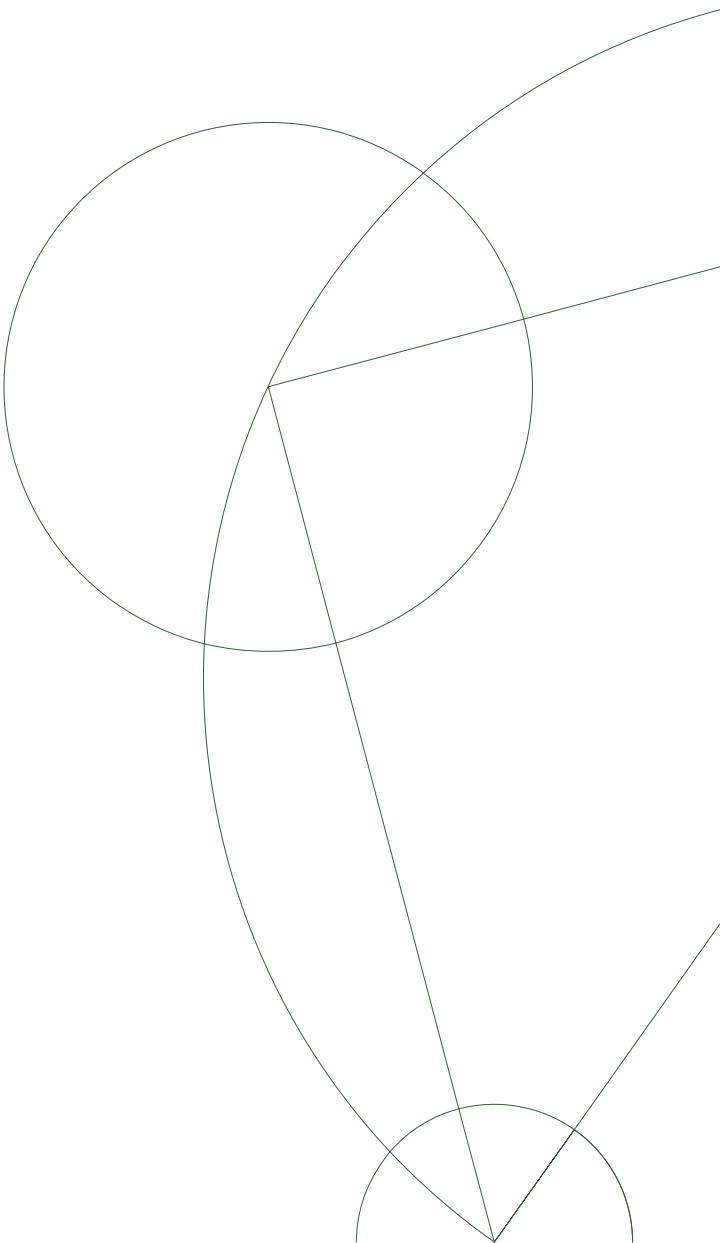
FACULTY OF SCIENCE
UNIVERSITY OF COPENHAGEN

Bioinformatics Master Thesis

Development of label-free quantification methods in proteomics

Antonio Ortega Jiménez <rnq313@alumni.ku.dk>



Supervisors

Thomas Hamelryck <thamelry@gmail.com>

Mathias F. Gruber <mafg@novozymes.com>

Contents

List of Figures

Preface

TODO: get nomenclature straight: what is a sample and what is a fraction

Chapter 1

Introduction

1.1 Novozymes and biotechnology

Novozymes A/S NZ is a company whose line of business consists of the development of products performing chemical transformations in economically relevant processes. The application of these products, instead of conventional solutions, has the advantage of requiring less chemical substances, potentially simplifying industrial processes, reducing their costs and their environmental impact. Notorious examples of such applications include waste-water treatment, household care or fermentation. For example, a product called Fermax was recently released to prevent foam development during the sugarcane ethanol production (see figure 1.1).

The company's name is inspired by the molecular agents that power these transformations, **enzymes**. Enzymes consist of a extremely diverse family of molecules catalyzing a plethora of biochemical reactions happening in living beings, from bacteria to humans.

NZ takes advantage of these molecules by matching processes where a chemical transformation takes place, with an enzyme that could improve its ef-

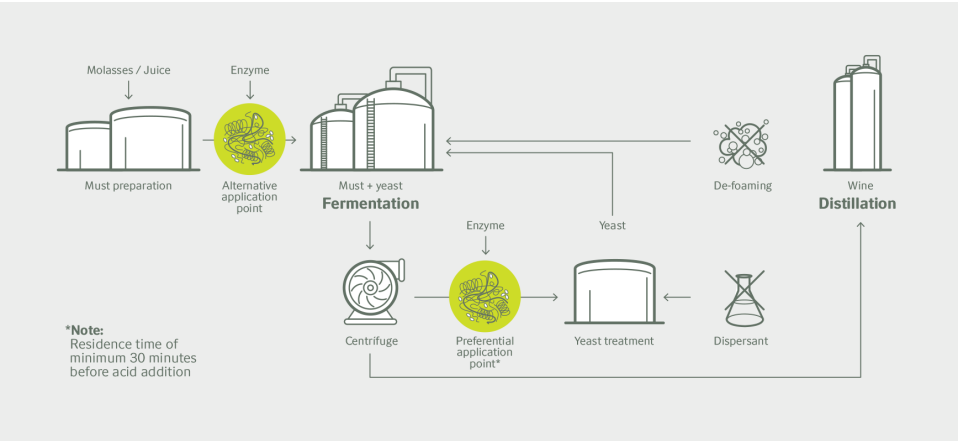


Figure 1.1 Schematic representation of a bioindustrial process optimised with a Novozymes product, marked as a green circle. Taken from Novozymes intranet.

ficiency, defined in terms of resource consumption, speed or waste subproduction. Moreover, NZ thrives to improve the performance of the selected enzymes in the target process by modifying its Wild Type (WT) form. Proteomics plays a key role on the research process required, thus NZ has a keen interested in being at the vanguard of research and tool development in this field.

1.2 Bioinformatics and proteomics

The majority of enzymes consist of proteins. But what are proteins? Proteins are molecules made up of 20 basic units, called aminoacids. All aminoacids share a common chemical structure, where a carbon atom (C_α) is covalently bonded to a hydrogen atom, a carboxyl group (COOH), an amino group (NH_2), and last but not least, a radical, also called side chain of the aminoacid. A slight deviation from this pattern exists in proline, where the radical is bound to the nitrogen atom, making it an iminoacid. The side

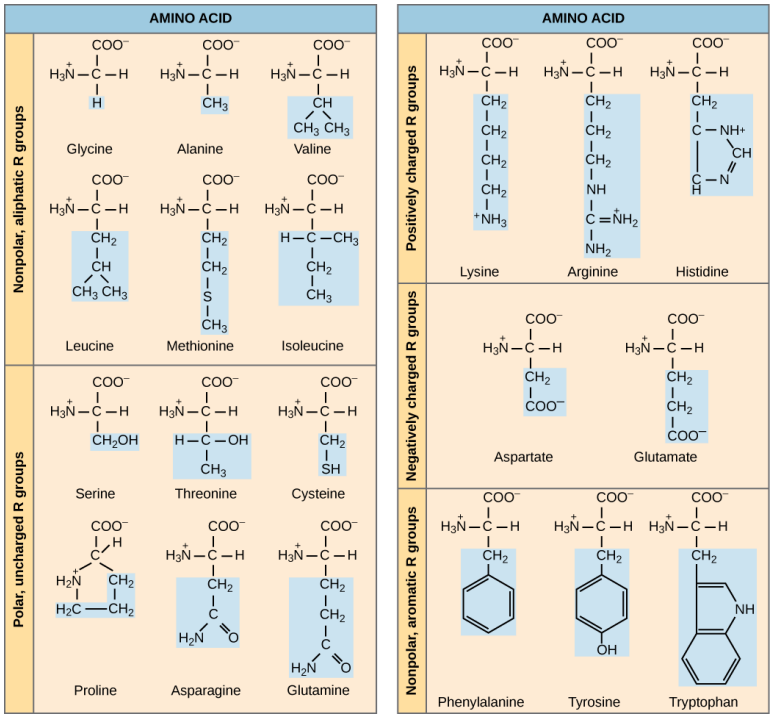


Figure 1.2 Molecular structure of the 20 aminoacids, organized by chemical properties into five groups. Taken from [1].

chain differs between aminaocids and creates the aminoacidic diversity (see figure 1.2).

The carboxyl group and amino group of different aminoacids can undergo a dehydration that generates a covalent bond, called peptidic bond, subsequently joining two aminoacids (see figure 1.3). The remaining amino and carboxyl groups can in turn be joined to new aminoacids, giving way for the formation of a biopolymer, where each subunit is an aminoacid. A peptide consists of a small chain of aminoacids joined this way. If made long enough, it acquires complex structures and becomes a protein proper [33].

Proteomics is the branch of bioinformatics whose purpose is the application of algorithms and data analysis to biological studies dealing with proteins. With proteomics, one endeavors to infer the protein composition of a sample,

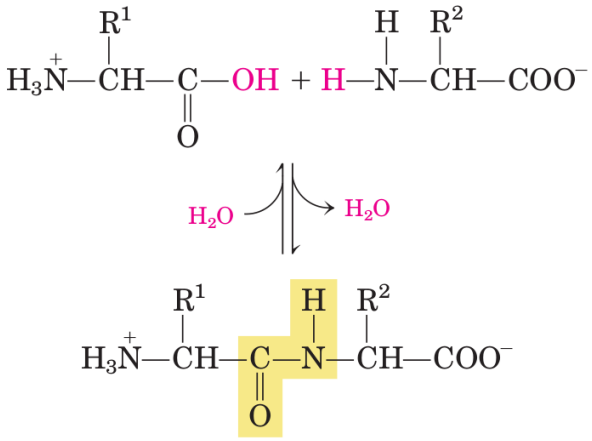


Figure 1.3 The peptidic bond brings together different aminoacids into a single protein. Taken from [33].

and eventually quantify its protein amounts.

More concretely, proteomics studies how proteins are capable of coordinating their functions by means of complex systems of regulation. Proteomics aims at unraveling the complexity, understanding how its powerful properties emerge from its individual components CITE NETWORK THEORY. For instance, the interplay of proteomics and biotechnology helps researchers learn which proteins are key in the onset of a frequent disturbance of these regulations: cancer [40]. The application of proteomics to biological problems is not restricted to medical problems like that, but also biotechnological processes like those mentioned above.

1.3 Objectives of the Thesis

In line with the goal of advancing NZ capabilities in proteomics research, with the final ambition of speeding the research turnover in the company, this project aimed at the following:

1. Develop an open-source, Linux based and easily deployable pipeline for the analysis of proteomics data, starting at the raw high-throughput data files and ending in the biological interpretation of the results.

2. Propose a label-free quantification probabilistic model that provides relative abundance estimates and a measurement of their uncertainty based on the available data.

3. Evaluate the tools with benchmark datasets to assess their performance and check they convey the biological phenomena captured in the data.

1.4 Structure of the Thesis

An overview over the main data collection technique used in proteomics, Mass Spectrometry (MS), and the ensuing computational data analysis steps is presented in chapter 2. An inspection of the developed pipeline is provided in chapter 3, while the probabilistic model is introduced in chapter 4. A benchmark of both tools on a NZ dataset is given in chapter 5. Finally, a conclusion of the work is given in chapter 6.

Chapter 2

Review on mass spectrometry (MS) and shotgun proteomics

The main source of data in proteomics is MS. However, different approaches to how these technique is used make two paradigms in proteomics analysis: top-down and bottom-up. In the top-down paradigm, intact proteins are directly used for the analysis. In the bottom-up paradigm (see figure 2.1), the proteins are first cleaved into smaller parts, and these parts are then used for identification, characterization, and quantification. These smaller parts are peptides [20], consisting of around 10 chained aminoacids. Such peptides acquire physicochemical properties fitting the requirements of the downstream analytical methods, mainly the mass spectrometer (MS), which performs the data acquisition. The bottom-up paradigm is most often used because peptides are much more suitable to analysis by mass spectrometry, as illustrated in section 2.2.3. This review will focus on the data-dependent approach within bottom-up proteomics, which currently is the most frequent workflow in proteomics [42]. In this approach, the equipment is configured

for the analysis of one peptide at a time, with the drawback of being biased for the most abundant peptides.

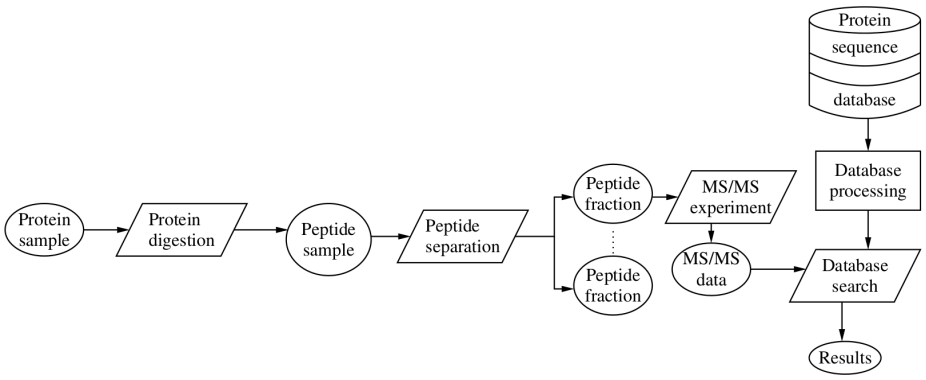


Figure 2.1 Diagram over a standard bottom-up proteomics analysis. Figure 1.3 from [20].

MS is performed by means of a mass spectrometer, an ensemble of pieces of equipment that can acquire mass measurements for eventually thousands of sample components. A detailed explanation of the sample processing required prior to MS is given in section 2.1, while an overview on mass spectrometers is given in section 2.2. The result of the MS analysis is a dataset that, with adequate computational analysis tools, is enough to perform the inference steps required to gather knowledge about the original protein sample. These inference steps can be condensed to the peptide and protein inference problems, explained in section 2.6. A third computational problem needs to be solved if quantitative, and not just qualitative information, is to be gained from the experiment. This is the quantification problem, explained in section 2.7.

A summary of the bottom-up approach MS analytical pipeline is provided in the rest of the chapter. It can be divided into two main steps:

1. MS analysis and data generation. Sections 2.1 to 2.3.

2. Computational analysis of data. Sections 2.4 to 2.7.

2.1 Sample processing

Prior to its introduction in the mass spectrometer for shotgun proteomics studies, a protein sample (I) is cleaved into peptides and (II) these peptides are separated by means of some physicochemical properties. This way, the analytical equipment works with one peptide at a time.

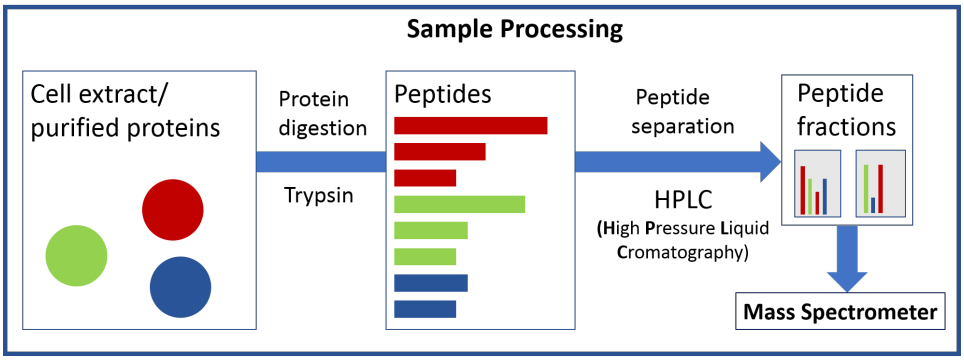


Figure 2.2 Diagram of the sample processing step prior to mass spectrometry analysis. First, proteins are denatured and digested with a specific protease like Trypsin. This yields a peptide mix that is separated into peptide fractions that can be introduced in the mass spectrometer.

2.1.1 Protein digestion

An MS experiment starts with the generation of a protein sample from the biological system of interest. Proteins are then denaturalized so as to remove bias due to the divergent properties acquired by folded proteins. Then, proteins are subjected to digestion with specific proteases i.e protein-cutting molecules, which cut the aminoacidic chains following a predictable pattern. Trypsin is the most frequently used protease. It cuts peptidic bonds whenever a positively charged residue, either Lysine (K) or Arginine (R),

lies on the carboxyl side of the peptidic bond. Since roughly 1/10 residues are either R or K, the average peptide length is 10 residues, as mentioned before. As demonstrated in 2.2.3, this length distribution is fitted to the resolution of the MS analyzer. Moreover, in as much R and K are positively charged aminoacids (see figure 1.2), the resulting peptide is guaranteed in most cases to be able to capture at least one charge, which is key in the MS workflow as described in section 2.2. All of these properties combined, together with its low prize, makes Trypsin the protease of choice in this step for most cases.

Even though proteases are very specific, the cleaving process is far from perfect, as there could be: [20]

1. Missed cleavages.
2. Unsuspected cleavages during the maturation/life cycle of the protein.
3. Unexpected cleavages occurring either in the wet-lab procedure of the proteolytic treatment.
4. Naturally occurring, intentionally or unintentionally induced chemical modifications.

Missed cleavages can happen due to steric impediments or the presence of specific aminoacids that can weaken the enzyme's function. This is the case of Trypsin whenever the residue on the other side of the peptidic bond is Proline. Altogether, a variability is created in the cleavage process that, though limited, needs to be taken care of in downstream analysis, as it could introduce biases in peptide observability.

The result of this process is a complex mix of peptides, made up by hundreds

or thousands of different molecules, following a length distribution given by the cleavage sites frequency and each protein's aminoacidic composition. A peptide separation step is required before introducing the sample in the spectrometer.

2.1.2 Peptide separation

If presented with the problem of analyzing a mixture of peptides, the capacities of mass spectrometers are easily overwhelmed by a too complex mixture, resulting in the analysis of only a minor part of the total protein of the sample. This can be surmounted by analyzing one peptide in the sample at a time. The required sample separation is achieved by High-Performance Liquid Cromatography (HPLC) methods, like reverse phase chromatography (separating on hydrophobicity) and strong cation exchange chromatography (separating on isoelectric point) [20].

During HPLC, the peptide mix is loaded into a column containing a stationary and a solid phase. These phases create an environment where peptides interact differently based on their physico-chemical properties, set by the nature of the phases. The output of the column, called elute, will consist of subsets or fractions of peptides leaving the column at different retention times (RT) i.e the amount of time passed before the peptide is observed in the mass spectrometer. Therefore, the input to the spectrometer will consist of one peptide at any given time.

2.2 The mass spectrometer

The mass spectrometer is the ensemble of pieces of equipment analyzing a peptide like as those generated following the workflow enunciated in sections 2.1 and 2.1.2. It consists of three main parts: an ion source, a mass analyzer, and a detector (see figure 2.3).

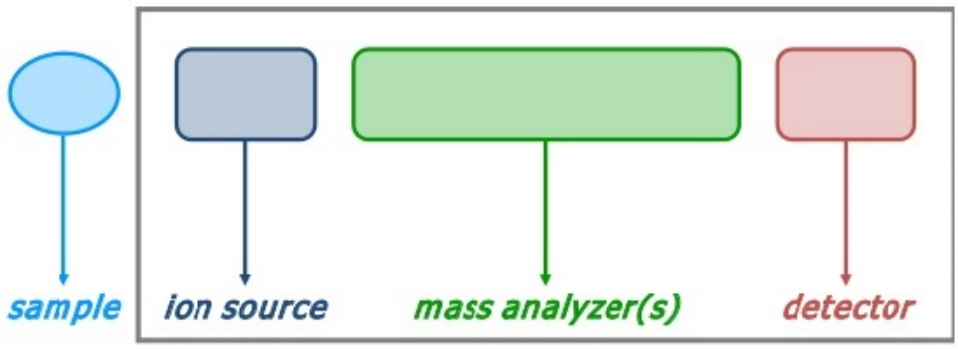


Figure 2.3 Schematic view of a mass spectrometer. Taken from [1]

2.2.1 The ion source

All mass spectrometers exploit the physical properties of mass and electric charge exhibited by the analyzed components. Ionization of the analytes is absolutely essential prior to any measurement, as analytes left uncharged will be unobservable to the equipment.

This step is performed in the ion source [20]. The most frequent ionization methods in proteomics are Matrix-Assisted Laser Desorption-Ionization (MALDI) and Electro Spray Ionization (ESI) [32]. Most peptides ionized by MALDI will acquire a single charge, whereas ESI can provide multiple charges (+2, +3, etc) too. Thus, the charge exhibited by an ion is not ob-

[1] https://www.slideshare.net/joachimjacob/bits-introduction-to-mass-spec-data-generation

vious when produced via ESI. Moreover, ESI can be run online with the right robotic equipment, while MALDI demands waiting time for vacuum generation. Finally, due to the chemical nature of the matrix components, MALDI ionizes more easily peptides containing aminoacids featuring aromatic rings (PYW), thus introducing a bias. Bias in ESI is less predictable. All the sources of bias introduced during ionization are grouped into the competitive ionization problem [46].

The acquired charge yields a mass/charge (m/z) ratio, a property that is applied in the downstream component separation and measurement steps.

2.2.2 The mass analyzer

The plethora of ion separation methods is reflected upon the range of different analyzers available, mainly time of flight (TOF), Ion trap (IT) and quadrupole (Q). These apply different principles to perform the same task: separation (analysis) of the ion mix by the m/z ratio.

Moreover, two other analyzers exist which combine mass analysis with intensity measurement. These are Fourier Transform Ion Cyclotron Resonance (FT-ICR) and Orbitrap. They both register cylotron resonance frequencies that are Fourier transformed into the spectrum space. Remarkably, FT-ICR exhibits great resolving power, at the cost of high maintenance costs and difficult operability [20].

2.2.3 The detector

Detectors measure the intensity of an incoming ion signal. The ion's m/z ratio is known thanks to the previous mass analysis step. Performed for

enough m/z ratios, the detector can produce a MS spectrum, which shows the intensity of ion current over an m/z range. Some topics in signal detection in MS need to be discussed.

On the one hand, the precision of the signal measurement is given by its mass resolution It is conventionally defined as the closest distinguishable separation between two peaks of equal height and width [15]. The resolution decreases as the m/z ratio increases because small increments in the m/z ratio become negligible at high m/z ratios. This is one of the reasons why proteins are better fit for analysis when digested into peptides, as m/z are reduced, thus increasing the mass resolution.

On the other hand, due to the natural occurrence of isotopes, particularly ^{13}C, the same peptide will induce the measurement of several signals with very close m/z values. They constitute the isotopic envelope of the ion (see figure 2.4), and represent the signal created by peptides containing an increasing number of ^{13}C atoms. Every time a ^{12}C is replaced by ^{13}C, the mass increases by 1 Da. Even though the natural abundance of ^{13}C is 1.1 %, the sheer number of carbon atoms in a peptide makes it likely that at least one or even more carbon atoms will be ^{13}C, eventually driving the pure ^{12}C signal to comparatively small intensity values, and down to intensities below the background noise. Such event can be problematic if it entails that the ^{13}C peak is confused for the ^{12}C peak.

The resolution achieved by modern equipment allows for the distinction of each individual signal in most isotopic envelopes. Remarkably, the separation across peaks in the envelope can be used to infer the charge of the peptide, as increases of 1 Da at charge 1 will induce a separation of 1 m/z, while at charge 2 it will be $1/2 = 0.5$ m/z, at 3 $1/3 = 0.33$ m/z, and so on

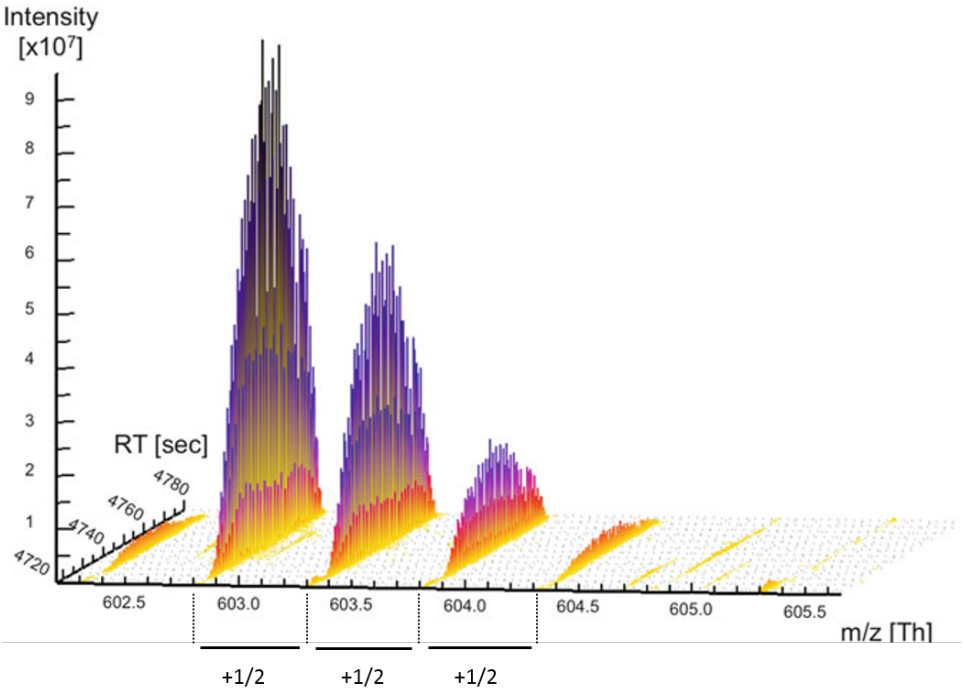


Figure 2.4 A doubly charged isotopic envelope with its monoisotopic ion measured at 602.8 Th. Each peak in the envelope is separated by 0.5 Th. This is explained by the peptide having 2 positive charges that make every extra Da in the ion mass account for 1/2 extra Th. Adapted from [32].

(see figure 2.4).

It is up to the MS technician to decide on the best pieces of equipment according to their availability and particularities of the dataset.

2.3 Tandem MS workflow

Shotgun proteomics analyses make use of two or more mass spectrometers connected in series, giving rise to the so-called Tandem MS (MS/MS) workflow. In this setting, each mass spectrometer collects a different type of spectra and thus different information (see figure 2.6. An extra spectrome-

[3]https://www.slideshare.net/joachimjacob/bits-introduction-to-mass-spec-data-generation

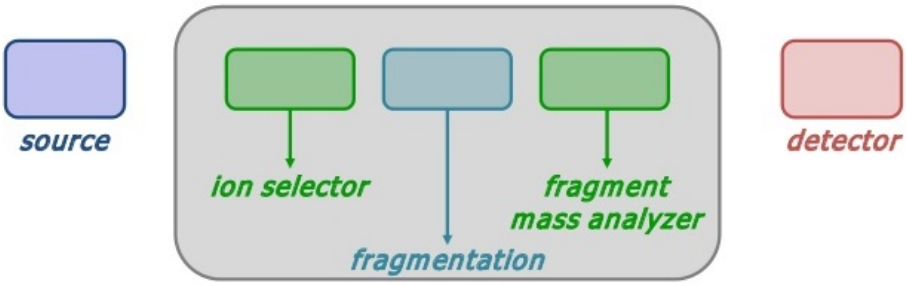


Figure 2.5 Illustration of the tandem MS workflow. The first spectrometer acts as an ion selector, that not only registers spectra, but also lets through ions with a given m/z ratio. The second spectrometer does not perform mass analysis, but instead provides the medium where peptide fragmentation (see section 2.3.1) occurs. Finally the third spectrometer records fragment mass spectra. Taken from [3].

ter, usually a quadrupole, is introduced in between.

- The first spectrometer records the intensity versus m/z ratio of the peptides eluting from the column at a given time and is used to filter ions exhibiting a selected m/z ratio.
- The ions filtered in the first spectrometer undergo fragmentation in the second spectrometer.
- The last spectrometer records the intensity versus m/z ratio of the fragments produced in the previous step. The produced spectrum can be use to read out the peptide sequence.

2.3.1 Fragmentation

The information that can be extracted from MS1 scans consists of the m/z ratio and retention time of the peptide ion, but not its sequence. Having the latter is paramount if the spectrum is to be matched to a theoretical peptide.

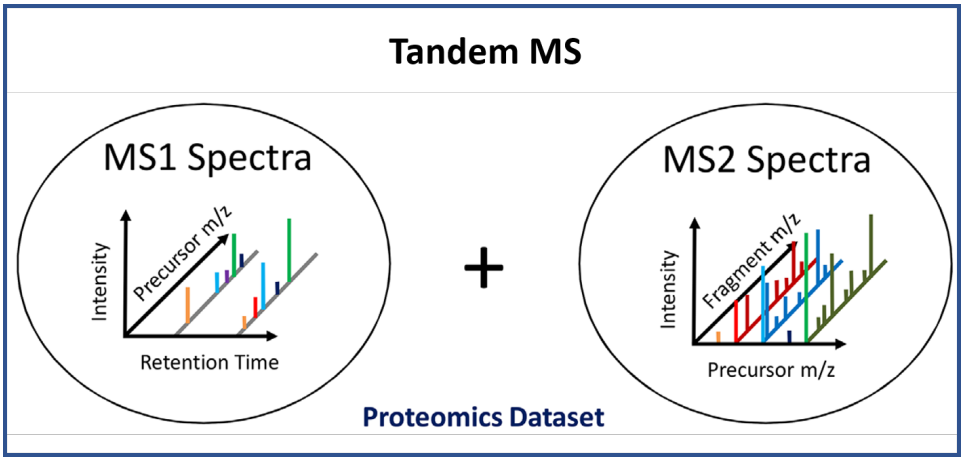


Figure 2.6 Illustration of the different spectra collected in tandem MS. MS1 spectra record precursor intensity vs m/z ratio at different times. MS2 spectra record the same magnitudes but the signal is generated by the fragments produced during fragmentation by the ion filtered in the first spectrometer. Altogether, they enable peptide identifications. Figure adapted from [49].

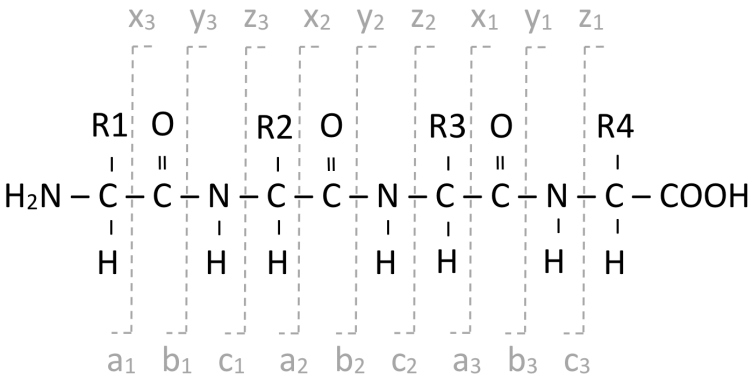


Figure 2.7 The common fragments and their relation to the peptide sequence can be organized into 2 groups of 3 series each. The abc fragments keep the N-terminal residue, while xyz keep the C-terminal one. Specific fragmentation techniques make fragments belonging to one series more likely than others. Other fragments are possible but much less likely. The fragment nomenclature was introduced in [38].

Fortunately, the peptide sequence of the ion can be inferred if fragmentation is performed. During peptide fragmentation, bonds along the peptidic chain are broken, turning the peptide into smaller fragments. These fragments will

consist of truncated versions of the original peptide at different positions, thus making it possible to read an m/z ratio difference between any pair of fragment ions. The difference can be exploited to deduce which aminoacid makes up for that difference. If this process is repeated for enough pairs of contiguous fragments, with the right software, a sequence can be read from the MS2 spectrum, as explained in section 2.4.

2.4 Spectra processing: search engines

Computational analysis of MS data starts with the matching of the spectra to a referene proteome (see figure 2.8). MS search engines are capable of performing the crucial step of peptide to spectrum matching (PSM). During this step, search engines perform *in silico* prediction of the peptides produced in the protein digestion step and their expected spectrum.

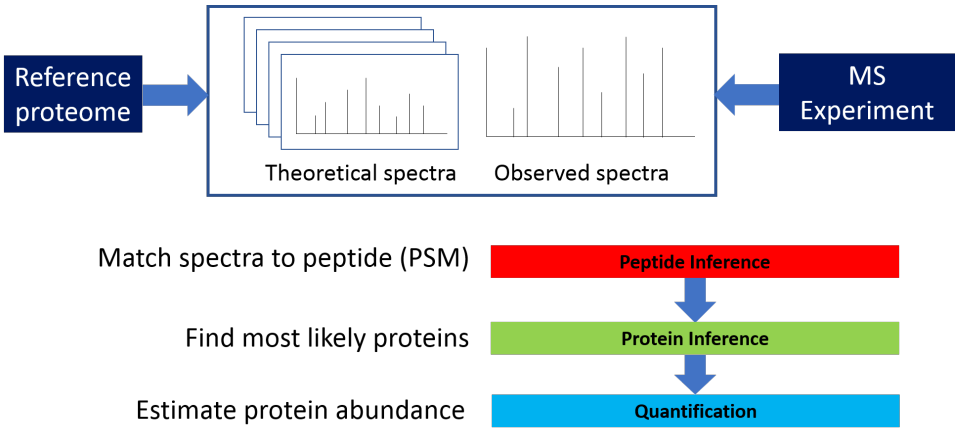


Figure 2.8 Diagram of the MS computational analysis. The PSM process maps the observed spectra to a list of peptides in the reference proteome that could have generated them. In other words, the software infers what peptides were introduced in the spectrometer. The analysis continues during protein inference and quantification.

Given the stochastic nature of the protein cleavage and spectra recording

processes, the resulting data exhibit variability manifested in both missing and spurious peaks. Furthermore, random (wrong) matches can be returned by the PSM process when running against a sufficiently big database. This translates to the generation of multiple matches, of which one, if any, will be correct. Therefore, the lists of matches need to be somehow ranked by goodness-of-fit. The issue is addressed by search engines through the deployment of statistical models that provide scoring systems. Assuming the correct protein is present in the database, a good scoring system should give the best score to the right peptide. Under this circumstances, if repeated for several peptides, enough evidence for the presence of individual proteins can be collected.

Multiple search engines exist that implement different matching and scoring algorithms. The most modern ones include MS-GF+, MS-Amanda, Comet, X!Tandem or Andromeda. Notably, the results of each individual search engine can be combined to gather their strengths, at the expense of an increased computational cost and time [41].

2.5 Validation and quality control

The scoring systems implemented in search engines provide the best matches, but they are bound to contain false identifications. Nevertheless, these scores can be used to apply a filter that aims at minimizing the amount of errors.

A common filter is the false discovery rate (FDR), usually set to 1%, indicating that after the its application, only one out of a hundred filtered matches are expected to be false positives (wrong matches).

The most commonly used method to compute the FDR of a list of matches is the target-decoy search. Using this method, the search engine replicates the matching process, using the same spectra, but instead against a decoy database. The decoy database is generated by reversing or more generally applying a randomization technique upon the sequences present in the original database (target).

All matches to the decoy are by definition wrong. Since the basic properties of the decoy (size, composition, etc) remain identical to those of the target the amount of matches to the decoy exhibiting more than a given score s can be regarded as an estimate of the number of false identifications ($\hat{n}_{fp}$) in the list of target results exhibiting at least the same score. This is because the existence of shared properties entails that random matches are equally likely to happen in both databases [14]. Together with the number of PSMs passing a given score in the target ($n_{tp} + n_{tp}$), the FDR can be computed using equation 2.1.

$$FDR = \frac{\hat{n}_{fp}}{n_{fp} + n_{tp}} \tag{2.1}$$

The equation tells us that we can compute the FDR at any score by counting how many decoy hits have a greater score ($\hat{n}_{fp}$), and dividing by the length of our target hits list. Thus, the score that makes the FDR equal to a predefined value, frequently 0.01 or 1 %, can be computed and used as threshold for the target hits.

The minimal FDR at which a given PSM is considered a valid match constitutes the PSM's q-value, i.e it is the smallest FDR we can allow while still keeping the PSM. Related to the q-value, the Posterior Error Proba-

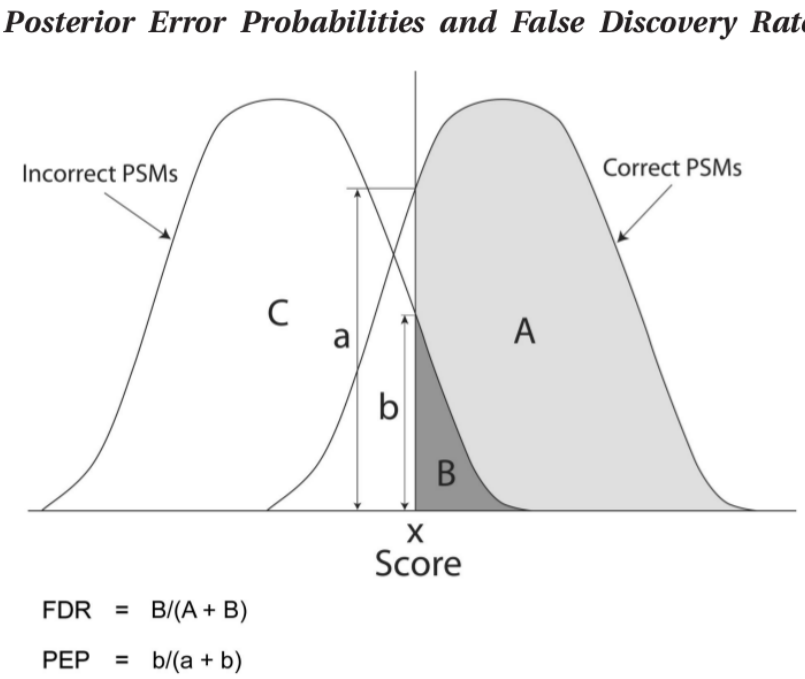


Figure 2.9 Visualizing FDR and PEP. The FDR at a given score s is defined as the ratio between False Positives and the sum of True and False Positives found in the list of PSMs of score greater than s. The PEP corresponds to the same ratio, but only at a specific score, hence its alternative name of local-FDR. Figure from [22].

bility (PEP) is an estimate of the probability of a given PSM of being an incorrect assignment (see figure 2.9). The PSM's confidence is just defined as $1 - PEP$ [34]. PEP can be computed from the decoy search results and provides another useful measurement of the uncertainty in the target results.

2.6 Peptide and protein inference

Two steps in protein identification can be distinguished:

1. **Peptide inference**: infer the peptides present in the sample.

2. **Protein inferece *proper***: based on the inferred peptides, infer what

proteins generated them. This is not trivial as peptides are degenerate and frequently map to more than one protein.

The result of the PSM step returns the inferred list of peptides. The ensemble of proteins most likely to have generated the list of peptides stemming from the filtered PSMs can be inferred using different algorithms. The degenerate nature of peptides is dealt with the Occam's razor principle, which states that the most likely solution is the simplest one. Thus, protein inference algorithms aim at explaining the maximum amount of peptides using the least amount of proteins.

2.7 Protein quantification

The combination of all the aforementioned computational analyses yields a list of protein identities that reports the protein composition i.e qualitative information of the original sample. However, in most proteomics applications, quantitative data can be of great interest, as many biological phenomena are manifested mainly through changes in the protein abundances, rather than protein presence alone. For instance, cancer cells in response to a drug could modulate the abundance of several proteins without removing them from the cytosol or introducing new ones.

Protein quantification pipelines can be classified based on whether isobaric labelling was used (label-based) or not (label-free). These are explained in subsection 2.7.1. If the label-free approach is employed, more distinctions can be made based on:

- The proxy used for quantification: spectral counting (SC) or extracted ion currents (XIC). These are explained in 2.7.2

- The way the data are brought to the protein level from the peptide level: summarization-based vs. peptide-based, explained in 2.7.3.

2.7.1 Label-based and label-free approaches

Two paradigms exist in protein quantification: label-based and label-free. In label-based quantification, originally identical peptides from a number of different samples are made distinguishable by their masses via the incorporation of a label. All label-based methods simultaneously analyze several samples in each experiment, removing the difficulties associated with between-run variability [20]. The finite number of "plexes" available for a given label sets the limit to how many samples can be differentially quantified [9]. Different techniques, like Stable Isotope Labeling by Amino acids in Cell culture (SILAC) or Isotope-Coded Affinity Tags (ICAT), differ in the nature of the label and the way it is introduced. Remarkably, peptide labeling costs can be quite high. This, together with the limited amount of samples that can be compared makes a case for label-free quantification.

In the label-free quantification approach, peptides from different samples are not labelled differently and are thus distinguished by their presence in different, independent MS runs. In order to account for the introduced inter-run variability in peptide identifications and RT, a match-between-runs (MBR) processing step can be carried out (see figure 2.10).

This way, extra identifications are attained through a reanalysis of the spectra, factoring in the information collected in replicate runs. The RT and precursor mass of unidentified spectra in one replicate is matched to that of identified spectra in the remaining replicates. As a consequence, more identifications with a lesser fraction of missing datapoints are achieved.

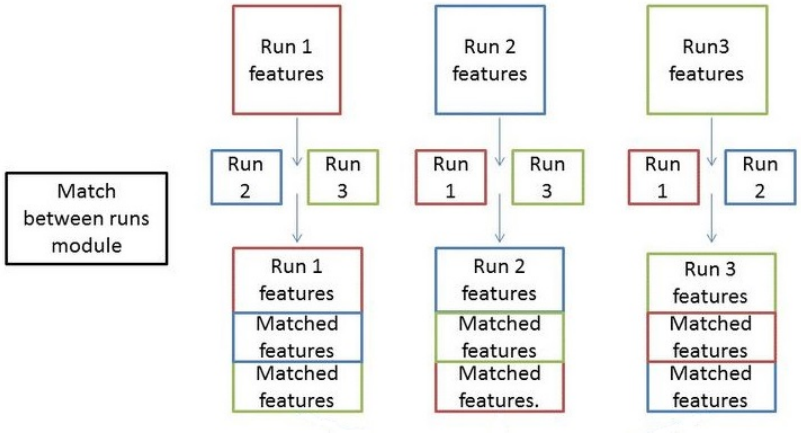


Figure 2.10 Illustration of the MBR step. **A** the information gathered from matches in replicate runs is put to use during a reanalysis of the spectra. Modified from supplementary information in [1].

2.7.2 SC and XIC based quantification

Quantification can be SC or XIC based.

On the one hand, spectrum counting based quantification is the simplest method in proteomics. The number of peptides a peptide is detected in the spectrometer is detected as proxy for its abundance. It relies on the rationale that highly abundant peptides will have a higher intensity and are thus more likely to trigger the acquisition of MS/MS spectra. These methods have the advantage that they are very simple to implement and don't require any further data processing.

On the other hand, XIC based methods rely on intensity measurements at any level of the MS workflow as proxies for protein abundance. A wide range of algorithms are available to process these data and output estimates of protein abundance. All of them require an intensive preprocessing step, usually including (I) taking the log_2 intensity to make the distributions symmetrical and thus fit for diverse parametric tests, and (II) quantile normalization to

address between-runs variability in the intensity measurements. They can be classified in the bases of which MS level is used as proxy for the protein abundances and on whether or not a summarization step is performed to aggregate peptide-level data into protein-level data, or not.

As stated in [9], *although the abundance of proteins and the probability of their peptides being selected for MS/MS sequencing are correlated to some extent, XIC-based methods should clearly be superior to SC given sufficient resolution and optimal algorithms. These advantages are most prominent for low-intensity protein/peptide species, for which a continuous intensity readout is more information-rich than discrete counts of spectra.* For this reason, only the XIC approach will be regarded in the rest of the manuscript.

For a more robust XIC based quantification, a feature extraction step is usually executed to extract the apex intensity of the identified peak clusters, as shown in figure 2.11.

2.7.3 XIC-peptide-based models for label-free quantification

The data collected in the mass spectrometer refers to peptides originating from a latent protein composition, given by the original sample. However, the data interpretation requires the transfer of these peptide-level data into the protein level. This can be done by either (I) performing an aggregation of the peptide-level data, where a summary value of the peptide-level data is taken as representative for the protein-level data, or (II) performing the protein quantification directly at the peptide-level by means of linear regression models.

As stated in [18] *peptides originating from the same protein can indeed be*

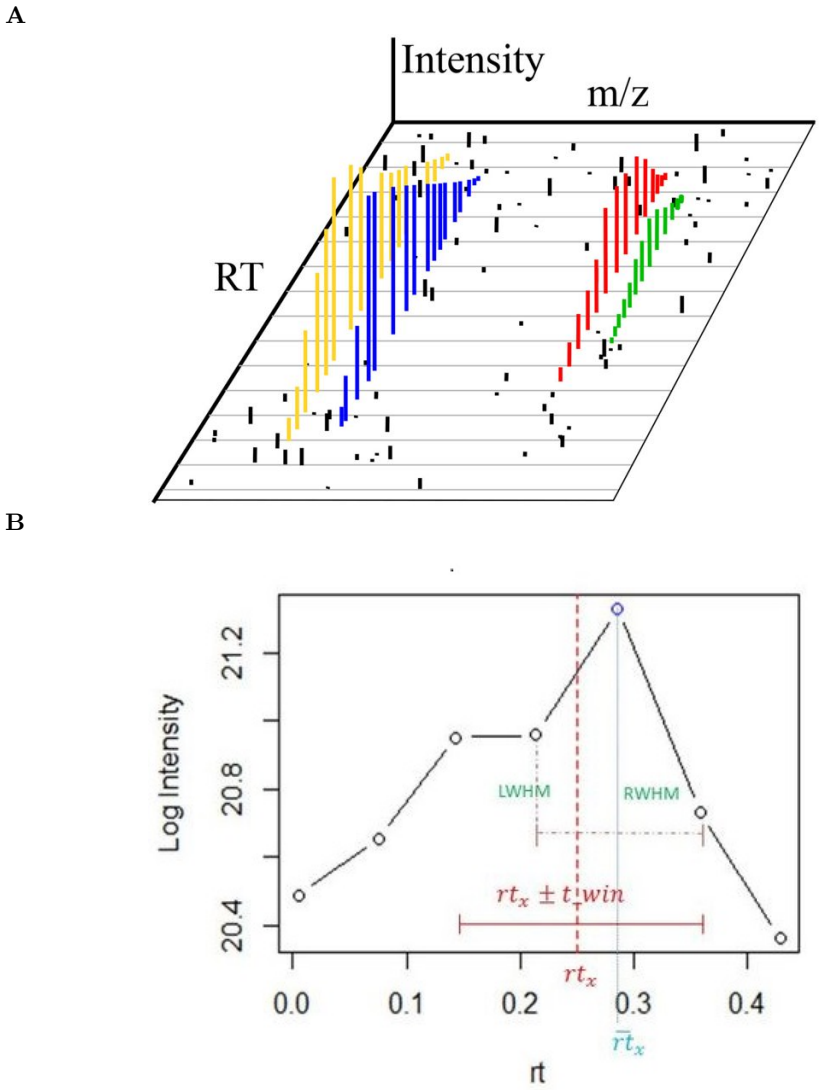


Figure 2.11 Illustration of the apex intensity extraction step.**A** Visualization of a 3D peak cluster on the RT / m/z plane. An ion current at the same m/z ratio, thus corresponding to the same precursor ion, is detected with varying degrees of intensity over a more or less narrow time window, spreading the signal over time. Taken from [43]. **B** A refinement analysis of these data attempts to fit a mathematical model of the signal over time and extract a representative measurement, such as the highest (apex) MS1 intensity. Modified from supplementary information in [1].

considered technical replicates and theoretically should lead to similar abundance estimates. However, the summarization of the peptide intensities into

protein expression values is cumbersome, and most summarization-based methods do not correct for differences in peptide characteristics or for the between-sample differences in the number of peptides that are identified per protein. This might introduce bias and differences in uncertainty between the aggregated protein expression values, which are typically ignored in downstream data analysis steps.

It is for this reason that peptide-based models offer the statistical framework required to learn as much from the data as possible. This translates into improved results when compared to the other aforementioned methods [18]. This hypothesis is the motivation for the method explained in chapter 4.

Chapter 3

A label-free quantification proteomics pipeline

Summary

A pipeline making use of the set of tools published by the Compomics and StatOmics groups SearchGUI [2], PeptideShaker [48], moFF [1] and MSqRob [17] , was developed to support complete label-free protein quantification analyses using the most recent advances in the field with open-source software. The pipeline can be run on Linux computer clusters to perform (I) peptide to spectrum matching against a reference database, (II) quality control and filtering, (III) MBR and feature extraction, (IV) protein inference and (V) relative quantification. Its output can be passed to follow-up analyses in R or Python to get a biological interpretation of the results. A benchmark of its performance was accomplished using the proteome benchmark dataset published in [9]. The results were comparable to those achieved by the MaxQuant [9] software, excepting a bias produced by the sample fractionation of this dataset.

3.1 Introduction

3.1.1 Background

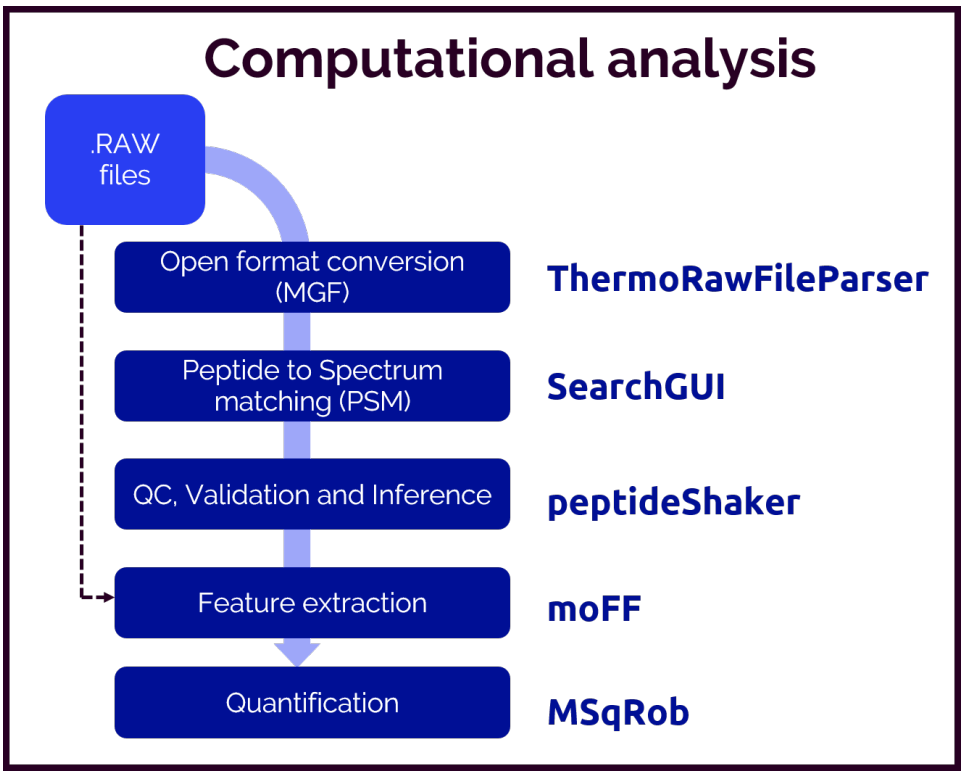


Figure 3.1 Schema of the presented pipeline. First, .RAW files produced by the spectrometer are converted to an open format like the Mascot Generic Format (MGF). After that, spectra are searched by means of a search engine to perform the PSM step. Thereafter, PSMs are validated and the most likely set proteins is inferred. If quantative information is to be extracted, a quantification step attempting to estimate protein quantities is executed. The biological interpretation of the pipeline results can be achieved by interacting with public databases using R/Bioconductor or Python thanks to their open format.

Several proteomics pipelines are available on the internet under different licensing conditions. Many are released as closed-source commercial software, where information on how the program works is kept from the user. This is a serious drawback as it hinders the study of the implemented models and its customisation. Open-source, free alternatives, like the Trans Pro-

teomic Pipeline (TPP) [12] or openMS [44] are nevertheless available for the community, but they either lack good documentation, good broad base or format exchangeability. MaxQuant, a freeware but closed-source, monolithic proteomics analysis suite [8], is developed for Windows and has only very recently been ported to Linux [42]. Nevertheless it has been extremely successfully adopted by the scientific community due to its ease of use and comprehensive pipeline.

3.1.2 Goals

The development of a pipeline attempting to achieve the following goals will be described in this chapter.

1. Fully command line, documented and Linux-supported interface for easy automation and scalability.
2. Open-source for easy customisation and extension.
3. Free-licensed and cost-free, so anybody with the knowledge can run it, democratizing the analyses.

3.2 Materials and Methods

3.2.1 Data generation and loading

The proteome benchmark dataset from [9] was reanalysed starting at the output RAW files available at the PRIDE repository [1]. Briefly, the *Homo sapiens* and *E. coli (strain K12)* proteomes were mixed in 1:1 (condition

[1] https://www.ebi.ac.uk/pride/archive/projects/PXD000279

L) and 1:3 (condition H) proportions, with 3 replicates for each combination. Moreover, each of the 3 replicates of the 2 conditions was analysed over 24 fractions. This experimental setting thus generated a total of $2 \times 3 \times 24 = 144$ RAW files. One file was missing in the repository. The ThermoRawFileParser [2] program was used to convert RAW files to the MGF open format.

The MQ+LFQ pipeline results were obtained starting at the Supplemental table 1 from the MaxLFQ paper supplemental data [3] [9], which is available as an excel file. The results of the MQ+Rob pipeline employed the Levenberg-Marquandt minimised peptide intensities stored in the peptides.txt file contained in the spectraHeLaEColi.zip in the supplemental data.

3.2.2 Decoy database preparation and search

The spectra saved in the MGF files obtained in the previous step were passed to the MS-GF+ search engine [24] by means of the SearchGUI `SearchCLI` tool version 3.3.3 [2] utility. The search parameters were set using the `IdentificationParametersCLI`. In order to account for potential post-translational modifications, the search was conducted allowing for the following variable modifications: oxidation of M and deamidation of N and Q. Moreover, C carbamidomethylation was set as fixed modification. The enzyme was set to semispecific Trypsin, allowing for a non-tryptic cleavage on any side of the peptide. Up to two missed cleavages were allowed. The precursor tolerance was 10 ppm and the fragment tolerance 0.5 Da.

The target database was created by combining the Uniprot proteomes for *E.*

[2] https://github.com/compomics/ThermoRawFileParser
[3] http://www.mcponline.org/content/13/9/2513/suppl/DC1

coli (strain K12) (UP000000625) and *Homo sapiens* (UP000005640), downloaded in June 2018. The decoy database was created using the `FastaCLI` utility in SearchGUI by reversing all sequences in the target.

3.2.3 Quality control and validation

The SearchGUI results were filtered using the default built-in checks available in the PeptideShaker version 1.16.22 utility `PeptideShakerCLI` [48] [4]. By default, the FDR was set to 1%. PEP and confidence statistics were computed using the PeptideShaker built-in algorithms. Output was extracted via the Default PSM report txt file, available in the `ReportCLI` utility.

3.2.4 Data refinement

The moFF command line utility [1] was applied to perform (I) match-between-runs and (II) extract MS1 apex intensity of each peak cluster. This required passing the original RAW files, together with the Default PSM report from PeptideShaker. Output was exported to a peptide summary file, containing one row per peptide and for every peptide, the detected apex intensity in each sample.

3.2.5 Quantification

Relative quantification was performed using the MSqRob utility by passing the peptide summary file from moFF. Prior to quantification, the data was preprocessed using the `preprocess_MSnSet()` function. In a nutshell, (I) MS1 apex intensities were log_2 transformed, (II) quantile normalized, (III)

[4] https://github.com/compomics/peptide-shaker/issues/300

peptides belonging to protein groups that contained one or more proteins that were also present in a smaller protein group were discarded [17], and (IV) protein groups with only 1 peptide were dropped.

Once preprocessing is done, a ridge regression model with Huber weights and empirical Bayes estimation of protein variance, implemented in the MSqRob package [17], was fit to every protein individually. The peptide and fraction effects were considered random, while the condition was treated as a fixed effect. The significance of the treatment effect differences was assessed through a Student's T test, implemented in the `test.contrast()` function. Only protein groups that could be unambiguously mapped to one of the organisms were considered for the analysis.

Quantification in the MQ+LFQ pipeline was executed using the LFQ intensity columns stored in the supplemental data file. Intensities were log2 transformed and averaged. The difference between conditions H and L was taken as estimate of the log2FC. Significance was evaluated by means of the two-tailed Welch Two Sample t-test implemented in the `t.test()` function in R. P-values were corrected using the FDR method implemented in the `p.adjust()` function in R.

3.2.6 Code implementation

3.3 Results

3.3.1 Evaluation of the PSM step

The preprocessing of the RAW files produced by the mass spectrometer into the MGF open format enabled searchGUI to dispose of the registered

spectra. PeptideShaker quality control and filtering capabilities (see figure 3.2) carried out the required search results validation. As expected, matches to the target and decoy exhibited similar score distributions at low score values, while a divergence is observed at higher score values. Likewise, the m/z error was found to be closer to 0 on validated PSMs than on those which did not pass the 1 % FDR filter. The application of this filter implied that the FNR (false negative rate) was set to 5 %, i.e 5 out of every 100 discarded matches were estimated to be true positives.

The 1% FDR cutoff selected PSMs with a score higher than 78, which translated to a confidence of at least 65%.

The combination of both programs enabled the identification and validation (matching) of thousands of spectra with high confidence in all samples. However, when compared to the total amount of spectra available, the percentage of matched spectra was on average 37.4%, with a marked decrease starting at fraction 14 (see figure 3.3).

3.3.2 Protein inference

Total peptides	Unique peptides	Non-unique	Proteins	Protein groups
46595	27384	19211	25056	10039

Table 3.1 Protein inference results. The counts of the different molecular entities detected by peptideShaker are displayed.

A total of 46595 peptides were inferred to be present in at least one of the fractions (see table 3.1). Of them more than 27 thousand mapped to a unique protein. However, a significant amount mapped to more than one protein. Such peptides are called non-unique peptides. The 25056 detected

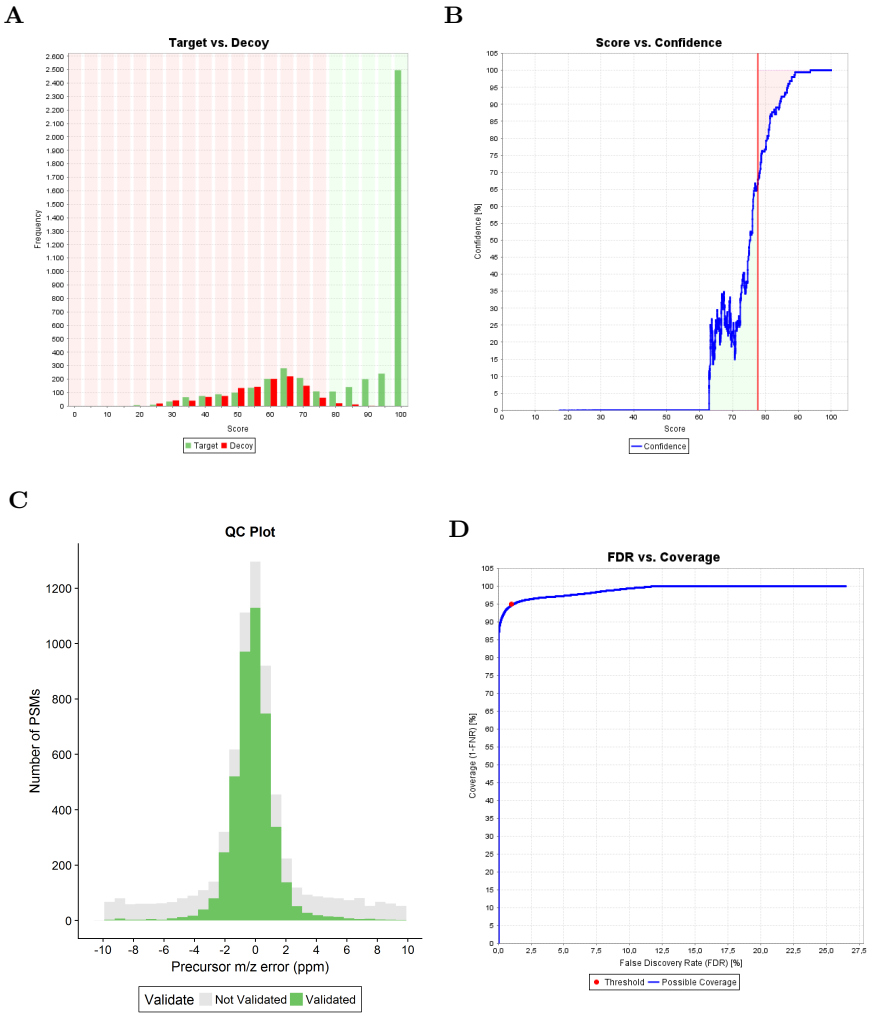


Figure 3.2 Quality control and validation of the 13th fraction of the third replicate in condition L A Score distribution for matches to the decoy and the target databases. **B** Evolution of the PSM score with confidence. The implemented cutoff at FDR of 1 % is displayed with a red vertical line. **C** Distribution of the difference between the predicted and measured m/z values, segregated by validation status. **D** ROC curve built upon the number of false positives and negatives estimated from the decoy search. The cutoff is displayed as a red dot.

proteins were grouped into 10039 protein groups. Protein groups are peptide generating entities for which enough data to confirm the presence of at least one of them is available, but not to exactly assess which of them. Thus, protein inference algorithms create them when a does not uniquely map to

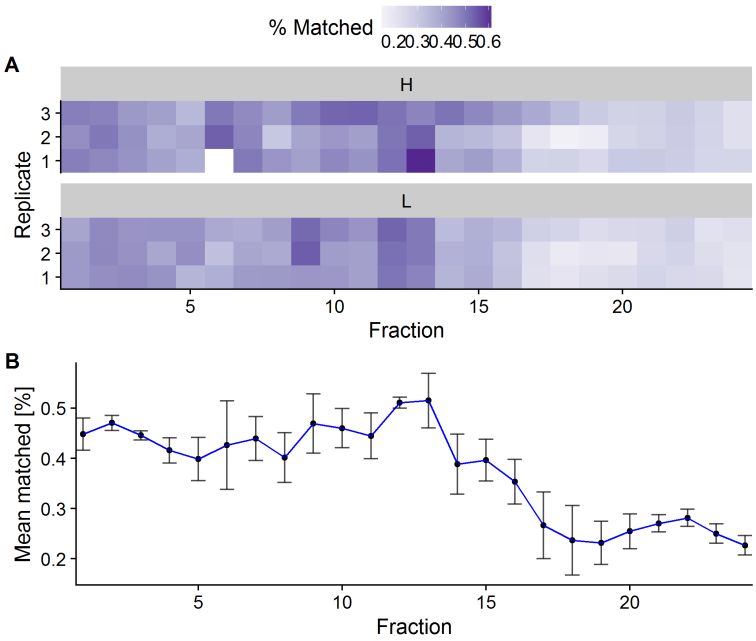


Figure 3.3 The total number of spectra per sample ranged between 5894 and 20249. **A** Percentages are encoded with a blue palette, the darker, the higher, and viceversa. **B** The mean for each analysed fraction across conditions and replicates is displayed together with error bars to represent the standard deviation. The sixth fraction of the first replicate in condition H was missing in the PRIDE data repository.

a protein because it is not possible to distinguish which protein it truly comes from.

Protein groups are very frequent due to several factors. For example, protein isoforms, consisting of protein sequences differing in potentially only one aminoacid, are very difficult to resolve. PeptideShaker executes a smart protein grouping by harnessing the annotation of proteins and classifying protein groups based on how well the annotation backs the protein group.

However, in the present analysis the quality of the protein group was not taken into account, as explained in the Materials and Methods.

3.3.3 MBR and apex intensity extraction evaluation

The match between runs step allowed for increased identifications by transferring successful matches between replicate runs. The results of this process for the 13th fraction of the L condition is shown in figure 3.4.

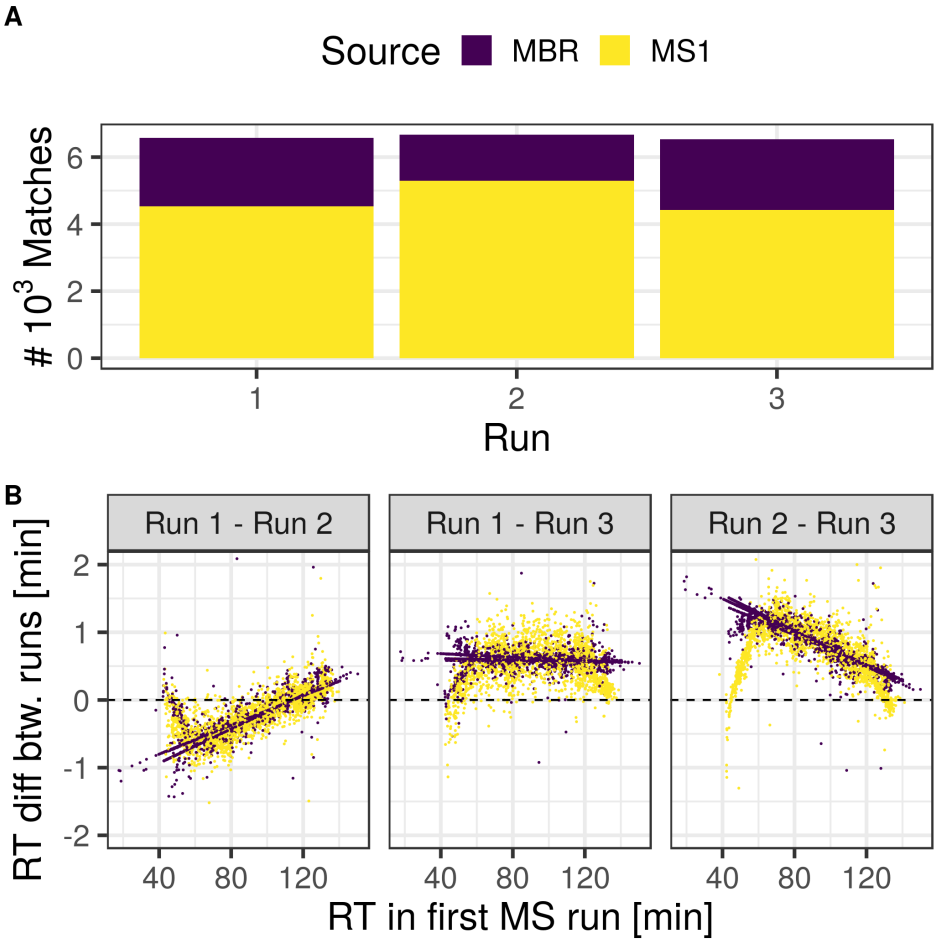


Figure 3.4 Match Between Runs with moFF. **A** Count of identifications on each run segreagated by source. More than 4k spectra were identified and validated by SearchGUI+peptideShaker. In this particular case, hundreds of new identifications were accomplished. **B** Match visualization. Every dot represents a peptide shared across 2 runs. The coordinate system illustrates its retention time on the first run on the x axis and the difference with the second run on the y axis. The color depicts whether the identification was carried out during the PSM process (MS1), or thanks to a cross-identification achieved by the MBR module (MBR).

Once as many identifications as possible were gathered, a refinement of the measured MS1 intensity can be implemented to select the apex of every peak cluster, which yields robust intensity measurements for each sample (see figure 3.5).

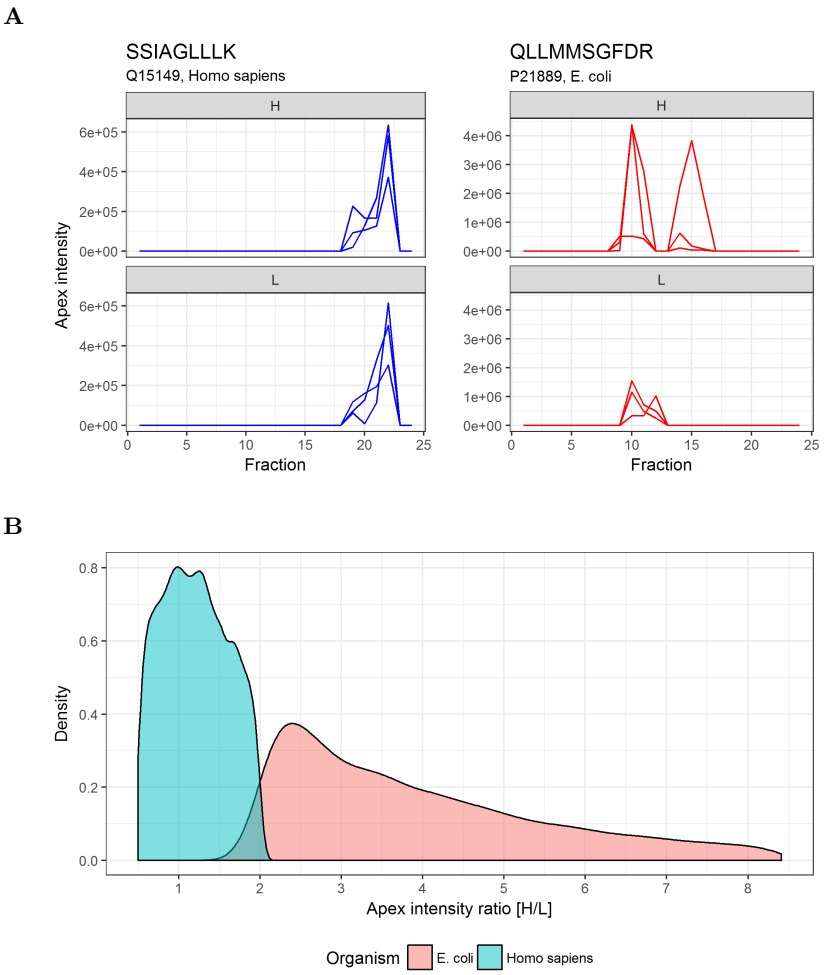


Figure 3.5 A The apex intensity profile across fractions for 2 different peptides, one from *Homo sapiens*, and one from *E. coli*. The figure illustrates the intrinsic intensity variability between technical replicates, particularly in the case of the *E. coli* QLLMMSGFDR peptide, as it was almost non-existent in one of the runs. **B** The expected pattern of overall similar intensities for the *Homo sapiens* data and 3-fold higher intensities for the *E. coli* data in condition H was observed, confirming an acceptable performance of the protocol.

3.3.4 Quantification benchmark

The extracted apex MS1 intensities were used as proxy for peptide abundance to estimate protein $\log_2$(ratios), or *log2-fold-changes* (log2FC) across condition H and condition L. During quantification, it is important to take into account as many of the effects influencing protein quantification. In this case, besides the different H and L conditions, peptide, run and fraction effects could be distinguished. The MSqRob quantification engine treats effects differently depending on whether or not they are random, or fixed. Fixed effects should have a consistent impact in the ion currents measured in the spectrometer, whereas random effects are truly random and thus unpredictable.

The result of the process was the successful quantification of 5307 protein groups out of 10039 (see table 3.2).

	log2FC	qval	Protein	*Organism*
1	-4.76E-01	1.42E-24	P78527	*Homo sapiens*
2	6.80E-01	6.75E-24	P0A8V2	*E. coli*
3	8.50E-01	4.39E-22	P25516	*E. coli*
4	7.07E-01	2.05E-18	P63284	*E. coli*
5	1.04E+00	1.21E-17	P37095	*E. coli*
6	7.78E-01	1.51E-16	P13029	*E. coli*
7	8.89E-01	5.62E-16	P77804	*E. coli*
8	8.53E-01	1.72E-15	P23721	*E. coli*
9	-6.97E-01	1.41E-14	P09874	*Homo sapiens*
10	1.37E+00	1.41E-14	P37666	*E. coli*

Table 3.2 Results of the quantification pipeline. The 10 protein groups with the lowest *q-val* as estimated by MSqRob are shown. Notably, eight correspond to proteins belonging to *E.coli*. The two human proteins exhibited in both cases an absolute value of the log2FC below 1.

The performance of the quantification can be evaluated thanks to the experimental design of the dataset. Since the *E. coli* proteome was mixed on

a 3:1 ratio in condition H, a log2FC of $log2(3) = 1.58$ is expected for proteins coming from this organism. Likewise, human proteins should have a log2FC of 0. As such, the evaluation can be reformulated as a classification task were the quantification algorithm tries to distinguish between proteins coming from one or the other organism.

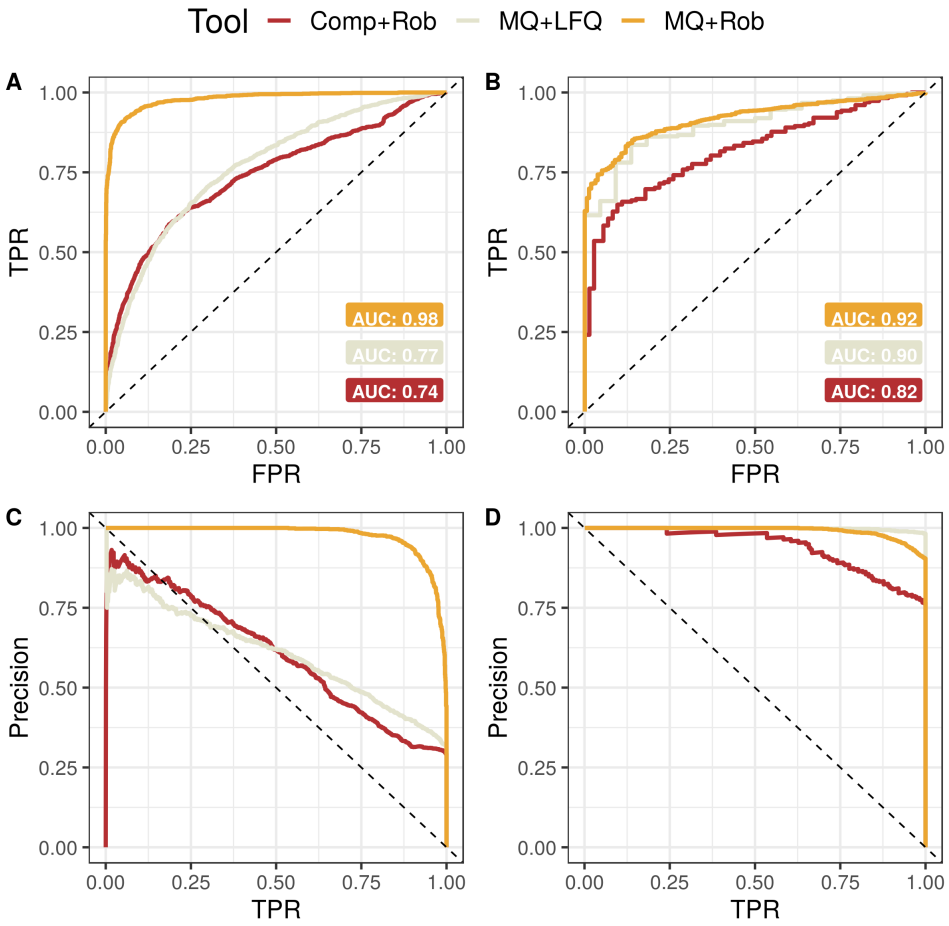


Figure 3.6 ROC and PR curves displaying the behaviour of 3 different label-free quantification pipelines publicly available. **Comp+Rob** is the pipeline presented in the previous sections. **MQ+LFQ** consists a standard MaxQuant upstream analysis combined with its LFQ quantification engine. Finally, **MQ+Rob** blends MaxQuant upstream analysis with the MSqRob quantification engine. **A, B** ROC curve produced by the three algorithms with no log2FC filter, and keeping proteins with log2FC greater than 1, respectively. **C, D** PR curve produced by the three algorithms with no log2FC filter, and keeping proteins with log2FC greater than 1, respectively.

Some of the most frequent evaluations of the performance of a binary classifier are Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves [4], which demonstrate the performance of the algorithm at different cutoff values of a predictor variable. Together, they show the False Positive Rate (FPR), the True Positive Rate (TPR) or recall, and the precision exhibited by the algorithm.

In the present problem, they can be used to show how good the *q-val* associated to a protein group is at separating the two proteomes. Thus, *E. coli* proteins are treated as positives, and those from *Homo sapiens* are treated as negatives. Moreover, a filter based on the log2FC can be applied to enrich the data in positives and facilitate the classification task. In order to compare the results of the here presented pipeline (Comp+Rob) with those achieved by other available tools, the analysis was repeated using the data processing implemented in MaxQuant [8] at different steps. The pipeline combining MaxQuant and MaxLFQ (MQ+LFQ) was fully carried out using the MaxQuant suite, whereas the combination of MaxQuant and MSqRob (MQ+Rob) used MaxQuant up to the fraction normalization step.

The result of this analysis is shown in figure 3.6. Remarkably, Comp+Rob achieved similar results to those resulting from MQ+LFQ. Nonetheless, the latter performed clearly better when the log2FC filter was applied, indicating that a combined fold change and q-value criteria is best for declaring proteins as differentially abundant. In all cases MQ+Rob was as good as MQ+LFQ, if not much better.

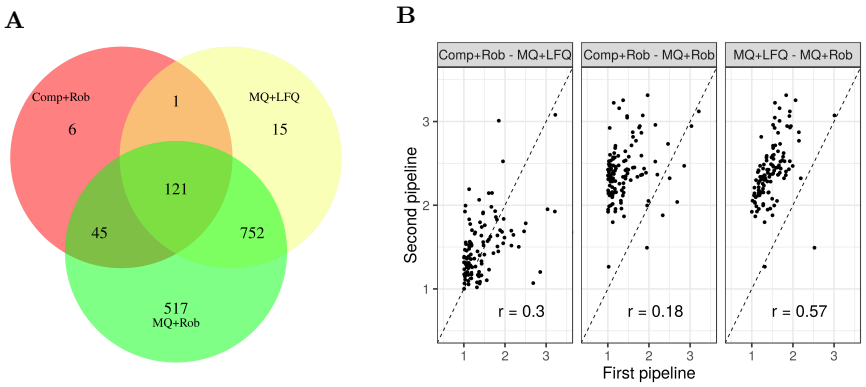


Figure 3.7 Agreement between the different pipelines. **A** The Venn diagram displays the number of *E. coli* proteins found to be differentially abundant by each pipeline. **B** The log2FC estimated in all three pipelines for the 121 shared proteins is plotted by pairs to evaluate their correlation.

In order to visualize the level of agreement of the pipelines for the assignment of differentially abundant proteins, a Venn diagram was constructed (see figure 3.7 A). Venn diagrams provide a straightforward visualization of the overlap of several ensembles of proteins. The scrutiny of the counts reveals a high degree of agreement, with most proteins detected by Comp+Rob or MQ+LFQ being detectd by MQ+Rob too. Remarkably, 517 proteins were detected by MQ+Rob alone, which further confirmed its resolving power. Analyzing the log2FC pairwise-correlation (see figure 3.7 B) reveals that the highest correlation was found between pipelines using MaxQuant as upstream spectra processor, as expected. Even though the observed correlation was low, the estimated log2FC was always greater than 1. The MQ+Rob pipeline systematically estimated log2FC values greater than the other 2.

Additionally, the accuracy with which each program assigns a log2FC estimate to each protein group, and how significantly different from the null value of 0 it was found to be, provides further insights on the pros and cons of each pipeline, as shown in figure 3.8. This analysis confirmed that the

pipeline providing the best separation between organisms was MQ+Rob.

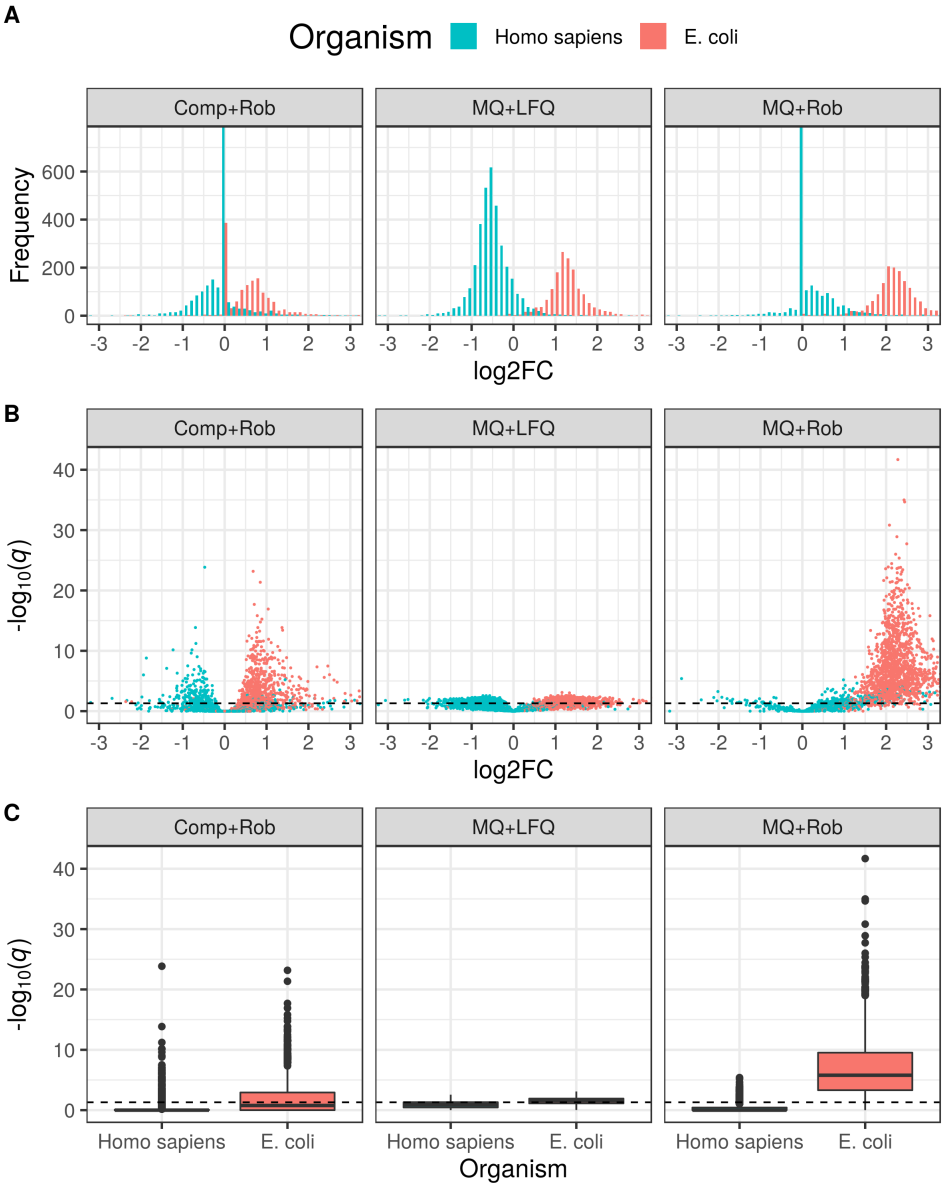


Figure 3.8 Overview over the different pipelines'performance. Quantification results are segregated by the source organism. **A** Histogram of the log2FC estimates. **B** Volcano plot, where every protein group is represented by a dot and the coordinates map to its log2FC on x and the minus logarithm of the q-value on y. **C** Boxplots of the minus logarithm of the q-value distribution.

3.4 Discussion

3.4.1 Improvement of the PSM step

The low attained matching rate manifests existing room for improvement in the currently available tools. As explained in section 2.4, it has been shown that the combined usage of multiple search engines can increase identifications, since the different statistical frameworks implemented in each of them compensate each other's caveats. Only MS-GF+ was used in thi work for simplicity. Furthermore, *de novo* search engines are available and supported by SearchGUI, and the latest developments in the field, like the IdentiPy engine [30] are progressively incorporated to the tool [5]. Their usage is predicted to improve identification rates and quality of the matches.

The most important reason why many spectra remain unidentified is the presence of post-translational modifications (PTMs), which exponentially increase the search space, forcing most workflows to discard many of the peptides featuring a PTM. New approaches to the problem are emerging, mainly machine learning methods for the handling of unexpected modifications [16]. Moreover, the prediction of MS2 peak intensities patterns from peptide sequences promises to increase the amount of evidence available during the PSM process, thus boosting correct identifications [25] [11].

Finally, the extremely low matching rates in the latter fractions translated to decreased contributions to the number of identifications. This indicated that decreasing the number of fractions would not have had a major impact in the experiment's depth.

[5] https://github.com/compomics/peptide-shaker/issues/309

3.4.2 Improvement of the feature extraction step

While the apex intensity is a good estimate of the peptide abundance, other methods are available that could potentially provide more robust data, such as the area of the peak. This is the approach taken by the recently published tool RawQuant [26], or the feature detection tool in the OpenMS pipeline [44]. Moreover, in order to fine tune the performance of the MBR and Apex intensity extraction steps, several input arguments can be adapted to each dataset. These are the spectrometer's m/z resolution, the time window used during the apex search and whether or not to activate outlier filtering and what weights to use.

3.4.3 Improvement of the quantification step

The deployed relative quantification approach successfully manages to estimate the log2FC of several of proteins in the dataset with high confidence. However, it was not found to be the best performing of the assayed pipelines in the benchmark dataset. Several axes of improvement are clear from the results.

Management of sample fractionation

Sample fractionation allows for a deeper analysis of the peptide mix, which in turn leads to increased identifications and an increased amount of data to be analyzed. However, it psimultaneously provides an extra random effect that difficults quantification if not handled properly.

The way the MaxLFQ engine solves the problem is by aggregating the data from the same sample across fractions using a ponderated average. One nor-

malizing weight is determined for every sample via the *Levenberg-Marquant minimisation of the overall proteome variation* $H(N)$ as defined in [9]. A custom implementation was written in Python using the scipy module, but the results were not satisfactory.

MSqRob treats the fractions as a random effect, as explained in section 3.3.4. As a consequence, the statistical model in MSqRob can be overwhelmed by the inconsistency resulting from a 24-plex sample fractionation, leading to null log2FC estimates for many *2*E. coli proteins. This is clearly manifested in figure 3.8 as the bins at log2FC of 0, which represent protein groups for which MSqRob did not find enough evidence to discard the null hypothesis. While no *E. coli* proteins fell in this category in the MQ-MSqRob pipeline, many did in the Comp+Rob, proving room for improvement in this aspect.

In any case, the way sample fractionation is handled should depend upon the fractionation method. For example, membrane fractionation can be exploited when working with membrane proteins [31].

To sum up, the presently implemented fraction handling is correct and provides good results, but fine-tuning of the pipeline at this step is expected to improve the results of the quantification. Thanks to the open-source nature of the software, this is indeed possible.

The pitfalls of mass spectrometry

The application of mass spectrometry to proteomics is a very hot topic in research, driving innovations in the way the different components of the spectrometer work. One recent example is the development of the timsTOF™ Pro (Trapped Ion Mobility Spectrometry-Time Of Flight) system by Bruker,

which supports PASEF (Parallel Accumulation and SErial Fragmentation) spectrometry [6]. This technology adds an extra dimension peptide separation process, leading to cleaner spectra. Notwithstanding, the developments also reveal the immature state of the technology. Slight inconsistencies in the protein extraction and digestion, bias in the peptide ionization, presence of unexpected PTMs, and detector saturation and insensitivity, remain as problems contributing to the eventual measurement variability characteristic of mass spectrometry [35]. Thus, adequate preprocessing of the peptide dataset, and care to ignore outliers is preeminent. Indeed, the input file for the MQ+LFQ consisted of a preprocessed file which removed the most problematic proteins [9], which explains why more than 500 extra proteins could be quantified when using MSqRob (see figure 3.7 A).

Support for absolute quantification

While relative quantification like that performed by MSqRob is enough to infer differential abundance of proteins across conditions, absolute quantification would provide an estimate of protein quantities that would support many other analyses, and facilitate comparison between datasets, in a way similar to what the normalised read counts in FPKM (Fragments Per Kilobase per Million Reads) does in transcriptomics. On the other hand, MaxLFQ provides absolute estimates, albeit less robust.

[6] https://www.prnewswire.com/news-releases/bruker-launches-the-timstof-pro-mass-spectrometer-to-enable-the-revolutionary-pasef-method-for-next-generation-proteomics-300520791.html

Quantification of uncertainty

All the quantification approaches enunciated until now make use of different frequentist approaches to the problem, that evaluate how significantly different the data are from what would be expected under a null hypothesis of a null ratio across the pair of conditions. Yet, it does not provide probabilistic interpretations of the uncertainty behind the estimate. The development of a Bayesian framework which endeavors at solving this issue is presented in chapter 4.

3.4.4 Applicability for Novozymes data

The SearchGUI and PeptideShaker recommend using Uniprot-formatted databases for optimal performance of their tools [7]. This is due to the software making use of the annotations to improve the protein inference step. In any case, a consistent formatting in the FASTA databases is required to used the tools. Thus, the development of sequence databases featuring consistent FASTA identifiers is highly advisable to make the best use of the data and improve the interplay with these and many other tools.

3.5 Conclusion

An open-source, cost-free, platform-agnostic and customisable label-free relative quantification pipeline assembling publicly available tools was presented in this work. The pipeline is capable of providing qualitative information on the protein composition of a sample using the latest proteomics

[7] https://github.com/compomics/searchgui/wiki/DatabaseHelp

search engines and validation software. Moreover, it can be extended to support more complex tasks, like *De novo* search, or the study of PTMs, thanks to its non-monolithic nature and usage of open-formats. Likewise, the openness of the data in all steps allows for quality control checks along the way. However, the comparison against other workflows shows the room for improvement in the quantification step when dealing with fractionated data, albeit deficiencies in this aspect should have no impact in non-fractionated datasets.

Either the pipeline developed in the presented chapter or those based on MaxQuant are ready to be deployed in a real environment and provide NZ with useful insights on its proteomics data for free.

Chapter 4

BayesQuant: Probabilistic estimation of protein ratios

Summary

The current label-free quantification methods reviewed in section 2.7 rely on frequentist statistics, which return point estimates of model parameters such as the estimate of the abundance ratio (fold change) across conditions. However, a Bayesian approach to this problem, that we know of, is lacking in the literature. As a response to this shortcoming, a statistical model, named BayesQuant, was written in the probabilistic programming framework PyMC3 and tested on the same benchmark dataset from chapter 3). The execution of the three steps required when doing Bayesian modelling, mainly (I) model implementation, (II) computation of posterior probabilities and (III) model checking, will be described for this particular problem in the present chapter, together with a discussion on its usability, its strengths and its weaknesses.

4.1 Introduction

4.1.1 Frequentist and Bayesian statistics

Both the LFQ and MSqRob quantitication engines presented in chapter 3 took a frequentist approach to the problem of protein quantification. Such approaches, establish a null hypothesis H_0, complementary to the actual hypothesis being tested (alternative hypothesis H_1). The information about the phenomena being modelled is put to use to measure how likely this null hypothesis is via Null-Hypothesis Significance Testing (NHST).

$$H_0 : \log_2(FC) = 0$$

$$H_1 : \log_2(FC) \neq 0$$

The results of this analysis will depend not only of the observed data, but also the data generation process [27]. This way, both tools returned point estimates of the log2FC between a pair of conditions and a p-value, which provides a measurement of the probability of the null hypothesis. The null hypothesis usually states that the true value of the parameter is 0. Thus, p-values do not say anything about the probability of the alternative (our) hypothesis.

The Bayesian statistical framework provides an alternative point of view by revolving the role played by the model parameters and the observed data. While frequentist statistics treats the data as random and the parameters as fixed, a Bayesian framework swaps their roles and yields a probability distribution for any model parameter given the provided data. Bayesian analyses

rely on the observed data and prior knowledge about the phenomenon only, and comes equipped with an equation that mathematically formalizes how to introduce these two dependencies in a valid way. This is the so called Bayes's theorem.

$$P(\theta|y) = \frac{P(y|\theta)P(\theta)}{P(y)} \tag{4.1}$$

The Bayes theorem is an extremely versatile tool taking a role analogous to that of statistics like T, F, or χ^2. Unlike them, which are tailored to specific scenarios, it can be used compute the probability distribution of almost any parameter in any model. 4 terms can be distinguished in its expression

1. $P(\theta)$: the **prior** probability distribution of the model parameter, introduces previous knowledge about the phenomenon being modelled.

2. $P(y|\theta)$: the probability distribution of the observed data (y) over the possible parameter space. It is also known as **likelihood** of the model, and its duty is updating our beliefs about the phenomenon by capturing the information in the data.

3. $P(y)$: the probability of the data, defined as the marginal probability of the data given a parameter value, for all possible values. It acts as a normalizing constant that makes the resulting distribution a true probability distribution adding up to 1. It is also known as the data **evidence**.

4. $P(\theta|y)$: the updated probability distribution of the model parameter, with the information extracted from the observed data. Since it reflects the beliefs about the phenomena after observing data, it is called

posterior probability distribution,as opposite to the prior.

The Bayes's theorem formulated above is thus read as *the posterir probability of the parameter θ is equal to the **prior** probability distribution times the likelihood divided by a constant.*

It can be applied to the quantification problem, where θ turns into the log2FC parameter we try to estimate. All is needed is the computation of the posterior probability distribution. Unfortunately, this is tractable analytically for simple models only. However, Markov Chain-Monte Carlo (MCMC) and Variational Inference (VI) methods can be used to approximate this posterior. Both of which are now possible to run thanks to the advent of modern computing.

4.1.2 Inference methods: MCMC and VI

The posterior distribution can be approximated (inferred) by means of MCMC and VI methods. On the one hand, MCMC methods simulate sampling from the posterior distribution by constructing an ergodic Markov chain on θ whose stationary distribution is the posterior p(θ | y) [**Blei2017**]. Monte Carlo sampling is performed on the Markov Chain, (hence the name of the technique) to randomly explore the parameter space with some heuristics. This heuristic guarantees convergence of the empirical estimate with the true posterior, provided a big enough sample size. Sampling convergence is defined as the status reached by the sampler when it estimates a distribution of probability that does not change anymore, regardless of how much longer the sampler runs [47] [1]. The first sampling algorithms, like Metropolis-

[1] http://www.cs.jhu.edu/ jason/tutorials/variational.html

Hastings [6] [2] [3] have given way to the much more efficient sampler NUTS [19].

On the other hand, VI methods, very recently developed and introduced in PyMC3, provide a fast alternative to MCMC methods, yet they are not guaranteed to asymptotically approximate the true posterior. In VI, optimization, instead of sampling, is employed as way to approximate the posterior [**Blei2017**]. More concretely, a family of densities Q over the parameters θ is defined. The goal of VI is to find the single density q within the family Q that best approximates $p(\theta|y)$. This approximate density should be complex enough to reach a good approximation, and at the same time simple enough to be computationally easy to work with. The best candidate is defined as the one minimising the Kullback-Leibler (KL) divergence (see equation 4.2).

$$q^*(\theta) = \operatorname{argmin} KL(q(\theta) \parallel p(\theta|x)) \tag{4.2}$$

where $q(\theta)$ stands for the whole family of densities, $p(\theta|x)$ stands for the true posterior and KL stands for the KL divergence. $q^*(\theta)$ is the best candidate density.

From [4] *the term variational is used because the best q in Q is picked. The term derives from the "calculus of variations," which deals with optimization problems that pick* [...] *a particular q in Q, specified by setting some*

[2] A very nice tutorial on how Metropolis-Hastings works http://twiecki.github.io/blog/2015/11/10/mcmc-sampling/

[3] Likewise, this resource provides very illustrative explanations of Hamiltonian-MC and NUTS http://elevanth.org/blog/2017/11/28/build-a-better-markov-chain/

[4] http://www.cs.jhu.edu/ jason/tutorials/variational.html

variational parameters i.e the knobs on Q that need to be turned to get a good approximation.

One of the terms yielded by the decomposition of the conditional density embodied by the posterior is the evidence of the data $p(x)$. $p(x)$ is computationally intractable, thus making the computation of the exact KL divergence impossible. A lower bound up to an additive constant can however be computed thanks to Jensen's inequality [21] This approximate amount is the Variational or Evidence Lower BOund (ELBO). *The ELBO is the negative KL divergence of plus logp(x), which is a constant with respect to q(z). Thus maximizing the ELBO is equivalent to minimizing the KL divergence* [**Blei2017**].

Finally, several density approximation families are available. The mean-field family of densities is one of the most simple, as it ignores all dependencies between latent variables, and treats them as independent. From http://www.cs.jhu.edu/ jason/tutorials/variational.html the mean-field approximations works by *pretending that the variables are just behaving that way "on their own." The mean-field method throws away all of the interactions.* A generic mean-field family member is shown in equation 4.3.

$$q(\theta_1, ..., \theta_m) = \prod_{j=1}^{m} q(\theta_j) \tag{4.3}$$

The demanding computational cost of the MCMC and VI schemes has only been recently met by the power of modern computers, thus making the approach feasible.

4.1.3 Model checking

However the posterior distribution is obtained, a proper Bayesian analysis is not finished without a model checking step that looks at potential problems happenning in the fitting step, and confirms the goodness of fit of the model to the data.

- Evidence for non-convergence must be collected. Albeit lack of evidence of non-convergence does not guarantee convergence, the existence of evidence definitely proofs it [27].
- Neither reaching convergence nor the best VI approximation guarantee that the fitted model actually captures the data generation process. Therefore, the true posterior could be a bad reflection of the natural process being model and the best approximation will not solve it. Posterior Predictive Checks are carried out to measure how well the model captured the data properties [27].

4.1.4 Probabilistic programming

The infrastructure providing the inference and model checking algorithms is available under several implementations in different programming languages, like R (JAGS [36]) and Python (PyMC3 [39]). All these frameworks make use of a programming paradigm called **probabilistic programming**.

The management of uncertainty in statistics is done by means of probability distributions, which account for the possible values that a parameter could take, together with how likely they are. Probabilistic programming offers the framework to build complex statistical models by storing probability distributions as variables of the program. Moreover, probabilistic

programming packages supply the tools needed to perform inference with these models by taking experimental evidence (data) and fitting the distributions to the data using Bayes'theorem. The fit of the model to the data entails the computation of the posterior probability distribution, which is tallied using algorithms provided with the probabilistic programming tool.

PyMC3 is a mature Python module dedicated to support probabilistic programming. It features intuitive model specification syntax, modern and powerful MCMC sampling algorithms like the No-U-Turn-Sampler (NUTS) as well as Automatic Differentiation Variational Inference (ADVI) [5]. Therefore, it provides the tools required to build a Bayesian model for the probabilisitc estimation of protein log2FC in proteomics datasets, that is, together with an estimate of its uncertainty.

4.1.5 Goals

1. Develop a Bayesian model to estimate log2FC in proteomics datasets together with the uncertainty of the estimate.

2. Make it scalable and fast for usability in ordinary modern computers.

4.2 Materials and Methods

4.2.1 Data input

The peptides.txt and proteinGroups.txt files produced by MaxQuant [8] after the analysis of the dataset published in [9] were used as input for the

[5] https://github.com/pymc-devs/pymc3

protein	Organism	H1	H2	H3	L1	L2	L3
P0A8I8	E. coli	25.13	25.24	24.39	21.71	22.67	22.40
P0A8I8	E. coli	21.49	23.10	23.38	21.34	22.65	21.25
P0A8I8	E. coli	24.10	24.54	23.81	19.88	20.87	20.30
A6NDG6	Homo sapiens	25.08	24.98	23.83	22.54	22.52	24.96
A6NDG6	Homo sapiens	22.70	24.32	23.00	22.29	23.27	23.91
A6NDG6	Homo sapiens	21.18	22.51	23.25	22.30	21.75	23.61

Table 4.1 Sample data input for the Bayesian model. Every row represents a unique peptide. The first column refers to the protein it was found to map to in the protein inference step. The second column is an annotation field, in this case indicating the protein's organism. The remaining columns indicate the $log_2(Intensity)$ registered for each peptide in the corresponding run. In this case, three peptides were observed for the proteins with ids *P0A8I8* and *A6NDG6*.

modelling algorithm when running without sequence modelling, and processed using the `preprocess_MaxQuant()` and `MSnSet2protdata()` functions in the MSqRob [17] package, similar to what was done in section 3.2.2.

Peptides missing in any of the samples available were dropped from the analysis.

The final state of the data is shown in table 4.1 for a single protein.

4.2.2 Sequence feature extraction

When running with active sequence modelling, the peptide sequences and their neighborhood in the protein was extracted from *E. coli* and *Homo sapiens* proteome databases (3.2.2). The protein neighborhood was defined by a window spanning 15 aminoacids on both sides of the peptide. It was extracted using the seqinr [5] and GenomicRanges [28] packages implemented in R and Bioconductor.

Features to model the peptide effect based on sequence were extracted us-

ing the Biopython [7] built-in module `ProtParam.ProteinAnalysis()`. The following properties were extracted: amino-acid percentage, peptide length, molecular weight, mass/length ratio, isoelectric point, aromaticity and instability.

4.2.3 Hierarchical modelling

The MS1 intensity measurements are affected by two main sources of variation:

1. The fixed effect that the researcher aims at unraveling with proteomics (treatment effect)

2. Random noise produced by experimental procedures, sequence bias, etc, which lead to a loss of data quality and resolving power.

A mass spectrometry-proteomics workflow aims at providing estimates of the treatment effect, upon which the biological interpretation of the results is done, while minimising the remaining unwanted effects. The mathematical framework implemented to model these effects was similar to the one used by MSqRob:

$$y_{ijkl} = \beta^0_{ijkl} + \beta^{treatment}_{ij} + \beta^{peptide}_{ik} + \beta^{run}_{il} + \epsilon_i \tag{4.4}$$

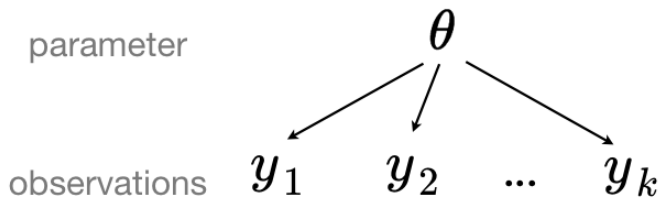
where y_{ijkl} stands for the $\log_2$-transformed intensity registered in treatment j, peptide k and run l. This way, separate treatment, peptide and run effects are distinguished. i stands for the protein index, which is evidently the same for all measurements belonging to peptides mapped to the same protein. ϵ models the noise that cannot be explained with the three effects

aforementioned, and is the same for all measurements associated to the same protein.

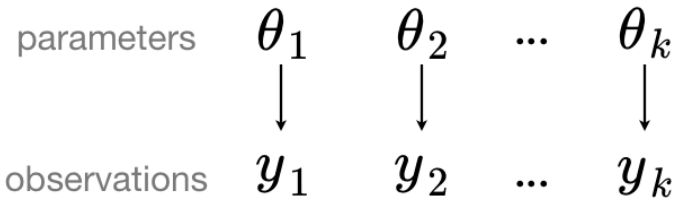
For example, the measurement $y_{P,A,p,1}$ corresponds to a measurement mapped to protein P. The measurement is impacted by a fixed effect, produced by treatment A, but also the peptide effect in peptide p and the run effect in run 1. An intercept term was also added to account for the starting intensity value.

The data is thus affected by different effects acting independently, and in order to model them properly, multilevels need to be defined. Multilevel effects can be modelled with three approaches: pooled, unpooled, and hierarchically (see figure 4.1). Hierarchical modeling is used when the sampling variance is not the only source of variation among the parameters, but they still share a dependency. The intercept and the three effects were modeled as independent hierarchies.

Pooled



Unpooled



Hierarchical

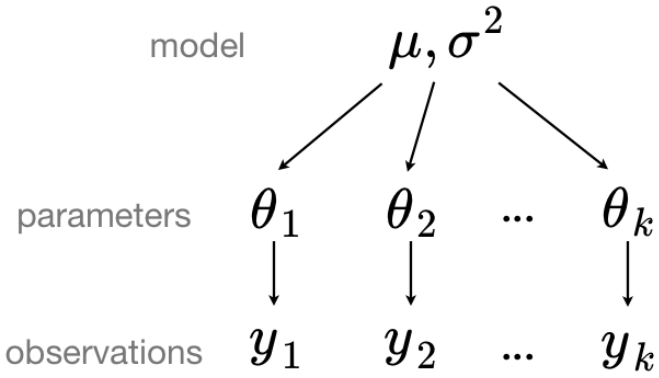


Figure 4.1 Alternative multilevel modelling approaches. A pooled model assumes that the parameter distribution is the same for all the datapoints. The unpooled model represents the opposite case, there every datapoint is assumed to be modelled by a different probability distribution of the parameter for each. Finally, a hierarchical model settles for a middle ground where the parameters will not be exactly the same but not completely different either. This is achieved by generating a global distribution from which the parameter for each datapoint is sampled [6].

This way, independent and identically distributed (IID) priors were set for

[6] https://docs.pymc.io/notebooks/multilevel_modeling.html

the intercept, treatment, peptide and run effects. The particular effect observed on each peptide was then modelled as a value sampled from the corresponding IID prior. A diagram of the resulting model is displayed in figure 4.2).

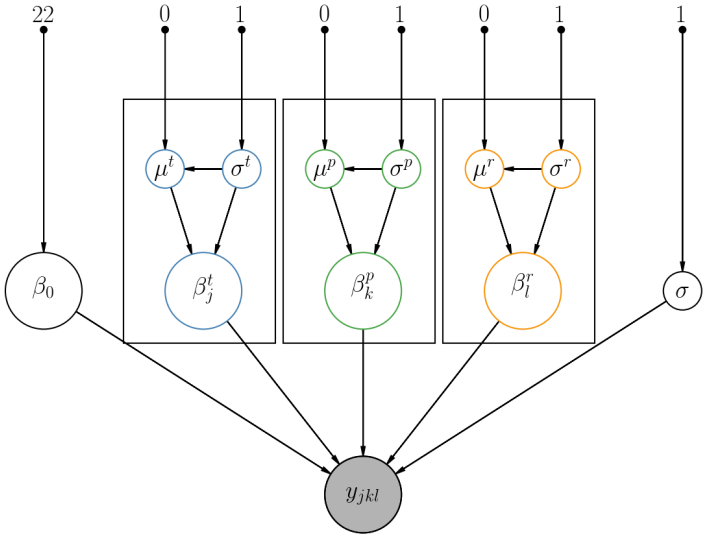


Figure 4.2 Diagram of the Bayesian model developed in the present work, without sequence modelling of the peptide effect. It is represented as a Directed Acyclic Graph (DAG) where nodes represent hyperparameters or random variables (in a circle) and directed edges represent the dependencies between them. The nodes on the top depict the hyperparameters of the model, governing the prior probability distributions represented by the nodes to which their edges lead to. The remaining nodes articulate the probabilistic model and all lead to a final node (y_{ijkl}) which represents the modelling of the observed data with the instantiated model. The model hierarchies are organized in boxes. The one corresponding to the treatment effect can be read as *the treatment effect β_{ij}^t observed in the data is modelled as a probability distribution conditional on the value of μ^t and σ^t, which are in turn probability distributions governed by the hyperparameters 0 and 1.*

4.2.4 Prior probability distribution specification

Both because of the little knowledge on the value of the parameters governing the data generation process and to provide a prior that everyone can

agree upon, non informative priors were provided to the model. This was condensed in the specification of Normal and Half Normal distributions for all the random stochastic variables in the model. The hyperparameters were selected to be as non informative as possible.

4.2.5 Posterior probability distribution computation

The VI mean-field strategy was applied to approximate the posterior probability distribution of the log2FC. ELBO optimisation was run for 40k iterations. Optimisation was validated by visual inspection of ELBO traceplots over the 40k iterations. Once optimisation was validated, 10k samples were drawn from the approximate distribution to simulate the true posterior.

4.2.6 Model checking

Posterior predictive checks of the posterior distribution were carried out using the `sample_ppc()` function in PyMC3. 500 datapoints for each peptide, consisting of 6 numbers, one for each run, were simulated. The resulting 6D data was projected onto a 2D plane via Principal Component Analysis. The covariance matrix required was computed on the whole dataset. The visual evaluation of the differences between simulated and observed data was used to assert the correctness of the models.

4.2.7 PyMC3 implementation

In order to get started with PyMC3, we first need to import it.

```
1 import pymc3 as pm
```

The model is initialized as a Python context manager. Within the context manager, the model is implemented by defining prior distributions for the model parameters and establishing the dependency relationships between them.

The code below formalizes in PyMC3 code the bias in MS1 intensity measurements due to a random effect.

```
with pm.Model() as model:
        sigma = pm.HalfNormal('sigma', 1)
        mu = pm.Normal('mu', mu=0, sd=sigma)
```

Which is equivalent to the following statistical notation:

$$\sigma \sim \mathcal{HN}(1)\,. \tag{4.5}$$

$$\mu \sim \mathcal{N}(0,\, \sigma)\,. \tag{4.6}$$

and can be read as *the prior probability distribution of the random effect follows a normal distribution with mean 0 and standard deviation σ. In turn, σ's prior probability distribution follows a half normal distribution with standard deviation 1.* 1 and 0 act as hyperparameters of the model, and introduce the state of beliefs or knowledge on the system prior to seeing the data.

The hierarchical structure of the model is set by the definition of a parameter distribution from which the value for each element (peptide) being modelled

is sampled. This can be done by defining a new normal distribution where its μ and σ are set to the random variables defined above. Im PyMC3 code,

```
5         betae = pm.Normal("betae", mu, sigma, n_elements)
```

which is equivalent to:

$$\beta_i^e \sim \mathcal{N}(\mu, \sigma)\,. \tag{4.7}$$

and can be read as *the probability distribution of the bias observed in the i^{th} element due to the effect here modelled is said to follow a normal distribution with mean μ and standard deviation σ both defined as random probability distributions above.*

Finally, the equation 4.4 defined above closes the model and binds the data to the model parameters. Its PyMC3 implementation is the following:

```
6         epsilon = pm.HalfNormal('epsilon', 1)
7         m = pm.Deterministic("m", beta)
8         obs = pm.Normal("obs", m, epsilon, observed=y)
```

which is equivalent to

$$
\begin{aligned}
\epsilon &\sim \mathcal{HN}(1)\,. \\
m &= \sum_{i=1} \beta_i\,. \\
y &\sim \mathcal{N}(m, \epsilon)
\end{aligned}
\tag{4.8}
$$

and is read as *the observed data is modelled by a normal distribution with mean m and standard deviation ϵ. m is a random deterministic variable resulting from the sum the effects defined above. ϵ follows a new half normal distribution with standard deviation 1.* A deterministic variable is a random variable that acquires a fixed value if all random variables it has a dependency on take a fixed value too, i.e its stochasticity disappears if its parameters are fixed.

4.3 Results

4.3.1 Running BayesQuant

BayesQuant takes a peptide_summary_intensity_moFF_run.tab (moFF) or a peptides.txt file (MaxQuant) as input, containing a peptide per row. Preprocessing using the MSqRob `preprocess_MSnSet()` function, as well as other data munging functions, is required.

```
Rscript prepare_BayesQuant.R --pepf peptides.txt \
    --filetype MaxQuant --exp exp_annotation.tsv \
    --output .
```

The R script outputs the input file for BayesQuant. Its data is introduced in BayesQuant via the following code:

```
# Load the module
from BayesQuant import BayesQuant
bayesquant = BayesQuant()
# Read the dataset
bayesquant.read_data(data="ms1_intensities.tsv")
# Compile a model for proteins with 3 peptides
model = bayesquant.compile_model(n_peptides=3)
# Load a specific protein
bayesquant.load_data("A6NDG61")
```

The snippet above loads the program, reads in the data, selects the data for a specific protein, and builds a model that matches the number of peptides observed. The posterior distribution can be computed using pure MCMC or VI methods, as stated in section 4.2.5. The VI procedure will now be explained, due to its significantly better performance.

```
# VI approximation: much faster and pretty accurate
trace_advi = bayesquant.fit(model_name=p, n_draws=40000)
```

The `bayesquant.fit()` function carries out two tasks:

- Finds the best mean-field approximation $q(\theta)$ to the true posterior $p(\theta|x)$. This approximate distribution is easier to work with.
- Sample from the approximate posterior, returning a collection of sam-

ples called trace.

The values stored in the trace obtained via VI will follow an approximation to the true posterior, and contain the results of the modelling process. Once it is obtained, the inference is complete and model checking ought to be performed to validate the results as explained in section 4.1.3. It is run with:

```
12 # Run Posterior Predictive Checks
13 bayesquant.ppc()
```

A step by step exposition of the plots and results produced by the program when ran on the proteins P0A818 (*E. coli*) and A6NDG6 (*Homo sapiens*) will follow.

4.3.2 VI optimisation evaluation

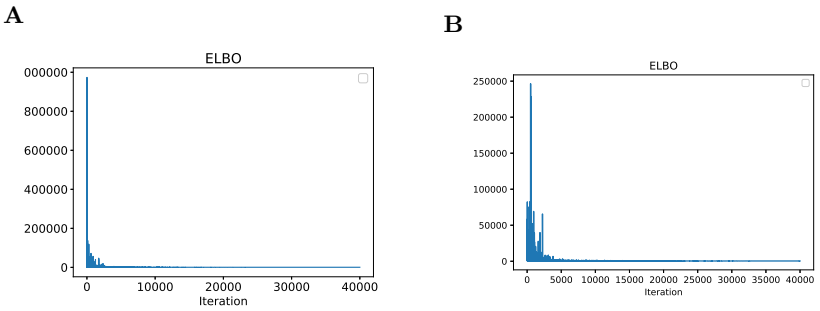


Figure 4.3 Progression of ELBO maximisation during the VI approximation for proteins P0A818 (*E. coli*, **A**) and A6NDG6 (*Homo sapiens*, **B**) over 40k iterations.

After the `.fit()` method is run, a plot of the maximisation of the ELBO is produced automatically. Even though the ELBO took extremely low

(negative) values at start (see figure 4.3), it was maximised fast to values around 0, indicating that the mean-field approximation was acceptably good.

4.3.3 Sampling from the approximation to the posterior

The 10k traces produced by BayesQuant from the VI approximation function exhibited a clear random walk behaviour, which decreases the posibility that the approximation was wrong. Moreover, the effects caused by the different peptides and runs were estimated and separated from the overall observed noise, allowing for a better estimation of the log2FC (see figure 4.4).

This way, a different posterior distribution was computed for the impact of each "peptide" and "run" in all the measurements collected for each protein.

The study of the log2FC posterior probability distribution (see figure 4.5) shows the most likely parameter values and empowers Bayesian Null Hypothesis Testing. The 95% High Probability Density Interval (HPDI), which is the narrowest interval containing 95% of the total probability, indicates the most likely true values of the parameter. Presence or absence of overlap between the 95% HPDI and the Region Of Practical Equivalence (ROPE) provides a formal way of discarding a point value of the parameter. The ROPE is a small interval considered to be essentually the same as the null value [27]. In this and all future applications in the thesis, the null value is set to 0, and the interval is given a width of 0.4 on each side.

The analysis indicated that P0Q8I8 (*E. coli*) most likely has a log2FC between 1.4 and 1.7, and it'significantly different from 0, as no overlap was observed between HPDI and ROPE. On the other hand, the human protein was found to acquire a log2FC most likely between -0.1 and 0.192. This

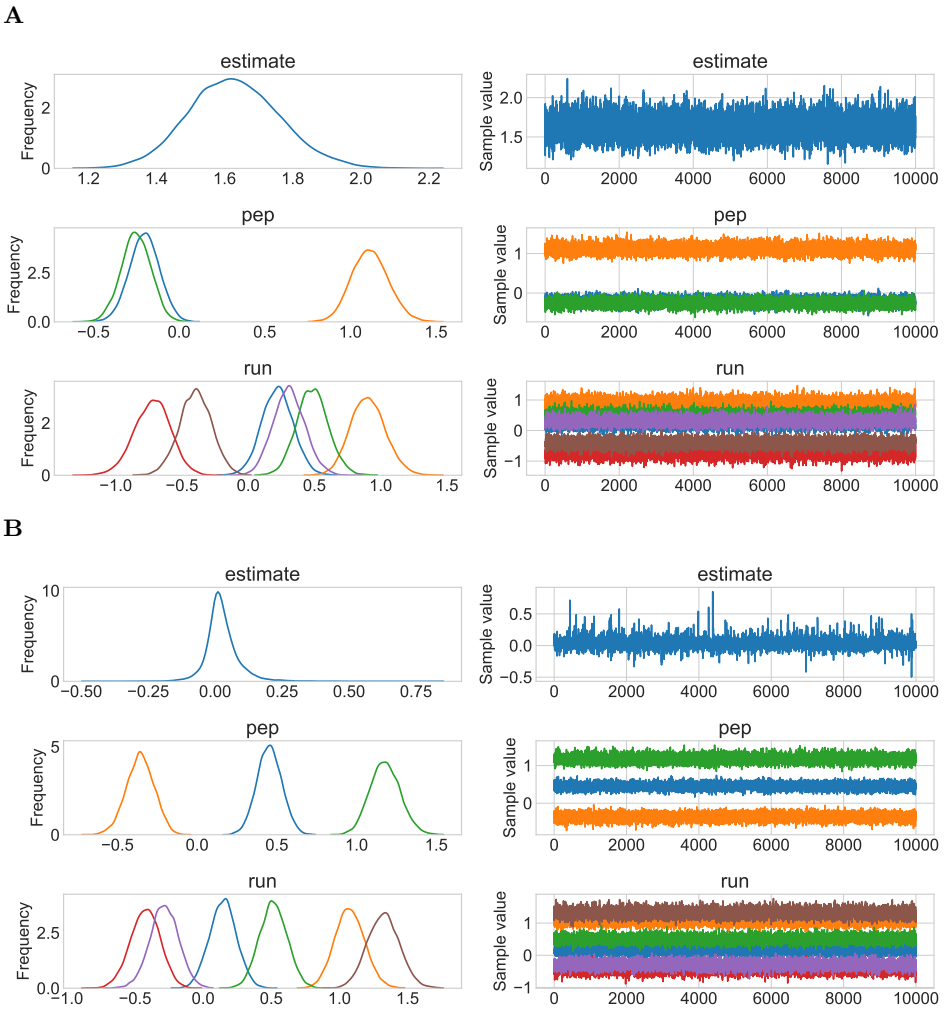


Figure 4.4 Traceplot of the model fit for proteins P0A818 (*E. coli*, **A**) and A6NDG6 (*Homo sapiens*, **B**). The left panel shows the frequency density over the parameter space for several model parameters: (I) the log2FC estimate (difference of treatment effects), (II) the effects in the three observed peptides, and (III) the effects in the six available runs. The right panel displays the sampled values from the VI approximation stored in the trace.

range is fully contained within the ROPE, which indicates that the log2FC was practically 0 (see figure 4.5).

The posterior predictive checks on both proteins pinpoints that the fitted models captured a data generation process approximating what was observed

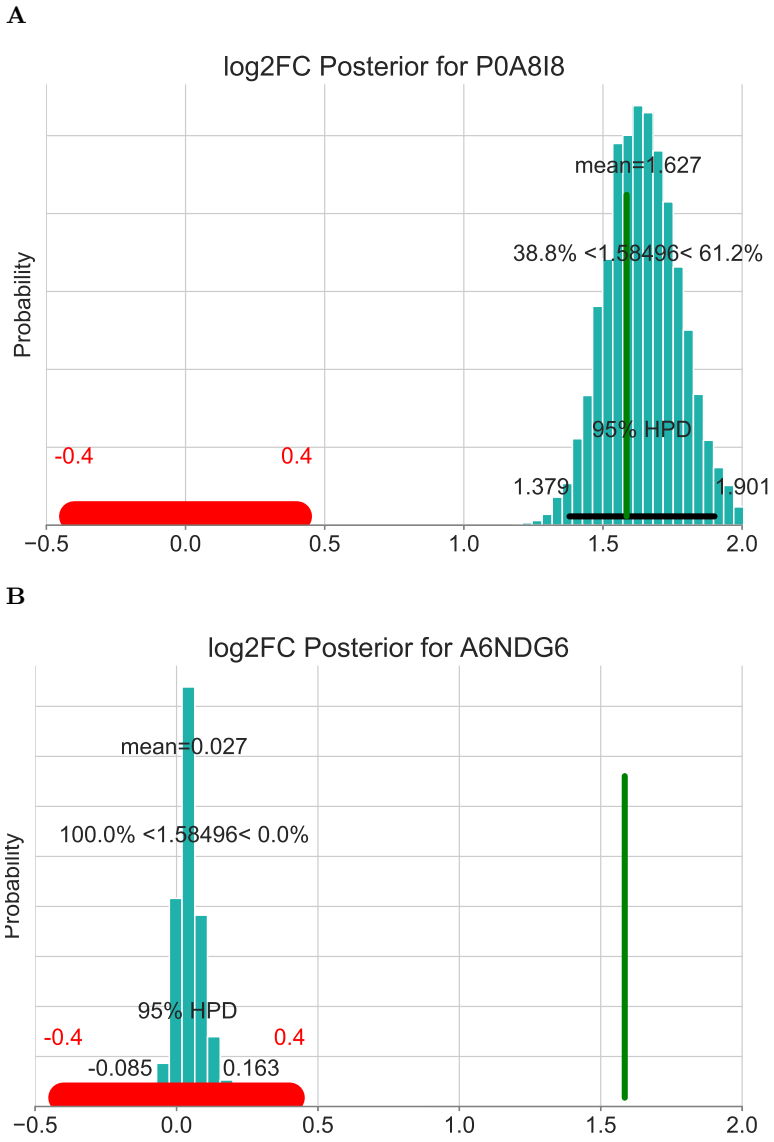


Figure 4.5 Annotated posterior probability distribution for the log2FC estimate for proteins P0A818 (*E. coli*, **A**) and A6NDG6 (*Homo sapiens*, **B**). The bar height is mapped to the probability mass in the corresponding interval. The 95% HPDI is marked with a black line. The ROPE defined as a 0.4 window around 0 is shown in red. A green vertical line marks the expected log2FC estimate for *E. coli* proteins ($\log_2(3)$).

(see figure 4.6). The three observed peptides were found to be undistinguishable from the 500 simulated ones.

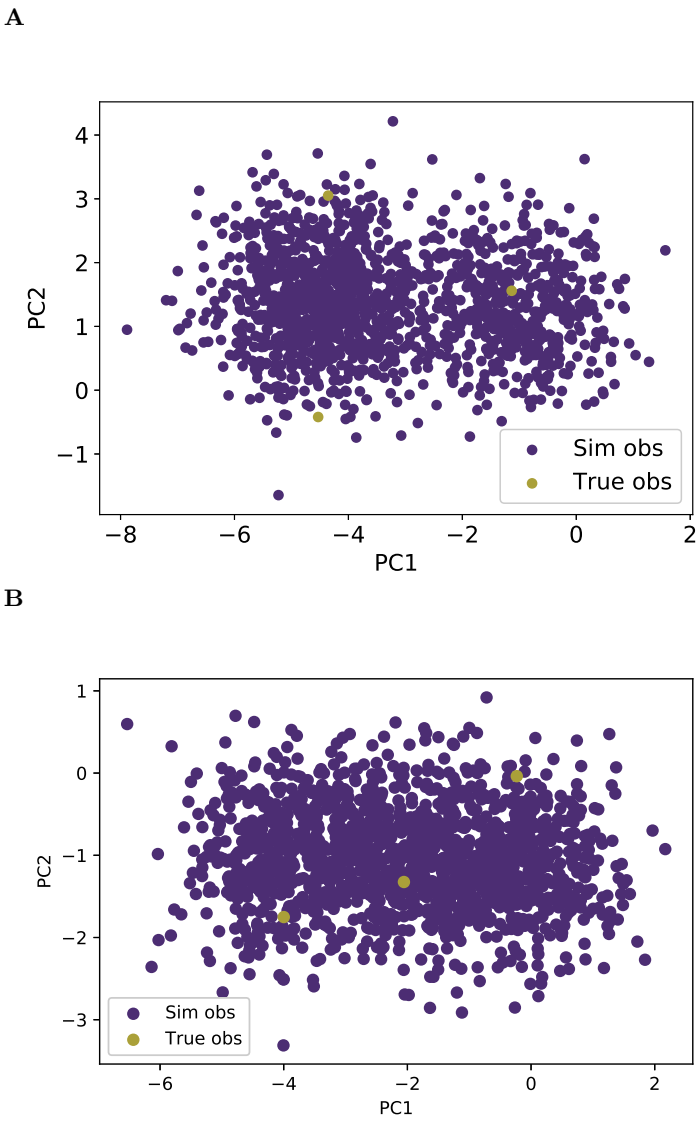


Figure 4.6 Projection of the posterior predictive checks on the 2D plane capturing most variance. Simulated datapoints are shown in blue, whereas actual observations are shown in red.

In order to further validate BayesQuant's performance, it was tested on groups of 5 proteins, one group from each proteome. Each pair of groups was made by proteins with a different number of observed peptides (2, 3, 4, 6, 7, 10) (see figure 4.7). The more peptides, the more accurate the

estimation will be as more data will be available. Since the HPDI reflects the uncertainty, a decrease on its width was expected to be observed (see figure 4.7).

Indeed, a general trend of width reduction was observed with the increased amount of peptides. For example, the widest intervals are observed on proteins with just two peptides available, and the narrowest on proteins with ten. Moreover, the HPDI tended to align more strongly around an agreement value as well.

The consistent observation that *Homo sapiens* proteins got a log2FC estimate around 0, and *E. coli* proteins got an estimate which overlapped the predefinned ROPE in only three cases (two of which on proteins with two peptides), confirmed the validity of the quantification framework.

4.3.4 Extended model: peptide effect with sequence features

Given the open-source nature of the program, it is possible to build upon it and extend its functionality. The possibility of making use of the peptide sequences to model the peptide effect using a linear model was considered and put to practice. The code was thus extended with the following snippets of code:

```
bayesquant.read_data(
    # table with ms1 intensity data
    data_path="data/ms1_intensities.tsv",
    # table with sequence features
    features_path="data/features.tsv")
```

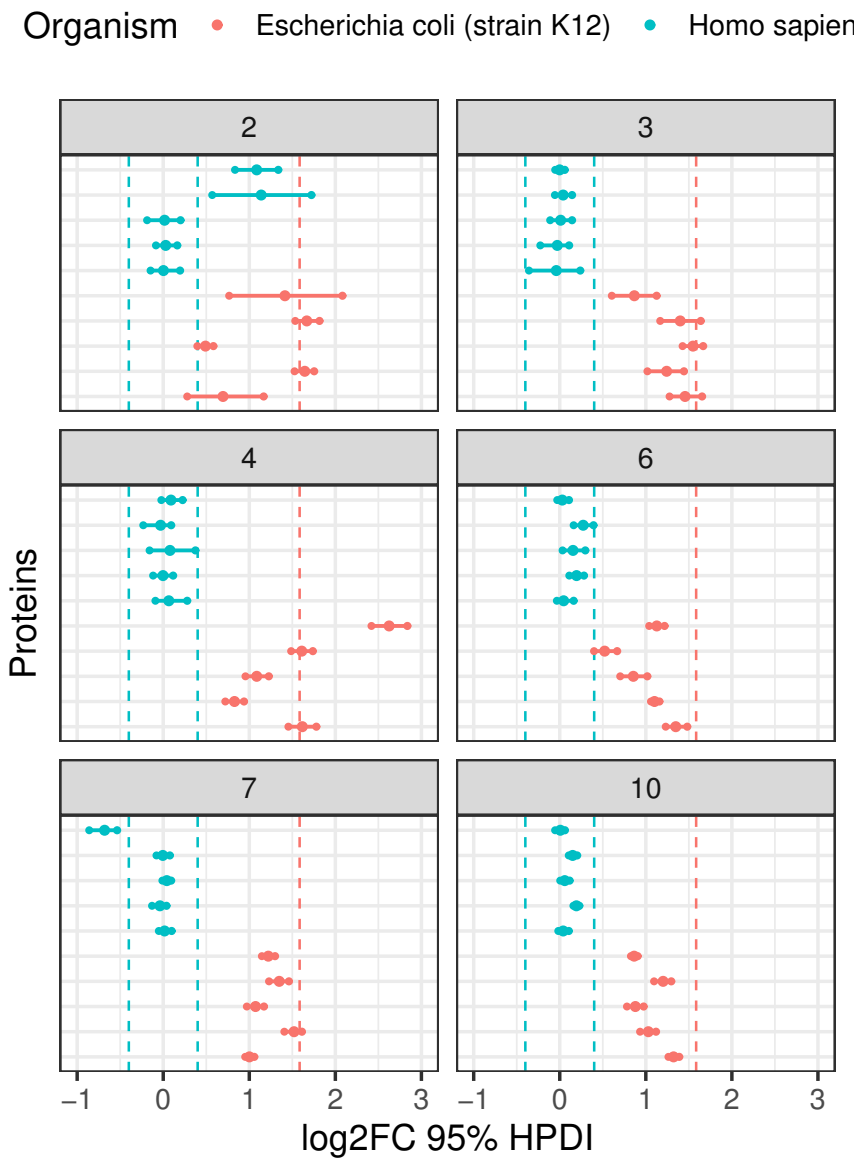


Figure 4.7 Visualization of the 95% HPDI inferred from 5 bacterial and 5 human proteins with 2, 3, 4, 6, 7 and 10 peptides. The intervals are represented by horizontal lines. The dot represents the mean of the whole distribution, and it will be centered in the interval if the distribution is symmetrical. Vertical blue lines represent the ROPE defined as in figure 4.5, whereas the red line represents the expected estimate for bacterial proteins.

```
# Specification of priors
sigma_theta = pm.HalfNormal('sigma_theta', sd=1)
theta = pm.Normal('theta', mu=0, sd=sigma_theta, shape = (n_features, 1))
theta_inter = pm.Normal('theta_inter', mu=0, sd=sigma_theta)
mu_pep = pm.Normal("mu_pep",
  mu=theta_inter + pm.math.sum(features.dot(theta)),
  sd=sigma_pep)
```

which means that if the peptide effect is modelled as a function of the sequence, μ^p is redefined to:

$$\sigma^\theta \sim \mathcal{HN}(1)$$

$$\theta_k \sim \mathcal{N}(0, \sigma^\theta)$$

$$\mu^p \sim \mathcal{N}(\sum_{k=0}^{k=K} x_k \theta_k, \sigma^p)$$

where θ_k is the weight of the k^{th} feature, and x_k is the numerical value of the k^{th} feature. σ^θ is the prior for the standard deviation of the Normal all feature weights are sampled from.

The same workflow as in section 4.3.3 was ran with the extended model, and the result is shown in figure 4.8. The analysis illustrates the null predictive power of the sequence properties extracted, as all converge extremely strongly to a value of 0. The result is equivalent to not having defined any weights at all, and instead having run the basic model.

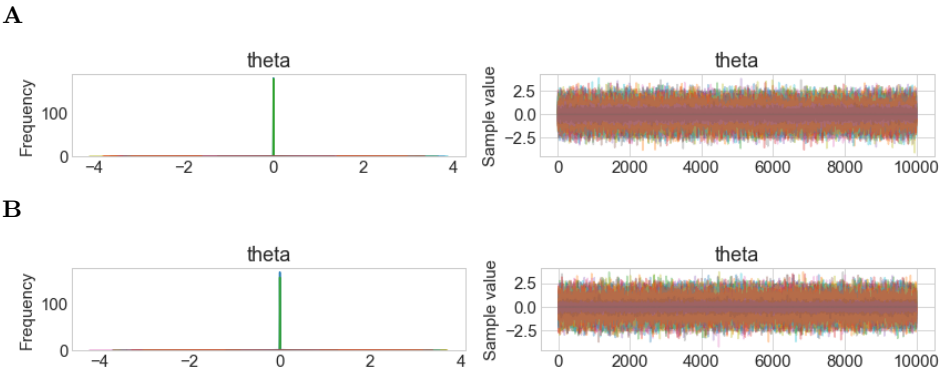


Figure 4.8 Traceplot obtained after fitting the extended model on the test proteins mentioned in 4.3.3. No feature was found to have any predictive power.

4.4 Discussion

4.4.1 Improving usability: parallelization

While BayesQuant performed well on the majority of proteins, it took around 30 seconds to approximate each protein'posterior distributions using VI, and more than two minutes with MCMC methods. This translates to a dataset of 1k proteins taking 500 minutes (>8h) to be processed. However, the average modern computer comes with several processors that could support the parallelization of the program in several threads, decreasing the total computing time.

4.4.2 Further robustness and validation checks

A solid way of checking the results are not flawed is the specification of alternative priors. Unless precise prior knowledge is available, the selection of one prior over another prior to model the beliefs about the system should

not have any impact on the results. It is for that reason that models built upon different priors could be run to further validate the program.

4.4.3 Advanced model comparison

Besides the predefinition of a ROPE and the check of overlap with the 95 % HPDI, more advanced model comparison methods making use of the Bayes factor are available. The Bayes factor is defined as the ratio of the posterior probability of the models given the data [27]. It could be used to measure how much more likely a model taking into account a treatment effect is compared to another one where the treatment effect is considered null i.e the log2FC is set to 0. The result would provide an alternative criteria for the declaration of a protein being differentially abundant.

4.4.4 Sequence-based modelling of the peptide effect

Unfortunately, during the development of the project, it was acknowledged that the peptide effect is a very difficult phenomenon to capture in a simple Bayesian model as the one presented in this work. This is due to the motley and and multilevel nature of the noise caused by the aminoacidic sequences:

- The protein neighborhood, which might affect the protease's efficiency.
- The competitive ionization problem, introduced in chapter 2 (2.2.1) implies that ionization prediction cannot be achieved without models that do not replicate this process mathematically. Thus, the ionization of a peptide ought to be predicted taking into account the peptide mix with which it accessed the ion source as a whole.

- The mobile proton model [3], which states that the peptide's fragmentation pattern changes drastically depending on the number of charges and its distribution on the sequence. The charge distribution is hypothesised to be in turn determined by the position of positive residues that can allocate the charges. This phenomenon implies that the PSM step is more difficult as a different spectrum is expected for molecules with the same peptide sequence but different charges. The decreased identifications and quality of the peaks makes feature extraction algorithms like the Apex module in moFF would extract worse MS1 intensities, in turn injecting noise in the quantification data.

- The characterization of powerful sequence properties that specifically capture the differential ionization efficiency is undone work, as most feature extractors are designed to solve different problems, like protein sructure prediction.

Modelling such a complex phenomenon will require accordingly complex predictive architectures and the introduction of proper neural networks, capable of deciphering the intrincate pattern mapping sequence to noise in the spectrometer's measurements.

4.4.5 Posterior assessment of the effects

The characterization of the uncertainty behind a log2FC estimate can be used to better inform downstream analysis programs and eventually help interpret the biological results. For example, a functional analysis program could decide to drop proteins declared differentially abundant if the uncertainty is greater than a threshold. A Gene Set Enrichment Analysis (GSEA) could make use of the probabilistic information to refine the results of the

enrichment.

Moreover, the estimation of the different run effects could be used to help MS technicians assess the presence of batch effects in their experiments, and correct them in future analyses.

4.5 Conclusion

A Bayesian framework for the relative quantification of protein abundance rations (Fold-Change, FC) using MS1 intensity was presented in this work. Unlike the currently prevailing methods in the field, it provides uncertainty estimates that provide a direct interpretation about the accuracy of the quantification. Moreover, preliminary results on the sequence modelling of the peptide bias revealed that more complex architectures and sequnce features are required to solve this problem. Finally, the method can be used to investigate the presence of batch effects in MS experiments for their minimisation in future. While the software can be integrated in any pipeline producing peptide-level data, further testing and improvements are probably required for optimal performance.

Chapter 5

Pipeline benchmarking on NZ data

Summary

A benchmark dataset was run through the programmes presented in chapters 3 and 4 to showcase their performance. The experimental desgin attempted to capture a protein profile change reflecting a known biological process. Thus, a successful computational analysis of the dataset should also reflect the phenomenon. The results confirmed that the computational analysis successfully captured this response. It was concluded that the software can be used in future experiments where the biological phenomena underlying the data are not known.

5.1 Introduction

The protein ratios across conditions are not known in ordinary datasets In order to judge the resolving power of the mass spectrometry and shotgun

proteomics workflows presented in this work, an in-house dataset attempting to capture an immunological response was generated. Therefore, even though true protein ratios were not known *a priori*, the analysis should find proteins related to the immunological response to be significantly differentially abundant.

Moreover, a function inference step from the estimated protein log2FC values is required to arrive to biological interpretations like that. These kind of analysis is common to all omics (genomics, transcriptomics, ...) and can be completed with Gene Set Enrichment Analysis GSEA and Pathway Analysis, among others.

5.1.1 Goals

1. Gauge the analytical power of the computational analyses explained in the previous chapters.
2. Propose a function inference analytical pipeline to complete the proteomics workflow.

5.2 Materials and Methods

5.2.1 Sample preparation

THP-1 cells were grown in RPMI1640 (Sigma) supplemented with 10% FBS (SeraLab) and 1% pen/strep (Gibco). Stimulation of THP-1 cells was done in 48-well plates (Gibco) with 2x106 cells/well. Cells were exposed to 500 ng/mL LPS (E. coli O111:B4) (positive control condition, *PC*) or nothing (negative control condition, *NC*) for 48 hours in 37 degrees, 5% CO_2. After

stimulation supernatants were collected and cell pellets were washed in PBS (Gibco). Both supernatants and pellets were frozen (-20 degrees). Only the pellet samples were considered for the analysis. EGWS, Esben Gjerløff Wedebye Schmidt.

5.2.2 Mass Spectrometry analysis

AEGI

5.2.3 Computational analysis

The resulting .RAW files were processed with the two different pipelines fully developed in the present work. (I) Compomics+MSqRob and (II) Compomics+BQ (Compomics and BayesQuant). The search settings were identical to those used in chapter 3, except for the proteome databases employed, which consisted of *Homo sapiens* only. The validation filters, MBR, MS1 Apex intensity, and MSqRob preprocessing parameters were also set to those used previously, for simplicity. The two pipelines used thus diverged only in the quantification engine, being identical in all previous steps.

In the Compomics+MSqRob pipeline, a protein was declared to be differentially abundant (DAP) between the 2 conditions assayed if the result of the student's-T test implemented in MSqRob returned a q-value less than 0.05. The criteria in the Compomics+BayesQuant pipeline was the absence of overlap between the ROPE [-0.4, 0.4] and the 95% HPDI.

5.2.4 Biological inference

The gProfiler R package [3] was used to run automatic GSEA and provide a biological interpretation of the results.

Pathways analysis was executed by calling the ComPath Webserver [13]. UniprotKB IDs were transformed to "Gene name" IDs, as required by the tool using the Retrieve/ID mapping tool from Uniprot. The KEGG [23] and Reactome [10] databases were used as source.

Protein interactions were explored by querying the STRING database [45] with the list of DAPs produced by MSqRob.

5.3 Results

5.3.1 Compomics+MSqRob

Data on 6444 peptides, as collected in the peptide summary file from moFF, was passed to MSqRob, resulting in the quantification of 1433 proteins. A histogram of the log2FC reveals that in most cases, not enough evidence was collected to declare the log2FC different from 0 (see figure 5.1 A). Remarkably, the distribution was shifted toward negative values, depicting that most of the proteins for which an non-null estimate was provided were more abundant in the *PC* condition. A volcano plot (see figure 5.1 B), reveals that the majority of the 49 protein groups passing the significance criteria ($qval < 0.05$), displayed an absolute value of the log2FC less than 1. However, all were considered differentially abundant. The 32 protein groups composed by a single protein (single protein DAPs) are shown in table 5.1.

	Protein	Name	log2FC	qval
1	P02792	Ferritin light chain	9.03E-01	2.32E-04
2	Q04760	Lactoylglutathione lyase	-7.24E-01	2.32E-04
3	P13796	Plastin-2	-6.54E-01	2.32E-04
4	P28066	Proteasome subunit alpha type-5	-6.88E-01	8.24E-04
5	P09874	Poly [ADP-ribose] polymerase 1	-3.68E-01	2.48E-03
6	P53396	ATP-citrate synthase	-3.79E-01	3.25E-03
7	P50552	Vasodilator-stimulated phosphoprotein	-6.95E-01	1.43E-02
8	P54577	Tyrosine–tRNA ligase, cytoplasmic	-4.21E-01	2.71E-02
9	Q15393	Splicing factor 3B subunit 3	-1.18E+00	2.88E-02
10	P08567	Pleckstrin	-1.26E+00	2.96E-02
11	P50990	T-complex protein 1 subunit theta	-3.60E-01	2.96E-02
12	Q7RTV0	PHD finger-like domain-containing protein 5A	-1.26E+00	2.96E-02
13	O75083	WD repeat-containing protein 1	-6.55E-01	3.12E-02
14	P17812	CTP synthase 1	-4.60E-01	3.12E-02
15	Q14498	RNA-binding protein 39	-1.50E+00	3.29E-02
16	P60903	Protein S100-A10	1.13E+00	3.29E-02
17	P26038	Moesin	-3.91E-01	3.37E-02
18	O75368	SH3 domain-binding glutamic acid-rich-like p	-6.15E-01	3.37E-02
19	Q8TEM1	Nuclear pore membrane glycoprotein 210	-8.24E-01	3.54E-02
20	Q01518	Adenylyl cyclase-associated protein 1	-3.51E-01	3.54E-02
21	Q08211	ATP-dependent RNA helicase A	-4.19E-01	3.54E-02
22	Q14566	DNA replication licensing factor MCM6	-4.33E-01	3.68E-02
23	P30086	Phosphatidylethanolamine-binding protein 1	-6.31E-01	3.68E-02
24	O43143	RNA helicase DHX15	-1.30E+00	3.80E-02
25	O75691	SMPC 20 homolog	-1.26E+00	3.80E-02
26	P27797	Calreticulin	-4.63E-01	3.80E-02
27	P41218	Myeloid cell nuclear differentiation antigen	-4.49E-01	3.80E-02
28	Q15056	Eukaryotic translation initiation factor 4H	-3.62E+00	3.98E-02
29	P04406	Glyceraldehyde-3-phosphate dehydrogenase	-1.90E+00	3.98E-02
30	P00492	Hypoxanthine-guanine phosphoribosyltransferase	-4.54E-01	4.12E-02
31	P50440	Glycine amidinotransferase, mitochondrial	-1.02E+00	4.50E-02
32	P62857	40S ribosomal protein S28	-5.50E-01	4.98E-02

Table 5.1 THP-1 MSqRob results. 32 single-protein groups were found to be differentially abundant under significance criteria of q-value less than 0.05. The protein id, the gene name, the point estimate of its log2FC and its significance is shown

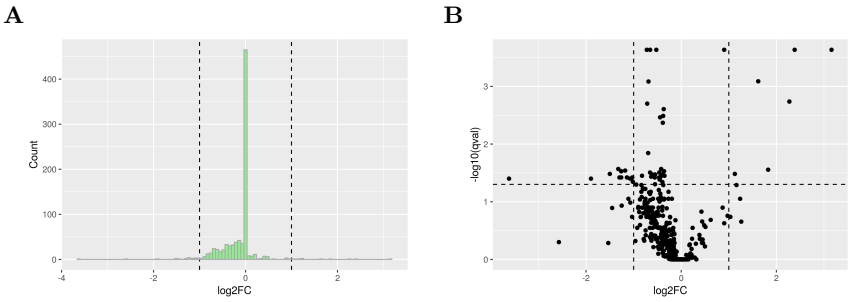


Figure 5.1 Compomics+MSqRob results. **A** Histogram of the estimated log2FC. **B** Volcano plot, showing the point estimate of the log2FC on the x axis, and the minus logarithm of the p-value on the y-axis.

5.3.2 Compomics+BayesQuant

After preprocessing and missing-data handling, a dataset of 1641 peptides was input to BayesQuant, and 269 proteins were quantified. A histogram of the mean of the inferred posterior distributions manifests the centrality of its distribution, with most values within the predefined ROPE of [-0.4, 0.4]. In line with what was observed in the results above, the distribution was found to be skewed toward negative values (see figure 5.2A). Only four proteins were assigned a 95% HPDI not touching the ROPE (see figure 5.2B and table 5.2). The distribution of the HPDI width summarises the overall uncertainty in the estimation process. Most proteins had a very narrow HPDI, as shown in figure 5.2C. As expected, the width of the interval exhibited some degree of correlation with the mean of the posterior, as shown in 5.2D.

Protein	log2FC	HPDI start	HPDI end	Peptides
O75475	-0.61	-0.80	-0.43	2
O15144	-0.75	-0.98	-0.51	3
P02792	0.50	0.41	0.58	4
P04406	-1.09	-1.45	-0.77	4

Table 5.2 DAP declared by the Compomics+BayesQuant pipeline.

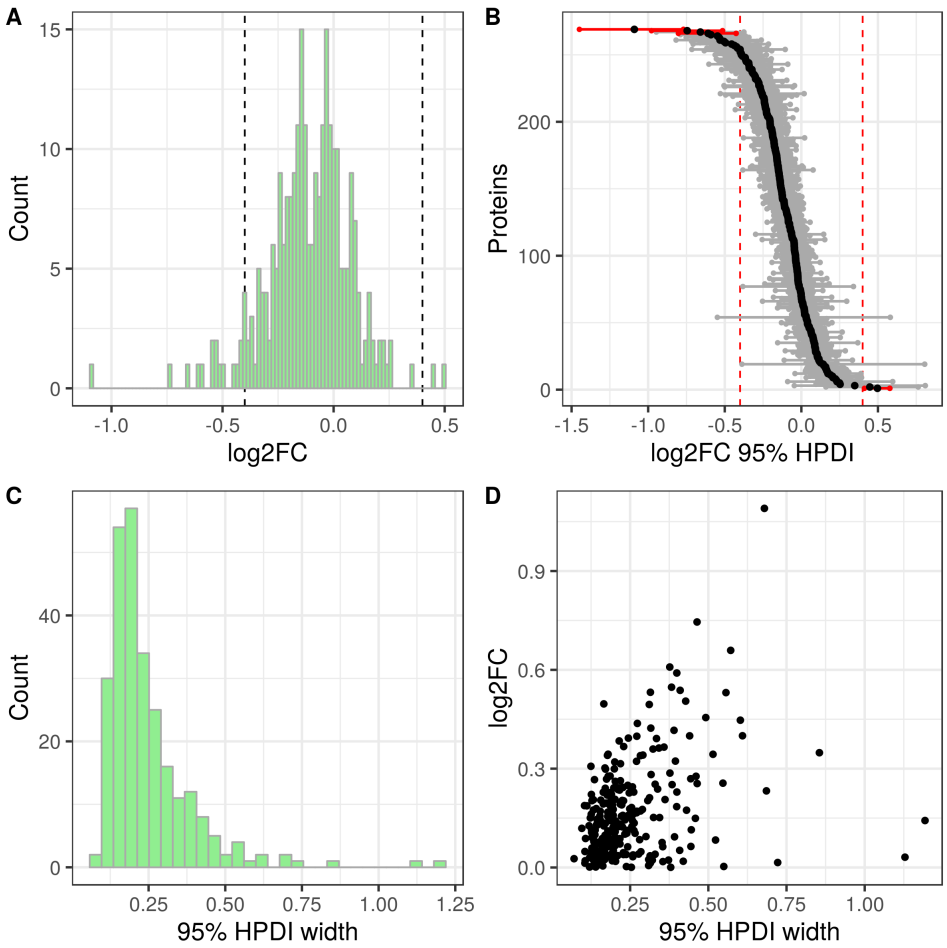


Figure 5.2 BayesQuant results. **A** Histogram of the mean of the posterior distributions. **B** Dumbbell plot, representing the 95% HPDI for all proteins. Black dots represent the mean, and intervals not overlapping the ROPE are shown red. **C** Histogram of the 95% HPDI width, which can be used as proxy for the overall uncertainty in the quantified data. **D** Scatter plot of the 95% HPDI width and the posterior log2FC mean.

Given the scarcity of the results, it was decided to continue the analyses with MSqRob output only.

5.3.3 Functional analysis

A GSEA performed on the list of 32 single-protein DAPs produced by MSqRob (see table 5.1) illustrates the most overrepresented biological processes, cellular compartments and molecular functions, compared to the human background. The 20 most significant terms found in the dataset are displayed in table 5.3.

	Term ID	Term Name	Query	Term	Ov	P-value
1	GO:0070062	extracellular exosome	32	2774	19	1.17e-05
2	GO:1903561	extracellular vesicle	32	2793	19	1.31e-05
3	GO:0043230	extracellular organelle	32	2795	19	1.33e-05
4	GO:0005615	extracellular space	32	3958	20	6.76e-04
5	GO:0035578	azurophil granule lumen	32	90	5	8.87e-04
6	GO:0044421	extracellular region part	32	4119	20	1.34e-03
7	GO:0002376	immune system process	32	2949	17	1.74e-03
8	GO:0060205	cytoplasmic vesicle lumen	32	335	7	2.54e-03
9	GO:0031983	vesicle lumen	32	336	7	2.59e-03
10	GO:0005576	extracellular region	32	4925	21	4.95e-03
11	GO:0003723	RNA binding	32	1787	13	5.50e-03
12	GO:0065003	protein-containing complex assembly	32	1803	13	6.08e-03
13	GO:0003725	double-stranded RNA binding	32	65	4	9.71e-03
14	GO:0005766	primary lysosome	32	153	5	1.23e-02
15	GO:0042582	azurophil granule	32	153	5	1.23e-02
16	HPA:006020_13	caudate; neuronal cells	14	311	5	1.35e-02
17	GO:0031982	vesicle	32	4271	19	1.38e-02
18	GO:1990904	ribonucleoprotein complex	32	836	9	1.44e-02
19	GO:0001775	cell activation	32	1359	11	1.60e-02
20	HPA:022020_13	hippocampus; neuronal cells	14	331	5	1.81e-02

Table 5.3 GSEA results. The 20 most significant terms are displayed. The columns *Query*, *Term* and *Ov* indicate the size of the query, the number of proteins under the term, and the overlap between them. The size of the query differs in the GO and HPA terms repositories due to nomenclature mismatches.

On the one hand, nine terms (1, 2, 3, 4, 6, 8, 9, 10 and 17) were cell compartment terms associated with cellular secretion and the extracelullar space. On the other hand, five terms (5, 7, 14, 15, 19) were related to

the immunological response. The remaining terms were connected to RNA functions and the nervous system.

5.3.4 Pathway analysis

The UniprotKB to Gene name mapping of the set of 49 DAPs (including multiprotein groups) returned by MSqRob returned a list of 51 entities. The list was supplied to the ComPath Pathway Enrichment tool to highlight which cellular pathways were overrepresented compared to the human background. 50 entities were mapped to the database (only *C12orf42* was unmapped). In agreement with the GSEA results, a handful of pathways linked to the immune response were found to be significantly enriched (see table 5.4) (I) Immune System, (II) Innate Immune System, (III) Necroptosis (IV) Programmed Cell Death, (V) Regulation of Apoptosis, (VI) Leukocyte transedothelial migration. Moreover, the finding of several RNA related pathways exposes the presence of potential cell reprogramming, essential in any change in protein profile.

5.3.5 Protein interaction analysis

The same list supplied to ComPath was searched in the STRING database [45] to screen for protein-protein interactions in the list. The database returns a protein association network (see figure 5.3), that helps interpreting the information contained in gene sets. 49/51 terms were mapped to the resource and 15 of them were found to be included in the "immune system process" set, placing this biological process as the third most enriched with an FDR of 0.0387. 27 genes belonged to the "cellular nitrogen compound metabolic process" (FDR 0.0128).

	Pathway	q-val	Ov	Size	DB
1	Neutrophil degranulation	0.00e+00	9	479	reactome
2	Immune System	0.00e+00	15	2050	reactome
3	Innate Immune System	0.00e+00	10	1109	reactome
4	Metabolism of RNA	0.00e+00	10	666	reactome
5	Processing of Pre-mRNA	2.00e-04	5	240	reactome
6	mRNA Splicing	1.10e-03	4	180	reactome
7	mRNA Splicing	1.20e-03	4	188	reactome
8	Metabolic pathways	1.60e-03	8	1282	kegg
9	Carbon metabolism	2.90e-03	3	116	kegg
10	Spliceosome	4.00e-03	3	134	kegg
11	Pentose phosphate pathway	4.20e-03	2	30	kegg
12	Base excision repair	4.90e-03	2	33	kegg
13	Ribosome	5.30e-03	3	153	kegg
14	Necroptosis	6.10e-03	3	162	kegg
15	Apoptosis	6.60e-03	3	168	reactome
16	Programmed Cell Death	6.90e-03	3	171	reactome
17	Regulation of Apoptosis	1.08e-02	2	53	reactome
18	Aminoacyl-tRNA biosynthesis	1.55e-02	2	66	kegg
19	Glycolysis / Gluconeogenesis	1.59e-02	2	68	kegg
20	Leukocyte transendothelial migration	2.96e-02	2	112	kegg

Table 5.4 Pathway Enrichment results. 50 proteins were mapped to pathway databses and an enrichment was performed. The 10 most significant pathways found in Reactome and Kegg are shown arranged by decreasing significance. The columns *Ov* and *Size* indicate the number of proteins shared between pathway and query, and the number of proteins included in the pathway, respectively. The input was always 49 set to protein groups.

5.4 Discussion

A discussion over the major issues and strenghts detected while obtaining these results follows.

5.4.1 Increasing the experiment data-throughput

Six samples organized in two different treatments of three biological replicates without sample prefractionation each were analyzed. The resulting data throughput allowed for the performance of biological inference, as

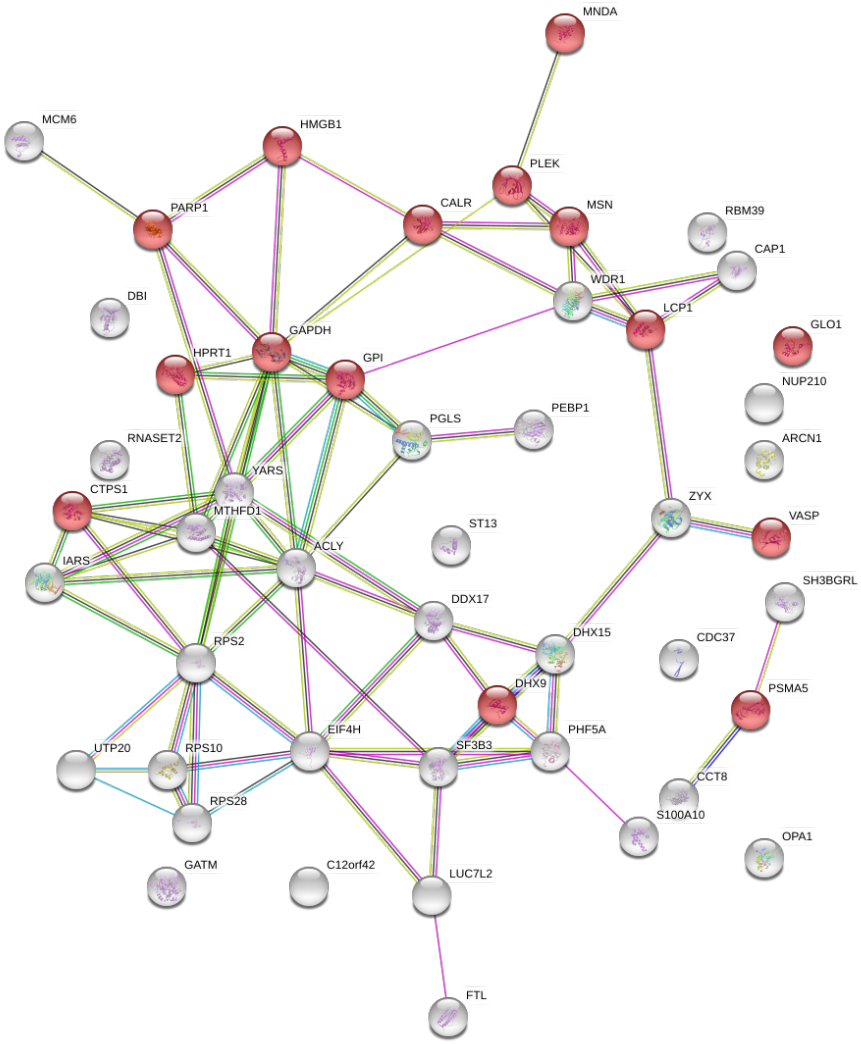


Figure 5.3 STRING interaction network. Every node represents a protein in the list, and an edge between any pair represents evidence for some kind of association or interaction between them. Evidence is either experimentally determined or predicted using diverse methods. Red colored nodes are included in the "immune system process" set.

it successfully captured the undergoing biological processes. Nevertheless, more in-depth results would have been possible with (I) sample prefractionation, which improves peptide separation and allows for the collection and identification of more spectra [37], and (II) the inclusion of technical replicates to differentiate the biological variability from experimental noise.

5.4.2 Missing data handling

One of the most frequent issues found when analyzing proteomics datasets is the handling of missing values [29]. The presence of missing data in many of the rows in the data passed to BayesQuant led to the dropping of 4803 out of 6443 peptides, producing a peptide dataset of only 1641 data points that accounted for 350 protein groups. Thanks to the missing data handling implemented in MSqRob, this tool quantified 1433 proteins, enough to continue the analysis (see table 5.5). However, the fact that the number of observed groups was 2787 implies that the management of missing values could potentially be improved in both tools, and particularly in BayesQuant.

Tool	Peptides	Proteins
MSqRob	6444	1433
BayesQuant	1641	350

Table 5.5 Peptide input and quantified proteins for each tool in the THP1 dataset.

Three different types of missing values have been defined [50]:

1. Missing Completely At Random (MCAR): Unpredictable, caused by random errors in the data acquistion step and affecting the whole dataset.

2. Missing At Random (MAR): Caused by other observed variables. Citing [50]: *Inaccurate peak detection and deconvolution of co-eluting compounds can be called MAR.*

3. Missing Not At Random (MNAR): Caused by signals under the limit of detection (LOD) of the spectrometer.

While all three are relevant, MNAR is particularly important because it introduces a bias in the dataset, where proteins are more likely to be quan-

tified if their abundance is high enough in all samples for their peptide signals to be above the LOD. Thus, proteins exhibiting abundances below the LOD due to the experimental treatment are bound to generate peptide data featuring missing values that, if not handled properly, are discarded. Subsequently, DAPs or in general proteins with variant abundance become less likely to be quantified, difficulting the extraction of protein sets.

Missing data should be imputed at the peptide level. The implementation of a module performing this task prior to quantification would improve the accuracy and throughput of the quantification process. Several approaches have been proposed, some within a probabilistic framework [29]. For example, in cases of clear MNAR, imputation could be based on an inferred global LOD, so that missing values are replaced with it. Work on this line could potentially boost the performance of BayesQuant.

5.4.3 Applicability for Novozymes data

NZ data is frequently confidential and cannot be uploaded to public servers like STRING. However, many of the databases consulted in this work can be built locally, thus removing the need of data leaving the company's facilities.

5.5 Conclusion

A fine biological interpretation of the results depends on the original question that motivated the study, and thus the work on this chapter did not focus on that, but instead showed how it can be started with the outputsability of the tools presented in the previous chapters. The results indicated that these tools can be deployed to perform computational analyses and biological

inference on protein samples with open-source software with no cost for NZ. However, the absence of proper missing data handling developed into the inability to quantify hundreds of proteins, signaling what are the next steps to improve the tools.

Chapter 6

Conclusion of the Thesis

The pipelines presented in chapter 3 and the MaxQuant suite offered the easiest to deploy and most output rich of all the cost-free pieces of software available for the proteomics community. All of them have pros and cons, and their combined usage was found to provide the best results, underlining the importance of the development of programs implementing open data formats exchangeable across tools. More functionalities can be added, including *de novo* and PTM search, and missing data management. BayesQuant, the probabilistic framework for relative quantification, presented in chapter 4 can be passed any peptide level dataset and provide estimates of the uncertainty behind traditional log2FC estimates and nuisance effects. It can be customised to expand its currently implemented model. The lack of success in the attempt to model the peptide bias using the aminoacidic sequence reveals the complexity of the competitive ionization problem. Recently developed tools like RawQuant could be applied in future work to extract from RAW files a feature-rich representation of the collected spectra that could be used to model this phenomenon with a Deep Learning approach. The results from chapter 5 manifest that NZ can at the same time decrease computational costs and access the state of the art in proteomics research

by making use of the latest open-source software, and it is encouraged to continue on that line.

Bibliography

[1] Andrea Argentini et al. “MoFF: A robust and automated approach to extract peptide ion intensities”. In: *Nature Methods* 13.12 (2016), pp. 964–966.

[2] Harald Barsnes and Marc Vaudel. “SearchGUI: a highly adaptable common interface for proteomics search and de novo engines”. In: *Journal of Proteome Research* (2018), acs.jproteome.8b00175.

[3] Robert Boyd and Árpád Somogyi. “The mobile proton hypothesis in fragmentation of protonated peptides: A perspective”. In: *Journal of the American Society for Mass Spectrometry* 21.8 (2010), pp. 1275–1278.

[4] Andrew P. Bradley. “The use of the area under the ROC curve in the evaluation of machine learning algorithms”. In: *Pattern Recognition* 30.7 (1997), pp. 1145–1159.

[5] Delphine Charif and Jean R. Lobry. “SeqinR 1.0-2: A Contributed Package to the R Project for Statistical Computing Devoted to Biological Sequences Retrieval and Analysis”. In: Springer, Berlin, Heidelberg, 2007, pp. 207–232.

[6] Siddhartha Chib and Edward Greenberg. “Understanding the Metropolis-Hastings Algorithm”. In: *The American Statistician* 49.4 (1995), pp. 327–335.

[7] Peter J A Cock et al. “Biopython: freely available Python tools for computational molecular biology and bioinformatics.” In: *Bioinformatics (Oxford, England)* 25.11 (2009), pp. 1422–3.

[8] Jürgen Cox and Matthias Mann. “MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide

protein quantification". In: *Nature Biotechnology* 26.12 (2008), pp. 1367–1372. arXiv: nbt.1511 [10.1038].

[9] Jürgen Cox et al. "Accurate Proteome-wide Label-free Quantification by Delayed Normalization and Maximal Peptide Ratio Extraction, Termed MaxLFQ". In: *Molecular & Cellular Proteomics* 13.9 (2014), pp. 2513–2526.

[10] David Croft et al. "The Reactome pathway knowledgebase". In: *Nucleic Acids Research* 42.D1 (2014), pp. D472–D477.

[11] Sven Degroeve, Lennart Martens, and Igor Jurisica. "MS2PIP: A tool for MS/MS peak intensity prediction". In: *Bioinformatics* 29.24 (2013), pp. 3199–3203.

[12] Eric W Deutsch et al. "A Guided Tour of the Trans-Proteomic Pipeline". In: 10.6 (2011), pp. 1150–1159.

[13] Daniel Domingo-Fernandez et al. "ComPath: An ecosystem for exploring, analyzing, and curating pathway databases". In: *bioRxiv* (2018), p. 353235.

[14] J. E. Elias and Gygi. S. P. "Target-Decoy Search Strategy for Mass Spectrometry-Based Proteomics". In: *Current Protocols in Protein Science* 604.SUPP.60 (2010), pp. 55–71. arXiv: arXiv:1011.1669v3.

[15] Alan G. Marshall et al. "Mass Resolution and Mass Accuracy: How Much Is Enough?" In: *Mass Spectrometry* 2.Special_Issue (2013), S0009–S0009.

[16] R. Gabriels, Lennart 1978-promotor (viaf)169280744 Martens, and Andrea copromotor Argentini. *Detectie van onverwachte post-translationele modificaties in zeer grote volumes publieke proteomics data*. 2017.

[17] Ludger J. E. Goeminne, Kris Gevaert, and Lieven Clement. "Peptide-level Robust Ridge Regression Improves Estimation, Sensitivity, and Specificity in Data-dependent Quantitative Label-free Shotgun Proteomics". In: *Molecular & Cellular Proteomics* 15.2 (2016), pp. 657–668.

[18] Ludger J.E. Goeminne et al. "Summarization vs peptide-based models in label-free quantitative proteomics: Performance, pitfalls, and data analysis guidelines". In: *Journal of Proteome Research* 14.6 (2015), pp. 2457–2465.

[19] Matthew D. Hoffman and Andrew Gelman. “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo”. In: (2011). arXiv: 1111.4246.

[20] L. Martens I. Eidhammer, K. Flikka and S.-O. Mikalsen. *Computational Methods for Mass Spectrometry Proteomics*. Wiley, 2008. ISBN: 9780470512975.

[21] Johan Ludwig William Valdemar Jensen. “Sur les fonctions convexes et les inégalités entre les valeurs moyennes”. In: *Acta mathematica* 30.1 (1906), pp. 175–193.

[22] Lukas Käll et al. “Posterior error probabilities and false discovery rates: Two sides of the same coin”. In: *Journal of Proteome Research* 7.1 (2008), pp. 40–44.

[23] M Kanehisa and S Goto. “KEGG: kyoto encyclopedia of genes and genomes.” In: *Nucleic acids research* 28.1 (2000), pp. 27–30.

[24] Sangtae Kim and Pavel A. Pevzner. “MS-GF+ makes progress towards a universal database search tool for proteomics”. In: *Nature Communications* 5 (2014), pp. 1–10.

[25] Ufuk Kirik, Jan C Refsgaard, and Lars J Jensen. “Improving peptide-spectrum matching by fragmentation prediction using Hidden Markov Models”. In: *bioRxiv Bioinformatics* (2018).

[26] Kevin A. Kovalchik et al. “Parsing and Quantification of Raw Orbitrap Mass Spectrometer Data Using RawQuant”. In: *Journal of Proteome Research* 17.6 (2018), pp. 2237–2247.

[27] John K. Kruschke. *Doing Bayesian data analysis : a tutorial with R, JAGS, and Stan*, p. 759. ISBN: 9780124058880.

[28] Michael Lawrence et al. “Software for Computing and Annotating Genomic Ranges”. In: *PLoS Computational Biology* 9.8 (2013). Ed. by Andreas Prlic, e1003118.

[29] Cosmin Lazar et al. “Accounting for the Multiple Natures of Missing Values in Label-Free Quantitative Proteomics Data Sets to Compare Imputation Strategies”. In: *Journal of Proteome Research* 15.4 (2016), pp. 1116–1125.

[30] Lev I. Levitsky et al. “IdentiPy: An Extensible Search Engine for Protein Identification in Shotgun Proteomics”. In: *Journal of Proteome Research* 17.7 (2018), pp. 2249–2255.

[31] Anne Marmagne et al. “Purification and Fractionation of Membranes for Proteomic Analyses”. In: *Arabidopsis Protocols*. Ed. by Julio Salinas and Jose J Sanchez-Serrano. Totowa, NJ: Humana Press, 2006, pp. 403–420. ISBN: 978-1-59745-003-4.

[32] Hamid Mirzaei. *Modern Proteomics – Sample Preparation, Analysis and Practical Applications*. Vol. 919. 2016. ISBN: 978-3-319-41446-1.

[33] David L. Nelson. *Lehninger : principles of biochemistry*. 5th ed. New York : W. H. Freeman and Co, 2008. ISBN: 9780716771081.

[34] Alexey I. Nesvizhskii. “A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics”. In: *Journal of Proteomics* 73.11 (2010), pp. 2092–2123.

[35] Paul D. Piehowski et al. “Sources of Technical Variability in Quantitative LC–MS Proteomics: Human Brain Tissue Sample Analysis”. In: *Journal of Proteome Research* 12.5 (2013), pp. 2128–2137.

[36] Martyn Plummer. “DSC 2003 Working Papers JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling”. In: ().

[37] Pier Giorgio Righetti et al. “Prefractionation techniques in proteome analysis: The mining tools of the third millennium”. In: *Electrophoresis* 26.2 (2005), pp. 297–319.

[38] P. Roepstorff and J. Fohlman. “Proposal for a Common Nomenclature for Sequence Ions in Mass Spectra of Peptides”. In: *Biological Mass Spectrometry* 11.11 (1984), pp. 601–601.

[39] John Salvatier, Thomas V. Wiecki, and Christopher Fonnesbeck. “Probabilistic programming in Python using PyMC3”. In: *PeerJ Computer Science* 2 (2016), e55.

[40] Punit Saraon et al. “High-throughput proteomics reveals that enzymes of the ketogenic pathway are upregulated during prostate cancer progression”. In: *Cancer Research* 72.8 Supplement (2012), pp. 4119–4119. ISSN: 0008-5472.

[41] David Shteynberg et al. “Combining Results of Multiple Search Engines in Proteomics”. In: *Molecular & Cellular Proteomics* 12.9 (2013), pp. 2383–2393.

[42] Pavel Sinitcyn, Jan Daniel Rudolph, and Jürgen Cox. “Computational Methods for Understanding Mass Spectrometry–Based Shotgun Proteomics Data”. In: *Annual Review of Biomedical Data Science* 1.1 (2018), annurev–biodatasci–080917–013516.

[43] Rob Smith et al. “Proteomics, lipidomics, metabolomics: A mass spectrometry tutorial from a computer scientist’s point of view”. In: *BMC Bioinformatics* 15.Suppl 7 (2014).

[44] Marc Sturm et al. “OpenMS – An open-source software framework for mass spectrometry”. In: *BMC Bioinformatics* 9.1 (2008), p. 163.

[45] Damian Szklarczyk et al. “The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible”. In: *Nucleic Acids Research* 45.D1 (2017), pp. D362–D368.

[46] Keqi Tang, Jason S. Page, and Richard D. Smith. “Charge competition and the linear dynamic range of detection in electrospray ionization mass spectrometry”. In: *Journal of the American Society for Mass Spectrometry* 15.10 (2004), pp. 1416–1423. arXiv: NIHMS150003.

[47] Viet Hung Tran. “Copula Variational Bayes inference via information geometry”. In: 2018 (2018), pp. 1–23. arXiv: 1803.10998.

[48] Marc Vaudel et al. “PeptideShaker enables reanalysis of MS-derived proteomics data sets: To the editor”. In: *Nature Biotechnology* 33.1 (2015), pp. 22–24. arXiv: arXiv:1011.1669v3.

[49] Kenneth Verheggen et al. “Anatomy and evolution of database search engines-a central component of mass spectrometry based proteomic workflows”. In: *Mass Spectrometry Reviews* (2017), pp. 1–15.

[50] Runmin Wei et al. “Missing Value Imputation Approach for Mass Spectrometry-based Metabolomics Data”. In: *Scientific Reports* 8.1 (2018), pp. 1–10.