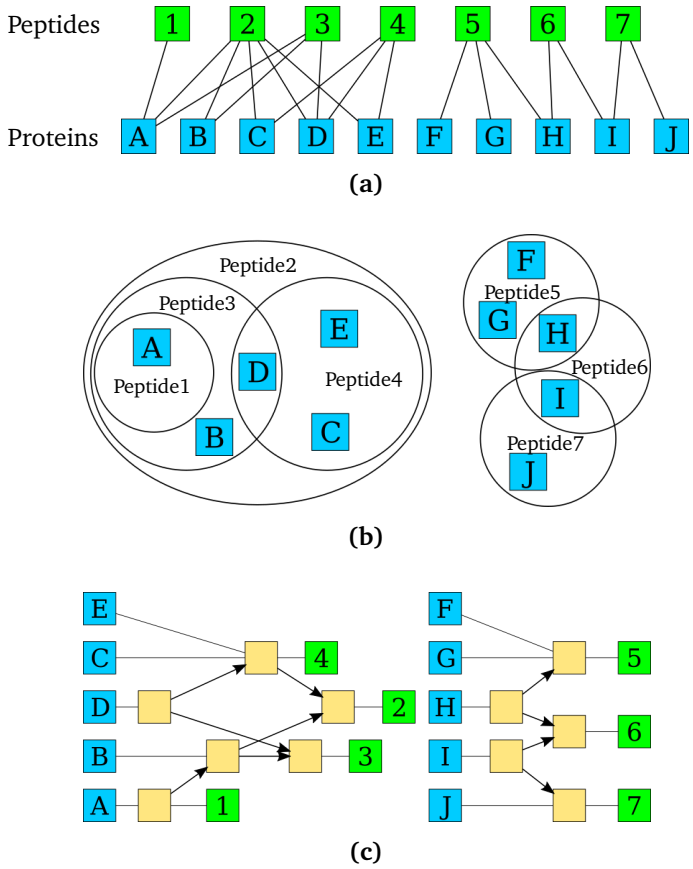


Figure 6.8: Compilation of the search engine results into a directed graph is performed in three steps. PSMs can be easily grouped into peptides according to their amino acid sequences; therefore, PSMs are left out in this figure. The connection information between peptides (green) and proteins (blue) is stored in a map shown in (a), where each peptide belongs to one or more proteins. This map can be divided into closed clusters, where each peptide maps to all its proteins and there is no mapping from one cluster to any other, as depicted by two such clusters in (b). The information on these closed clusters can be processed in parallel to create a set of acyclic graphs shown in (c), where it is easy to retrieve for a given protein all peptides (and PSMs) or vice versa by following the connections between nodes. This data structure is the actual intermediate format used by PIA to quickly retrieve information. The yellow group nodes store no additional information, but they are necessary to connect the remaining nodes correctly and to uphold the set of rules given in the text (see Section 6.4.2).

Structure of the Intermediate PIA XML File

The compiled identifications are stored in an XML file. This XML file is created, and subsequently loaded for an analysis, using JAXB. Listing F.2 shows an example of a PIA intermediate file. The structure is split into five parts: input information, the PSMs, accessions, peptides and groups. The input information is mainly used to carry settings, which are needed to create a standard compliant export after an analysis. It consists of three lists: `<filesList>` contains

information of the actual compiled search result files, <Inputs> of the matched databases and <AnalysisSoftwareList> of the softwares used for identification and processing prior to the compilation. For all these information, mainly tags also used in mzIdentML are used. This greatly facilitates an export, for example the <Inputs> and <AnalysisSoftwareList> can be copied to an mzIdentML file, as can most of the tags included in a <file> element.

The <spectraList> holds a collection of <spectrumMatch> elements. These elements represent one PSM each, containing all mandatory and optional information discussed in the importers section (Section 6.4.2). Additionally, it maps via the fileRef argument to the actual search engine results file it originates from. Accessions of all compiled files are listed in the <accessionsList>. If an accession was defined in more than one imported file, it has multiple <FileRef> tags. It may also contain multiple Description and SearchDatabaseRef tags, if these were defined. The <Sequence>, though, is always unique and warnings are given, if data with differing sequences for the same accession are imported. The <peptide> tags inside the <peptidesList> contain, besides their sequence, references to all associated PSMs. Furthermore, the respective accessions are linked and the actual position of the peptide sequence in the protein sequence, if this was reported by the search engine.

To completely represent the directed graphs of the generated intermediate structure, <group> tags are needed. These are collected inside the <groupList>. Each group may be connected to any number of accessions, peptides and other groups, as explained in the previous section concerning the compilation of the intermediate structure.

Filters

Filters are one of the most important settings in a PIA analysis. They work on either of the three levels of PIA and have a big impact on the outcome of the protein inference. They are implemented inside the modeller package and are, except for the PSM and peptide level score filters and the PSMTopIdentificationFilter, instantiated by the RegisteredFilters class. This class is an enumeration and returns appropriately initialized instances of SimpleTypeFilters, a template class of the basic AbstractFilter. For the score filters and the PSMTopIdentificationFilter additional settings of the basis score are necessary, therefore they have special classes.

All instantiations of a filter need a comparator (greater than, equal, contains, regular expression...), the compared argument (a numerical or string value) and whether the filter should pass the respective objects fulfilling the comparator and argument or the complement. Based on these settings filters can be created, to either refine the PSMs and peptides passed to an inference algorithm or the elements passed to the creation of a report.

Table 6.1 shows an overview of the filters implemented at the time of writing this thesis. The names of the filters should suffice to explain their filtering behaviour.

Table 6.1: The currently implemented filters on the three levels.

PSM Level Filters	Peptide Level Filters	Protein Level Filters
PSM scores	peptide scores	PROTEIN_SCORE_FILTER
CHARGE_FILTER	PEPTIDE_ACCESSIONS_FILTER	PROTEIN_RANK_FILTER
DELTA_MASS_FILTER	PEPTIDE_DESCRIPTION_FILTER	NR_UNIQUE_PEPTIDES_PER_PROTEIN_FILTER
DELTA_PPM_FILTER	PEPTIDE_FILE_LIST_FILTER	NR_GROUP_UNIQUE_PEPTIDES_PER_PROTEIN_FILTER
MZ_FILTER	PEPTIDE_MISSED_CLEAVAGES_FILTER	PROTEIN_ACCESSIONS_FILTER
PSM_ACCESSIONS_FILTER	PEPTIDE_MODIFICATIONS_FILTER	PROTEIN_DESCRIPTION_FILTER
PSM_DESCRIPTION_FILTER	PEPTIDE_SEQUENCE_FILTER	PROTEIN_FILE_LIST_FILTER
PSM_FILE_LIST_FILTER	PEPTIDE_SOURCE_ID_LIST_FILTER	PROTEIN_MODIFICATIONS_FILTER
PSM_MISSED_CLEAVAGES_FILTER	PEPTIDE_UNIQUE_FILTER	PROTEIN_SEQUENCE_LIST_FILTER
PSM_MODIFICATIONS_FILTER	NR_PSMS_PER_PEPTIDE_FILTER	NR_PEPTIDES_PER_PROTEIN_FILTER
PSM_RANK_FILTER	NR_SPECTRA_PER_PEPTIDE_FILTER	NR_PSMS_PER_PROTEIN_FILTER
PSM_SEQUENCE_FILTER		NR_SPECTRA_PER_PROTEIN_FILTER
PSM_UNIQUE_FILTER		
PSM_SOURCE_ID_FILTER		
NR_ACCESSIONS_PER_PSM_FILTER		
NR_PSMS_PER_PSM_SET_FILTER		

Export

PIA allows exporting the PSM, peptide and protein results into different file formats for further processing or storage of the results. Each level allows the export into a basic CSV format, that can easily be processed by R or Excel, but also to the PSI standardised mzTab format. The PSM and protein level allow additional export into mzIdentML, which has no native concept for peptides at the moment. As explained in Section 3.5.2, mzIdentML has an elaborate framework to report protein ambiguities and groups, which is supported by the PIA exports. The export of peptide level information to both PSI formats is work in progress at the time of writing the thesis. A full support is planned for the version 1.2. of the mzIdentML standard.

To improve the interoperability between PIA and OpenMS, also an idXML exporter is available. These exports contain `indistinguishable_group` tags introduced in OpenMS 2.0 and thus the created files can be used as input for the *ProteinQuantification* tool of OpenMS, to set the reported groups.

6.4.3 KNIME Nodes

The KNIME-SDK, a special version of the Eclipse IDE[131], allows the creation and testing of nodes. A node must be embedded in a plug-in project and consists, besides some files for the project's settings, of at least four classes, the `NodeDialog`, `NodeFactory`, `NodeModel` and `NodeView`. The `NodeDialog` handles the setting of input parameters before running the node, the `NodeModel` includes all the logic as well as the storage and retrieval of settings and special objects created by the node, the `NodeView` allows visualising the results and the `NodeFactory` is an entry point which provides KNIME with all necessary information about the node.

As PIA is completely written in Java, all of its functionality could straight forward be implemented into the KNIME environment. To install new plug-ins into KNIME, repository sites containing the latest builds of the extensions can be contacted. PIA currently resides on the official "trunk" or "nightlies" repository of KNIME and thus allows an easy installation and updating by the user. To ensure continuous integration of new features, the source code of the PIA KNIME nodes is fetched each night from the GitHub repository by the build servers of KNIME, tested and, if no errors occurred, it is deposited in the repository.

6.4.4 Web Frontend using JavaServer Faces

The web frontend uses JavaServer Faces (JSF) to combine a platform independent visualisation with Java libraries running in the background. To create a modern interface and enable some special features like the uploading of files and full Ajax support, the open source framework

library RichFaces is used. To run a JSF application, a servlet container like Apache Tomcat must be running on the server.

A JavaServer Faces application uses so called beans to handle the logic behind a rendered web page. PIA's web frontend consists of only three beans: the `ViewerBean`, the `CompilerBean` and a `CompilationManager`. The `ViewerBean` manages loading of a compiled PIA project and all the modelling steps needed by a PIA analysis. This includes the visualisation of the three layers, mainly with lists, but also tasks like the FDR estimation and starting the actual protein inference and the setting of filters. Instances of the `CompilerBean` contain the logic for the graphical interface behind the creation of a new PIA compilation. Both beans are `session scoped`, which means they are instantiated once for a user calling the respective page via a web browser. This also controls, that a user can open only one project at a time. The `CompilationManager` on the other hand is started as soon as the servlet container starts the application and has only one instance during the whole runtime of the server. The manager is called to list the projects stored on the server and schedules the actual compilations. This ensures, that only a limited number of compilations are run in parallel to prevent the application from using up too much main memory.

Chapter 7

Conclusion and Outlook

Protein inference, the step of creating a list of proteins from peptide spectrum matches, remains a challenge in current mass spectrometry based bottom-up proteomics. There is no solution for this problem, only different ways on how to infer highly reliable protein results are possible. Therefore, it is important to always keep this problem in mind when analysing the results of a proteomics experiment and always think of protein groups - maybe their representatives - instead of single proteins in a report. The only way to circumvent this altogether is to use top-down MS proteomics. These methods are evolving and might replace the bottom-up approaches in the future. This will have several inherent benefits like the possibility to distinguish protein isoforms more accurately and also make most protein inference strategies obsolete. At the time of writing this thesis, the machines need much more improvement to allow high-throughput mass spectrometry proteomics of complex samples, though.

Though the approaches of bottom-up proteomics are improved greatly in recent years, there are still several challenges. One of the main issues LC-MS faces is, that it is not possible to collect MS or MS/MS data of each peptide in a sample. Even if a given peptide is present in the sample, it might not be detected due to insufficient ionisation. The current method of choice for the identification in untargeted experiments is the data dependent acquisition. A peptide, respectively its m/z value, which was selected for fragmentation is usually excluded for the next 10–30 seconds. While this excludes the re-measuring of high intensive peptides, it can be observed, that a peptide is often measured at the very beginning of its ion trail and after its peak was reached. This forfeits the possibility to measure a more intense and thus better suited MS/MS spectrum at the highest intensity. Another problem is the fact, that the window for the selection of parent ions is relatively large and thus it is possible to accidentally fragment the ions of multiple peptides at once. These chimeric spectra are usually hard to match to peptides, as many of the m/z peaks are interpreted to be noise by several search engines. An emerging trend at the time of writing is *data independent acquisition* (DIA) or *SWATH*, a technique in which relatively large, static fragmentation windows are used to fragment all

ions in the analysed m/s range. These experiments are still often bottom-up, but in theory all peptides should be fragmented at (almost) each retention time. The identification of the resulting chimaric MS/MS spectra though is still a big issue.

In this thesis I presented a new protein inference engine called "PIA - Protein Inference Algorithms", which is open-source and completely written in Java. Besides the usage of PIA from the command line, three more user friendly methods were implemented: a web-based user interface, the implementation into PRIDE Inspector and the development of a KNIME node. All of these graphical frontends allow a browsable in-depth analysis of the results on the PSM, peptide and protein level. Furthermore, an intuitive visualisation of the relations between these three layers allows an explanation of the reasoning for reporting a certain protein or discarding the evidence of another. This also allows to inspect the complex relations when analysing protein isoforms, as shown in Section 5.4.

Currently, three different inference methods are implemented, which can be combined with different scoring algorithms and a multitude of filters. PIA supports the native formats of several search engines as well as the standard format mzIdentML as input. The generated protein lists can either be stored into easily parsable formats like CSV and mzTab, but also in a more comprehensive way into the community standard mzIdentML. PIA also includes an implementation for merging peptide spectrum matches obtained from multiple search engines, but can additionally import merges from other tools. One of the basic principles of PIA is the fact, that it fully supports protein ambiguity groups and sub-groups instead of reporting single accessions.

In Chapter 5, PIA and four other protein inference methods were assessed. The analysis showed, that the assessed algorithms agree on most of the reported protein groups, though there are several differences between them. Some of the algorithms strongly depend on the underlying protein database complexity, especially the Bayesian approaches. PIA, as one of the parsimonious approaches, is relatively robust against these changes in complexity. Additionally it was highlighted, that the combination of search results from multiple peptide identification algorithms increases the overall reported number of proteins as well as their quality, regarding the number of peptides per protein. It was also shown, that the FDR as well as the q-value calculated by the target-decoy-approach, under certain circumstances, cannot control the false discovery rate sufficiently. On the set of given metrics, PIA performed very well and in most cases outperformed the other methods. Furthermore, most of the analysed implementations for protein inference have some restrictions on the data input. They either need the PSM data in a special format or allow only one type of score, respectively probability. PIA on the other hand has no restrictions on the used scores, as long as they are in the PSI ontology.

One drawback of the current implementation though is the parsing of the intermediate PIA XML files. As these can become very large, especially when combining many search files, the memory consumption rises accordingly. This, though, could be compensated with a new parser,

which allows for indexing and accessing only the data, that is needed at a given time point. This could also be achieved with the usage of a graph-database as storage for the intermediate data instead of an XML format.

To give scientists using PIA an even bigger insight into the data, it is planned to enhance the visualisation with a spectrum viewer. This could be achieved by connecting PIA with the files containing the originally identified spectra. Furthermore, it would be possible to implement a connection to databases containing e.g. gene-ontology (GO) terms and visualise these in the generated outputs.

Also the methods for protein inference could be improved with further knowledge. For example the peptide quantities could be used to improve not only the inferred proteins, but also the results of protein quantifications, as the assignment of quantities of shared peptides is still an unsolved problem in MS-based label free quantification.

These results show that PIA as a freely available open-source tool can valuably contribute to the field of proteomics. In combination with the workflow environment KNIME, it allows the creation and execution of any kind of complex workflow, including any pre- or post-processing of the results. Thus, PIA does not entirely solve the problem of protein inference, which might never be possible for bottom-up MS proteomics due to shared peptide sequences. But it is a versatile tool, that can be used in many ways and workflows, to help scientists working in MS proteomics to create high-quality protein results, using the peptide identifications originating from virtually any spectrum identification algorithm.

Bibliography

[1] Uszkoreit J., et al. (2015). PIA: An intuitive protein inference engine with a web-based user interface. *Journal of Proteome Research*, 14(7):2988–2997. ix, 45, 81, 123

[2] Audain E. and Uszkoreit J., et al. (2017). In-depth analysis of protein inference algorithms using multiple search engines and well-defined metrics. *Journal of Proteomics*, 150:170–182. ix, 45, 123

[3] Watson J. D. and Crick F. H. (1953). Molecular structure of nucleic acids. *Nature*, 171(4356):737–738. 1

[4] Human Genome Sequencing Consortium International (2004). Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945. 1

[5] Mardis E. R. (2008). The impact of next-generation sequencing technology on genetics. *Trends in Genetics*, 24(3):133–141. 1

[6] Miller N. A., et al. (2015). A 26-hour system of highly sensitive whole genome sequencing for emergency management of genetic diseases. *Genome Medicine*, 7(1). 1

[7] The 1000 Genomes Project Consortium (2010). A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–1073. 1

[8] The 1000 Genomes Project Consortium (2012). An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65. 1

[9] International Human Genome Sequencing Consortium (2004). Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945. 1

[10] Ezkurdia I., et al. (2014). Multiple evidence strands suggest that there may be as few as 19 000 human protein-coding genes. *Human Molecular Genetics*, 23(22):5866–5878. 1

[11] The C. elegans Sequencing Consortium (1998). Genome sequence of the nematode C. elegans: A platform for investigating biology. *Science*, 282(5396):2012–2018. 1

[12] Smith L. M., et al. (2013). Proteoform: a single term describing protein complexity. *Nat Meth*, 10(3):186–187. 1

[13] Wasinger V. C., et al. (1995). Progress with gene-product mapping of the Mollicutes: Mycoplasma genitalium. *ELECTROPHORESIS*, 16(1):1090–109. 2

[14] Chick J. M., et al. (2015). A mass-tolerant database search identifies a large proportion of unassigned spectra in shotgun proteomics as modified peptides. *Nat Biotechnol*, 33(7):743–749. 2

[15] Eng J. K., McCormack A. L., and Yates J. R. (1994). 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*, 5(11):976–989. J Am Soc Mass Spectrom. 2, 22

[16] Perkins D. N., Pappin D. J., Creasy D. M., and Cottrell J. S. (1999). Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20(18):3551–67. 2, 23

[17] Craig R. and Beavis R. C. (2004). TANDEM: matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–7. 2, 23

[18] Bandeira N. (2007). Spectral networks: a new approach to de novo discovery of protein sequences and posttranslational modifications. *Biotechniques*, 42(6):687, 689, 691 passim. 2, 20

[19] Bertsch A., et al. (2009). De novo peptide sequencing by tandem MS using complementary CID and electron transfer dissociation. *Electrophoresis*, 30(21):3736–47. 2, 20

[20] Wolters D. A., Washburn M. P., and Yates J. R. (2001). An automated multidimensional protein identification technology for shotgun proteomics. *Anal Chem*, 73(23):5683–90. 2, 9

[21] Serwe M., Blüggel M., and Meyer H. E. *Microseparation Techniques V: High Performance Liquid Chromatography*, pages 67–85. Wiley-VCH Verlag GmbH (2007). 2, 9, 11

[22] Nesvizhskii A. I. and Aebersold R. (2005). Interpretation of shotgun proteomic data: the protein inference problem. *Mol Cell Proteomics*, 4(10):1419–40. 3, 29

[23] Nesvizhskii A. I., Keller A., Kolker E., and Aebersold R. (2003). A statistical model for identifying proteins by tandem mass spectrometry. *Anal Chem*, 75(17):4646–58. 3, 54

[24] Searle B. C. (2010). Scaffold: a bioinformatic tool for validating MS/MS-based proteomic studies. *Proteomics*, 10(6):1265–9. 3

[25] Ma Z. Q., et al. (2009). IDPicker 2.0: Improved protein assembly with high discrimination peptide identification filtering. *J Proteome Res*, 8(8):3872–81. 3

[26] Eisenacher M., et al. (2012). Search and decoy: the automatic identification of mass spectra. *Methods Mol Biol*, 893:445–88. 3, 27

[27] Jones A. R., Siepen J. A., Hubbard S. J., and Paton N. W. (2009). Improving sensitivity in proteome studies by analysis of false discovery rates for multiple search engines. *Proteomics*, 9(5):1220–9. 3, 26, 61, 82

[28] Nahnsen S., Bertsch A., Rahnenführer J., Nordheim A., and Kohlbacher O. (2011). Probabilistic consensus scoring improves tandem mass spectrometry peptide identification. *J Proteome Res*, 10(8):3332–43. 27, 56

[29] Kwon T., Choi H., Vogel C., Nesvizhskii A. I., and Marcotte E. M. (2011). MSblender: A probabilistic approach for integrating peptide identifications from multiple database search engines. *J Proteome Res*, 10(7):2949–58. 3, 82

[30] Ong S.-E., et al. (2002). Stable isotope labeling by amino acids in cell culture, SILAC, as a simple and accurate approach to expression proteomics. *Molecular & Cellular Proteomics*, 1(5):376–386. 8

[31] Shevchenko A., Tomas H., Havlis J., Olsen J. V., and Mann M. (2007). In-gel digestion for mass spectrometric characterization of proteins and proteomes. *Nat. Protocols*, 1(6):1754–2189. 8

[32] Tanca A., Biosa G., Pagnozzi D., Addis M. F., and Uzzau S. (2013). Comparison of detergent-based sample preparation workflows for LTQ-Orbitrap analysis of the Escherichia coli proteome. *PROTEOMICS*, 13(17):2597–2607. 8

[33] Wiśniewski J. R., Zougman A., Nagaraj N., and Mann M. (2009). Universal sample preparation method for proteome analysis. *Nature Methods*, 6(5):359–362. 8

[34] Glatter T., et al. (2012). Large-scale quantitative assessment of different in-solution protein digestion protocols reveals superior cleavage efficiency of tandem lys-c/trypsin proteolysis over trypsin digestion. *J. Proteome Res.*, 11(11):5145–5156. 8

[35] Alpert A. J. (2008). Electrostatic repulsion hydrophilic interaction chromatography for isocratic separation of charged solutes and selective isolation of phosphopeptides. *Anal. Chem.*, 80(1):62–76. 10

[36] Tanaka K., et al. (1988). Protein and polymer analyses up to m/z 100,000 by laser ionization time-of-flight mass spectrometry. *Rapid Commun. Mass Spectrom.*, 2(8):151–153. 10

[37] Karas M. and Hillenkamp F. (1988). Laser desorption ionization of proteins with molecular masses exceeding 10,000 daltons. *Anal. Chem.*, 60(20):2299–2301. 10

[38] O'Farrell P. H. (1975). High resolution two-dimensional electrophoresis of proteins. *Journal of Biological Chemistry*, 250(10):4007–4021. 10

[39] Rappsilber J., Moniatte M., Nielsen M. L., Podtelejnikov A. V., and Mann M. (2003). Experiences and perspectives of MALDI MS and MS/MS in proteomic research. *International Journal of Mass Spectrometry*, 226(1):223–237. 10

[40] Nobel Media AB 2014. The Nobel Prize in Chemistry 2002. `http://www.nobelprize.org/nobel_prizes/chemistry/laureates/2002/`. [Online; accessed 28.04.2016]. 11

[41] Rayleigh L. (1882). On the equilibrium of liquid conducting masses charged with electricity. *Philosophical Magazine Series 5*, 14(87):184–186. 11

[42] Yost R. A. and Enke C. G. (1978). Selected ion fragmentation with a tandem quadrupole mass spectrometer. *Journal of the American Chemical Society*, 100(7):2274–2275. 12

[43] Kingdon K. H. (1923). A method for the neutralization of electron space charge by positive ionization at very low gas pressures. *Physical Review*, 21(4):408–418. 12

[44] Makarov A. (2000). Electrostatic axially harmonic orbital trapping: a high-performance technique of mass analysis. *Analytical Chemistry*, 72(6):1156–1162. 12

[45] Hu Q., et al. (2005). The Orbitrap: a new mass spectrometer. *J. Mass Spectrom.*, 40(4):430–443. 12

[46] Scheltema R. A., et al. (2014). The q exactive HF, a benchtop mass spectrometer with a pre-filter, high-performance quadrupole and an ultra-high-field orbitrap analyzer. *Molecular & Cellular Proteomics*, 13(12):3698–3708. 12

[47] Gross J. H. Isotopic composition and accurate mass. In *Mass Spectrometry*, pages 67–116. Springer Science and Business Media (2010). 12

[48] Ong S.-E. and Mann M. (2005). Mass spectrometry–based proteomics turns quantitative. *Nat Chem Biol*, 1(5):252–262. 13

[49] Wells J. M. and McLuckey S. A. Collision-induced dissociation (CID) of peptides and proteins. In *Methods in Enzymology*, pages 148–185. Elsevier BV (2005). 14

[50] Papayannopoulos I. A. (1995). The interpretation of collision-induced dissociation tandem mass spectra of peptides. *Mass Spectrometry Reviews*, 14(1):49–73. 14

[51] Stryer L. *Biochemie*, chapter Struktur und Funktion der Proteine, pages 15–45. Spektrum Akademischer Verlag GmbH (1988). 14

[52] Johnson R. S., Martin S. A., Biemann K., Stults J. T., and Watson J. T. (1987). Novel fragmentation process of peptides by collision-induced decomposition in a tandem mass spectrometer: differentiation of leucine and isoleucine. *Analytical Chemistry*, 59(21):2621–2625. 15

[53] Brodbelt J. S. (2016). Ion activation methods for peptides and proteins. *Analytical Chemistry*, 88(1):30–51. 15

[54] Gillet L. C., et al. (2012). 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*, 11(6):O111.016717–O111.016717. 20

[55] Keich U. and Noble W. S. (2015). On the importance of well-calibrated scores for identifying shotgun proteomics spectra. *Journal of Proteome Research*, 14(2):1147–1160. 22

[56] Diament B. J. and Noble W. S. (2011). Faster SEQUEST searching for peptide identification from tandem mass spectra. *Journal of Proteome Research*, 10(9):3871–3879. 23, 99

[57] Eng J. K., Jahan T. A., and Hoopmann M. R. (2012). Comet: An open-source MS/MS sequence database search tool. *PROTEOMICS*, 13(1):22–24. 23

[58] Pappin D., Hojrup P., and Bleasby A. (1993). Rapid identification of proteins by peptide-mass fingerprinting. *Current Biology*, 3(6):327–332. 23

[59] Fenyö D. and Beavis R. C. (2003). A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes. *Anal. Chem.*, 75(4):768–774. 23

[60] Geer L. Y., et al. (2004). Open mass spectrometry search algorithm. *Journal of Proteome Research*, 3(5):958–964. 24

[61] Kim S. and Pevzner P. A. (2014). MS-GF+ makes progress towards a universal database search tool for proteomics. *Nat Comms*, 5:5277. 24

[62] Kim S., Gupta N., and Pevzner P. A. (2008). Spectral probabilities and generating functions of tandem mass spectra: A strike against decoy databases. *Journal of Proteome Research*, 7(8):3354–3363. 24

[63] Kim S., et al. (2010). The generating function of CID, ETD, and CID/ETD pairs of tandem mass spectra: Applications to database search. *Molecular & Cellular Proteomics*, 9(12):2840–2852. 24

[64] Elias J. E. and Gygi S. P. (2007). Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat Methods*, 4(3):207–14. 25, 26, 84

[65] Nesvizhskii A. I. (2010). A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *Journal of Proteomics*, 73(11):2092–2123. 25

[66] Benjamini Y. and Hochberg Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300. 25

[67] Reidegeld K. A., et al. (2008). An easy-to-use decoy database builder software tool, implementing different decoy strategies for false discovery rate calculation in automated ms/ms protein identifications. *Proteomics*, 8(6):1129–37. 25, 47

[68] Jeong K., Kim S., and Bandeira N. (2012). False discovery rates in spectral identification. *BMC Bioinformatics*, 13(Suppl 16):S2. 25

[69] Käll L., Storey J. D., MacCoss M. J., and Noble W. S. (2008). Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *Journal of Proteome Research*, 7(1):29–34. 26, 84

[70] Shteynberg D., et al. (2011). iProphet: Multi-level integrative analysis of shotgun proteomic data improves peptide and protein identification rates and error estimates. *Molecular & Cellular Proteomics*, 10(12):M111.007690–M111.007690. 27, 55

[71] The UniProt Consortium (2015). UniProt: a hub for protein information. *Nucleic Acids Research*, 43(D1):D204–D212. 28

[72] Wheeler D. L., et al. (2007). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*, 35(Database):D5–D12. 28

[73] UniProtKB. UniProtKB help on splice variants, http://www.uniprot.org/help/var_seq, accessed 14.12.2015. 28

[74] Alpi E., et al. (2014). Analysis of the tryptic search space in UniProt databases. *PROTEOMICS*, 15(1):48–57. 28, 39

[75] Jones A. R., et al. (2012). The mzIdentML data standard for mass spectrometry-based proteomics results. *Mol Cell Proteomics*, 11(7):M111.014381. 30, 34, 82

[76] Seymour S. L., et al. (2014). A standardized framing for reporting protein identifications in mzIdentML 1.2. *PROTEOMICS*, 14(21-22):2389–2399. 30, 35

[77] Orchard S., et al. (2013). Preparing to work with big data in proteomics - a report on the HUPO-PSI Spring Workshop. *PROTEOMICS*, 13(20):2931–2937. 33

[78] Taylor C. F., et al. (2007). The minimum information about a proteomics experiment (MIAPE). *Nature Biotechnology*, 25(8):887–893. 33

[79] Mayer G., et al. (2014). Controlled vocabularies and ontologies in proteomics: Overview, principles and practice. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*, 1844(1):98–107. 34

[80] Martens L., et al. (2010). mzML - a community standard for mass spectrometry data. *Molecular & Cellular Proteomics*, 10(1):R110.000133–R110.000133. 34

[81] Griss J., et al. (2014). The mzTab data exchange format: Communicating mass-spectrometry-based proteomics and metabolomics experimental results to a wider audience. *Molecular & Cellular Proteomics*, 13(10):2765–2775. 34, 82

[82] Tanca A., et al. (2013). Evaluating the impact of different sequence databases on metaproteome analysis: Insights from a lab-assembled microbial mixture. *PLoS ONE*, 8(12):e82981. 39

[83] Klimek J., et al. (2008). The standard protein mix database: a diverse data set to assist in the production of improved peptide and protein identification software tools. *J Proteome Res*, 7(1):96–103. 46, 65

[84] Quandt A., et al. (2014). Using synthetic peptides to benchmark peptide identification software and search parameters for MS/MS data analysis. *EuPA Open Proteomics*, 5:21–31. 46, 65

[85] Chawade A., Sandin M., Teleman J., Malmström J., and Levander F. (2015). Data processing has major impact on the outcome of quantitative label-free LC-MS analysis. *Journal of Proteome Research*, 14(2):676–687. 46

[86] Ramakrishnan S. R., et al. (2009). Integrating shotgun proteomics and mRNA expression data to improve protein identification. *Bioinformatics*, 25(11):1397–1403. 46, 47

[87] Kley R. A., et al. (2013). A combined laser microdissection and mass spectrometry approach reveals new disease relevant proteins accumulating in aggregates of filaminopathy patients. *Mol Cell Proteomics*, 12(1):215–27. 46, 84

[88] Chambers M. C., et al. (2012). A cross-platform toolkit for mass spectrometry and proteomics. *Nat Biotechnol*, 30(10):918–20. 47

[89] Proteome Informatics Research Group (iPRG). iPRG homepage. `https://abrf.org/research-group/proteome-informatics-research-group-iprg`. [Online; accessed 23.04.2016]. 47, 66

[90] Ahn J.-M., et al. (2014). Proteogenomic analysis of human chromosome 9-encoded genes from human samples and lung cancer tissues. *J. Proteome Res.*, 13(1):137–146. 48

[91] Yuan Z.-F., Lin S., Molden R. C., and Garcia B. A. (2014). Evaluation of proteomic search engines for the analysis of histone modifications. *J. Proteome Res.*, 13(10):4470–4478. 48

[92] Li Y. F., et al. (2009). A Bayesian approach to protein inference problem in shotgun proteomics. *Journal of Computational Biology*, 16(8):1183–1193. 54, 86

[93] Li Y. F., Arnold R. J., Tang H., and Radivojac P. (2010). The importance of peptide detectability for protein identification, quantification, and experiment design in MS/MS proteomics. *Journal of Proteome Research*, 9(12):6288–6297. 54

[94] Huang T. and He Z. (2012). A linear programming model for protein inference problem in shotgun proteomics. *Bioinformatics*, 28(22):2956–2962. 54

[95] Serang O., MacCoss M. J., and Noble W. S. (2010). Efficient marginalization to compute protein posterior probabilities from shotgun mass spectrometry data. *Journal of Proteome Research*, 9(10):5346–5357. 54, 55, 86

[96] Berthold M. R., et al. KNIME: The KoNstanz Information MinEr. In *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. Springer (2007). 54, 56, 82, 91

[97] Sturm M., et al. (2008). OpenMS - an open-source software framework for mass spectrometry. *BMC Bioinformatics*, 9:163. 54, 56, 82

[98] Li Y. F. and Radivojac P. (2012). Computational approaches to protein inference in shotgun proteomics. *BMC Bioinformatics*, 13(16):S4. 55

[99] Deutsch E. W., et al. (2015). Trans-proteomic pipeline, a standardized data processing pipeline for large-scale reproducible proteomics informatics. *PROTEOMICS - Clinical Applications*, 9(7-8):745–754. 55

[100] Bertsch A., Gröpl C., Reinert K., and Kohlbacher O. (2010). OpenMS and TOPP: Open Source Software for LC-MS data analysis. *Data Mining in Proteomics*, page 353–367. 56

[101] Pfeifer N., Leinenbach A., Huber C. G., and Kohlbacher O. (2007). Statistical learning of peptide retention behavior in chromatographic separations: a new kernel-based approach for computational proteomics. *BMC Bioinformatics*, 8(1):468. 57

[102] Serang O. and Käll L. (2015). Solution to statistical challenges in proteomics is more statistics, not less. *Journal of Proteome Research*, 14(10):4099–4103. 58

[103] Editorial (2015). The difficulty of a fair comparison. *Nat Meth*, 12(4):273–273. 58

[104] Claassen M., Reiter L., Hengartner M. O., Buhmann J. M., and Aebersold R. (2011). Generic comparison of protein inference engines. *Molecular & Cellular Proteomics*, 11(4):O110.007088–O110.007088. 58

[105] Huang T., Wang J., Yu W., and He Z. (2012). Protein inference: a review. *Brief Bioinform*, 13(5):586–614. 58, 82

[106] Shteynberg D., Nesvizhskii A. I., Moritz R. L., and Deutsch E. W. (2013). Combining results of multiple search engines in proteomics. *Molecular & Cellular Proteomics*, 12(9):2383–2393. 61

[107] Omenn G. S., et al. (2015). Metrics for the Human Proteome Project 2015: Progress on the human proteome and guidelines for high-confidence protein identification. *Journal of Proteome Research*, 14(9):3452–3460. 69, 74

[108] Reiter L., et al. (2009). Protein identification false discovery rates for very large proteomics data sets generated by tandem mass spectrometry. *Molecular & Cellular Proteomics*, 8(11):2405–2417. 73

[109] Pedrioli P. G. A., et al. (2004). A common open representation of mass spectrometry data and its application to proteomics research. *Nature Biotechnology*, 22(11):1459–1466. 74

[110] Kirkwood K. J., Ahmad Y., Larance M., and Lamond A. I. (2013). Characterization of native protein complexes and protein isoform variation using size-fractionation-based quantitative proteomics. *Mol Cell Proteomics*, 12(12):3851–3873. 77

[111] Tay A. P., et al. (2015). Proteomic validation of transcript isoforms, including those assembled from rna-seq data. *J Proteome Res*, 14(9):3541–3554.

[112] Tutakhel O. A. Z., et al. (2016). Alternative splice variant of the thiazide-sensitive nacl cotransporter: a novel player in renal salt handling. *Am J Physiol Renal Physiol*, 310(3):F204–F216. 77

[113] Schrötter A., et al. (2013). FE65 regulates and interacts with the Bloom syndrome protein in dynamic nuclear spheres - potential relevance to Alzheimer's disease. *J Cell Sci*, 126(Pt 11):2480–92. 84

[114] Maerkens A., et al. (2013). Differential proteomic analysis of abnormal intramyoplasmic aggregates in desminopathy. *Journal of Proteomics*, 90(0):14 – 27. Special Issue: FROM GENOME TO PROTEOME: OPEN INNOVATIONS. 84

[115] Vizcaíno J. A., et al. (2013). The PRoteomics IDEntifications (PRIDE) database and associated tools: status in 2013. *Nucleic Acids Res*, 41(Database issue):D1063–9. 85

[116] Körting G., et al. (2006). Protein extractor; from peptide ID to protein ID. *Molecular & Cellular Proteomics*, 5(10):S216–S216. 88

[117] O'Madadhain J., Fisher D., White S., and Boey Y. The JUNG (Java Universal Network/Graph) framework. Technical report, UCI-ICS (2003). 89

[118] Apache Software Foundation: Los Angeles, CA. Apache Tomcat. `http://tomcat.apache.org`. [Online; accessed 14.03.2016]. 90

[119] Oracle Corporation and/or its affiliates. GlassFish Server. `https://glassfish.java.net`. [Online; accessed 14.03.2016]. 90

[120] de la Garza L., et al. (2013). From the desktop to the grid: conversion of KNIME workflows to gUSE. *5th International Workshop on Science Gateways*. 91

[121] Aiche S., et al. GenericKnimeNodes. `https://github.com/genericworkflownodes/GenericKnimeNodes`. [Online; accessed 14.03.2016]. 91

[122] Vizcaíno J. A., et al. (2015). 2016 update of the PRIDE database and its related tools. *Nucleic Acids Research*, 44(D1):D447–D456. 94

[123] Vizcaíno J. A., et al. (2014). ProteomeXchange provides globally coordinated proteomics data submission and dissemination. *Nature Biotechnology*, 32(3):223–226. 94

[124] Wang R., et al. (2012). PRIDE inspector: a tool to visualize and validate MS proteomics data. *Nature Biotechnology*, 30(2):135–137. 95

[125] Perez-Riverol Y., et al. (2015). PRIDE Inspector Toolsuite: Moving toward a universal visualization tool for proteomics data standard formats and quality assessment of ProteomeXchange datasets. *Molecular & Cellular Proteomics*, 15(1):305–317. 95

[126] Perez-Riverol Y., et al. (2015). ms-data-core-api: an open-source, metadata-oriented library for computational proteomics. *Bioinformatics*, 31(17):2903–2905. 95

[127] The Apache Software Foundation. Apache Maven. `https://maven.apache.org/`. [Online; accessed 19.04.2016]. 98

[128] Helsens K., Martens L., Vandekerckhove J., and Gevaert K. (2007). MascotDatfile: an open-source library to fully parse and analyse MASCOT MS/MS search results. *Proteomics*, 7(3):364–6. 99

[129] Muth T., Vaudel M., Barsnes H., Martens L., and Sickmann A. (2010). XTandem parser: an open-source library to parse and analyse X!Tandem MS/MS search results. *Proteomics*, 10(7):1522–4. 99

[130] Reisinger F., et al. (2012). jmzIdentML API: A java interface to the mzIdentML standard for peptide and protein identification data. *Proteomics*, 12(6):790–4. 100

[131] The Eclipse Foundation. Eclipse. https://www.eclipse.org/. [Online; accessed 21.04.2016]. 104

Appendix A

Abbreviations

2D-DIGE	two-dimensional Differential Gel Electrophoresis
CID	Collision Induced Dissociation
CV	Controlled Vocabulary
DDA	Data Dependent Acquisition
DIA	Data Independent Acquisition
DNA	Deoxyribonucleic acid
ESI	Electrospray Ionisation
FDR	False Discovery Rate
FP	False Positive
HUPO	Human Proteome Organization
IDE	Integrated Development Environment
JSF	JavaServer Faces
LC	Liquid Chromatography
LC-MS/MS	Liquid chromatography coupled to tandem mass spectrometry
MALDI	Matrix Assisted Laser Desorption/Ionisation
MGF	Mascot Generic Format
MIAPE	Minimum Information About a Proteomics Experiment
MPC	Medizinisches Proteom-Center (an institute of the Ruhr-Universität Bochum)
MS	(Ion) Mass Spectrum or Mass Spectrometry
MS/MS	Tandem Mass Spectrometry
PAG	Protein Ambiguity Group
PP	ProteinProphet
PRIDE	PRoteomics IDEntifications (database)
PSI	Proteomics Standards Initiative (of teh HUPO)
PSM	Peptide Spectrum Match
SDK	Software Development Kit
TDA	Target-Decoy Approach
TP	True Positive
TPP	Trans-Proteomic Pipeline

Appendix B

Contributions

Chapter 5

Section 5.2

This work is part of the main PIA manuscript "PIA: An Intuitive Protein Inference Engine with a Web-Based User Interface"[1] published in the *Journal of Proteome Research*. Additionally to myself, Alexandra Maerkens, Yasset Perez-Riverol, Helmut E. Meyer, Katrin Marcus, Christian Stephan, Oliver Kohlbacher and Martin Eisenacher contributed to the work.

Section 5.3

This work was published before in "In-depth analysis of protein inference algorithms using multiple search engines and well-defined metrics"[2] in the *Journal of Proteomics*. In addition to myself and Enrique Audain (we contributed equally as first authors), Timo Sachsenberg, Julianus Pfeuffer, Xiao Liang, Henning Hermjakob, Aniel Sanchez, Martin Eisenacher, Knut Reinert, David L. Tabb, Oliver Kohlbacher and Yasset Perez-Riverol contributed to the work.

Chapter 6

The contents of this chapter were partially published before in the *Journal of Proteome Research*[1], as was Section 5.2

Appendix C

List of publications

Peer reviewed journal articles

Ordered by descending date of publication.

- Audain E, **Uszkoreit J**, Sachsenberg T, Pfeuffer J, Liang X, Hermjakob H, Sanchez A, Eisenacher M, Reinert K, Tabb DL, Kohlbacher O, Perez-Riverol Y. In-depth analysis of protein inference algorithms using multiple search engines and well-defined metrics. *J Proteomics*. 2016 Aug 4;150:170-182.

  Enrique Audain and Julian Uszkoreit contributed equally to this work

  This contains the comparison of inference algorithm as shown in Section 5.3.

- Perez-Riverol Y, Gatto L, Wang R, Sachsenberg T, **Uszkoreit J**, da Veiga Leprevost F, Fufezan C, Ternent T, Eglen SJ, Katz DS, Pollard TJ, Konovalov A, Flight RM, Blin K, Vizcaino JA. Ten Simple Rules for Taking Advantage of Git and GitHub. *PLoS Comput Biol* 12(7): e1004947. 14 July 2016

- Maerkens A, Olivé M, Schreiner A, Feldkirchner S, Schessl J, **Uszkoreit J**, Barkovits K, Güttsches AK, Theis V, Eisenacher M, Tegenthoff M, Goldfarb LG, Schröder R, Schoser B, van der Ven PFM, Fürst DO, Vorgerd M, Marcus K, Kley RA. New insights into the protein aggregation pathology in myotilinopathy by combined proteomic and immunolocalization analyses. *Acta Neuropathologica Communications.* 2016 Feb; 4(1):1-20.

  This publication used PIA for parts of the analysis.

- Perez-Riverol Y, Xu QW, Wang R, **Uszkoreit J**, Griss J, Sanchez A, Reisinger F, Csordas A, Ternent T, Del-Toro N, Dianes JA, Eisenacher M, Hermjakob H, Vizcaino JA. PRIDE Inspector Toolsuite: moving towards a universal visualization tool for proteomics data standard formats and quality assessment of ProteomeXchange datasets. *Mol Cell Proteomics.* 2016 Jan;15(1):305-17.

Parts of PIA are implemented in PRIDE Inspector as described in this publication.

- **Uszkoreit J**, Maerkens A, Perez-Riverol Y, Meyer HE, Marcus K, Stephan C, Kohlbacher O, Eisenacher M. PIA: An Intuitive Protein Inference Engine with a Web-Based User Interface. *J Proteome Res.* 2015 Jul 2;14(7):2988-97.

  This is the main publication of PIA.

- Perez-Riverol Y, **Uszkoreit J**, Sanchez A, Ternent T, del Toro N, Hermjakob H, Vizcaíno JA, Wang R. ms-data-core-api: An open-source, metadata-oriented library for computational proteomics. *Bioinformatics.* 2015 April 24.

  This publication describes the basis of PIA's implementation into PRIDE Inspector.

- **Uszkoreit J**, Plohnke N, Rexroth S, Marcus K, Eisenacher M. The bacterial proteogenomic pipeline. *BMC Genomics.* 2014, 15(Suppl 9):S19.

  In this paper, the PSM FDR estimations and combinations of search results using PIA are used for proteogenomics approaches.

- Seymour SL, Farrah T, Binz PA, Chalkley RJ, Cottrell JS, Searle BC, Tabb DL, Vizcaíno JA, Prieto G, **Uszkoreit J**, Eisenacher M, Martínez-Bartolomé S, Ghali F, Jones AR. A standardized framing for reporting protein identifications in mzIdentML 1.2. *Proteomics.* 2014 Aug 4.

- Nensa FM, Neumann MH, Schrötter A, Przyborski A, Mastalski T, Susdalzew S, Looβe C, Helling S, El Magraoui F, Erdmann R, Meyer HE, **Uszkoreit J**, Eisenacher M, Suh J, Guénette SY, Röhner N, Kögel D, Theiss C, Marcus K, Müller T. Amyloid beta a4 precursor protein-binding family B member 1 (FE65) interactomics revealed synaptic vesicle glycoprotein 2A (SV2A) and sarcoplasmic/endoplasmic reticulum calcium ATPase 2 (SERCA2) as new binding proteins in the human brain. *Mol Cell Proteomics.* 2014 Feb;13(2):475-88.

  This publication used PIA for parts of the analysis.

- Maerkens A, Kley RA, Olivé M, Theis V, van der Ven PF, Reimann J, Milting H, Schreiner A, **Uszkoreit J**, Eisenacher M, Barkovits K, Güttsches AK, Tonillo J, Kuhlmann K, Meyer HE, Schröder R, Tegenthoff M, Fürst DO, Müller T, Goldfarb LG, Vorgerd M, Marcus K. Differential proteomic analysis of abnormal intramyoplasmic aggregates in desminopathy. *J Proteomics.* 2013 Sep 2

- Walzer M, Qi D, Mayer G, **Uszkoreit J**, Eisenacher M, Sachsenberg T, Gonzalez-Galarza FF, Fan J, Bessant C, Deutsch EW, Reisinger F, Vizcaíno JA, Medina-Aunon JA, Albar JP, Kohlbacher O, Jones AR. The mzQuantML data standard for mass spectrometry-based quantitative studies in proteomics. *Mol Cell Proteomics.* 2013 Aug;12(8):2332-40.

- Poschmann G, Lendzian A, **Uszkoreit J**, Eisenacher M, Borght AV, Ramaekers FC, Meyer HE, Stühler K. A combination of two electrophoretical approaches for detailed proteome-based characterization of SCLC subtypes. *Arch Physiol Biochem.* 2013 Jul;119(3):114-25.

- Schrötter A, Mastalski T, Nensa FM, Neumann M, Loosse C, Pfeiffer K, Magraoui FE, Platta HW, Erdmann R, Theiss C, **Uszkoreit J**, Eisenacher M, Meyer HE, Marcus K, Müller T. FE65 regulates and interacts with the Bloom syndrome protein in dynamic nuclear spheres - potential relevance to Alzheimer's disease. *J Cell Sci.* 2013 Jun 1;126(Pt 11):2480-92.

  This publication used PIA for parts of the analysis.

- Kley RA, Maerkens A, Leber Y, Theis V, Schreiner A, van der Ven PF, **Uszkoreit J**, Stephan C, Eulitz S, Euler N, Kirschner J, Müller K, Meyer HE, Tegenthoff M, Fürst DO, Vorgerd M, Müller T, Marcus K. A combined laser microdissection and mass spectrometry approach reveals new disease relevant proteins accumulating in aggregates of filaminopathy patients. *Mol Cell Proteomics.* 2013 Jan;12(1):215-27.

  This publication used PIA for parts of the analysis.

- Müller T, Schrötter A, Loosse C, Helling S, Stephan C, Ahrens M, **Uszkoreit J**, Eisenacher M, Meyer HE, Marcus K. Sense and nonsense of pathway analysis software in proteomics. *J Proteome Res.* 2011 Dec 2;10(12):5398-408.

Book chapters

- Eisenacher M, Kohl M, Turewicz M, Koch MH, **Uszkoreit J**, Stephan C. Search and decoy: the automatic identification of mass spectra. *Methods Mol Biol.* 2012;893:445-88.

Appendix D

Supporting Tables

Table D.1: Table showing the fraction of unique tryptic peptides in common databases. This table shows the results of an *in silico* digestion of often used databases from the UniProtKB. Shown are the results of human and mouse entries of Swiss-Prot, the human and mouse proteomes and the complete Swiss-Prot of the UniProtKB release 2015_11. The protein sequences of each database were digested using the tryptic regular expression [RK]|{P}, allowing 1, 2 and 3 missed cleavages (m) and discarding peptides shorter than 6 or longer than 45 amino acids. The total number of accessions, the number of accessions having at least one unique peptide, the total number of peptides and the number of unique peptides are given together with the corresponding percentages of the whole database.

Database	accessions	m	accessions with unique peptides (%)	peptides	unique peptides (%)
Swiss-Prot (*H. sapiens*)	20,194	0	19,926 (98.67)	581,909	561,261 (96.45)
		1	19,976 (98.92)	1,503,676	1,455,430 (96.79)
		2	19,990 (98.99)	2,450,916	2,379,336 (97.08)
Proteome (*H. sapiens*)	70,075	0	52,636 (75.11)	660,251	356,336 (53.97)
		1	59,679 (85.16)	1,705,209	924,710 (54.23)
		2	61,356 (87.56)	2,780,870	1,516,776 (54.54)
Swiss-Prot (*M. musculus*)	16,740	0	16,665 (99.55)	498,064	485,477 (97.47)
		1	16,691 (99.71)	1,280,698	1,252,243 (97.78)
		2	16,695 (99.73)	2,079,548	2,038,603 (98.03)
Proteome (*M. musculus*)	49,235	0	35,376 (71.85)	629,766	342,905 (54.45)
		1	38,918 (79.05)	1,622,683	888,260 (54.74)
		2	39,680 (80.60)	2,641,145	1,453,169 (55.02)
Swiss-Prot	549,832	0	398,238 (72.43)	6,451,378	5,002,565 (77.54)
		1	409,011 (74.39)	17,230,789	13,599,567 (78.93)
		2	411,566 (74.85)	28,675,635	22,941,276 (80.00)

Table D.2: The numbers of reported true positive, false negative, false positive, the resulting precision, recall and F1 score and the total number of protein groups for the iPRG 2008 dataset using the Swiss-Prot database. The table shows the respective numbers for each inference algorithm and each combination of search engines (XT=X!Tandem, MA=Mascot, MS=MS-GF+).

Run	TP	FN	FP	precision	recall	F1 score	total number of groups
PIA XT+MA+MS	231	27	5	0.98	0.90	0.94	304
PIA MA+MS	231	27	5	0.98	0.90	0.94	304
PIA XT+MS	228	30	5	0.98	0.88	0.93	298
PIA XT	228	30	5	0.98	0.88	0.93	297
PIA MA+MS	220	38	4	0.98	0.85	0.91	282
ProteinLP XT+MA+MS	231	27	20	0.92	0.90	0.91	319
ProteinLP XT+MA	231	27	20	0.92	0.90	0.91	319
ProteinLP XT+MS	228	30	19	0.92	0.88	0.90	314
ProteinLP XT	228	30	19	0.92	0.88	0.90	314
PIA MA	208	50	4	0.98	0.81	0.89	258
ProteinLP MA+MS	217	41	16	0.93	0.84	0.88	283
ProteinProphet MA+MS	203	55	1	1.00	0.79	0.88	249
ProteinLP MA	208	50	13	0.94	0.81	0.87	267
ProteinLP MS	209	49	15	0.93	0.81	0.87	261
ProteinProphet MA	198	60	1	0.99	0.77	0.87	242
ProteinProphet XT+MA+MS	192	66	1	0.99	0.74	0.85	199
ProteinProphet XT+MA	192	66	1	0.99	0.74	0.85	199
ProteinProphet XT+MS	191	67	1	0.99	0.74	0.85	197
ProteinProphet XT	191	67	1	0.99	0.74	0.85	197
ProteinProphet MS	190	68	2	0.99	0.74	0.84	224
PIA MS	154	104	3	0.98	0.60	0.74	159
Fido MA+MS	217	41	162	0.57	0.84	0.68	441
Fido MS	113	145	100	0.53	0.44	0.48	229
Fido XT+MS	98	160	112	0.47	0.38	0.42	225
Fido XT	98	160	112	0.47	0.38	0.42	225
Fido XT+MA+MS	99	159	120	0.45	0.38	0.42	236
Fido XT+MA	99	159	120	0.45	0.38	0.42	236
MSBayesPro XT+MA+MS	57	201	0	1.00	0.22	0.36	77
MSBayesPro XT+MA	56	202	0	1.00	0.22	0.36	73
Fido MA	73	185	95	0.43	0.28	0.34	173
MSBayesPro MA+MS	48	210	0	1.00	0.19	0.31	62
MSBayesPro XT	26	232	0	1.00	0.10	0.18	31
MSBayesPro MS	19	239	0	1.00	0.07	0.13	23
MSBayesPro XT+MS	15	243	0	1.00	0.06	0.11	20
MSBayesPro MA	15	243	0	1.00	0.06	0.11	18

Appendix E

Supporting Figures

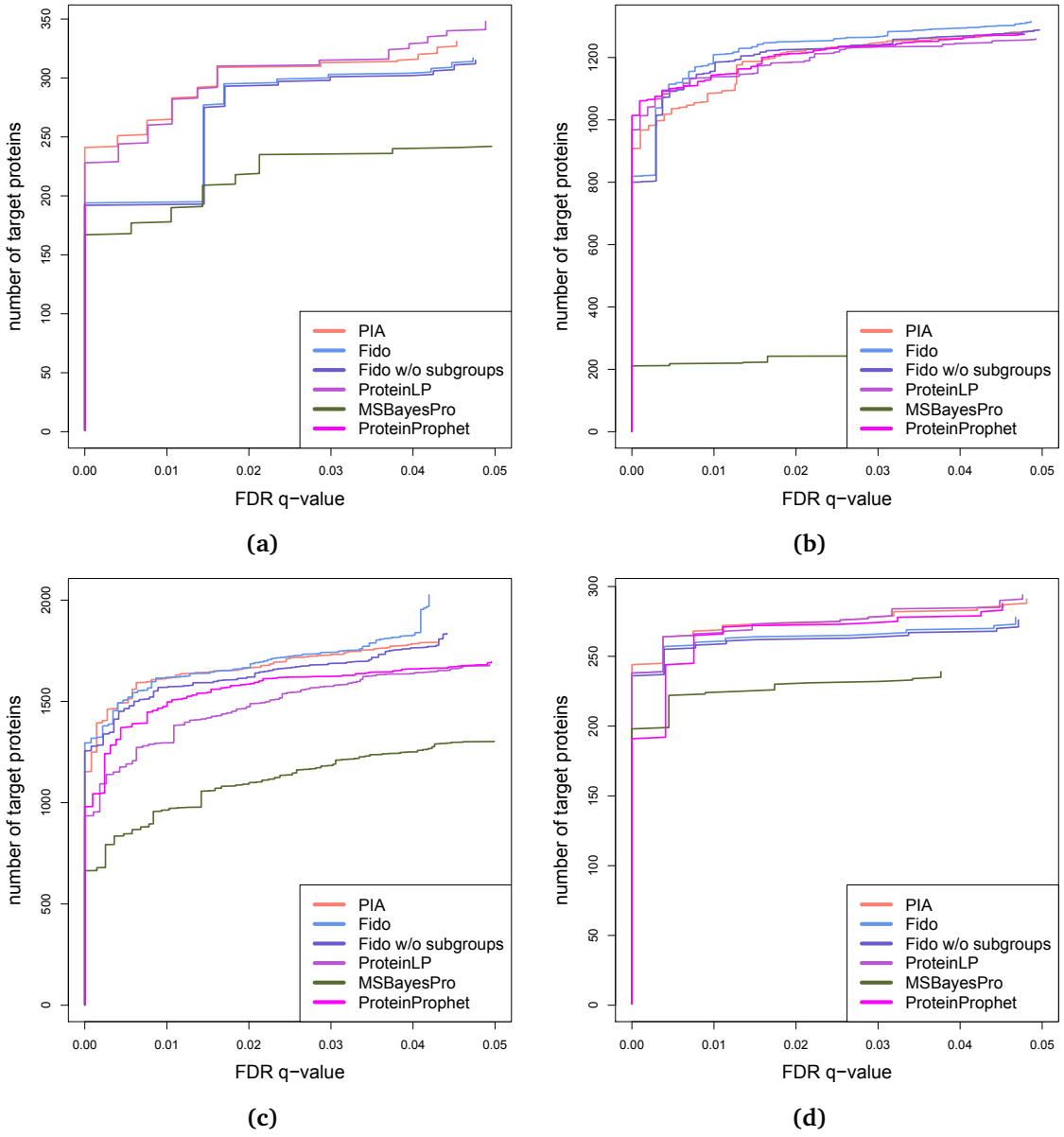
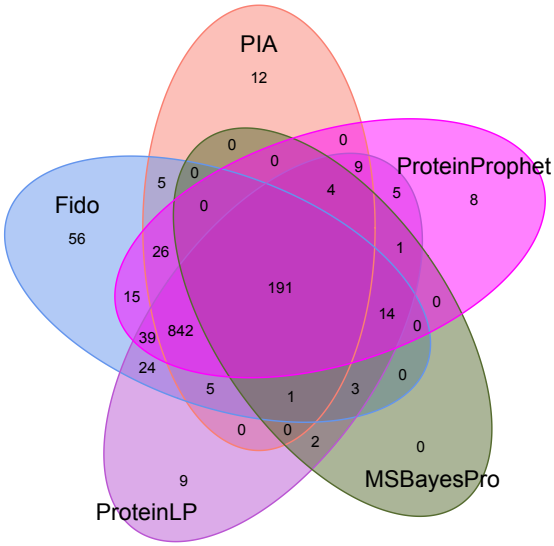
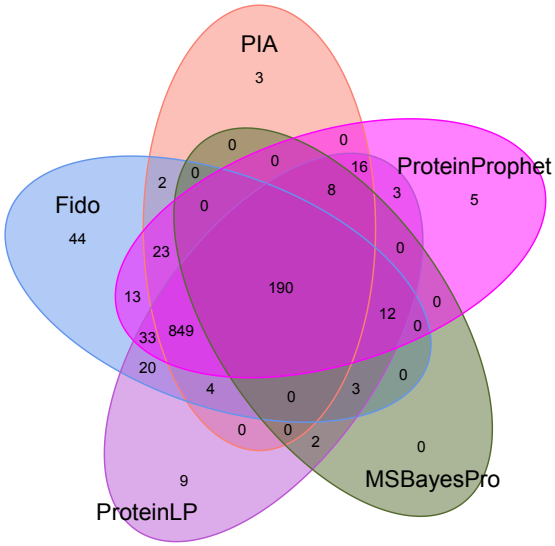


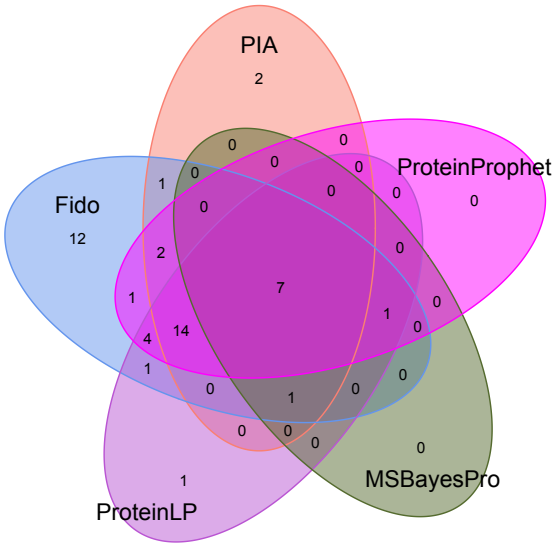
Figure E.1: Pseudo-ROC curves showing the number of reported protein groups against the FDR q-value for the four analysed datasets using the respective Swiss-Prot database and the combination of all three search engines: a) iPRG 2008, b) yeast, c) PXD000603, and d) PXD001118 dataset. The plots indicate that the main trend is similar for all inference algorithms. Depending on the dataset, different algorithms perform worse than others for certain q-value ranges, like Fido in the iPRG 2008 dataset, PIA in the yeast dataset and ProteinLP in the PXD000603 dataset. Only MSBayesPro performs significantly poorer than all other methods on all datasets.



(a) All protein groups



(b) True positives



(c) False positives

Figure E.2: Venn diagrams showing the number of reported protein for each inference algorithm at 1% FDR for the ground truth yeast dataset using the Swiss-Prot database. (a) Number of all reported proteins for the yeast dataset. (b) Number of reported proteins included in the reference set, i.e. the true positives. (c) Number of reported proteins that are not in the reference set and thus false positives. It can be seen that the overlap of all algorithms excluding MSBayesPro is in all three cases the largest number. Though MSBayesPro reports no groups uniquely and 88% of its reported groups are reported in consensus with all other methods. Though most of the reported protein groups are true positives, especially interesting are also the 7 collectively reported (and also the 14 collectively excluding MSBayesPro) groups, which are false positives. This might indicate an error or incompleteness in the determination of the true positives for this dataset.

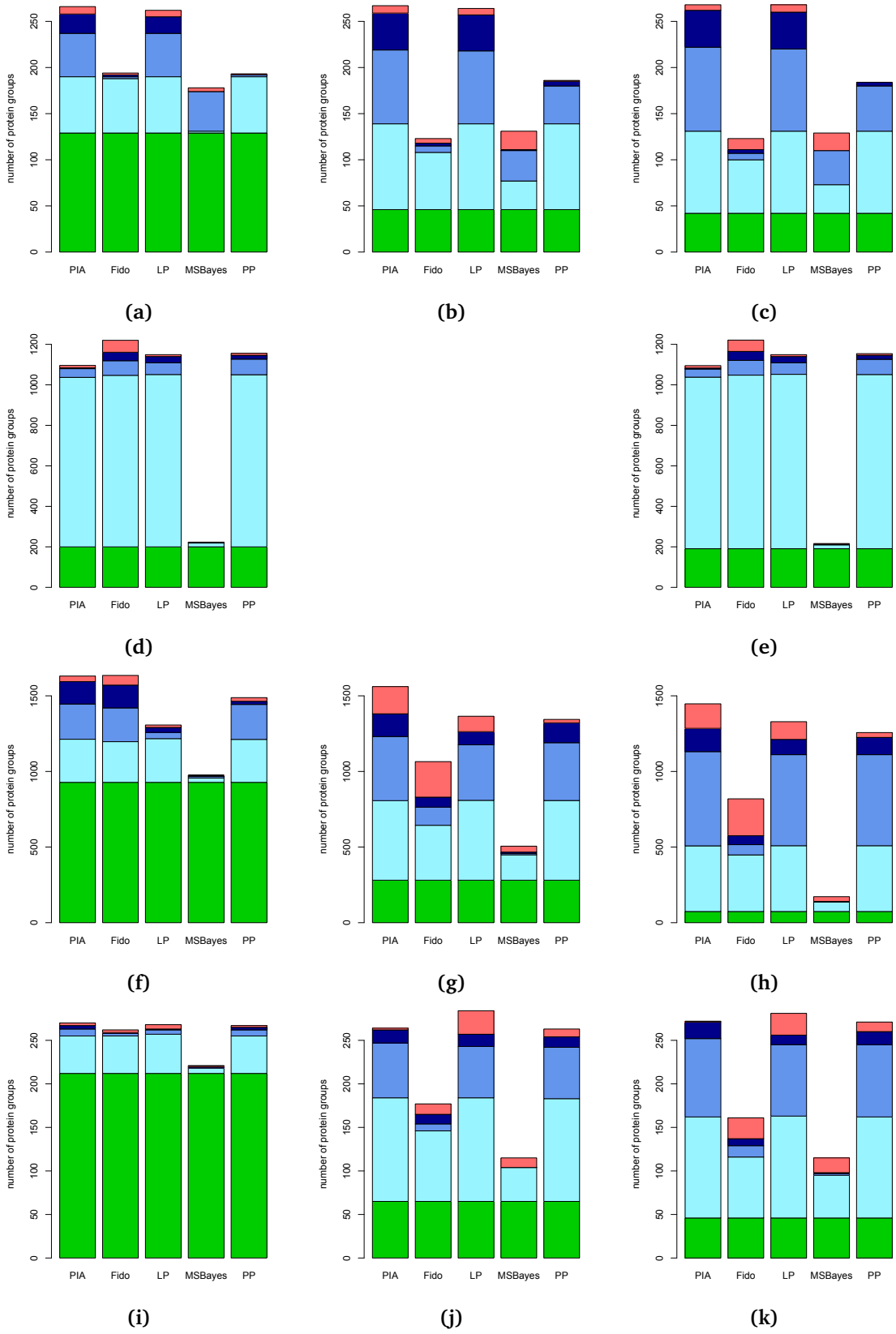
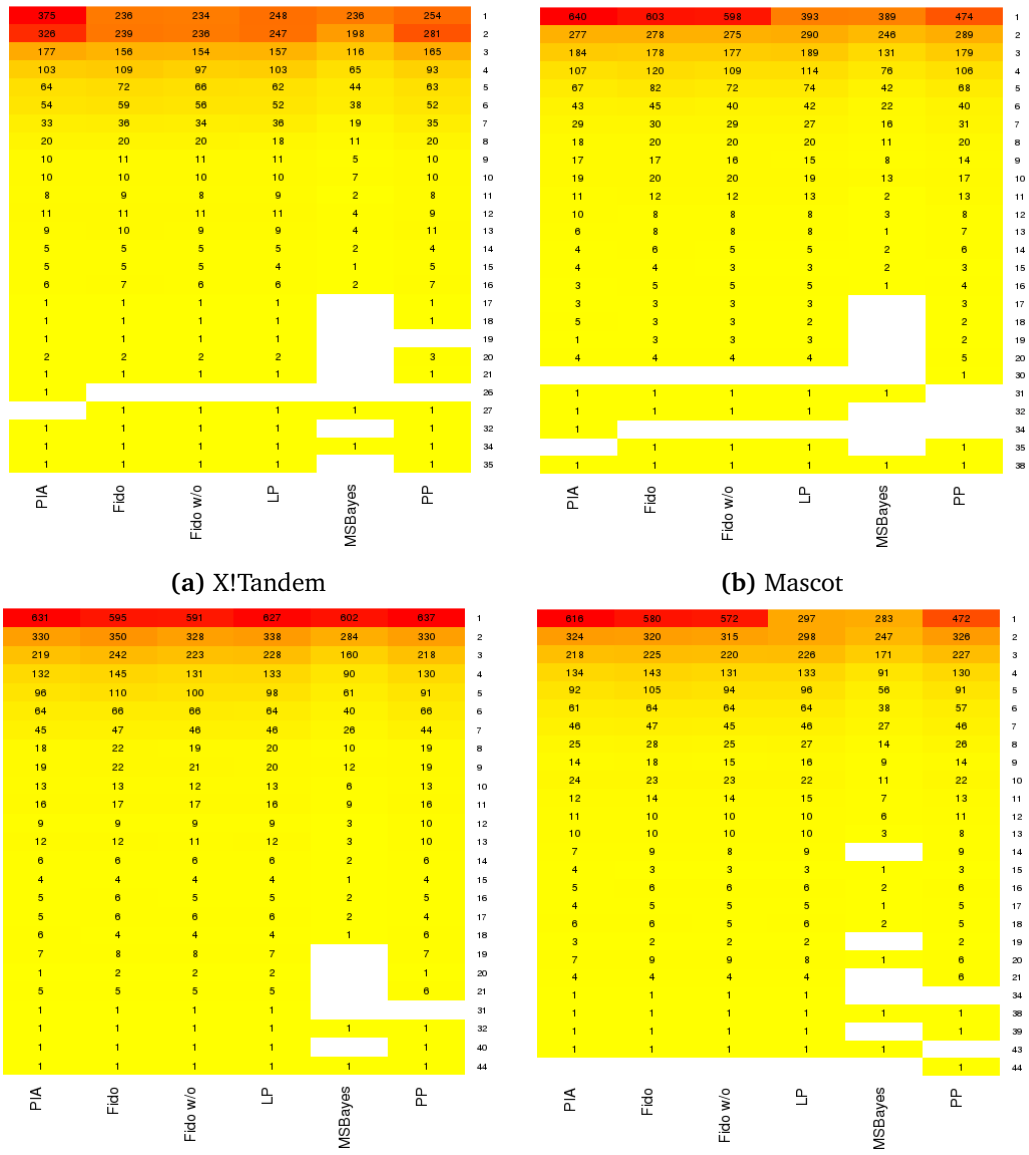


Figure E.3: Number of protein groups reported for the datasets using different inference algorithms and databases. Number of protein groups under a 1% FDR q-value for the iPRG 2008 (a-c), yeast (d, e), PXD000603 (f-h) and PXD001118 (i-k) dataset with the corresponding Swiss-Prot (a, d, f, i), UniProt proteome (b, d, g, j) and UniProt proteome with isoforms (c, e, h, k) databases. For the colour codes, please see Figure 5.10 and Section 5.3.8.



(a) X!Tandem **(b)** Mascot

(c) MS-GF+ **(d)** Merged

Figure E.4: Numbers of identified peptides per protein. The graphics show the numbers of peptides identified per protein in a heatmap-like plot for identifications from a) X!Tandem, b) Mascot, c) MS-GF+ and d) combination of all for the PXD000603 dataset and the Swiss-Prot database. It can be seen, that the most proteins are identified with relatively few peptides, while only few proteins have ten or more peptides in this dataset. With the single search engines, PIA, Fido and ProteinProphet report on average 5.7% of proteins with ten or more peptides, while with the merge of the PSM results they report 6.5% with at least ten peptides, and also numerically at least 5 proteins more with these many peptides. This shows that also on protein level a qualitative improvement is yielded by merging search results.

Appendix F

Additional Material

Listing F.1: Example for a PIA analysis pipeline XML file.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tool docurl="http://www.medizinisches-proteom-center.de" name="pipeline"
      version="0.1.23">
  <description>
    This file contains a pipeline for execution by PIA
  </description>
  <PARAMETERS>
    <!-- PSM settings -->
    <NODE name="PSMCreatePSMSets">
      <ITEM name="create sets" value="yes" type="string"/>
    </NODE>
    <NODE name="PSMAddPreferredFDRScore">
      <ITEM name="score name" value="mascot_score" type="string"/>
    </NODE>
    <NODE name="PSMAddPreferredFDRScore">
      <ITEM name="score name" value="xtandem_expect" type="string"/>
    </NODE>
    <NODE name="PSMSetAllTopidentificationsForFDR">
      <ITEM name="number of top identifications" value="1" type="string"/>
    </NODE>
    <NODE name="PSMSetAllDecoyPattern">
      <ITEM name="decoy pattern" value="s." type="string"/>
    </NODE>
    <NODE name="PSMCalculateAllFDR"/>
    <NODE name="PSMCalculateCombinedFDRScore"/>

    <!-- peptide settings -->
    <NODE name="PeptideConsiderModifications">
```

```
      <ITEM name="consider␣modifications" value="no" type="string"/>
    </NODE>

    <!-- protein settings -->
    <NODE name="ProteinAddInferenceFilter">
      <ITEM name="filtername" value="psm_score_filter_psm_combined_fdr_score"
          type="string" />
      <ITEM name="comparison" value="LEQ" type="string"/>
      <ITEM name="value" value="0.01" type="string"/>
      <ITEM name="negate" value="no" type="string"/>
    </NODE>
    <NODE name="ProteinInfereProteins">
      <ITEM name="inference" value="inference_spectrum_extractor"
      type="string"/>
      <ITEM name="scoring" value="scoring_multiplicative" type="string"/>
      <ITEM name="used␣score" value="combined_fdr_score" type="string"/>
      <ITEM name="used␣spectra" value="best" type="string"/>
    </NODE>
  </PARAMETERS>
</tool>
```

Listing F.2: Example of a PIA intermediate XML file.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:jPiaXML date="2016-04-08T17:19:43.862+02:00" name="example file"
    xmlns:ns2="http://psidev.info/psi/pi/mzIdentML/1.1"
    xmlns:ns3="http://www.medizinisches-proteom-center.de/PIA/piaintermediate">
<filesList>
  <file fileName="/tmp/example_1.idXML" format="idXML" id="1" name="example_1.idXML">
    <AnalysisCollection>
      <ns2:SpectrumIdentification spectrumIdentificationProtocol_ref="identProtocol_1_1"
          id="specIdent_1_1">
        <ns2:SearchDatabaseRef searchDatabase_ref="searchDB_1"/>
      </ns2:SpectrumIdentification>
    </AnalysisCollection>
    <AnalysisProtocolCollection>
      <ns2:SpectrumIdentificationProtocol analysisSoftware_ref="software_1"
          id="identProtocol_1_1">
        <ns2:Enzymes>
          <ns2:Enzyme missedCleavages="2" id="enzyme_1_1">
            <ns2:SiteRegexp>(?&lt;=[KR])(?!P)</ns2:SiteRegexp>
          </ns2:Enzyme>
        </ns2:Enzymes>
        <!-- and other search parameters as in mzIdentML -->
      </ns2:SpectrumIdentificationProtocol>
    </AnalysisProtocolCollection>
  </file>
  <!-- more files -->
</filesList>

<Inputs>
  <ns2:SearchDatabase location="" id="searchDB_1" name="swissprot-decoy-20160210.fasta">
    <ns2:DatabaseName>
      <ns2:userParam name="UniProt Swiss-Prot Decoy"/>
    </ns2:DatabaseName>
  </ns2:SearchDatabase>
</Inputs>

<AnalysisSoftwareList>
  <ns2:AnalysisSoftware version="2.5.1" uri="http://www.matrixscience.com/" id="software_1"
      name="mascot">
    <ns2:SoftwareName>
      <ns2:cvParam cvRef="PSI-MS" accession="MS:1001207" name="Mascot"/>
    </ns2:SoftwareName>
  </ns2:AnalysisSoftware>
```

```
</AnalysisSoftwareList>

<spectraList>
  <spectrumMatch charge="3" deltaMass="0.00174" fileRef="1" id="19293" isDecoy="false"
      massToCharge="651.68085" missed="0" retentionTime="3713.77"
      sequence="FIQVCTQLQVLTEAFR" spectrumIdentificationRef="specIdent_1_1">
    <sourceID>index=9548</sourceID>
    <Title>651.68085_3713.77_controllerType=0 controllerNumber=1 scan=9548_LC5</Title>
    <Score cvAccession="MS:1001171" name="Mascot␣Ion␣Score" value="23.93"/>
    <Score cvAccession="MS:1001172" name="Mascot␣Expect" value="0.343687"/>
    <Modification description="Carbamidomethyl" location="5" mass="57.0214" residue="C"/>
  </spectrumMatch>
  <!-- many more spectrum matches -->
</spectraList>

<accessionsList>
  <accession acc="Q9UBV8" id="15640">
    <Sequence>MASYPYRQGCPGAAGQAPGAPPGSYYPGPPNSGGQYGSGLPPG...</Sequence>
    <FileRef fileRef="1"/>
    <SearchDatabaseRef searchDatabaseRef="searchDB_1"/>
    <Description fileRefID="1">PEF1_HUMAN Peflin OS=Homo sapiens</Description>
  </accession>
  <!-- more accessions-->
</accessionsList>

<peptidesList>
  <peptide id="12591">
    <Sequence>FIQVCTQLQVLTEAFR</Sequence>
    <spectrumRefList>
      <spectrumRef spectrumRefID="19292"/>
      <spectrumRef spectrumRefID="39679"/>
    </spectrumRefList>
    <occurrences>
      <occurrence accessionRefID="15640" end="258" start="243"/>
    </occurrences>
  </peptide>
  <!-- more peptides-->
</peptidesList>

<groupsList>
  <group id="1089" treeId="1045">
    <accessionsRefList>
      <accessionRef accRefID="12590"/>
```

```
      <accessionRef accRefID="12591"/>
    </accessionsRefList>
    <peptidesRefList>
      <peptideRef pepRefID="9704"/>
    </peptidesRefList>
  </group>
  <group id="1117" treeId="1064">
    <accessionsRefList>
      <accessionRef accRefID="3080"/>
    </accessionsRefList>
    <peptidesRefList/>
    <childrenRefList>
      <childRef childRefID="1116"/>
      <childRef childRefID="1114"/>
    </childrenRefList>
  </group>
  <!-- more groups-->
</groupsList>
</ns3:jPiaXML>
```