technische universität
dortmund

Dissertation

in Fulfillment of the Requirements for the Degree of

Doktor der Naturwissenschaften

Improvement of protein quantification for proteins with shared peptides by using bipartite peptide-protein graphs

Submitted to the Faculty of Statistics

of the TU Dortmund University

by

Karin Ulrike Schork

May 16, 2024

Date of Oral Examination: December 04, 2024

Referees:

Prof. Dr. Jörg Rahnenführer

Prof. Dr. Martin Eisenacher

Prof. Dr. Katja Ickstadt

Acknowledgements

First, I would like to thank Prof. Dr. Martin Eisenacher and Prof. Dr. Jörg Rahnenführer for supervising this thesis project and the many helpful discussion over the last few years. Many thanks also to Prof. Dr. Julian Uszkoreit and Dr. Michael Turewicz for providing helpful comments on the characterization of the peptide-protein graphs and their contribution to the resulting paper.

I would also like to thank my current and former colleagues from the Medical Bioinformatics group for the support throughout the years: Maike Weber, Dominik Lux, Robin Grugel, Dirk Winkelhardt, Dr. Michael Kohl, Dr. Markus Stepath, Anika Frericks-Zipper and Awien Barwari. Many thanks also to all my current and former colleagues at the Medizinisches Proteom-Center for creating such a great working atmosphere. Special thanks also to my students Arne Junge, Katharina Neuhaus, Nils Achterfeldt, Sofian Faiz and Bürkan Kalaycik who contributed to this topic with their bachelor theses, study projects and internships.

Last but not least I thank my family and friends for the support.

Contents

1 Introduction

Proteins are an important class of biomolecules which fulfill important tasks, for example as enzymes or transporter proteins. As many different diseases are associated with changes in the proteome, research on potential protein biomarkers, e.g. for disease diagnosis, is promising. The state-of-the-art method for proteomics biomarker studies is bottom-up mass spectrometry. In this technique, the intact proteins are enzymatically digested to peptides (shorter amino acid chains) before measurement. The data is used to identify and quantify peptides. Quantifications of different peptides within the same sample cannot be compared, only measurements of the same peptide between different samples. Therefore, peptide quantifications are often used in form of peptide ratios, which are the ratios of the peptide intensities in two different samples. For further analysis and interpretation of the results, the peptide quantifications have to be re-assembled to protein quantifications. This step, called protein quantification, is complicated due to the existence of shared peptides which could stem from multiple different proteins.

Bipartite graphs are often used to represent the relationship between proteins and peptides. Figure 1.1 (p. 2) shows an example of a bipartite peptide-protein graph (or more precise: a connected component of a larger graph considering all peptides measured in a data set) together with the corresponding peptide ratios. There are two proteins, protein $Prot1$ and protein $Prot2$ with their corresponding quantified peptides. Peptide $Pep1$ is unique, as it is only assigned to protein $Prot1$. Peptide $Pep2$ is shared between protein $Prot1$ and protein $Prot2$. The peptide ratios r_1 and r_2 are the ratios of the measured peptide intensities between two different samples. The protein ratios R_1 and R_2 are unknown and have to be estimated from the peptide ratios.

There is a large variety of methods for solving the protein quantification problem. While some algorithms employ basic methods like simple arithmetic means of peptide intensities, most methods use more rigorous statistical methods. One category of

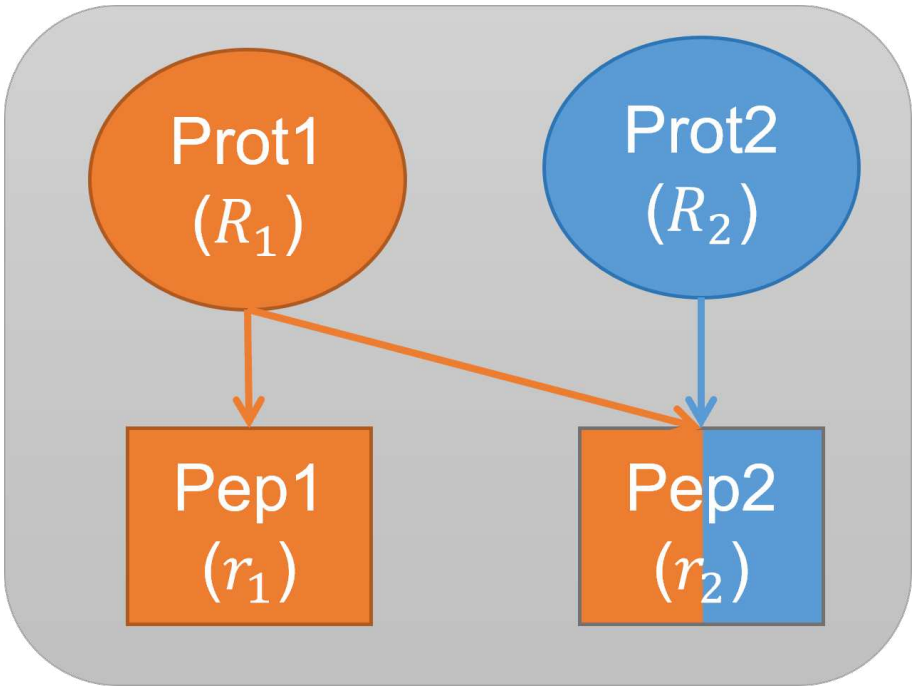


Figure 1.1: Example of a bipartite peptide-protein graph with two protein nodes $Prot1$ and $Prot2$, one unique peptide node $Pep1$ and one shared peptide node $Pep2$. The coloring of the nodes indicates which peptide belongs to which protein. The peptide ratios r_1 and r_2 are measured, while the protein ratios R_1 and R_2 are unknown.

methods filters out unreliable or low-quality peptides before summarizing them to proteins, for example by using hierarchical clustering based on pairwise correlations between peptides. Another large category sets up linear or non-linear models that are solved. These models contain only the peptide quantities or further information on the experimental design (e.g., treatment groups or replicates). Furthermore, some methods employ methods from the field of Bayesian statistics to estimate the protein quantities and their variation.

Despite the large number and variety of available methods, there is still no consensus how to properly make use of shared peptide information during protein quantification. Some commonly used algorithms even rely solely on the unique peptides and do not make use of the information present in the quantities of shared peptides. This makes it impossible to quantify proteins without unique peptides, like $Prot2$ in the example above (figure 1.1). However, proteins without unique peptides can make up a large part of all potential proteins in a sample and quantifying them could be crucial for the subsequent analysis of the data.

The main aim of this PhD project is to develop a novel method that is able to gain quantitative information about proteins without unique peptides and improve the quantification of proteins with unique peptides by adequately using shared peptides. This method called bppgQuant uses bipartite graphs that represent the relationship between proteins and peptides to estimate protein quantities. For each peptide, a nonlinear equation is formed, which relates the unknown protein ratios to the measured peptide ratios. A least-squares approach is applied to estimate the protein ratios by minimizing the sum of squared error terms over all peptides in a bipartite graph. As the set of equations is often underdetermined, multiple optimal solutions are possible, especially for proteins without unique peptides. Therefore, a strategy to calculate an estimate of the whole solution space is developed.

BppgQuant is then evaluated on four different data sets with known protein ratios by spike-in proteins or mixtures of different organisms in pre-defined ratios. The results are compared to the existing protein quantification methods SCAMPI and PQP.

This thesis is structured as follows. Chapter 2 gives an overview about the biological and technical background of bottom-up mass spectrometry, as well as the foundations of peptide quantification, protein inference and protein quantification. In chapter 3 a review of existing state-of-the-art protein quantification methods is provided. Bipartite peptide-protein graphs occurring in different data sets, which are the foundation for the development of the bppgQuant method are characterized in chapter 4. In Chapter 5 the proposed protein quantification method bppgQuant is described in detail, evaluated and compared to the existing methods SCAMPI and PQP. A general discussion and an outlook on further future improvements and developments is given in chapter 6.

2 Background

This chapter describes the biological and technical background of mass spectrometry-based bottom-up proteomics (sections 2.1 - 2.3) and the resulting protein inference and protein quantification problem (section 2.4).

2.1 Proteins and proteomics

The structural units of proteins are amino acids, which consist of a central carbon atom that is connected to a single hydrogen, an amino group (-NH_2), a carboxyl group (-COOH) and a side chain (Müller-Esterl, 2018). The basic structure of an amino acid is depicted in figure 2.1.

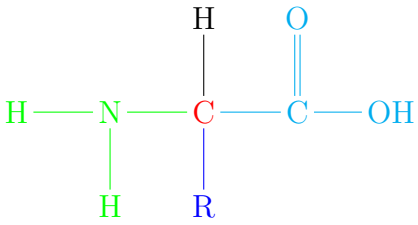


Figure 2.1: General structure of an amino acid with a central carbon atom (red), an amino group (-NH_2, green), a carboxyl group (-COOH, light blue), a hydrogen atom (black) and a side chain -R (dark blue), see Müller-Esterl, 2018, p. 27, taken from Schork, 2016.

There are 21 different amino acids that occur in proteins in the human organism, the so-called proteinogenic amino acids (Müller-Esterl, 2018, pp. 26-29 and 218). They differ in their side chains which may consist of a simple hydrocarbon chain (e.g., lysine) or more complex structures containing carbon rings (e.g., tryptophan) or sulfur (e.g., methionine) (Müller-Esterl, 2018, pp. 28-29). Two amino acids can be connected via covalent peptide bonds. During the chemical reaction, the amino group of the first amino acid is connected to the carboxyl group of the second, while

a water molecule is split off (Müller-Esterl, 2018, p. 31), see figure 2.2. The resulting molecule is called a dipeptide. If multiple amino acids are connected like this in a chain, the resulting molecule is called an oligopeptide or polypeptide (depending on the number of amino acids) or, in short, peptide. Proteins are longer polypeptides (plain amino acid sequences, primary structure) that form secondary structures (alpha helices or beta sheets due to hydrogen bonds) and tertiary structures (non-covalent interactions between secondary structures) (Müller-Esterl, 2018, pp. 74-78). In some proteins two or more polypeptide chains together build a quarternary structure, e.g., hemoglobin which consists of four subunits (Müller-Esterl, 2018, p. 79).

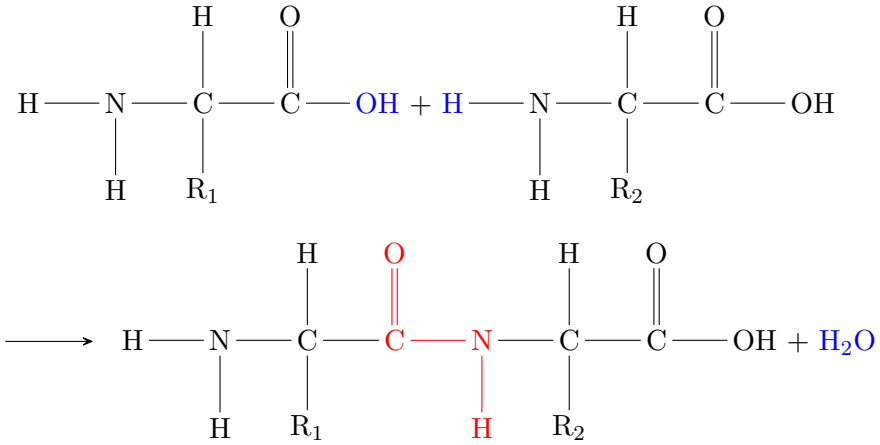


Figure 2.2: Two amino acids connect via a peptide bond (red) and form a dipeptide, while a water molecule (blue) is formed, see Müller-Esterl, 2018, p. 31, taken from Schork, 2016.

The end of the peptide containing the carboxyl group is called C-terminus, while the end with the amino group is called N-terminus. For representing amino acid sequences, the so called one-letter-code is used from N- to C-terminus (Müller-Esterl, 2018, pp. 27-29). Each of the proteinogenic amino acids are represented by one specific letter (e.g., M for methionine).

The genes in the DNA (desoxyribonucleic acid) contain the information about the amino acid sequence of the proteins. The genes are transcribed to mRNA (messenger ribonucleic acid), which is then translated into the amino acid sequence at the ribosome. The genetic code contains nucleotide triplets for the proteinogenic amino acids (Müller-Esterl, 2018, p. 217). However, after translation the amino acids can be further modified, for example oxidized or phosphorylated, which is known under the term post-translational modifications (PTMs), leading to a larger variety of pro-

teins beyond the differences in amino acid sequence (Smith et al., 2013; Carbonara et al., 2021). Some of these modifications have a biological origin, e.g., phosphorylation of the amino acids serine, threonine or tyrosine, that has an important function for signal transduction inside the cells (Müller-Esterl, 2018, pp. 65-66). On the other hand, some modifications are artificially introduced during sample preparation in the laboratory, like carbamidomethylation of cysteine (Kuznetsova et al., 2020).

Proteins are the acting molecules in cells and fulfill many different important functions for example as enzymes, transport proteins or as parts of the cytoskeleton (Müller-Esterl, 2018, pp. 62-63, 338-349 and 466). The set of proteins in a cell or organism is called the proteome. Different diseases are associated with changes in the proteome and therefore proteins are a common target for biomarker studies in biomedicine. Proteins can be detected in different body fluids like urine, blood or cerebrospinal fluid and may serve as biomarkers for the diagnosis and prognosis of a wide range of different diseases. Protein biomarkers have been researched for example for different types of cancer (D'Costa et al., 2016; Borrebaeck, 2017), neurodegenerative diseases like Alzheimer's or Parkinson's disease (Blennow, 2004; Parnetti et al., 2013), cardiovascular disease (Lyngbakken et al., 2019) or COVID-19 infections (Fraser et al., 2020). The field that covers protein analysis is called proteomics and aims at the analysis of the whole proteome in a cell, tissue or organism. Besides medicine, proteins are a subject of research also in other fields, e.g., microbiology (Wagner et al., 2024), plant research (Jorrin Novo, 2021) or forensics (Duong et al., 2021).

2.2 Mass spectrometry-based bottom-up proteomics

The current state-of-the-art method for analyzing the proteome of a sample is bottom-up proteomics based on mass spectrometry (Aebersold and Mann, 2016). Mass spectrometry in general is a technique to measure the mass-to-charge ratio (m/z) and intensity of ions (Aebersold and Mann, 2003). An overview over the general workflow in bottom-up proteomics is given in figure 2.3 (p, 8). Intact proteins can be measured by mass spectrometry via the so-called "top-down proteomics" approach, however this technique faces many challenges for high-throughput use due to the large size and complexity of proteins (Brown et al., 2020; Karch et al., 2022). The term "bottom-up proteomics" refers to a technique where, as a part of sam-

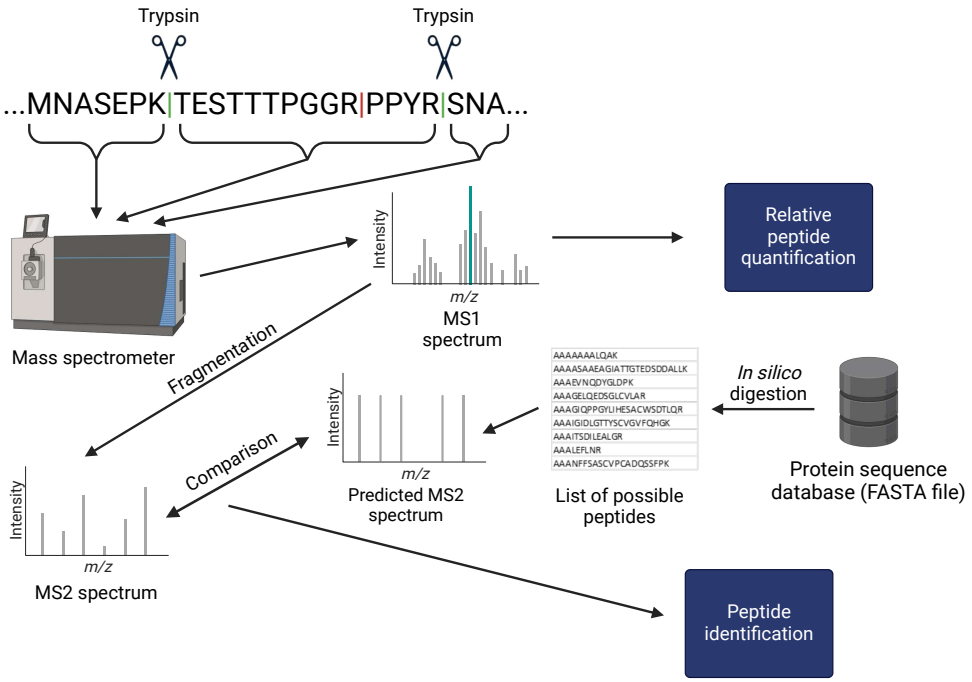


Figure 2.3: General workflow of mass-spectrometry-based bottom-up proteomics. The intact proteins are digested to peptides via the enzyme trypsin. The peptides are measured via mass spectrometry, resulting in MS1 and MS2 spectra. MS1 spectra can be used for relative peptide quantification. The precursor ions belonging to the highest peaks in the MS1 spectra are fragmented and MS2 spectra are acquired. A protein sequence database (FASTA file) is *in silico* digested resulting in a list of possible peptides. For these, theoretical spectra are predicted that are compared with the measured MS2 spectra, resulting in the identification of peptides. Created with `BioRender.com`.

ple preparation, proteins are cleaved into smaller peptides before measurement via mass spectrometry. A commonly used enzyme for protein cleavage is trypsin, which cleaves at the C-terminal side of the amino acids lysine (K) and arginine (R), except if they are followed by a proline (P) (Zhang et al., 2013, see also the peptide sequence in figure 2.3 at the top). This digestion usually generates peptides that are well usable for mass spectrometry, considering their number of amino acids, mass and potential of charged side chains at the C-terminus after ionization (Shuken, 2023).

Before entering the mass spectrometer, the peptides are often separated by a high performance liquid chromatography (HPLC) system (Shi et al., 2004). To avoid that all peptides enter the mass spectrometer at the same time, the peptides pass

a column with a duration that depends on their physico-chemical properties, especially the hydrophobicity (Shuken, 2023). The time at which a peptide elutes from the column is called retention time. The eluting peptides are then ionized, i.e., they become charged molecules, for example by the so-called electrospray ionization (Shuken, 2023).

Inside the mass spectrometer the mass-to-charge ratios (m/z) of the peptide ions (also called precursor ions) are measured by the mass analyzer. The detector measures the corresponding ion intensities. The graph of m/z on the x-axis and intensity on the y-axis is called MS1 spectrum. In the data dependent acquisition (DDA) type measurement, the precursor ions corresponding to the highest peaks in the MS1-spectrum are isolated and further fragmented, e.g., by collision with an inert gas (collision-induced dissociation). During this fragmentation the precursor ions break at different peptide bonds, resulting in a variety of smaller fragment ions. The m/z and intensities of these fragment ions are also measured and result in the so-called MS2 spectrum. The pattern of peaks in the MS2 spectrum together with the mass of the precursor ion can be used to identify the sequence of the precursor peptide ion (Shuken, 2023).

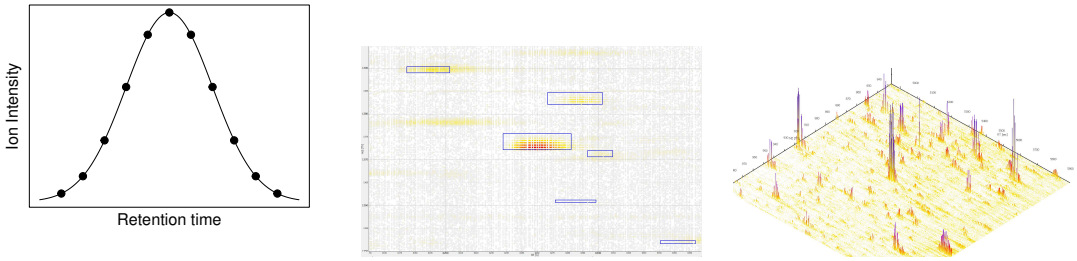
As a basis for peptide identification, protein sequence databases like UniProt (Bateman et al., 2023) are used. In most cases only the sequences of proteins expected to be present in the sample is used, e.g., the proteome of the corresponding organism. Commonly, the protein sequence databases are saved as so-called FASTA files. The protein sequences are digested *in silico*, meaning that the trypsin digestion is mimicked using programming code. The theoretical observable peptide sequences are used to generate theoretically expected MS2 spectra, which can be obtained by calculating the expected m/z of all possible fragment ions and often assigning the same intensity to all peaks belonging to the same type of ion (Eng et al., 1994). Machine and deep learning models have been developed for predicting fragment ion intensities, leading to more accurate theoretical spectra, e.g. MS^{2}PIP (Degroeve et al., 2013) or Prosit (Gessulat et al., 2019). The theoretical and measured MS2 spectra are compared using peptide identification software tools that are also called peptide search engines. A score for the spectrum similarity is calculated, for example the cross correlation (Eng et al., 2008). When a measured spectrum matches to a theoretical one, this is called peptide-spectrum match (PSM). During identification, non-sense protein or peptide sequences are introduced, the so-called decoy sequences, e.g., by reversing or shuffling the original protein sequences from the database (Elias

and Gygi, 2010). They serve as a trap for the identification algorithm and can be used to calculate the false discovery rate (FDR). Then, usually the result list is filtered so that an FDR lower than 1% is achieved to avoid too many false peptide identifications (Shuken, 2023).

2.3 Relative peptide quantification

The data produced by the mass spectrometer can also be used to quantify peptides. Different methods exist for peptide quantification. These methods can be split into two main categories: label-free and label-based quantification (Bantscheff et al., 2012). In label-based proteomics, the peptides or proteins are labelled, for example metabolically by including heavy amino acids using the SILAC approach (Stable Isotope Labeling by Amino acids in Cell culture, Ong et al., 2002) or chemically by TMT (Tandem Mass Tags, Thompson et al., 2003). This allows a simultaneous measurement of multiple samples within a single mass spectrometry run, which leads to a reduction of technical variance (Megger et al., 2014). The different samples can then be distinguished during data analysis by mass shifts (SILAC) or specific reporter fragment ions that split off during fragmentation of the precursor (TMT). However, there are also drawbacks of labelled proteomics, e.g., additional costs for the labels and inaccuracies due to an imperfect labelling efficiency. Additionally, for example TMT data show problems with batch effects and a high proportion of missing values (Brenes et al., 2019). In this thesis, the focus is on label-free quantification techniques, where each sample is processed without labels and separately until the data analysis step (Bantscheff et al., 2012). There are different possibilities to quantify peptides in label-free proteomics, spectral counting, XIC-based or feature-based quantification, which will be explained in the following.

The most simple method for peptide quantification is spectral counting. For each peptide sequence the number of MS2 spectra associated with it (peptide-spectrum-matches, PSMs) are counted. The main idea is that a higher peptide abundance will lead to more MS2 spectra. Different approaches have been proposed to improve spectral counting based quantification, for example NSAF (normalized spectral abundance factor, Zhang et al., 2010), emPAI (exponentially modified protein abundance index, Ishihama et al., 2005) or APEX (absolute protein expression measurement, Lu et al., 2007). Spectral counting-based methods have largely been replaced by the



(a) XIC **(b)** 2D MS1 map with features **(c)** 3D MS1 map

Figure 2.4: (a) Schematic representation of an extracted ion chromatogram (XIC). The dots represent intensity measurements. (b) 2-dimensional representation of an MS1 map, the blue rectangles represent detected features. (c) 3-dimensional representation of an MS1 map. The parts (b) and (c) are screenshots from the ToppView tool (Sturm and Kohlbacher, 2009).

other, more accurate methods for most applications (Geis-Asteggiante et al., 2016; Milac et al., 2012).

Another method is the XIC-based peptide quantification (Shuken, 2023). XIC stands for "eXtracted Ion Chromatogram". For a specific peptide ion (with its specific m/z value) data points for the intensity in the MS1 spectra are collected over the retention time, see figure 2.4(a). The precursor intensity is then calculated as the area under the curve (or the intensity of the highest data point). The resulting intensity is unitless. If a peptide sequence was identified multiple times (e.g., with different charge states or PTMs), the precursor intensities can be aggregated to peptide intensities.

The third option for peptide quantification is the feature-based quantification. A so called MS1 map is used, a representation of the data with m/z, retention time and MS1 intensity as the three axes (see figure 2.4(b) and (c) for a 2- and 3-dimensional representation of the map, respectively). In this map a feature is a set of peaks or an area in the m/z and retention time dimensions that belongs to a specific precursor ion (with a specific charge state). Features can be detected by utilizing the characteristic isotopic patterns. In nature, atoms of elements like carbon, oxygen, hydrogen and nitrogen (which are all commonly present in peptides) occur in their common form but also with additional neutrons in the atom core, which are called isotopes. For example, carbon occurs in the light form ^{12}C (6 neutrons), but also as heavy carbon ^{13}C (7 neutrons). Especially carbon has a comparably high proportion of the heavier isotopes with over 1% of ^{13}C (Valkenborg et al., 2012). Amino acids

and therefore also peptides contain a lot of carbon atoms so that molecules with one, two or even more heavy carbon atoms occur in nature with a high probability. These isotopologues (same peptide sequence with different isotope composition) can be distinguished by the mass spectrometer as an additional neutron leads to an additional mass of approximately one Dalton. This results in mass traces (retention time vs. intensity) that have a characteristic distance on the m/z axes (depending on the charge state of the ion), which are called isotopic patterns. Different algorithms exist for the detection of features, e.g., Dinosaur (Teleman et al., 2016) or the FeatureFinder in OpenMS (Weisser and Choudhary, 2017). After detection, these features can be quantified by integrating over the isotope pattern and the mass trace. Either the peaks within the m/z and retention time area are summed up, or a two-dimensional distribution is fitted to the data and the estimated model parameters are used to calculate the volume of the feature (Nahnsen et al., 2013). Another possibility is to model each mass trace as a normal or normal-exponential hybrid distribution (Weisser et al., 2013; Lan and Jorgenson, 2001).

XIC- and feature-based quantification can also be summarized as intensity-based quantification in contrast to spectral counting. For each of the three described methods the quantification is only relative. As different peptides can have different detection probability and ionization efficiencies, the intensities of two peptides with

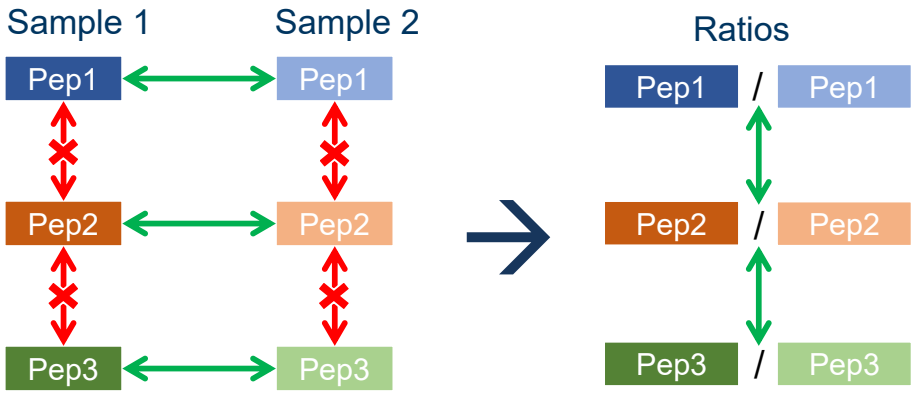


Figure 2.5: Intensities of different peptides cannot be compared directly because of different ionization efficiencies (red, crossed-out arrows). Only intensities of the same peptide over different samples can be compared (green arrows). Therefore, often peptide ratios are calculated, which can be compared for different peptides (green arrows on the right).

the same concentration within a sample can vary a lot (Liigand et al., 2019). Because of this, it is not possible to compare intensities of different peptides within the same sample directly (see figure 2.5, p. 12). However, the comparison between intensities of the same peptide across different samples is possible. Therefore, often peptides ratios between samples or sample groups are calculated.

2.4 The protein inference and quantification problem

While identified and quantified peptides can already serve as biomarkers, calculations on protein level are often preferred because they allow a functional interpretation of the data, e.g., by enrichment analysis for pathways or gene ontology terms (Haw et al., 2011; Xie et al., 2002). Protein inference describes the problem of reporting a list of proteins that is thought to be in the sample considering the peptide-level data at hand (Huang et al., 2012; Li and Radivojac, 2012). This step would be trivial if each identified peptide was unambiguously assignable to a single protein (unique peptides). However, depending on the data set, there may exist a large amount of shared peptides, that can belong to multiple proteins. Reasons for the existence of shared peptides include alternative splicing, protein isoforms or proteins with similar functional groups (Nesvizhskii and Aebersold, 2005). These shared peptides complicate the protein inference step. For an identified peptide shared between two or more proteins it is not clear, from which or from how many proteins it actually stems (protein ambiguity). There is also no direct evidence for a protein, if no unique peptide is identified.

The relationship between proteins and the corresponding peptides that are produced during tryptic digestion is frequently represented by a bipartite graph in bottom-up proteomics (Pavlopoulos et al., 2018), and used to aid protein inference and quantification (Zhang et al., 2007; Gerster et al., 2010; Bamberger et al., 2018; Pfeuffer et al., 2020). Two examples of bipartite peptide-protein graphs are shown in figure 2.6 (p. 14).

Many common algorithms for protein inference are based on the Occam's razor principle, e.g., IDPicker (Zhang et al., 2007) or PIA (Uszkoreit et al., 2015). Given the set of identified peptide sequences, the smallest set of proteins is reported that can explain all of the peptides (set covering problem). Some methods take into account that there is an insecurity in the identification of peptides reflected by a peptide

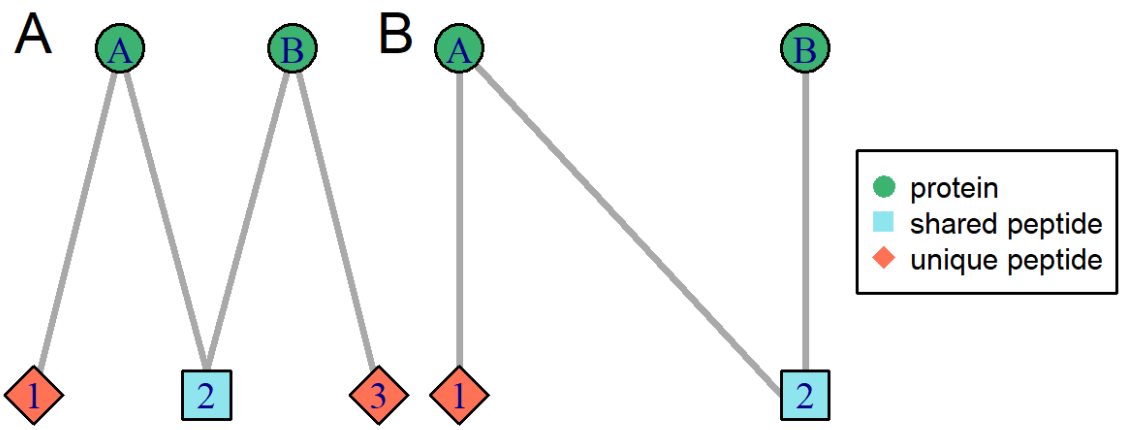


Figure 2.6: Examples of two bipartite graphs representing the relationship between peptides and proteins. Protein nodes are shown as green circles and peptide nodes as red diamonds (unique to a protein node) or blue squares (shared between protein nodes). Nodes may be collapsed and may contain multiple protein accessions or peptide sequences (for details see section 4.2, p. 41).

score or probability, e.g., ProteinProphet (Nesvizhskii et al., 2003) or MSBayesPro (Li et al., 2009). Also, inference algorithms that make use of the quantitative peptide data have been developed, like ProteinClusterQuant (Bamberger et al., 2018) or Quantifere (Lukasse and America, 2014).

Often, the protein inference step includes the formation of protein groups to reflect the protein ambiguity (Jones, 2016). Multiple protein accessions within the database are put together in a protein group, if the inference algorithm cannot decide which protein to report, e.g., because they have the same set of identified peptides. Protein inference is further complicated by the fact that the uniqueness and sharedness of a peptide is always defined for one specific database situation. E.g., a peptide may be unique in a certain database, but shared if more protein sequences are included.

Regarding protein quantification, a similar problem occurs. Peptide quantities have to be summarized to protein quantities. Again, shared peptides are a problem, as the measured peptide intensities may have been influenced by multiple proteins. It is a not clear, which proportion of a certain shared peptide intensity comes from which corresponding protein. Therefore, some quantification methods exclusively use unique peptides or employ a ”razor peptide” approach, which assigns a shared peptide completely to the corresponding protein with highest evidence (e.g., highest number of identified unique peptides) (Cox and Mann, 2008). An overview over methods for protein quantification is given in chapter 3.

3 Existing methods for protein quantification

As explained above, in bottom-up mass spectrometry, proteins are first digested to peptides before measurement. Therefore, the measured intensities are first on peptide and not protein level. There is a large variety of different approaches for summarizing peptide-level quantities in the literature (Blein-Nicolas and Zivy, 2016). This chapter will give an overview about existing statistical methods for this task for label-free, untargeted and intensity-based data (therefore excluding methods for label-based, targeted or spectral counting based data). Furthermore, methods that specialize on absolute quantification are excluded, as this usually involves using reference proteins or peptides that have to be spiked into the sample before measurement (Kettenbach et al., 2011).

In this section, 20 different statistical protein quantification methods are described. These methods can be divided into different categories based on the underlying methodology. There are simple methods that use basic summarization of the acquired peptide quantities, e.g., arithmetic means (section 3.1). Other methods also employ these techniques, but filter the peptides beforehand, e.g., regarding their pairwise correlation or number of missing values (section 3.2). More complex methods rely on linear or non-linear models or equation systems that are solved (section 3.3), while others rely on Bayesian models (section 3.4). Finally, there are some methods that do not fit into one of the other categories (section 3.5). In section 3.6, the methods are compared regarding their handling of shared peptides and ability to quantify proteins without unique peptides.

For better comparability and readability of the model equations, not the original notation of the corresponding papers but a common notation is used as shown in table 3.1 (p. 16). Peptide and protein indices may refer to the set of proteins and peptides within a single connected component of a bipartite peptide-protein graph

or the whole data set, depending on the method. Individual additional notation is introduced at the respective paragraphs for these methods.

Table 3.1: Notation used for explaining the different protein quantification methods.

Notation	Description
$i = 1, \ldots, m$	Index for proteins / protein nodes
$j = 1, \ldots, n$	Index for peptides / peptide nodes
$k = 1, \ldots, K$	Index for the (technical or biological) replicates
$t = 1, \ldots, T$	Index for the treatment or experimental group
$p_{j,X}$	Measured intensity of peptide j in sample X, shortened to p_j if sample is not important
r_j	Measured ratio of peptide j between two samples
$P_{i,X}$	Unknown intensity of protein i in sample X, shortened to P_i if sample is not important
R_i	Unknown ratio of protein i between two samples
δ_{ij}	Indicator if peptide j belongs to protein i
μ	Overall mean or intercept of the model
α	Coefficient for the peptide effect
β	Coefficient for the effect of the experimental condition or treatment group
γ	Coefficient for the effect of biological or technical replicates
ε	Error term

3.1 Basic methods

There are some basic methods that are based on simple averaging of the peptide intensities to obtain protein intensities.

The **topN** method (Silva et al., 2006) calculates the arithmetic mean of corresponding peptide intensities, but using only the peptides with the N highest intensities:

$$\widehat{P}_i = \frac{1}{N} \sum_{j=n-N+1}^{n} p_{(j)}. \tag{3.1}$$

$p_{(1)}, \ldots, p_{(n)}$ are the peptide intensities belonging to protein i ordered from lowest to highest intensity. The high intense peptide ions are assumed to have nearly 100% ionization efficiency and therefore their intensities are more reliable for calculating the protein intensities. It has been shown that a value of N = 3 leads to good

results when calculating the average intensity, as the resulting protein intensity correlates well with the true protein concentrations. However, this method works worse with short proteins with a lower number of available peptides (Silva et al., 2006). Because of its simplicity, this method is popular and implemented in different software solutions.

In the **iBAQ** approach ("intensity-based absolute quantification", Schwanhäusser et al., 2011), the intensities of all corresponding peptides are summed up and divided by the number of theoretically observable peptides N_o:

$$\widehat{P}_i = \frac{\sum_{j=1}^{n} p_j}{N_o}. \tag{3.2}$$

As observable peptides all peptides with a length between 6 and 30 amino acids are considered, derived from an *in silico* digestion of the corresponding protein sequence. Peptides with missed cleavages are not counted. In the original manuscript it is shown that the iBAQ value can be calibrated by using spiked-in proteins to obtain an absolute measure of protein abundance. However, if it is used without the calibration (like it is done in implementations like in MaxQuant), it is simply a relative protein quantification measure.

3.2 Methods based on peptide filtering

A second category of methods relies on the pre-filtering of peptides before applying a summarization method. From this point of view also TopN could fall into this category, as it filters out all peptides that are not among the N highest abundant ones. Filtering out unreliable peptide quantities before protein quantification may in general improve the quality of the peptide data and therefore directly the quality of the calculated protein ratios or intensities. A popular way to filter peptides is by correlation. The basic rational behind this is that peptides belonging to the same protein should in theory correlate well with each other when observed over multiple samples. If one or more peptides do not correlate well with other peptides of the same protein, wrong peptide identification, post-translational modifications or protein isoforms are among the possible reasons (Forshed et al., 2011).

The **Quantifere** method (Lukasse and America, 2014) filters out shared peptides before protein quantification, however, defines uniqueness and sharedness of proteins based on a protein inference algorithm that makes use of the quantitative peptide-level data. First, a standard protein inference based on the Occam's razor principle is applied on each peptide-protein graph. Inferred proteins are called primary proteins, while not inferred proteins, that have shared peptides with the primary proteins, are called secondary proteins. Figure 3.1A shows an example. With Occam's razor, only protein 2 is inferred, therefore it is a primary protein. The proteins 1 and 3 are secondary proteins.

All secondary proteins do not have unique peptides (otherwise they would have been directly inferred by Occam's razor). The aim of the Quantifere method is to additionally infer some of the secondary proteins by using the quantitative peptide

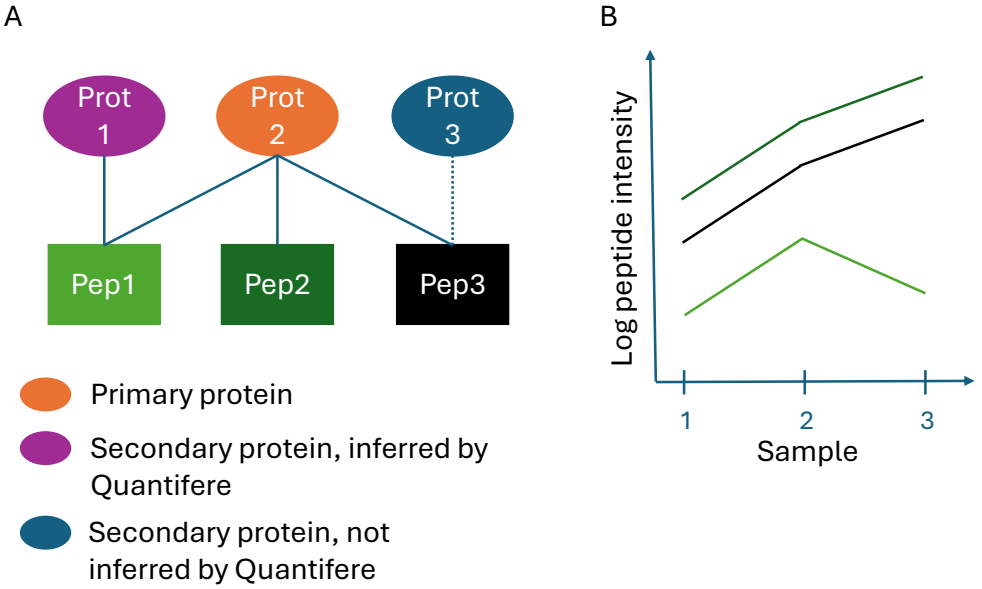


Figure 3.1: Example for Quantifere's protein inference step (Lukasse and America, 2014). A) The peptide-protein graph with inferred primary protein 2 and secondary proteins 1 and 3. Protein 2 is directly inferred by Occam's razor because of the unique peptide 2. B) Graphical representation of the intensity patterns of the three peptides. Peptide 2 and 3 correlate well with each other, while peptide 1 does not correlate well with the other two peptides. Therefore, based on Quantifere's algorithm, the secondary protein 1 is inferred, while protein 3 is not inferred and removed from the graph (indicated by the dotted edge in figure A). Peptide 3 would then be fully assigned to protein 2 for the quantification step.

information. This is done by looking at the pairwise correlations between the peptide intensities (for an example see figure 3.1B). All peptides that are shared by at least one common protein are clustered using agglomerative hierarchical clustering with average linkage. The distance measure D is based on the Pearson's correlation coefficient $\mathrm{Corr}(j_1, j_2)$ between the intensity vectors of two peptides j_1 and j_2:

$$D_{j_1,j_2} = 1 - \mathrm{Corr}(j_1, j_2) \tag{3.3}$$

The clustering algorithm is stopped, when a further iteration would cause the mean correlation between members of a cluster C with $|C| \geq 2$ elements to fall below the threshold τ:

$$\frac{1}{\binom{|C|}{2}} \sum_{j_1 \in C} \sum_{\substack{j_2 \in C \\ j_2 > j_1}} \mathrm{Corr}(j_1, j_2) < \tau. \tag{3.4}$$

τ can be chosen chosen by the user, guided by a comparison of distributions of correlations between pairs of peptides from the same and different sets of proteins. If more than one cluster is found with this approach, this is a hint that the corresponding secondary protein is also present in the sample and causes one or more shared peptides to deviate from the intensity pattern of the other peptides. A secondary protein is inferred if it has at least one peptide that does not cluster with a peptide belonging to another protein within the same graph. E.g, in the example in figure 3.1A, protein 1 would be inferred as a secondary protein, as peptide 1 does not correlate well with the other peptides. Protein 3 is not inferred, as its peptide 3 correlates well with peptide 2.

Non-inferred secondary proteins are removed from the bipartite graph (in the example in figure 3.1A this is indicated by the dotted edge between protein 3 and peptide 3). By this, some previously shared peptides can become unique for the primary proteins (in the example peptide 3). For each primary protein i the now unique peptides are summed up to obtain an estimate for the protein intensity:

$$\hat{P}_i = \sum_{j=1}^{n} p_j \cdot \tilde{\delta}_{ij} \cdot I_{\left\{\sum_{l=1}^{m} \tilde{\delta}_{lj} = 1\right\}}. \tag{3.5}$$

$\tilde{\delta}_{ij}$ is the indicator if peptide j belongs to protein i after Quantifere's protein inference step. The Quantifere procedure aims to improve quantification of primary proteins in contrast to ignoring secondary proteins in general. In the current Quantifere version, secondary proteins cannot be quantified, as very accurate absolute peptide intensity would be needed, as stated by the authors.

The **PQPQ** method ("Protein Quantification by Peptide Quality control", Forshed et al., 2011) works similar to Quantifere regarding correlation and clustering but takes into account the general quality of the peptide identification. First, it defines high-confidence peptides which have a search engine score above a user-defined threshold τ_1 (this depends on the individual search engine). Pearson's correlation coefficient between all high-confidence peptides belonging to the protein of interest are calculated. Peptide pairs are considered as well-correlated if the corresponding p-value (H_0 : Corr ≤ 0 vs. H_1 : Corr > 0) is below a threshold τ_2 (user-defined, default is 0.4). The peptide with the highest intensity that correlates well with most of the high-confidence peptides is chosen as the so-called model peptide for this specific protein. An agglomerative hierarchical clustering is calculated based on the peptide correlation (same distance measure as used in Quantifere, see equation 3.3 on page 19). If more than one cluster is detected, a model peptide is defined for each cluster. In this case, likely two or more proteoforms are present in the data. The set of peptides to use for quantification of a protein i (termed G_i) is then defined as the corresponding cluster members plus low-confidence peptides that correlate well with the model peptide. Finally, the protein ratios are calculated as the arithmetic mean of the peptide ratios of all peptides in the corresponding cluster:

$$\hat{R}_i = \frac{1}{|G_i|} \sum_{j \in G_i} r_j. \tag{3.6}$$

The **PeCorA** method ("PEptide CORrelation Analysis", Dermit et al., 2021) searches for peptides deviating from the patterns of all other peptides stemming from the same protein. First, the peptide-level intensities are centered (mean over all samples $\bar{p}_j$ is substracted), scaled (divided by standard deviation over all samples s_j) and log-transformed:

$$p^*_{j,k} = \log\left(\frac{p_{j,k} - \bar{p}_j}{s_j}\right) \tag{3.7}$$

For each peptide j_1 that is unique to a protein i, a linear model is calculated. This model incudes the peptide group X_1 (the specific peptide j_1 against all other peptides of the protein i) and the treatment effect β_t plus the interaction term of both ($\alpha \times \beta$):

$$p_j^* = \mu_i + \alpha_j X_1 + \beta_t + (\alpha \times \beta)_{jt} + \varepsilon_{ijt}. \tag{3.8}$$

It is tested if there is a significant interaction effect between the treatment group and the peptide group ($H_0 : (\alpha \times \beta) = 0$ vs. $H_1 : (\alpha \times \beta) \neq 0$). An existing interaction (after correcting for multiple testing over the different peptides with the Benjamini-Hochberg procedure) would hint to a different behaviour of the current peptide compared to all other peptides over the different treatment or experimental groups. PeCorA's main aim is to use the detected differences between peptide patterns to detect differentially regulated proteoforms in the data. However, it can also be used as a pre-filtering of discordant (presumably low-quality) peptides before summarizing them to protein intensities.

The **MSstats** R package (Tsai et al., 2020) employs a multi-step procedure to ensure a sufficient quality of peptide intensities before using them to calculating protein intensities.

First, peptides with many missing values are detected by modelling the number of observed peptide intensities larger than 0 for a given protein by a binomial distribution. The corresponding parameter π_i (probability of observing a peptide intensity larger than 0) for each protein is estimated by

$$\hat{\pi}_i = \frac{1}{J} \cdot \sum_{j=1}^{n} \delta_{ij} \cdot \frac{N_{ij}}{K} \tag{3.9}$$

with $J = \sum_{j=1}^{n} \delta_{ij}$ the number of peptides for protein i and $N_{ij} = \sum_{k=1}^{K} I_{\{p_{jk}>0\}}$ the number of peptide intensities larger than 0 for peptide j over all samples. Peptides with a number of missing values that exceeds the 1% quantile of this binomial distribution are excluded from the data set.

As a second step, the following linear model is set up, modelling the log-intensity of the different corresponding peptides $j = 1, \ldots, n$ of a specific protein i:

$$\log(p_{jk}) = \mu_i + \alpha_{ij} + \gamma_{ik} + \varepsilon_{ijk} \tag{3.10}$$

with $\sum_{j=1}^{n} \alpha_{ij} = 0$, $\sum_{k=1}^{K} \gamma_{ik} = 0$, $E[\varepsilon_{ijk}] = 0$ and $Var[\varepsilon_{ijk}] = \sigma_i^2$. μ_i is the mean of the log-transformed intensities of protein i over all samples, α_{ij} is the peptide effect and γ_{ik} is the sample effect. The parameters are estimated by the robust M-estimation method to reduce the effect of missing values, outliers and noise.

Third, for each protein an estimate $\tilde{\sigma}_i^2$ of the representative protein variation is calculated, based on the empirical Bayes model used in the limma method (Ritchie et al., 2015). Single outlier data points within a peptide are removed when the corresponding residual of the linear model (equation 3.10, p. 22) fulfills the three-sigma rule for outlier detection:

$$|\hat{\epsilon}_{ijk}| > 3 \cdot \tilde{\sigma}_i. \tag{3.11}$$

As a fourth step, the residuals are used to determine noisy features by comparing the variance of the residuals with the representative protein variation. Those are also excluded for protein quantification. After removing the noisy peptides, the model in equation 3.10 (p. 22) is calculated again and the estimated log-transformed protein intensities are then estimated as

$$\widehat{\log(P_{i,k})} = \hat{\mu}_i + \hat{\gamma}_{ik}. \tag{3.12}$$

The **apQuant** method ("Accurate and Precise QUANTification", Doblmann et al., 2019) calculates seven quality metrics on the chromatograms (XIC-based peptide quantification) of each peptide and a set of decoy peptides. These metrics are based on retention time differences (between different runs or isotopes within the same run), retention time duration and mass deviations from the theoretical mass. These data are then used as the input of the Percolator tool (Käll et al., 2007), which uses a support vector machine to classify targets and decoys and refines the calculation of the false discovery rate (FDR). Peptides resulting in an FDR >5% by this approach are excluded before calculating iBAQ or TopN values (see section 3.1, p. 16). By this procedure inconsistencies and errors in the quantification of peptides are detected

and removed before protein summarization. Shared peptides are handled by the razor-peptide-approach similar to the MaxLFQ algorithm (see section 3.5, p. 30).

3.3 Methods based on (non)-linear models or equation systems

Many protein quantification methods set up linear or non-linear models of varying complexity to estimate the unknown protein quantity based on the measured peptide quantities. In some more complex models, information on experimental or treatment groups as well as technical or biological replicates are incorporated. In the previous section 3.2 (Methods based on peptide filtering), the methods PeCorA and MSStats already use linear models. However, the focus of these methods was not to build the model for protein quantification directly, but to use the gained information about the model coefficients or residuals to filter out peptides before summarization to proteins. Therefore, they were classified as methods using peptide filtering approaches and not linear models.

In the **PQP** method (Dost et al., 2009; Dost et al., 2012), for each peptide in a peptide-protein graph, an equation is formed that connects the unknown protein intensities B_i and A_i, $i = 1, \ldots, n$ in the samples A and B with the measured peptide ratios between samples A and B. As the equations are usually not exactly solvable, the aim is to find estimates for the protein intensities that have minimal errors. The sum of absolute errors is minimized by using a linear programming approach. The protein intensities in sample A are scaled to a total sum of 100 to remove one degree of freedom. The optimization problem is then formulated as:

$$\min \sum_{j=1}^{m} |\varepsilon_j| \tag{3.13}$$

subject to

$$\sum_{i=1}^{m} A_i = 100 \text{ and } A_i, B_i \geq 0 \quad \forall i = 1, \ldots, m \tag{3.14}$$

with

$$\begin{aligned} \varepsilon_j = & \sum_{i=1}^{m} \delta_{ij} \cdot A_i - r_j \cdot \sum_{i=1}^{m} \delta_{ij} \cdot B_i \; \forall j = 1, \ldots, n, \quad \text{if } r_j \geq 1 \\ \varepsilon_j = r_j \cdot & \sum_{i=1}^{m} \delta_{ij} \cdot B_i - \sum_{i=1}^{m} \delta_{ij} \cdot A_i \; \forall j = 1, \ldots, n, \quad \text{if } r_j \leq 1. \end{aligned} \tag{3.15}$$

The method **SCAMPI** ("Statistical Model for Protein quantification", Gerster et al., 2014) models the measured peptide intensities in the following linear model in relationship to the sum of unknown protein intensities

$$\log_{10}(p_j) = \alpha_0 + \alpha_1 \sum_{j=1}^{n} \delta_{ij} \log_{10}(P_i) + \varepsilon_j. \tag{3.16}$$

It is assumed that all $\log_{10}(P_i)$ as well as all ε_i are independent and identical normal distributed, $\log(P_i) \sim N(\mu_i, 1)$ and $\varepsilon_i \sim N(0, \sigma_i^2)$. $\alpha_0, \alpha_1, \mu_i$ and σ_i are unknown parameters that need to be estimated. For parameter estimation, maximum likelihood estimation (MLE) as well as indirect least squares estimation (ILSE) approaches are compared. The MLE approach makes use of the assumed normal distribution of the log-transformed protein intensities. As there is an equation for each single peptide, also for shared ones, a protein score $\widehat{\log(P_i)}$ is calculated for each single protein with at least one quantified peptide, even if no unique peptide is present.

The **HIquant** method (Malioutov et al., 2019) models the peptide intensities as the sum of the corresponding unknown protein intensities times a peptide-specific nuisance parameter z_j:

$$p_{j,k} = z_j \cdot \sum_{i=1}^{m} \delta_{ij} P_{j,k}. \tag{3.17}$$

This can also be written in matrix notation as

$$\begin{pmatrix} p_{1,1} & \ldots & p_{1,K} \\ \vdots & \ddots & \vdots \\ p_{n,1} & \ldots & p_{n,k} \end{pmatrix} = \begin{pmatrix} z_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & z_n \end{pmatrix} \begin{pmatrix} \delta_{11} & \ldots & \delta_{1m} \\ \vdots & \ddots & \vdots \\ \delta_{n1} & \ldots & \delta_{nm} \end{pmatrix} \begin{pmatrix} P_{1,1} & \ldots & P_{1,K} \\ \vdots & \ddots & \vdots \\ P_{m,1} & \ldots & P_{m,K} \end{pmatrix} \tag{3.18}$$

or short as

$$\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{\Delta}\boldsymbol{P}. \tag{3.19}$$

$\boldsymbol{X} \in \mathbb{R}^{n\times K}$ is the matrix of measured peptide intensities with different peptides in rows and the different samples in columns. $\boldsymbol{\Delta} \in \mathbb{R}^{n\times m}$ is the known design matrix (known relationship between proteins and peptides containing the δ_{ij}), while the nuisance parameters $\boldsymbol{Z} \in \mathbb{R}^{n\times n}$ (diagonal matrix with z_j on the diagonal) and matrix of the protein intensities $\boldsymbol{P} \in \mathbb{R}^{m\times K}$ are unknown. Different solver algorithms are available in HIQuant to solve the equation system, e.g., a quadratic programming based solver or a coordinate descent based solver. HIQuant uses shared peptides in the equations but builds the model only on proteins with at least one unique peptide. That means that a quantification of proteins without unique peptides is impossible.

Karpievitch et al., 2009 propose a linear model that introduces effects of the treatment or experimental group. The model formula for peptide j that belongs to protein i in sample k of the treatment group t is given by:

$$\log_2(p_{j,tk}) = \mu_i + \alpha_{ij} + \beta_{it} + \varepsilon_{ijtk}. \tag{3.20}$$

μ_i is the mean intensity of protein i over all samples, α_{ij} is the effect of peptide j on protein i with $\sum_{j=1}^{n} \alpha_{ij} = 0$ and β_{it} is the group effect with $\sum_{t=1}^{T} \beta_{it} = 0$. The error term is assumed to be normal distributed with a variance specific to the peptide but independent from the treatment group: $\varepsilon_{ijtk} \sim N(0, \sigma_{ij}^2)$. A maximum likelihood approach is used under the assumption of a normal distribution to estimate the protein intensities μ_i and the group effects β_{it}. Mechanisms for left censored peptide intensities from low-abundant peptides (below the quantification limit) and random missing values are incorporated into the likelihood function.

The **all-P** model (Blein-Nicolas et al., 2012) is a log-linear mixed effect model that models the peptide intensities depending on treatment group t and replicate k. The model formula is given by

$$\log(p_{j,tk}) = \log\left(\sum_{j=1}^{n} \delta_{ij} P_{it}\right) + \log(\alpha_j) + \beta_{tk} + \gamma_k + \varepsilon_{jtk}. \tag{3.21}$$

α_j is the coefficient of proportionality that allows to calculate peptide intensities measured by MS by the true abundance (peptide-specific factor due to different ionization efficiencies). γ_k is the effect of the biological variation between replicates, β_{tk} stands for the technical variation in treatment group t and replicate k. P_{it} is considered a fixed effect, while $\log(\alpha_j) \sim N(0, \sigma_\alpha^2)$, $\gamma_k \sim N(0, \sigma_\gamma^2)$ and $\beta_{tk} \sim N(0, \sigma_\beta^2)$ are considered as random effects following normal distributions. For shared peptides the sum has more than one summand, leading to a non-linear model. To calculate the estimated protein intensities, a Bayesian hierarchical framework is used.

The **MSqRobSum** (Sticker et al., 2020) method models the log-transformed protein intensity as a mixed model depending on the peptides, sample and experimental group. The original MSqRob model formula is given by (Goeminne et al., 2016)

$$\log(p_{jtk}) = \mu + \alpha_j + \beta_t + \gamma_k + \varepsilon_{jtk}. \tag{3.22}$$

In MSqRobSum this model is estimated by a regression analysis with two stages. First, the model

$$\log(p_{jk}) = \alpha_j + \gamma_k + \varepsilon_{jk}. \tag{3.23}$$

is calculated via the robust regression with M-estimation. This model summarizes the peptide intensities to protein intensities. As the second step the model

$$\log(P_{jk}) = \mu + \beta_t + \varepsilon_{tk}. \tag{3.24}$$

is calculated to estimate the effect of the experimental group.

A different linear mixed effect model is used by **Clough et al., 2012** that incorporates effects from the peptide, experimental group, biological replicate as well as the interaction between peptide and experimental group. The following formula is used:

$$\log(P_{jtkl}) = \mu + \alpha_j + \beta_t + \gamma_k + (\alpha \times \beta)_{jt} + \varepsilon_{jtkl} \tag{3.25}$$

with $\alpha_1 = \beta_1 = (\alpha \times \beta)_{j1} = (\alpha \times \beta)_{1t} = 0$ and $\varepsilon_{jtkl} \sim N(0, \sigma_{\varepsilon,jtk}^2)$. The interaction term between peptide and condition $(\alpha \times \beta)$ is used which can only be considered

if at least two peptides are quantified for a specific protein. An estimate for the protein intensity can then be calculated as:

$$\widehat{\log(P_{jtkl})} = \widehat{\mu} + \frac{1}{n}\sum_{j=1}^{n}\widehat{\alpha}_j + \widehat{\beta}_t + \frac{1}{K}\sum_{k=1}^{K}\widehat{\gamma}_k + \frac{1}{n}\sum_{j=1}^{n}\widehat{(\alpha \times \beta)}_{jt}. \tag{3.26}$$

3.4 Methods based on Bayesian models

Another category of protein quantification methods uses approaches and models from the field of Bayesian statistics.

Diffacto (Zhang et al., 2017) adapts a Bayesian factor analysis FARMS ("Factor Analysis for Robust Microarray Summarization") that previously has been used in the context of microarray data (Hochreiter et al., 2006). FARMS uses the equation system

$$\boldsymbol{x} = \boldsymbol{\lambda} \cdot z + \boldsymbol{\varepsilon}. \tag{3.27}$$

$x_j = \log(r_j), j = 1, \ldots, n$ are the log-transformed peptide ratios, $z \in \mathbb{R}$ is the factor corresponding to the log-transformed relative protein concentration, $\boldsymbol{\lambda} \in \mathbb{R}^n$ is the vector of the loadings and $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is the vector of error terms. The following (multivariate) normal distributions are assumed:

$$z \sim N(0,1),\ \boldsymbol{\varepsilon} \sim N(\mathbf{0},\ \boldsymbol{\Psi}), \boldsymbol{x} \sim N(\mathbf{0}, \boldsymbol{\lambda}\boldsymbol{\lambda}^T + \boldsymbol{\Psi}). \tag{3.28}$$

In the Bayesian factor analysis, the prior distribution of the diagonal noise variance matrix $\boldsymbol{\Psi}$ is chosen as non-informative. The prior distribution for the loadings $\boldsymbol{\lambda}$ is the product of n rectified normal distributions:

$$p(\boldsymbol{\lambda}) = \prod_{j=1}^{n} p(\lambda_j) \tag{3.29}$$

with

$$\lambda_j = \max\{y_j, 0\} \text{ with } y_j \sim N(\mu_\lambda, \sigma_\lambda) \text{ and } \sigma_\lambda^2 = \rho \cdot \frac{1}{n} \cdot \sum_{j=1}^{n} Var(x_j). \tag{3.30}$$

The hyperparameters ρ and μ_λ are estimated by the Expectation-Maximization algorithm. The loadings $\boldsymbol{\lambda}$ are transformed to a weight vector $\boldsymbol{w} = (w_1, ..., w_n)^T \in \mathbb{R}^n$ by the following equation:

$$w_j = \begin{cases} 0, \text{ if } \lambda_j < \frac{1}{2} \cdot \max\limits_{l=1,...,n} \lambda_l \\ \lambda_j, \text{ otherwise} \end{cases} \tag{3.31}$$

The weights are equal to the loadings, except if the loading is lower than half the maximum loading. In this case the weight is set to zero. This leads to an exclusion of peptides that do not correlate well with other peptides and a higher weighting of correlated peptides. Finally, the log-transformed protein ratio is calculated by a weighted mean of the corresponding peptide ratios:

$$\widehat{\log(R_i)} = \frac{\sum\limits_{j=1}^{n} w_j \cdot \log(r_j)}{\sum\limits_{j=1}^{n} w_j}. \tag{3.32}$$

It is not clear how Diffacto incorporates shared peptides, however, the software documentation mentions that shared peptides are used by default.

The **Triqler** method (The and Käll, 2019; Truong et al., 2023) employs a Bayesian model to calculate protein ratios by incorporating different types of error sources that may influence the estimate on the way from the raw mass spectra to the protein ratios. These error sources include ionization efficiency, probability of errors during the peptide identification, as well as group effects and missing values. A graphical representation of the probablistic graphical model for a single protein can be found in figure 3.2 (p. 29).

An estimate for the mean protein quantity in a experimental group $z_t, t = 1, \ldots, T$ and the corresponding variance in a group $s_t, t = 1, \ldots, n$ is the final outcome of the model. These two variables influence the protein intensity P_{tk} for each replicate

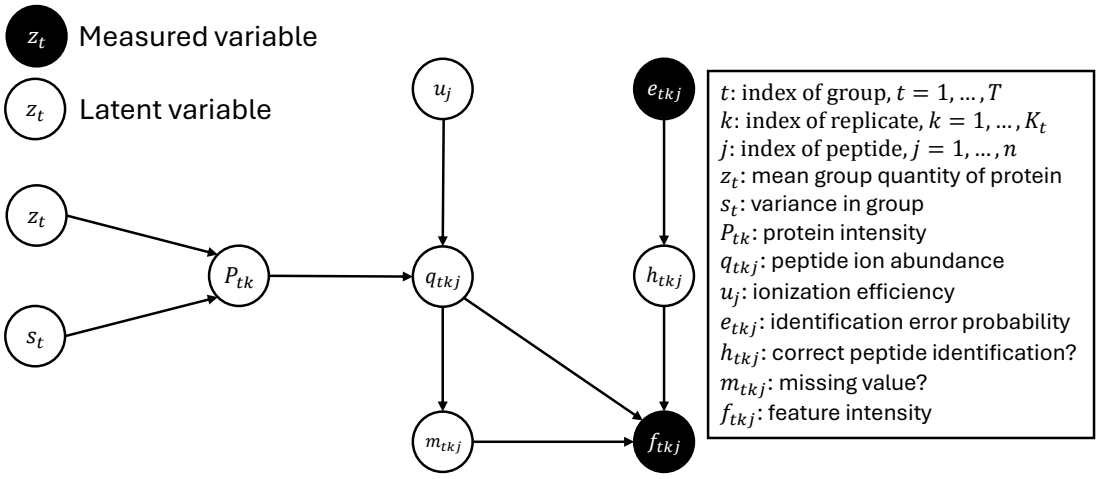


Figure 3.2: Probablistic graphical model for a single protein in the Triqler method, adapted from figure S1 in (The and Käll, 2019).

$k = 1, \dots, K_t$ in group t. Together with the peptide-specific ionization efficiency $u_j, j = 1, \dots, n$, the protein intensity has influence on the abundance q_{tkj} of the respective peptide ion j. The peptide ion abundance is influenced by two further error sources. The first one is a hidden variable h_{tkj} which represents if the peptide identification was correct or not, which is influenced by the error probability e_{tkj}, which is calculated by the search engine. Second, the hidden variable m_{tkj} models a censoring mechanism that may lead to a missing value in the final feature intensity f_{tkj}.

Different prior distributions are used for the different variables. The standard deviation s_t is modelled as a gamma distribution with parameters a_s and b_s: $s_t \sim \text{Gamma}(a_s, b_s)$. s_t and z_t are used as a parameters for a hyperbolic secant distribution, which models the drawing of the protein intensities from the group population, leading to the following probability distribution:

$$\text{hypsec}(x; z_t, s_t) = \frac{1}{\pi \cdot s_t} \text{sech}\left(\frac{x - z_t}{s_t}\right) \tag{3.33}$$

Another hyperbolic secant distribution is used for modelling the log-transformed protein intensity $\log(P_{tk})$ relative to the corresponding protein intensity in all other replicates ($k = 1, \dots, K_t$) with parameters μ_y and σ_y. The difference between the log-transformed feature intensity and the log-transformed peptide ion abundance

$\log(f_{tkj}) - \log(q_{tkj})$ is also represented by a hyperbolic secant distribution with parameters μ_d and σ_d. Uninformative priors are used for the ionization efficiency u_j and the mean protein quantity per group z_t. Finally, the prior distribution for the measured log-tranformed feature intensity $\log(f_t kj)$ is chosen as a censored normal distribution, accounting for a rising probability of a missing value for peptides with low intensities:

$$\text{censnorm}(x) = \left(\frac{1}{2} + \frac{1}{2} \cdot \tanh\left(\frac{x - \mu_m}{\sigma_m}\right)\right) \cdot N(x|\mu_f, \sigma_f). \tag{3.34}$$

In total, ten hyperparameters have to be estimated, a_s and b_s for the group variance, μ_y and σ_y for the protein intensity, μ_m and σ_m for the censoring mechanism, μ_d and σ_d for the uncertainty in quantifying a feature and μ_f and σ_f for the feature intensity. The hyperparameters are estimated by an empirical Bayes approach. For example, μ_f, σ_f, μ_m and σ_m are estimated by fitting the corresponding censored normal distribution (see equation 3.34) to a histogram of the log-transformed measured feature intensities. μ_y and μ_d are expected to be close to zero, as no systematic bias is expected.

After estimating the hyperparameters, the posterior distribution for the protein intensity per experimental group can be calculated, which highlights the uncertainty of the estimated protein ratios in comparison to methods that only deliver a single point estimate. Currently, the probablistic graphical model represents a single protein and only unique peptides are used.

3.5 Other methods

This category describes methods that cannot be assigned to any of the categories described above.

The **MaxLFQ** algorithm (Cox et al., 2014) is implemented in the MaxQuant software. By default, shared peptides are fully assigned to the protein which has the most evidence (razor peptide-approach). A normalization step reduces differences between the different samples and is especially suited for fractionated samples. The actual protein quantification algorithm works as follows: For all quantified peptides, the intensity ratios r_{j,k_1k_2} between all pairwise combinations of two samples $k_1, k_2 \in$

$\{1, \ldots, K\}$ are calculated. If peptide intensities are missing in at least one of the two conditions, this peptide ratio is set to `NA`. G_{i,k_1k_2} is defined as the set of indices of peptides that belong to protein i and have a non-missing peptide ratio:

$$G_{i,k_1k_2} = \{j : \delta_{ij} = 1 \wedge r_{j,k_1k_2} \neq \texttt{NA}\}. \tag{3.35}$$

The protein ratio for a specific combination of samples is calculated as the median of the corresponding non-missing peptide ratios. However, this is only seen as valid if at least two peptide ratios could be calculated for this specific sample pair (i.e., if both samples had intensities larger than zero for these two peptides):

$$\widehat{R}_{i,k_1k_2} = \begin{cases} \underset{j \in G_{i,k_1k_2}}{\text{median}}(\{r_{j,k_1k_2}\}) \\ \texttt{NA} \text{ if } \; |G_{i,k_1k_2}| < 2 \end{cases} \tag{3.36}$$

As a final step, the set of calculated protein ratios over the different sample pairs is used to calculate estimates for the protein intensities. This is done by solving a non-linear equation system:

$$\widehat{R}_{i,k_1k_2} = \frac{P_{k2}}{P_{k1}} \quad \forall \; k_1, k_2 \text{ with } \widehat{R}_{i,k_1k_2} \neq \texttt{NA} \tag{3.37}$$

If a protein has only invalid protein ratios for the comparison of one specific sample with any other sample, the protein intensity for this sample is set to zero:

$$\widehat{P}_k = 0 \quad \forall \; k = 1, \ldots, K \text{ if } \widehat{R}_{i,k_1k_2} = \texttt{NA} \quad \forall \; k_1, k_2 \in \{1, \ldots, K\} \tag{3.38}$$

The reasoning behind this approach is that consistent peptide identifications across samples are seen as a sign of high quality. If a certain sample quantifies a different set of peptides than the other samples for a certain protein, it is seen as unreliable for contributing to the protein quantification.

gpGrouper (Saltzman et al., 2018) is a gene-centric algorithm for inference and quantification. One gene can be the template for multiple proteins, e.g., through the effect of alternative splicing, leading to protein isoforms. However, isoforms often cannot be distinguished by bottom-up mass spectrometry due to the lack of unique

peptides for the specific isoform, this is why a gene-centric approach is proposed here. The method is an improvement of the iBAQ method (see section 3.1, p. 16). The intensities of shared peptides are divided between the corresponding genes by using the ratio of the unique-to-gene peptides. This method is specifically designed for situations where multiple organisms are mixed, like for xenograft models, as it avoids protein groups that overlap between species. Especially peptides unique to a certain organism can help to calculate the fraction of involved species.

The **ProRata** algorithm (Pan et al., 2006) was first described for labelled peptides but was later adapted for label free quantification (Wang et al., 2013). It uses a profile likelihood algorithm to calculate point as well as confidence interval estimations. In brief, peptide ratios as well as signal-to-noise ratios are incorporated into a likelihood function. The probability distribution of logarithmized peptide ratios is modelled as a mixture distribution between a normal distribution and a uniform distribution $0.85 \cdot N(\mu_j, \sigma_j) + 0.15 \cdot U$. The uniform distribution part is introduced to limit the influence of outliers that cannot be modelled by the normal distribution alone. The likelihood function is evaluated on a grid of possible values in the interval $[-7, 7]$ for the log-transformed protein ratios to find the maximum.

The **directLFQ** method (Ammar et al., 2023) aims at the fast processing of large data sets. This method defines a sample intensity trace for a specific sample k as the vector of log2-transformed intensities for different precursor ions:

$$\boldsymbol{p}_{\text{sample},k} = (\log_2(p_{1,k}), \ldots, \log_2(p_{n,k}))^T. \tag{3.39}$$

Similarly, a ion intensity trace for a specific peptide j is defined as the vector of log2-transformed peptide intensities over all samples:

$$\boldsymbol{p}_{\text{ion},j} = (\log_2(p_{j,1}), \ldots, \log_2(p_{j,K}))^T. \tag{3.40}$$

In an iterative procedure, the two most closest traces (lowest variance of differences) are identified and shifted above each other by adding and substracting the median of the differences, respectively. Then, the two traces are combined by averaging them. This procedure is repeated until all traces are aligned. This procedure is done once on the sample intensity traces for normalization and then on the ion intensity traces. Then, the protein intensities are calculated as the median of the

normalized ion intensity trace. The log2-transformation is undone and a further scaling is applied so that the total sum of peptide intensities matches the total sum of protein intensities.

3.6 Usage of shared peptides

Looking at the large variety of available protein quantification methods summarized in this chapter, the question arises why the development of a further method is necessary. It has been shown that using shared peptides can clearly improve the protein quantification (Zhang et al., 2015; Jin et al., 2008; Dost et al., 2012), however, many methods rely heavily on the unique peptides.

In table 3.2 (p. 35) the usage of shared peptides as well as the ability to quantify proteins without unique peptides is shown for the methods described in sections 3.1 to 3.5. Out of the 20 methods, nine clearly state the usage of shared peptides for protein quantification, six methods do not use shared peptides and for five it is not clear or depends on the search engine output. From the nine methods using shared peptides, apQuant and MaxLFQ use a so-called "razor-peptide" approach where each shared peptide is completely assigned to the protein with highest evidence, e.g., the protein with most unique peptides. Karpievitch et al., 2009 use a similar approach, but randomly assign the shared peptides. While these approaches make use of shared peptides, their nature, meaning that their intensities are influenced by all the underlying protein intensities, is ignored and may lead to a bias in protein quantification.

Naturally, older methods like topN, iBAQ or ProRata may remove shared peptides or do not mention them in the method description. Using only unique peptides makes the quantification process easier and at the start of quantitative proteomics this may have been a good compromise (Ong and Mann, 2005). Largely, also the "two-peptide" rule was applied, demanding at least two identified unique peptides to infer a protein (to rule out one-hit-wonders that could have been found because of a single identification error). This rule was criticized later (Gupta and Pevzner, 2009) but is often still employed in the context of protein quantification.

Even if a method uses shared peptides in their algorithm, it does not automatically mean that all proteins without unique peptides can also be quantified. For example

razor peptide approaches like proposed by MaxLFQ assign each shared peptide to one protein based on further evidence, which strongly favors proteins with unique peptides, like the Occam's razor protein inference strategy. Proteins without unique peptides may only be quantified in cases where those shared peptides are not also present in other proteins with more prior evidence, e.g., in form of unique peptides. As another example, HIquant uses shared peptides in their equation system, but require at least one unique peptide for each protein to quantify it. Even modern method like Triqler, MSStats or PECorA do not incorporate shared peptides. This may be explained by the rising complexity of the methods itself so it is possible that the avoidance of the additional complexity level of shared peptides may make these methods more feasible. Only PQP and SCAMPI mention explicitly the possibility to quantify proteins without unique peptides.

In summary, even many modern protein quantification methods neglect the influence of shared peptides and quantification without unique peptides is a challenge that only a few methods approach. As the number of shared peptides and proteins without unique peptides is high (as shown later in section 4.4.3, see p. 53), there is great potential in developing a method particularly focusing on this issue. In chapter 5 the method bppgQuant is proposed that takes into account shared peptides and also allows quantification of proteins without any unique peptide.

Table 3.2: Summary of protein quantification methods and information about their usage of shared peptides.

Type	Method	Shared peptides used?	Quant. without unique peptides possible?	Comment
Basic	topN (Silva et al., 2006)	Unclear	Unclear	Depends on implementation
	iBAQ (Schwanhäusser et al., 2011)	Unclear	Unclear	Depends on implementation
Filtering	Quantifere (Lukasse and America, 2014)	No	No	Shared peptides only used for inference
	PQPQ (Forshed et al., 2011)	No	No	Only isoform-unique peptides are used
	PeCorA (Dermit et al., 2021)	No	No	
	MSStats (Tsai et al., 2020)	No	No	
	apQuant (Doblmann et al., 2019)	Yes	Depends	Razor peptide approach
(non-)linear model	PQP (Dost et al., 2012)	Yes	Yes	
	SCAMPI (Gerster et al., 2014)	Yes	Yes	
	HIquant (Malioutov et al., 2019)	Yes	No	
	(Karpievitch et al., 2009)	Yes	No	Shared peptides are randomly assigned to one of the proteins
	all-P (Blein-Nicolas et al., 2012)	Yes	Unclear	
	MSqRobSum (Sticker et al., 2020)	Unclear	Unclear	
	(Clough et al., 2012)	Unclear	Unclear	
Bayes	Triqler (The and Käll, 2019)	No	No	
	Diffacto (Zhang et al., 2017)	Yes	Unclear	Software allows setting to exclude shared peptides, default is FALSE
Other	MaxLFQ (Cox et al., 2014)	Yes	Depends	Razor peptide approach
	gpGrouper (Saltzman et al., 2018)	Yes	No	
	ProRata (Pan et al., 2006)	No	No	Shared peptides are removed
	directLFQ (Ammar et al., 2023)	Depends	Unclear	Depends on the output of the search engine

4 Characterization of bipartite peptide-protein graphs

The relationship between proteins and the corresponding peptides, which are produced during trypsin digestion, is frequently represented by bipartite graphs in bottom-up proteomics (Pavlopoulos et al., 2018) and used to aid protein inference and quantification (Zhang et al., 2007; Gerster et al., 2010; Bamberger et al., 2018; Pfeuffer et al., 2020). Here, these graphs will be called bipartite peptide-protein graphs. While these graphs are frequently used as a representation, a comprehensive analysis of those occurring either in real quantitative data sets or in theory by an *in silico* digestion of a protein sequence database has not been performed yet. The only step into this direction is the graphical representation of the bipartite graphs of a complete data set shown by Bamberger et al., 2018, however no further analysis of this has been done.

In chapter 5 the structure of the bipartite graphs will be used to define a new method for protein quantification, called bppgQuant. As a basis for this, an overview of the behaviour and the occurrence of graph types is necessary to judge the complexity of the protein quantification problem and to highlight the importance of shared peptides. Also, analyzing the results regarding the occurrence of proteins without unique peptides is useful, as this will be a focus of bppgQuant. In this chapter 4, the bipartite peptide-protein graphs that occur in four different data sets are characterized. Graphs are generated on basis of the measured quantitative peptide-level data sets and on basis of the theoretical *in silico* digestion of the corresponding protein databases (FASTA files) to peptides.

The contents of this chapter is largely based on the following publication:

Schork K, Turewicz M, Uszkoreit J, Rahnenführer J, Eisenacher M (2022). Characterization of peptide-protein relationships in protein ambiguity groups via bipartite graphs. PLOS ONE 17(10): e0276401. `https://doi.org/10.1371/journal.pone.0276401`.

As a first author I was responsible for developing the methodology, analyzing the data, implementing the corresponding software, visualization and writing of the original draft. Julian Uszkoreit assisted with the preparation of the peptide-level data. Michael Turewicz, Jörg Rahnenführer and Martin Eisenacher contributed to and had influence on the methodology and editing of the manuscript draft. As colast authors, Jörg Rahnenführer and Martin Eisenacher took over the supervision of the project.

The analysis in this paper was repeated for this thesis by putting more emphasis on comparability between data sets, meaning that the peptide-level quantitative data are now produced with exactly the same software versions, the same search engine settings and the same version of the UniProt database (Bateman et al., 2023). Additionally, a fourth data set D4 was added which is a mixture of human and mouse proteins (for details see section 4.1). Furthermore a section on the influence of missed cleavages on the graphs is added. Also, the computational construction of the bipartite graphs was optimized by building them from edgelists instead of adjacency matrices (for details see section 4.3). The computation time was further decreased by using more build-in functions from the `igraph` R package (Csárdi and Nepusz, 2006) to deal with the graphs.

The chapter is structured as follows. First the used data sets, the corresponding protein databases and the preprocessing of the data is described (section 4.1). Then, the construction of the bipartite peptide-protein graphs based on the database and the quantitative data is explained (section 4.2). A section about the software implementation and technical challenges follows (section 4.3). Then, the graphs that occur in the different data sets and situations are characterized and analyzed (section 4.4). Finally, the results are summarized and discussed in section 4.5 together with a perspective on how to use this knowledge for protein inference and quantification.

4.1 Data sets, protein databases and preprocessing

To characterize the bipartite peptide-protein graphs, four proteomics test data sets stemming from samples of different organisms were chosen. These data sets contain proteins with known protein ratios between samples and are therefore suitable to assess the quality of the protein quantification method by comparing the expected protein ratios with the estimated proteins ratios (results shown in chapter 5). Graphs were constructed from the peptide-level quantitative data and from theoretical peptides acquired by an *in silico* digestion of the corresponding protein databases. In the following, details on the composition of the data sets, the corresponding protein sequence databases used for *in silico* digestion and the peptide identification using database searches are described. Details on the expected ratios and quantification information are given later in section 5.3 (p. 83).

For data set D1 (Barkovits et al., 2020; Uszkoreit et al., 2022) 13 non-mouse proteins were spiked into a lysate of the mouse cell line C2C12 in different concentrations. The corresponding protein sequence database (FASTA-file) consisted of in total 55,691 protein entries, 55,286 proteins from the UniProt reference proteome of mus musculus (UP000000589, version v202205, only canonical sequences), the 13 spike-in proteins, the contaminant database provided by Andromeda (245 entries, Cox et al., 2011), as well as 147 spike-in contaminants (Barkovits et al., 2020).

In data set D2 (Ramus et al., 2016a; Ramus et al., 2016b), yeast was used as the background organism. The UPS1 (Universal Proteomics Standard Set, Merck KGaA, 2023) was spiked into yeast lysate, which contains 48 human proteins in equimolar amounts. The fasta file consists of 6,060 proteins from the UniProt reference proteome for saccharomyces cerevisiae (UP000002311, version v202205, only canonical sequences), the 48 UPS proteins (Merck KGaA, 2023) and the 245 Andromeda contaminants (Cox et al., 2011).

In data set D3 (Cox et al., 2014), proteins extracted from HeLa cells (human) were mixed with proteins from the bacterium *E. coli* in a defined ratio. The protein sequence database consists of 81,837 proteins of the UniProt homo sapiens reference proteome (UP000005640, version v202205, only canonical sequences), 4,448 proteins of the UniProt escherichia coli reference proteome (UP000000625, version v202205, only canonical sequences) and the 245 contaminant sequences from Andromeda , (Cox et al., 2011), summing up to in total 86,530 protein entries.

For data set D4 (Saltzman et al., 2018), proteins from human HeLa as well as mouse NIH-3T3 cells were mixed in different proportions. In total 137,368 proteins were in the database, 81,837 from the UniProt homo sapiens reference proteome (UP000005640, version v202205, only canonical sequences), 55,286 from the UniProt reference proteome of mus musculus (UP000000589, version v202205, only canonical sequences) and 245 Andromeda contaminant sequences (Cox et al., 2011).

An overview over the different data sets can be found in table B.1 in the appendix (p. 171). All four data sets were measured in data-dependent acquisition mode (DDA). The data sets contain between two and nine experimental conditions, which differ in the concentrations of spike-in proteins or mixture ratio of the different organisms. Each of those conditions was measured in three replicates. The raw files were downloaded using the respective ProteomeXchange identifiers (PXD numbers) from the PRIDE or MassIVE databases (Perez-Riverol et al., 2022; Deutsch et al., 2023). The peptides were identified and quantified by using the MaxQuant software version 2.3.0.0 (Cox and Mann, 2008; Cox et al., 2014), which internally uses the search engine Andromeda (Cox et al., 2011). The enzyme "Trypsin" was used (cleavage after R and K, but not before P) and up to two missed cleavages were allowed. The precursor mass tolerance was set to 20ppm for the first search and 4.5 ppm for the main search, while fragment mass tolerance was set to 0.5 Da. The FDR cutoff was set to 1%. Label-free quantification (LFQ) was enabled and decoys were generated by randomly shuffling the original protein sequence. A maximum of three modifications per peptide were allowed. Carbamidomethylation of cysteine was set as a fixed modification and oxidation of methionine and acetylation of the protein N-term as variable modifications.

For calculating theoretical graphs from the corresponding protein sequence databases ("database graphs", abbreviated with D*_fasta), the protein sequences were *in silico* digested, meaning that the tryptic digestion is mimicked by programming code and deriving which peptide sequences could occur in theory. The cleavage rules for trypsin (after K and R, if not followed by P) were used. The maximal peptide length was set to 50; for the minimum length values between five and nine were investigated to find a suitable value (see section 4.4.1, p. 45). The maximal number of allowed missed cleavages was set to integer values between zero and four, again a suitable value was searched for (see section 4.4.2, p. 51).

For calculating the graphs from the quantitative peptide-level data ("quantitative graphs", abbreviated with D*_quant), the resulting `peptides.txt` tables from the

MaxQuant output were processed as follows. Peptide sequences belonging to decoy proteins were removed. Intensities of zero were considered as a missing value. The same thresholds from the database graphs for the maximal number of missed cleavages and the minimal peptide length were also applied to the quantitative graphs to ensure comparability. The raw peptide intensities (stemming from a XIC-based peptide quantification) were normalized to remove potential technical bias. As the actual intensities are not relevant in this chapter, more details about the normalization are given later in section 5.3 (p. 83). The technical replicates of the same experimental condition were averaged for each data set. If a peptide had two or more missing values out of three replicates, the corresponding averaged peptide intensity was set to `NA`. For each possible pairwise comparison of experimental conditions, peptide ratios from the averaged peptide intensities were calculated. Only peptides with a valid ratio ($\neq$ `NA`) were considered for constructing the graphs.

4.2 Construction of bipartite peptide-protein graphs

The relationship between peptides and the corresponding proteins from a bottom-up proteomics experiment can be represented by a bipartite graph (Schork et al., 2022). A bipartite graph G has a node set $N(G)$ and an edge set $E(G)$. The nodes are split into two different sets, $N_1(G)$ and $N_2(G)$ so that no edges exist within a node set, but only from one node set to the other (Rahman, 2017, pp. 17-18). In the field of proteomics, $N_1(G)$ represents the protein nodes and $N_2(G)$ the peptide nodes. An edge exists between a protein and a peptide node if the peptide could stem from the protein. The large bipartite graph calculated from a whole data set or database consists of so-called connected components which are not connected with each other. Figure 2.6 (p. 14) shows two example of such connected components. They will be simply called "graphs" in the following for simplicity.

The details of the computational construction deviates from the description in Schork et al., 2022, because in the meantime the procedure was optimized, especially regarding the runtime. For more details see section 4.3. The starting point for the construction of the bipartite graphs is a table with peptide sequences and proteins that they may have originated from. This table is generated in case of database graphs during the *in silico* digestion. For the quantitative graphs, the quantified peptides were mapped to the protein sequences in the respective FASTA

file. The table is converted into an edgelist, which contains one line per edge in the graph with the peptide in the first and the protein in the second column. Each protein accession and each peptide sequence in the edgelist is considered as a node.

To be able to better compare the graphs and to visualize them, the graphs are simplified. For this, peptide and protein nodes are collapsed if possible (Schork et al., 2022). Protein nodes that have the exactly same set of peptides cannot be distinguished (protein ambiguity), also not during protein inference or quantification. Therefore these protein nodes are collapsed into a single node, containing both protein accessions, i.e., they form a protein group (see first step in figure 4.1). Note that by this procedure, formerly shared peptides can become unique for the respective collapsed protein node as indicated by the change of colouring (e.g., peptide 1 and 2 in figure 4.1). For further simplification and better visualization, peptide nodes belonging to the exact same set of peptide nodes were collapsed (see second step in figure 4.1). For details on the node collapsing algorithm, see section 4.3. In this thesis, uniqueness of peptide nodes is defined based on these collapsed graphs, i.e., a peptide node is unique if it belongs to only one protein node, which may consist of multiple protein accession numbers. This also means that there may be more protein ambiguity than visible from the graph structures, as protein nodes may contain multiple non-distinguishable protein accessions.

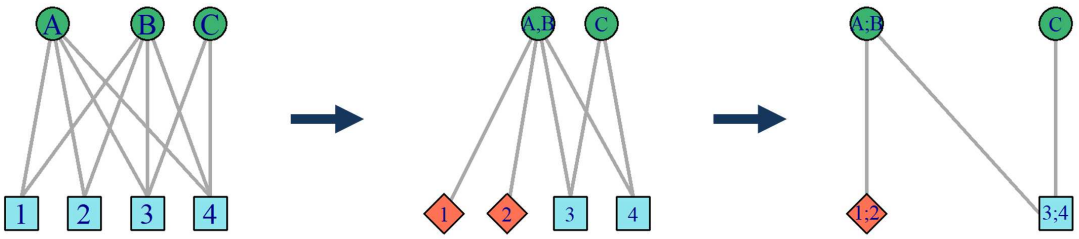


Figure 4.1: Process of collapsing protein and peptide nodes in the bipartite peptide-protein graph. Formerly shared peptides may become unique for the specific collapsed protein node, which is the definition of uniqueness used within this thesis.

After collapsing protein and peptide nodes, from the reduced edgelist, a bipartite graph can directly be constructed using the `graph_from_edgelist()` function of the `igraph` R package (Csárdi and Nepusz, 2006). After the whole bipartite graph is constructed, connected components are generated by using the `igraph` function `decompose()`. A connected component is a subgraphs for which all nodes are connected by paths inside itself, but no node is connected to any node outside of the subgraph. These components contain then a set of protein nodes that are connected

via chains of shared peptides. These connected components can be analyzed separately regarding the protein quantification, as all other proteins in the graph do not influence the peptide intensities or ratios (Schork et al., 2022). The connected components will be called "bipartite peptide-protein graphs" or simply "graphs" in this thesis.

As explained above, the bipartite graphs are constructed on two different levels for each data set (Schork et al., 2022). First, the theoretical graphs from the *in silico* digestion of the respective protein sequence database are calculated ("database graphs", abbreviated with D*_fasta). On the other hand, the quantitative peptide-level data is used ("quantitative graphs", abbreviated with D*_quant). As explained above, only peptides with a valid ratio for a specific pairwise comparison of experimental groups are considered. As some peptides may not be consistently identified or quantified in all samples or groups, different sets of peptides are used for each pairwise comparison. Therefore, the graph generation is done separately for each comparison (see table B.1 in the appendix, p. 171, for the number of comparisons for each data set).

To compare the graph structure between data sets and between the quantitative and database graphs, isomorphism classes are built. Two graphs G and H are isomorphic, if the structure of the graphs is the same. Formally, if there is a bijective function $f : N(G) \rightarrow N(H)$ that maps the node set of a graph G to the one of a graph H so that

$$(u, v) \in E(G) \Leftrightarrow (f(u), f(v)) \in E(H) \; \forall u, v \in N(G), \tag{4.1}$$

the two graphs are called isomorphic (Rahman, 2017, pp. 22-23). For the use case of bipartite graphs, like the peptide-protein graphs, it is also required that the function f only maps protein nodes with protein nodes and peptide nodes with peptide nodes (Schork et al., 2022). Otherwise, for example, an M-shaped graph with two protein nodes and three peptide nodes may be seen as isomorphic to a W-shaped graph with three protein nodes and two peptide nodes. Those graph types are however extremely different regarding their behaviour for protein quantification. While the M-shaped graph has one unique peptide per protein nodes, the W-shaped graph has no unique peptide at all. Therefore, an adapted version of the `isomorphic()`-function of the `igraph` package was implemented, that first transforms the edges into directed edges (from protein edges to peptide edges) before testing isomorphy with

the VF2 algorithm (Cordella et al., 2004). Isomorphism classes are sets of isomorphic graphs. In an iterative procedure, for a specific graph all isomorphic graphs in the list of remaining graphs are searched. These together build a new isomorphism class and are deleted from the list of remaining graphs. This procedure is repeated until no graph is left in the list. Representative graphs of those isomorphism classes can then be visualized (for an example see figure 4.3, p. 55).

4.3 Software implementation and technical challenges

The code used for this thesis was implemented using R version 4.3.1 (R Core Team, 2023) and RStudio version 2023.09.1 (Posit team, 2023). The development of the R package `bppg` (Bipartite Peptide-Protein Graphs) was started, that contains functionality for generating, visualizing and characterizing the bipartite peptide-protein graphs, see `https://gitlab.ruhr-uni-bochum.de/mpc-bioinformatics/bppg`. For handling the graphs the R package `igraph` (Csárdi and Nepusz, 2006) was used. A complete list of R packages used for implementations for the analyses for chapters 4 and 5 can be found in table B.9 in the appendix (p. 176). The programming code for all analyses and visualizations in this thesis can be found on Zenodo under the following link: `https://zenodo.org/records/11120878`.

In a former version of the code (used for the paper Schork et al., 2022), for the graph generation, first the biadjacency matrix was constructed from the list of peptides and corresponding proteins. A biadjacency matrix has rows reflecting one node type and columns reflecting the other one. It is a special case of a standard adjacency matrix for bipartite graphs, which leaves out the submatrices full of zeros that are not necessary because there cannot be any edges between nodes of the same type. For the peptide-protein graphs this matrix is large but contains mostly zeros, this is why the class `dgCMatrix` for sparse matrices from the `matrix` package (Bates et al., 2024) was used to save RAM. For the collapsing of the protein and peptide nodes, simply duplicated rows and columns of the biadjacency matrix had to be removed. However this approach did not keep information on which nodes were merged, which is not important for analyzing the graphs structures but is crucial for interpreting the quantification results later in chapter 5. The execution of this step was time consuming and an existing implementation that kept the collapsed node information was inefficient.

In his Bachelor's thesis Arne Junge investigated how to improve the efficiency of graph generation for database graphs (Junge, 2022). The best found solution was to first generate edgelists that are easily obtained from the *in silico* digestion results. Edgelists are two-column dataframes that contain the two nodes connected by an edge in each row. The edgelist could be directly by using the function `graph_from_edgelist()` from the `igraph` package (Csárdi and Nepusz, 2006) to generate the bipartite graph object. By using this alternative way of graph generation, the runtime could be extremely decreased from 2,959 seconds to 21 seconds for the human reference database, an over 99% decrease of runtime. In their study project Katharina Neuhaus, Sofian Faiz and Nils Achterfeldt adapted this procedure also for generating the quantitative graphs, with a special focus on the efficient collapsing of peptide and protein nodes while keeping the node information (Neuhaus et al., 2023). Before, the collapsing was directly applied by removing duplicated rows and columns from the biadjacency matrix. They developed an algorithm to collapse nodes by an efficient use of the `aggregate()`-function applied directly to the edgelist.

4.4 Characterization of occurring bipartite peptide-protein graphs

In the following, bipartite peptide-protein graphs generated from an *in silico* tryptic digestion of protein sequence databases ("database graphs") as well as from quantitative peptide data ("quantitative graphs") are characterized.

4.4.1 General overview and impact of minimal peptide length

Quantitative and database bipartite peptide-protein graphs are constructed using the respective peptide intensities as well as the corresponding protein sequence databases. For the database graphs, appropriate values for the maximal number of missed cleavages and the minimal and maximal number of amino acids per peptide have to be defined. The aim is to make the *in silico* digestion as comparable as possible to the quantified peptides. The peptide size is an important factor because too small or too large peptides cannot be measured with mass spectrometry (Swaney et al., 2010). For the minimum, six or seven amino acids are common default set-

tings in search engines, e.g., X!Tandem (Bjornson et al., 2008) or Andromeda (Cox et al., 2011). Nine is the required length for reporting identified peptides by the Human Proteome Project of the Human Proteome Organization (HUPO-HPP, Deutsch et al., 2019). The maximal peptide length may be defined by the number of amino acids or by weight (e.g., Andromeda has a default limit of 4600 Da, Cox et al., 2011).

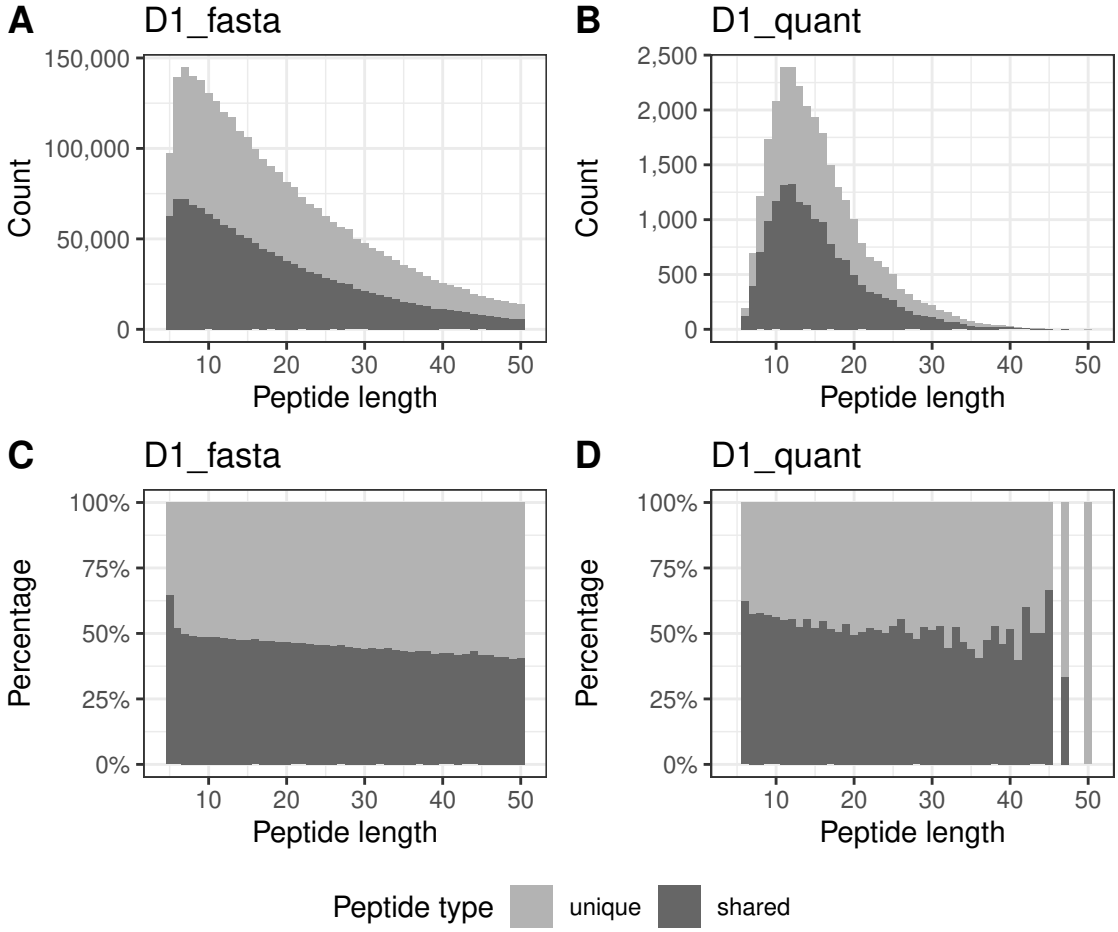


Figure 4.2: Count and percentages of shared and unique peptide sequences depending on the peptide length. The peptide length is given in amino acids. As an example, here the values for data set D1 are shown. Uniqueness is here defined as belonging to only one protein node, which may consist of multiple protein accessions.

For the analysis of the four data sets, Andromeda settings with two missed cleavages, minimum six amino acids per peptide and a maximum of 4600 Da was used. For the *in silico* digestion, first a minimum of five and a maximum of 50 amino acids was used. The distribution of peptide length, separated for unique and shared peptides is shown in figure 4.2, exemplary for data set D1 (the plots for data sets D2-D4 can be found in the appendix as figures A.1, A.2 and A.3, pp. 147 - 149). It can be seen

that small peptides with up to six amino acids are seldomly quantified, even tough they are frequent in the database graphs (figure 4.2A and B). The peptides of length five show a larger fraction of shared peptides compared to any longer peptides (figure 4.2C). Peptides longer than 40 amino acids become less common in the database graphs and are almost never quantified. Due to their low number, they are not expected to have a great effect on the bipartite graphs, this is why this influence is not investigated further. An upper limit of 50 was used, and also the quantitative data was filtered according to this threshold.

The minimal peptide length, however, may have a large influence on the database graphs, as there is a high number of small peptides compared to the quantitative graphs (compare figure 4.2A to B). Also, there is an increased degree of sharedness for small peptide sizes (see figure 4.2C). In the following, the influence of the minimal peptide length on the theoretical graphs is investigated, with values from five (extreme lower limit) to six and seven (common default values for search engines) to nine (HUPO-HPP recommendation, Deutsch et al., 2019).

Table 4.1: Influence of different minimal peptide lengths on the database bipartite graphs for D1_fasta. * In terms of number of protein nodes.

	min 5 AA	min 6 AA	min 7 AA	min 8 AA	min 9 AA
Protein accessions	55,620	55,610	55,599	55,563	55,532
Protein nodes	55,265	55,247	55,221	55,163	55,097
Peptide sequences	2,970,833	2,873,662	2,734,624	2,589,855	2,449,783
Peptide nodes	141,143	110,356	102,657	100,238	98,583
Unique peptide nodes	49,285	49,203	49,046	48,851	48,646
Shared peptide nodes	91,858	61,153	53,611	51,387	49,937
Edges	481,244	301,669	262,369	249,290	240,420
Graphs	2,734	10,140	15,374	16,605	17,133
Graphs with 1 protein node	1,860	4,430	5,865	6,392	6,617
Isomorphism classes	193	1,502	2,730	2,828	2,866
Largest graph*					
Protein nodes	50,671	27,550	1,793	1,010	1,004
Protein nodes percentage	91.69%	49.87%	3.25%	1.83%	1.82%
Peptide nodes	134,719	63,594	4,352	2,646	2,547
Second largest graph*					
Protein nodes	25	91	1,183	752	295
Peptide nodes	73	177	3,105	1,528	903

Table 4.1 shows numbers for the database graphs generated for minimal peptide length between five and nine amino acids (AA) for D1_fasta. For a minimum of five amino acids, 55,620 protein accessions have at least one theoretical peptide between

five and 50 amino acids, which is almost all proteins in the fasta file. From these, 55,265 protein nodes are formed, meaning that only seldomly protein nodes are collapsed. This can be easily explained, as an *in silico* digestion of a whole fasta file was used and each protein sequence should be unique, meaning that the probability of producing at least one unique peptide is extremely high. In contrast, the almost three million peptide sequences are collapsed to only 140,000 peptide nodes, which highly reduces the complexity of the bipartite graphs. From these around $\frac{2}{3}$ are shared peptide nodes. In total 2,734 graphs are formed, 1,860 (68.0 %) of them are the most simple case with only one protein node and one unique peptide node. This case is trivial for protein inference, as the only protein has unique evidence and the protein quantification can be done, e.g., by the mean of the peptide intensities or ratios, as no shared peptides are involved in these graphs. The graphs can be grouped into 193 isomorphism classes. The largest graph is extremely large and contains over 50,000 protein nodes and 134,000 peptide nodes. It is responsible for 91.69 % of all protein nodes, 95.45% of all peptide nodes and 97.86% of all edges. In contrast, the second largest graph is comparably small with only 25 protein nodes and 73 peptide nodes. The huge size of the largest graph can be explained by the high amount of small shared peptides that connect protein nodes with each other (compare with the distribution of peptide lengths in figures 4.2A and C, p. 46).

When the minimal number of amino acids per peptide is increased to six, the number of covered protein accessions or nodes only decreases slightly. Around 100,000 peptides with the size of five amino acids are removed in this step, that correspond to around 30,000 peptide nodes. Most of the missing peptide nodes are shared. This leads to a splitting of the largest graph to only about half its size with 27,550 protein nodes (49.87% of all protein nodes). In total the number of graphs is almost four times as high, with over 10,000 graphs that can be grouped into 1,500 isomorphism classes. All of this together suggests that the largest graph at five amino acids length is broken into pieces that were previously only connected via shared peptides of length five.

When increasing the minimal peptide length to seven amino acids this trend is continued. The largest graph splits further into smaller pieces. The largest graph contains only 1,800 protein nodes (3.25% of all nodes), while the number of graphs almost doubled. The largest and second largest graph have also a similar size with 1,800 and 1,200 protein nodes. When the minimal peptide length is increased further to eight or nine amino acids, the trend continues but is slowed down. The largest

graph has around 1,000 protein nodes and does not shrink much when removing peptides of length eight. The number of graphs rises at a slower pace and the number of isomorphism classes increases only slightly.

The question is which limit for the minimum peptide length should be chosen. The aim is to have a fair comparison between these database graphs and quantitative graphs constructed from the quantitative peptide-level data. A low limit of five or six leads to a huge largest graph that is difficult to handle. On the other hand, a high limit nine leads to the loss of more than 500,000 peptide sequences and over 40,000 peptide nodes in total. For the further analysis, the limit of seven amino acids was chosen, as it is the lowest value where the largest and second largest graph are of comparable size and the largest graph is small enough to handle (below 5% of all protein nodes). With this limit, the graphs become comparable to the quantitative graphs, where peptides with length five were not searched and with length six were extremely seldomly quantified (see figure 4.2, p. 46), without the risk of removing too many peptides.

For data set D2_fasta the situation is similar but not the same (see table B.2 in the appendix, p. 172). Overall, the number of peptides and proteins is much lower than for D1, because the data set is based on yeast samples that are a lot less complex than the mouse samples. For a limit of five amino acids the largest graph again has a large size, with 4,504 of in total 6,275 protein nodes. 71.78 % of protein nodes, 85.10% of peptide nodes and 91.75% of edges are covered by this single graph, which is extremely high but lower than for D1. At a length of six amino acids, the largest graph is split into many smaller subgraphs and the largest and second largest graphs are already of comparable size with 116 and 71 protein nodes, respectively. The largest graphs makes up for only 1.85% of all protein nodes. When increasing the minimal peptide length further, the largest graph gets smaller for seven amino acids but then stays at a similar size; the number of protein nodes even stays the same. On the other hand, the number of isomorphism classes has a peak at 66 at a minimum of six amino acids. For this data set a minimal peptide length of six was chosen for the following analysis because the largest graph has already fallen apart to a sufficient degree and not too many different graph types are lost.

The data set D3_fasta shows a similar behaviour as D1 (see table B.3 in the appendix, p. 172). However, the number of proteins and peptides is larger than for D1_fasta. A small enough largest graph is however only reached for a minimal peptide length of eight. Then, the largest graph contains 3.11% of all protein nodes but is still over

five times as large as the second largest graph. This issue is not solved by increasing the minimal peptide length further to nine amino acids. Therefore, the decision is to use a minimal peptide length of eight amino acids for this data set, to ensure that the largest graph can be handled and that not too much information gets lost.

Data set D4_fasta has the largest fasta file with the reference proteome of mouse and human combined. As a result, the absolute values of accessions, protein and peptide nodes, as well as edges is the highest among the four data sets (see table B.4 in the appendix, p. 173). The human and mouse proteome share a lot of peptides. With a minimal amino acid length of five, the number of shared peptide nodes is twice as high as the unique peptide nodes. This ratio becomes smaller for rising minimal peptide lengths. For a minimum of nine amino acids, the number of shared peptide nodes is still considerably higher than those of unique peptide nodes. Like for the other data sets, the largest graph contains the majority of proteins nodes for five amino acids (95.2%, the highest value among all data sets). Like for D3_fasta, a threshold of eight amino acids seems to be reasonable, as the largest graphs then contains only 3.35% of all protein nodes. The number of over 21,000 graphs is comparable to D3, however, the over 7,800 isomorphism classes make this the most complex of all four data sets.

In summary, the minimal peptide length has a huge impact on the database bipartite peptide-protein graphs. If the minimal peptide length is too small, the largest graph incorporates the majority of protein nodes, which is not manageable for comparison with the quantitative graphs. On the other hand, an increase of the minimal peptide length leads to an information loss because many small peptides would be removed. Therefore finding a good balance is important. It was recognized that the optimal minimal peptide length depends on the data set, in particular the total number of protein nodes and the complexity. While for the less complex yeast data set D2, a threshold of six amino acids seems to be reasonable, for the mouse data set D1 a limit of seven is the better choice. D3 and D4 are even more complex as mixtures of human with *E. coli* or mouse, respectively, and would require an even larger limit of eight amino acids. In all cases, the chosen minimal peptide length leads to a largest graph containing less than 5% of all protein nodes. In the following sections, the above mentioned individual minimal peptide lengths for the different data sets were used for the *in silico* digestion of the protein sequence databases. For comparability, also the quantitative peptide-level data was filtered accordingly before constructing the quantitative graphs.

4.4.2 Influence of missed cleavages

During tryptic digestion of proteins in the laboratory, potential cleavage sites may be missed. The quality of the trypsin (Burkhart et al., 2012), neighbouring amino acids (Lawless and Hubbard, 2012) or nearby post-translational modifications like phosphorylation (Gershon, 2014) have been identified as potential factors for missed cleavages. Peptides with up to two missed cleavages are usually also quantified, as this is a common default value for the respective search engine settings. Therefore, also the maximal number of missed cleavages have to be taken into account as a parameter for the *in silico* digestion of the protein sequence database, as they may have an influence on the generated graphs. For building the database graphs from the FASTA file, zero to four allowed missed cleavages were tested, using the determined minimal peptide length from section 4.4.1 (p. 45) for each data set.

Table 4.2: Influence of different maximal allowed missed cleavages on the database bipartite graphs for D1_fasta. * In terms of number of protein nodes.

	MC 0	MC 1	MC 2	MC 3	MC 4
Protein accessions	55,487	55,579	55,599	55,602	55,602
Protein nodes	54,359	55,122	55,221	55,239	55,242
Peptide sequences	604,082	1,627,185	2,734,624	3,746,279	4,589,424
Peptide nodes	86,646	98,934	102,657	104,081	104,827
Unique peptide nodes	42,501	47,770	49,046	49,398	49,558
Shared peptide nodes	44,145	51,164	53,611	54,683	55,269
Edges	203,154	246,840	262,369	269,008	272,486
Graphs	16,907	15,795	15,374	15,164	15,064
Graphs with 1 protein node	6,670	6,023	5,865	5,809	5,784
Isomorphism classes	3,021	2,862	2,730	2,652	2,622
Largest graph*					
Protein nodes	934	1,085	1,793	5,800	6,273
Peptide nodes	1,751	2,684	4,352	14,265	15,487
Second largest graph*					
Protein nodes	519	887	1,183	139	139
Peptide nodes	923	1,761	3,105	469	522

A summary of the graph characteristics for data set D1_fasta are shown in table 4.2. When more missed cleavages are allowed, the number of peptide sequences rises dramatically (from 600,000 at zero missed cleavages to over 4.5 million at four missed cleavages). At the same time, the number of peptide nodes only rises from 86,000 to 104,000. This is because the larger miscleaved peptide sequences will often be found in the same node as at least one of their smaller subsequences. If at least one of the

subsequences is unique, the larger miscleaved peptide will automatically be unique too. If both subsequences are shared, the longer peptide may be shared or become unique, if only one protein contains the two subsequences in this specific order. The number of shared as well as unique peptide nodes rises a bit when the number of allowed missed cleavages is increased. The number of graphs decreases a bit from around 17,000 to around 15,000. The size of the largest graph goes up continuously, while the size of the second largest graph goes up until two missed cleavages with up to 1,183 protein nodes, with a large drop to only 139 protein nodes for up to three or four missed cleavages. In this case the largest graph makes up over 20% of the total protein nodes. Some protein accessions only gain large enough peptides by introducing missed cleavages. For example the protein P62947 with the amino acid sequence "MRAKWRKKRMRRLKRKRRKMRQRSK" contains so many lysins (K) and arginines (R) that only with at least three missed cleavages tryptic peptides with at least seven amino acids occur. Another phenomenon is that protein nodes with multiple protein accessions may break up into different nodes when newly unique peptides are generated by additional missed cleavages. A maximum peptide length of 50 is used in all cases. Missed cleavages may cause some peptides to exceed this limit. On the other hand, the sequence of a peptide with missed cleavages consists of two or more sub-sequences. These sub-sequences may be of small size (below the minimum peptide length, even a single amino acid is possible), so not necessarily counted as a peptide. In summary, allowing three or four missed cleavages may lead to large largest graphs, while limiting the peptides to zero or one missed cleavage greatly reduces the amount of considered peptides. For D1, therefore it was decided to keep a limit of two missed cleavages as the threshold.

For data set D2_fasta (see table B.5, p. 173) the situation is similar, however, due to the smaller fasta file, the effects are not as strong. For example, the largest graph stays comparably small also for up to four missed cleavages (3.8% of all protein nodes). The gain in protein and peptides nodes compared to two missed cleavages is extremely small. Also, the number of graphs does not change much. Especially the number of unique peptide nodes is only raised by seven when going from two to four missed cleavages, so there is not much gain when allowing higher numbers of missed cleavages. For data sets D3_fasta and D4_fasta (see tables B.6 and B.7 in the appendix, p. 174) the situation is also similar. For three or four missed cleavages the proportion of the largest graph on all protein nodes rises slightly over 5%.

In general, the phenomenon of missed cleavages that arise during tryptic digestion in the laboratory needs to be addressed by the data analysis. For the bipartite peptide-protein graphs allowing many missed cleavages per peptide would potentially lead to more unique peptides, which could in principle facilitate the protein inference and quantification. However, for all four data sets, the increase in unique peptide nodes almost stagnates when more than two missed cleavages are allowed. It was decided to use the common threshold of up to two missed cleavages for the *in silico* digestion, also the peptide identification was done using this setting.

4.4.3 Occurring types of graphs

In this section the occurring graph types arising from the quantitative peptide-level data ("quantitative graphs") and the theoretical *in silico* digestion ("database graph") are compared (Schork et al., 2022). For data set D1, all theoretical peptides between seven and 50 amino acids are considered, for D2 between six and 50 and for D3 and D4 between eight and 50, according to the findings of section 4.4.1 (p. 45). To have a fair comparison, peptides outside this range are also omitted from the quantitative proteomics data. The number of removed peptides is low, as smaller or large peptides are seldomly quantified (see figures 4.2, p. 46 and A.1 - A.3, pp. 147 - 149). According to the analysis in section 4.4.2 (p. 51), a maximal number of two missed cleavages were allowed in situations, during *in silico* digestion and the database search for the quantitative data. As the graph type is expected to strongly influence the protein inference and quantification, having a look at the most common graph types is important. A comparison between the database and the quantitative graphs can give a hint if the characteristics of the theoretical graphs are transferable to graphs generated from the real-world quantitative data sets. The quantitative graphs can at most be as complete as the database graphs, but are expected to be smaller due to missing, non-quantified peptides.

D1_fasta has 15,374 graphs that can be separated into 2,730 isomorphism classes, see table 4.3 (p. 54). In figure 4.3(a) (p. 55) representative graphs of the ten largest isomorphism classes are plotted with information on their absolute and relative frequency. For the quantitative data set, 13 spike-in proteins were added to a mouse cell lysate in five different concentrations. When calculating the peptide ratios between the possible pairs of concentrations, ten different comparisons are

Table 4.3: Number of graphs and graph types. Note that the number of quantitative graphs cannot be directly compared to the number of database graphs or between data sets. The reason for this is that the numbers here are the sum of the number of graphs over all pairwise comparisons of groups.

Type	Variable	D1	D2	D3	D4
database	number of graphs	15,374	4,900	21,660	21,412
database	number of graph types	2,730	66	4,633	7,898
quantitative	number of graphs	26,902	22,952	4,097	86,592
quantitative	∅ nr. of graphs per comparison	2,690	638	4,097	4,123
quantitative	number of graph types	416	19	348	7,322

obtained. All graphs over all comparisons put together lead to 26,902 graphs that can be grouped into 416 isomorphism classes.

The most simple graph type with only one protein and one peptide node is the most common in all cases and data sets. For D1_fasta it makes up about 38% of all graphs. This graph is extremely easy and trivial to handle during protein inference and quantification. Although it is the most common graph, it is small and makes up only 10.6% of all protein nodes. The proportion is much higher for D1_quant, almost 60% of all graphs and 31.2% of all protein nodes often the quantitative graphs. The second most common graph type is the M-shaped type that has two protein nodes, both having a unique peptide node each and one shared peptide node. Almost 20 % of all graphs are of this type and it makes up 11.1% of the protein nodes. This graph type is also quite easy to handle due to the presence of the unique peptides, with the additional information from the shared peptide. However, the ratio of the shared peptide may not fit to those of the unique peptides and therefore may cause a contradiction, as will be explained in section 5.1.1 (p. 72). This graph type is extremely common in the database graphs, but makes up only 0.45% of all graphs for the quantitative graphs. In the top 10 isomorphism classes of the database graphs there are four (on the ranks three, four, six and nine) that have three protein nodes, each connected to a unique peptide node. They differ in their number of and their connection to shared peptide nodes. More or less they will behave similarly to the M-shaped graph during protein quantification. The same holds for the seventh and tenth most common class that contain four protein nodes and four unique peptide nodes. Strikingly, in the quantitative graphs those graphs are uncommon. Except the most simple graph type and the M-shaped graph (with only 0.45%), the ten largest isomorphism classes for the quantitative graphs do not contain any graph where each protein node has a unique peptide node.

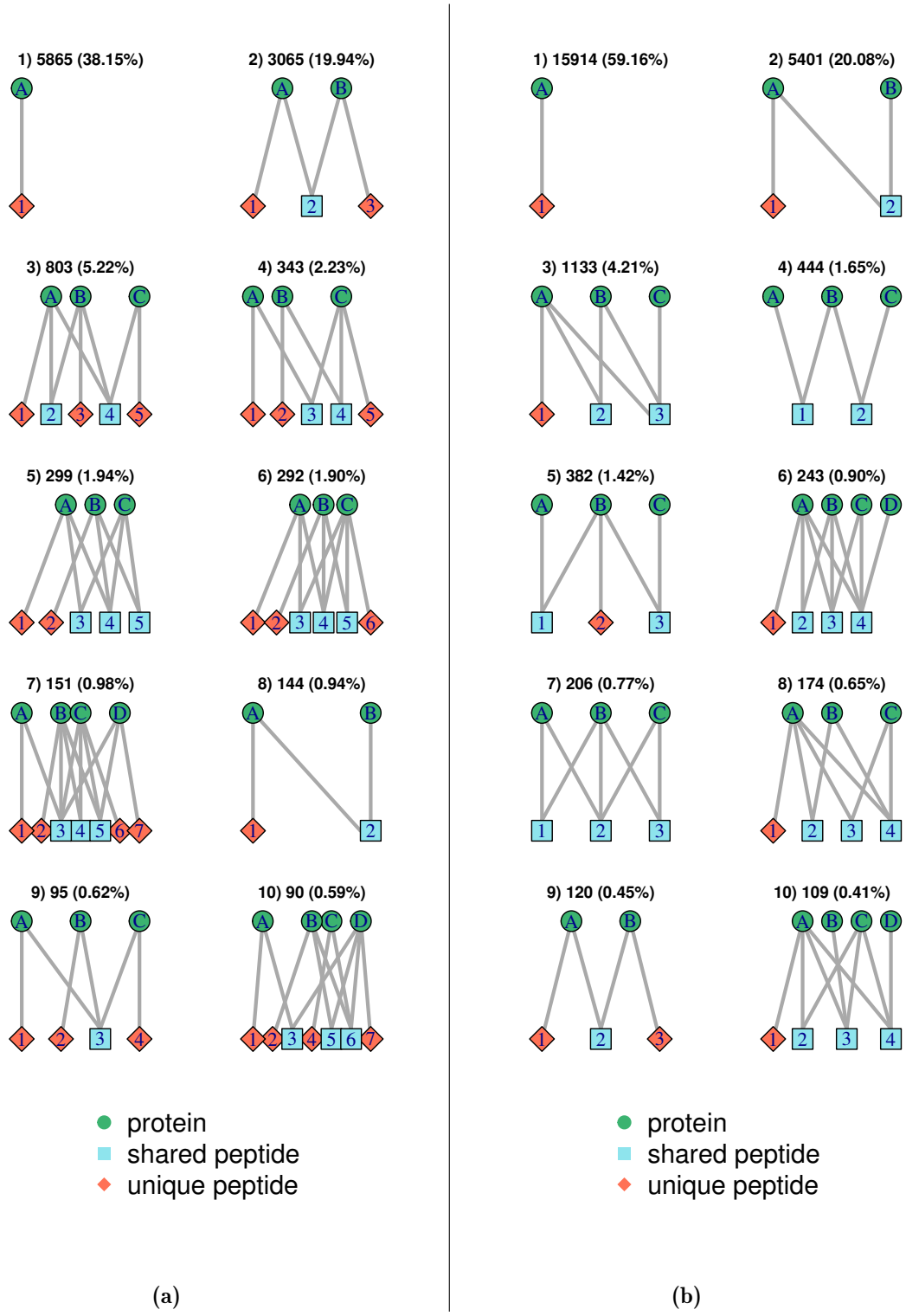


Figure 4.3: Representative bipartite graphs of the ten largest isomorphism classes found in data set D1. (a) D1_fasta, (b) D1_quant, with rank, number of occurrences and percentage of all graphs.

For the fifth and eighth most common graph types for D1_fasta, not all proteins have a unique peptide node. That means there are protein nodes with connections only to shared peptides. These cases are much more difficult to handle for protein quantification, which will be explained examplarily for the N-shaped graphs in section 5.1.1 (p. 72). With many common protein quantification methods these proteins would not be quantified. Quantitative graphs that contain protein nodes without unique peptides are much more common than database graphs. For example, the N-shaped graph makes up more than 20% of the quantitative graphs here while it was below 1% of the database graphs. In the top 10 quantitative graphs for D1_quant there are even two graph types (fourth and seventh rank) that contain no unique peptide node at all. These are extremely hard to handle for protein quantification and would most likely not be considered in common protein quantification strategies. From the remaining 2,720 database graph types, outside of the top 10, 2,460 only occur once in the whole D1_fasta. Many of these graphs are quite large and are unique due to their complex combination of peptide and protein nodes and the edges. These graphs contain in total 29,029 proteins nodes, that means 52.6% of all protein nodes in total. For D1_quant 98 database graph types occur only once, which contain only 1.5% of all protein nodes.

For data set D2 (see figure A.4 in the appendix, p. 150) the *in silico* digestion leads to 4,900 database graphs within 66 isomorphism classes, so much less than for D1. In D2_quant 22,952 quantitative graphs are present (see table 4.3, p. 54); the high number is due to the high number of comparisons between the nine different concentration groups, namely 36 comparisons. The average number of quantitative graphs per comparison is thus also much less than for D1_quant. The most simple graph type again is the most common one, although with an extremely high proportion of 87% for the database and 89% for the quantitative graphs. For the database graphs again the M-shaped graph is the second most common with 8.55%. For the quantitative graphs this graph type is less common with 4.2%, but the decline is less pronounced than for D1. In the top 10 database graphs of D2_fasta, all protein nodes have at least one unique peptide node. For the quantitative graphs, there are seven out of the ten most common graph types with at least one protein node without a unique peptide. Two of them even do not contain any unique peptide at all. The second most common quantitative graph type here is the N-shaped graph with over 5%, that does not even exist in the database graphs in D2_fasta at all. As the most simple graph type is so prominent, the 10 most common graph types

already contain 98.7% and 99.6% of all graphs and 88.7% and 98.1% of all protein nodes for database and quantitative graphs, respectively.

The data set D3 behaves more similarly to D1 than to D2 (see figure A.5 in the appendix, p. 151). The proportion of the most simple graph is with almost 50% higher for the database graphs in D3_fasta than in D1_fasta. With 65% it is also a bit higher than the almost 60% when comparing the quantitative graphs in D3_quant and D1_quant. The top 10 isomorphism lists for the database graphs in D3_fasta and D1_fasta are very similar, although the order is a bit different. For the quantitative graphs, the top 10 list contains even more graph types (four) without any unique peptide than for D1_quant, while each except the most simple graph type has less unique peptide nodes than protein nodes, i.e., there is at least one protein node without a connection to a unique peptide node. The M-shaped graph is missing from the top ten here.

For D4 (see figure A.6 in the appendix, p. 152), human and mouse samples were mixed in different ratios and measured. Therefore it is expected that the graphs are similar to D1 and D3. However, also graphs containing both mouse and human protein nodes are present, as mouse and human share many peptides. Compared to the other data sets, the proportion of the most simple graph type is much smaller, only 33% and 37% for D4_fasta and D4_quant, respectively. Like for the other data sets, the M-shaped graph is the second most common graph type for the database graphs and the N-shaped for the quantitative graphs. In total, the top 10 lists are very similar to those of D3 for quantitative and database graphs. Due to seven different pairwise comparisons, D4_quant has by far the most quantitative graphs in total with over 80,000, but a similar number of graphs per comparison as D3_quant (see table 4.3, p. 54). However, D4_quant and D4_fasta have by far the most different graph types and therefore D4 can be seen as the most complex of the four data sets.

Figure 4.4 (p. 58) shows the distribution of numbers of the different node types for all four data sets, for database and quantitative graphs. This gives a summary of the structure of the occurring bipartite peptide-protein graphs. Protein nodes and peptide nodes are the two main node types, while the peptide nodes can be subdivided into unique and shared peptide nodes. The most simple graph type, which is extremely common in all considered situations, is the only one with only one protein and one peptide node. It is clearly visible as the highest bar in the barplot, for all cases. Higher numbers of protein or peptide nodes become less and

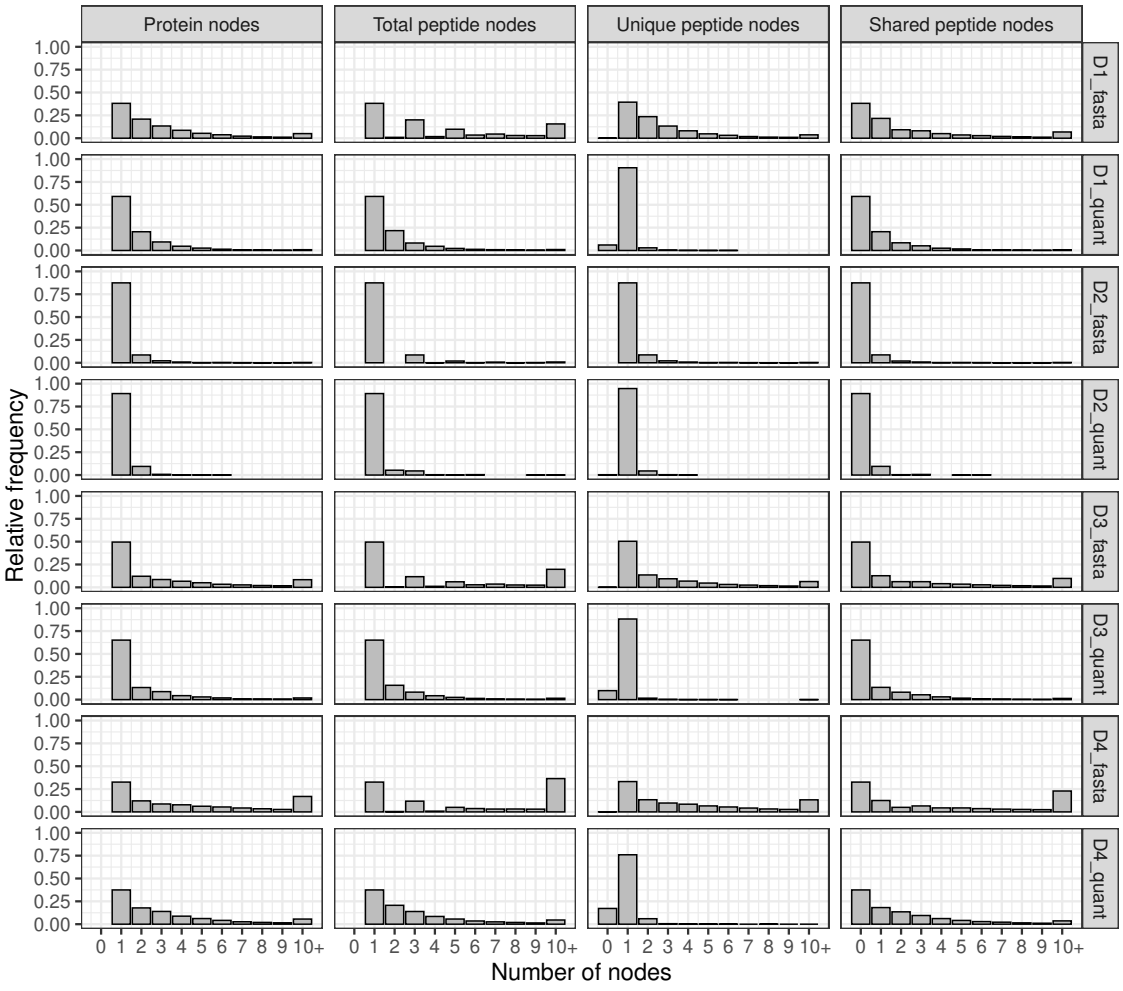


Figure 4.4: Distribution of numbers of different node types for the bipartite peptide-protein graphs (compare to figure S2 in Schork et al., 2022). The leftmost column shows the distribution of number of protein nodes for D1, D2, D3 and D4 for database graphs (*_fasta) and quantitative graphs (*_quant). On the x-axis the number of protein nodes are shown and on the y-axis the proportion of graphs with this exact number of protein nodes in comparison to all graphs. E.g., the bar at $x = 1$ is the percentage of graphs with only one protein node compared to all graphs. The sum of all bar heights adds up to one for each subfigure. Similarly, the distribution of the number of total peptide nodes, unique peptide nodes and shared peptide nodes are shown in the following columns. The rightmost bar with the label "10+" comprises the values of 10 and above.

less common. It can be seen that the distribution of the number of protein nodes looks similar among D1, D3 and D4. D2 is an exception with an extreme high number of graphs with only on protein node (already observed in the top 10 graph types before) and a rapid decline of the frequency of larger graphs. It can also be seen that there is a shift of the distribution when comparing the database to the quantitative graphs, to smaller graphs with less protein nodes. For the total number of peptide nodes, also a decline is visible. However, for the database graphs, the distribution shows gaps for even numbers of peptide nodes. E.g., graphs with two peptide nodes are way less common than those with three peptide nodes. This also corresponds to the top 10 graphs, e.g., an M-shaped graph with three peptide nodes is extreme common, while an N- or W-shape with two peptide nodes is not. Whereas for the quantitative graphs, many unique peptides were not quantified, leading to a reduction of M-shaped graphs to N-shaped graphs. Similar things may also happen to larger graphs, causing the gaps in the total peptide node distribution for the database graphs, that vanishes for the quantitative graphs.

For D4, the number of larger graphs with more than 10 peptide nodes is the highest among all data sets. Again, for D2 it is visible that most graphs are rather small. For the unique peptide nodes it can be observed that there is also a steady decline for the database graphs, except for D2 with the large fraction of the most simple graph type. D4_quant shows the highest number of quantitative graphs with zero unique peptide nodes, which are the hardest case for protein quantification. It can also be seen that graphs with many shared peptide nodes are more common in D1, D3 and D4 than in D2 and a bit less common for the quantitative graphs than for the database graphs.

Aggregated characteristics over all graphs in a data set are shown in figure 4.5 (p. 60), which may serve as a summary of the findings above. The percentage of the most simple graph (the only possible graph type with only protein node, see the light blue bar in figure 4.5A), rises when comparing the database graphs to the quantitative graphs. The difference is small for D2, as the percentage of the most simple graph type are already extremely high (87.4% on database and 89.2% on quantitative level). For D4, the difference is also smaller than for D1 or D3, possibly explained by the higher proportion of shared peptides, as can be seen in figure 4.5C. For D1, D3 and D4 a large percentage of graphs has more than two protein nodes (up to 67.4% for D4_fasta) and therefore at least one shared peptide, which effects an even higher percentage of all protein nodes.

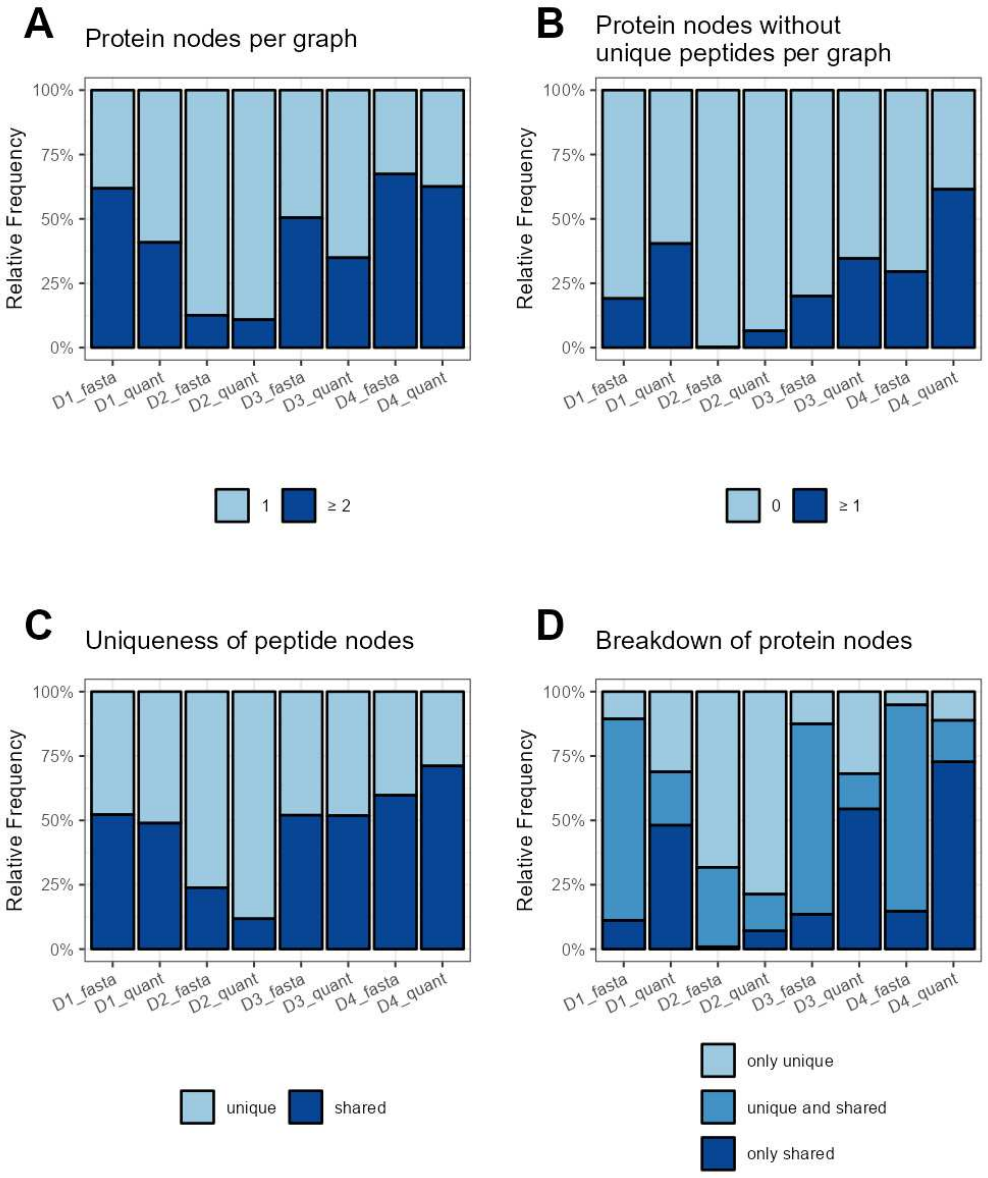


Figure 4.5: Aggregated characteristics of bipartite peptide-protein graphs over the different data sets (compare to figure 3 in Schork et al., 2022). Uniqueness of the peptide nodes is here defined as belonging to only one protein node, which may consist of multiple protein accessions. A: Percentages of graphs with one or with more than one protein node. The light blue bars reflects the percentage of the most simple graph type with only one protein node. The dark blue bars represents the percentage of all larger graphs with two or more protein nodes. B: Percentages of graphs with none vs. one or more protein nodes, that are not connected to any unique peptide node. In the graphs represented by the light blue bars, each protein node has at least one unique peptide. In the graphs represented by the dark blue bars, at least one protein node exists without a unique peptide. C: Percentages of unique vs. shared peptide nodes in relation to all peptide nodes across all bipartite peptide-protein graphs. D: Percentages of protein nodes with only unique peptides vs. unique and shared peptides vs. only shared peptides in relation to all protein nodes across all bipartite peptide-protein graphs.

For the protein quantification and the proposed novel bppgQuant method in this thesis, the protein nodes with zero unique peptide (and therefore only shared peptides) are the most interesting, as they are hard to quantify and many current methods would simple ignore them. Figure 4.5B shows the percentage of graphs with at least one protein node that has only shared peptides. This percentage is much higher for the quantitative graphs than would estimated from the protein sequence databases, probably because many theoretically existing unique peptides are not quantified. The percentage of such quantitative graphs also depends on the data set. While the percentage is quite small for D2_quant (6.5%), it reaches 40.4% and 34.6% for D1_quant and D3_quant, respectively. The highest percentage can be found in data set D4_quant, where 61.5% of all quantitative graphs contain at least one protein node with only shared peptides.

In figure 4.5C, the percentage of unique and shared peptide nodes are presented. The percentage of shared peptides is less for the quantitative graphs than for the database graphs, except for D4, where also many peptides shared between the two organisms are present. Again, this percentage depends highly on the data set, it is much lower in D2 (23.8% for D2_fasta and 11.8% for D2_quant) and extremely high in D4 (59.7% in D4_fasta and 71.1% in D4_quant). In figure 4.5D the protein nodes are separated into those with "only unique", "only shared" or "unique and shared" peptides. The percentage of protein nodes with only shared as well as only unique rises, while the number of protein nodes with both peptide node types decreases heavily, when comparing the database graphs with the quantitative graphs. The number of protein nodes without unique peptides is the highest for data set D4 (14.7% for D4_fasta and 72.8% for D4_quant) and the lowest for data set D2 with only 0.9% for D2_fasta and 7.1% for D2_quant.

4.4.4 Influence of including isoforms

So far FASTA databases and corresponding quantitative data were analyzed containing only so-called "canonical" protein sequences. However, in UniProt also so-called isoforms are annotated which have a similar sequence as a canonical one, because they originate from the same gene or a gene family (UniProt, 2024a). Which sequence is chosen by UniProt as the canonical one is depending on a set of different criteria (UniProt, 2024c). Isoforms may be caused by different biological processes, for example alternative splicing (Müller-Esterl, 2018, pp. 208-213). The DNA is

transcribed to mRNA to copy the genetic information. The mRNA then contains coding parts, the exons, and non-coding parts, the introns. The introns are then removed by a procedure called splicing. Alternative splicing is a phenomenon, where from an mRNA multiple forms can be generated for example by removing a different set of exons. This leads to different proteins with similar sequence which are stemming from the same gene. Protein isoforms are an important research topic because even similar protein sequences may have a different function. This is why methods focusing on distinguishing between isoforms have been developed, e.g., IsoformResolver (Meyer-Arendt et al., 2011), COPF (Bludau et al., 2021), SEPepQuant (Dou et al., 2023) or SpliceVista (Zhu et al., 2014). This task is difficult because the high sequence overlap leads to a lot of shared peptides.

Large changes in the sequences may be annotated with completely different accession numbers in UniProt, for example the isoforms alpha and beta of the Lamina-associated polypeptide 2, which have the accession numbers P42166 and P42167, respectively (UniProt, 2024a). If isoform sequences are similar they are annotated with the accession number of the corresponding canonical sequence followed by a dash and a number, e.g., O14524-2, which is an isoform of NEMP1, the nuclear envelope integral membrane protein 1, where a sequence of 73 amino acids is missing compared to the canonical sequence (UniProt, 2024b). In the following, the term "isoform" is used to represent proteins that are additionally exported, when "canonical + isoform sequences" are downloaded from UniProt.

The influence of including isoforms on the database and quantitative bipartite peptide-protein graphs are analyzed exemplarily for data set D3 in this section, see also Schork et al., 2022. D3 is a human data set and the isoform annotation for *homo sapiens* proteins is expected to be good in UniProt. As a consequence, the impact of adding isoforms is expected to be the highest for this data set, without the interference of two large proteomes, like in dataset D4. As they have similar sequences, isoforms are expected to introduce sharedness between canonical proteins and their isoforms, leading potentially to more complicated graphs containing also proteins that do not have any unique peptide. While preparing the quantitative data and *in silico* digesting the database, the exact same settings as for D3 were used, namely peptides of lengths between eight and 50 amino acids and up to two missed cleavages.

The fasta files for data set D3 with isoforms (called D3_iso in the following) contains in total over 100,000 entries with over 22,000 additional isoform entries compared

Table 4.4: Unique and shared peptides stemming from an *in silico* digestion of D3_fasta and D3_iso_fasta (without and with isoforms). * Unlike the rest of this thesis, unique is defined here as unique for a protein accession. However, the difference between protein nodes and accessions is negligible when looking at the database level. ** Isoform-unique peptides are defined as shared peptides that are shared only between isoforms of the same protein.

	without isoforms			
with isoforms	unique	shared	not existing	sum
unique*	1,195,122	0	97,693	1,292,815
isoform-unique**	441,430	0	10,154	451,584
shared	26,673	1,320,439	109	1,347,221
sum	1,663,225	1,320,439	107,956	

to the canonical database. Of these isoforms, only 12 stem from *E. coli*, the vast majority from human. As a note for comparison with the other data sets, yeast has 29 and mouse 8,331 annotated isoforms in the reference proteome.

Table B.8 (appendix, p. 175) contains the direct comparison of total graph characteristics for data set D3 with and without isoforms for the database as well as the quantitative graphs. For the database graphs, the inclusion of isoforms leads to 26.4% more protein accessions, but only around 110,000 additional peptide sequences, i.e., 3.6%. The number of shared peptide nodes increases more than the unique peptide nodes. The number of graph decreases slightly while the number of isomorphism classes raises from 4,600 to 6,100. From the additional peptides 90.5% are unique (see table 4.4). Peptides that are unique without isoforms become shared in 28.1% of the cases, however, the majority remain isoform-unique (meaning that they are shared by two or more isoforms of the same family).

For the quantitative graphs, the number of quantified peptides decreases slightly from 20,500 to 19,500, which is a phenomenon known in proteomics when increasing the database size due to an increased probability for false positive hits (Kumar et al., 2017). However, the number of peptide nodes rises, potentially because additional peptides are often unique and form a new separate peptide node. Like for the database graphs, the number of graphs decreases while the number of isomorphism classes rises. The percentage of shared peptide nodes rises from 51.8% to 55.6%.

Comparing the ten largest isomorphism classes with and without isoforms (figure A.7, p. 153, vs. figure A.5, p. 151), it is noticeable that the top 10 graphs do not change much. Some graph types become a bit more or less common and the order

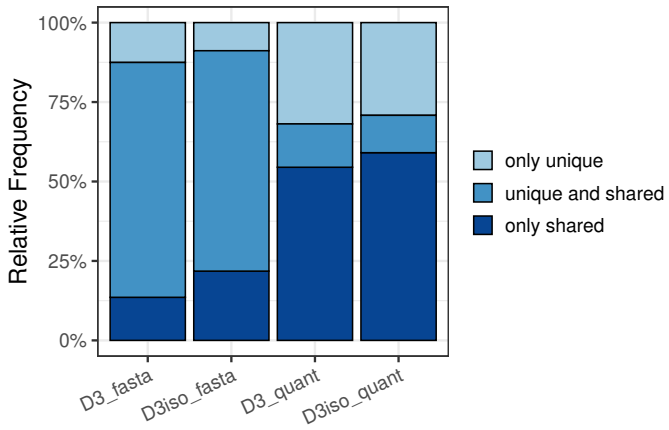


Figure 4.6: Percentages of protein nodes with ”only unique” vs. ”unique and shared” vs. ”only shared” peptides in relation to all protein nodes across all database and quantitative bipartite peptide-protein graphs for data set D3 with and without isoforms, compare to figure 4 in Schork et al., 2022.

of the top 10 changes slightly. However, it can be noticed that the proportion of the four graph types without any unique peptides rises slightly (e.g., the W-shaped graph goes from 2.76% to 3.84% and increases from fourth to third rank in the list). Over all graphs, it can be observed that the percentage of protein nodes with only shared peptides rises, from 13.6% to 21.8% for D3_iso_fasta and from 54.5% to 59.1% for D3_iso_quant, while the percentage of protein nodes with only unique peptides decreases (see figure 4.6).

4.5 Summary and discussion

In this chapter the bipartite peptide-protein graphs were characterized, which represent the relationship between peptides in proteins in bottom-up proteomics. These graphs are frequently used to represent the protein inference or the protein quantification problem. In chapter 5 the novel quantification method bppgQuant is proposed that makes use of these graph structures. As described in section 3.6, many current protein quantification methods do not fully exploit the potential of shared peptides and usually cannot quantify proteins without unique peptides. To get an overview about the situation in different data sets the bipartite peptide-protein graphs were generated and compared. It was particularly interesting how large the fraction of

protein nodes without unique peptides is, which is the most difficult case for protein quantification and the focus of bppgQuant.

To find suitable settings for the *in silico* digestion needed for constructing the database graphs, different values for the minimal peptide length and the maximal allowed number of missed cleavages were compared. The minimal peptide length has a large impact, as including too small peptides may lead to a largest graph containing almost all protein nodes, which is hard to handle and also to compare with the quantitative graphs later on. The smallest minimal peptide length leading to a handleable largest graph (less than 5% of all protein nodes) was chosen, which is between six and eight amino acids depending on the data sets. The threshold of six was chosen for D2_fasta which is based on yeast samples and is therefore the least complex data set with the least proportion of "sharedness" between proteins. Therefore the impact of small peptides is lower in comparison to for example D3_fasta or D4_fasta which both contain human proteins. For these two data sets the threshold of eight amino acids was chosen, as including peptides with lengths of seven or lower led to a too large largest graph. The respective thresholds were also applied to the quantitative peptide-level data for constructing the quantitative graphs. Although different thresholds were used for the different data sets, it is still possible to compare them. In contrast, allowing the largest graph to be too large (too small threshold) or to lose too many peptides (too large threshold) would be worse regarding the comparability. The influence of missed cleavages on the graphs was also investigated, but is much smaller than the influence of the minimal peptide length. Therefore it was decided to keep using up to two missed cleavages, which is an often-used default setting and was also used for the quantitative data during database search.

Database as well as quantitative graphs were generated for four different data sets and their corresponding protein sequence databases of varying organisms and complexity. In all cases the smallest possible graph with only one peptide and one protein node is the most common. This case is extremely easy to handle for protein inference and quantification. For inference, the protein can be reported directly because of the existence of a unique peptide. For quantification, simply the mean ratio for all unique peptide sequences that are present in the single peptide node could be used. In this case, there is no other protein that may influence a shared peptide ratio and therefore also the estimation of the protein ratio. This strategy however will always lead to a joint estimation of the ratio of all protein accessions

inside a collapsed protein node. In consequence a unique solution for the ratio of the protein node may be provided, however, the interpretation of the result is still ambiguous because the different protein accessions within a node cannot be resolved. The collapsing of protein nodes has much more impact on the quantitative graphs, where theoretically possible unique peptides are not quantified, than on the database graphs. For the quantitative graphs, between 8.4% (data set D2) and 40.6% (data set D4) of the protein nodes contain more than one protein accession, with a maximum of 73 accessions in a single node in data set D4. For D4, the average number of protein accessions per protein node is almost 2. For the database graphs, where all theoretically unique peptides are present, only up to 1% of protein nodes contain more than one accession for D4. Collapsing of protein nodes may here be caused by protein entries with differences only in a short amino acid sequence, so that these short unique peptides are not considered during graph construction.

When comparing the database graphs with the quantitative graphs two opposing effects can be observed. On the one hand large database graphs split up into smaller quantitative graphs when non-quantified peptide nodes vanish, leading for example to a higher proportion of the most simple graph type, which is easy to quantify. On the other hand, protein nodes or even whole quantitative graphs without unique peptides become more common. These are the hardest case for protein quantification. Missing peptides may also lead to an additional collapsing of protein nodes, leading to smaller graphs on average. These findings can be observed for example by looking at the ten largest isomorphism classes or characteristics aggregated over all graphs. Overall these differences highlight the importance of looking at the structures of the quantitative graphs for a specific data set and not relying on an unspecific view on the database graphs from an *in silico* digestion. Depending on the data set only a small proportion of the theoretically possible peptides will be identified and quantified in a sample. That a certain proportion of unique peptides does not get quantified may explain the higher proportion of protein nodes without unique peptides in the quantitative graphs. On the other hand, missing shared peptides may lead to a loss of connections between protein nodes, which causes graphs to break apart and may explain the higher proportion of smaller and less complex graphs.

Only a small part of the theoretically possible peptides are quantified, explaining the differences between database graphs and quantitative graphs. The missing quantification can have different reasons. First, the detectability of peptides is not only

influenced by their concentration in the sample, but also by a lot of different physico-chemical properties (Qeli et al., 2014). Second, not all protein entries of the database will be present in each cell type or body fluid, as many proteins are tissue-specific (see for example the Human proteome Atlas, Uhlén et al., 2015). However, it can be assumed that data sets measured with more modern mass spectrometers produce bipartite peptide-protein graphs closer to the comprehensive database graphs, as more peptides are identified and quantified. This has to be investigated further.

Throughout the thesis "uniqueness" of peptides is defined based on protein nodes (with an exception in table B.8, p. 175, to illustrate the influence of isoforms). I.e., a unique peptide is unique for a single protein node, which may contain multiple protein accessions because of the node collapsing step. This collapsing of protein nodes is relatively rare for the database graphs, as almost all protein sequences are distinguishable by looking at their theoretical peptides. However, many theoretically possible peptides are not quantified, which leads to more protein accessions that cannot be distinguished anymore. E.g., in dataset D3_quant, over 14,000 protein accessions fit to the quantified peptides that were collapsed to around 8,300 protein nodes. These accessions within the same protein node also have to be treated as a single protein group during protein quantification and will only get a common protein ratio estimated. Therefore, defining uniqueness of peptides regarding the protein nodes makes perfectly sense in context of protein quantification. Other definitions of uniqueness, e.g., on the level of protein accessions, isoform families or even genes (Saltzman et al., 2018) may be suitable for other use cases.

Besides the differences between quantitative and database graphs, also differences between the four analyzed data sets were observed. Especially the graphs from data set D2 showed a lower complexity compared to the other data sets. With over 87%, the smallest possible graph type was much more frequent than for the other data sets for the quantitative as well as the database graphs. Also, the proportion of shared peptides and therefore the proportion of protein nodes without any unique peptide is comparably smaller. This may be explained by the different complexities of the main organisms behind the data sets, which is also reflected by their genome sizes. While human (3,100 Mb, National Center for Biotechnology Information, 2022) and mouse (2,700 Mb, National Center for Biotechnology Information, 2020) have more or less comparable genome sizes, the genome of yeast, which is the basis for data set D2, is considerably smaller with only 12.1 Mb (National Center for Biotechnology Information, 2014). Mb stands for megabase, i.e. one million nucleotide base pairs

of the DNA. These effects make the protein quantification overall easier for D2 than for D1, D3 or D4. In consequence, it would be thinkable that less complex protein quantification algorithms, even if they do not make appropriate usage of shared peptides, would work sufficiently for this data set.

The data set D4 has been added compared to the paper Schork et al., 2022. The corresponding samples are a mixture of the proteomes of mouse and human cell lines. The mouse and human proteome have a certain degree of similarity, which leads to a large overlap of theoretical or quantified peptides. As a consequence, D4 has overall the largest proportion of shared peptides and therefore also the largest proportion of protein nodes without unique peptides. This data set will resemble the most difficult challenge for protein quantification because of this.

It has to be noted that not all of the four data sets were measured on the same type of mass spectrometry instrument. D1 was measured on a Q Exactive HF, D2 and D3 on an LTQ Orbitrap and D4 with an Orbitrap Fusion Lumos, all by Thermo Fisher Scientific. Therefore, the comparability between the data sets could be challenged. However, already for the theoretical database graphs, differences between the different data sets can be observed, especially for data set D2, while D1, D3 and D4 show a more or less comparable behaviour. Also, for the quantitative graphs the differences between D2 and D3 are very large, although those data sets were measured both with the LTQ Orbitrap instrument type. On the other hand, D1 and D3 show large similarities but are measured with two different instrument types. While different instruments may certainly have an effect on the amount and set of quantified peptides (Elias et al., 2005), the results of the graph comparisons indicate that the differences between yeast and mouse or human cells are larger than the differences between the instruments. Especially because in contrast to Schork et al., 2022, the MaxQuant versions, UniProt protein sequence database versions as well as the search engine settings were completely harmonized across data sets, a comparison of the different data sets based on the bipartite peptide-protein graphs is still possible here.

The influence of including isoforms was exemplarily analyzed for data set D3. The proportion of shared peptides and in consequence the number of protein nodes with only shared peptides rises whe protein isoforms are added to the database. The term isoform refers to a set of similar proteins that share a lot of their amino acid sequence, e.g., because they stem from the same gene region by alternative splicing. In UniProt, known isoforms are annotated and it is possible to export FASTA files

containing only canonical sequences or additionally also the isoforms. However, when using the UniProt reference proteomes, still potential isoforms may be present in the canonical database because these entries are not always reviewed or annotated as isoforms. Because of the high sequence similarity, isoforms are prone to have many shared peptides with the corresponding canonical sequence, in some cases even no unique peptides are existing or not quantified. However, quantifying the isoforms separately may be of high relevance for biomarker and molecular research, as isoforms may have different functions inside the cells (Buée et al., 2000; Wei et al., 2012). A challenge for this aim is, that for two specific isoforms only shared peptides may be quantified, so these two accessions would end up in a collapsed protein node and only a common protein ratio could be estimated.

Besides trypsin, also other proteases exist that are used for the enzymatic digestion for proteins to peptides. For example, the enzyme Glu-C has a completely different cleavage behaviour, as it cleaves after the amino acid glutamic acid (E) and therefore creates completely different peptides during digestion compared to trypsin (Zhang et al., 2013). Most likely, the results from this chapter cannot directly be transferred to other digestion enzymes, as the graph structure may heavily be affected.

In contrast to the paper Schork et al., 2022, the four data sets were preprocessed with the same software version of MaxQuant version, using also the same current version of the UniProt database. This was done to ensure a better comparability between the data sets. However, only small and minor differences in the results between the Schork et al., 2022 and this chapter here were observed for the data sets D1-D3. An additional data set D4 was added, where proteins from human and mouse cells were mixed in different ratios. There are many peptides shared also between mouse and human, making it the most complex of the four data sets and a good test case for the protein quantification later on.

In conclusion, the database and quantitative bipartite graphs generated from different data sets and corresponding protein sequence databases were characterized. It is not clear how well the results could be transferred to other organisms, especially those that are not human or model organisms and may have a less well characterized proteome. In the following chapter 5, the bipartite peptide-protein graphs together with quantitative information on the peptide level will be used to improve protein quantification, especially for proteins without unique peptides.

5 Novel protein quantification method

In this chapter the novel protein quantification method bppgQuant is presented, which makes use of the bipartite graph structures. It calculates protein ratios from peptide ratios, while incorporating shared peptides and focusing on gaining quantitative information for protein nodes without unique peptides. This chapter is structured as follows: In section 5.1 the methodological derivation of bppgQuant is explained. The implementation of bppgQuant is explained in section 5.2. Then, the four quantitative data sets are explained further together with the necessary preprocessing steps, see section 5.3. The performance of bppgQuant is compared with the methods SCAMPI and PQP on these data sets in section 5.4. The discussion of the results is given in section 5.5.

5.1 bppgQuant method for protein quantification

In this section the novel method bppgQuant (Bipartite Peptide-Protein Graph-based protein QUANTification) is presented. It is based on an equation system and subsequent optimization to estimate protein ratios out of peptide ratios. In section 5.1.1, the general equation system that represents the relationships between peptide and protein ratios is explained and solved exemplarily for two simple cases. In section 5.1.2 the equation system is extended to an optimization problem. The following section 5.1.3 explains how cases with multiple optimal solutions are handled. Finally, in section 5.1.4 an approach is presented that quantifies only a subset of protein nodes that sufficiently contribute to the measured peptide ratios, to decrease the number of inaccurate quantification results. In the following it is assumed that bipartite peptide-protein graphs have been constructed based on quantified peptides as described in section 4.2. Furthermore, peptide ratios r_j between two experimental

conditions for these quantified peptides are available. In this section the notation described in table 3.1 (p. 16) is followed.

5.1.1 Equation system for the relationship between peptide and protein ratios

In the following, each connected component of the bipartite peptide-protein graph, containing m protein and n peptide nodes is handled separately. It is assumed that peptide ratios between two samples A and B are given (A and B can also be groups of samples, e.g., technical replicates of the same experimental condition, that are summarized beforehand). The ratio r_j for peptide j can be calculated from the measured peptide intensities and is therefore known. Let A_i and B_i be the unknown quantities of protein i in samples A and B, respectively. Let R_i be the unknown ratio of protein i, namely $R_i = \frac{B_i}{A_i}$. Let $C_i = \frac{Ai}{\sum A_i}$ be the proportion of protein i in sample A compared to the other proteins in the same connected component. For each peptide ratio, an equation can be formulated (first under the simplified assumption that there is no noise in the measurements), which models the relationship between the unknown protein ratios and the measured peptide ratios.

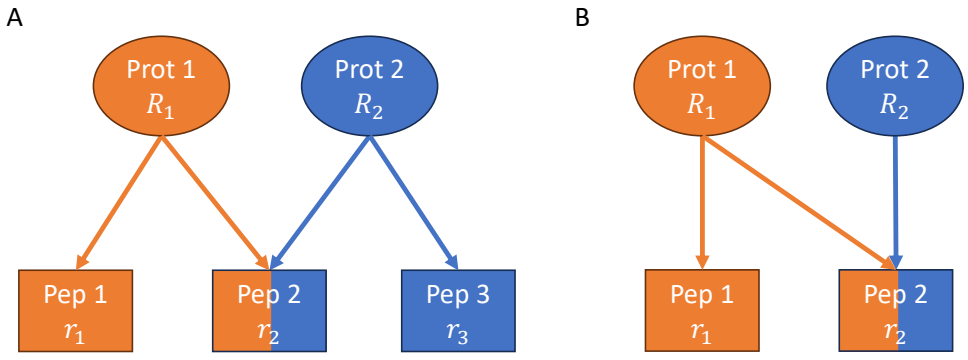


Figure 5.1: Examples of two simple bipartite graph types. The colour gradient of the shared peptides indicate that the peptide ratio is influenced by both protein ratios. A: Example of an M-shaped graph with two protein nodes and three peptide nodes. B: Example of an N-shaped graph with two protein nodes and two peptide nodes.

Equation for a unique peptide

For a unique peptide j, like peptide 1 in figure 5.1A, which exclusively belongs to protein 1, the peptide ratio is the same as the ratio of the corresponding protein i (if noise in the measurements is ignored), see figure 5.1A. Therefore the equation for a unique peptide can be formulated as

$$r_j = R_i \Leftrightarrow r_j - R_i = 0 \Leftrightarrow \frac{r_j}{R_i} = 1 \Leftrightarrow \log(r_j) - \log(R_i) = 0. \tag{5.1}$$

Equation for a peptide shared by two or more proteins

The example graph in figure 5.1A contains a peptide that is shared by exactly two protein nodes (peptide 2, shared by protein nodes 1 and 2). The measured peptide ratio r_2 is influenced by both protein ratios, as indicated by the colour gradient. It can be described as a combination of the unknown quantities of the proteins in the samples A and B ($A_1, A_2, B_1, B_2 > 0$). It is assumed, that the intensity of a shared peptide j in each sample is given as the sum of the intensities of all corresponding proteins times a peptide-specific factor α_j (e.g., if the ionization efficiency of a peptide is low, it will not get a very high intensity. However, this applies to both sample A and B, so it will be cancelled out). The equation for a peptide j shared by two proteins i_1 and i_2 is given by

$$r_j = \frac{\alpha_j \cdot (B_{i_1} + B_{i_2})}{\alpha_j \cdot (A_{i_1} + A_{i_2})} = \frac{B_{i_1} + B_{i_2}}{A_{i_1} + A_{i_2}} \tag{5.2}$$

if $A_{i_1} + A_{i_2} \neq 0$, i.e. at least one of the two proteins have to be present in sample A. In a general setting, the whole connected component consists of $m \geq 2$ proteins ($i = 1, \ldots, m$) and n peptides $j = 1, \ldots, n$. δ_{ij} is an indicator variable that is 1, if peptide j is a tryptic peptide of the sequence of protein i, and 0 otherwise. In this general case, the ratio of peptide j can then be represented as follows:

$$
\begin{aligned}
r_j &= \frac{\sum_{i=1}^{m} B_i \cdot \delta_{ij}}{\sum_{k=1}^{m} A_k \cdot \delta_{kj}} \\
&= \sum_{i=1}^{m} \left[B_i \cdot \delta_{ij} \cdot \frac{1}{\sum_{k=1}^{m} A_k \cdot \delta_{kj}} \right] && \mid B_i = R_i \cdot A_i \\
&= \sum_{i=1}^{m} \left[R_i \cdot A_i \cdot \delta_{ij} \cdot \frac{1}{\sum_{k=1}^{m} A_k \cdot \delta_{kj}} \right] \\
&= \sum_{i=1}^{m} \left[R_i \cdot \frac{A_i}{\sum_{l=1}^{m} A_l} \cdot \delta_{ij} \cdot \frac{\sum_{l=1}^{m} A_l}{\sum_{k=1}^{m} A_k \cdot \delta_{kj}} \right] \\
&= \sum_{i=1}^{m} \left[R_i \cdot \frac{A_i}{\sum_{l=1}^{m} A_l} \cdot \delta_{ij} \cdot \frac{1}{\sum_{k=1}^{m} \frac{A_k}{\sum_{l=1}^{m} A_l} \cdot \delta_{kj}} \right] && \mid C_i := \frac{A_i}{\sum_{l=1}^{m} A_l};\ \sum_{i=1}^{m} C_i = 1 \\
&= \sum_{i=1}^{m} \left[R_i \cdot C_i \cdot \delta_{ij} \cdot \frac{1}{\sum_{k=1}^{m} C_k \cdot \delta_{kj}} \right] \\
&= \sum_{i=1}^{m} \left[R_i \cdot w_{ij} \right]
\end{aligned}
\tag{5.3}
$$

The peptide ratio of peptide j can be interpreted as an weighted sum of protein ratios. The weight w_{ij} for a certain protein is > 0 if the peptide belongs to the protein and zero otherwise. The weights include the factors $C_i, \ldots, C_m$, which are also unknown. They represent the proportion of the intensity of protein i in sample A, compared to the summed up intensity of all proteins in this connected component. The existence of these factors makes sense, as a shared peptide ratio will be influenced more by the ratio of high-abundant proteins. As a further condition, the sum of all C_i needs to be 1.

Solving the equation system

For a given connected component with m proteins and n peptides, there is one equation per peptide and the equation system to be solved is the following:

$$r_j = \sum_{i=1}^{m} \left[R_i \cdot C_i \cdot \delta_{ij} \cdot \frac{1}{\sum_{k=1}^{m} C_k \cdot \delta_{kj}} \right] \quad \forall j = 1, \ldots, n \tag{5.4}$$

under the conditions that $R_i > 0$ and $0 < C_i < 1 \ \forall i = 1, \ldots, m$ and $\sum_{k=1}^{m} C_k = 1$. This is a non-linear equation system with n equations, $2m$ unknowns, one equality constraint and several inequality constraints. For non-linear equation systems, obtaining information about the solvability of these is not as straightforward as for linear equation systems. For small peptide-protein graphs and therefore small equation systems, exemplarily the solvability can be assessed.

For example, in case of $m = 2$ protein nodes and $m = 3$ peptide nodes (M-shaped graph, see figure 5.1A, p. 72) the following can be derived: Peptide 1 and 3 are unique for protein 1 and 2, respectively, while peptide 2 is shared by both proteins. The peptide ratios r_1, r_2 and r_3 are calculated from the measured peptide intensities, while the protein ratios R_1, R_2 as well as the weights C_1 and C_2 are unknown. The indicators δ_{ij} can be derived from the graph structure, in this case $\delta_{11} = \delta_{12} = \delta_{22} = \delta_{23} = 1$ and $\delta_{13} = \delta_{21} = 0$. Using that $C_1 + C_2 = 1$, the following equations are obtained.

$$\begin{aligned} r_1 &= R_1 \\ r_2 &= R_1 \cdot C_1 + R_2 \cdot C_2 = r_1 \cdot C_1 + r_3 \cdot (1 - C_1) = C_1(r_1 - r_3) + r_3 \\ r_3 &= R_2 \end{aligned} \tag{5.5}$$

In any case, the protein ratios equal the peptide ratios of the corresponding unique peptides. The measured ratio of the shared peptide r_2 determines, if the equation system can be completely solved also for the weights C_1 and C_2. If $r_1 = r_3$, it follows from the second equation that also $r_2 = r_3$, meaning that all measured peptide ratios must be the same, otherwise there is a contradiction. If $r_1 \neq r_3$, it follows that:

$$\begin{aligned} r_2 &= C_1(r_1 - r_3) + r_3 \\ \Leftrightarrow r_2 - r_3 &= C_1(r_1 - r_3) \\ \Leftrightarrow \frac{r_2 - r_3}{r_1 - r_3} &= C_1. \end{aligned} \tag{5.6}$$

As a proportion, C_1 needs to be between 0 and 1. If $r_1 > r_3$ it follows that:

$$\begin{aligned} C_1 \geq 0 \Leftrightarrow \frac{r_2 - r_3}{r_1 - r_3} \geq 0 \quad &\Leftrightarrow r_2 - r_3 \geq 0 \Leftrightarrow r_2 \geq r_3 \\ C_1 \leq 1 \Leftrightarrow \frac{r_2 - r_3}{r_1 - r_3} \leq 1 \quad &\Leftrightarrow r_2 - r_3 \leq r_1 - r_3 \Leftrightarrow r_2 \leq r_1. \end{aligned} \tag{5.7}$$

On the other hand, if $r_1 < r_3$:

$$\begin{aligned} C_1 \geq 0 \Leftrightarrow \frac{r_2 - r_3}{r_1 - r_3} \geq 0 \quad &\Leftrightarrow r_2 - r_3 \leq 0 \Leftrightarrow r_2 \leq r_3 \\ C_1 \leq 1 \Leftrightarrow \frac{r_2 - r_3}{r_1 - r_3} \leq 1 \quad &\Leftrightarrow r_2 - r_3 \geq r_1 - r_3 \Leftrightarrow r_2 \geq r_1. \end{aligned} \tag{5.8}$$

This means that regardless of whether r_1 or r_3 is the larger of the two ratios, the ratio of the shared peptide r_2 has to be between these two unique peptide ratios for the equation system to be solvable. If this condition is not fulfilled, the equation system is not solvable (i.e., there is a contradiction). In theory, this situation should not occur, if the peptide ratios strictly follow the equation and there is no noise. However, in practice, the mass spectrometry measurements are affected by different sources of noise which can cause the shared peptide ratio to deviate from this rule.

Another small example is the N-shaped graph, as shown in figure 5.1B (p. 72). Here, protein 1 has one unique and one shared peptide, while protein 2 has only connections to the shared peptide. In this case two equations and one constraint are present in the equation system, together with four unknowns (R_1, R_2, C_1, C_2). This is not enough to solve the system uniquely, however, still a set of possible solutions can be derived. In this case, the following equations are derived:

$$
\begin{aligned}
r_1 &= R_1 \\
r_2 &= R_1C_1 + R_2C_2 = r_1(1 - C_2) + R_2C_2 = r_1 - r_1C_2 + R_2C_2 \\
&\Rightarrow r_2 - r_1 = C_2(R_2 - r_1) \\
&\Rightarrow \frac{r_2 - r_1}{C_2} + r_1 = R_2
\end{aligned}
\tag{5.9}
$$

or equivalent:

$$\Rightarrow C_2 = \frac{r_2 - r_1}{R_2 - r_1}.$$

The relationship between R_2 and C_2 cannot be further resolved without a further equation. However, as C_2 has to be greater than 0 and smaller than 1, there are three different situations:

$$
\begin{aligned}
r_2 > r_1 &\Rightarrow R_2 > r_1 \wedge R_2 > r_2 \Rightarrow R_2 > r_2 \\
r_2 < r_1 &\Rightarrow R_2 < r_1 \wedge R_2 < r_2 \Rightarrow R_2 < r_2 \\
r_2 = r_1 &\Rightarrow R_2 = r_1 = r_2
\end{aligned}
\tag{5.10}
$$

That means that r_2 is the lower or upper bound for R_2, depending on the constellation for r_1 and r_2. This can also be formulated by the following limits:

$$
\begin{aligned}
C_2 = 1 &\Rightarrow R_2 = \frac{r_2 - r_1}{1} + r_1 = r_2 \\
r_2 > r_1 &\Rightarrow \lim_{C_2 \to 0} \frac{r_2 - r_1}{C_2} + r_1 = \infty \\
r_2 < r_1 &\Rightarrow \lim_{C_2 \to 0} \frac{r_2 - r_1}{C_2} + r_1 = -\infty
\end{aligned}
\tag{5.11}
$$

That means that R_2 has the solution interval (r_2, ∞) if $r_2 > r_1$ or $(-\infty, r_2)$, if $r_2 < r_1$.

In summary, in this case the protein ratio R_1 (protein is connected to the unique as well as the shared peptide) is fixed at the level of the unique peptide ratio. On the other hand, for protein ratio R_2 an upper or lower bound of the ratio can be given, depending on the constellation of measured peptide ratios. The limit is always the shared peptide ratio, and the protein ratio is always farther away from the unique peptide ratio r_1. This makes sense, as the peptide ratio r_2 otherwise cannot have emerged from the two protein ratios.

5.1.2 Optimization problem

Exactly solving the above mentioned equation system in equation 5.4 (p. 75) to obtain a single solution for all R_i and C_i is only possible in a few rare cases like the M-shaped graph, if no contradiction occurs. In general, most of the equation systems will not be solvable. For example, often two or more peptide sequences inside the same peptide node exist, which leads to the same right-hand-side of the formula. Although these peptides are expected to have the same ratio in theory, in practice the biological and technical variability of the measurements will lead to different values and therefore different left-hand-sides of the formula. This directly leads to a contradiction in the equation system. This is why instead of solving the equation system exactly, it can be transformed into an optimization problem by introducing error terms that should be minimized. If a system is exactly solvable, this optimization should lead to the exact solution with an error term of zero (at least approximately, due to numerical errors).

The measured ratio of peptide j can be modeled with a multiplicative error term $\exp(\varepsilon_j)$, assuming a normal distribution for ε_j. As ratios are modeled here, multiplicative error terms seem more reasonable than additive error terms. Multiplicative errors also guarantee to obtain symmetric results on the multiplicative scale, e.g., that using the reciprocal of the peptide ratios will lead to the reciprocal of the protein ratio estimates:

$$\begin{aligned} r_j &= \sum_{i=1}^{m} \left[R_i \cdot \frac{C_i \cdot \delta_{ij}}{\sum_{k=1}^{m} C_k \cdot \delta_{kj}} \right] \cdot \exp(\varepsilon_j) \rightarrow \varepsilon_j \sim N(0, \sigma^2) \\ \Rightarrow \log(r_j) &= \log\left(\sum_{i=1}^{m} \left[R_i \cdot \frac{C_i \cdot \delta_{ij}}{\sum_{k=1}^{m} C_k \cdot \delta_{kj}} \right] \right) + \varepsilon_j. \end{aligned} \tag{5.12}$$

If protein ratios R_i and weights C_i were given, an estimation for the peptide ratios r_j could be given by the equation 5.12. As multiplicative errors are used, the ε_j are the difference between the estimated and the measured peptide ratio, on logarithmic scale. With this, an optimization problem is set up, using the least squares approach, i.e., by minimizing the sum of squared error terms over all peptides in the same

bipartite graph. For each bipartite graph, there exists one error term per peptide, i.e. n error terms. The optimization problem is then formulated as

$$\underset{R_i, C_i, i=1,\ldots,m}{\arg\min} \left(\sum_{j=1}^{m} \varepsilon_j^2 \right) \tag{5.13}$$

$$\text{with } \varepsilon_j = \log(r_j) - \log \left(\sum_{i=1}^{m} \left[R_i \cdot \frac{C_i \cdot \delta_{ij}}{\sum_{k=1}^{m} C_k \cdot \delta_{kj}} \right] \right)$$

with respect to $R_i \geq 0$, $C_i \in [0, 1]$ and $\sum_{i=1}^{m} C_i = 1$.

With equation 5.13 there is a non-linear optimization problem with an equality constraint and several inequality constraints for each bipartite peptide-protein graph. The R package `Rsolnp` with the function `solnp()` can handle this kind of optimization problems and was used here to calculate estimates for the protein ratios R_i as well for the weights C_i. The `solnp()` function uses an augmented Lagrange multiplier method for optimizing nonlinear function as described in Ye, 1987. As starting values for the optimization algorithm $\frac{1}{m}$ is used for $C_i \; \forall \; i = 1, \ldots, m$. For the R_i the geometric means of the unique peptide ratios are used. For proteins without unique peptides the geometric means over all associated peptide ratios are used. These are simple a priori estimates for the protein ratios and weights, if no further information is available.

5.1.3 Handling systems with many optimal solutions

As shown in section 5.1.1 in equations 5.9 - 5.11 (page 77) for the small N-shaped graph, it is possible that multiple solutions lead to the same minimal error (in this specific case an error term of 0). This can happen also in many types of larger graphs, especially when protein nodes without unique peptides are present. As shown before, the solutions then depend on the relationship between the protein ratios R_i and the factors C_i.

However, in the optimization strategy described above using the `rsolnp()` function, only one of the possible optimal solutions is found. Ideally, the whole interval of

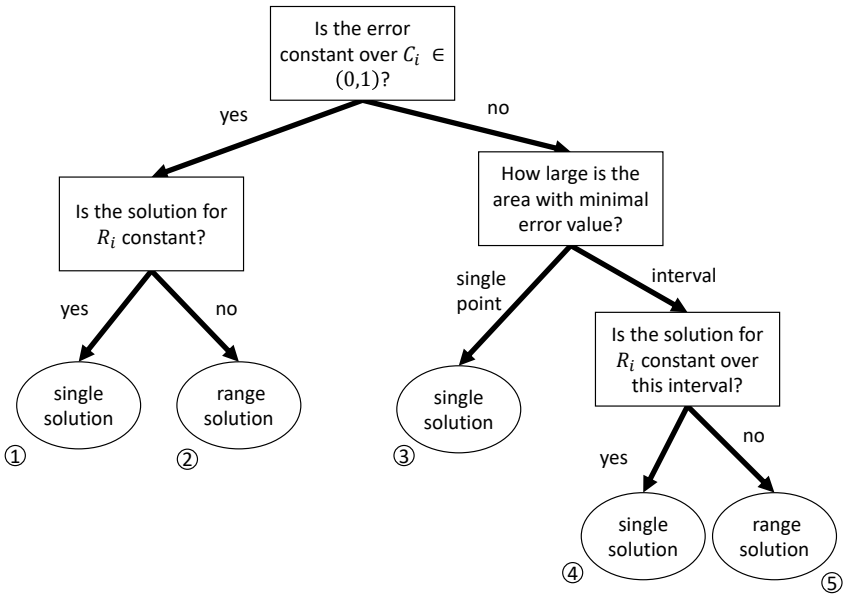


Figure 5.2: Decision tree for determining point or range solutions from the result tables obtained after iterating over a grid of different C_i values and optimization.

optimal solutions could be assessed. To study the behavior of the optimization and to get an idea of the solution space, the following strategy was employed. For each protein, a grid for the corresponding factor C_i is constructed between 0.001 and 0.999 with step sizes of 0.001 leading to 999 grid points. The values 0 and 1 were excluded because they lead to computational problems, i.e. division by 0. Additionally, more grid points on the borders are added, namely $0.0001, \ldots, 0.0009$ and $0.9991, \ldots, 0.9999$. The specific C_i is fixed on one value of the grid after another, while the corresponding protein ratios as well as all other variables are optimized. By this, for each value of C_i estimates of the protein ratios are obtained as well as an error term, leading to a large result table.

By comparing the results for the error, it can become clear if there is a unique solution with a minimal error, or if multiple solutions can reach the same minimal error term. In total, six situations are considered as shown in figure 5.2. First, it is determined if the error term is constant for the whole range of $C_i \in (0, 1)$. For this, a small tolerance of 10^{-10} was used to account for numerical variations in the optimization step. If the error term is constant, the obtained estimated values for R_i may be constant or not. Here, a much smaller tolerance of 10^{-4} was used on the log2-transformed protein ratios. Differences of protein ratios of this magnitude are usually irrelevant for real applications. If the values for R_i are constant, this

solution is given as a point solution (case 1). If it is not constant, the minimum and maximum value is given as upper and lower bound of the range solution (case 2). In case the error term is not constant over the whole interval of $C_i \in (0, 1)$, it is determined if the error term is constant at its minimum value for at least a part of the C_i values, using again the tolerance of 10^{-10}. If there is only a single data point within the tolerance of the minimal error term, the corresponding R_i value is reported as a point solution (case 3). Otherwise, it is checked if R_i is constant (with 10^{-4} tolerance) for the C_i interval leading to the minimal error term. Depending on this, a point solution (case 4) or a range solution (case 5) is reported. Case 6, which is not shown in the figure, occurs when the graph is of the most simple graph type with only one protein and one or more unique peptide sequences. In this case, the only weight C_1 is always 1 and no variation of the C_i is needed. Then, the optimization as explained in section 5.1.2 (p. 78) will be applied, which in this simple case simplifies to calculating the geometric mean of the unique peptide ratios for the corresponding protein node. In general, a disadvantage of this approach to obtain multiple solutions is the longer running time, especially for larger systems with many protein nodes. Reducing the grid size however may result in inaccurate results.

5.1.4 Selection of a subset of protein nodes

In the sections before a protein ratio or interval of ratios is calculated for each protein node present in a bipartite peptide-protein graph. This follows the assumption, that from each protein node, at least one accession is actually present in at least one of the samples (see equation 5.3, p. 74 which requires that at least one protein abundance is larger than zero, $\exists\ k \in \{1, \ldots, m\}$ with $A_k \neq 0$, otherwise the denominator of the fraction is zero). However, this assumption may not hold in many situations, as not all proteins in a FASTA database are necessarily expressed in a specific tissue or body fluid and therefore may be absent from all samples.

A variable-selection-like approach was used to reduce the number of protein nodes and only keeps those that sufficiently contribute to the explanation of the measured peptide ratios. First, the bppgQuant method is applied to the whole bipartite peptide-protein graph to get a reference value for the error term. As obtaining all possible solutions is irrelevant at this step, the approach described in section 5.1.3 (p. 79) is not applied, but a single optimization step. If a protein node is removed,

the error term will stay constant or increase, as fewer proteins are present to explain differences between the peptide ratios. For each protein node it is checked if it can be removed or if that would leave peptide nodes unconnected (e.g., if the protein node has unique peptides). If it can be deleted, it is removed from the graphs and the minimization of the error term is repeated. If the error term is less than 5% larger than the reference error, this branch is investigated further and one by one, more protein nodes are deleted, until the error term is too high or at least one peptide is unconnected. The elimination of a protein node may cause the bipartite graph to break into connected components. In this case the error terms resulting from the two or more resulting graphs are summed up. Finally, the smallest subset of protein nodes that lead to an error term within the threshold of 5% from the reference error is selected. In case of a tie, the first observed subset is chosen, meaning that currently the final outcome depends on the order of the protein nodes in the graph.

The reasoning behind this approach is the following: If a protein node is important to explain differences in the measured peptide ratios, its deletion will lead to a much higher error term. If a protein node was for example not present in the sample and does not contribute to the peptide ratios, its deletion will only have a small effect on the error term. Examples of this approach will be shown in section 5.4.6 (p. 116).

5.2 Software implementation

The code used for this thesis was implemented using R version 4.3.1 (R Core Team, 2023) and RStudio version 2023.09.1 (Posit team, 2023). The bipartite peptide-protein graphs were generated from the quantitative peptide-level data using the developed `bppg` package, see section 4.3 (p. 44). The optimization of the error term was done using the `Rsolnp` package (Ghalanos and Theussl, 2015) and executed on a server using the `batchtools` package (Lang et al., 2017). Normalization of the quantitative peptide-level data was done using the loess normalization from the `limmma` package (Ritchie et al., 2015) or the LTS normalization from the `vsn` package (Huber et al., 2002). ROC curves and AUC calculations were performed using the `pROC` package (Robin et al., 2011). Visualizations of the results were generated using the `ggplot2` package (Wickham, 2016) while using functionality from the `cowplot` (Wilke, 2020), `ggpubr` (Kassambara, 2023) and `scales` package (Wickham and Seidel, 2022). An overview over further R packages used for the analyses in

chapters 4 and 5 is given in table B.9 in the appendix (p. 176), including version numbers. The programming code for all analyses and visualizations in this thesis can be found on Zenodo under the following link: `https://zenodo.org/records/11120878`.

5.3 Test data sets and data preparation

In this section the four data sets used for method evaluation and comparison are described as well as the preparation of the quantitative peptide-level data.

5.3.1 Test data sets with known protein ratios

To evaluate the proposed method for calculating protein ratios from peptide ratios and to compare it with other established methods, different gold standard data sets are used, as already explained in section 4.1. The idea behind such data sets is to add certain proteins in varying concentrations into a stable background proteome (spike-in proteins). By this, the expected protein ratios of the spike-in proteins are known, while mimicking a complex sample by using a whole background proteome. In some cases there are only some spike-in proteins (data set D1 and D2), in others the whole proteome of different organisms are mixed leading to a less controlled but more complex setting (data set D3 and D4). In the following, details on these gold standard data sets are given regarding the expected ratios and spiked-in proteins that were omitted in section 4.1.

For data set D1 (Barkovits et al., 2020; Uszkoreit et al., 2022), 13 non-mouse proteins were spiked into 20 µg of C2C12 lysate in different concentrations, as shown in table B.10. Different combinations of concentrations between 0.1 and 10 pmol of spike-in proteins lead to 5 experimental conditions. Each of the states was measured in three replicates, leading to 15 samples in total. By building ratios between the different states, a wide variety of theoretical present ratios of 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100 can be reached.

In data set D2 (Ramus et al., 2016a; Ramus et al., 2016b), yeast was used as the background organism. The UPS1 (Universal Proteomics Standard Set) (Merck KGaA, 2023) was spiked into yeast lysate, which contains 48 human proteins in

equimolar amounts. Nine increasing UPS1 concentration levels were used, ranging from 50amol/µg, up to 50,000 amol/µg. Therefore, ratios up to 1000 can be reached when comparing condition 9 with condition 1. Each state was measured in three replicates, leading to 27 samples in total.

In data set D3 (Cox et al., 2014), 60 µg HeLa cell lysate (human cells) were mixed with either 10 µg or 30 µg of E.coli lysate, each fractionated into 24 fractions and measured in three technical replicates. Therefore, by building ratios between the two states, a theoretical ratio of three is obtained for all E.coli proteins. In contrast to D1 and D2, here the whole proteome of the E.Coli bacterium is spiked into the background and not a selected set of proteins. On the one hand, this leads to a higher number of proteins with a known ratio. On the other hand, not every protein is spiked-in separately, but as a lysate of the whole E.Coli cells. This can lead to deviations from the ratio of three for the individual proteins, however, on average the ratio of three is expected.

For the data set D4 (Saltzman et al., 2018), human HeLa as well as mouse NIH-3T3 cells were mixed in seven different ratios: 0%, 10%, 25%, 50%, 75%, 90% and 100% HeLa. The mixing and measurement was repeated two to three times for each condition. By comparing different mixtures, different expected ratios for human and mouse proteins can be obtained between 1.1 and 10. Even a comparison of pure human and pure mouse samples is possible.

An overview over the different data sets can be found in table B.1 (appendix, p. 171). The raw files were downloaded using the respective PXD number from PRIDE (Perez-Riverol et al., 2022). The data were further processed as explained in section 4.1, without including isoforms.

In the following, pairwise ratios between experimental conditions will be considered. While in data set D3 only one pairwise comparison is possible, D2 has the most pairwise comparisons with 36 (nine experimental conditions). Data set D4 contains a 100% human and a 100% mouse condition, which would lead to expected ratios of zero or ∞ when compared to another condition. This scenario will lead to corresponding zero or ∞ peptide ratios, which bppgQuant cannot handle appropriately currently (see section 6.1 in the outlook chapter). Therefore, pairwise comparisons containing either 100% mouse or 100% human samples for D4 where omitted during analysis.

5.3.2 Normalization of the peptide-level data

The MaxQuant software (Cox and Mann, 2008) generates a `peptides.txt` file containing the results of peptide identification and quantification. The peptide data containing non-normalized ("raw") peptide intensities were processed as follows. Peptide sequences matching to decoy proteins were removed. Peptides with an intensity of 0 over all samples were removed. Peptide intensities were log2-transformed. The peptide intensities still contain technical variation or biases next to the biological variation. The process of normalization aims at the reduction of the technical variation, which may be caused by small changes in the experimental conditions, like temperature, pipetting errors or age of the chromatography column (Karpievitch et al., 2012). After normalization, the samples are better comparable and prepared for the following analysis (Rozanova et al., 2023). A general assumption of common normalization methods is that the large majority of proteins or peptides does not change between samples, so that a ratio around 1 is expected for the majority of proteins (Karpievitch et al., 2012). This assumption is reasonable in many applications, as for example a certain disease does not change the whole proteome of a cell, but often only specific pathways that are affected.

Local regression normalization (loess) is a method originally developed for microarray data (Bolstad et al., 2003), but has proven to work well also on proteomics data (Chawade et al., 2014; Välikangas et al., 2018). Briefly, the loess normalization method generates MA-Plots based on the average and difference of the log-transformed intensities of two samples. A local linear regression line is fitted to the point cloud to account for potential differences in bias between high and low-abundant proteins. Using the assumption that the majority of proteins does not change between samples, the local regression line is shifted and the data points are back-transformed. To normalize more than two samples, the fast cyclic loess approach is applied, where each sample is normalized to an average representative sample and the whole procedure is repeated several times (Ballman et al., 2004). For each of the four data sets, the loess normalization was applied using the `normalizeBetweenArrays()` function of the limma package. This method is compared with the non-normalized data. Naturally, the protein quantification can only work properly with a good quality of peptide intensities, so a good normalization is required here. For data sets D1 and D2, the assumption that the majority of proteins do not change can reasonably be made, as only a few proteins are spiked into a stable

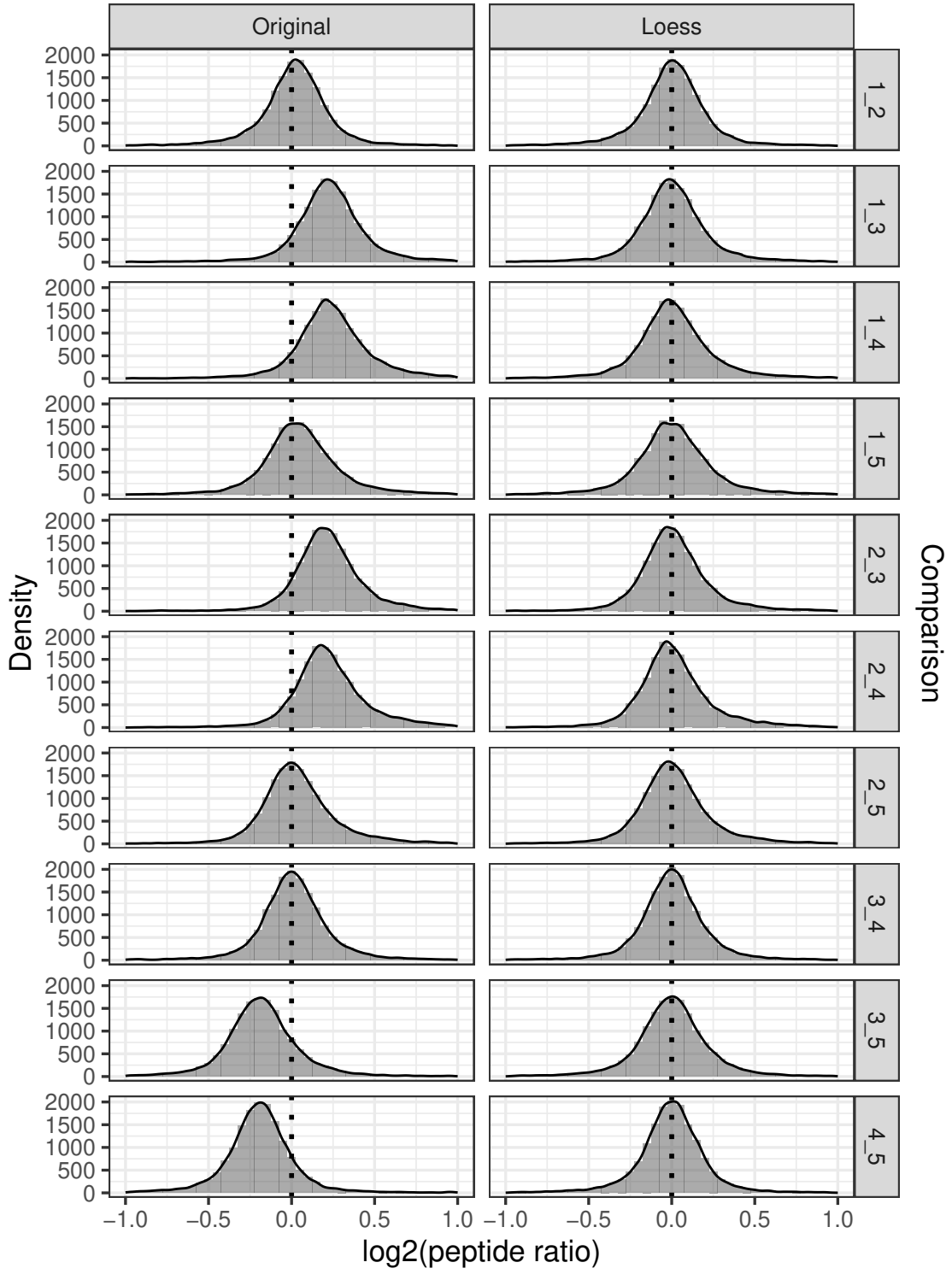


Figure 5.3: Histograms and density estimates of log2-transformed peptide ratios in data set D1 for the ten different pairwise condition comparisons. Original, non-normalized values are shown on the left and loess-normalized values on the right. The vertical dotted line describes the expected ratio of 1 (0 on log2-scale). For better visibility, the x-axes were cut at −1 an 1 on log2-scale.

background. Therefore, global differences between samples will largely be caused by technical variation.

Figure 5.3 shows histograms of peptide ratios before normalization and after loess normalization for data set D1. Without normalization for some pairwise condition comparisons, e.g. comparing condition 1 and 3 or 4 and 5, the distribution of peptide ratios shows a clear bias. Some other comparisons like 2 and 5 show much less technical bias. The loess normalization clearly improve the situation by shifting the distributions closer to the expected value of 0 on log2-scale. It can be concluded that the loess normalization works reasonable well for the peptide data of data set D1. For data set D2 the situation is similar the technical bias in the non-normalized data is not as severe as for D1, but still the loess normalization reduces the bias in the peptide-level quantitative data (see figure A.8 in the appendix, page 154).

In the data set D3, 60 µg HeLa cell lysate (human cells) were mixed with either 10 µg or 30 µg of *E. coli* lysate. Here the question is if the change in the overall proteome of these samples is too high for fulfilling the assumptions of a loess normalization. A problem with this data set is that the total amount of the samples is different (70 µg vs. 90 µg). This has an effect on the measured peptide intensities and will lead to peptide intensities that deviate from the expected ratios. As can be seen

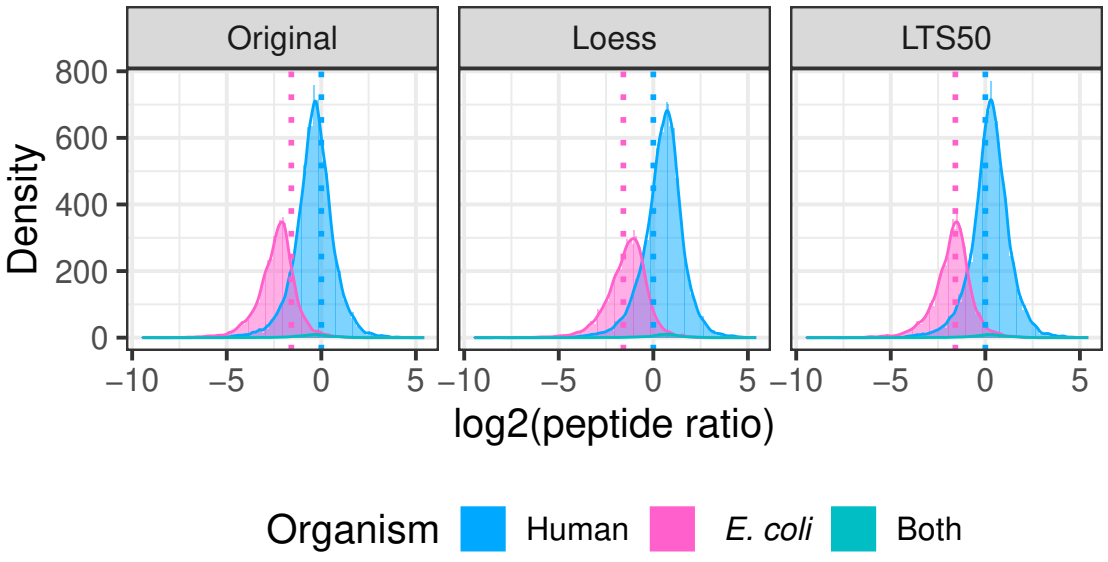


Figure 5.4: Histograms and density estimates of ratios of the human background peptides and the spiked-in *E. coli* peptides without normalization and after loess or LTS normalization for the data set D3. The expected ratios of 1 for the human peptides and $\frac{1}{3}$ for the *E. coli* peptides are shown as a dashed vertical line.

in figure 5.4, the distributions are shifted to the left for the non-normalized data. However, the loess normalization is not able to normalize the data properly, but overall shifts the data too far to the right. To overcome this issue the least trimmed sum of squares normalization (LTS) was applied (Huber et al., 2002). The LTS normalization is similar to a loess normalization, however, it can cope with a larger part of the data changing. The local regression line is not calculated by minimizing the error term over all samples, but only by a proportion λ of the data with the smallest error term. This makes the method robust against larger changes of the proteome between samples (Hahne et al., 2008, pp. 71-78). Here, the parameter λ was set to the minimal allowed value of 0.5. The LTS-normalization performs better than loess as is allows the violation of the stable proteome assumption, see figure 5.4 (p. 87). But still, while the *E. coli* peptides are distributed well around the expected value, the human background peptides are still shifted to the right. The difference between the two histograms is not as expected, most probably because of measurement effects caused by the fact that different amounts of total protein were measured. This is something that could only be fixed by re-measurement and a more appropriate design of the data set. However the data set is still used with the LTS normalization, as this one at least properly normalizes the *E. coli* peptides. For the interpretation of the resulting protein ratios the not 100% optimal normalization of the peptide ratios has to be considered.

For data set D4, the difference in total measured protein amount are not a problem as always 50 µg of total protein was measured and only the proportion of human and mouse proteins varies between the different conditions (Saltzman et al., 2018). However, there is no stable background proteome over all samples and when comparing two samples with each other, always a large part of the proteome changes. Therefore the normalization is an even greater challenge for this data set. Figure 5.5 (p. 89) shows the peptide ratio distribution for different normalization strategies. For the non-normalized data, the medians are not too far away from the expected ratios, however depending on the comparison there are small shifts to the right, e.g. for 010_050 or 010_090. The LTS normalization shows the best results, however, in some cases the distribution of the human peptides is still biased. This effect is the most prominent in comparisons that contain the samples of the 25% HeLa / 75% mouse group, which might be a sign of an issue with this specific underlying samples. In all comparisons, there are a lot of peptides that are shared between mouse and human (category "both"). These may serve as a stable background and it can be seen that the LTS normalization manages to shift this distribution close to a ratio

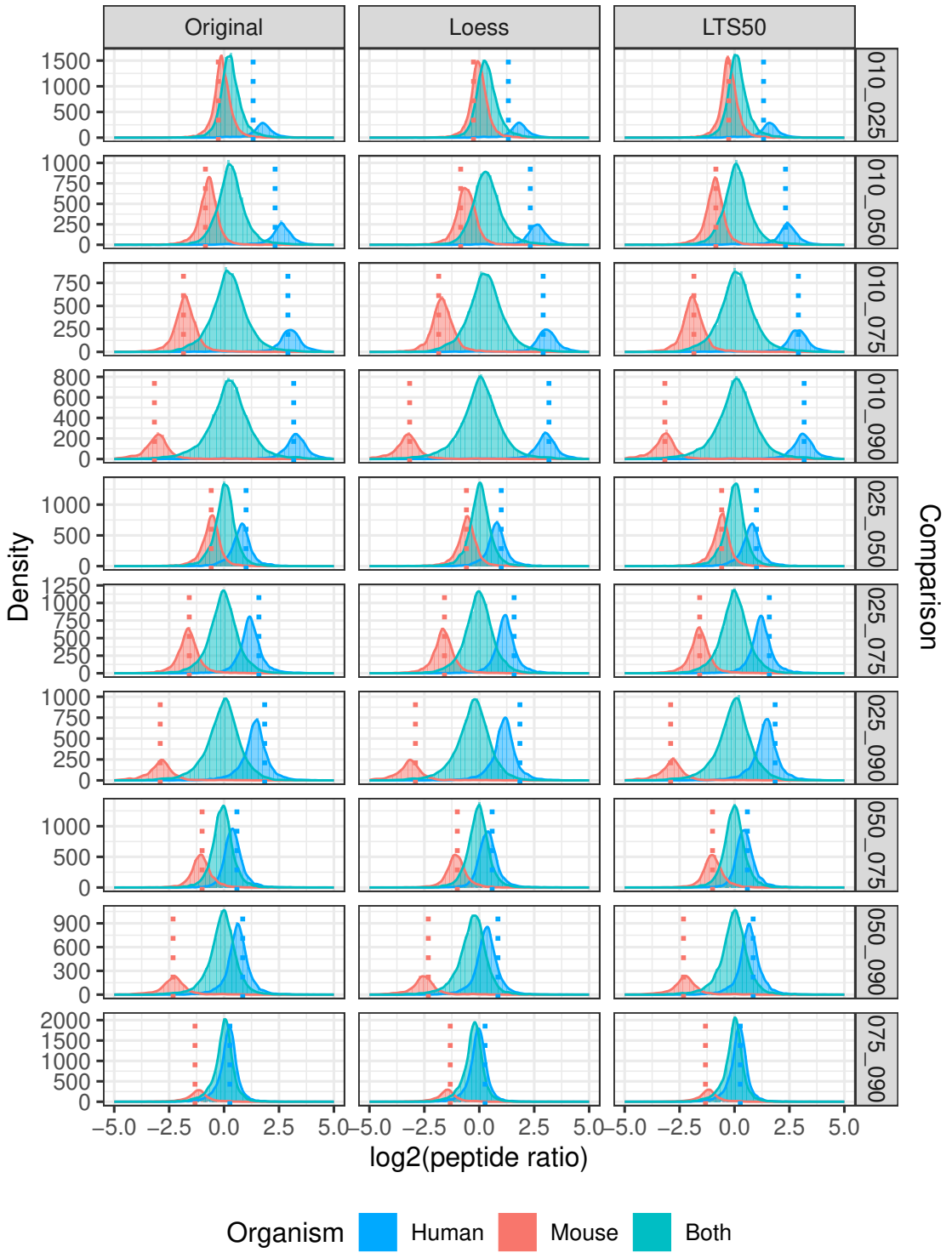


Figure 5.5: Histograms and density estimated of peptide ratios in data set D4, without normalization, with the loess normalization and LTS normalization. The expected ratios for human and murine peptides are shown as vertical dashed lines. For better visibility, the x-axes were cut at −5 an 5.

of 1 (0 on log2-scale). It was decided to continue with the LTS-normalized samples, although the normalization may not be perfect (see also the discussion chapter).

So in summary, the peptide intensities for D1 and D2 were normalized with the loess normalization, while D3 and D4 were normalized with the LTS normalization (with $\lambda = 0.5$) to account for larger changes in the proteome between samples.

5.3.3 Data preparation for bppgQuant, PQP and SCAMPI

For calculation of the bipartite peptide-protein graphs, a comprehensive edgelist is first generated, using all theoretical peptides from an in silico-digestion of the corresponding fasta files. Peptides outside of the desired peptide length (minimum 6,7 or 8 depending on the data set and maximum 50 for all) were removed. The technical replicates of the same sample (e.g., the same experimental condition, same amount of spike-ins) are aggregated by calculating the mean intensity for each peptide. This is only done if at least 60% of the samples (here in this case this usually means 2 out of 3 replicates) have a valid value, otherwise the mean intensity is treated as a missing value. Then, peptide ratios between all possible pairwise combinations of experimental conditions are calculated. If a peptide has missing values in at least one of the two compared samples, the ratio is treated as missing. The complete edgelist is now filtered for each comparison by peptides with valid peptide ratio. This edgelist is used to collapse protein nodes. In contrast to chapter 4, peptide nodes are not collapsed here, as all individual peptide nodes and the corresponding ratios are needed for the optimization problem of the new protein quantification method. The bipartite peptide-protein graphs are then generated from this edgelist, while the peptide ratios are added as node attributes to the corresponding peptide nodes. These graphs can directly be processed by the bppgQuant approach.

The PQP model was implemented by Dost et al. in C# and is provided as an executable file. The data has to be provided in a specific format as a `.txt` file, see table B.11 (appendix, p. 177). For each bipartite graph (called CC for connected component) the data starts with `>CC` and the graph number, followed by a row with number of protein nodes and peptides separated by comma. In the next line, the known peptide ratios are given separated by comma. The following lines consist of a representation of the adjacency matrix of the graph. The bipartite graphs and the structure of the collapsed protein nodes can be directly translated into this format.

For the peptide ratios, the reciprocal of the previously calculated peptide ratios was used, as PQP expects them in the form of A/B while B/A was calculated. This makes the correct interpretation of the results more straightforward. The PQP executable is then executed via the `system2()` R function. Except the input file and output file name no further arguments are necessary.

The output is again provided as a `.txt` file with a special format (see table B.12 in the appendix, p. 177). After the graph ID, the cost is given. This is the error term of the optimal solution. In the next two lines the estimated intensities of the protein nodes for sample B and A are given, respectively. Last, PQP recalculates the peptide ratios from the estimated protein ratios using their model formula (i.e., without the error term). This output scheme is further processed by calculating QB/QA to obtain the protein ratios and by mapping to the protein node names by using the graph ID. If $QB = QA = 0$, the protein ratio is treated as an `NA`. If $QB = 0$ and $QA \neq 0$, the protein ratio is set to the minimum of obtained protein ratios for the specific dataset and for $QA = 0$ and $QB \neq 0$ the maximum is used.

SCAMPI is implemented in the R package `protiq`. The input data have to be prepared in a certain format. SCAMPI requires three data frames as an input, one containing peptide IDs, sequences and intensities, one containing protein IDs and accessions and one containing the edge list of the graphs. Those three dataframes are combined in a scampi object and run trough the `checkInputData()` function to check for validity of the input and perform log10 transformation.

SCAMPI calculates on peptide intensities of individual samples instead of peptide ratios between conditions. Therefore, a SCAMPI run for each individual sample is calculated and the SCAMPI output is obtained. The output contains the estimations for the model parameters, α, β, μ and τ as well as estimated score for the proteins (compare to equation 3.16, page 24). To calculate protein ratios, the scores are averaged over the replicates of the experimental conditions and the means are subtracted from each other for the different comparisons between two conditions. This leads to estimates of the protein ratios on log10 scale. In SCAMPI, two possibilities to estimate the model parameters are implemented, MLE and ILSE, which produce similar results in most cases. However, the behaviour of ILSE shows higher robustness (Gerster et al., 2014), so we chose to focus only on these results.

5.4 Results of a method comparison on test data sets

The four data sets D1, D2, D3 and D4 differ in their type and number of spike-in proteins. For the data sets D1 and D2 only a few proteins (13 and 48, respectively) were spiked into a stable background proteome (mouse and yeast cells, respectively). The background proteins are expected to have a ratio distributed around 1, whereas the spike-in proteins have different defined expected ratios depending on the concentrations that were spiked in. For the data sets D3 and D4, two proteomes (human and *E. coli* or human and mouse) were mixed in different ratios. While the human background proteins still have an expected ratio around 1 for D3, for D4 no stable background over the different experimental conditions exists. Because of the different characteristics of the four data sets, in the following, the results are presented in different subsections. An overview over the number of quantified protein groups for the different methods is given in section 5.4.1. The results for the data sets D1 and D2 are split into section 5.4.2 for the background proteins and 5.4.3 for the spike-in proteins. The results for D3 and D4 are shown in section 5.4.4, followed by a summary and consideration of the range solutions in section 5.4.5. Finally, the results of bppgQuant after protein selection (as described in section 5.1.4) are shown in section 5.4.6.

In the following the proposed bppgQuant method is compared with SCAMPI (Gerster et al., 2014) and PQP (Dost et al., 2009; Dost et al., 2012). In section 3.6 (p. 33) those were identified as the only two methods that mention the possibility to quantify proteins without shared peptides, which is also in the focus of bppgQuant.

For bppgQuant, especially for proteins with only shared peptides, the underlying equation system may not have a single solution with a minimal error term, but a whole range of possible solutions, which will be reported as an interval with a lower and upper border. Some of these intervals are extremely large, because the lower border is close to a ratio of zero. As corresponding large intervals on the other side of 0 (with an upper border going to infinity) do not exist, this may be a sign of numerical problems during optimization, for example due to contradictions between peptide ratios. These large intervals are also hardly usable for biological interpretation later on, this is why they were removed from the results. To determine the threshold for cutting off the results, the interval length was plotted against the interval centers, see figure A.9 in the appendix (p. 155). Interestingly, for all four data sets the interval length for intervals with a positive interval center on log

scale stops at a value around 14-15, while for negative intervals also larger lengths are possible. This asymmetry indicates that these large intervals are computational artifacts (likely due to contradictions in the equations). Therefore, interval solutions with a length higher than 15 on log2-scale were removed, to ensure to reduction of these artifacts while not removing valuable interval solutions.

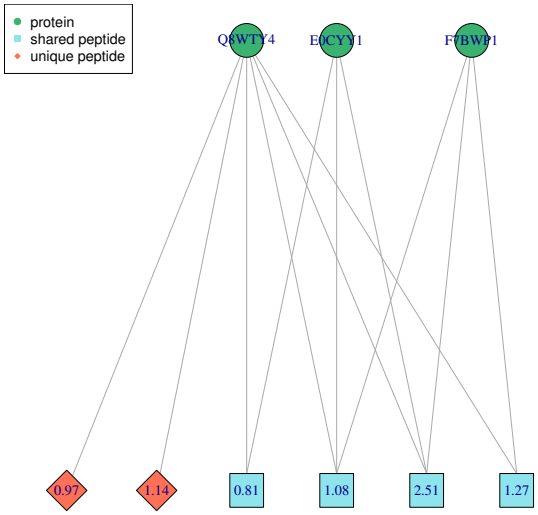


Figure 5.6: Example of a bipartite peptide-protein graph with one protein ratio estimated as 0. The peptide nodes are annotated with their respective measured peptide ratios (rounded). BppgQuant estimates the following protein ratios (rounded): Q8WTY4: 1.05, E0CYY1: 0.00, F7BWP1: 1.51.

A similar phenomenon occurs for the point estimates for the bppgQuant method. In the current version of bppgQuant, for some protein nodes an extreme ratio of zero or close to zero is estimated. As bppgQuant currently does not handle on/off proteins and peptides (see also the discussion in section 5.5, p. 118) and such extreme ratios do not occur in the other direction (meaning very large ratios going to infinity), this is likely again a numerical artifact from the optimization step. Figure 5.6 shows an example of a bipartite peptide-protein graph where one protein node ratio is estimated as close to zero. The protein Q8WTY4 has two unique peptide and its estimated ratio is 1.05, which is the geometric mean of the two unique peptide ratios. This protein contains also all four present shared peptides. F7BWP1 is connected to three of the shared peptides and a protein ratio of 1.51 is estimated. The third protein, E0CYY1 has three shared peptide sequences with a high variability in their

peptide ratios: $0.81, 1.08$ and 2.51. The last one seems to be an outlier, which causes a contradiction for this protein node. Derived from the optimization problem in equation 5.13 (p. 79), the protein ratio needs to be smaller than 1.05 (the already estimated ratio of protein Q8WTY4) to have a small error term for the peptide node with ratio 0.81. On the other hand, it needs to be larger than the ratio of F7BWP1 (1.51) to be closer to the outlying peptide ratio of 2.51. Also other examples show that estimated peptide ratios close to zero may be a hint for contradictions or outliers in the peptide ratios. For the method comparisons, estimated protein ratios below 2^{-15} were set to `NA` (threshold chosen to be comparable to the one for the range solutions). It has to be noted that the omission of small estimated protein ratios or the large intervals does not influence the quantification of spike-in proteins as the most extreme expected ratios are $\frac{1}{1000}$ (data set D2).

5.4.1 Number of quantified protein nodes

The protein quantification methods bppgQuant, PQP and SCAMPI were applied to all bipartite peptide-protein graphs that were constructed from the peptide-level quantitative data sets D1, D2, D3 and D4 that were introduced in section 5.3 (p. 83). As a result, estimates for the unknown protein ratios are obtained.

In a bipartite peptide-protein graph, three different types of protein nodes can be distinguished based on the peptide nodes they are connected with: protein nodes with only unique peptides, with only shared peptides and with unique and shared peptides. During protein quantification, protein nodes with unique peptides are easier to handle, because the unique peptide ratios are only influenced by the corresponding protein. In contrast, protein nodes with only shared peptides are hard to quantify. Three types of estimates are possible for each protein node: a single solution, a range solution (only for bppgQuant) or no solution (error during calculation or `NA` as output).

Figure 5.7 (p. 95) shows the relative frequency of the different solution types for the different protein node types, data sets and quantification methods. The absolute numbers for this can be found in table B.13 in the appendix (p. 178). These are summed up numbers over all pairwise comparisons of the experimental conditions. The absolute numbers are not comparable between data sets as each data set has a different number of conditions and is based on different organisms.

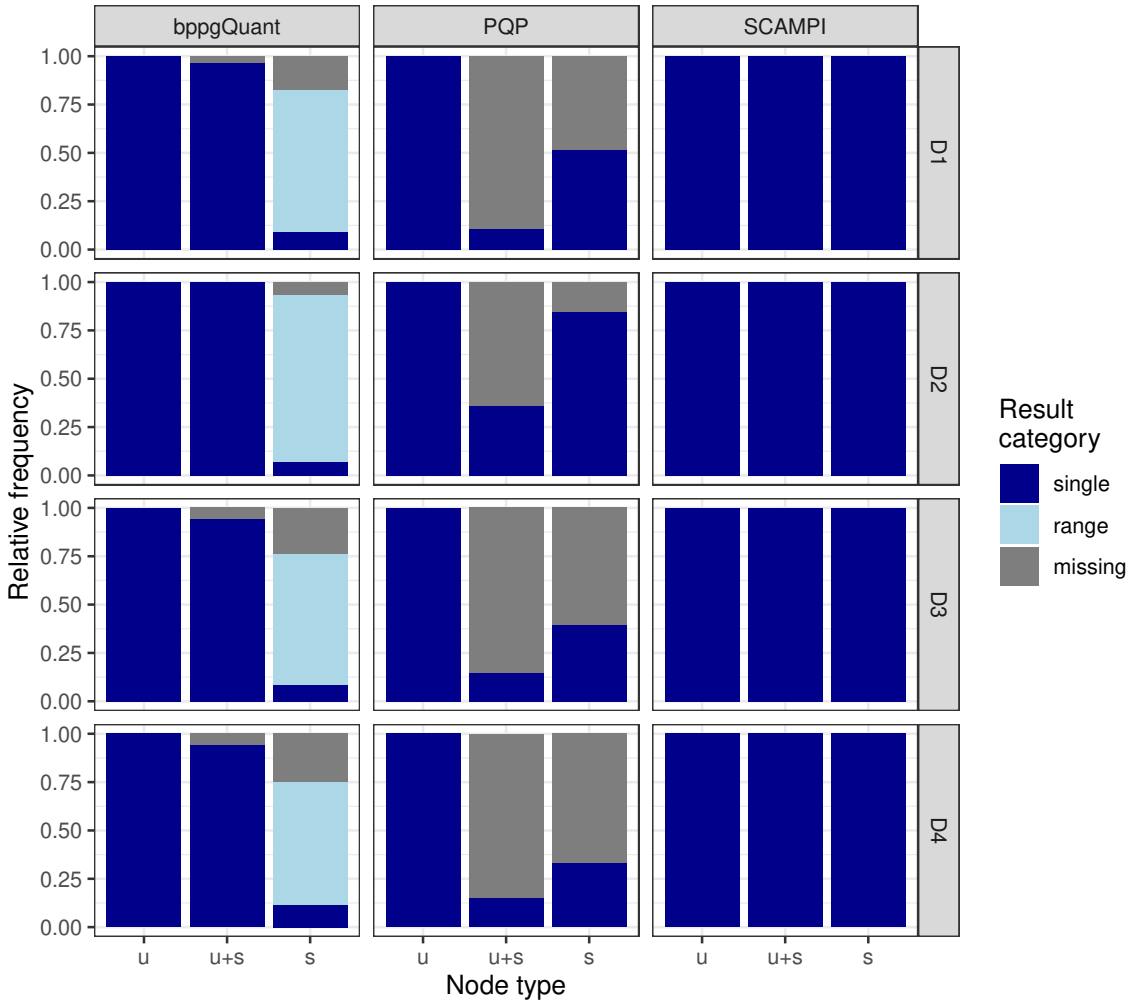


Figure 5.7: Barplots of relative frequencies of solution types for the different protein node types, data sets and protein quantification methods. u: only unique peptides, u+s: unique and shared peptides, s: only shared peptides.

Protein nodes with only unique peptides can always be quantified by all methods in all data sets. This is often a trivial case. For example in bppgQuant this case simplifies to the calculation of the geometric mean of the unique peptide ratios. For protein nodes with unique and shared peptides, bppgQuant obtains a single estimate for the majority of cases (94.1% - 100.0% depending on the data set). For each bipartite peptide-protein graph the calculation was stopped after three hours, leading to 0-5.9% of missing solutions for this protein node type. For this node type bppgQuant only computes single solutions, except for nine protein nodes in data set D4, which obtain a range solution (0.02%). In contrast, protein nodes with only shared peptides lead to a range solution in most cases (63.5-86.5%). The proportion of missing solutions (between 6.6% and 25.0%) is higher than for the nodes with

unique and shared peptides. Differences between the four data sets can be observed. While there are only a few missing solutions for bppgQuant on the data set D2, this proportion becomes higher in the more complex data sets D3 and D4. In those, also the proportion of single solutions for protein nodes without unique peptides is higher (8.5% and 11.4%) compared to D2 (below 1%).

The PQP method has a high proportion of missing solutions for protein nodes with only shared (up to 67.2% for D4) or unique and shared peptides (up to 89.3% for D1). Missing solutions for protein ratios occur for this method when both protein intensities for a pairwise comparison are estimated as zero. Interestingly, the amount of missing solutions is much higher in protein nodes with unique and shared peptides, although they should be easier to quantify than nodes with only shared peptides. PQP seems to have difficulties quantifying these kind of protein nodes. Depending on the data set only 10.7-35.7% of these nodes are quantified, while they are not much of a problem for bppgQuant or SCAMPI. Again the influence of the data set can be observed, as in D2 a higher proportion of proteins could be quantified. On the other hand, SCAMPI produces solutions for all existing protein nodes, completely independent of the presence of unique or shared peptides.

5.4.2 Results for the background proteome for data sets D1 and D2

In this section the quantification results for the background proteins of the data sets D1 and D2 are described. For these proteins a ratio of 1 (0 on log2-scale) is expected, however this is not explicitly controlled for every single protein. It is expected that the biological variability of the cells have an influence on this additionally to the expected technical variability of the measurements.

In figure 5.8 (p. 97) histograms with density estimation of all single solutions (no range solutions for bppgQuant) for the protein ratios for the background mouse proteins in data set D1 over all 10 comparisons are shown, split by the sharedness/uniqueness of their corresponding peptides. Corresponding medians, MADs (median absolute deviation from the expected value) and total number can be found in table 5.1. The robust measures median and MAD were chosen to limit the influence of outliers in the protein ratio estimation.

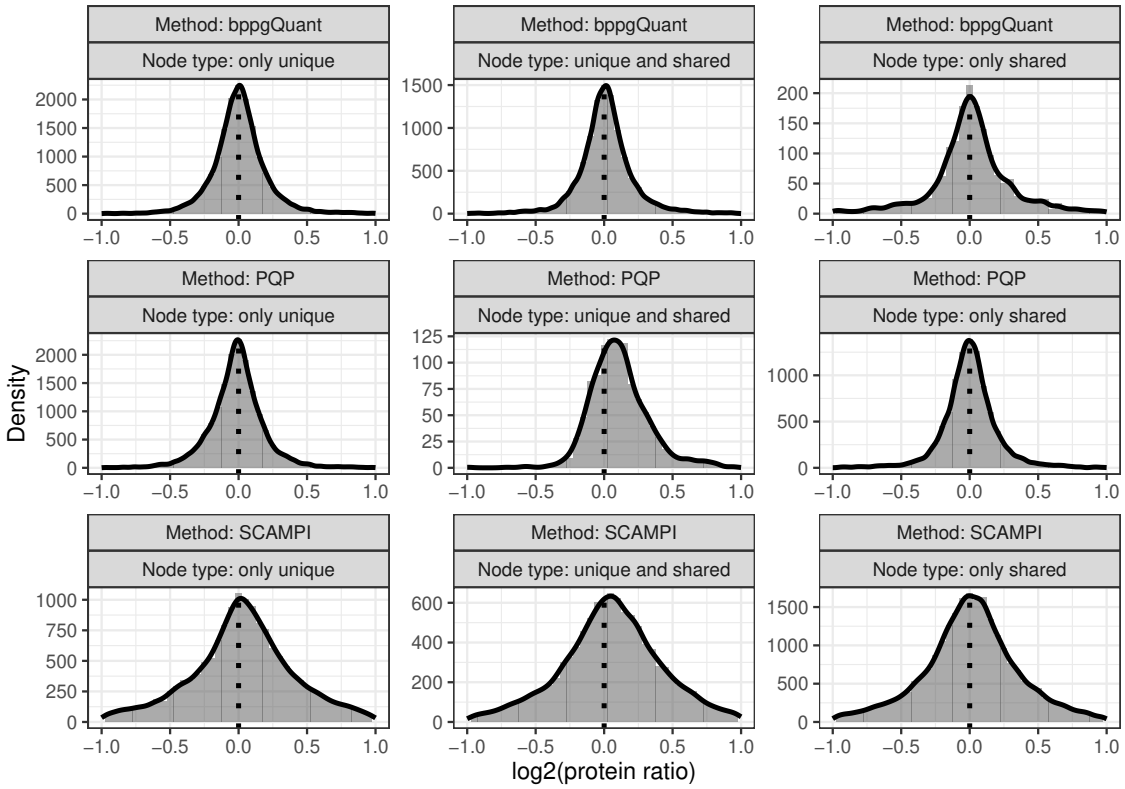


Figure 5.8: Histograms and density plots for the estimated protein ratios for the background proteins in data set D1 summarized over all 10 comparisons. Only the murine background proteins (no spike-ins or contaminants) are shown here. Only the single solutions (no range solutions) are visualized here. The x-axes are cut at -1 and 1 for better visibility. Please consider the different ranges of the y-axes.

For proteins with only unique peptide nodes, the estimates of the three methods are symmetrically distributed around the expected log-ratio of zero. The medians are close to zero, while bppgQuant has the one nearest to zero with 0.0022. The distributions of bppgQuant and PQP are more narrow than SCAMPI, reflected by a more than doubled median absolute deviation from the expected values of zero (MAD) for SCAMPI (0.3591 compared to around 0.1476). For proteins with unique and shared peptides, PQP shows an overestimation of the ratios which can be seen in the histograms and by the bias in the medians. SCAMPI also has a much larger median than bppgQuant with 0.0571 and a small shift to the right is also visible in the histogram. For bppgQuant the distribution is still around zero with a MAD similar to the proteins ratios with only unique peptides (around 0.1415). However, it has to be noted that PQP has a problem with quantifying this type of nodes and

Table 5.1: Median, MAD (median absolute difference from the expected ratio) and n (number of quantified protein nodes) of estimated log2 protein ratios for data set D1. The best median and MAD values per node type are printed in bold.

Method	Node type	Median	MAD	n
bppgQuant	only unique	**0.0022**	**0.1476**	15,709
PQP	only unique	-0.0108	0.1511	15,709
SCAMPI	only unique	0.0317	0.3591	15,709
bppgQuant	unique and shared	**0.0091**	**0.1415**	10,038
PQP	unique and shared	0.0920	0.1976	1,112
SCAMPI	unique and shared	0.0571	0.3562	10,389
bppgQuant	only shared	**0.0104**	0.2873	2,173
PQP	only shared	-0.0226	**0.2135**	12,500
SCAMPI	only shared	0.0117	0.3119	24,232

quantifies 90% less protein nodes compared to SCAMPI or bppgQuant (see table 5.1).

The distribution for proteins with only shared peptides for bppgQuant is broader and contains heavier tails compared to the other node types. More ratios have a higher distance to the expected zero, which leads to the higher MAD value of 0.2873. However this behaviour is expected as quantifying protein nodes without any unique peptide is the hardest case and it is a good result to even obtain single solutions (in contrast to a range solution) in this case. The SCAMPI results show a similarly large MAD and the distribution looks extremely similar to the other protein node types. In this case, PQP shows the best distribution with a lower MAD of 0.2135, while bppgQuant's median is closest to the expected value of 0 with 0.0104. However, here for bppgQuant only the single solution type is incorporated into the calculation of the MAD and median, which explains the small n compared to the other methods.

For data set D2 the overall situation is similar to D1, see figure 5.9 (p. 99) and table 5.2 (p. 100). For the methods bppgQuant and PQP the distributions are narrow around zero. In D2, there are overall less shared peptides and less proteins without unique peptides (see also figure 4.5, p. 60). Also because bppgQuant has a large proportion of range solutions for the protein nodes without unique peptides, the corresponding histogram represents less than 10% of protein nodes than for PQP or SCAMPI, making it appear less smooth. For SCAMPI the distributions are much wider for all protein node types (also wider than for D1, as here the x-axis is from -2 to 2 compared to -1 to 1 for D1.), which is also represented by much larger MADs compared to the other two methods. For the protein nodes without unique

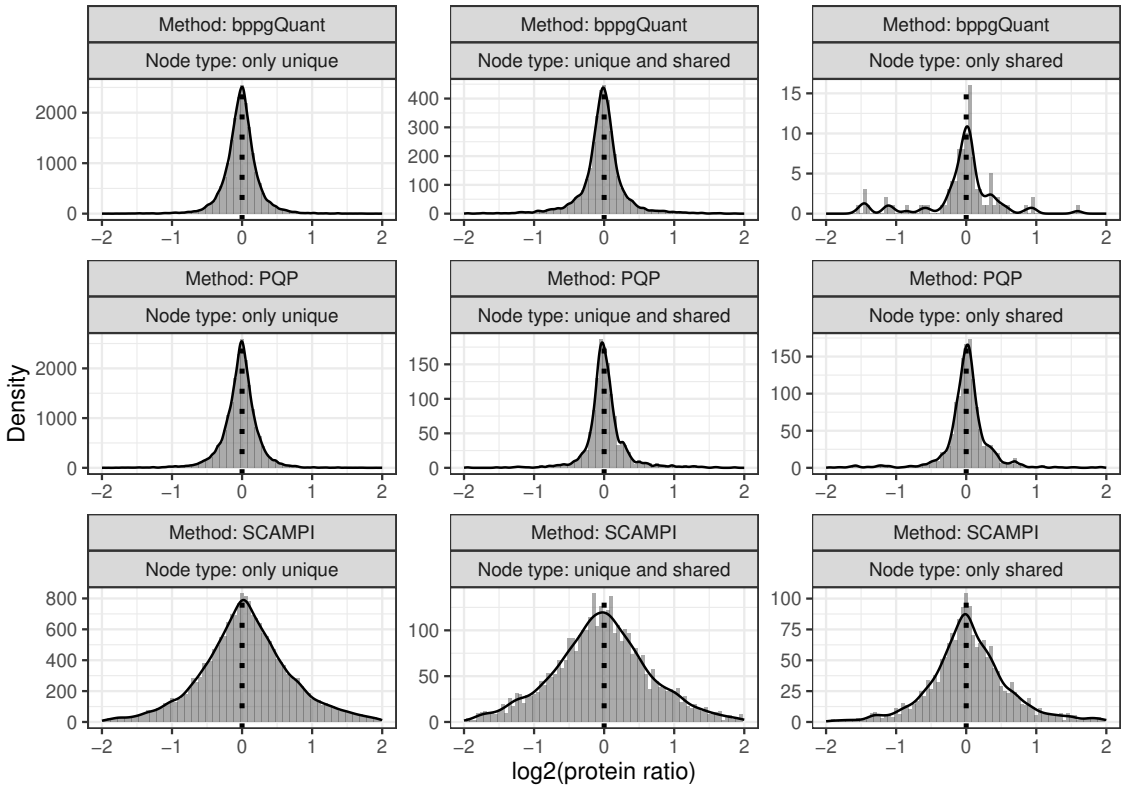


Figure 5.9: Histograms and density plots for the estimated protein ratios for the background proteins in data set D2 summarized over all 10 comparisons. Only the yeast background proteins (no spike-ins or contaminants) are shown here. Only the single solutions (no range solutions) are visualized here. The x-axes are cut at -2 and 2 for better visibility. Please consider the different ranges of the y-axes.

peptides, PQP shows the best fitting distribution for the estimates with a median of 0.0013 and an MAD of 0.2063.

5.4.3 Results for spike-in proteins in data sets D1 and D2

In data set D1 13 non-mouse proteins were spiked into a mouse cell background in varying amounts, resulting in five conditions and ten pairwise comparisons of these conditions, see table B.10 (appendix, p. 176). It has to be noted that some proteins are more prone to missing peptide intensities if they were spiked into the sample in low amounts (missing not at random because of the individual lower quantification limit (Lazar et al., 2016)). Therefore, some of the spike-in proteins may not have quantified peptides in some of the five experimental conditions and therefore no protein ratio can be calculated by any of the three quantification methods (see figure

Table 5.2: Median, MAD (median absolute difference from the expected ratio) and n (number of quantified protein nodes) of estimated log2 protein ratios for data set D2. The best median and MAD values per node type are printed in bold.

Method	Node type	Median	MAD	n
bppgQuant	only unique	**-0.0118**	**0.1719**	20,240
PQP	only unique	-0.0139	0.1751	20,240
SCAMPI	only unique	0.0406	0.5647	20,240
bppgQuant	unique and shared	-0.0152	0.1766	3,660
PQP	unique and shared	**-0.0075**	**0.1473**	1,307
SCAMPI	unique and shared	-0.0308	0.6469	3,660
bppgQuant	only shared	0.0412	0.3371	127
PQP	only shared	**0.0013**	**0.2063**	1,518
SCAMPI	only shared	0.0209	0.4580	1,809

A.10 in the appendix, p. 156). As was already shown in figure 5.7 (p. 95), SCAMPI quantifies all protein nodes that are present in the graphs, so all proteins with at least one assigned peptide. Only in the comparison of condition 4 and 5 all 13 spiked-in proteins have at least one quantified peptide and can be quantified by SCAMPI. The least number of quantifiable spike-in proteins is in the comparison of condition 2 and condition 4 and condition 1 with 2 with only eight proteins. BppgQuant is able to quantify most of the quantifiable spike-in proteins too, in most comparisons it quantifies one protein less than SCAMPI. In contrast, PQP only quantifies between one and seven proteins and in most cases less than half of the proteins that the other methods can quantify. More than half of the protein nodes belonging to spike-in proteins have unique and shared peptides, a node category that is difficult to handle for PQP, as was already observed earlier.

The estimated and expected protein ratios for the different quantification methods are compared in figure 5.10 (p. 101). All methods show a correlation with the expected ratios (Pearson correlation coefficient for bppgQuant: 0.989, for PQP: 0.986 and for SCAMPI: 0.901). However, SCAMPI has a much larger variance in the estimates than bppgQuant or PQP, reflected also by the lower Pearson correlation. The mean absolute deviation (MAD) for the spike-in proteins over all comparisons is the smallest for bppqQuant with 0.4829, directly followed by PQP with 0.5161. SCAMPI has by far the worst MAD with 2.6800. All three methods show a systematic underestimation of log-ratios smaller than zero and an overestimation of ratios larger than zero. This phenomenon can already be seen on the corresponding peptide level ratios in the original publication for the DDA data (Barkovits et al., 2020).

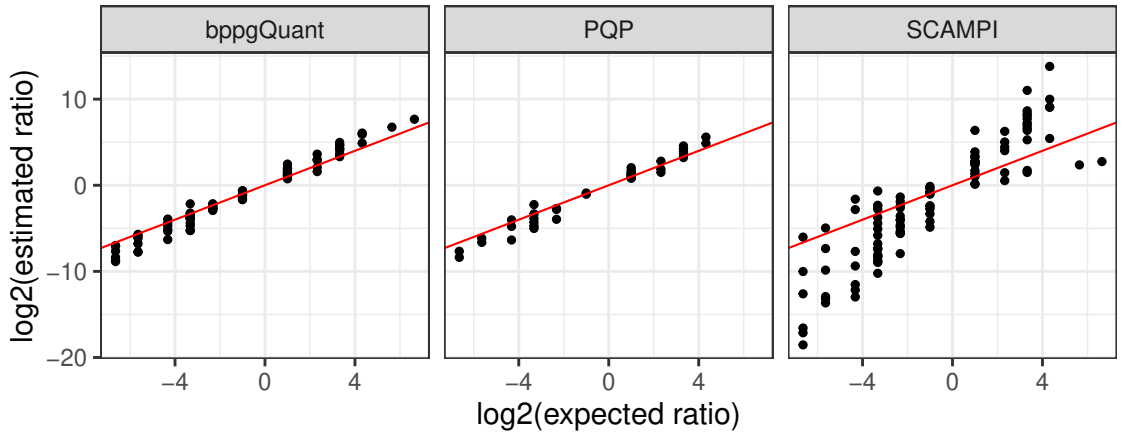


Figure 5.10: Scatter plot of expected versus estimated protein ratios for the spike-in proteins in date set D1 for bppgQuant, PQP and SCAMPI. The red line represents the angle bisector.

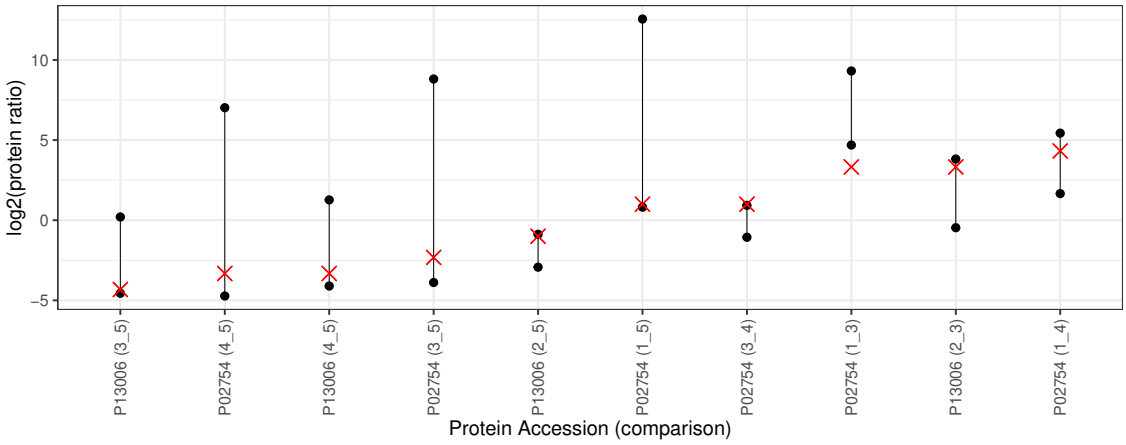


Figure 5.11: Range solutions for the spike-in proteins in data set D1 from the bppgQuant method. The red crosses mark the expected protein ratio.

For two spike-in proteins in some comparisons there is a range solution calculated by the bppgQuant method, namely P13006 and P02754 (see figure 5.11). It can be seen that the results are mostly in line with the expected ratios. Either the intervals cover the expected ratio or one of the interval borders is relatively close to the expected ratio.

In data set D2 the UPS1 standard mixture with 48 proteins was mixed into a stable yeast proteome in varying concentrations. In contrast to data set D1, where the different spike-in proteins can have different concentrations within the same condition, the UPS1 mix contains all proteins in equimolar amounts and is spiked into the sample as a whole. Figure A.11 (appendix, p. 156) shows the number of protein

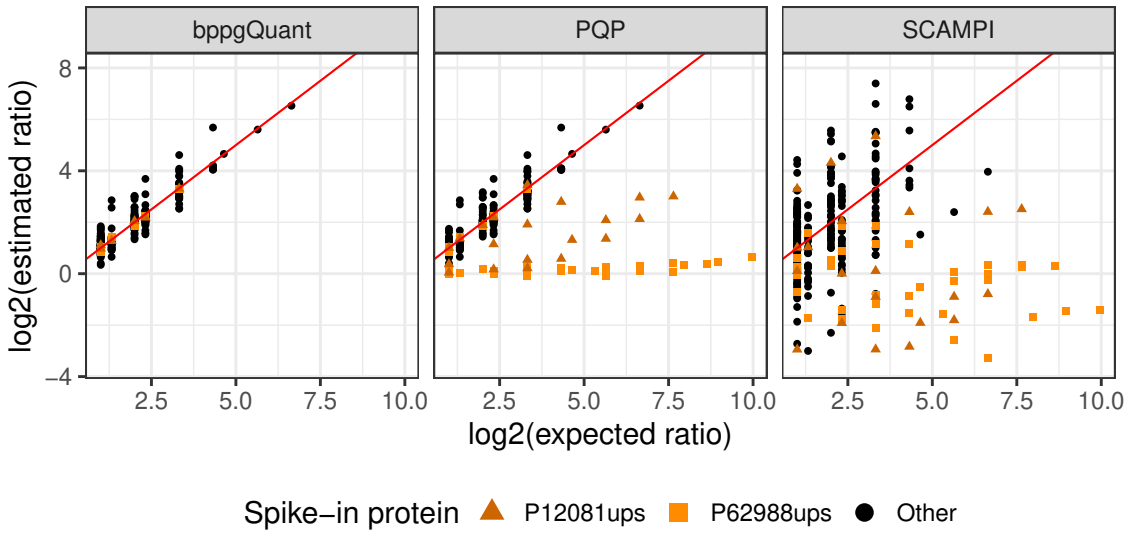


Figure 5.12: Scatter plot of expected versus estimated protein ratios for the spike-in proteins in date set D2. The red line represents the angle bisector.

ratios estimated for each comparison and method. It is noticeable that for comparisons with at least one of the low concentrations, the number of quantified UPS proteins is extremely low. For all comparisons with 50 amol/µg and 125 amol/µg as the smaller concentration only one or none of the UPS proteins is quantified in each method. For 250 amol/µg it is up to two, for 500 amol/µg up to three and for 2500 amol/µg up to nine. For the comparison of the two largest concentrations, 25,000 and 50,000 amol/µl, 47 out of 48 UPS proteins could be quantified. The reason for this behaviour is that the lower concentrations seem to lie below the detection limit for most of the spike-in peptides. This has also been observed in the original publication of the data set (Ramus et al., 2016b). In most cases, bppgQuant and SCAMPI quantify the same number of spike-in protein per comparison and PQP a bit less. The lower amount of protein nodes for PQP however is not as severe as for data set D1. This could be explained by the fact that almost 75% of the spike-in proteins in D2 have only unique peptides (the easiest case for quantification), while this percentage was considerably lower in data set D1 with 32%.

Figure 5.12 shows the relationship between expected and estimated ratios for the spike-in proteins for the different methods for data set D2. There is more variability in the estimates for bppgQuant than for data set D1, however the estimated ratios correlate well with the expected ratios (Pearson correlation: 0.937). For PQP the situation looks similar, except outliers with greatly underestimated ratios, causing the Pearson correlation to drop to 0.239. SCAMPI also contains such outliers and

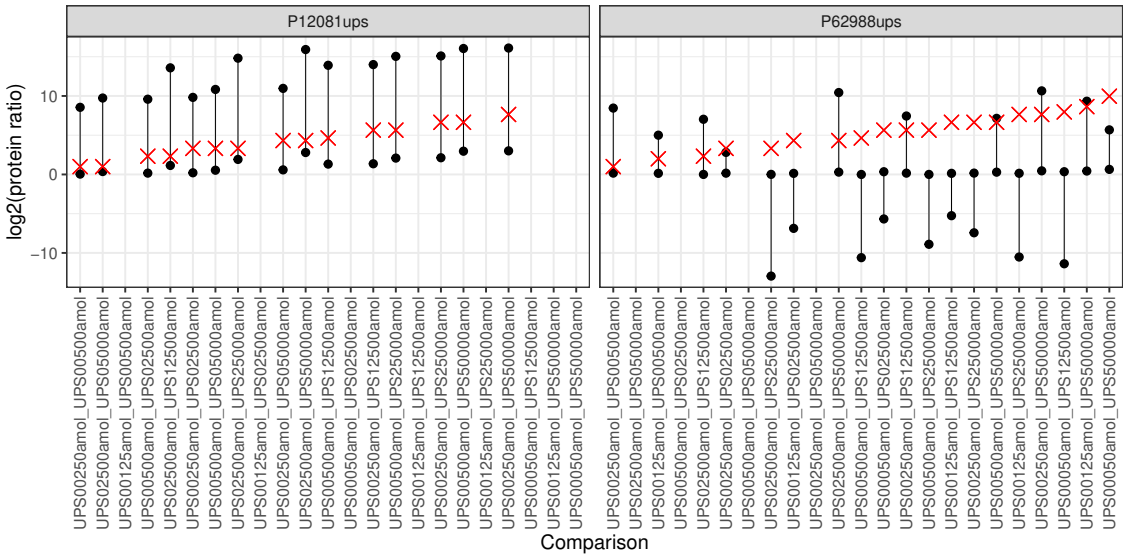
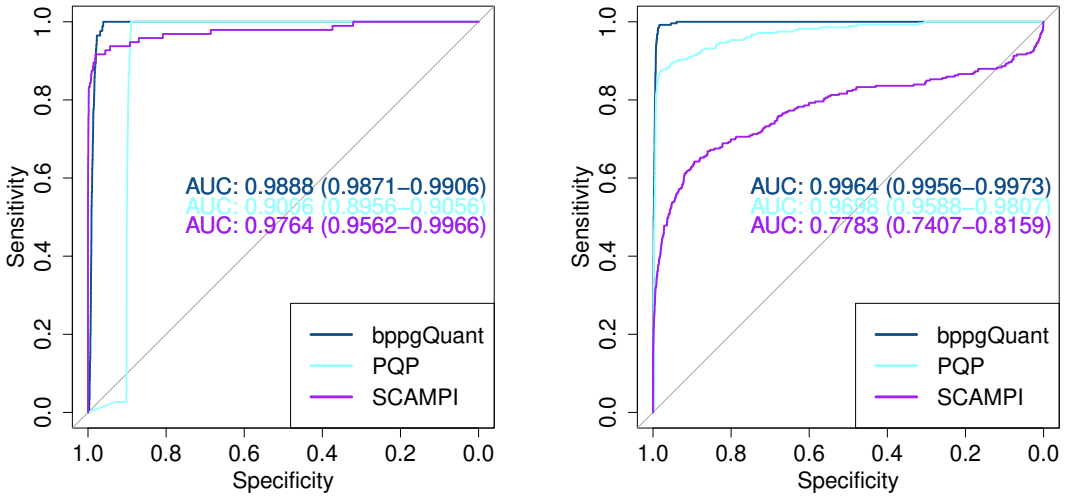


Figure 5.13: Range solutions for data set D2 from the bppgQuant method for the spike-in proteins P12081ups and P62988ups. The red crosses represent the expected protein ratio.

shows a large variability also for ratios of the other spike-in proteins. A lot of estimates are even below zero on log2-scale, while the lowest expected ratio was one on log2-scale. The Pearson correlation is therefore extremely low with a value of -0.027, indicating no linear relationship between estimated and expected protein ratios. The outliers for PQP and SCAMPI mostly from two proteins, P62988ups and P12081ups. These proteins were quantified with a range solution by bppgQuant, that is why there are missing in this figure. Overall, bppgQuant again has the smallest MAD with 0.1499, followed by PQP with 0.2240 and SCAMPI with 1.0151.

Figure 5.13 shows the range solutions for the two spike-in proteins P12081ups and P62988ups for the bppgQuant method. While the estimated intervals include the expected protein ratio in 100% of the cases for P12081ups, this is only the case in 42.1% for P62988ups. For some comparisons, the interval largely underestimates the protein ratios. This protein has many missing peptide intensities for low UPS concentrations, which may be the cause of this wrong estimation. This problem will be addressed further in the discussion section 5.5 (p. 118).

Additionally it is also interesting if the spike-in proteins can be distinguished from the background proteins using the estimated protein ratios. Receiver Operating Characteristics curves (ROC-curves) are calculated for separating the two classes ”spike-in” and ”not spike-in”. Contaminants were removed before the calculation of



(a) Data set D1. **(b)** Data set D2.

Figure 5.14: ROC curves and AUC values (with confidence interval) for distinguishing spike-in from background proteins in data sets D1 (a) and D2 (b) using the estimated protein ratios from bppgQuant, PQP and SCAMPI. Contaminants (also spike-in contaminants for D1) were removed before calculation. Only single solutions (no range solutions) regardless of the node type were used for the calculation. For D1, absolute values of the log2-transformed protein ratios were used.

the ROC curves and the corresponding area under the curve (AUC) values. For data set D1, the expected ratios of the spike-in proteins are smaller or larger than 1. To calculate the ROC-curves therefore the absolute values of the log2-transformed estimated ratios are used, as the spike-in ratios are expected to be further away from 1 than those of the background proteins.

Figure 5.14 show the ROC curves for data set D1 and D2, the ratios for all pairwise condition comparisons were analyzed together. For data set D1, all three methods show a very good AUC value of over 0.9. BppQuant shows the best AUC value with 0.9888, followed by SCAMPI with 0.09764, while PQP shows the worst with 0.9006. For data set D2, bppgQuant and PQP again show high AUC values with 0.9964 and 0.9698, respectively, while SCAMPI has the worst AUC with 0.7783.

5.4.4 Results for data sets D3 and D4

The data sets D3 and D4 are different from D1 and D2 as they do not have a few spike-in proteins but whole proteomes of two different organisms, that are mixed in defined ratios.

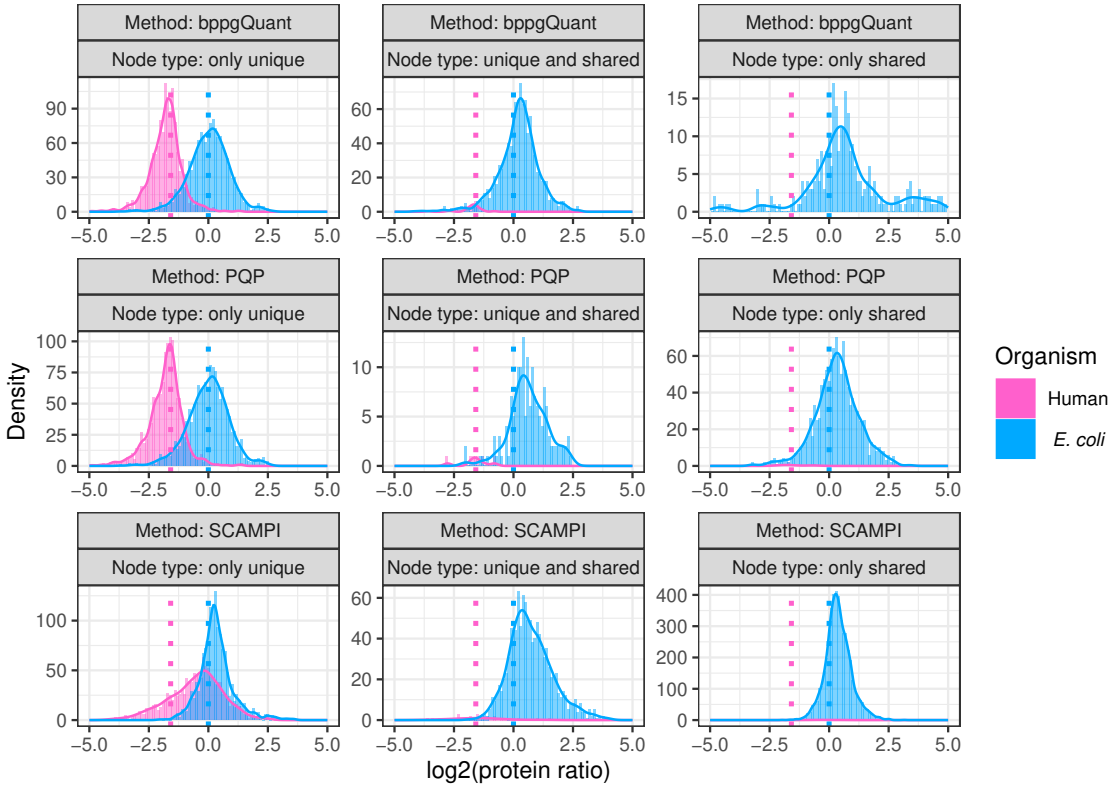


Figure 5.15: Histograms and density plots for the estimated protein ratios in data set D3. Contaminant proteins were removed. Only the single solutions (no range solutions) are visualized here. The x-axes are cut at -5 and 5 for better visibility. Please consider the different ranges of the y-axes.

In data set D3, 60 µg of human proteins were mixed with 10 or 30 µg of *E. coli* proteins. In theory, when comparing the two experimental conditions, protein ratios of 1 (0 on log2-scale) for the human proteins and $\frac{1}{3}$ (-1.0986 on log-scale) for the *E. coli* proteins would be expected.

Histograms of the estimated protein ratios for the data set D3, for the human and *E. coli* peptides are shown in figure 5.15 with corresponding median and MAD values in table 5.3 (p. 106). For the protein nodes with only unique peptides, PQP has a median of 0.0587 which fits best to the expected value, closely followed by bppgQuant. However, SCAMPI has the lowest MAD with 0.6235 caused by a

Table 5.3: Median, MAD and total number of quantified proteins (n) for the different organisms (Org.), methods and node types (NT, u = only unique, u+s = unique and shared, s = only shared) for data set D3. The expected ratio for human and *E. coli* proteins are shown in column "ER". The best median and MAD values per organism/node type are printed in bold.

Org.	Method	NT	ER	Median	MAD	n
Human	bppgQuant	u	0.0000	0.0764	0.7752	1,405
Human	PQP	u	0.0000	**0.0587**	0.7921	1,405
Human	SCAMPI	u	0.0000	0.2781	**0.6235**	1,405
Human	bppgQuant	u+s	0.0000	**0.2647**	**0.7465**	1,043
Human	PQP	u+s	0.0000	0.5715	0.9683	159
Human	SCAMPI	u+s	0.0000	0.6142	1.0437	1,096
Human	bppgQuant	s	0.0000	0.6531	2.2268	386
Human	PQP	s	0.0000	**0.1607**	1.5991	1,787
Human	SCAMPI	s	0.0000	0.3499	**0.6222**	4,507
E. coli	bppgQuant	u	-1.0986	-1.7514	**0.5540**	1,250
E. coli	PQP	u	-1.0986	**-1.7181**	0.5682	1,250
E. coli	SCAMPI	u	-1.0986	-0.3953	1.9220	1,250
E. coli	bppgQuant	u+s	-1.0986	-1.7012	**0.2386**	28
E. coli	PQP	u+s	-1.0986	-1.5842	0.3303	7
E. coli	SCAMPI	u+s	-1.0986	**-1.5248**	1.0485	28
E. coli	PQP	s	-1.0986	**-1.7404**	**0.6727**	8
E. coli	SCAMPI	s	-1.0986	-0.2555	1.9710	9

narrow but shifted distribution (median value of 0.2781). For the *E. coli* proteins, the distributions of bppgQuant and PQP are shifted to the left of the expected value of -1.0986, while SCAMPI has a wide distribution shifted to the right. For the node types with unique and shared or only shared peptides, only a small number of *E. coli* peptides exists, as the number of shared peptides is low in general here. For human protein nodes with unique and shared peptides all distributions are shifted to the right, while bppgQuant shows by far the best median with 0.2647 and the lowest MAD with 0.7465. For the *E. coli* protein nodes, SCAMPI has the best median and bppgQuant the lowest MAD, however, these numbers are only based on 28 protein nodes. For human protein nodes with only shared peptides, all three methods again show a distribution shifted to the right of the expected ratio. Here, PQP has the best median and SCAMPI with the narrow distribution the lowest MAD. The results of the *E. coli* proteins are however hard to interpret as only a few of them fall into this node category.

Regarding the interpretation of the results, not only closeness to the expected ratio is important. Even more important in practice is the ability to distinguish between

background and spiked-in proteins. In a real case scenario this would be stable proteins that do not change between samples and proteins that show a differential expression. Therefore a ROC analysis was conducted trying to differentiate between human and *E. coli* proteins based on their estimated protein ratios (see figure 5.16).

The ROC curves of bppgQuant and PQP follow a similar shape, while bppgQuant has a better overall sensitivity. With an area under the curve (AUC) of 0.9404 bppgQuant performs clearly better in distinguishing human and *E. coli* proteins than PQP (AUC = 0.8392) and SCAMPI (AUC = 0.7675). This is in line with the results from before, where the ratio distributions estimated by SCAMPI show a much higher overlap between organisms.

For data set D4, proteins of human (HeLa) and mouse cells were mixed in different concentrations. Only pairwise comparisons of conditions that do not contain the 100% HeLa or the 0% HeLa samples will be considered here, as bppgQuant currently is not able to quantify on/off proteins in a suitable manner (see the discussion section 5.5, p. 118 for more details on this topic). As an example, the histograms for the comparison of the mixture with 90% HeLa and 10% HeLa (the comparison with the largest expected sample differences) is shown in figure 5.17 (p. 108, other comparisons can be found in the appendix in figures A.12 to A.20 in the appendix,

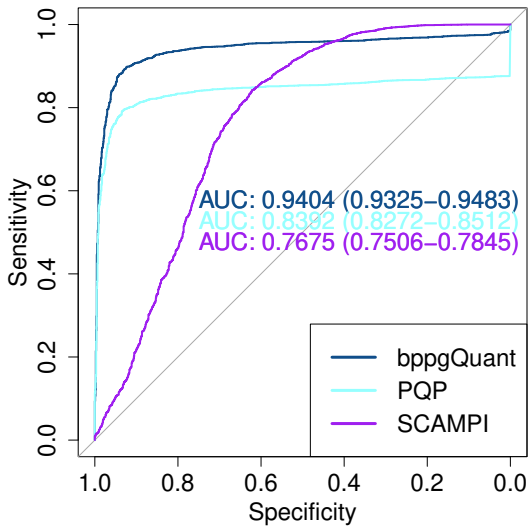


Figure 5.16: ROC curves and AUC values (with confidence interval) for distinguishing *E. coli* proteins from human proteins in data sets D3 using the estimated protein ratios from bppgQuant, PQP and SCAMPI. Contaminants were removed before calculation. Only single solutions (no range solutions) regardless of the node type were used for the calculation.

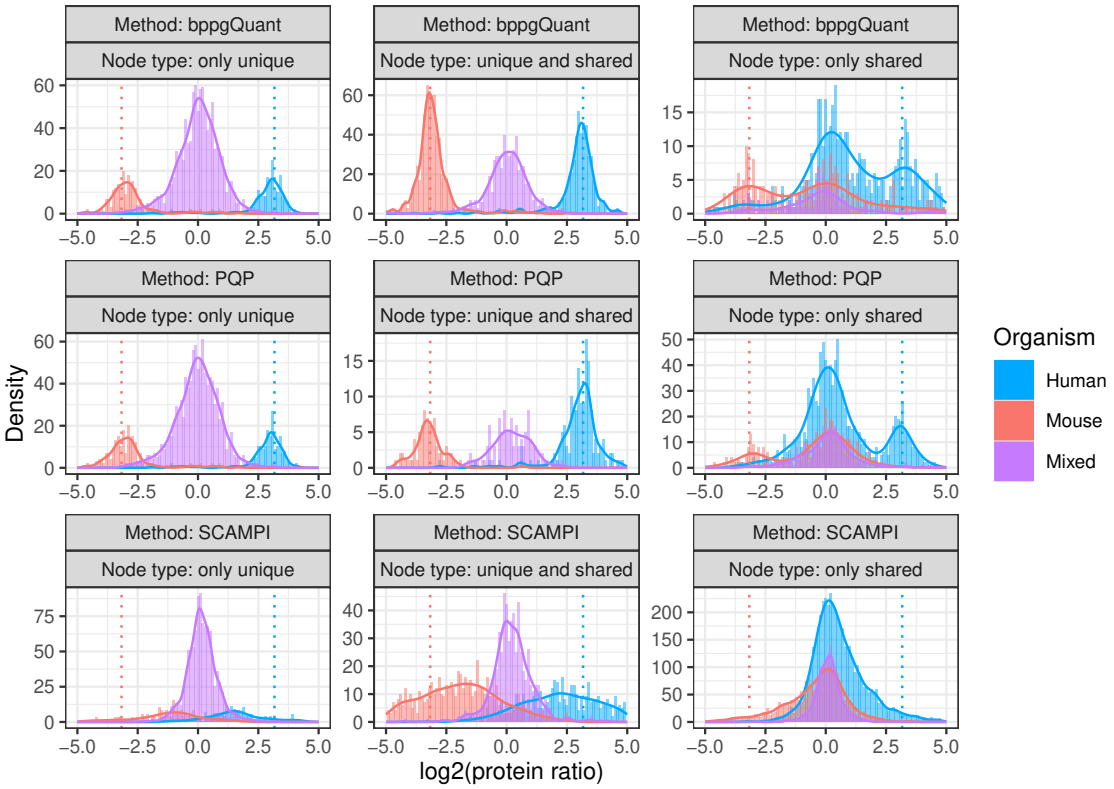


Figure 5.17: Histograms and density plots for the estimated protein ratios in data set D4 comparing the 10% human / 90% mouse sample with the 90% human / 10% mouse samples (comparison 010_090). Contaminant proteins were removed. Only single solutions (no range solutions) regardless of the node type were used. The x-axes are cut at -5 and 5 for better visibility. Please consider the different ranges of the y-axes.

p. 157-165, and the corresponding table B.14, p. 184). For human proteins a ratio of 9 (3.1699 on log2-scale) and for mouse proteins a ratio of $\frac{1}{9}$ (-3.1699 on log2-scale) is expected as shown by the vertical dashed lines in the respective group colours. In light blue, the ratios of mixed protein nodes are presented. In these nodes, protein accessions from human as well as mouse proteins are collapsed. Therefore it cannot be decided if the protein node is stemming from human or mouse and an expected protein ratio is not explicitly given.

For bppgQuant and this specific comparison, the distributions of human and mouse proteins fit well to their expected ratios for the protein nodes with only unique peptides or unique and shared peptides (also the median estimated ratios are close to the expected ratios, see table 5.4, p. 109). There is a large proportion of mixed protein nodes that contain human as well as mouse proteins. These show a wide distribution that is centered around a ratio of 0 on log2-scale. The ratios for the

Table 5.4: Median, MAD and total number of quantified proteins (n) for the different organisms (Org.), methods and node types (NT, u = only unique, u+s = unique and shared, s = only shared) for data set D4, comparison of 10% human / 90% mouse samples with 90% human, / 10% mouse samples (010_090). The expected ratio for human and murine proteins are shown in column "ER". The best median and MAD values per organism/node type are printed in bold.

Org.	Method	NT	ER	Median	MAD	n
Human	bppgQuant	u	3.1699	**3.0170**	**0.4196**	185
Human	PQP	u	3.1699	2.9908	0.4449	185
Human	SCAMPI	u	3.1699	1.5032	2.5024	185
Human	bppgQuant	u+s	3.1699	3.0684	**0.4359**	495
Human	PQP	u+s	3.1699	**3.1159**	0.4920	156
Human	SCAMPI	u+s	3.1699	2.7325	2.3052	528
Human	bppgQuant	s	3.1699	**1.1058**	4.4376	674
Human	PQP	s	3.1699	0.1804	5.0436	1,551
Human	SCAMPI	s	3.1699	0.3789	**4.1426**	4,686
Mouse	bppgQuant	u	-3.1699	-2.9794	**0.5410**	203
Mouse	PQP	u	-3.1699	**-3.0204**	0.5769	203
Mouse	SCAMPI	u	-3.1699	-1.0968	3.2020	203
Mouse	bppgQuant	u+s	-3.1699	**-3.1470**	**0.4197**	638
Mouse	PQP	u+s	-3.1699	-3.2536	0.4850	79
Mouse	SCAMPI	u+s	-3.1699	-2.2831	2.4753	685
Mouse	bppgQuant	s	-3.1699	-0.4146	5.0100	340
Mouse	PQP	s	-3.1699	**-0.8699**	5.7965	662
Mouse	SCAMPI	s	-3.1699	-0.1933	**4.4174**	2,324

protein nodes with unique and shared peptides show a similar distribution, however, the mixed protein nodes are less prominent. For the proteins with only shared peptides, the distributions of human and mouse proteins are bimodal with a peak near their expected ratio and a peak near zero (where also the mixed protein nodes are located). Also the histograms are more coarse as there are less protein nodes with only shared peptides that obtain a fixed solution. The distance of the median to the true value and the MAD values are also worse than for the other node types. Many protein nodes of this type receive a range solution, which is not incorporated into the histograms or the median and MAD calculations.

For PQP the results are similar and median and MAD values are similar to those of bppgQuant for protein nodes with only unique or unique and shared proteins. As could be seen before in figure 5.7 (p. 95), PQP has difficulties quantifying protein nodes with unique and shared peptides and quantifies much less proteins in this category, especially for mouse. SCAMPI quantifies the protein nodes considerably

Table 5.5: Area under the ROC curve (AUC) for the different methods for distinguishing between human and mouse proteins in data set D4 for the different pairwise condition comparisons. The largest AUC value in each row is printed in bold.

Comparison	bppgQuant	PQP	SCAMPI
010_025	**0.8224**	0.6743	0.5567
010_050	**0.8790**	0.7526	0.7100
010_075	**0.8745**	0.7585	0.7551
010_090	**0.8690**	0.7278	0.7814
025_050	**0.8847**	0.7465	0.7104
025_075	**0.8977**	0.7652	0.7522
025_090	**0.8716**	0.7521	0.7326
050_075	**0.8897**	0.7388	0.6791
050_090	**0.8730**	0.7206	0.6634
075_090	**0.8192**	0.6467	0.5457

worse than the other two methods taking the expected ratios into account. For protein nodes with only unique or unique and shared peptides the distributions of the log-ratios for human and mouse are much broader than for the other methods, which also leads to medians far away from the expected value and a much higher MAD than for PQP or bppgQuant (e.g., for the human protein nodes, the MAD is larger than 2 for SCAMPI and below 0.5 for the other methods). For protein nodes with only shared peptides, all three distributions are centered close to zero with a large overlap. There is no much separation between mouse and human protein nodes. Still, likely because of the more narrow distribution compared to the other methods (and few outliers for PQP and bppgQuant, which are outside of the shown x-axis range), SCAMPI has the smallest MADs in this situation.

Overall it can be seen that quantifying proteins with only shared peptides is hard for all methods, as the medians have a larger distance to the expected ratios and the MAD values are up to 10 times higher than for protein nodes that have unique peptides.

For the other pairwise comparisons, the situation is similar, however in some a bias of the estimated ratios especially for the human proteins can be observed. For example, for the comparison of HELA010 and HELA025 (see figure A.12, p. 157), the distribution of the human protein nodes are shifted to the right of the expected value. This is in correspondence with the distribution of normalized peptide ratios, see figure 5.5 (p. 89). Also, in this comparison the proportion of wrongly quantified protein ratios close to 0 is much higher.

To assess the ability of the different methods to distinguish between mouse and human protein nodes based on their estimated ratios, ROC curves were calculated (over all node types together). The corresponding area under the curve (AUCs) can be found in table 5.5 (p. 110). It can be seen that bppqQuant has the highest AUC for all comparisons, between 0.8192 and 0.8977. SCAMPI has the worst AUC in all cases except for 010_090 and is close to random guessing for 010_025 and 075_090, which have the smallest ratio differences between the organisms. Visualizations of the corresponding ROC curves can be found in figure A.21 (appendix, p. 166).

5.4.5 Summary and consideration of range solutions

A summary of the result from the sections 5.4.2 to 5.4.4 is given in table 5.6 (p. 112). For each situation, the best performing of the three methods (bppgQuant, PQP or SCAMPI) is shown regarding the median with the lowest distance to the expected value, the lowest MAD and the highest AUC value. The results for median and MAD are split by the type of protein node (only unique, unique and shared or only shared peptide nodes), while the AUC values were calculated on the whole data set. For data sets D1 and D2, additional MAD values for the spike-in proteins were calculated, while the calculations are split by organism for D3 and D4. For D1 and D2, the results over all pairwise condition comparisons are summarized in one row. This is not possible for D4 because here no stable background proteome exists. D3 only has one condition comparison.

BppgQuant has the overall best MAD values in many cases for protein nodes with only unique as well as unique and shared peptides. For only shared peptides, SCAMPI has the lowest value for all parts of D4 and the human proteins in D3. For the median, either PQP or bppgQuant have the best value, independent of the type of protein node, however the difference between the methods is comparably low in most cases. BppgQuant has the best MAD value for the spike-in proteins in data sets D1 and D2. Strikingly, also bppgQuant has constantly the best AUC value for distinguishing spike-ins from background proteins (D1 and D2) or two mixed organisms (D3 and D4).

In summary it can be seen that SCAMPI performs worst of the three methods. It has the lowest MAD for protein nodes with only shared peptides in many situations, but often a strong bias in the median of the estimated protein ratios and an overall

Table 5.6: Summary of the results of the method comparison of bppgQuant, SCAMPI and PQP for the protein node quantification on the four data sets D1, D2, D3 and D4. In each cell, the method with the best median, MAD or AUC is written and the cell is coloured with a method specific-colour for better visibility.

Data	Part	Median	MAD	Median	MAD	Median	MAD	MAD	AUC
		u	u	u+s	u+s	s	s	spike-ins	total
D1	Background	bppgQuant	bppgQuant	bppgQuant	bppgQuant	bppgQuant	PQP	bppgQuant	bppgQuant
D2	Background	bppgQuant	bppgQuant	PQP	PQP	PQP	PQP	bppgQuant	bppgQuant
D3	Human	PQP	SCAMPI	bppgQuant	bppgQuant	PQP	SCAMPI		bppgQuant
D3	*E. coli*	PQP	bppgQuant	SCAMPI	bppgQuant	PQP	PQP		
D4	010_025 Human	PQP	PQP	PQP	bppgQuant	bppgQuant	SCAMPI		bppgQuant
D4	010_025 Mouse	PQP	bppgQuant	bppgQuant	bppgQuant	PQP	SCAMPI		
D4	010_050 Human	PQP	PQP	PQP	bppgQuant	bppgQuant	SCAMPI		bppgQuant
D4	010_050 Mouse	PQP	bppgQuant	bppgQuant	bppgQuant	PQP	SCAMPI		
D4	010_075 Human	bppgQuant	bppgQuant	bppgQuant	bppgQuant	bppgQuant	SCAMPI		bppgQuant
D4	010_075 Mouse	PQP	bppgQuant	bppgQuant	bppgQuant	PQP	SCAMPI		
D4	010_090 Human	bppgQuant	bppgQuant	PQP	bppgQuant	bppgQuant	SCAMPI		bppgQuant
D4	010_090 Mouse	PQP	bppgQuant	bppgQuant	bppgQuant	PQP	SCAMPI		
D4	025_050 Human	bppgQuant	bppgQuant	PQP	bppgQuant	bppgQuant	SCAMPI		bppgQuant
D4	025_050 Mouse	PQP	bppgQuant	PQP	bppgQuant	PQP	SCAMPI		
D4	025_075 Human	bppgQuant	bppgQuant	PQP	bppgQuant	bppgQuant	SCAMPI		bppgQuant
D4	025_075 Mouse	PQP	bppgQuant	PQP	bppgQuant	PQP	SCAMPI		
D4	025_090 Human	bppgQuant	bppgQuant	PQP	bppgQuant	bppgQuant	SCAMPI		bppgQuant
D4	025_090 Mouse	PQP	bppgQuant	bppgQuant	bppgQuant	PQP	SCAMPI		
D4	050_075 Human	bppgQuant	bppgQuant	PQP	bppgQuant	bppgQuant	SCAMPI		bppgQuant
D4	050_075 Mouse	PQP	PQP	bppgQuant	bppgQuant	PQP	SCAMPI		
D4	050_090 Human	bppgQuant	bppgQuant	PQP	PQP	bppgQuant	SCAMPI		bppgQuant
D4	050_090 Mouse	PQP	PQP	PQP	bppgQuant	bppgQuant	SCAMPI		
D4	075_090 Human	bppgQuant	bppgQuant	PQP	bppgQuant	bppgQuant	SCAMPI		bppgQuant
D4	075_090 Mouse	PQP	bppgQuant	bppgQuant	bppgQuant	bppgQuant	SCAMPI		

worst performance in the ROC analysis. In 17 of the 23 cases with SCAMPI as the best performing method, bppgQuant performs better than PQP.

In table 5.6 (p. 112) for the bppgQuant method only the cases with a single solution for the protein node ratio are considered. This is because range solutions cannot easily be visualized in a histogram, be incorporated in ROC analysis or in the calculation of medians and MADs. However, this may be an unfair comparison regarding the protein nodes with only shared peptides, where bppgQuant produces between 63.5% and 86.5% range solutions depending on the data set.

A way to visualize the range solutions is shown in figure 5.18 (p. 114), exemplarily for the comparison of HELA010 with HELA090 samples in data set D4. Each horizontal line represents the range solution for one protein node reaching from the lower to the upper border of the resulting interval. The solutions are sorted by organism and from bottom to top by increasing length of the interval. The expected ratios for the human and mouse proteins are shown as dashed vertical lines. The results fit well to the observations made for the single solutions. For human and mouse protein nodes, there are intervals fitting well to the expected ratios and others that are lying around zero. This corresponds to the bimodal distribution that was visible for the single solutions, see figure 5.17 (p. 108). Furthermore the intervals for mixed protein ratios concentrate around 0, like the distributions for the mixed single solutions. Further examples of such graphics for the data sets D1, D2 and D3 can be found in the appendix as figures A.22, A.23 and A.24, pp. 167-169. For D1 and D2 only the background proteins are plotted here. For data set D3, there are range solutions only for human proteins, so no distinction between organisms is necessary.

For D1, D2 and D3 the intervals also seem to lie around the expected protein ratio of zero, however, on a closer look it is visible that most intervals start or end at zero and do not contain zero. Depending on the data set, only between 24.50% and 34.63% contain the 1, while this percentage is rising to 75.62 and 85.37% of overlap with the interval $\left[\frac{1}{1.2}, 1.2\right]$ for data sets D2 and D1, respectively (see table 5.7). For data set D3, the initial coverage of the expected ratio is the highest with 34.63%, however it only rises to 48.35% for the larger interval. This may be explained by a small shift of the interval gaps to the right which is visible in figure A.24 (p. 169) and can already be observed also for the single solutions in figure 5.15 (p. 105).

It is not directly possible to compare these range solutions with the point solutions for the same protein nodes that PQP or SCAMPI obtain. Therefore, two ways

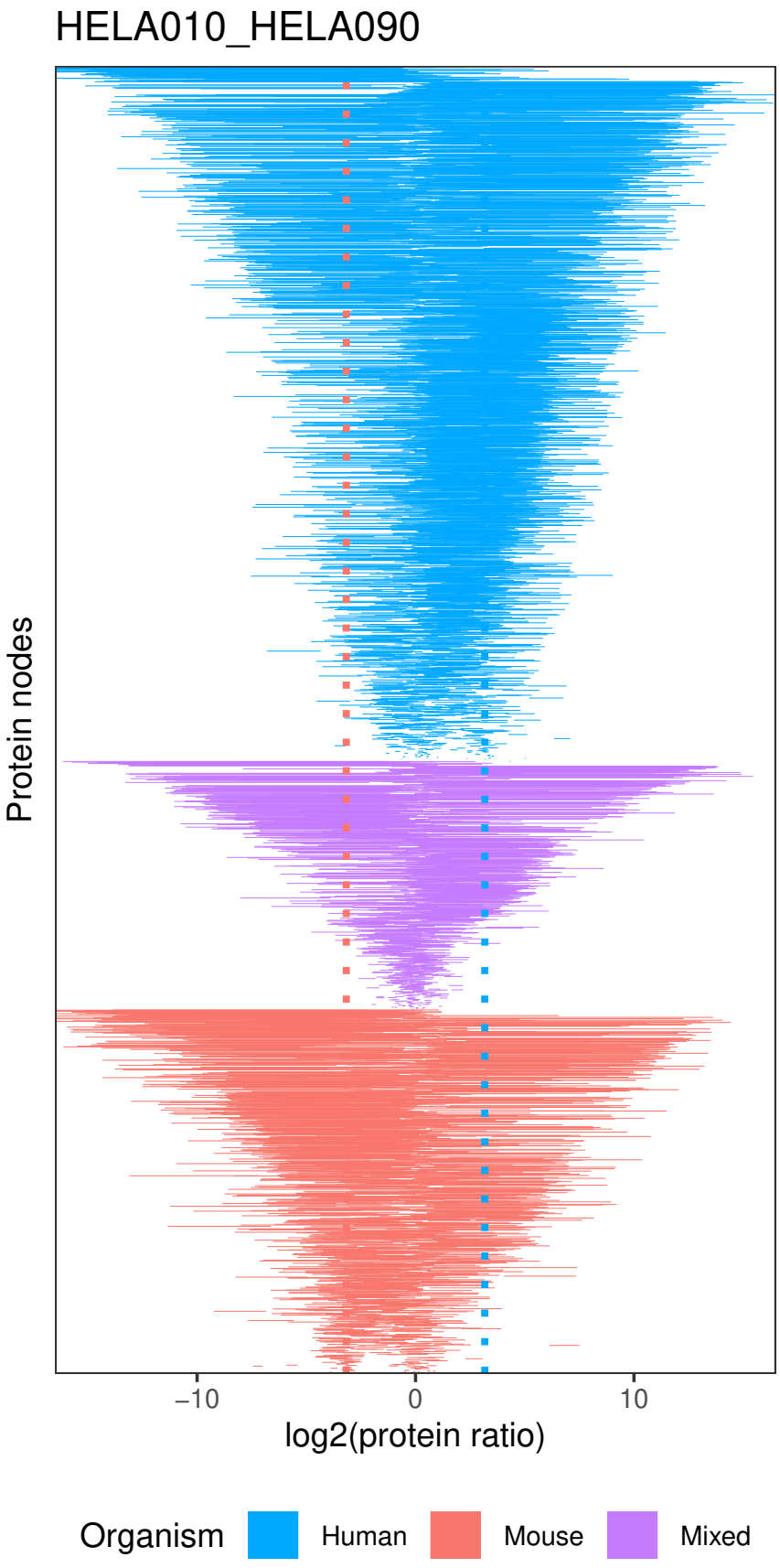


Figure 5.18: Graphical representation of range solutions for the comparison of 10% human / 90% mouse samples with 90% human / 10% mouse samples in D4. Each horizontal line represents a range solution for one protein node, sorted by organisms and decreasing interval length from top to bottom.

Table 5.7: Percentages of range solutions for data sets D1, D2 and D3 that cover the expected ratio of 1 or overlap with narrow intervals around 1.

Data set	contains 1	overlap with $\left[\frac{1}{1.1}, 1.1\right]$	overlap with $\left[\frac{1}{1.2}, 1.2\right]$
D1	30.06%	67.71%	85.37%
D2	24.50%	57.58%	75.62%
D3	34.63%	41.94%	48.35%

of simplifying the range solution to a single estimated ratio are considered. First, naturally, the center of the interval calculated as the mean of the interval borders (on log2-scale) is one possibility. As a second (conservative) option, one of the interval borders is used, namely the one that is more closer to 0, again on log2-scale. These two options are calculated and compared to the results from PQP and SCAMPI.

Table 5.8 shows the results when using the interval border closest to 0 on log2-scale for the range solutions that otherwise would not be comparable with solutions from PQP or SCAMPI. As range solutions almost exclusively occur for protein nodes with only shared peptides, only those where considered in this table. Additionally MAD values of the spike-in proteins for D1 and D2 and the results of the ROC analysis are shown, as they are also effected by results for protein nodes with only shared peptides. Asterisks show cases where the best performing method changes compared to table 5.6 (p. 112). Especially for the MAD values, which was dominated by SCAMPI, bppgQuant_0 shows good results and is the best method in 16 out of 24 cases. It also retains the best ranking in all except one ROC analysis, although the AUC values are in general lower than when ignoring the range solutions. However, in five cases the best median is lost to PQP or SCAMPI, while it is only gained in one case.

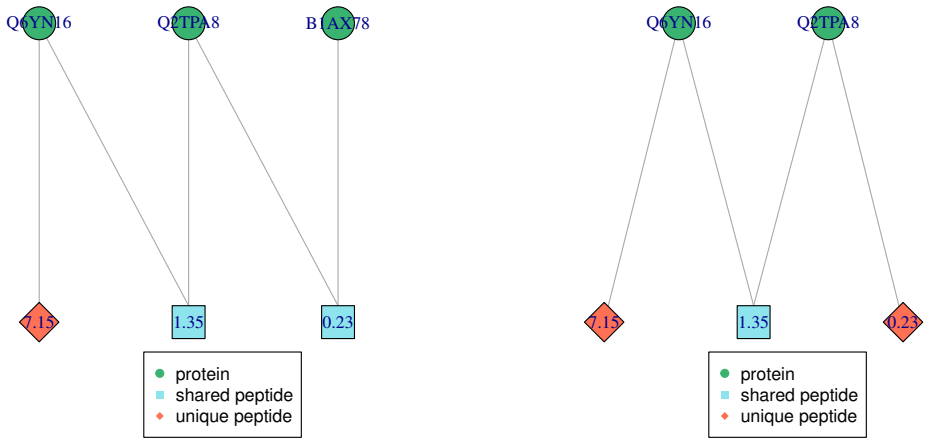
A corresponding table for considering the interval center for range solutions (bppgQuant_c) is shown in the appendix in table B.15 (p. 185). BppgQuant_c performs better regarding median values close to the expected value, however it loses rank 1 in the ROC analysis in more than half of the cases to SCAMPI or PQP. This indicates that for the range solutions on bppgQuant the center of the interval is a worse estimator for the true ratio than the conservative usage of the interval border closest to zero. This fits to the observations in figure 5.18 (p. 114), where many intervals start or end near the expected ratios, but in many cases do not cover it.

Table 5.8: Result summary using the interval border closer to 0 for range solutions stemming from bppgQuant (bppgQuant_0). Asterisks (*) indicate that the best performing method changed compared to table 5.6.

Data set	Part	Median	MAD	MAD	AUC
		Only shared	Only shared	Spike-ins	Total
D1	Background	*SCAMPI	*bppgQuant_0	bppgQuant_0	bppgQuant_0
D2	Background	PQP	*bppgQuant_0	bppgQuant_0	bppgQuant_0
D3	Human	PQP	SCAMPI		bppgQuant_0
D3	*E. coli*	*bppgQuant_0	*bppgQuant_0		
D4	010_025 Human	bppgQuant_0	*bppgQuant_0		bppgQuant_0
D4	010_025 Mouse	PQP	*bppgQuant_0		
D4	010_050 Human	*SCAMPI	SCAMPI		bppgQuant_0
D4	010_050 Mouse	PQP	*bppgQuant_0		
D4	010_075 Human	*SCAMPI	*bppgQuant_0		bppgQuant_0
D4	010_075 Mouse	PQP	*bppgQuant_0		
D4	010_090 Human	bppgQuant_0	SCAMPI		*SCAMPI
D4	010_090 Mouse	PQP	*bppgQuant_0		
D4	025_050 Human	bppgQuant_0	SCAMPI		bppgQuant_0
D4	025_050 Mouse	PQP	SCAMPI		
D4	025_075 Human	bppgQuant_0	*bppgQuant_0		bppgQuant_0
D4	025_075 Mouse	PQP	*bppgQuant_0		
D4	025_090 Human	bppgQuant_0	*bppgQuant_0		bppgQuant_0
D4	025_090 Mouse	PQP	*bppgQuant_0		
D4	050_075 Human	bppgQuant_0	SCAMPI		bppgQuant_0
D4	050_075 Mouse	PQP	*bppgQuant_0		
D4	050_090 Human	bppgQuant_0	*bppgQuant_0		bppgQuant_0
D4	050_090 Mouse	*PQP	SCAMPI		
D4	075_090 Human	bppgQuant_0	*bppgQuant_0		bppgQuant_0
D4	075_090 Mouse	*PQP	SCAMPI		

5.4.6 Selection of a subset of protein nodes

Especially in the bppgQuant results for the protein nodes with only shared peptides in data set D4 a bimodal distribution can be observed (see figures 5.17, p. 108, and 5.18, p. 114). While a part of the results align well with the expected ratios, there is a part of wrong solutions that gather around a ratio of 1, for both single and range solutions. It is possible that these deviations from the true ratios may in part be caused by protein nodes not present in the samples. As introduced in section 5.1.4 (p. 81), a selection of only protein nodes that sufficiently contribute to the differences between peptide ratios is possible. This procedure was exemplarily applied to the comparison of HELA010 and HELA090 samples in data set D4.

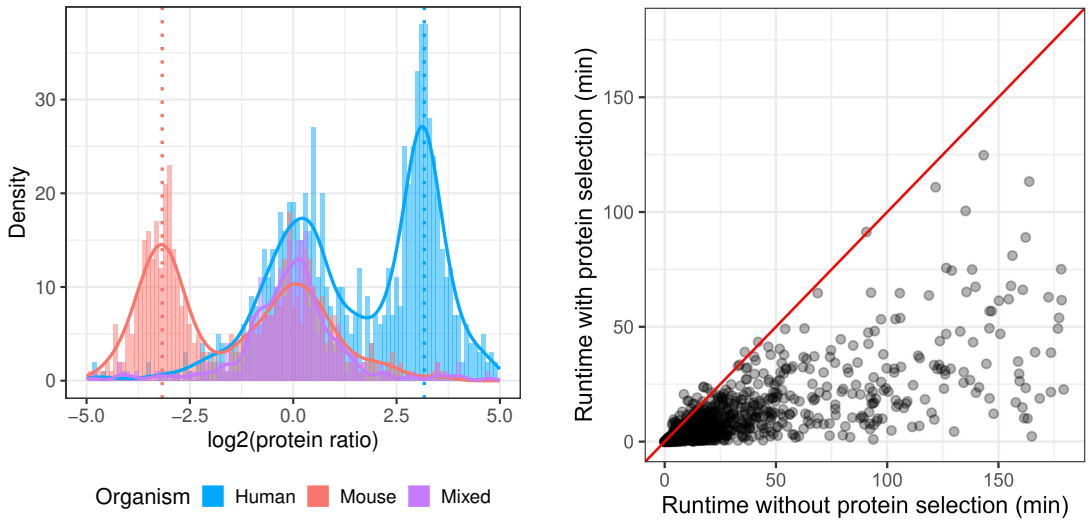


(a) Without protein selection. **(b)** With protein selection.

Figure 5.19: Example of a bipartite peptide-protein graph before (a) and after (b) protein node selection.

An example is shown in figure 5.19. The graph contains three protein nodes, only one of them has a unique peptide. For the whole graph, the error term after optimization is $< 10^{-15}$. When removing the protein node Q2TPA8, the error term dramatically rises to 1.39. In this case the graph would break into two smaller graphs. If instead B1AX78 is removed, the error term stays extremely small at $< 10^{-15}$. Further removal of protein nodes is not possible, as this would leave unconnected peptides.

Of formerly 7,292 protein nodes with only shared peptides in this comparison, 3,907 (53.6%) were removed. Of the 4,406 that are left, for 1,021 (23.2%) at least one peptide becomes unique because of the removal of other protein nodes. The largest impact of this procedure is expected for protein nodes that formerly have only shared peptides. A histogram of the corresponding estimated ratios is shown in figure 5.20(a) (p. 118). This can be compared with the corresponding figure when using all protein nodes (figure 5.17 top right, p. 108). It can be observed that when selecting only relevant protein nodes, the proportion of wrongly quantified human or mouse proteins is reduced, as the histogram peaks now are higher at the expected values than at the wrong value of 0. The median changes from formerly 1.1058 for human and -0.4146 for mouse (see table 5.4, p. 109) to much better fitting values of 2.8759 and -3.0111, respectively. The expected log2-ratios are 3.1699 for



(a) Estimated protein ratios.

(b) Runtime comparison.

Figure 5.20: (a) Histograms and density estimation for the estimated protein ratios calculated by bppgQuant with protein selection on the comparison of 10% human / 90% mouse with 90% human / 10% mouse samples (comparison 010_090) in data set D4. For better visibility, the x-axis was cut at -5 and 5. (b) Comparison of run times in minutes per bipartite peptide-protein graphs of bppgQuant with and without protein node selection, calculated on the condition comparison 010_090 in data set D4.

human and −3.1699 for mouse, respectively. Also, the MAD decreases sharply from 4.4376 to 0.9252 for human and from 5.0100 to 0.7023 for mouse. Over all protein nodes, the AUC rises from 0.8690 to 0.8982. Another advantage of the protein node selection procedure is the reduced run time. As shown in figure 5.20(b), the runtime is decreased for almost all bipartite peptide-protein graphs. Additionally, protein ratios could be calculated for 56 additional graphs within the set limit of three hours.

5.5 Discussion

In this chapter the novel protein quantification algorithm bppgQuant was proposed that solves optimization problems generated from the bipartite peptide-protein graphs to estimate protein ratios from measured peptide ratios. In contrast to many other existing methods it explicitly calculates ratios also for protein nodes without any unique peptide, in many cases by producing a range solution that consists of an interval of possible optimal protein ratios. The results from bppgQuant were compared with two other methods, PQP and SCAMPI, on four different data sets

with known expected protein ratios. While the median of protein ratios calculated by PQP are often the closest to the expected ratio (closely followed by bppgQuant), bppgQuant has the lowest MAD in many cases for protein nodes with only unique and unique and shared peptides. For protein nodes with only shared peptides, SCAMPI has the lowest MAD in most cases, however, there are large distances between median protein ratios and expected values. Also, the ROC analysis showed the overall lowest AUC for SCAMPI. BppgQuant showed overall the best performance in distinguishing spike-in from background proteins or different organisms within the same data set and in accurately quantifying the spike-in proteins in data sets D1 and D2.

The three compared protein quantification methods differ in the number of protein nodes that could be quantified in the different data sets (see figure 5.7, p. 95). SCAMPI computes protein ratios for every single protein node, irrespective of the node type. PQP does not give a solution for many protein nodes with "unique and shared" or "only shared" peptides. In principle, bppgQuant can estimate a single protein ratio or range solution for each protein node. However, some limitations lead to not reporting all possible results. First, the computation of the estimate for the protein ratio for a bipartite graph was limited to three hours. As explained in section 5.1.3 (p. 79), to calculate also multiple possible solutions for the same graph, the optimization is repeated several times on a grid of values for the weights C_i, which may lead to a longer run time. Additionally, it was observed that the optimization procedure took extremely long to converge for some graphs, especially when the C_i grid point was close to 1 or 0, likely because of numerical difficulties (e.g., division by a value close to 0). However, such grid points may be necessary to consider, as these are used to improve the estimation of the potential lower and upper limits of the range solutions, which may be a better estimator than the center of the intervals, as was shown in section 5.4.5 (p. 111). Adjusting of the internal parameters of the optimization algorithm, e.g., tolerances and stop criteria, may decrease the runtime in these cases without losing accuracy for other graphs. Second, there are protein ratios estimated as close to zero by bppgQuant. As explained in section 5.4 (p. 92), there are single solutions and lower limits of range solutions with values close to zero. It was decided to omit these outlying results for the method comparison. These may be caused by contradictions in the peptide ratios (caused by outliers in the peptide ratios), as shown in an example in figure 5.6 (p. 93). This issue may be addressed in the future by appropriate handling of peptide outliers, as discussed in the outlook chapter, section 6.2.

The PQP method is able to calculate a protein ratio for 32.7-84.3% of the protein nodes with "only shared" peptides, which is the most difficult node type to quantify. However, it only quantifies even a smaller proportion (10.7-35.7%) of protein nodes with "unique and shared" peptides and also shows a bias in the calculated protein ratios for this node type, e.g. for data set D1 (see figure 5.8, p. 97). It is not completely clear why PQP has problems with quantifying this type of node, as the present unique peptides should make it easier than quantifying proteins with no unique peptides at all. In many of these cases, PQP calculates intensities of 0 in both compared conditions, leading to an undefined protein ratio despite the presence of one or more unique peptides. As another problem it was observed that the PQP method is not robust against using the reciprocal of the peptide ratios as an input (which, ideally should lead to the reciprocal of the corresponding originally calculated protein ratios), especially for ratios close to 1 (see figure A.25 in the appendix, p. 170). This may be explained by the minimization of the sum of absolute error terms causing an asymmetric consideration of ratios larger or smaller than 1.

SCAMPI calculates protein intensities for every single protein accession that has any connection to at least one quantified peptide in the data set. However, in summary, SCAMPI showed the worst results compared to bppgQuant and PQP, which cannot properly be explained at the moment. In the original paper (Gerster et al., 2014), the obtained protein scores are correlated with known protein concentrations and show acceptable results. However in this thesis, it showed wider protein ratio distributions than PQP or bppgQuant for data sets D1 and D2 (figure 5.8, p. 97, and figure 5.9, p. 99) and higher variability in the estimation of spike-in protein ratios (figure 5.10, p. 101, and figure 5.12, p. 102). On data sets D3 and D4 SCAMPI's protein ratio estimates showed the worst ability to distinguish between the different organisms especially for protein nodes with only shared peptides, see for example figure 5.17 (p. 108).

In contrast to other protein quantification methods, bppgQuant reports intervals of possible solutions in cases where multiple solutions are optimal with respect to the optimization problem. This happens almost exclusively in protein nodes without any unique peptides. This highlights the complexity of the problem in these cases and that many other methods may hide this complexity by providing a single solution, e.g., like SCAMPI or PQP. However, this unusual type of solution poses challenges,

e.g., with regard to further statistical analysis of the results (see section 6.3) or visualization for the end user (see section 6.4).

As described in section 5.1.4 (p. 81) not all protein nodes within a bipartite peptide-protein graph may be present in each sample. For example, many proteins are specific to a certain kind of tissue or body fluid. This kind of information is for example collected in the Human Proteome Atlas (Uhlén et al., 2015). Some proteins may be present in the sample but difficult to measure by standard approaches in mass spectrometry, e.g., hydrophobic membrane proteins which require special treatment (Yang et al., 2023). Another possible reason for not found proteins is the usage of a too large FASTA file, e.g. containing protein sequences from not present organisms, which influences the the calculation of the false discovery rate and may lead to less identified peptides. However, the bppgQuant method assumes that each protein node in a graph is present in at least one of the two considered samples, see equation 5.3 (p. 74), which requires that $\exists\ k \in \{1, \ldots, m\} : A_k \neq 0$. To tackle this problem it would be possible to apply a protein inference algorithm beforehand to remove some protein nodes from the bipartite peptide-protein graphs. However, popular algorithms like Occam's razor do not take the measured peptide ratios into account and would most likely eliminate most protein nodes that do not have unique peptides. Another possibility would be to pre-filter protein nodes, e.g., based on the specific organ or body fluid by using resources like the Human Proteome Atlas (Uhlén et al., 2015). However, the assignment of proteins to body fluids or organs is still part of ongoing research and may not be complete. Furthermore, in the medical context, expression of atypical proteins in a specific tissue or body fluid may be a sign for a disease and should not be excluded on principle.

In section 5.1.4 (p. 81) an iterative method to remove protein nodes that do not sufficiently contribute to explaining the present measured peptide ratios is proposed. Based on the idea that important protein nodes would largely increase the error term when removed, the set of protein nodes is reduced before applying bppgQuant to get a single or range solution for the ratios of the remaining proteins. It was shown that this procedure improves the bimodal protein ratio distribution exemplarily on data set D4 (see figure 5.20, p. 118), where a part of unexpected quantification results may have been caused by protein nodes not present in the samples. While this procedure reduced the number of quantifiable protein nodes, it may increase the accuracy and reliability of the remaining results. Additionally, this decreases the overall runtime of the bppgQuant method, although overhead for removing the

protein nodes and calculating the error terms exist. This may be due to the fact that the complexity of the graphs is reduced (e.g. some peptides become unique), leading to potentially less numerical issues like described above. The protein node selection can be adapted by changing the parameter of tolerated error increase, which was set to 5% here. This parameter has to be optimized to further improve this approach. A smaller value would lead to removing less protein nodes (and potentially less improvement on the ratio distribution), while a larger value will lead to removing more protein ratios, but larger error terms that could be a sign of inaccurate protein ratio estimations.

In the four used test data sets, few single spike-ins are mixed into a stable background proteome (data sets D1 and D2) or two proteomes of different organisms are mixed (D3 and D4). However, these data sets have some disadvantages for evaluating the performance of bppgQuant. In spike-in data sets like D1 and D2, the spike-in proteins are often chosen so that they have unique peptides and, if possible, do not share too many peptides with the stable background to avoid interferences. Evaluating methods based on their ability to quantify proteins with shared peptides is therefore often limited to stable background proteins (with expected protein ratios of 1), or very few spike-in proteins with shared peptides. Data sets with mixed proteomes like D3 and D4 may offer a larger pool of proteins with shared peptides with expected ratios other than 1. In data set D3, proteomes of human and *E. coli* are mixed. However, *E. coli* itself has comparably few shared peptides and even less that are shared with the human proteins. For D4, human and mouse proteomes were mixed, which show a large fraction of shared peptides between those organisms.

In theory, from those four data sets especially D4 is suitable to assess the ability of quantifying protein nodes with shared peptides. However, it is difficult o properly normalize the peptide intensities for D3 and D4 (see section 5.3.2, p. 85). For D3, unequal protein amounts are measured in the two experimental groups, causing a deviation from the expected ratios. For D4, the normalization becomes even more challenging, as no stable background exist between the different experimental conditions. For those two data sets, the LTS normalization was used, which tolerates violations of the normalization assumption that the majority of proteins do not change between samples. While this normalization worked better than the usual loess normalization, it could not completely eliminate those issues. Those normalization issues make the reliance on the known expected protein ratios difficult, as it is not clear if they are really met in the underlying peptide-level data.

Therefore, additional to comparing the estimated protein ratios to the expected ratios, ROC analyses were calculated. Here, the expected protein ratios do not play a role, only the ability of the method to distinguish between the two organisms or between background and spike-in proteins, depending on the data set. Also in real scenarios, e.g. biomarker discovery, detecting the differentially expressed proteins is much more important than the exact estimation of the protein ratios. A typical workflow would contain different stages, first detecting potential biomarkers in bottom-up proteomics data and later verifying those using the more precise targeted proteomics approach (Nakayasu et al., 2021).

In this thesis results of the proposed bppgQuant method were compared against results from SCAMPI and PQP, because for these two methods also the possibility to quantify proteins without unique peptides is claimed. However, these two methods are not well known and not commonly used in the proteomics field. For gaining the acceptance of end-users, e.g. biologists, it would be necessary to compare the results to well-known tools, e.g. MaxQuant, especially regarding the number of quantified protein nodes. In this thesis, the raw data were processed with MaxQuant and the respective quantitative peptide-level data were used, therefore, also the protein quantification results from MaxQuant are available. MaxQuant uses the so-called "razor peptide" principle which means that shared peptides are assigned to the single protein group with the highest further evidence (e.g., by the highest number of unique peptides). Therefore, shared peptides will usually be assigned to protein groups that have unique peptides. A protein group without unique peptides may be quantified only in rare cases, if those shared peptides do not belong to any protein that contains unique peptides and are therefore assigned to the specific protein group as razor peptides. BppgQuant on average is able to quantify 59-93% more protein nodes than quantified protein groups by MaxQuant for data sets D1, D3 and D4, using raw intensities (see figure B.16 in the appendix, p. 185). For data set D2, bppqQuant and MaxQuant quantify roughly the same number of protein nodes, possibly explained by the lower complexity of the D2 data set and less protein nodes without unique peptides. When comparing the results to the LFQ-values (normalized protein intensities by MaxQuant, which introduces missing value) with the bppgQuant results after protein selection, bppgQuant quantifies 1.53 to 2.63 times as many protein nodes / groups than MaxQuant. However, it has to be noted that a direct comparison of the ratios calculated by bppgQuant and from the MaxQuant intensities is complicated because MaxQuant builds its own protein groups during the protein inference step, which are not completely identical to the

protein nodes formed by bppgQuant. In conclusion, especially data sets stemming from samples from complex organisms like mouse or human can benefit from using bppgQuant regarding the number of quantified protein nodes and accuracy of the quantifications. As many protein nodes without unique peptides can additionally be quantified, proteins that were previously ignored in biomarker studies can now be assessed.

6 Overall discussion and outlook

In this thesis a novel protein quantification method, called bppgQuant was developed. It estimates protein ratios from measured peptide ratios while utilizing the structure of the bipartite peptide-protein graphs. In summary, it showed good performance on different data sets, also compared to SCAMPI (Gerster et al., 2014) and PQP (Dost et al., 2009; Dost et al., 2012).

In chapter 4, bipartite peptide-protein graphs for the four test data sets D1-D4 were characterized. Those were used for quantification of protein nodes in chapter 5. For the characterization, protein and peptide nodes were collapsed if they exhibit the same set of edges. For the quantification, the peptide nodes were not collapsed, but each peptide sequence was represented by a separate peptide node. This was done to make use of the individual peptide ratios instead of summarizing them and using them to obtain an error term for each individual peptide. This also ensures that each peptide ratio has overall the same impact on the results. If the peptide nodes would have been collapsed, for example several unique peptides of a certain protein may together have the same impact as a single shared peptide, which may lead to a bias in the quantification. However, the results of the characterization in chapter 4 are still valid and can be used to assess the difficulty of quantification, as this is not influenced by multiple peptide nodes of the same "type", i.e., with the same set of edges. For example, an N-shaped graph is still difficult to quantify, no matter if only one or multiple unique or shared peptide nodes exist.

The characteristics of the bipartite peptide-protein graphs could be used more extensively in the future to improve the bppgQuant method. For very small bipartite peptide-protein graphs, formulas for the solutions (single or range solutions) can be directly derived from the equation system in equation 5.4 (p. 75) without the optimization step, as was shown for N- and M-shaped graphs in section 5.1.1 (p. 72). It would be interesting to see if such analytical solutions could be also derived for larger graphs, possibly by using the analytical results from smaller graphs,

which can be found as subgraphs in the larger graphs. As a basis for this, a efficient method to find smaller subgraphs in larger bipartite peptide-protein graphs is needed, which was the topic of Katharina Neuhaus' Bachelor's thesis (Neuhaus, 2023). She compared two algorithms, VF2 (Cordella et al., 2004) and LAD (Solnon, 2010) for subgraph matching from the `igraph` R package, which were adapted to be usable for the special case of bipartite peptide-protein graphs. For example, the general node type (protein node or unique peptide node or shared peptide node) had to be distinguished, which was solved by colouring the different node types. With the results of the subgraphs matching it is possible to find smaller bipartite graphs patterns, where an analytical solution is known, in larger graphs and investigate the possibility to transfer (parts of) the solution to the larger graph.

Furthermore, different improvements and optimizations to the bppgQuant method can be made, which will be discussed in the following sections. In cooperation with Prof. Martin Eisenacher and Prof. Jörg Rahnenführer, the proposal *"Improved protein quantification with mathematical optimization techniques using bipartite peptide-protein graph structures"* was written, which was approved by the DFG (project number 532401634). The improvements of bppgQuant proposed in sections 6.1, 6.3, 6.4 and partly 6.5 are part of this future DFG project.

6.1 Handling of missing values and on/off peptides

There is often a high proportion of missing intensity values in bottom-up proteomics data on the peptide level (Webb-Robertson et al., 2015). Low-abundant proteins that are close to the detection limit of the mass spectrometer may for example lead to missing values in the peptide-level data (missing not at random, MNAR). Other factors may cause missing at random (MAR) or missing completely at random (MCAR) values, for example noisy spectra that are not identified or the stochastic nature of choosing precursor ions for fragmentation in data-dependent acquisition proteomics (Matafora and Bachi, 2021). On/off proteins or peptides are present in one experimental group, while completely missing in the other. At the moment bppgQuant does not make use of peptides with too many missing values or on/off peptides. A valid peptide ratios is here defined as having at least two out of three valid values in each compared group. This actually makes it impossible to detect or correctly quantify on/off proteins.

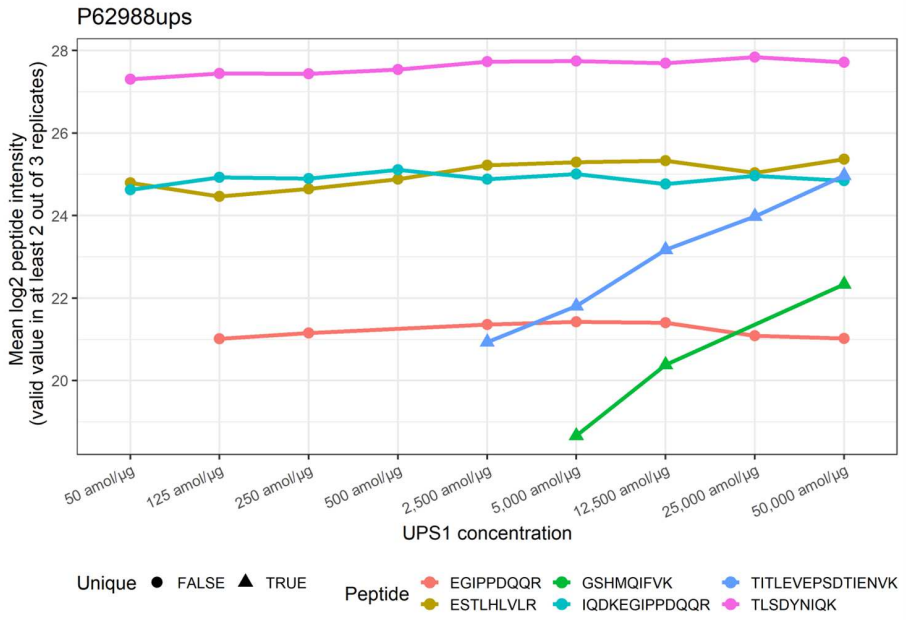


Figure 6.1: Peptide intensities of unique and shared peptides of the spiked-in protein P62988ups in data set D2. Taken from the DFG proposal "Improved protein quantification with mathematical optimization techniques using bipartite peptide-protein graph structures".

An example for this is given in figure 6.1. It shows the peptide intensities associated with the spiked-in UPS protein P62988ups from data set D1. It can be seen that in total six peptides were quantified, two unique and four shared peptides. While the shared peptides show an almost constant pattern over the different UPS1 standard concentrations, the unique peptides rise with rising concentration. The shared peptides are not quantified in the lower concentration groups, only starting at 2500 or 5000 amol/µg, respectively. In the lower concentrations, MNAR missing values occur because it is below the detection limit (see also Ramus et al., 2016b). If a low and a high concentration are compared, no valid peptide ratio can be calculated for these two unique peptides. Therefore, for this protein node only the shared peptides will be used, which suggest a more or less stable intensity pattern leading to peptide ratios close two 1 for the different comparisons and in consequence an incorrectly estimated protein ratio.

Missing values and especially on/off peptides and proteins are a challenge for protein quantification with the bppgQuant method. To overcome this problem two main approaches are thinkable. The first one is to incorporate the information about on/off peptides into the bipartite peptide-protein graphs, similarly to Bamberger et al., 2018, where peptide nodes with a log-transformed ratio of $-\infty$ or ∞ exist.

Bamberger et al., 2018 use the bipartite graphs only for detection of isoforms, not for quantification. However, the presence of such on/off peptide nodes may help to detect on/off proteins and avoid incorrect ratio calculations as in the example above. The second approach is to impute missing peptide intensities. A large variety of missing value imputation methods have been tested for proteomics data, however the mixture of different missingness types makes this very challenging (Shen et al., 2022; Egert et al., 2021; Goeminne et al., 2020). Also the usage of imputation methods for left-censored data (i.e. the missing peptide intensities lower than the lower limit of quantification) may be helpful (Gardner and Freitas, 2021; Wei et al., 2018).

6.2 Handling of outliers

Even unique peptides of the same protein node do not have the same measured peptide ratios between two samples because of technical variability and other influences that may affect the underlying peptide intensities. Additionally, outliers in the peptide intensities may occur (Tsai et al., 2020; Erhard and Zimmer, 2012), which may propagate into outliers in the peptide ratios, see an example in figure 5.6 (p. 93). Such outlying peptide ratios may cause problems during protein quantification, e.g. because they may lead to contradictions in the graph with resulting numerical issues during minimization of the error term.

BppgQuant currently does not especially handle outliers or filter them beforehand. Pre-filtering of outliers as it is done within MSStats (Tsai et al., 2020) may be a valuable pre-processing step for bppgQuant. Furthermore, in his internship Bürkan Kalaycik investigated the influence of peptides with missed cleavages on the structure of the bipartite graphs, see section 4.4.2 (p. 51) as well as on the distribution of the peptide ratios (Kalaycik, 2023). He found out that peptides with missed cleavages or fully cleaved parts of them are prone to be outliers. This may be explained by different cleavage behaviour in the different samples, i.e. one sample may have a different percentage and distribution of missed cleavages as another sample. This may cause miscleaved peptide ratios between these two samples to deviate from the original protein ratio. This phenomenon has to be investigated further together with other possible reasons for outliers, like unconsidered post-translational modifications or isoforms (Erhard and Zimmer, 2012). The aim would

be to come up with a solution, e.g. pre-filtering of peptides with missed cleavages or summarizing intensities from miscleaved peptides and their cleaved sub-sequences or from the same peptide sequences with different PTMs.

6.3 Estimation of uncertainty

One important aim of protein quantification is the subsequent comparison of two or more experimental groups or conditions to find differentially expressed proteins that may serve as biomarkers. Currently, bppgQuant calculates only estimates for the protein ratios without providing estimates for their variability, which would be necessary to construct a statistical test or to compute confidence intervals.

Deriving a formula for the theoretical variance of the estimated protein ratios is complicated, as the procedure to calculate the protein ratios is complex due to the optimization step and the influence of the underlying graph structure. Bootstrapping would allow to estimate the variance by repeatedly drawing random samples with replacement from the data and to consequently form confidence intervals (Efron and Narasimhan, 2020). For small sample sizes, which are not uncommon in proteomics, special methods may be needed, e.g. the pooled sample method by Dwivedi et al., 2017.

6.4 Implementation of a user-friendly tool

An important step would be to provide a user-friendly software tool that allows also non-experts in programming to apply bppgQuant to their quantitative peptide data. Based on the R package `bppg` offering a graphical user interface as an RShiny web application (Chang et al., 2023) would be straightforward. Also, incorporation of bppgQuant into nextflow workflows (Di Tommaso et al., 2017) is possible, for example directly following peptide identification and quantification steps. An appropriate and intuitive visualization of the results, especially for the novel concept of range solutions will be crucial. To ensure interoperability with other proteomics software tools, reporting of the results in established standard formats like mzTab (Griss et al., 2014) or mzQuantML (Walzer et al., 2013) should be an aim. On the other hand, allowing different output formats from popular peptide quantification

software other than MaxQuant as input files for bppgQuant, like Proteome Discoverer (Orsburn, 2021), Spectronaut (Bruderer et al., 2015) or DIA-NN (Demichev et al., 2019), would greatly enhance the usability of bppgQuant.

6.5 Further test data sets and simulation

The four spike-in data sets used in chapter 5 are useful to evaluate different protein quantification methods. However, they all have certain disadvantages. Often, single spike-in proteins, like in data sets D1 and D2 are chosen so that they have unique peptides and if possible share as few as possible peptides with the background, to reduce interference effects. For data sets like D3 or D4 that consist of mixtures of proteomes from different organisms, normalization issues may occur as stated in the discussion chapter 5.5 (p. 118). The simulation of data sets may serve as an approach to overcome such problems, especially to test cases that may occur in real data sets but are seldomly found in spike-in data sets. For example, the so called plasmode approach may be used (Gadbury et al., 2008), which uses real data sets and adds some artificial known effects. The advantage of this approach is that many properties from the real data set are kept and that scenarios of different complexity can be investigated. The plasmode method needs to be adapted for proteomics data, especially regarding the relationship between proteins and their unique and shared peptides.

In section 6.1 the handling of missing values, especially in context of on/off peptides and proteins is mentioned. Test data sets with known missing proteins can mostly be found in context of protein inference. One test data set that focusing on shared peptides, the one proposed by The et al., 2018. As this data was created with protein inference in mind, protein mixtures with known missing proteins, that however share peptides with present proteins are present. In the data set D4 used in this thesis, also samples with 100% HeLa or 100% mouse are present. These were not used in this thesis, because bppgQuant cannot handle on/off peptides yet. However, it would also be a good test data set for this scenario.

As further step it would be to test bppgQuant on a data set from a real-world application. However, in such data sets it is difficult to assess the quality of the protein quantification, as it is not clear what protein ratios to expect for specific proteins. One possibility would be to use the ratios calculated by bppgQuant to

find significantly differentially expressed proteins between the experimental groups and assess the meaningfulness of the results on the basis of GO or pathway analyses. However, this would require that it is possible to calculate a statistical test on the ratios generated by bppgQuant, which is currently not possible as an approach to estimate the uncertainty of the estimates still needs to be developed, see also section 6.3 (p. 129).

So far bppgQuant was only tested on data measured with the data-dependent acquisition method (DDA) and preprocessed with the MaxQuant software (Cox and Mann, 2008). Other software tools for quantifying peptides from DDA data exist, like Proteome Discoverer (Orsburn, 2021) or OpenMS (Röst et al., 2016). These tools use different peptide identification algorithms internally as well as different strategies for processing of the peptide intensities, which could in consequence influence the performance of protein quantification. Therefore, testing bppgQuant on peptide-level quantitative data from different sources would be beneficial.

In DDA only the most precursor ions are selected for further fragmentation. As an alternative, the data-independent acquisition approach exists, where all precursor ions are fragmented within defined m/z windows. DIA has therefore the ability to quantify more peptides, especially low-abundant ones, that could have an impact on the bipartite peptide-protein graphs as well as on the quantification (Bilbao et al., 2015). It also showed an improved accuracy for peptide quantification which could positively influence the performance of bppgQuant. Therefore bppgQuant needs to be evaluated on peptide-level data produced by DIA analysis, e.g. by the software tools Spectronaut (Bruderer et al., 2015) or DIA-NN (Demichev et al., 2019). Similarly, bppgQuant was only evaluated on label-free proteomics data, however it should in principal also work with quantitative peptide data from labelled approaches like SILAC or TMT (Ong et al., 2002; Thompson et al., 2003, see also section 2.3).

Because it focuses on the quantification of proteins with "unique and shared peptides" or even "only shared peptides", bppgQuant may be additionally useful in two research areas within proteomics, which should be investigated in the future. First, it could help to quantify different proteoforms or isoforms, as they often have a large sequence similarity (Plubell et al., 2022). Second, metaproteomics would be a use case, which refers to analysing samples that are a mixture of different organisms (Wilmes et al., 2015), mostly referring to microbes, e.g. in context of gut samples or marine

samples, which share a lot of peptides or even whole proteins (Blakeley-Ruiz and Kleiner, 2022).

References

Aebersold, R. and Mann, M. (2003). Mass spectrometry-based proteomics. *Nature* 422, pp. 198–207. DOI: 10.1038/nature01511.

Aebersold, R. and Mann, M. (2016). Mass-spectrometric exploration of proteome structure and function. *Nature* 537, pp. 347–355. DOI: 10.1038/nature19949.

Ammar, C., Schessner, J. P., Willems, S., Michaelis, A. C., and Mann, M. (2023). Accurate Label-Free Quantification by directLFQ to Compare Unlimited Numbers of Proteomes. *Molecular and Cellular Proteomics* 22.7, 100581. DOI: 10.1016/j.mcpro.2023.100581.

Ballman, K. V., Grill, D. E., Oberg, A. L., and Therneau, T. M. (2004). Faster cyclic loess: normalizing RNA arrays via linear models. *Bioinformatics* 20.16, pp. 2778–2786. DOI: 10.1093/BIOINFORMATICS/BTH327.

Bamberger, C., Martínez-Bartolomé, S., Montgomery, M., Pankow, S., Hulleman, J. D., Kelly, J. W., and Yates, J. R. (2018). Deducing the presence of proteins and proteoforms in quantitative proteomics. *Nature Communications* 9.1, 2320. DOI: 10.1038/S41467-018-04411-5.

Bantscheff, M., Lemeer, S., Savitski, M. M., and Kuster, B. (2012). Quantitative mass spectrometry in proteomics: critical review update from 2007 to the present. *Analytical and bioanalytical chemistry* 404.4, pp. 939–965. DOI: 10.1007/S00216-012-6203-4.

Barkovits, K., Pacharra, S., Pfeiffer, K., Steinbach, S., Eisenacher, M., Marcus, K., and Uszkoreit, J. (2020). Reproducibility, Specificity and Accuracy of Relative Quantification Using Spectral Library-based Data-independent Acquisition. *Molecular and Cellular Proteomics* 19.1, pp. 181–197. DOI: 10.1074/mcp.RA119.001714.

Bateman, A., Martin, M. J., Orchard, S., Magrane, M., Ahmad, S., Alpi, E., Bowler-Barnett, E. H., Britto, R., Bye-A-Jee, H., Cukura, A., Denny, P., Dogan, T., Ebenezer, T. G., Fan, J., Garmiri, P., Costa Gonzales, L. J. da, Hatton-Ellis, E., Hussein, A., Ignatchenko, A., Insana, G., Ishtiaq, R., Joshi, V., Jyothi, D., Kandasaamy, S., Lock, A., Luciani, A., Lugaric, M., Luo, J., Lussi, Y., MacDougall, A., et al. (2023). UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research* 51.D1, pp. D523–D531. DOI: 10.1093/NAR/GKAC1052.

Bates, D, Maechler, M, and Jagan, M (2024). Matrix: Sparse and Dense Matrix Classes and Methods. R package version 1.6-5. URL: https://CRAN.R-project.org/package=Matrix.

Bengtsson, H. (2022). R.utils: Various Programming Utilities. R package version 2.12.2. URL: https://CRAN.R-project.org/package=R.utils.

Bilbao, A., Varesio, E., Luban, J., Strambio-De-Castillia, C., Hopfgartner, G., Müller, M., and Lisacek, F. (2015). Processing strategies and software solutions for data-independent acquisition in mass spectrometry. *PROTEOMICS* 15.5-6, pp. 964–980. DOI: 10.1002/PMIC.201400323.

Bischl, B, Lang, M, Horn, D, Richter, J, and Surmann, D (2022). BBmisc: Miscellaneous Helper Functions for B. Bischl. R package version 1.13. URL: https://CRAN.R-project.org/package=BBmisc.

Bjornson, R. D., Carriero, N. J., Colangelo, C., Shifman, M., Cheung, K. H., Miller, P. L., and Williams, K. (2008). X!!Tandem, an Improved Method for Running X!Tandem in Parallel on Collections of Commodity Computers. *Journal of proteome research* 7.1, pp. 293–299. DOI: 10.1021/PR0701198.

Blakeley-Ruiz, J. A. and Kleiner, M. (2022). Considerations for constructing a protein sequence database for metaproteomics. *Computational and Structural Biotechnology Journal* 20, pp. 937–952. DOI: 10.1016/J.CSBJ.2022.01.018.

Blein-Nicolas, M., Xu, H., Vienne, D. de, Giraud, C., Huet, S., and Zivy, M. (2012). Including shared peptides for estimating protein abundances: A significant improvement for quantitative proteomics. *PROTEOMICS* 12.18, pp. 2797–2801. DOI: 10.1002/pmic.201100660.

Blein-Nicolas, M. and Zivy, M. (2016). Thousand and one ways to quantify and compare protein abundances in label-free bottom-up proteomics. *Biochimica et biophysica acta* 1864.8, pp. 883–895. DOI: 10.1016/J.BBAPAP.2016.02.019.

Blennow, K. (2004). Cerebrospinal Fluid Protein Biomarkers for Alzheimer's Disease. *Neurotherapeutics* 1.2, pp. 213–225. DOI: 10.1602/NEURORX.1.2.213.

Bludau, I., Frank, M., Dörig, C., Cai, Y., Heusel, M., Rosenberger, G., Picotti, P., Collins, B. C., Röst, H., and Aebersold, R. (2021). Systematic detection of functional proteoform groups from bottom-up proteomic datasets. *Nature Communications* 12.1, 3810. DOI: https://doi.org/10.1038/s41467-021-24030-x.

Bolstad, B. M., Irizarry, R. A., Åstrand, M., and Speed, T. P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* 19.2, pp. 185–193. DOI: 10.1093/BIOINFORMATICS/19.2.185.

Borrebaeck, C. A. (2017). Precision diagnostics: moving towards protein biomarker signatures of clinical utility in cancer. *Nature Reviews Cancer* 17.3, pp. 199–204. DOI: 10.1038/nrc.2016.153.

Brenes, A., Hukelmann, J., Bensaddek, D., and Lamond, A. I. (2019). Multibatch TMT Reveals False Positives, Batch Effects and Missing Values. *Molecular and Cellular Proteomics* 18.10, pp. 1967–1980. DOI: 10.1074/MCP.RA119.001472.

Brown, K. A., Melby, J. A., Roberts, D. S., and Ge, Y. (2020). Top-down proteomics: challenges, innovations, and applications in basic and clinical research. *Expert review of Proteomics* 17.10, pp. 719–733. DOI: 10.1080/14789450.2020.1855982.

Bruderer, R., Bernhardt, O. M., Gandhi, T., Miladinović, S. M., Cheng, L. Y., Messner, S., Ehrenberger, T., Zanotelli, V., Butscheid, Y., Escher, C., Vitek, O., Rinner, O., and Reiter, L. (2015). Extending the Limits of Quantitative Proteome Profiling with Data-Independent Acquisition and Application to Acetaminophen-Treated Three-Dimensional Liver Microtissues. *Molecular And Cellular Proteomics* 14.5, pp. 1400–1410. DOI: 10.1074/MCP.M114.044305.

Buée, L., Bussière, T., Buée-Scherrer, V., Delacourte, A., and Hof, P. R. (2000). Tau protein isoforms, phosphorylation and role in neurodegenerative disorders. *Brain Research Reviews* 33.1, pp. 95–130. DOI: 10.1016/S0165-0173(00)00019-9.

Burkhart, J. M., Schumbrutzki, C., Wortelkamp, S., Sickmann, A., and Zahedi, R. P. (2012). Systematic and quantitative comparison of digest efficiency and specificity reveals the impact of trypsin quality on MS-based proteomics. *Journal of Proteomics* 75.4, pp. 1454–1462. DOI: 10.1016/J.JPROT.2011.11.016.

Carbonara, K., Andonovski, M., and Coorssen, J. R. (2021). Proteomes Are of Proteoforms: Embracing the Complexity. *Proteomes* 9.3, 38. DOI: 10.3390/PROTEOMES9030038.

Chang, W, Cheng, J, Allaire, J, Sievert, C, Schloerke, B, Xie, Y, Allen, J, McPherson, J, Dipert, A, and Borges, B (2023). shiny: Web Application Framework for R. R package version 1.8.0. URL: https://CRAN.R-project.org/package=shiny.

Charif, D. and Lobry, J. R. (2007). SeqinR 1.0-2: A Contributed Package to the R Project for Statistical Computing Devoted to Biological Sequences Retrieval and Analysis. *Structural Approaches to Sequence Evolution. Biological and Medical Physics, Biomedical Engineering*. Ed. by Bastolla, U., Porto, M., Roman, H. E., and Vendruscolo, M. Springer, Berlin, Heidelberg. Chap. 10, pp. 207–232. DOI: 10.1007/978-3-540-35306-5_10.

Chawade, A., Alexandersson, E., and Levander, F. (2014). Normalyzer: A tool for rapid evaluation of normalization methods for omics data sets. *Journal of Proteome Research* 13.6, pp. 3114–3120. DOI: 10.1021/PR401264N.

Clough, T., Thaminy, S., Ragg, S., Aebersold, R., and Vitek, O. (2012). Statistical protein quantification and significance analysis in label-free LC-MS experiments with complex designs. *BMC Bioinformatics* 13.S6. DOI: 10.1186/1471-2105-13-S16-S6.

Cordella, L. P., Foggia, P., Sansone, C., and Vento, M. (2004). A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.10, pp. 1367–1372. DOI: 10.1109/TPAMI.2004.75.

Cox, J., Hein, M. Y., Luber, C. A., Paron, I., Nagaraj, N., and Mann, M. (2014). Accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction, termed MaxLFQ. *Molecular and Cellular Proteomics* 13.9, pp. 2513–2526. DOI: 10.1074/mcp.M113.031591.

Cox, J. and Mann, M. (2008). MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology* 26.12, pp. 1367–1372. DOI: 10.1038/nbt.1511.

Cox, J., Neuhauser, N., Michalski, A., Scheltema, R. A., Olsen, J. V., and Mann, M. (2011). Andromeda: A Peptide Search Engine Integrated into the MaxQuant Environment. *Journal of Proteome Research* 10.4, pp. 1794–1805. DOI: 10.1021/pr101065j.

Csárdi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems* 1695. URL: https://igraph.org.

D'Costa, J. J., Goldsmith, J. C., Wilson, J. S., Bryan, R. T., and Ward, D. G. (2016). A Systematic Review of the Diagnostic and Prognostic Value of Urinary Protein Biomarkers in Urothelial Bladder Cancer. *Bladder Cancer* 2.3, pp. 301–317. DOI: 10.3233/BLC-160054.

Dahl, D. B., Scott, D., Roosen, C., Magnusson, A., and Swinton, J. (2019). xtable: Export Tables to LaTeX or HTML. R package version 1.8-4. URL: http://xtable.r-forge.r-project.org/.

Degroeve, S., Martens, L., and Jurisica, I. (2013). MS2PIP: a tool for MS/MS peak intensity prediction. *Bioinformatics* 29.24, pp. 3199–3203. DOI: 10.1093/BIOINFORMATICS/BTT544.

Demichev, V., Messner, C. B., Vernardis, S. I., Lilley, K. S., and Ralser, M. (2019). DIA-NN: neural networks and interference correction enable deep proteome coverage in high throughput. *Nature Methods* 17.1, pp. 41–44. DOI: 10.1038/s41592-019-0638-x.

Dermit, M., Peters-Clarke, T. M., Shishkova, E., and Meyer, J. G. (2021). Peptide Correlation Analysis (PeCorA) Reveals Differential Proteoform Regulation. *Journal of Proteome Research* 20.4, pp. 1972–1980. DOI: 10.1021/acs.jproteome.0c00602.

Deutsch, E. W., Bandeira, N., Perez-Riverol, Y., Sharma, V., Carver, J. J., Mendoza, L., Kundu, D. J., Wang, S., Bandla, C., Kamatchinathan, S., Hewapathirana, S., Pullman, B. S., Wertz, J., Sun, Z., Kawano, S., Okuda, S., Watanabe, Y., Maclean, B., Maccoss, M. J., Zhu, Y., Ishihama, Y., and Vizcaíno, J. A. (2023). The ProteomeXchange consortium at 10 years: 2023 update. *Nucleic Acids Research* 51.D1, pp. D1539–D1548. DOI: 10.1093/NAR/GKAC1040.

Deutsch, E. W., Lane, L., Overall, C. M., Bandeira, N., Baker, M. S., Pineau, C., Moritz, R. L., Corrales, F., Orchard, S., Van Eyk, J. E., Paik, Y. K., Weintraub, S. T., Vandenbrouck, Y., and Omenn, G. S. (2019). Human Proteome Project Mass Spectrometry Data Interpretation Guidelines 3.0. *Journal of Proteome Research* 18.12, pp. 4108–4116. DOI: 10.1021/ACS.JPROTEOME.9B00542.

Di Tommaso, P., Chatzou, M., Floden, E. W., Barja, P. P., Palumbo, E., and Notredame, C. (2017). Nextflow enables reproducible computational workflows. *Nature Biotechnology* 35.4, pp. 316–319. DOI: 10.1038/nbt.3820.

Doblmann, J., Dusberger, F., Imre, R., Hudecz, O., Stanek, F., Mechtler, K., and Dürnberger, G. (2019). ApQuant: Accurate Label-Free Quantification by Quality Filtering. *Journal of Proteome Research* 18.1, pp. 535–541. DOI: 10.1021/acs.jproteome.8b00113.

Dost, B., Bandeira, N., Li, X., Shen, Z., Briggs, S., and Bafna, V. (2009). Shared Peptides in Mass Spectrometry Based Protein Quantification. *Research in Computational Molecular Biology. RECOMB 2009. Lecture Notes in Computer Science*. Ed. by Batzoglou, S. Vol. 5541. Springer, Berlin, Heidelberg, pp. 356–371. DOI: 10.1007/978-3-642-02008-7_26.

Dost, B., Bandeira, N., Li, X., Shen, Z., Briggs, S. P., and Bafna, V. (2012). Accurate Mass Spectrometry Based Protein Quantification via Shared Peptides. *Journal of Computational Biology* 19.4, p. 337. DOI: 10.1089/CMB.2009.0267.

Dou, Y., Liu, Y., Yi, X., Olsen, L. K., Zhu, H., Gao, Q., Zhou, H., and Zhang, B. (2023). SEPepQuant enhances the detection of possible isoform regulations in shotgun proteomics. *Nature Communications* 14.1, pp. 1–15. DOI: 10.1038/s41467-023-41558-2.

Dowle, M. and Srinivasan, A. (2023). data.table: Extension of 'data.frame'. R package version 1.14.8. URL: https://CRAN.R-project.org/package=data.table.

Duong, V. A., Park, J. M., Lim, H. J., and Lee, H. (2021). Proteomics in Forensic Analysis: Applications for Human Samples. *Applied Sciences* 11.8, 3393. DOI: 10.3390/APP11083393.

Dwivedi, A. K., Mallawaarachchi, I., and Alvarado, L. A. (2017). Analysis of small sample size studies using nonparametric bootstrap test with pooled resampling method. *Statistics in Medicine* 36.14, pp. 2187–2205. DOI: 10.1002/SIM.7263.

Efron, B. and Narasimhan, B. (2020). The Automatic Construction of Bootstrap Confidence Intervals. *Journal of Computational and Graphical Statistics* 29.3, pp. 608–619. ISSN: 15372715. DOI: 10.1080/10618600.2020.1714633.

Egert, J., Brombacher, E., Warscheid, B., and Kreutz, C. (2021). DIMA: Data-Driven Selection of an Imputation Algorithm. *Journal of Proteome Research* 20.7, pp. 3489–3496. DOI: 10.1021/ACS.JPROTEOME.1C00119.

Elias, J. E. and Gygi, S. P. (2010). An Overview of Label-Free Quantitation Methods in Proteomics by Mass Spectrometry. *Methods in Molecular Biology*. Ed. by Hubbard, S and Jones, A. Vol. 604. August. Humana Press, pp. 55–71. DOI: 10.1007/978-1-60761-444-9_5.

Elias, J. E., Haas, W., Faherty, B. K., and Gygi, S. P. (2005). Comparative evaluation of mass spectrometry platforms used in large-scale proteomics investigations. *Nature Methods* 2.9, pp. 667–675. DOI: 10.1038/nmeth785.

Eng, J. K., Fischer, B., Grossmann, J., and MacCoss, M. J. (2008). A fast SEQUEST cross correlation algorithm. *Journal of Proteome Research* 7.10, pp. 4598–4602. DOI: 10.1021/PR800420S.

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, pp. 976–989. DOI: 10.1016/1044-0305(94)80016-2.

Erhard, F. and Zimmer, R. (2012). Detecting outlier peptides in quantitative high-throughput mass spectrometry data. *Journal of Proteomics* 75.11, pp. 3230–3239. DOI: 10.1016/J.JPROT.2012.03.032.

Forshed, J., Johansson, H. J., Pernemalm, M., Branca, R. M., Sandberg, A. S., and Lehtiö, J. (2011). Enhanced information output from shotgun proteomics data by protein quantification and peptide quality control (PQPQ). *Molecular and Cellular Proteomics* 10.10, pp. 1–9. DOI: 10.1074/mcp.M111.010264.

Fraser, D. D., Cepinskas, G., Patterson, E. K., Slessarev, M., Martin, C., Daley, M., Patel, M. A., Miller, M. R., O'Gorman, D. B., Gill, S. E., Pare, G., Prassas, I., and Diamandis, E. (2020). Novel Outcome Biomarkers Identified with Targeted Proteomic Analyses of Plasma from Critically Ill Coronavirus Disease 2019 Patients. *Critical Care Explorations* 2.9, E0189. DOI: 10.1097/CCE.0000000000000189.

Gadbury, G. L., Xiang, Q., Yang, L., Barnes, S., Page, G. P., and Allison, D. B. (2008). Evaluating statistical methods using plasmode data sets in the age of massive public databases: An illustration using false discovery rates. *PLoS Genetics* 4.6, e1000098. DOI: 10.1371/journal.pgen.1000098.

Gardner, M. L. and Freitas, M. A. (2021). Multiple imputation approaches applied to the missing value problem in bottom-up proteomics. *International Journal of Molecular Sciences* 22.17. DOI: 10.3390/IJMS22179650/S1.

Geis-Asteggiante, L., Ostrand-Rosenberg, S., Fenselau, C., and Edwards, N. J. (2016). Evaluation of Spectral Counting for Relative Quantitation of Proteoforms in Top-Down Proteomics. *Analytical Chemistry* 88.22, pp. 10900–10907. DOI: 10.1021/ACS.ANALCHEM.6B02151.

Gershon, P. D. (2014). Cleaved and missed sites for trypsin, Lys-C, and Lys-N can be predicted with high confidence on the basis of sequence context. *Journal of Proteome Research* 13.2, pp. 702–709. ISSN: 15353893. DOI: https://doi.org/10.1021/pr400802z.

Gerster, S. and Buehlmann, P. (2013). protiq: Protein (identification and) quantification based on peptide evidence. R package version 1.2. URL: https://CRAN.R-project.org/package=protiq.

Gerster, S., Kwon, T., Ludwig, C., Matondo, M., Vogel, C., Marcotte, E. M., Aebersold, R., and Buhlmann, P. (2014). Statistical approach to protein quantification. *Molecular and Cellular Proteomics* 13.2, pp. 666–677. DOI: 10.1074/mcp.M112.025445.

Gerster, S., Qeli, E., Ahrens, C. H., and Bühlmann, P. (2010). Protein and gene model inference based on statistical modeling in k-partite graphs. *Proceedings of the National Academy of Sciences of the United States of America* 107.27, pp. 12101–12106. DOI: https://doi.org/10.1073/pnas.0907654107.

Gessulat, S., Schmidt, T., Zolg, D. P., Samaras, P., Schnatbaum, K., Zerweck, J., Knaute, T., Rechenberger, J., Delanghe, B., Huhmer, A., Reimer, U., Ehrlich, H. C., Aiche, S., Kuster, B., and Wilhelm, M. (2019). Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature Methods* 16.6, pp. 509–518. DOI: 10.1038/s41592-019-0426-7.

Ghalanos, A. and Theussl, S. (2015). Rsolnp: General Non-linear Optimization Using Augmented Lagrange Multiplier Method. R package version 1.16. URL: https://cran.r-project.org/package=Rsolnp.

Goeminne, L. J., Gevaert, K., and Clement, L. (2016). Peptide-level Robust Ridge Regression Improves Estimation, Sensitivity, and Specificity in Data-dependent Quantitative Label-free Shotgun Proteomics. *Molecular and Cellular Proteomics* 15.2, pp. 657–668. DOI: 10.1074/MCP.M115.055897.

Goeminne, L. J., Sticker, A., Martens, L., Gevaert, K., and Clement, L. (2020). MSqRob Takes the Missing Hurdle: Uniting Intensity- and Count-Based Proteomics. *Analytical Chemistry* 92.9, pp. 6278–6287. DOI: 10.1021/ACS.ANALCHEM.9B04375.

Griss, J., Jones, A. R., Sachsenberg, T., Walzer, M., Gatto, L., Hartler, J., Thallinger, G. G., Salek, R. M., Steinbeck, C., Neuhauser, N., Cox, J., Neumann, S., Fan, J., Reisinger, F., Xu, Q. W., Del Toro, N., Pérez-Riverol, Y., Ghali, F., Bandeira, N., Xenarios, I., Kohlbacher, O., Vizcaíno, J. A., and Hermjakob, H. (2014). The mzTab data exchange format: Communicating mass-spectrometry-based proteomics and metabolomics experimental results to a wider audience. *Molecular and Cellular Proteomics* 13.10, pp. 2765–2775. DOI: 10.1074/mcp.O113.036681.

Gupta, N. and Pevzner, P. A. (2009). False discovery rates of protein identifications: A strike against the two-peptide rule. *Journal of Proteome Research* 8.9, pp. 4173–4181. DOI: 10.1021/pr9004794.

Hahne, F., Huber, W., Gentleman, R., and Falcon, S. (2008). *Bioconductor Case Studies*. 1st ed. Springer, New York. DOI: 10.1007/978-0-387-77240-0.

Haw, R., Hermjakob, H., D'Eustachio, P., and Stein, L. (2011). Reactome pathway analysis to enrich biological discovery in proteomics data sets. *PROTEOMICS* 11.18, pp. 3598–3613. DOI: 10.1002/PMIC.201100066.

Hochreiter, S., Clevert, D. A., and Obermayer, K. (2006). A new summarization method for affymetrix probe level data. *Bioinformatics* 22.8, pp. 943–949. DOI: 10.1093/BIOINFORMATICS/BTL033.

Huang, T., Wang, J., Yu, W., and He, Z. (2012). Protein inference: a review. *Briefings in Bioinformatics* 13.5, pp. 586–614. DOI: 10.1093/BIB/BBS004.

Huber, W., Von Heydebreck, A., Sültmann, H., Poustka, A., and Vingron, M. (2002). Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics* 18.suppl_1, S96–S104. DOI: 10.1093/BIOINFORMATICS.

Ishihama, Y., Oda, Y., Tabata, T., Sato, T., Nagasu, T., Rappsilber, J., and Mann, M. (2005). Exponentially modified protein abundance index (emPAI) for estimation of absolute protein amount in proteomics by the number of sequenced peptides per protein. *Molecular and Cellular Proteomics* 4.9, pp. 1265–1272. DOI: 10.1074/MCP.M500061-MCP200.

Jin, S., Daly, D. S., Springer, D. L., and Miller, J. H. (2008). The effects of shared peptides on protein quantitation in label-free proteomics by LC/MS/MS. *Journal of Proteome Research* 7.1, pp. 164–169. DOI: 10.1021/PR0704175.

Jones, A. R. (2016). Protein Inference and Grouping. *New Developments in Mass Spectrometry, Proteome Informatics*. The Royal Society of Chemistry, pp. 93–115. DOI: 10.1039/9781782626732-00093.

Jorrin Novo, J. V. (2021). Proteomics and plant biology: contributions to date and a look towards the next decade. *Expert Review of Proteomics* 18.2, pp. 93–103. DOI: 10.1080/14789450.2021.1910028.

Junge, A. (2022). Automatisierte Erstellung und Auswertung von bipartiten Peptid-Protein-Graphen aus verschiedenen Proteinsequenzdatenbanken. Bachelor's thesis, Ruhr-Universität Bochum.

Kalaycik, B. (2023). Einfluss von missed cleavages auf bipartite Peptid-Protein-Graphen und Peptid-Ratios. Final internship presentation, Ruhr-Universität Bochum.

Käll, L., Canterbury, J. D., Weston, J., Noble, W. S., and MacCoss, M. J. (2007). Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature Methods* 4.11, pp. 923–925. DOI: 10.1038/nmeth1113.

Karch, K. R., Snyder, D. T., Harvey, S. R., and Wysocki, V. H. (2022). Native Mass Spectrometry: Recent Progress and Remaining Challenges. *Annual Review of Biophysics* 51, pp. 157–179. DOI: 10.1146/ANNUREV-BIOPHYS-092721-085421.

Karpievitch, Y. V., Dabney, A. R., and Smith, R. D. (2012). Normalization and missing value imputation for label-free LC-MS analysis. *BMC Bioinformatics* 13.S5. DOI: 10.1186/1471-2105-13-S16-S5.

Karpievitch, Y., Stanley, J., Taverner, T., Huang, J., Adkins, J. N., Ansong, C., Heffron, F., Metz, T. O., Qian, W. J., Yoon, H., Smith, R. D., and Dabney, A. R. (2009). A statistical framework for protein quantitation in bottom-up MS-based proteomics. *Bioinformatics* 25.16, pp. 2028–2034. DOI: 10.1093/BIOINFORMATICS/BTP362.

Kassambara, A. (2023). ggpubr: 'ggplot2' Based Publication Ready Plots. R package version 0.6.0. URL: https://CRAN.R-project.org/package=ggpubr.

Kettenbach, A. N., Rush, J., and Gerber, S. A. (2011). Absolute quantification of protein and post-translational modification abundance with stable isotope–labeled synthetic peptides. *Nature Protocols* 6.2, pp. 175–186. DOI: 10.1038/nprot.2010.196.

Kumar, D., Yadav, A. K., and Dash, D. (2017). Choosing an optimal database for protein identification from tandem mass spectrometry data. *Proteome Bioinformatics. Methods in Molecular Biology*. Vol. 1549. Humana Press, New York, pp. 17–29. DOI: 10.1007/978-1-4939-6740-7_3.

Kuznetsova, K. G., Solovyeva, E. M., Kuzikov, A. V., Gorshkov, M. V., and Moshkovskii, S. A. (2020). Modification of Cysteine Residues for Mass Spectrometry-Based Proteomic Analysis: Facts and Artifacts. *Biochemistry (Moscow) Supplement Series B: Biomedical Chemistry* 14.3, pp. 204–215. DOI: 10.1134/S1990750820030087.

Lan, K. and Jorgenson, J. W. (2001). A hybrid of exponential and gaussian functions as a simple model of asymmetric chromatographic peaks. *Journal of Chromatography A* 915.1-2, pp. 1–13. DOI: 10.1016/S0021-9673(01)00594-5.

Lang, M., Bischl, B., and Surmann, D. (2017). batchtools: Tools for R to work on batch systems. *Journal of Open Source Software* 2.10, 135. DOI: 10.21105/JOSS.00135.

Lawless, C. and Hubbard, S. J. (2012). Prediction of Missed Proteolytic Cleavages for the Selection of Surrogate Peptides for Quantitative Proteomics. *OMICS : a Journal of Integrative Biology* 16.9, p. 449. DOI: 10.1089/OMI.2011.0156.

Lazar, C., Gatto, L., Ferro, M., Bruley, C., and Burger, T. (2016). Accounting for the Multiple Natures of Missing Values in Label-Free Quantitative Proteomics Data Sets to Compare Imputation Strategies. *Journal of Proteome Research* 15.4, pp. 1116–1125. DOI: 10.1021/ACS.JPROTEOME.5B00981.

Li, Y. F., Arnold, R. J., Li, Y., Radivojac, P., Sheng, Q., and Tang, H. (2009). A Bayesian Approach to Protein Inference Problem in Shotgun Proteomics. *Journal of Computational Biology* 16.8, pp. 1183–1193. DOI: 10.1089/CMB.2009.0018.

Li, Y. F. and Radivojac, P. (2012). Computational approaches to protein inference in shotgun proteomics. *BMC Bioinformatics* 13.16, pp. 1–19. DOI: 10.1186/1471-2105-13-S16-S4.

Liigand, P., Kaupmees, K., and Kruve, A. (2019). Influence of the amino acid composition on the ionization efficiencies of small peptides. *Journal of Mass Spectrometry* 54.6, pp. 481–487. DOI: 10.1002/JMS.4348.

Lu, P., Vogel, C., Wang, R., Yao, X., and Marcotte, E. M. (2007). Absolute protein expression profiling estimates the relative contributions of transcriptional and translational regulation. *Nature Biotechnology* 25.1, pp. 117–124. DOI: 10.1038/NBT1270.

Lukasse, P. N. and America, A. H. (2014). Protein inference using peptide quantification patterns. *Journal of Proteome Research* 13.7, pp. 3191–3199. DOI: https://doi.org/10.1021/pr401072g.

Lyngbakken, M. N., Myhre, P. L., Røsjø, H., and Omland, T. (2019). Novel biomarkers of cardiovascular disease: Applications in clinical practice. *Critical Reviews in Clinical Laboratory Sciences* 56.1, pp. 33–60. DOI: 10.1080/10408363.2018.1525335.

Malioutov, D., Chen, T., Airoldi, E., Jaffe, J., Budnik, B., and Slavov, N. (2019). Quantifying Homologous Proteins and Proteoforms. *Molecular and Cellular Proteomics* 18.1, pp. 162–168. DOI: 10.1074/MCP.TIR118.000947.

Matafora, V. and Bachi, A. (2021). Missing Value Monitoring to Address Missing Values in Quantitative Proteomics. *Quantitative Methods in Proteomics. Methods in Molecular Biology*. Ed. by Marcus, K., Eisenacher, M., and Sitek, B. Vol. 2228. Humana, New York, pp. 401–408. DOI: 10.1007/978-1-0716-1024-4_27.

Megger, D. A., Pott, L. L., Ahrens, M., Padden, J., Bracht, T., Kuhlmann, K., Eisenacher, M., Meyer, H. E., and Sitek, B. (2014). Comparison of label-free and label-based strategies for proteome analysis of hepatoma cell lines. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics* 1844.5, pp. 967–976. DOI: 10.1016/J.BBAPAP.2013.07.017.

Merck KGaA (2023). *UPS1 & UPS2 Proteomic Standards*. URL: https://www.sigmaaldrich.com/DE/de/technical-documents/technical-article/protein-biology/protein-mass-spectrometry/ups1-and-ups2-proteomic (Accessed on 20.10.2023).

Meyer-Arendt, K., Old, W. M., Houel, S., Renganathan, K., Eichelberger, B., Resing, K. A., and Ahn, N. G. (2011). IsoformResolver: A peptide-centric algorithm for protein inference. *Journal of Proteome Research* 10.7, pp. 3060–3075. DOI: https://doi.org/10.1021/pr200039p.

Milac, T. I., Randolph, T. W., and Wang, P. (2012). Analyzing LC-MS/MS data by spectral count and ion abundance: two case studies. *Statistics and Its Interface* 5.1, pp. 75–87. DOI: 10.4310/SII.2012.V5.N1.A7.

Müller-Esterl, W. (2018). *Biochemie. Eine Einführung für Mediziner und Naturwissenschaftler*. Springer Berlin Heidelberg, Berlin, Heidelberg. DOI: 10.1007/978-3-662-54851-6.

Nahnsen, S., Bielow, C., Reinert, K., and Kohlbacher, O. (2013). Tools for label-free peptide quantification. *Molecular and Cellular Proteomics* 12.3, pp. 549–556. DOI: 10.1074/MCP.R112.025163.

Nakayasu, E. S., Gritsenko, M., Piehowski, P. D., Gao, Y., Orton, D. J., Schepmoes, A. A., Fillmore, T. L., Frohnert, B. I., Rewers, M., Krischer, J. P., Ansong, C., Suchy-Dicey, A. M., Evans-Molina, C., Qian, W. J., Webb-Robertson, B. J. M., and Metz, T. O. (2021). Tutorial: best practices and considerations for mass-spectrometry-based protein biomarker discovery and validation. *Nature Protocols* 16.8, pp. 3737–3760. DOI: 10.1038/s41596-021-00566-6.

National Center for Biotechnology Information (2014). Saccharomyces cerevisiae S288C genome assembly R64 - NCBI - NLM. URL: https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_000146045.2/ (Accessed on 20.02.2024).

National Center for Biotechnology Information (2020). Mus musculus genome assembly GRCm39 - NCBI - NLM. URL: https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_000001635.27/ (Accessed on 20.02.2024).

National Center for Biotechnology Information (2022). Homo sapiens genome assembly GRCh38.p14 - NCBI - NLM. URL: https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_000001405.40/ (Accessed on 20.02.2024).

Nesvizhskii, A. I. and Aebersold, R. (2005). Interpretation of Shotgun Proteomic Data. *Molecular And Cellular Proteomics* 4.10, pp. 1419–1440. DOI: 10.1074/MCP.R500012-MCP200.

Nesvizhskii, A. I., Keller, A., Kolker, E., and Aebersold, R. (2003). A statistical model for identifying proteins by tandem mass spectrometry. *Analytical Chemistry* 75.17, pp. 4646–4658. DOI: https://doi.org/10.1021/ac0341261.

Neuhaus, K. L. (2023). Subgraph matching in bipartite peptide-protein graphs. Bachelor's thesis, Ruhr-Universität Bochum.

Neuhaus, K., Faiz, S., and Achterfeldt, N. (2023). Automated generation and visualization of bipartite peptide-protein graphs from quantitative proteomics data sets. Study project report, Ruhr-Universität Bochum.

Ong, S. E., Blagoev, B., Kratchmarova, I., Kristensen, D. B., Steen, H., Pandey, A., and Mann, M. (2002). Stable Isotope Labeling by Amino Acids in Cell Culture, SILAC, as a Simple and Accurate Approach to Expression Proteomics. *Molecular and Cellular Proteomics* 1.5, pp. 376–386. DOI: 10.1074/MCP.M200025-MCP200.

Ong, S. E. and Mann, M. (2005). Mass spectrometry–based proteomics turns quantitative. *Nature Chemical Biology* 1.5, pp. 252–262. DOI: 10.1038/nchembio736.

Orsburn, B. C. (2021). Proteome Discoverer—A Community Enhanced Data Processing Suite for Protein Informatics. *Proteomes* 9.1. DOI: 10.3390/PROTEOMES9010015.

Pan, C., Kora, G., McDonald, W. H., Tabb, D. L., VerBerkmoes, N. C., Hurst, G. B., Pelletier, D. A., Samatova, N. F., and Hettich, R. L. (2006). ProRata: A quantitative proteomics program for accurate protein abundance ratio estimation with confidence interval evaluation. *Analytical Chemistry* 78.20, pp. 7121–7131. DOI: https://doi.org/10.1021/ac060654b.

Parnetti, L., Castrioto, A., Chiasserini, D., Persichetti, E., Tambasco, N., El-Agnaf, O., and Calabresi, P. (2013). Cerebrospinal fluid biomarkers in Parkinson disease. *Nature Reviews Neurology* 9.3, pp. 131–140. DOI: 10.1038/nrneurol.2013.10.

Pavlopoulos, G. A., Kontou, P. I., Pavlopoulou, A., Bouyioukos, C., Markou, E., and Bagos, P. G. (2018). Bipartite graphs in systems biology and medicine: a survey of methods and applications. *GigaScience* 7.4, pp. 1–31. DOI: 10.1093/GIGASCIENCE/GIY014.

Perez-Riverol, Y., Bai, J., Bandla, C., García-Seisdedos, D., Hewapathirana, S., Kamatchinathan, S., Kundu, D. J., Prakash, A., Frericks-Zipper, A., Eisenacher, M., Walzer, M., Wang, S., Brazma, A., and Vizcaíno, J. A. (2022). The PRIDE database resources in 2022: a hub for mass spectrometry-based proteomics evidences. *Nucleic Acids Research* 50.D1, pp. D543–D552. DOI: 10.1093/NAR/GKAB1038.

Pfeuffer, J., Sachsenberg, T., Dijkstra, T. M., Serang, O., Reinert, K., and Kohlbacher, O. (2020). EPIFANY: A Method for Efficient High-Confidence Protein Inference. *Journal of Proteome Research* 19.3, pp. 1060–1072. DOI: https://doi.org/10.1021/acs.jproteome.9b00566.

Plubell, D. L., Käll, L., Webb-Robertson, B. J., Bramer, L. M., Ives, A., Kelleher, N. L., Smith, L. M., Montine, T. J., Wu, C. C., and Maccoss, M. J. (2022). Putting Humpty Dumpty Back Together Again: What Does Protein Quantification Mean in Bottom-Up Proteomics? *Journal of Proteome Research* 21.4, pp. 891–898. DOI: 10.1021/acs.jproteome.1c00894.

Posit team (2023). RStudio: Integrated Development Environment for R. Boston, MA. URL: http://www.posit.co/.

Qeli, E., Omasits, U., Goetze, S., Stekhoven, D. J., Frey, J. E., Basler, K., Wollscheid, B., Brunner, E., and Ahrens, C. H. (2014). Improved prediction of peptide detectability for targeted proteomics using a rank-based algorithm and organism-specific data. *Journal of Proteomics* 108, pp. 269–283. DOI: 10.1016/J.JPROT.2014.05.011.

R Core Team (2023). R: A Language and Environment for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Rahman, M. S. (2017). *Basic Graph Theory. Undergraduate Topics in Computer Science*. Undergraduate Topics in Computer Science. Springer, Cham. DOI: 10.1007/978-3-319-49475-3.

Ramus, C., Hovasse, A., Marcellin, M., Hesse, A. M., Mouton-Barbosa, E., Bouyssié, D., Vaca, S., Carapito, C., Chaoui, K., Bruley, C., Garin, J., Cianférani, S., Ferro, M., Dorssaeler, A. V., Burlet-Schiltz, O., Schaeffer, C., Couté, Y., and Peredo, A. Gonzalez de (2016a). Spiked proteomic standard dataset for testing label-free quantitative software and statistical methods. *Data in Brief* 6, pp. 286–294. DOI: 10.1016/j.dib.2015.11.063.

Ramus, C., Hovasse, A., Marcellin, M., Hesse, A. M., Mouton-Barbosa, E., Bouyssié, D., Vaca, S., Carapito, C., Chaoui, K., Bruley, C., Garin, J., Cianférani, S., Ferro, M., Van Dorssaeler, A., Burlet-Schiltz, O., Schaeffer, C., Couté, Y., and Peredo, A. Gonzalez de (2016b). Benchmarking quantitative label-free LC-MS data processing workflows using a complex spiked proteomic standard dataset. *Journal of Proteomics* 132, pp. 51–62. DOI: 10.1016/j.jprot.2015.11.011.

Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., and Smyth, G. K. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research* 43.7, e47. DOI: 10.1093/NAR/GKV007.

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J. C., and Müller, M. (2011). pROC: An open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics* 12.1, pp. 1–8. DOI: 10.1186/1471-2105-12-77.

Rozanova, S., Uszkoreit, J., Schork, K., Serschnitzki, B., Eisenacher, M., Tönges, L., Barkovits-Boeddinghaus, K., and Marcus, K. (2023). Quality Control—A Stepchild in Quantitative Proteomics: A Case Study for the Human CSF Proteome. *Biomolecules* 13.3, 491. DOI: 10.3390/BIOM13030491.

Röst, H. L., Sachsenberg, T., Aiche, S., Bielow, C., Weisser, H., Aicheler, F., Andreotti, S., Ehrlich, H. C., Gutenbrunner, P., Kenar, E., Liang, X., Nahnsen, S., Nilse, L., Pfeuffer, J., Rosenberger, G., Rurik, M., Schmitt, U., Veit, J., Walzer, M., Wojnar, D., Wolski, W. E., Schilling, O., Choudhary, J. S., Malmström, L., Aebersold, R., Reinert, K., and Kohlbacher, O. (2016). OpenMS: a flexible open-source software platform for mass spectrometry data analysis. *Nature methods* 13 (9), pp. 741–748. DOI: 10.1038/NMETH.3959.

Saltzman, A. B., Leng, M., Bhatt, B., Singh, P., Chan, D. W., Dobrolecki, L., Chandrasekaran, H., Choi, J. M., Jain, A., Jung, S. Y., Lewis, M. T., Ellis, M. J., and Malovannaya, A. (2018). GpGrouper: A peptide grouping algorithm for gene-centric inference and quantitation of bottom-up proteomics data. *Molecular and Cellular Proteomics* 17.11, pp. 2270–2283. DOI: 10.1074/mcp.TIR118.000850.

Schauberger, P and Walker, A (2023). openxlsx: Read, Write and Edit xlsx Files. R package version 4.2.5.2. URL: http://CRAN.R-project.org/package=openxlsx.

Schork, K. (2016). Verbesserte Annotation von Massenspektren mit Algorithmen der Clusteranalyse. Master's thesis, Technische Universität Dortmund.

Schork, K., Turewicz, M., Uszkoreit, J., Rahnenführer, J., and Eisenacher, M. (2022). Characterization of peptide-protein relationships in protein ambiguity groups via bipartite graphs. *PLOS ONE* 17.10, e0276401. DOI: 10.1371/journal.pone.0276401.

Schwanhäusser, B., Busse, D., Li, N., Dittmar, G., Schuchhardt, J., Wolf, J., Chen, W., and Selbach, M. (2011). Global quantification of mammalian gene expression control. *Nature* 473.7347, pp. 337–342. DOI: 10.1038/nature10098.

Shen, M., Chang, Y. T., Wu, C. T., Parker, S. J., Saylor, G., Wang, Y., Yu, G., Van Eyk, J. E., Clarke, R., Herrington, D. M., and Wang, Y. (2022). Comparative assessment and novel strategy on methods for imputing proteomics data. *Scientific Reports* 12.1, pp. 1–11. DOI: 10.1038/s41598-022-04938-0.

Shi, Y., Xiang, R., Horváth, C., and Wilkins, J. A. (2004). The role of liquid chromatography in proteomics. *Journal of Chromatography A* 1053.1-2, pp. 27–36. DOI: 10.1016/J.CHROMA.2004.07.044.

Shuken, S. R. (2023). An Introduction to Mass Spectrometry-Based Proteomics. *Journal of Proteome Research* 22.7, pp. 2151–2171. DOI: https://doi.org/10.1021/acs.jproteome.2c00838.

Silva, J. C., Gorenstein, M. V., Li, G. Z., Vissers, J. P., and Geromanos, S. J. (2006). Absolute quantification of proteins by LCMSE: A virtue of parallel MS acquisition. *Molecular and Cellular Proteomics* 5.1, pp. 144–156. DOI: 10.1074/mcp.M500230-MCP200.

Smith, L. M., Kelleher, N. L., Linial, M., Goodlett, D., Langridge-Smith, P., Goo, Y. A., Safford, G., Bonilla, L., Kruppa, G., Zubarev, R., Rontree, J., Chamot-Rooke, J., Garavelli, J., Heck, A., Loo, J., Penque, D., Hornshaw, M., Hendrickson, C., Pasa-Tolic, L., Borchers, C., Chan, D., Young, N., Agar, J., Masselon, C., Gross, M., McLafferty, F., Tsybin, Y., Ge, Y., Sanders, I., Langridge, J., Whitelegge, J., and Marshall, A. (2013). Proteoform: a single term describing protein complexity. *Nature Methods* 10.3, pp. 186–187. DOI: 10.1038/nmeth.2369.

Solnon, C. (2010). AllDifferent-based filtering for subgraph isomorphism. *Artificial Intelligence* 174.12-13, pp. 850–864. DOI: 10.1016/J.ARTINT.2010.05.002.

Solymos, P and Zawadzki, Z (2023). pbapply: Adding Progress Bar to '*apply' Functions. R package version 1.7-2. URL: http://CRAN.R-project.org/package=pbapply.

Sticker, A., Goeminne, L., Martens, L., and Clement, L. (2020). Robust Summarization and Inference in Proteome-wide Label-free Quantification. *Molecular and Cellular Proteomics* 19.7, pp. 1209–1219. DOI: 10.1074/MCP.RA119.001624.

Sturm, M. and Kohlbacher, O. (2009). TOPPView: An open-source viewer for mass spectrometry data. *Journal of Proteome Research* 8.7, pp. 3760–3763. DOI: 10.1021/PR900171M.

Swaney, D. L., Wenger, C. D., and Coon, J. J. (2010). The value of using multiple proteases for large-scale mass spectrometry-based proteomics. *Journal of Proteome Research* 9.3, p. 1323. DOI: 10.1021/PR900863U.

Teleman, J., Chawade, A., Sandin, M., Levander, F., and Malmström, J. (2016). Dinosaur: A Refined Open-Source Peptide MS Feature Detector. *Journal of Proteome Research* 15.7, pp. 2143–2151. DOI: https://doi.org/10.1021/acs.jproteome.6b00016.

The, M., Edfors, F., Perez-Riverol, Y., Payne, S. H., Hoopmann, M. R., Palmblad, M., Forsström, B., and Käll, L. (2018). A Protein Standard That Emulates Homology for the Characterization of Protein Inference Algorithms. *Journal of Proteome Research* 17.5, pp. 1879–1886. DOI: 10.1021/ACS.JPROTEOME.7B00899.

The, M. and Käll, L. (2019). Integrated Identification and Quantification Error Probabilities for Shotgun Proteomics. *Molecular and Cellular Proteomics* 18.3, pp. 561–570. DOI: 10.1074/MCP.RA118.001018.

Thompson, A., Schäfer, J., Kuhn, K., Kienle, S., Schwarz, J., Schmidt, G., Neumann, T., and Hamon, C. (2003). Tandem mass tags: A novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS. *Analytical Chemistry* 75.8, pp. 1895–1904. DOI: https://doi.org/10.1021/ac0262560.

Truong, P., The, M., and Käll, L. (2023). Triqler for Protein Summarization of Data from Data-Independent Acquisition Mass Spectrometry. *Journal of Proteome Research* 22.4, pp. 1359–1366. DOI: https://doi.org/10.1021/acs.jproteome.2c00607.

Tsai, T. H., Choi, M., Banfai, B., Liu, Y., MacLean, B. X., Dunkley, T., and Vitek, O. (2020). Selection of features with consistent profiles improves relative protein quantification in mass spectrometry experiments. *Molecular and Cellular Proteomics* 19.6, pp. 944–959. DOI: 10.1074/mcp.ra119.001792.

Uhlén, M., Fagerberg, L., Hallström, B. M., Lindskog, C., Oksvold, P., Mardinoglu, A., Sivertsson, A., Kampf, C., Sjöstedt, E., Asplund, A., Olsson, I. M., Edlund, K., Lundberg, E., Navani, S., Szigyarto, C. A. K., Odeberg, J., Djureinovic, D., Takanen, J. O., Hober, S., Alm, T., Edqvist, P. H., Berling, H., Tegel, H., Mulder, J., Rockberg, J., Nilsson, P., Schwenk, J. M., Hamsten, M., Von Feilitzen, K., Forsberg, M., Persson, L., Johansson, F., Zwahlen, M., Von Heijne, G., Nielsen, J., and Pontén, F. (2015). Tissue-based map of the human proteome. *Science* 347.6220. DOI: 10.1126/SCIENCE.1260419.

UniProt (2024a). Alternative products. URL: https://www.uniprot.org/help/alternative_products (Accessed on 22.02.2024).

UniProt (2024b). NEMP1 - Nuclear envelope integral membrane protein 1 - Homo sapiens (Human). URL: https://www.uniprot.org/uniprotkb/O14524/entry (Accessed on 22.02.2024).

UniProt (2024c). What is the canonical sequence? Are all isoforms described in one entry? URL: https://www.uniprot.org/help/canonical_and_isoforms (Accessed on 24.04.2024).

Uszkoreit, J., Barkovits, K., Pacharra, S., Pfeiffer, K., Steinbach, S., Marcus, K., and Eisenacher, M. (2022). Dataset containing physiological amounts of spike-in proteins into murine C2C12 background as a ground truth quantitative LC-MS/MS reference. *Data in Brief* 43.108435. DOI: 10.1016/j.dib.2022.108435.

Uszkoreit, J., Maerkens, A., Perez-Riverol, Y., Meyer, H. E., Marcus, K., Stephan, C., Kohlbacher, O., and Eisenacher, M. (2015). PIA: An Intuitive Protein Inference Engine with a Web-Based User Interface. *Journal of Proteome Research* 14.7, pp. 2988–2997. DOI: https://doi.org/10.1021/acs.jproteome.5b00121.

Välikangas, T., Suomi, T., and Elo, L. L. (2018). A systematic evaluation of normalization methods in quantitative label-free proteomics. *Briefings in Bioinformatics* 19.1, pp. 1–11. DOI: 10.1093/BIB/BBW095.

Valkenborg, D., Mertens, I., Lemière, F., Witters, E., and Burzykowski, T. (2012). The isotopic distribution conundrum. *Mass Spectrometry Reviews* 31.1, pp. 96–109. DOI: 10.1002/MAS.20339.

Wagner, G., Maia, G. A., Barr, J. R., and Moura, H. (2024). Proteomics Fundamentals and Applications in Microbiology. *Recent Advancements in the Diagnosis of Human Disease*, pp. 209–236. DOI: 10.1201/9781003438595-7.

Walzer, M., Qi, D., Mayer, G., Uszkoreit, J., Eisenacher, M., Sachsenberg, T., Gonzalez-Galarza, F. F., Fan, J., Bessant, C., Deutsch, E. W., Reisinger, F., Vizcaíno, J. A., Medina-Aunon, J. A., Albar, J. P., Kohlbacher, O., and Jones, A. R. (2013). The mzQuantML data standard for mass spectrometry-based quantitative studies in proteomics. *Molecular and Cellular Proteomics* 12.8, pp. 2332–2340. DOI: 10.1074/mcp.O113.028506.

Wang, Y., Ahn, T. H., Li, Z., and Pan, C. (2013). Sipros/ProRata: a versatile informatics system for quantitative community proteomics. *Bioinformatics* 29.16, pp. 2064–2065. DOI: 10.1093/BIOINFORMATICS/BTT329.

Webb-Robertson, B. J. M., Wiberg, H. K., Matzke, M. M., Brown, J. N., Wang, J., McDermott, J. E., Smith, R. D., Rodland, K. D., Metz, T. O., Pounds, J. G., and Waters, K. M. (2015). Review, evaluation, and discussion of the challenges of missing value imputation for mass spectrometry-based label-free global proteomics. *Journal of Proteome Research* 14.5, pp. 1993–2001. DOI: 10.1021/PR501138H.

Wei, J., Zaika, E., and Zaika, A. (2012). P53 family: Role of protein isoforms in human cancer. *Journal of Nucleic Acids* 2012.687359. DOI: 10.1155/2012/687359.

Wei, R., Wang, J., Jia, E., Chen, T., Ni, Y., and Jia, W. (2018). GSimp: A Gibbs sampler based left-censored missing value imputation approach for metabolomics studies. *PLOS Computational Biology* 14.1, e1005973. DOI: 10.1371/JOURNAL.PCBI.1005973.

Weisser, H. and Choudhary, J. S. (2017). Targeted Feature Detection for Data-Dependent Shotgun Proteomics. *Journal of Proteome Research* 16.8, pp. 2964–2974. DOI: 10.1021/ACS.JPROTEOME.7B00248.

Weisser, H., Nahnsen, S., Grossmann, J., Nilse, L., Quandt, A., Brauer, H., Sturm, M., Kenar, E., Kohlbacher, O., Aebersold, R., and Malmström, L. (2013). An automated pipeline for high-throughput label-free quantitative proteomics. *Journal of Proteome Research* 12.4, pp. 1628–1644. DOI: 10.1021/PR300992U.

Wickham, H (2023). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.5.1. URL: https://CRAN.R-project.org/package=stringr.

Wickham, H. (2016). *ggplot2*. Use R! Springer International Publishing, Cham. DOI: 10.1007/978-3-319-24277-4.

Wickham, H., Averick, M., Bryan, J., Chang, W., D', L., Mcgowan, A., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Lin Pedersen, T., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., and Yutani, H. (2019). Welcome to the Tidyverse. *Journal of Open Source Software* 4.43, 1686. DOI: 10.21105/JOSS.01686.

Wickham, H. and Seidel, D. (2022). scales: Scale Functions for Visualization. R package version 1.2.1. URL: https://CRAN.R-project.org/package=scales.

Wickham, H., Vaughan, D., and Girlich, M. (2023). tidyr: Tidy Messy Data. R package version 1.3.0. URL: https://CRAN.R-project.org/package=tidyr.

Wilke, C. O. (2020). cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'. R package version 1.1.1. URL: https://CRAN.R-project.org/package=cowplot.

Wilmes, P., Heintz-Buschart, A., and Bond, P. L. (2015). A decade of metaproteomics: Where we stand and what the future holds. *PROTEOMICS* 15.20, pp. 3409–3417. DOI: 10.1002/PMIC.201500183.

Xie, H., Wasserman, A., Levine, Z., Novik, A., Grebinskiy, V., Shoshan, A., and Mintz, L. (2002). Large-Scale Protein Annotation through Gene Ontology. *Genome Research* 12.5, pp. 785–794. DOI: 10.1101/GR.86902.

Yang, H. C., Li, W., Sun, J., and Gross, M. L. (2023). Advances in Mass Spectrometry on Membrane Proteins. *Membranes* 13.5. DOI: 10.3390/MEMBRANES13050457.

Ye, Y. (1987). Interior Algorithms for Linear, Quadratic, and Linearly Constrained Non-Linear Programming. PhD thesis. Stanford University.

Zhang, B., Chambers, M. C., and Tabb, D. L. (2007). Proteomic parsimony through bipartite graph analysis improves accuracy and transparency. *Journal of Proteome Research* 6.9, pp. 3549–3557. DOI: 10.1021/PR070230D.

Zhang, B., Pirmoradian, M., Zubarev, R., and Kall, L. (2017). Covariation of Peptide Abundances Accurately Reflects Protein Concentration Differences. *Molecular and Cellular Proteomics* 16.5, pp. 936–948. DOI: 10.1074/MCP.O117.067728.

Zhang, Y., Fonslow, B. R., Shan, B., Baek, M. C., and Yates, J. R. (2013). Protein analysis by shotgun/bottom-up proteomics. *Chemical Reviews* 113.4, pp. 2343–2394. DOI: 10.1021/CR3003533.

Zhang, Y., Wen, Z., Washburn, M. P., and Florens, L. (2010). Refinements to label free proteome quantitation: How to deal with peptides shared by multiple proteins. *Analytical Chemistry* 82.6, pp. 2272–2281. DOI: 10.1021/AC9023999.

Zhang, Y., Wen, Z., Washburn, M. P., and Florens, L. (2015). Improving label-free quantitative proteomics strategies by distributing shared peptides and stabilizing variance. *Analytical Chemistry* 87.9, pp. 4749–4756. DOI: 10.1021/AC504740P.

Zhu, Y., Hultin-Rosenberg, L., Forshed, J., Branca, R. M., Orre, L. M., and Lehtiö, J. (2014). SpliceVista, a Tool for Splice Variant Identification and Visualization in Shotgun Proteomics Data. *Molecular and Cellular Proteomics* 13.6, pp. 1552–1562. DOI: 10.1074/MCP.M113.031203.

A Additional Figures

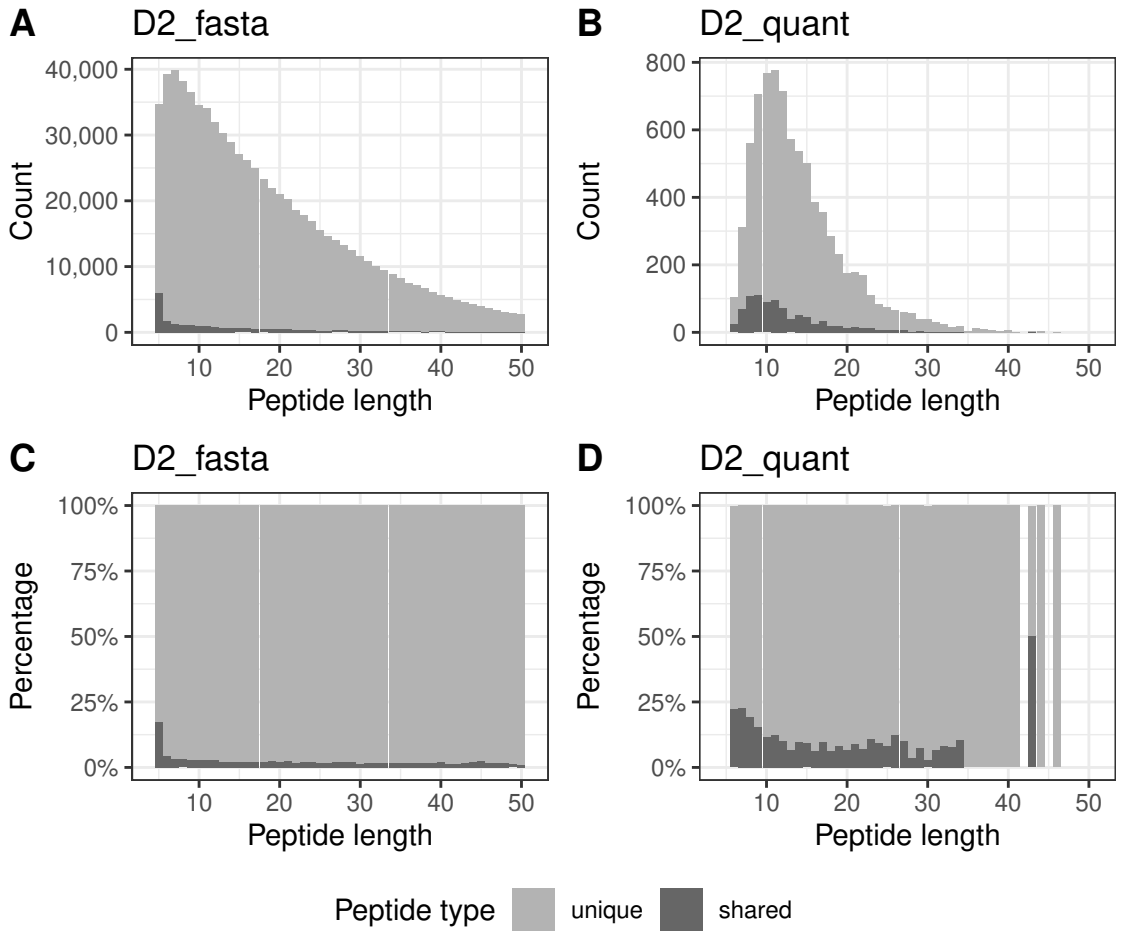


Figure A.1: Count and percentages of shared and unique peptide sequences depending on the peptide length. The peptide length is given in amino acids. As an example, here the values for data set D2 are shown. Uniqueness is here defined as belonging to only one protein node, which may consist of multiple protein accessions.

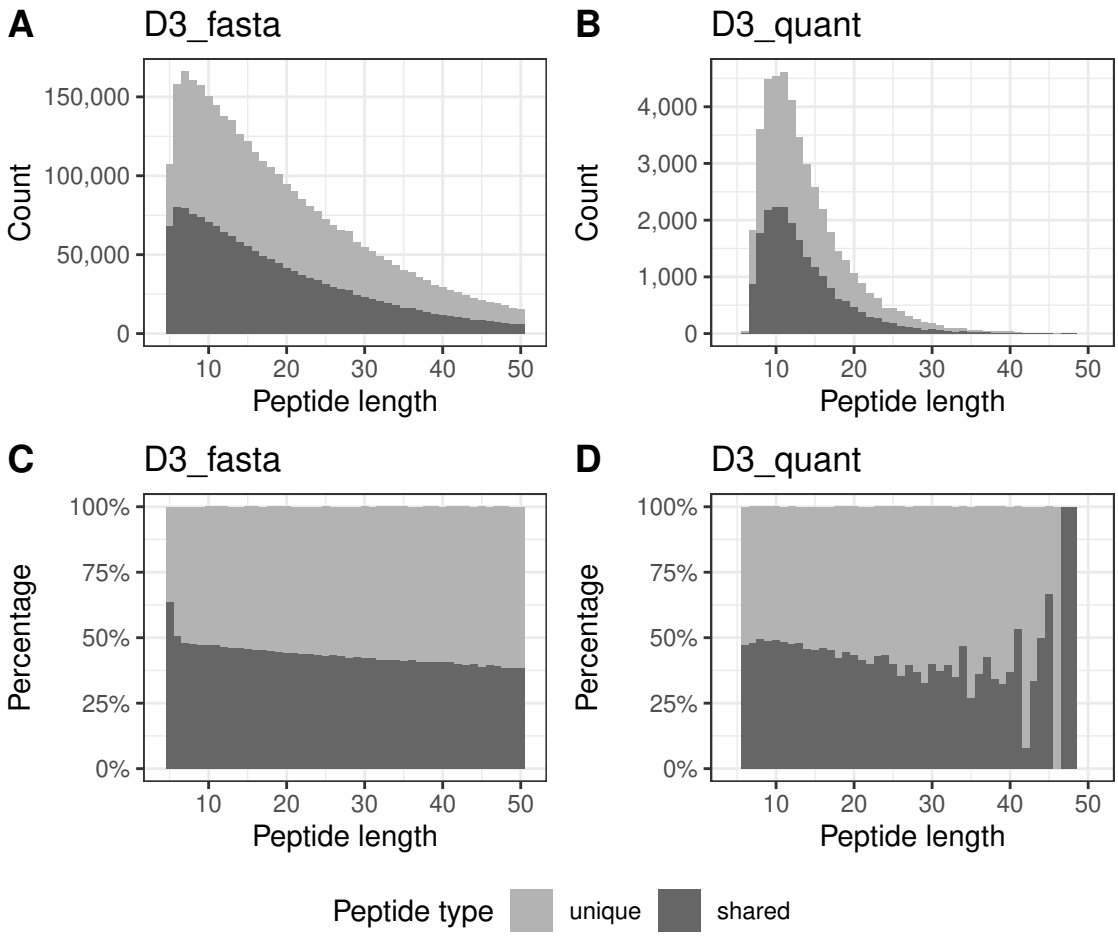


Figure A.2: Count and percentages of shared and unique peptide sequences depending on the peptide length. The peptide length is given in amino acids. As an example, here the values for data set D3 are shown. Uniqueness is here defined as belonging to only one protein node, which may consist of multiple protein accessions.

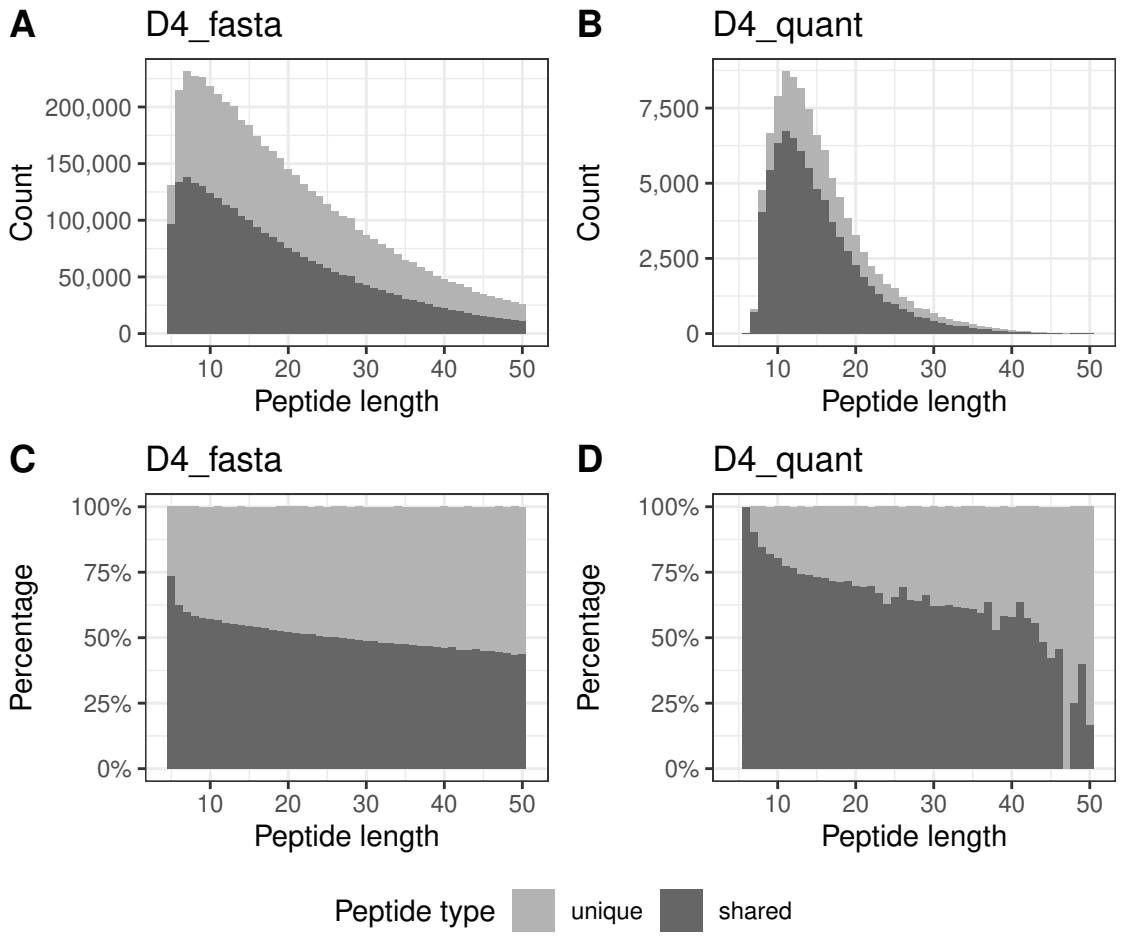


Figure A.3: Count and percentages of shared and unique peptide sequences depending on the peptide length. The peptide length is given in amino acids. As an example, here the values for data set D4 are shown. Uniqueness is here defined as belonging to only one protein node, which may consist of multiple protein accessions.

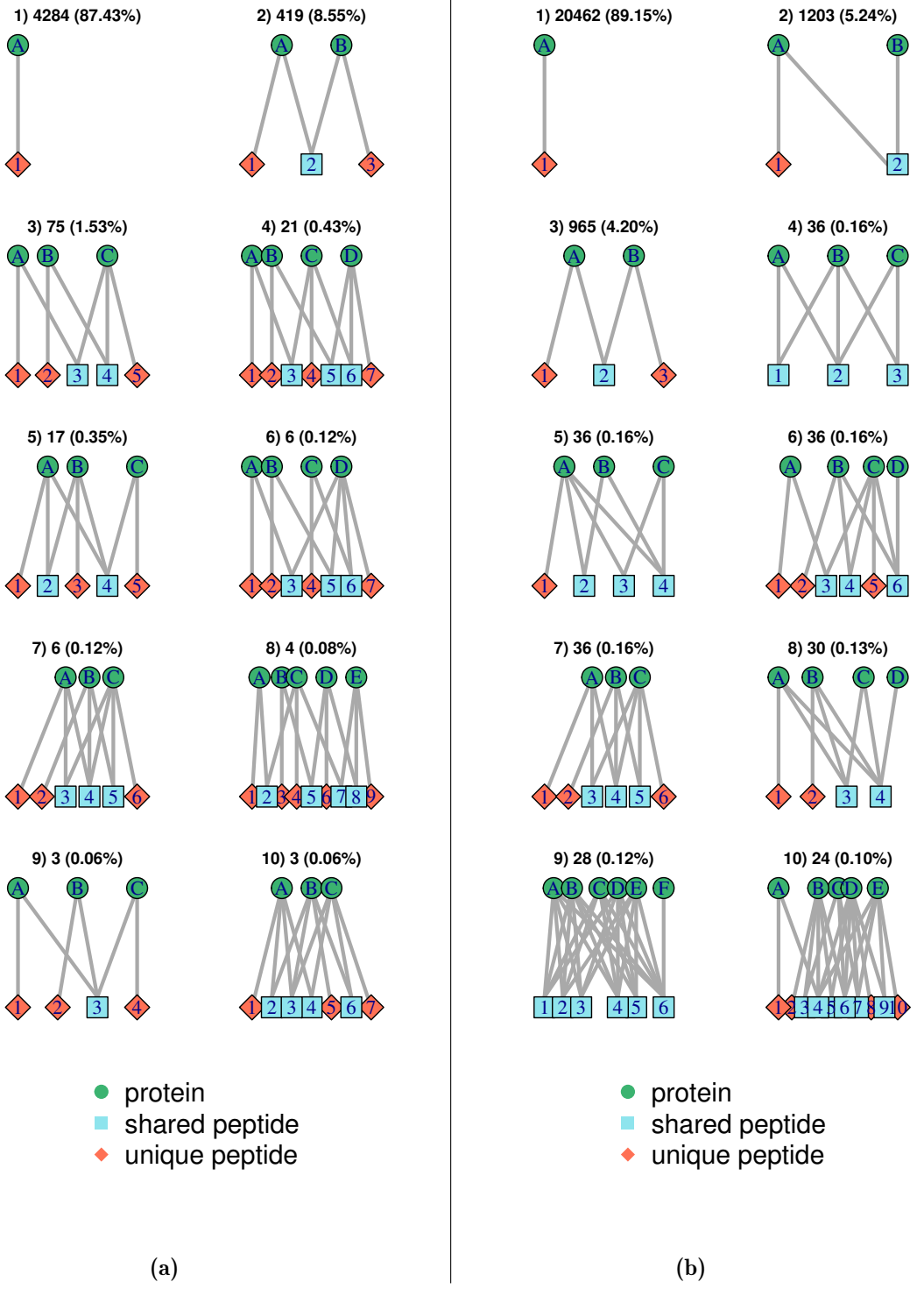


Figure A.4: Representative bipartite graphs of the ten largest isomorphism classes found in data set D2. (a) D2_fasta, (b) D2_quant, with rank, number of occurrences and percentage of all graphs.

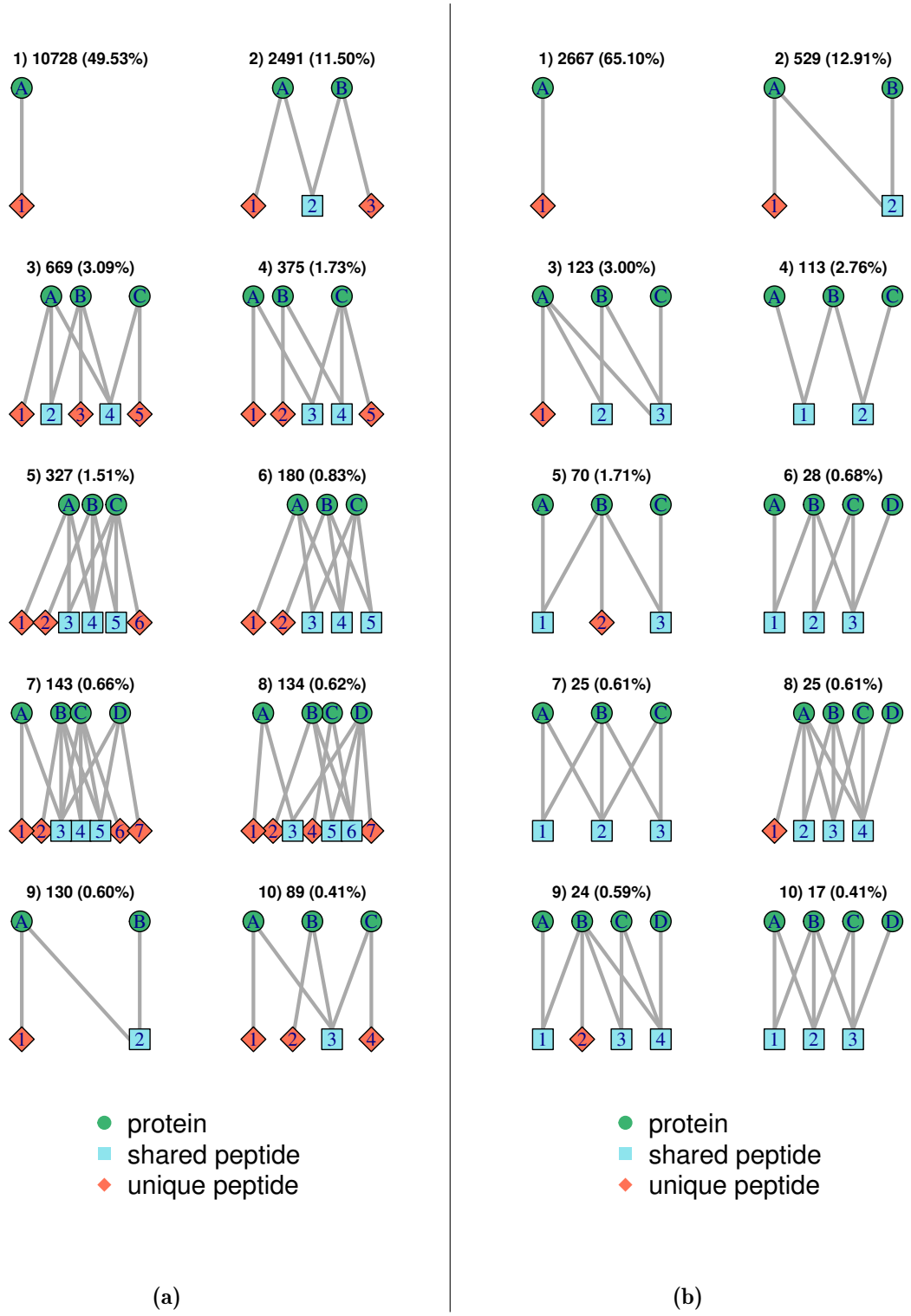


Figure A.5: Representative bipartite graphs of the ten largest isomorphism classes found in data set D3. (a) D3_fasta, (b) D3_quant, with rank, number of occurrences and percentage of all graphs.

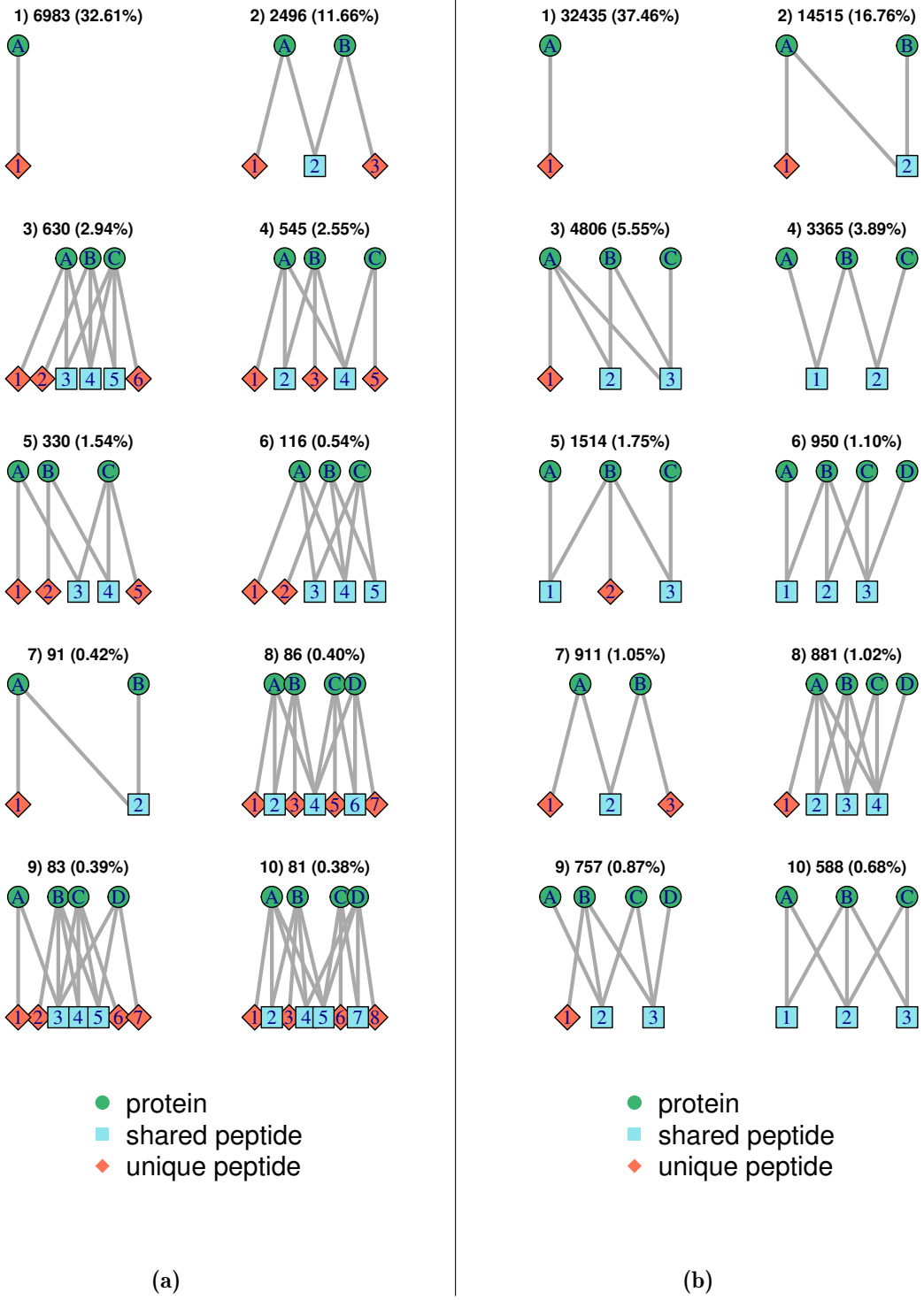


Figure A.6: Representative bipartite graphs of the ten largest isomorphism classes found in data set D4. (a) D4_fasta, (b) D4_quant, with rank, number of occurrences and percentage of all graphs.

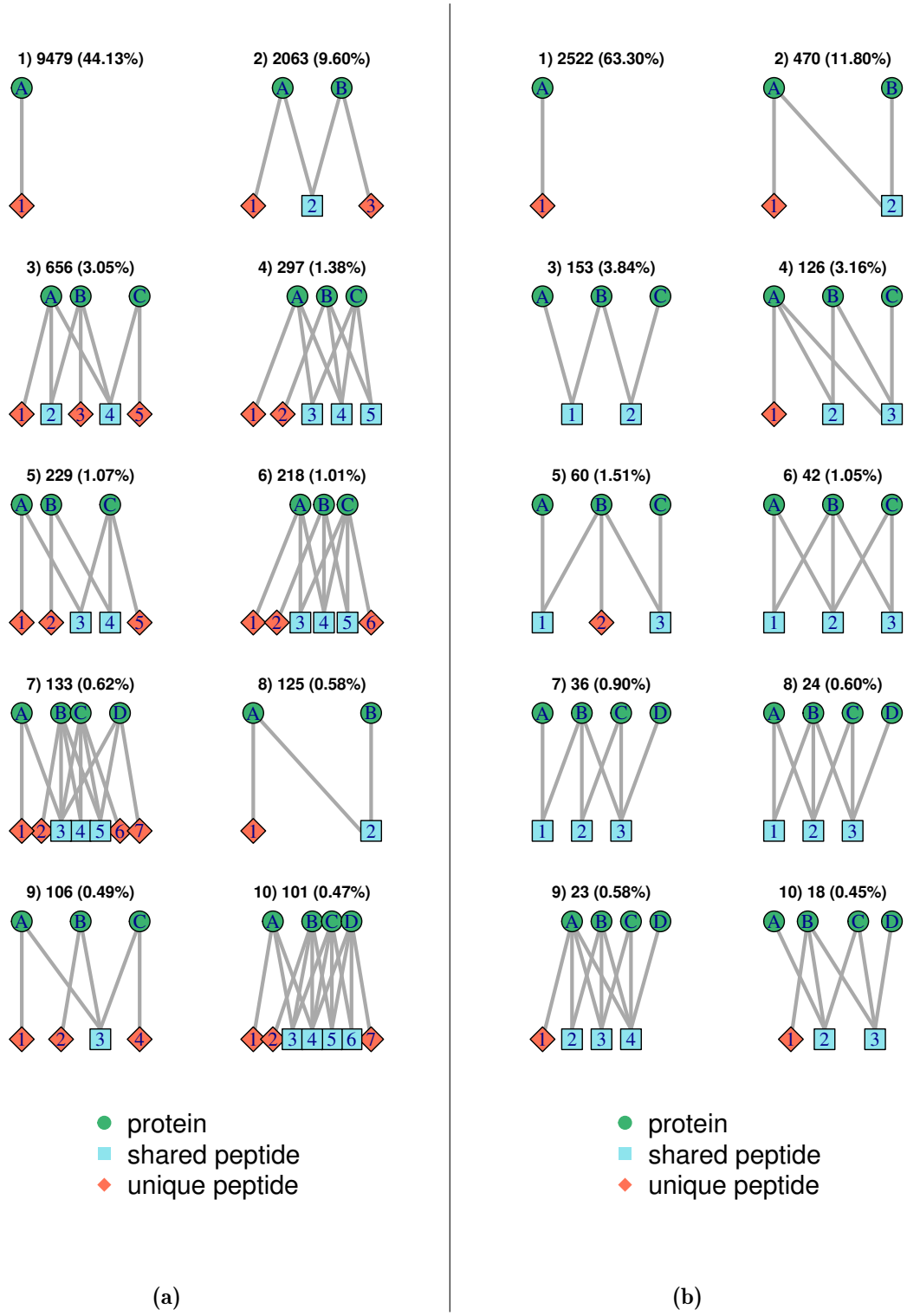


Figure A.7: Representative bipartite graphs of the ten largest isomorphism classes found in data set D3iso. (a) D3iso_fasta, (b) D3iso_quant, with rank, number of occurrences and percentage of all graphs.

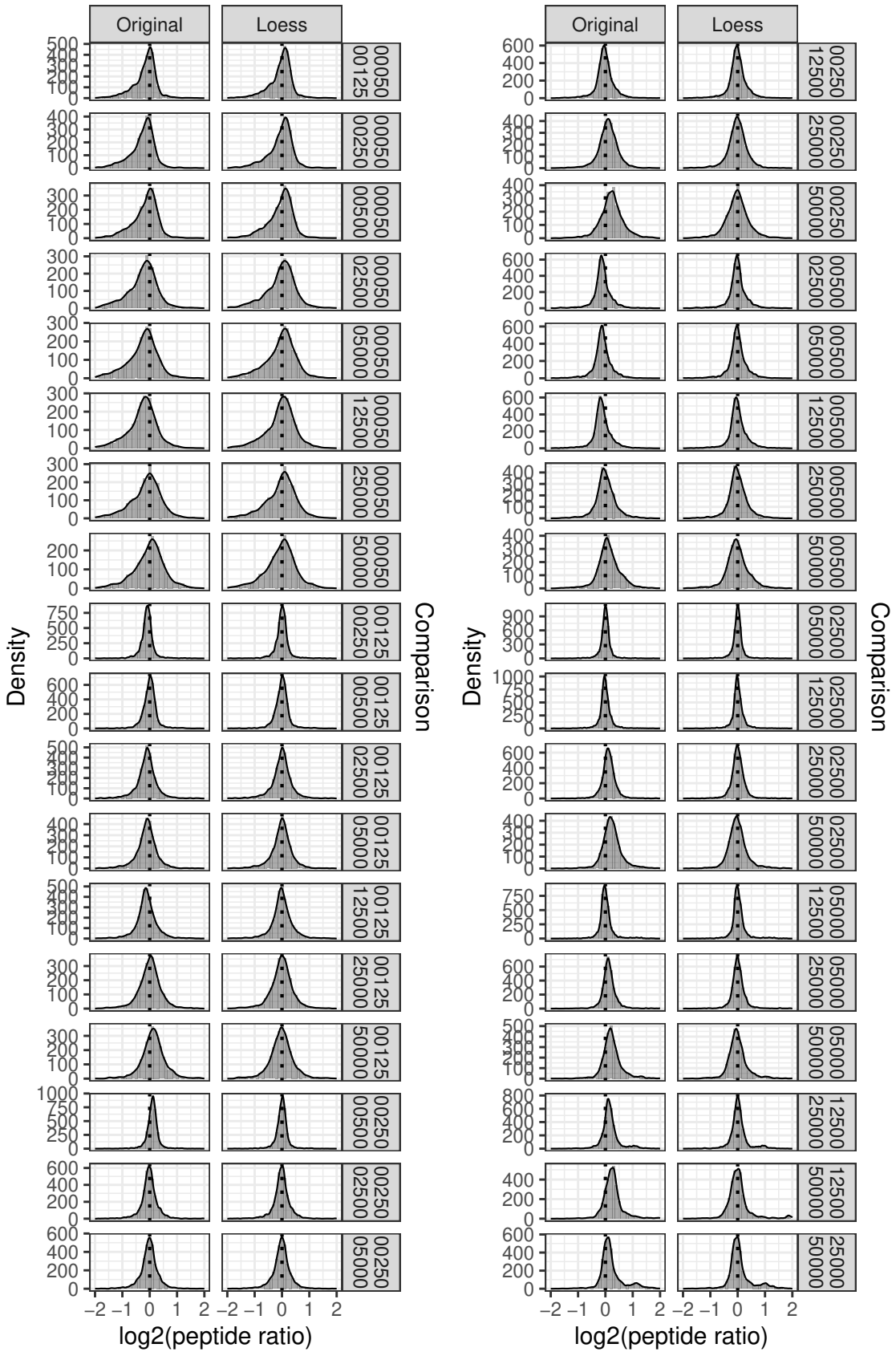


Figure A.8: Histograms and density estimates of log2-transformed peptide ratios in data set D2 for the 36 different pairwise condition comparisons. Original, non-normalized values are shown on the left and loess-normalized values on the right. The vertical dotted line describes the expected ratio of 1 (0 on log2-scale). For better visibility, the x-axes were cut at -2 an 2 on log2-scale.

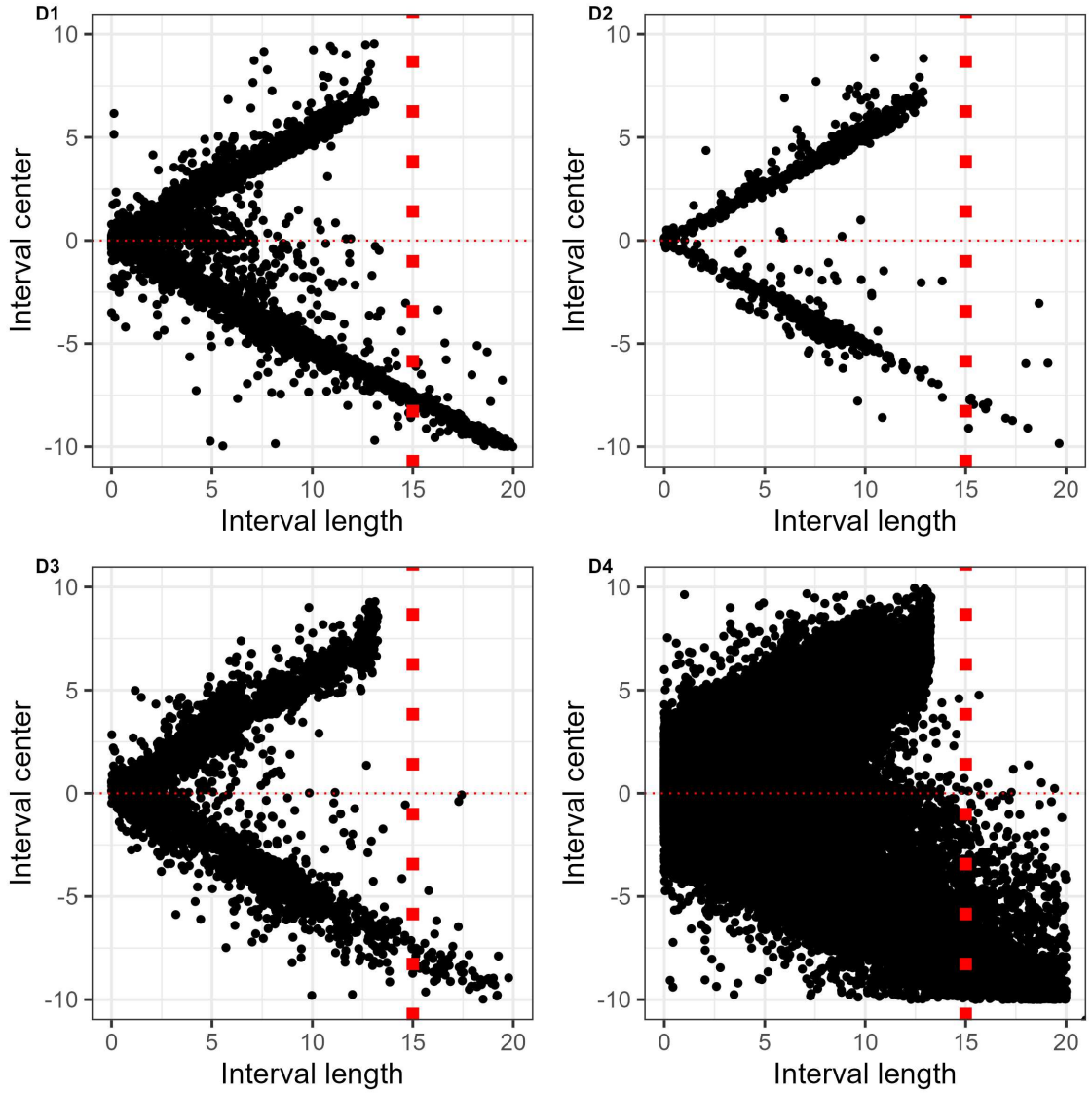


Figure A.9: Scatter Plot comparing the interval center and interval length for the range solutions. The proposed threshold for removing too large intervals at 15 is indicated by a red dashed vertical line.

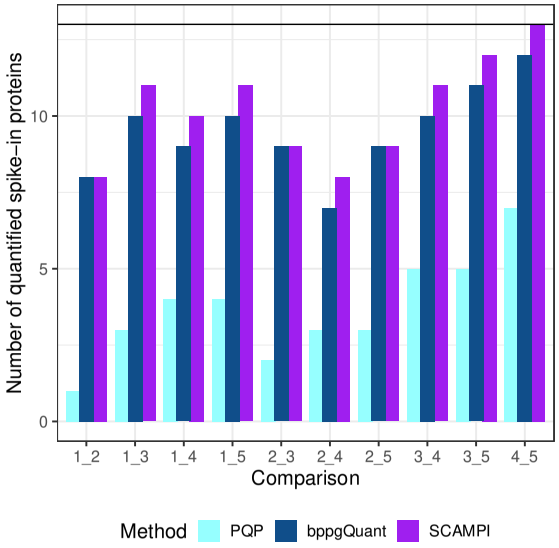


Figure A.10: Number of quantified spike-in proteins over the different pairwise condition comparisons and quantification methods in data set D1. The black horizontal line refers to the total number of spike-in proteins in this data set, namely 13.

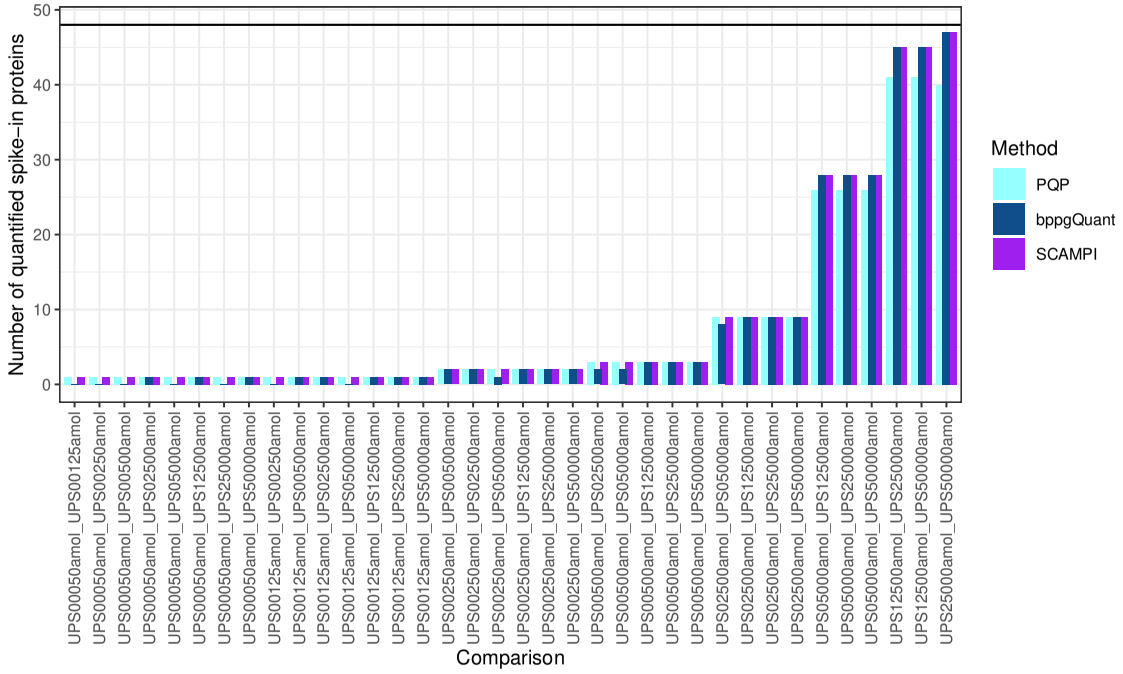


Figure A.11: Number of quantified spike-in proteins over the different pairwise condition comparisons and quantification methods in data set D2. The black horizontal line refers to the total number of spike-in proteins in this data set, namely 48.

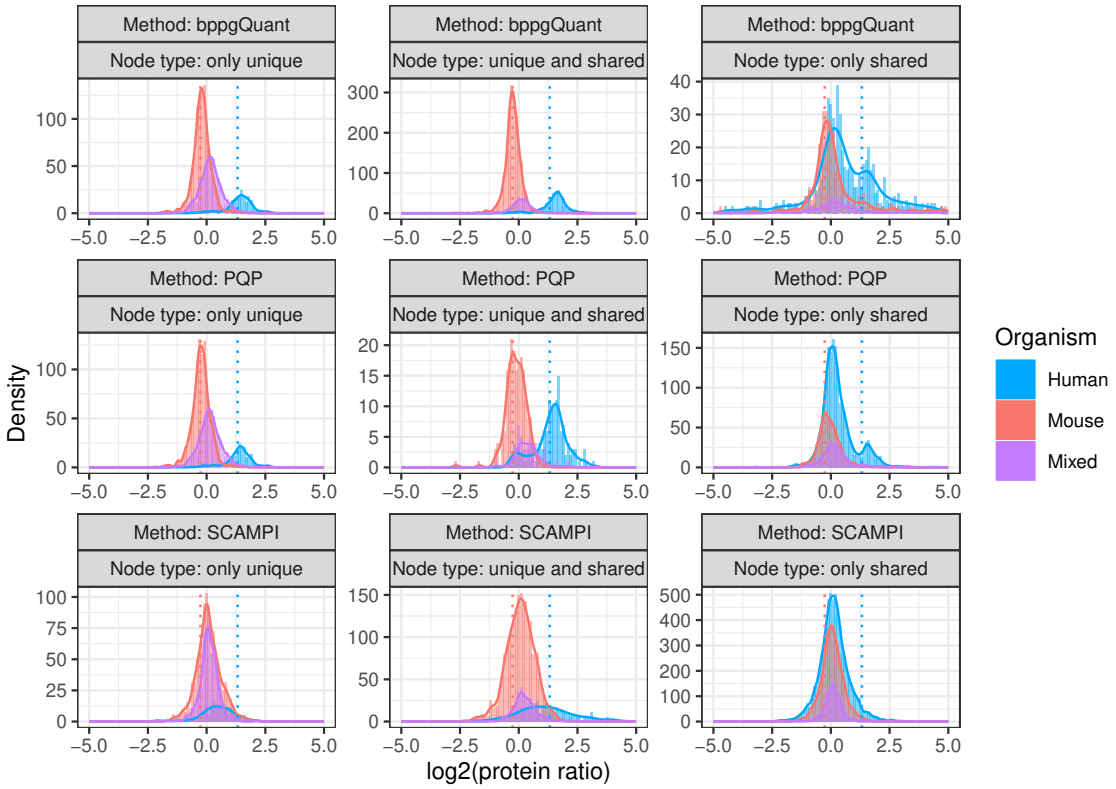


Figure A.12: Histograms and density plots for the estimated protein ratios in data set D4 comparing the 10% human / 90% mouse sample with the 25% human / 75% mouse samples (comparison 010_025). Contaminant proteins were removed. Only single solutions (no range solutions) regardless of the node type were used. The x-axes are cut at -5 and 5 for better visibility. Please consider the different ranges of the y-axes..

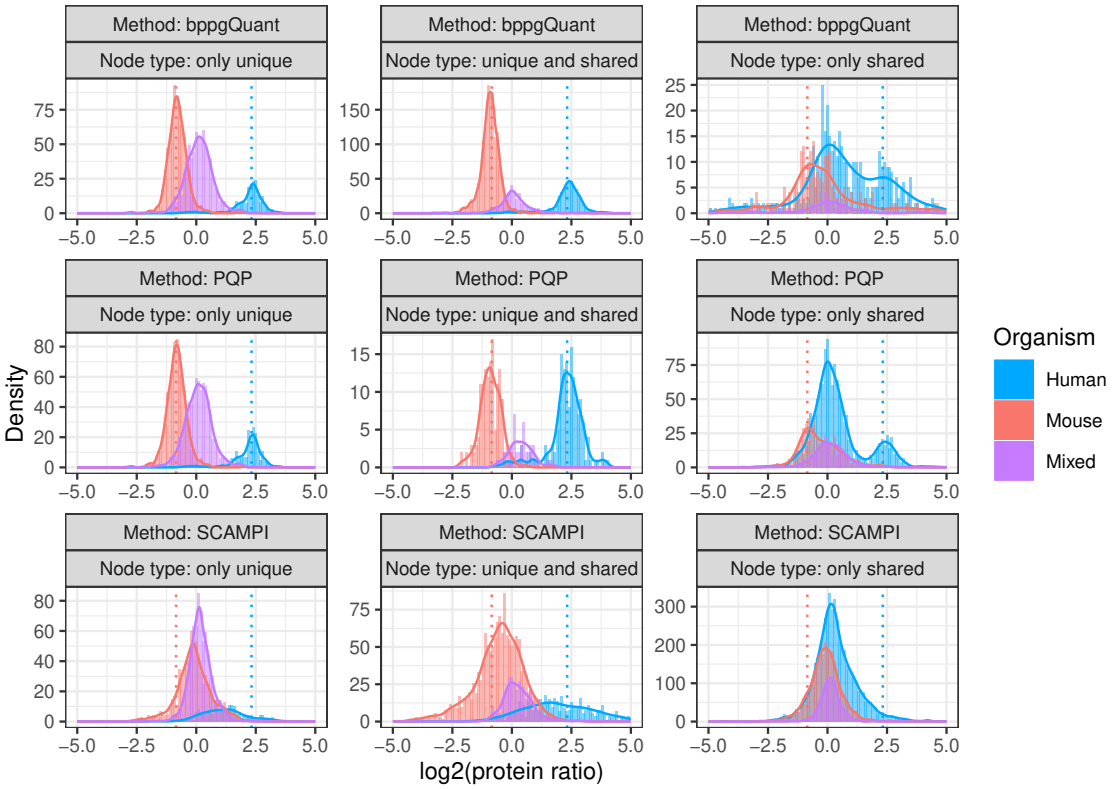


Figure A.13: Histograms and density plots for the estimated protein ratios in data set D4 comparing the 10% human / 90% mouse sample with the 50% human / 50% mouse samples (comparison 010_050). Contaminant proteins were removed. Only single solutions (no range solutions) regardless of the node type were used. The x-axes are cut at -5 and 5 for better visibility. Please consider the different ranges of the y-axes.

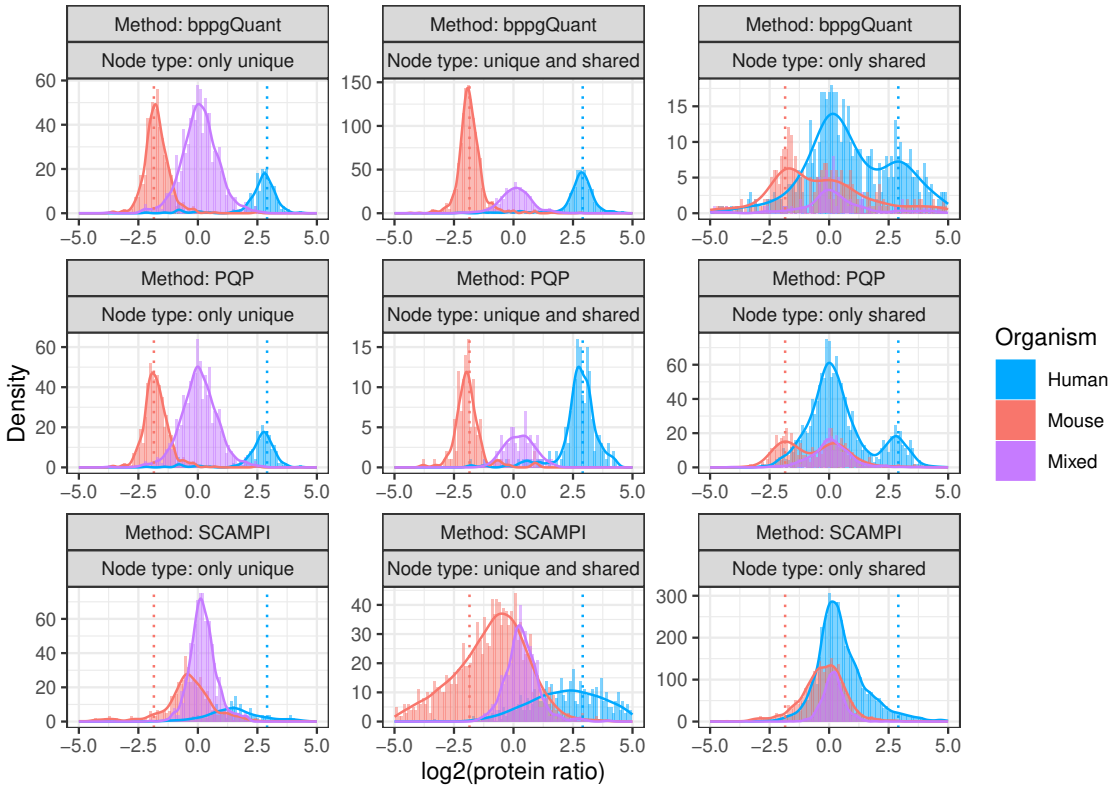


Figure A.14: Histograms and density plots for the estimated protein ratios in data set D4 comparing the 10% human / 90% mouse sample with the 75% human / 25% mouse samples (comparison 010_075). Contaminant proteins were removed. Only single solutions (no range solutions) regardless of the node type were used. The x-axes are cut at -5 and 5 for better visibility. Please consider the different ranges of the y-axes.

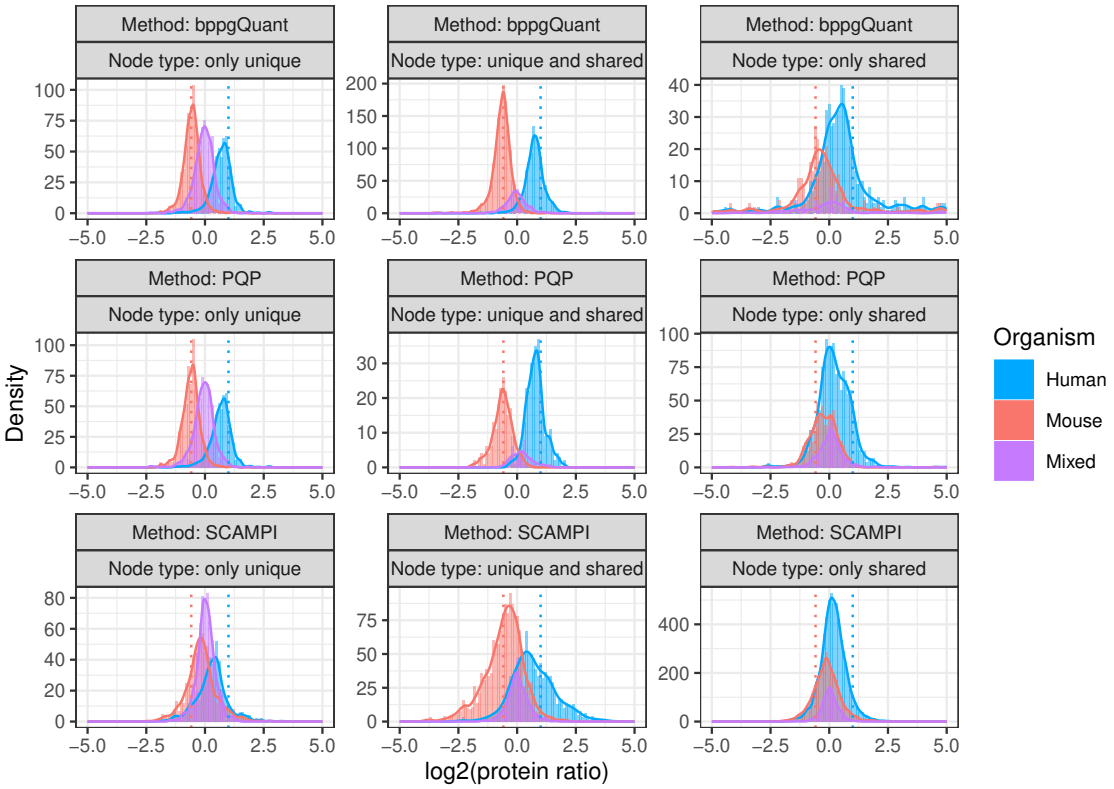


Figure A.15: Histograms and density plots for the estimated protein ratios in data set D4 comparing the 25% human / 75% mouse sample with the 50% human / 50% mouse samples (comparison 025_050). Contaminant proteins were removed. Only single solutions (no range solutions) regardless of the node type were used. The x-axes are cut at -5 and 5 for better visibility. Please consider the different ranges of the y-axes.

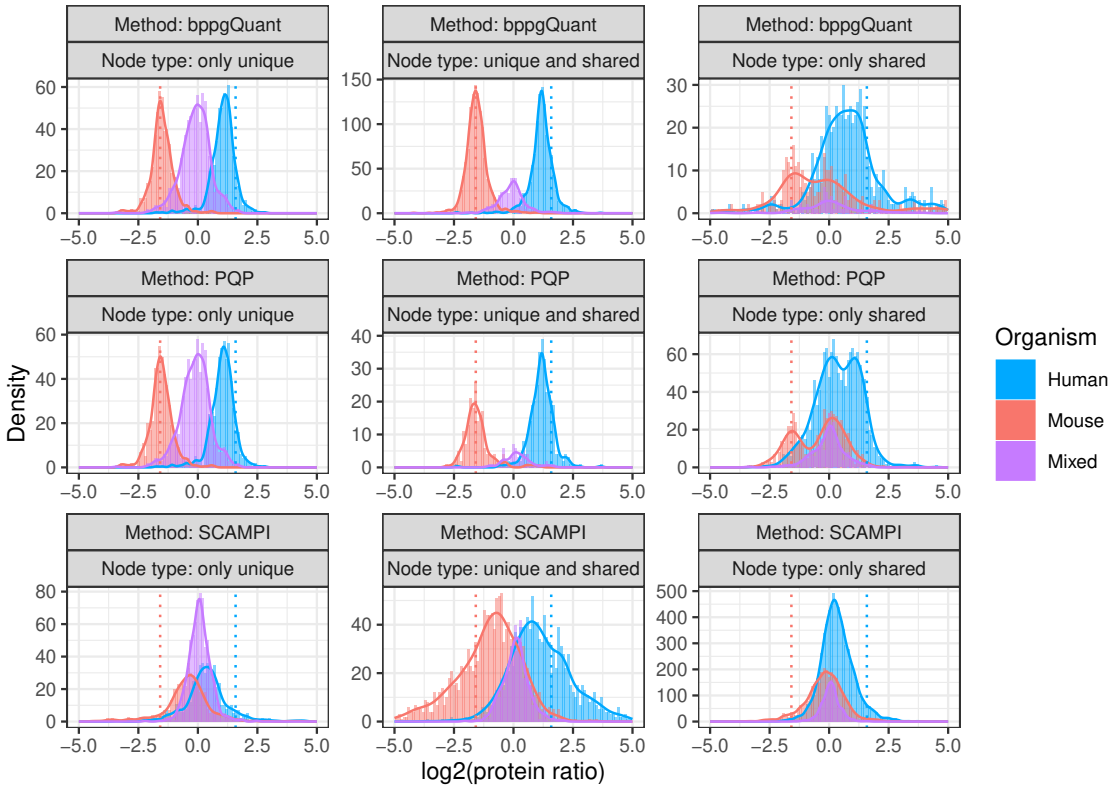


Figure A.16: Histograms and density plots for the estimated protein ratios in data set D4 comparing the 25% human / 75% mouse sample with the 75% human / 25% mouse samples (comparison 010_090). Contaminant proteins were removed. Only single solutions (no range solutions) regardless of the node type were used. The x-axes are cut at -5 and 5 for better visibility. Please consider the different ranges of the y-axes.

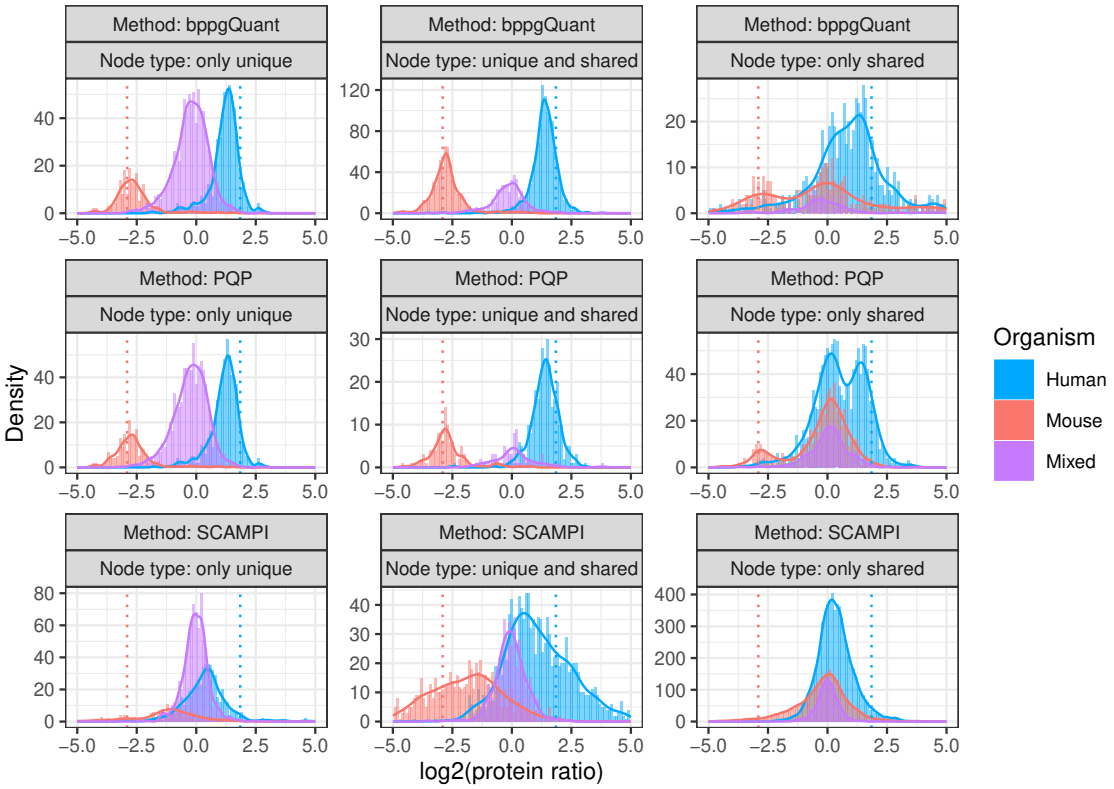


Figure A.17: Histograms and density plots for the estimated protein ratios in data set D4 comparing the 25% human / 75% mouse sample with the 10% human / 90% mouse samples (comparison 025_090). Contaminant proteins were removed. Only single solutions (no range solutions) regardless of the node type were used. The x-axes are cut at -5 and 5 for better visibility. Please consider the different ranges of the y-axes.

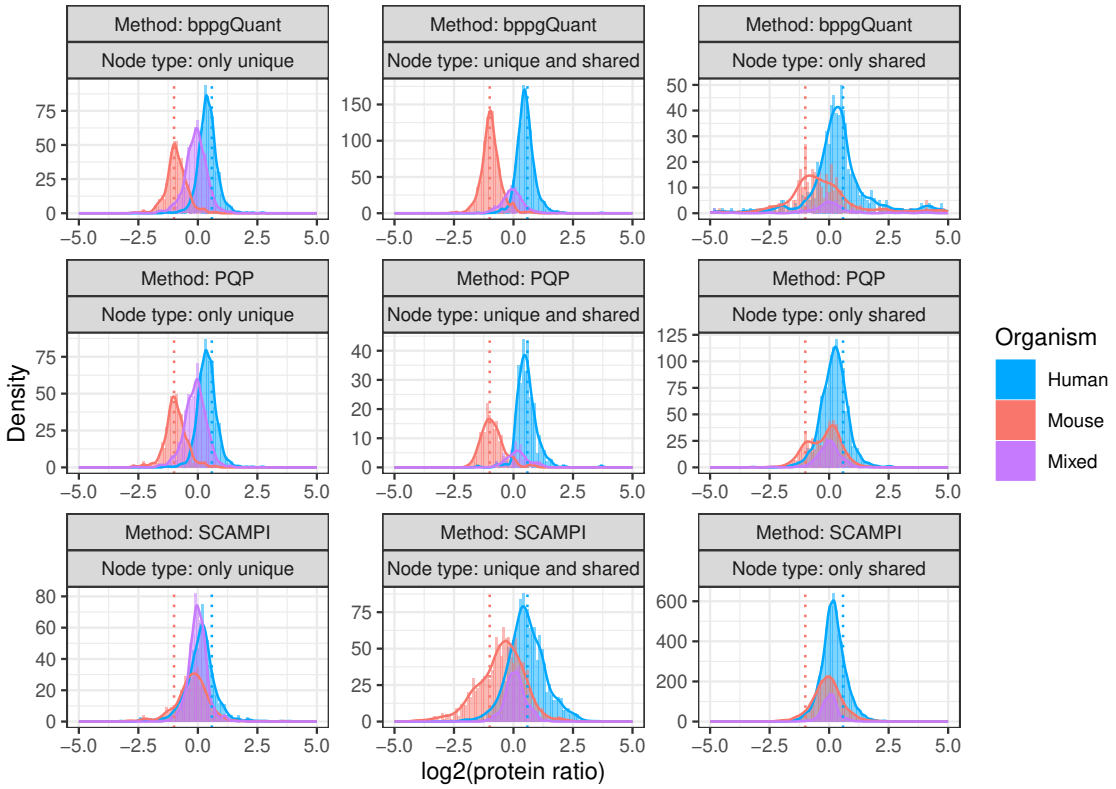


Figure A.18: Histograms and density plots for the estimated protein ratios in data set D4 comparing the 50% human / 50% mouse sample with the 75% human / 25% mouse samples (comparison 050_075). Contaminant proteins were removed. Only single solutions (no range solutions) regardless of the node type were used. The x-axes are cut at -5 and 5 for better visibility. Please consider the different ranges of the y-axes.

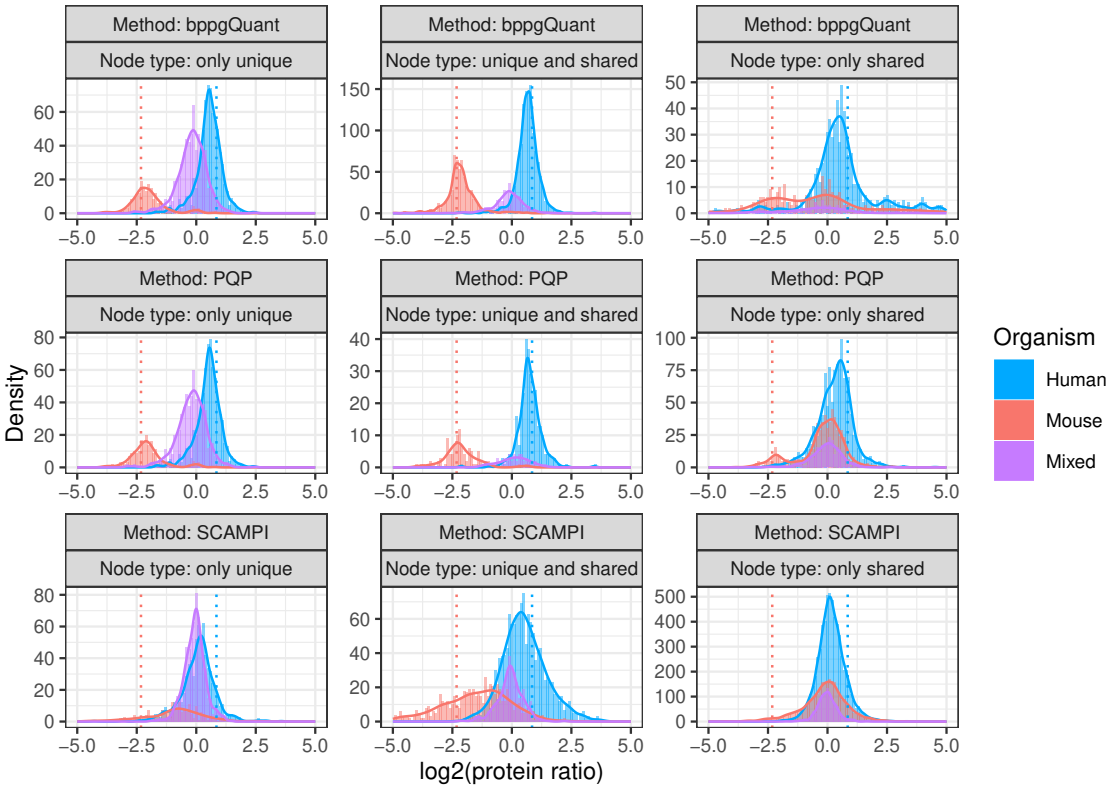


Figure A.19: Histograms and density plots for the estimated protein ratios in data set D4 comparing the 50% human / 50% mouse sample with the 90% human / 10% mouse samples (comparison 050_090). Contaminant proteins were removed. Only single solutions (no range solutions) regardless of the node type were used. The x-axes are cut at -5 and 5 for better visibility. Please consider the different ranges of the y-axes.

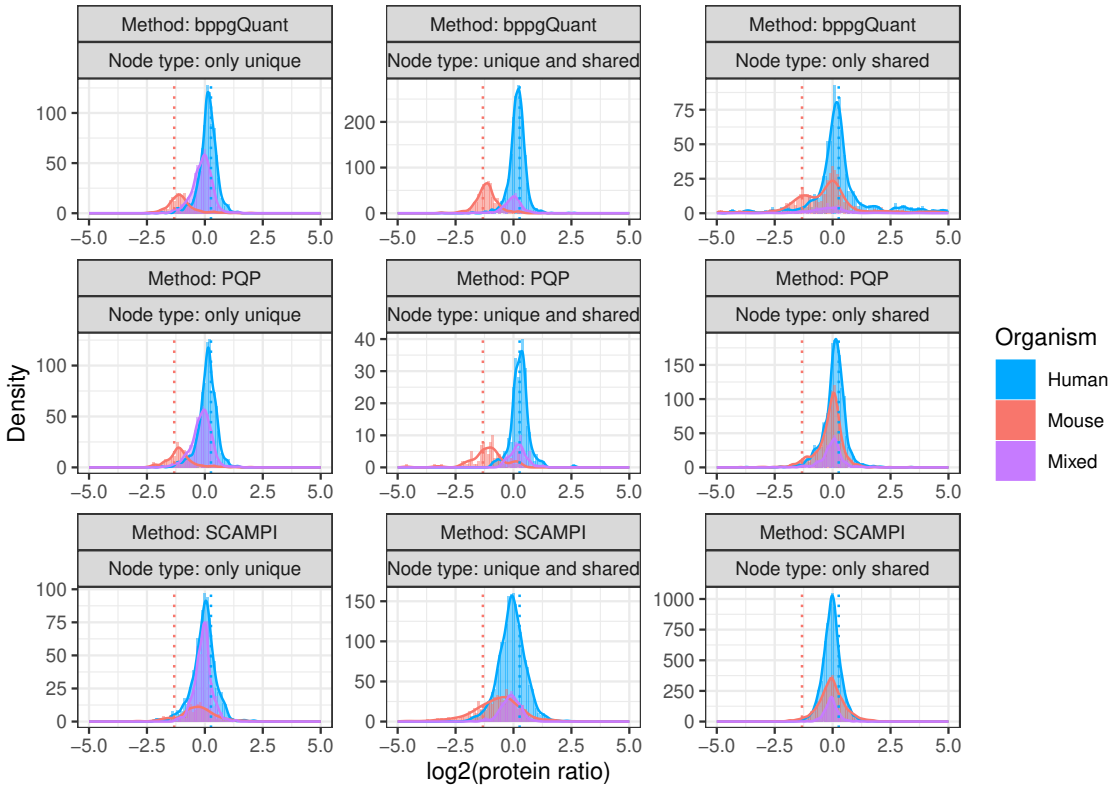
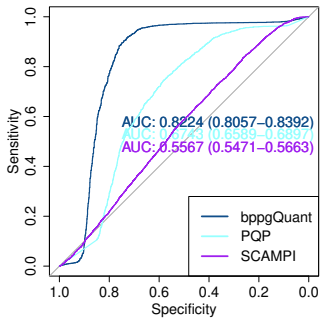
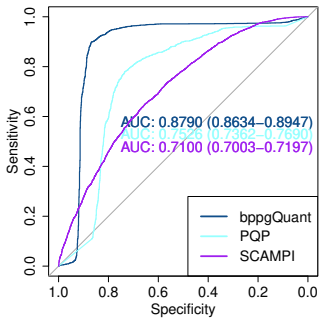


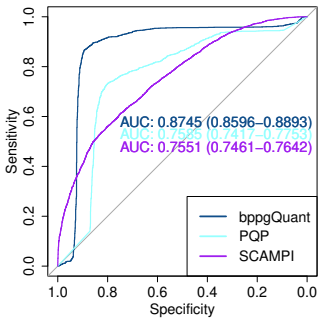
Figure A.20: Histograms and density plots for the estimated protein ratios in data set D4 comparing the 75% human / 25% mouse sample with the 90% human / 10% mouse samples (comparison 075_090). Contaminant proteins were removed. Only single solutions (no range solutions) regardless of the node type were used. The x-axes are cut at -5 and 5 for better visibility. Please consider the different ranges of the y-axes.



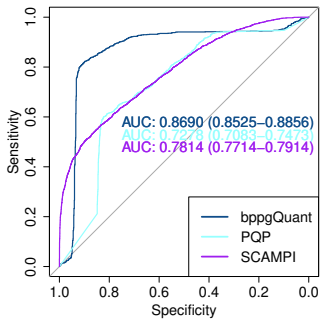
(a) HELA010_HELA025



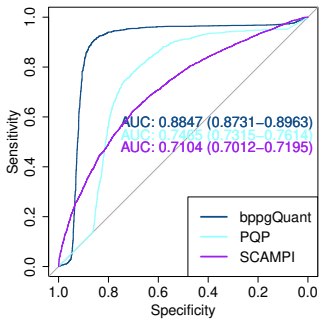
(b) HELA010_HELA050



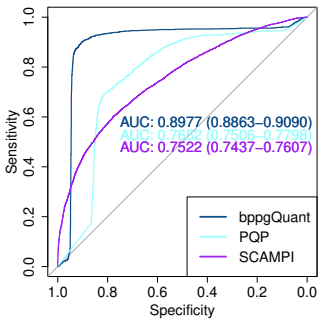
(c) HELA010_HELA075



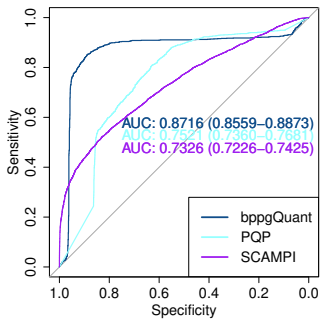
(d) HELA010_HELA090



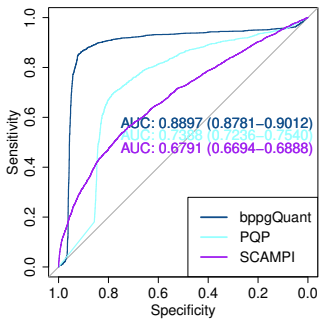
(e) HELA025_HELA050



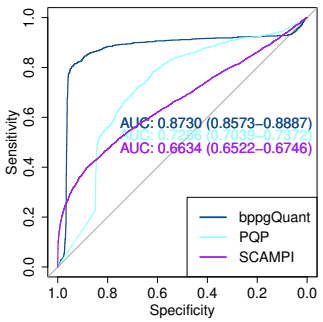
(f) HELA025_HELA075



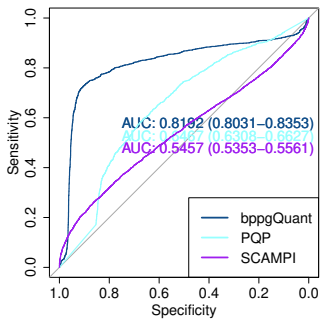
(g) HELA025_HELA090



(h) HELA050_HELA075



(i) HELA050_HELA090



(j) HELA075_HELA090

Figure A.21: ROC curves for distinguishing human from mouse proteins in data set D4.

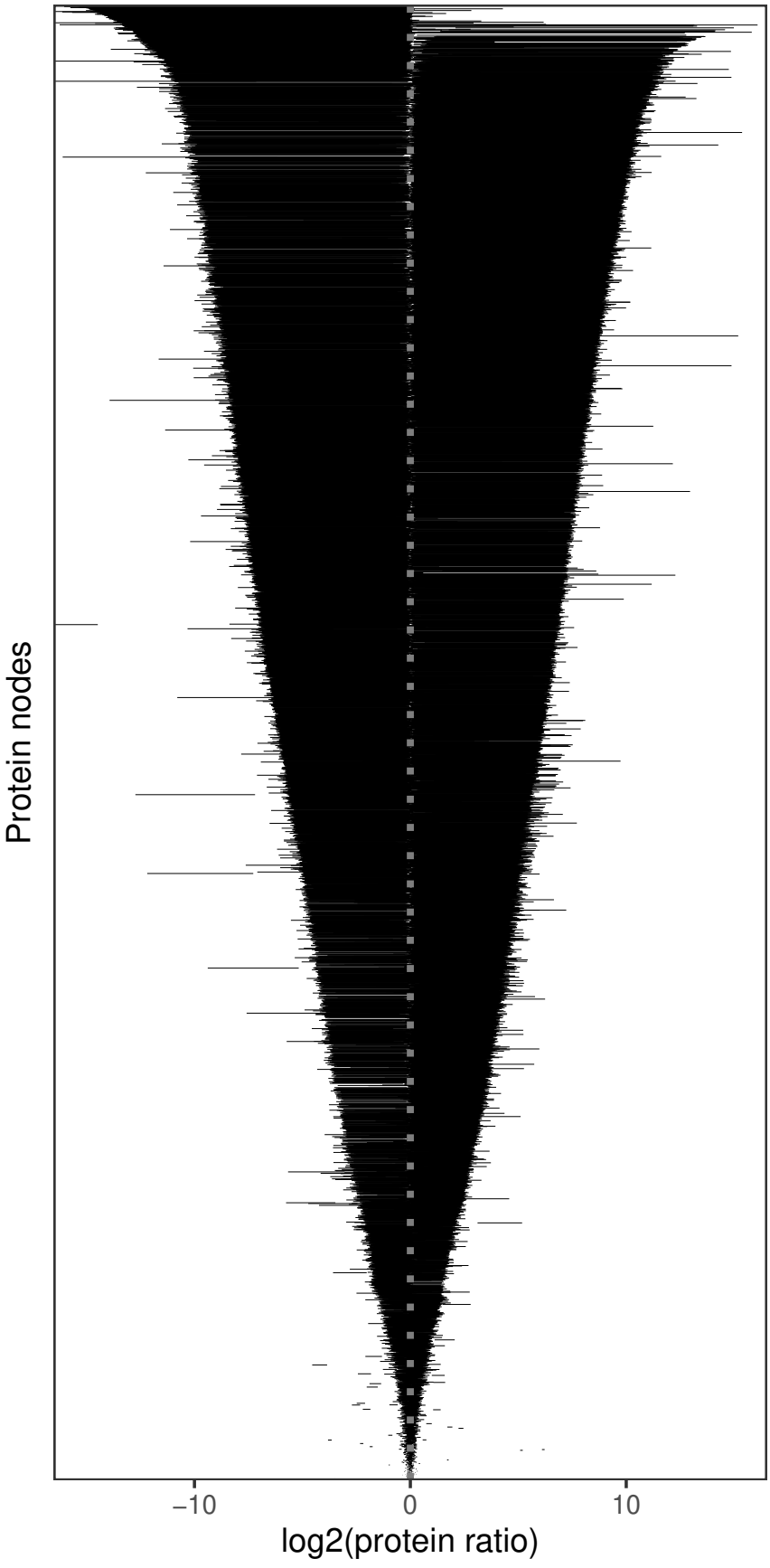


Figure A.22: Graphical representation of range solutions for data set D1 (over all pairwise condition comparisons). Each horizontal line represents a range solution for one protein node, sorted by decreasing interval length from top to bottom.

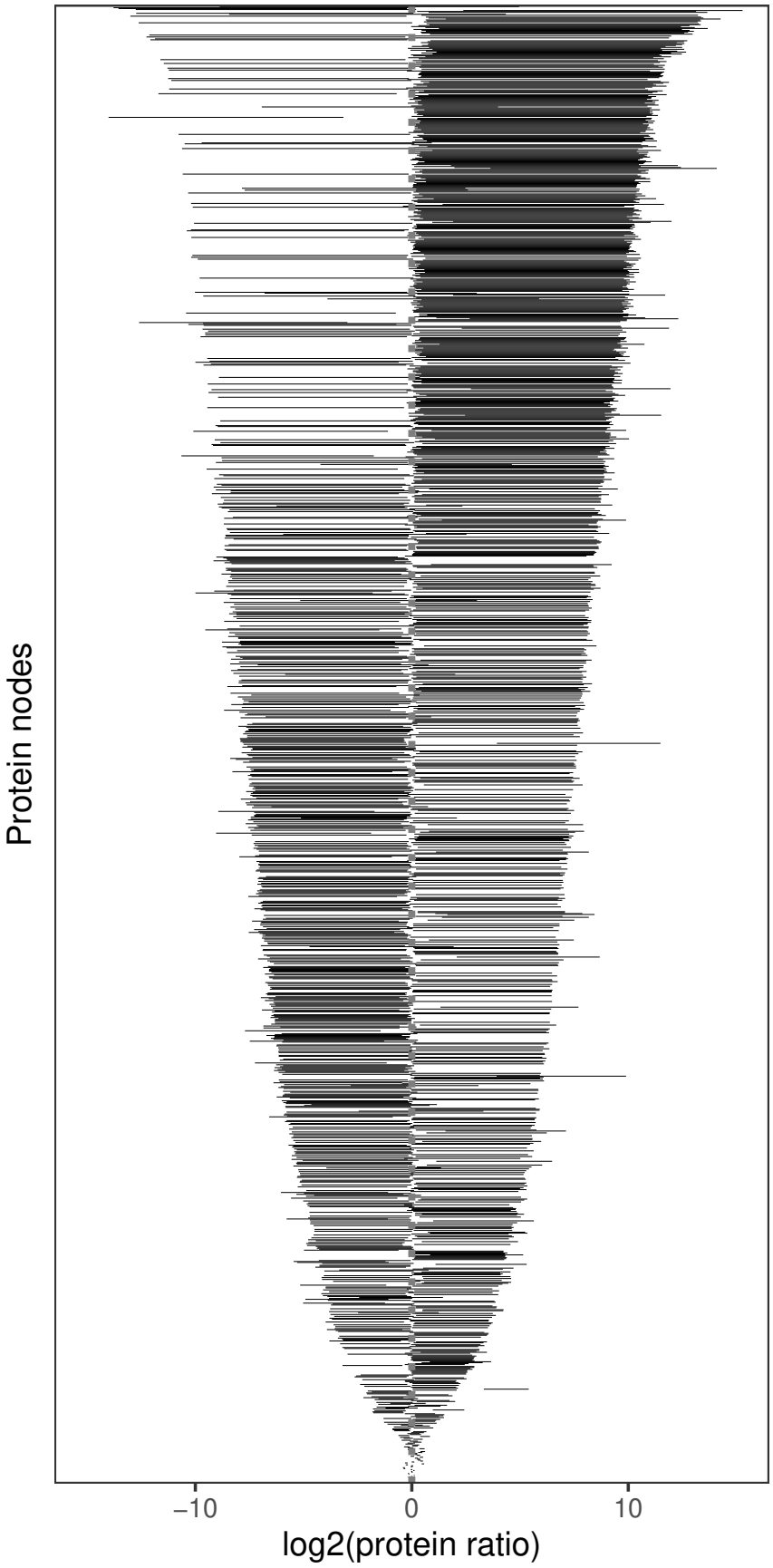


Figure A.23: Graphical representation of range solutions for data set D2 (over all pairwise condition comparisons). Each horizontal line represents a range solution for one protein node, sorted by decreasing interval length from top to bottom.

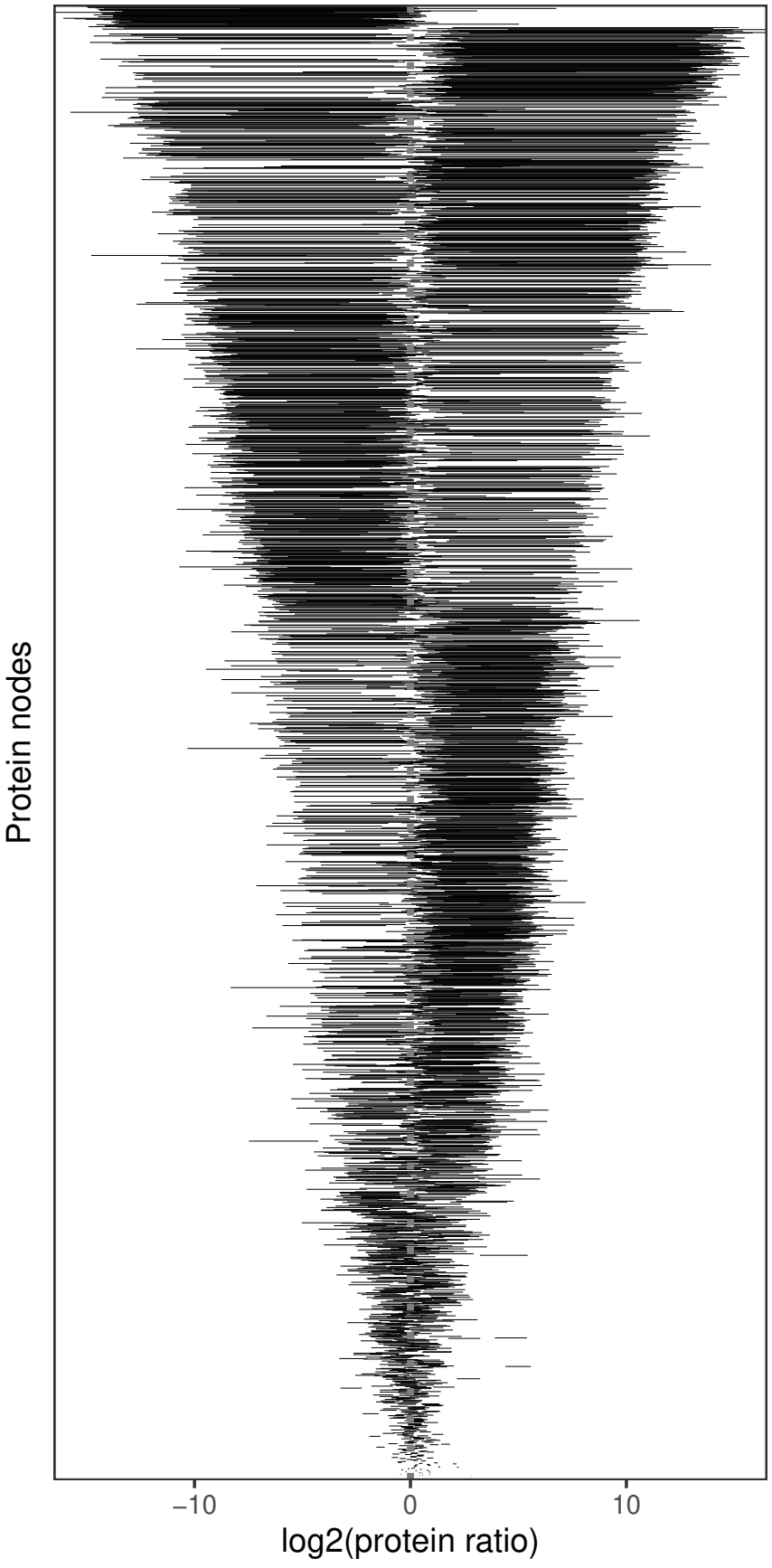


Figure A.24: Graphical representation of range solutions for data set D3. Each horizontal line represents a range solution for one protein node, sorted by decreasing interval length from top to bottom.

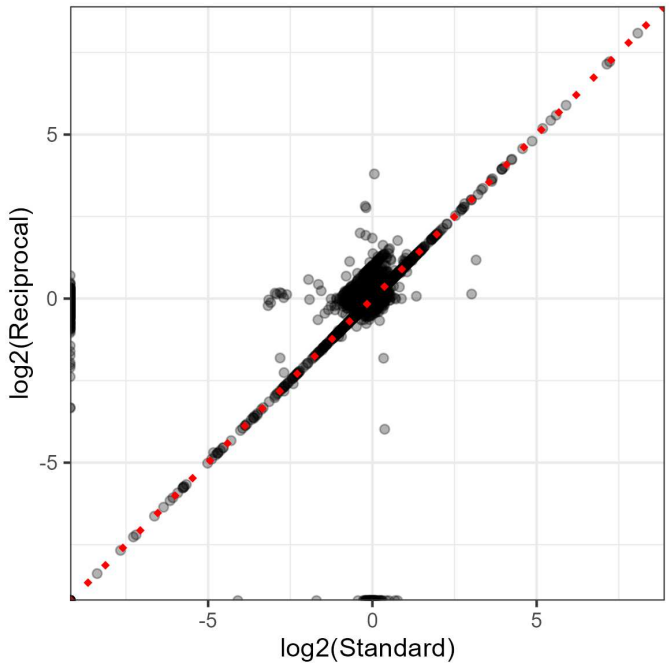


Figure A.25: Protein ratios estimated by PQP on data set D1 using the standard peptide ratios vs. using the reciprocal of the peptide ratios (and reporting 1/the obtained protein ratios). The points are expected to lie on the red dotted line, representing the angle bisector.

B Additional Tables

Table B.1: Overview over the four used test data sets. For the possible ratios, the reciprocal is given if the ratio is below 1. For the number of conditions and comparisons for data set D4, the numbers including and excluding the pure human and mouse samples are given.

	D1	D2	D3	D4
Background	C2C12 (mouse)	yeast	HeLa (human)	HeLa (human)
Spike-ins	13 non-mouse proteins	UPS1 (48 human proteins)	E. coli	NIH-3T3 (mouse)
Reference	Barkovits et al., 2020; Uszkoreit et al., 2022	Ramus et al., 2016a; Ramus et al., 2016b	Cox et al., 2014	Saltzman et al., 2018
PXD Number	PXD012986	PXD001819	PXD000279	PXD008560
Number of conditions (comparisons)	5 (10)	9 (36)	2 (1)	7 (21) / 5 (10)
Possible ratios	2 - 100	2 - 1000	3	1.1 - ∞ / 1.1 - 10

Table B.2: Influence of different minimal peptide lengths on the database bipartite graphs for D2_fasta. * In terms of number of protein nodes.

	min 5 AA	min 6 AA	min 7 AA	min 8 AA	min 9 AA
Protein accessions	6,347	6,346	6,344	6,344	6,344
Protein nodes	6,275	6,274	6,273	6,273	6,273
Peptide sequences	766,652	732,017	692,743	652,926	614,717
Peptide nodes	13,132	8,162	7,574	7,453	7,377
Unique peptide nodes	6,218	6,216	6,215	6,215	6,215
Shared peptide nodes	6,914	1,946	1,359	1,238	1,162
Edges	27,226	15,122	13,666	13,089	12,690
Graphs	1,585	4,900	5,481	5,571	5,614
Graphs with 1 protein node	1,456	4,284	5,055	5,202	5,274
Isomorphism classes	18	66	41	36	36
Largest graph*					
Protein nodes	4,504	116	69	69	69
Protein nodes percentage	71.78%	1.85%	1.10%	1.10%	1.10%
Peptide nodes	11,174	356	264	258	253
Second largest graph*					
Protein nodes	21	71	50	49	49
Peptide nodes	37	269	221	214	202

Table B.3: Influence of different minimal peptide lengths on the database bipartite graphs for D3_fasta. * In terms of number of protein nodes.

	min 5 AA	min 6 AA	min 7 AA	min 8 AA	min 9 AA
Protein accessions	86,411	86,388	86,362	86,316	86,250
Protein nodes	85,690	85,647	85,603	85,519	85,409
Peptide sequences	3,415,222	3,307,762	3,149,627	2,983,664	2,823,406
Peptide nodes	202,302	166,578	157,154	154,010	151,635
Unique peptide nodes	74,683	74,499	74,255	73,927	73,476
Shared peptide nodes	127,619	92,079	82,899	80,083	78,159
Edges	797,291	547,611	491,889	473,014	459,360
Graphs	4,372	13,869	20,180	21,660	22,303
Graphs with 1 protein node	3,597	8,103	10,102	10,728	11,072
Isomorphism classes	229	2,308	4,347	4,633	4,720
Largest graph*					
Protein nodes	79,328	45,242	7,027	2,661	2,283
Protein nodes percentage	92.58%	52.82%	8.21%	3.11%	2.67%
Peptide nodes	194,113	99,337	16,050	6,352	5,226
Second largest graph*					
Protein nodes	12	57	451	470	261
Peptide nodes	36	129	896	1,038	519

Table B.4: Influence of different minimal peptide lengths on the database bipartite graphs for D4_fasta. * In terms of number of protein nodes.

	min 5 AA	min 6 AA	min 7 AA	min 8 AA	min 9 AA
Protein accessions	137,180	137,147	137,111	137,029	136,938
Protein nodes	135,694	135,635	135,566	135,424	135,246
Peptide sequences	5,156,229	5,025,374	4,810,649	4,579,506	4,352,226
Peptide nodes	363,315	312,103	294,008	286,826	281,175
Unique peptide nodes	116,605	116,349	115,968	115,461	114,816
Shared peptide nodes	246,710	195,754	178,040	171,365	166,359
Edges	1,671,662	1,187,153	1,056,227	1,004,580	965,739
Graphs	4,182	10,874	18,870	21,412	22,531
Graphs with 1 protein node	3,349	5,278	6,359	6,983	7,445
Isomorphism classes	270	2,656	6,902	7,896	8,162
Largest graph*					
Protein nodes	129,183	96,612	21,916	4,532	3,557
Protein nodes percentage	95.20%	71.23%	16.17%	3.35%	2.63%
Peptide nodes	354,486	238,946	55,009	11,499	8,759
Second largest graph*					
Protein nodes	25	133	641	1,270	1,097
Peptide nodes	73	255	1,421	2,570	2,172

Table B.5: Influence of different maximal allowed missed cleavages on the database bipartite graphs for D2_fasta. * In terms of number of protein nodes.

	MC 0	MC 1	MC 2	MC 3	MC 4
Protein accessions	6,344	6,344	6,346	6,347	6,347
Protein nodes	6,263	6,273	6,274	6,275	6,276
Peptide sequences	168,302	442,363	732,017	995,386	1,215,143
Peptide nodes	7,484	7,960	8,162	8,256	8,316
Unique peptide nodes	6,193	6,211	6,216	6,220	6,223
Shared peptide nodes	1,291	1,749	1,946	2,036	2,093
Edges	12,066	14,201	15,122	15,486	15,773
Graphs	5,306	5,037	4,900	4,836	4,803
Graphs with 1 protein node	4,803	4,426	4,284	4,219	4,192
Isomorphism classes	48	59	66	66	65
Largest graph *					
Protein nodes	70	92	116	151	240
Peptide nodes	188	295	356	437	616
Second largest graph *					
Protein nodes	61	70	71	71	71
Peptide nodes	186	239	269	280	287

Table B.6: Influence of different maximal allowed missed cleavages on the database bipartite graphs for D3_fasta. * In terms of number of protein nodes.

	MC 0	MC 1	MC 2	MC 3	MC 4
Protein accessions	85,955	86,264	86,316	86,321	86,327
Protein nodes	83,300	85,189	85,519	85,579	85,589
Peptide sequences	622,605	1,738,342	2,983,664	4,133,562	5,093,784
Peptide nodes	124,813	146,888	154,010	156,626	157,866
Unique peptide nodes	60,272	70,781	73,927	74,788	75,088
Shared peptide nodes	64,541	76,107	80,083	81,838	82,778
Edges	360,026	444,474	473,014	485,411	491,912
Graphs	23,037	21,953	21,660	21,494	21,458
Graphs with 1 protein node	11,719	10,918	10,728	10,665	10,657
Isomorphism classes	4,901	4,764	4,633	4,557	4,536
Largest graph *					
Protein nodes	2,193	2,522	2,661	4,396	4,534
Peptide nodes	3,708	5,614	6,352	10,353	10,810
Second largest graph *					
Protein nodes	281	309	470	312	312
Peptide nodes	388	552	1,038	609	622

Table B.7: Influence of different maximal allowed missed cleavages on the database bipartite graphs for D4_fasta. * In terms of number of protein nodes.

	MC 0	MC 1	MC 2	MC 3	MC 4
Protein accessions	136,512	136,953	137,029	137,044	137,052
Protein nodes	131,856	134,913	135,424	135,520	135,537
Peptide sequences	937,856	2,641,950	4,579,506	6,400,349	7,943,571
Peptide nodes	227,336	271,750	286,826	292,579	295,419
Unique peptide nodes	93,600	110,358	115,461	116,993	117,634
Shared peptide nodes	133,736	161,392	171,365	175,586	177,785
Edges	731,598	932,652	1,004,580	1,035,367	1,050,997
Graphs	24,085	22,055	21,412	21,172	21,108
Graphs with 1 protein node	8,582	7,315	6,983	6,897	6,884
Isomorphism classes	8,451	8,145	7,896	7,772	7,736
Largest graph *					
Protein nodes	3,391	4,045	4,532	8,246	9,014
Peptide nodes	6,128	9,588	11,499	21,595	23,827
Second largest graph *					
Protein nodes	437	868	1,270	1,277	1,277
Peptide nodes	715	1,665	2,570	2,656	2,669

Table B.8: Comparison of database and quantitative bipartite graph characteristics without and with isoforms on data set D3 (with minimal peptide length of eight amino acids an up to two missed cleavages). * In terms of number of protein nodes.

	D3_fasta	D3_iso_fasta	D3_quant	D3_iso_quant
Protein accessions	86,316	108,318	14,665	17,940
Protein nodes	85,519	107,077	8,366	8,643
Peptide sequences	2,983,664	3,091,620	20,502	19,522
Peptide nodes	154,010	189,780	7,905	7,964
Unique peptide nodes	73,927	83,740	3,808	3,538
Shared peptide nodes	80,083	106,040	4,097	4,426
Edges	473,014	686,329	18,485	19,995
Graphs	21,660	21,481	4,097	3,984
Graphs with 1 protein node	10,728	9,479	2,667	2,522
Isomorphism classes	4,633	6,163	348	370
Largest graph*				
Protein nodes	2,661	3,325	46	44
Peptide nodes	6,352	7,460	51	50
Second largest graph*				
Protein nodes	470	693	37	41
Peptide nodes	1,038	1,367	39	42

Table B.9: R packages used to implement the code for analyses and visualizations for chapters 4 and 5.

Package	Version	References	usage
batchtools	0.9.17	Lang et al., 2017	run R scripts on server
BBmisc	1.13	Bischl et al., 2022	miscellaneous
cowplot	1.1.1	Wilke, 2020	visualization
data.table	1.14.8	Dowle and Srinivasan, 2023	handling large tables
ggplot2	3.4.4	Wickham, 2016	visualization
ggpubr	0.6.0	Kassambara, 2023	visualization
igraph	2.0.1.1	Csárdi and Nepusz, 2006	handling graphs
limma	3.56.2	Ritchie et al., 2015	loess normalization
Matrix	1.6-5	Bates et al., 2024	handling large matrices
openxlsx	4.2.5.2	Schauberger and Walker, 2023	handling xlsx files
pbapply	1.7-2	Solymos and Zawadzki, 2023	progress bars
pROC	1.18.4	Robin et al., 2011	ROC curves and AUC
protiq	1.2	Gerster and Buehlmann, 2013	SCAMPI method
Rsolnp	1.16	Ghalanos and Theussl, 2015	optimization
R.utils	2.12.2	Bengtsson, 2022	miscellaneous
scales	1.2.1	Wickham and Seidel, 2022	visualization
seqinr	4.2-36	Charif and Lobry, 2007	handling FASTA files
stringr	1.5.1	Wickham, 2023	string operations
tidyr	1.3.0	Wickham et al., 2023	miscellaneous
tidyverse	2.0.0	Wickham et al., 2019	miscellaneous
vsn	3.68.0	Huber et al., 2002	LTS normalization
xtable	1.8-4	Dahl et al., 2019	generating Latex tables

Table B.10: Amount of spike-in proteins (pmol) mixed with 20 µg of in the samples for data set D1. Taken from Barkovits et al., 2020.

Protein name	Accession	Cond. 1	Cond. 2	Cond. 3	Cond. 4	Cond. 5
α-Synuclein	P37840	1.0	10.0	0.5	0.1	5.0
β-Lactoglubolin	P02754	0.5	0.1	5.0	10.0	1.0
Fibrinogen α	P02671	10.0	5.0	1.0	0.5	0.1
Fibrinogen β	P02675	10.0	5.0	1.0	0.5	0.1
Fibrinogen γ	P02679	10.0	5.0	1.0	0.5	0.1
Glucose oxidase	P13006	0.1	1.0	10.0	5.0	0.5
Hemoglobin α	P69905	0.5	5.0	10.0	1.0	0.1
Hemoglobin β	P68871	0.5	5.0	10.0	1.0	0.1
Lipase 1	P20261	0.1	0.5	1.0	5.0	10.0
Lipase 2	P32946	0.1	0.5	1.0	5.0	10.0
Lipase 3	P32947	0.1	0.5	1.0	5.0	10.0
Lysozyme	P00698	5.0	10.0	0.1	0.5	1.0
Myoglobin	P68082	1.0	0.1	5.0	10.0	0.5

Table B.11: Input data scheme for the PQP implementation. Extracted from the PQP readme file (Dost et al., 2012).

`>CC:id`	ID of the connected component
`m, n`	number of proteins and peptides
`r1, r2, ..., rn`	measured peptide ratios
`1010`	row of the biadjacency matrix
`1101`	row of the biadjacency matrix
`0100`	row of the biadjacency matrix

Table B.12: Output data scheme for the PQP implementation. Extracted from the PQP readme file (Dost et al., 2012).

`>CC:id`	ID of the connected component
`Cost`	minimal error term
`QB1, QB2, ..., QBm`	estimated protein intensities in sample B
`QA1, QA2, ..., QAm`	estimated protein intensities in sample A
`r1, r2, ..., rn`	re-calculated peptide ratios

Table B.13: Absolute frequencies of solution types per protein node type, dataset and method. Please note that the absolute numbers are not directly comparable between data sets because of the different number of pariwise condition comparisons, which are aggregated here.

Data set	Node type	Result category	bppgQuant	PQP	SCAMPI
D1	unique	single	15,914	15,914	15,914
D1	unique + shared	single	10,201	1,131	10,553
D1	unique + shared	missing	352	9,422	0
D1	shared	single	2,231	12,685	24,572
D1	shared	range	17,994	0	0
D1	shared	missing	4,347	11,887	0
D2	unique	single	20,462	20,462	20,462
D2	unique + shared	single	3,692	1,318	3,692
D2	unique + shared	missing	0	2,374	0
D2	shared	single	127	1,563	1,854
D2	shared	range	1,604	0	0
D2	shared	missing	123	291	0
D3	unique	single	2,667	2,667	2,667
D3	unique + shared	single	1,076	167	1,141
D3	unique + shared	missing	65	974	0
D3	shared	single	388	1,799	4,558
D3	shared	range	3,082	0	0
D3	shared	missing	1,088	2,759	0
D4	unique	single	32,435	32,435	32,435
D4	unique + shared	single	43,750	6,863	46,493
D4	unique + shared	range	9	0	0
D4	unique + shared	missing	2,734	39,630	0
D4	shared	single	24,174	69,200	211,280
D4	shared	range	134,216	0	0
D4	shared	missing	52,890	142,080	0

Table B.14: Median, MAD and total number of quantified proteins (n) for the different organisms (Org.), methods and node types (NT, u = only unique, u+s = unique and shared, s = only shared) for data set D4. The expected ratio for human and murine proteins are shown in column "ER". The best median and MAD values per organism/node type are printed in bold.

Comparison	Org.	Method	NT	ER	Median	MAD	n
010_025	Human	bppgQuant	u	1.3219	1.4646	0.4308	203
010_025	Human	PQP	u	1.3219	**1.4204**	**0.4005**	203
010_025	Human	SCAMPI	u	1.3219	0.4846	1.2706	203
010_025	Human	bppgQuant	u+s	1.3219	1.6039	**0.5371**	457
010_025	Human	PQP	u+s	1.3219	**1.4406**	0.6017	140
010_025	Human	SCAMPI	u+s	1.3219	1.1228	1.1891	510
010_025	Human	bppgQuant	s	1.3219	**0.4187**	2.0331	758
010_025	Human	PQP	s	1.3219	0.1100	2.0189	2,296
010_025	Human	SCAMPI	s	1.3219	0.1132	**1.7989**	6,203
010_025	Mouse	bppgQuant	u	-0.2630	-0.2098	**0.2986**	993
010_025	Mouse	PQP	u	-0.2630	**-0.2338**	0.3158	993
010_025	Mouse	SCAMPI	u	-0.2630	0.0471	0.5529	993
010_025	Mouse	bppgQuant	u+s	-0.2630	**-0.2514**	**0.2610**	1,990
010_025	Mouse	PQP	u+s	-0.2630	-0.1257	0.4158	196
010_025	Mouse	SCAMPI	u+s	-0.2630	0.0834	0.6941	2,101
010_025	Mouse	bppgQuant	s	-0.2630	-0.0754	0.8216	442
010_025	Mouse	PQP	s	-0.2630	**-0.2304**	0.7220	964
010_025	Mouse	SCAMPI	s	-0.2630	0.0486	**0.5558**	3,789
010_050	Human	bppgQuant	u	2.3219	2.3443	0.3664	188
010_050	Human	PQP	u	2.3219	**2.3321**	**0.3409**	188
010_050	Human	SCAMPI	u	2.3219	1.1445	1.8282	188
010_050	Human	bppgQuant	u+s	2.3219	2.3957	**0.4097**	466
010_050	Human	PQP	u+s	2.3219	**2.3445**	0.4647	159
010_050	Human	SCAMPI	u+s	2.3219	1.9796	1.7124	509
010_050	Human	bppgQuant	s	2.3219	**0.6462**	3.4298	597
010_050	Human	PQP	s	2.3219	0.0985	3.6164	1,781
010_050	Human	SCAMPI	s	2.3219	0.2457	**3.0807**	4,990

Table B.14: (Continued)

Comparison	Org.	Method	NT	ER	Median	MAD	n
010_050	Mouse	bppgQuant	u	-0.8480	-0.7998	**0.3488**	720
010_050	Mouse	PQP	u	-0.8480	**-0.8219**	0.3549	720
010_050	Mouse	SCAMPI	u	-0.8480	-0.1138	1.1451	720
010_050	Mouse	bppgQuant	u+s	-0.8480	**-0.8990**	**0.3233**	1,380
010_050	Mouse	PQP	u+s	-0.8480	-0.8994	0.4390	143
010_050	Mouse	SCAMPI	u+s	-0.8480	-0.4615	1.0465	1,455
010_050	Mouse	bppgQuant	s	-0.8480	-0.3408	1.5887	321
010_050	Mouse	PQP	s	-0.8480	**-0.7053**	1.5239	729
010_050	Mouse	SCAMPI	s	-0.8480	-0.1008	**1.1469**	2,759
010_075	Human	bppgQuant	u	2.9069	**2.7671**	**0.4518**	191
010_075	Human	PQP	u	2.9069	2.7302	0.4803	191
010_075	Human	SCAMPI	u	2.9069	1.5333	2.1279	191
010_075	Human	bppgQuant	u+s	2.9069	**2.8327**	**0.4051**	490
010_075	Human	PQP	u+s	2.9069	2.8100	0.4473	157
010_075	Human	SCAMPI	u+s	2.9069	2.7112	2.1704	532
010_075	Human	bppgQuant	s	2.9069	**0.8090**	4.1412	727
010_075	Human	PQP	s	2.9069	0.1256	4.5745	1,884
010_075	Human	SCAMPI	s	2.9069	0.3477	**3.7987**	5,448
010_075	Mouse	bppgQuant	u	-1.8480	-1.7504	**0.4141**	511
010_075	Mouse	PQP	u	-1.8480	**-1.7909**	0.4269	511
010_075	Mouse	SCAMPI	u	-1.8480	-0.3701	2.2885	511
010_075	Mouse	bppgQuant	u+s	-1.8480	**-1.8671**	**0.3607**	1,251
010_075	Mouse	PQP	u+s	-1.8480	-1.9618	0.3984	115
010_075	Mouse	SCAMPI	u+s	-1.8480	-0.8343	2.0931	1,324
010_075	Mouse	bppgQuant	s	-1.8480	-0.3472	3.7270	376
010_075	Mouse	PQP	s	-1.8480	**-1.4096**	3.3273	753
010_075	Mouse	SCAMPI	s	-1.8480	-0.1994	**2.4578**	2,824
025_050	Human	bppgQuant	u	1.0000	**0.7379**	**0.4574**	510
025_050	Human	PQP	u	1.0000	0.7092	0.4984	510
025_050	Human	SCAMPI	u	1.0000	0.3163	1.0674	510
025_050	Human	bppgQuant	u+s	1.0000	0.7466	0**.4645**	986
025_050	Human	PQP	u+s	1.0000	**0.7736**	0.4831	298
025_050	Human	SCAMPI	u+s	1.0000	0.6062	0.9966	1,057

Table B.14: (Continued)

Comparison	Org.	Method	NT	ER	Median	MAD	n
025_050	Human	bppgQuant	s	1.0000	**0.4506**	1.5989	816
025_050	Human	PQP	s	1.0000	0.1060	1.7027	1,969
025_050	Human	SCAMPI	s	1.0000	0.1556	**1.2592**	5,928
025_050	Mouse	bppgQuant	u	-0.5850	-0.5488	**0.3035**	670
025_050	Mouse	PQP	u	-0.5850	**-0.5749**	0.3206	670
025_050	Mouse	SCAMPI	u	-0.5850	-0.1851	0.7350	670
025_050	Mouse	bppgQuant	u+s	-0.5850	-0.6214	**0.3030**	1,378
025_050	Mouse	PQP	u+s	-0.5850	**-0.6041**	0.3644	206
025_050	Mouse	SCAMPI	u+s	-0.5850	-0.4661	0.7456	1,472
025_050	Mouse	bppgQuant	s	-0.5850	-0.3460	1.0466	427
025_050	Mouse	PQP	s	-0.5850	**-0.4371**	1.1681	901
025_050	Mouse	SCAMPI	s	-0.5850	-0.1247	**0.7596**	3,066
025_075	Human	bppgQuant	u	1.5850	**1.0995**	**0.7386**	547
025_075	Human	PQP	u	1.5850	1.0722	0.7862	547
025_075	Human	SCAMPI	u	1.5850	0.3719	1.8452	547
025_075	Human	bppgQuant	u+s	1.5850	1.1871	**0.6232**	1,127
025_075	Human	PQP	u+s	1.5850	**1.1930**	0.6414	329
025_075	Human	SCAMPI	u+s	1.5850	1.0309	1.4236	1,215
025_075	Human	bppgQuant	s	1.5850	**0.8068**	2.3627	903
025_075	Human	PQP	s	1.5850	0.2737	2.6201	2,169
025_075	Human	SCAMPI	s	1.5850	0.2923	**1.9214**	6,762
025_075	Mouse	bppgQuant	u	-1.5850	-1.5278	**0.3668**	472
025_075	Mouse	PQP	u	-1.5850	**-1.5572**	0.3696	472
025_075	Mouse	SCAMPI	u	-1.5850	-0.3935	1.8624	472
025_075	Mouse	bppgQuant	u+s	-1.5850	-1.5689	**0.3506**	1,207
025_075	Mouse	PQP	u+s	-1.5850	**-1.6009**	0.3765	191
025_075	Mouse	SCAMPI	u+s	-1.5850	-0.9127	1.5567	1,320
025_075	Mouse	bppgQuant	s	-1.5850	-0.6338	2.6089	436
025_075	Mouse	PQP	s	-1.5850	**-1.1022**	3.0414	984
025_075	Mouse	SCAMPI	s	-1.5850	-0.1484	**2.1366**	3,347
025_090	Human	bppgQuant	u	1.8480	**1.2947**	**0.8362**	553
025_090	Human	PQP	u	1.8480	1.2779	0.8894	553
025_090	Human	SCAMPI	u	1.8480	0.3870	2.1942	553

Table B.14: (Continued)

Comparison	Org.	Method	NT	ER	Median	MAD	n
025_090	Human	bppgQuant	u+s	1.8480	1.3934	**0.7108**	1,133
025_090	Human	PQP	u+s	1.8480	**1.4200**	0.7225	305
025_090	Human	SCAMPI	u+s	1.8480	0.9613	1.8284	1,209
025_090	Human	bppgQuant	s	1.8480	**1.1177**	2.4369	840
025_090	Human	PQP	s	1.8480	0.3836	2.9433	1,932
025_090	Human	SCAMPI	s	1.8480	0.2855	**2.3208**	6,293
025_090	Mouse	bppgQuant	u	-2.9069	-2.6935	**0.5105**	189
025_090	Mouse	PQP	u	-2.9069	**-2.7597**	0.5509	189
025_090	Mouse	SCAMPI	u	-2.9069	-1.1426	2.8001	189
025_090	Mouse	bppgQuant	u+s	-2.9069	**-2.7882**	**0.4631**	621
025_090	Mouse	PQP	u+s	-2.9069	-2.7880	0.4849	106
025_090	Mouse	SCAMPI	u+s	-2.9069	-2.1194	2.1803	689
025_090	Mouse	bppgQuant	s	-2.9069	-0.2513	4.6488	400
025_090	Mouse	PQP	s	-2.9069	**-0.5438**	5.3190	966
025_090	Mouse	SCAMPI	s	-2.9069	-0.1185	**4.1407**	3,112
050_075	Human	bppgQuant	u	0.5850	**0.3930**	**0.3937**	708
050_075	Human	PQP	u	0.5850	0.3643	0.4273	708
050_075	Human	SCAMPI	u	0.5850	0.1488	0.7289	708
050_075	Human	bppgQuant	u+s	0.5850	0.4796	**0.3232**	1,257
050_075	Human	PQP	u+s	0.5850	**0.5105**	0.3595	325
050_075	Human	SCAMPI	u+s	0.5850	0.5063	0.7171	1,335
050_075	Human	bppgQuant	s	0.5850	**0.3612**	1.0244	818
050_075	Human	PQP	s	0.5850	0.1434	1.1391	2,100
050_075	Human	SCAMPI	s	0.5850	0.1738	**0.694**2	6,470
050_075	Mouse	bppgQuant	u	-1.0000	-0.9246	0.3699	440
050_075	Mouse	PQP	u	-1.0000	**-0.9471**	**0.3698**	440
050_075	Mouse	SCAMPI	u	-1.0000	-0.2041	1.2329	440
050_075	Mouse	bppgQuant	u+s	-1.0000	**-0.9597**	**0.3207**	1,098
050_075	Mouse	PQP	u+s	-1.0000	-0.9243	0.4424	183
050_075	Mouse	SCAMPI	u+s	-1.0000	-0.4303	1.1377	1,188
050_075	Mouse	bppgQuant	s	-1.0000	-0.4736	1.5932	444
050_075	Mouse	PQP	s	-1.0000	**-0.5621**	1.9512	988
050_075	Mouse	SCAMPI	s	-1.0000	-0.0711	**1.3912**	3,136

Table B.14: (Continued)

Comparison	Org.	Method	NT	ER	Median	MAD	n
050_090	Human	bppgQuant	u	0.8480	**0.5620**	**0.5078**	708
050_090	Human	PQP	u	0.8480	0.5486	0.5203	708
050_090	Human	SCAMPI	u	0.8480	0.1216	1.1069	708
050_090	Human	bppgQuant	u+s	0.8480	0.6775	0.4065	1,266
050_090	Human	PQP	u+s	0.8480	**0.7155**	**0.3786**	296
050_090	Human	SCAMPI	u+s	0.8480	0.4641	0.9779	1,326
050_090	Human	bppgQuant	s	0.8480	**0.4619**	**1.2823**	822
050_090	Human	PQP	s	0.8480	0.2667	1.6719	1,976
050_090	Human	SCAMPI	s	0.8480	0.1376	1.0857	6,112
050_090	Mouse	bppgQuant	u	-2.3219	-2.0879	0.5546	198
050_090	Mouse	PQP	u	-2.3219	**-2.0966**	**0.5463**	198
050_090	Mouse	SCAMPI	u	-2.3219	-0.7855	2.3615	198
050_090	Mouse	bppgQuant	u+s	-2.3219	-2.1930	**0.4166**	595
050_090	Mouse	PQP	u+s	-2.3219	**-2.2231**	0.5205	101
050_090	Mouse	SCAMPI	u+s	-2.3219	-1.4911	1.9175	655
050_090	Mouse	bppgQuant	s	-2.3219	**-0.5746**	3.5543	385
050_090	Mouse	PQP	s	-2.3219	-0.3761	4.0680	986
050_090	Mouse	SCAMPI	s	-2.3219	-0.0571	**3.3658**	2,969
075_090	Human	bppgQuant	u	0.2630	**0.1503**	**0.3053**	874
075_090	Human	PQP	u	0.2630	0.1334	0.3338	874
075_090	Human	SCAMPI	u	0.2630	-0.0028	0.5161	874
075_090	Human	bppgQuant	u+s	0.2630	0.1839	**0.2740**	1,845
075_090	Human	PQP	u+s	0.2630	**0.2803**	0.2899	273
075_090	Human	SCAMPI	u+s	0.2630	-0.0560	0.6332	1,936
075_090	Human	bppgQuant	s	0.2630	**0.1639**	0.6759	1,066
075_090	Human	PQP	s	0.2630	0.0652	0.8146	2,472
075_090	Human	SCAMPI	s	0.2630	-0.0334	**0.5033**	8,403
075_090	Mouse	bppgQuant	u	-1.3219	-1.0989	**0.4893**	188
075_090	Mouse	PQP	u	-1.3219	**-1.1322**	0.4931	188
075_090	Mouse	SCAMPI	u	-1.3219	-0.3150	1.5162	188
075_090	Mouse	bppgQuant	u+s	-1.3219	**-1.1603**	**0.4188**	605
075_090	Mouse	PQP	u+s	-1.3219	-1.0674	0.6527	89
075_090	Mouse	SCAMPI	u+s	-1.3219	-0.6437	1.3081	688

Table B.14: (Continued)

Comparison	Org.	Method	NT	ER	Median	MAD	n
075_090	Mouse	bppgQuant	s	-1.3219	**-0.2276**	1.9922	566
075_090	Mouse	PQP	s	-1.3219	-0.1409	2.1415	1,343
075_090	Mouse	SCAMPI	s	-1.3219	-0.0543	**1.8838**	4,055

Table B.15: Result summary using the interval center for range solutions stemming from bppgQuant (bppgQuant_c). Asterisks (*) indicate that the best performing method changed compared to table 5.6 (see page 112).

Data	Part	Median	MAD	MAD	AUC
		Only shared	Only shared	Spike-ins	Total
D1	Background	*SCAMPI	PQP	*PQP	*SCAMPI
D2	Background	PQP	PQP	bppgQuant_c	*PQP
D3	Human	PQP	SCAMPI		*PQP
D3	*E. coli*	PQP	PQP		
D4	010_025 Human	bppgQuant_c	SCAMPI		bppgQuant_c
D4	010_025 Mouse	PQP	SCAMPI		
D4	010_050 Human	bppgQuant_c	*bppgQuant_c		bppgQuant_c
D4	010_050 Mouse	PQP	SCAMPI		
D4	010_075 Human	bppgQuant_c	SCAMPI		bppgQuant_c
D4	010_075 Mouse	PQP	SCAMPI		
D4	010_090 Human	bppgQuant_c	*bppgQuant_c		*SCAMPI
D4	010_090 Mouse	*bppgQuant_c	SCAMPI		
D4	025_050 Human	bppgQuant_c	SCAMPI		*PQP
D4	025_050 Mouse	*bppgQuant_c	SCAMPI		
D4	025_075 Human	bppgQuant_c	SCAMPI		*PQP
D4	025_075 Mouse	*bppgQuant_c	SCAMPI		
D4	025_090 Human	bppgQuant_c	SCAMPI		bppgQuant_c
D4	025_090 Mouse	*bppgQuant_c	SCAMPI		
D4	050_075 Human	bppgQuant_c	SCAMPI		*PQP
D4	050_075 Mouse	*bppgQuant_c	SCAMPI		
D4	050_090 Human	bppgQuant_c	SCAMPI		*PQP
D4	050_090 Mouse	bppgQuant_c	SCAMPI		
D4	075_090 Human	bppgQuant_c	SCAMPI		*PQP
D4	075_090 Mouse	bppgQuant_c	SCAMPI		

Table B.16: Average number of quantified protein nodes or groups over all pairwise condition comparisons in the four data sets. For bppgQuant, the number for the standard version as well as for the version with protein node selection is given. For MaxQuant, the numbers based on the raw, non-normalized intensities and based on LFQ intensities (which can introduce additional missing values on protein level) are given.

Data set	bppgQuant standard	bppgQuant prot. selection	MaxQuant raw intensity	MaxQuant LFQ intensity
D1	4,851	4,144	3,046	1,915
D2	722	792	729	459
D3	7,538	6,359	4,675	2,970
D4	12,590	9,675	6,512	3,673

Eidesstattliche Erklärung

Hiermit erkläre ich, Karin Ulrike Schork, dass ich die vorliegende Dissertation mit dem Titel *Improvement of protein quantification for proteins with shared peptides by using bipartite peptide-protein graphs* selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Dissertation ist bisher keiner anderen Fakultät vorgelegt worden. Ich erkläre, dass ich bisher kein Promotionsverfahren erfolglos beendet habe und dass keine Aberkennung eines bereits erworbenen Doktorgrades vorliegt.

Bochum, den 16.05.2024