Bielefeld University
Faculty of Technology
The
Applied Neuroinformatics
Group

International Graduate School
Bioinformatics & Genome Research

Dissertation

Peak Intensity Prediction in Mass Spectra using Machine Learning Methods

Wiebke Timm

September 12, 2008

Der Technischen Fakultät der Universität Bielefeld vorgelegt zur Erlangung des akademischen Grades *Dr. rer. nat.*

Supervisors: Jun. Prof. Tim W. Nattkemper and Prof. Sebastian Böcker

This thesis was typeset with LaTeX using KOMA-Script. Hermann Zapf's Euler font was used for setting mathematics.

Gedruckt auf alterungsbeständigem Papier nach ISO 9706.

Contents

Tables and Figures

List of Tables

List of Figures

1. Introduction

Mass spectrometry (MS) is an indispensable technique for the analysis of proteins and peptides in life sciences. Various methods have been developed to allow the comparison of protein abundances in cells that differ in terms of their states. A growing number of studies in proteomics aim to quantitatively characterize proteomes for a better understanding of cellular mechanisms (see the overview article by Bantscheff *et al.* 2007). These studies use either chemical labeling or label-free methods for protein quantification.

Using isotopic labeling methods, protein mixtures are tagged with an isotope that can be used to tell samples apart by their mass shift and to directly compare peaks from different samples. Labeling methods include SILAC (Stable Isotope Labeling with Amino acids in Cell culture, Ong *et al.* 2002) and ICAT (Isotope Coded Affinity Tags, Gygi *et al.* 1999). These methods allow an accurate quantification relative to the tagged sample at the expense of additional experimental processing steps and immense costs for the labeling reagents.

In contrast, label-free methods use only the signal intensities or the number of detected peaks per peptide (spectral counts) to estimate peptide abundances. But peak intensities also depend on peptide ionization efficiencies, that are influenced by a peptide's composition and the chemical environment. In other words, the sensitivity of an MS device varies between peptides. Therefore, two equally abundant peptides will generally lead to different peak intensities. Absolute quantification using label-free methods is possible through the use of reference peptides at very high accuracy, for example Steen *et al.* (2005) and Mayr *et al.* (2006). But again, such methods require significant experimental effort. Consequently, label-free techniques are routinely used only for differential quantification, that is, the determination of concentration ratios between samples.

Nonetheless, label-free methods have several intrinsic advantages over labeling techniques. Obviously, they do not require the labor- and cost-intensive labeling. Also, there is no fundamental limit to the number of samples that can be compared. Unlike labeling techniques, label-free methods do not increase the mass spectral complexity. They have the potential to analyze a higher range of protein concentrations and to achieve a higher proteome coverage (Bantscheff *et al.*, 2007).

There exist two fundamentally different experimental setups for label-free quantification using MS: In both cases, proteins are digested and peptides are separated using liquid chromatography (LC). In the first case, the LC is directly coupled to an electrospray ionization

(ESI) mass spectrometer, which allows for a comparatively simple experimental setup and a rapid analysis of separated peptides. In the second case, LC fractions are spotted on target plates and analyzed using matrix-assisted laser desorption ionization (MALDI) MS (Ji and Li, 2005; Mirgorodskaya *et al.*, 2005; Neubert *et al.*, 2008). Using LC-MALDI is more time-consuming but has certain advantages such as a more efficient data-dependent analysis: The sample portions from the LC can be stored for several days and reanalyzed when necessary, so it is possible to acquire fragmentation ion spectra for all MS parent ions that are of interest. Spectra are easier to interpret and compare because mostly singly charged ions are observed.

If an estimate of the peptide-specific sensitivity were possible, this would allow the use of label-free techniques for absolute quantification. This would save the enormous costs of the labeling and facilitate the realization of more large comparative studies.

In this work, I find out if and how peptide-specific sensitivities of unknown peptides in MS spectra can be modeled. The approach to this problem presented in this work is a combination of simulation and supervised learning. Which properties of the peptides are most relevant to this problem is an important question in this context. It is understood that one cannot experimentally determine the peptide-specific sensitivity for all possible peptides. Machine learning methods have been designed to predict the response of a system using only examples of the system's input-response behavior. Their application facilitating the prediction of peptide-specific sensitivity for peptides that were not measured previously.

Most of the presented results are based on mass spectra from MALDI-TOF mass spectrometry. Proteins were separated with 2D polyacrylamide gel electrophoresis (2D-PAGE) prior to mass spectrometric analysis. In the corresponding experimental setup, all peptides in a spectrum have the same abundance, given a correct preparation, for gel spots consisting of one protein. Therefore, the peptide-specific sensitivity of the MS device can be accessed by comparing peak intensities in every such spectrum. A peptide-specific correction factor can then be calculated by dividing one over the corresponding peak intensity. A first application of the method to LC-ESI spectra is investigated, too.

This work constitutes an important step to facilitate the enhancement of label-free quantification accuracy: I show that the prediction of peptide-specific sensitivities is indeed feasible even on a small dataset. Knowledge extraction with feature selection methods leads to the rediscovery of known as well as new properties that are relevant for this problem. Least-angle regression (Efron *et al.*, 2004), a modern feature selection technique is evaluated for this purpose among others, and is shown to performs comparatively well on noisy MS data.

1.1. Outline

This thesis is organized in three parts. The first part contains background knowledge. Chapter 2 explains applications of MS in proteomics, and computational methods with a focus on existing quantitative methods. Chapter 3 introduces to supervised learning in general as well as the regression methods applied in this work. The remainder of the chapter deals with model selection and evaluation.

The second part motivates the scope, the approach, and the data used for this thesis. Chapter 4 features the scope of this work in the context of related work. An explanation of the modeling of the MS analysis workflow for proteins can be found in Chapter 5, as well as common sources of noise and errors. Chapter 6 presents data preprocessing, normalization, and analysis. Although the main focus of this work is on MALDI MS, another type of MS spectra (electrospray ionization, ESI) has been processed additionally. These different types of data make up two parts of this chapter.

The third part deals with the encoding of peptide sequences (feature vectors) as input for the learning methods and results of the prediction of peptide-specific sensitivities, as well as the feature selection approach and results. Chapter 7 introduces two types of sets of predictors (feature sets) as peptide encodings. Either only sequence information or additional chemical information about peptide components (amino acids) can be used. Four initial feature sets are introduced. Prediction results obtained with a ν-support vector regression on these feature sets are presented. First tests to transfer the method to ESI data are shown, too. Chapter 8 presents a comparison of feature subsets obtained with different methods (a forward stepwise selection heuristic, least-angle-regression, and an L_1-penalized generalized linear model). In addition, feature importance has been assessed with random forest regression. The chapter ends with a summary of properties and features that are most relevant for peptide-specific sensitivity prediction, and an evaluation of the feature selection methods. Additional analyses are gathered in Chapter 9, for instance the use of unlabeled data or a comparison to existing work.

Chapter 10 closes this thesis with a summary and discussion of the results, and an outlook to future prospects.

A key to notations can be found in Section A.1 of the appendix. Words in italic are explained in the glossary (Appendix B) if they are not explained where they appear first.

1.2. Publications

Parts of this thesis have been published or are under revision.

As first author:

- Timm, W., Böcker, S., and Nattkemper, T. W. (2006). Peak Intensity Prediction for PMF Mass Spectra Using Support Vector Regression. In *Applied Artificial Intelligence - Proceedings of the 7th International FLINS Conference*, pages 565–572. World Scientific.
- Timm, W., Scherbart, A., Böcker, S., Kohlbacher, O., and Nattkemper, T. W. (2008). Peak intensity prediction in MALDI-TOF mass spectrometry: a machine learning study to support quantitative proteomics. In *BMC Bioinformatics*. first revision under review.

Co-authorship:

- Scherbart, A., Timm, W., Böcker, S., and Nattkemper, T. W. (2007). Neural network approach for mass spectrometry prediction by peptide prototyping. *ArtiÞcial Neural Networks - ICANN 2007*. **4669**, pages 90–99. Springer.
- Scherbart, A., Timm, W., Böcker, S., and Nattkemper, T. W. (2007). Som-based peptide prototyping for mass spectrometry peak intensity prediction. In *Proceedings of Workshop on Self-Organizing Maps (WSOM'07)*.

2. Proteomics and mass spectrometry

Proteomics is the field of research that deals with the exploration of the whole proteome, i.e. the entirety of all proteins and post-translational modifications[1] of a cell or organism. The proteome of an organism is not fixed as is the genome, but is undergoing constant changes. A prominent example are the butterfly and the caterpillar. Both have the same genome, but the proteome is different. For higher organisms, the proteome even differs between different parts of the body. And it differs between diseased and healthy tissue. Proteomics research is aimed at the analysis of different states of the proteome of organisms. This should enable biologists to gain insights about the function of organisms. Medical scientists hope to develop agents to cure diseases like cancer or hereditary dysfunctions based on proteomics research.

Proteomics research is a fast-developing field, and yet today's technologies and methods are far from matching the needs to cope with the enormous complexity of living organisms. Large whole-proteome studies target simple organisms such as yeast (Ghaemmaghami *et al.*, 2003) or bacteria (Ishihama *et al.*, 2008). Mammalians are even more complex. The Human Proteome Organization (HUPO) coordinates the development of new technologies, techniques, and training to study aspects of the human proteome (Foster *et al.*, 2006; States *et al.*, 2006).

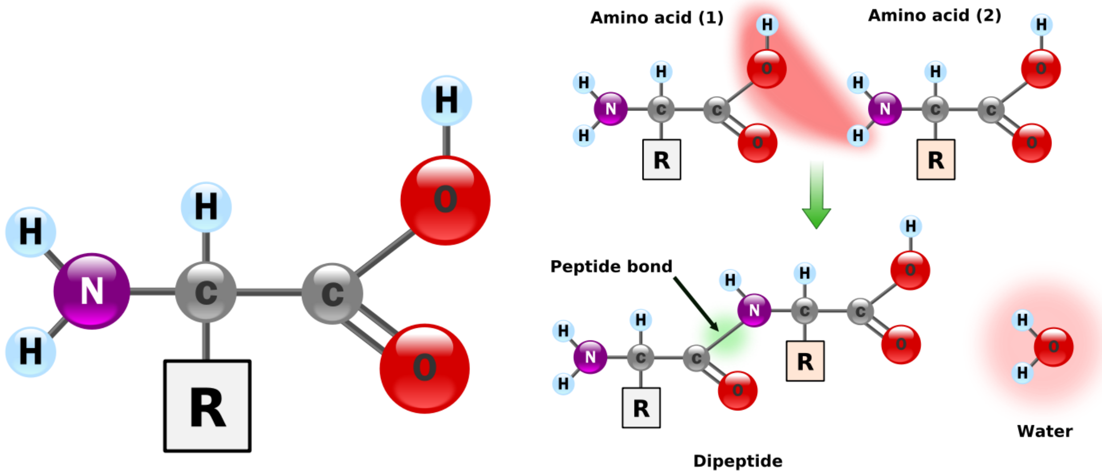
Being relatively young, the field of proteomics originates from protein analytics, a much older discipline. Classical protein analytics deals with structure and function of individual proteins. In contrast, proteomics aims to comprehend the entirety of proteins in a biological sample.

Proteomics methods concentrate on qualitative and quantitative description of proteins in cells and tissues as a snapshot under defined conditions. Today it is possible to identify single proteins relevant for a certain cell state from a mixture of more than thousands of proteins. Mass spectrometry is a key technology for high-throughput protein analysis.

[1] both are explained in this chapter

2.1. Proteins and peptides: an overview

Proteins govern a huge variety of cellular functions. Some (e.g. enzymes) determine which reactions take place, others have a key role in signaling and transport, and structural proteins form rigid elements allow cells to maintain size and shape or even generate mechanical forces (motor proteins). Not only is their presence or absence important for the state of a cell. Changes in the abundance of certain proteins can make the difference between a normal and a diseased or nonfunctional cell.



(a) Amino acid structure

(b) Peptide bond

Figure 2.1: Amino acids can link via a peptide bond to form dipeptides, polypeptides or even larger chains (proteins). The spheres denote atoms, the sticks bonds between them. All peptides have this structure, but the residual group (denoted by the "R") differs. Images by Yassine Mrabet, published under free document license in the Wikipedia (`http://www.wikipedia.com`).

A protein is a molecule that consists of a sequence of *amino acids* (see chemical structure in Fig. 2.1a). Two amino acids can link to each other via the *peptide bond* to form a dipeptide (Fig. 2.1b). Even more amino acids chained together are called polypeptide. Proteins are very large polypeptide chains. There are twenty different amino acids that occur in a protein naturally. Each has unique properties and a one-letter code (see Table A.4). In bioinformatics, peptides are usually modeled as strings over the alphabet $\mathcal{A}$ that consists of these 20 characters. For proteins, that is often not sufficient. They form higher-order structures depending on the amino acid sequence and their chemical environment. The amino acid sequence is referred to as the *primary structure* of a protein. Depending on its amino acid sequence, local structures are formed by hydrogen bonds. A protein may contain various of these local or *secondary structures*. Most common are the *alpha helix* and *beta sheet*. The *ter-*

tiary structure describes the three-dimensional conformation of the whole protein. A protein's tertiary structure is stabilized by a number of factors as its environment, disulphide bridges, salt bridges, hydrogen bonds, and steric constraints. For example, in a hydrophilic solution such as water, hydrophobic amino acids tend to be buried inside of the protein whereas hydrophilic amino acid can be found on the outside. Finally, multiple proteins can form large complexes. This is called the *quaternary structure*. Information on the structures of proteins can be found inside a world-wide repository, the Protein Databank (PDB, Berman *et al.* 2003, 2007). For further reading, a biochemistry textbook such as Berg *et al.* (2006) is recommended.

2.1.1. Post-translational modifications

Proteins are derived in multiple steps from the genetical code stored in the *DNA*. During *translation*, the protein is formed. Often, this is not the final protein, but it is modified before being transported to its target localization in the cell. This modification is called *post-translational modification* (PTM). There is a huge variety of possible PTMs. Additional signal sequences can control to what location in the cell the protein is directed. Additive modifications like phosphorylation, oxidation, or glycosylation play an important role in inter- and intracellular signaling. The working mechanism of these signaling pathways is under active research. For example, Chiarugi and Buricchi (2007) wrote a review about the dynamics and interaction between tyrosine phosphorylation and methionine oxidation. Methods for PTM analysis with MS are far from being routinely applied, but there are promising developments. Steen *et al.* (2006) analyze the behavior of phosphorylated peptides in MS. Earlier, they already presented an approach to label-free quantitation of protein phosphorylation stoichiometry with MS (Steen *et al.*, 2005). Glycosylations are probably the most complex modifications to analyze with MS: They can contain forked chains of various types of sugar molecules. Wuhrer *et al.* (2005) reviewed MS methods for glycolsylation analysis.

Modifications can be added to or removed from a protein. This does not only change the mass but also the chemical behavior of affected proteins or peptides. PTMs increase the complexity of the proteome dramatically, posing an additional challenge to proteomics applications.

2.2. Inside the machine – mass spectrometry

Mass spectrometry (MS) is a technique to measure mass-to-charge ratios of molecules. In combination with other techniques for protein separation and fragmentation of proteins, it allows protein identification in large complex mixtures. Mass spectrometry based methods for protein quantitation are under active development.

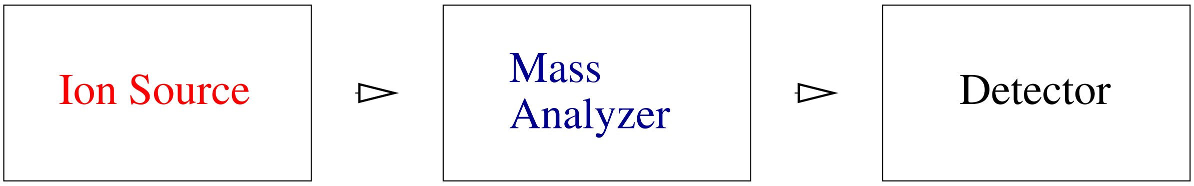


Figure 2.2: Principal concept of a mass spectrometry device: Molecules are ionized at the *ion source* and passed on to be separated by mass and charge with the *mass analyzer*. The number of ions for each mass are counted with the *detector*. The technique used for ionization, separation, and detection differ between different types of instruments.

The general setup of a simple mass spectrometer can be seen in Fig. 2.2. In the *ion source*, a mixture of molecules is ionized to facilitate the separation of these ions by their mass and charge with a *mass analyzer*. The resulting ions for each mass are then counted by a *detector*. Many ionization techniques and mass analyzers are available. Only so called soft ionization methods are introduced here, because these leave large ions such as peptides intact during ionization and are therefore used for protein analysis.

2.2.1. Ion source

Two soft ionization methods commonly used in proteomics are Matrix-assisted laser desorption ionization (MALDI) and electrospray ionization (ESI). There are more techniques available for the analysis of other sample types which are not introduced here.

Matrix-assisted laser desorption ionization (MALDI)

Laser-desorption ionization as an *ion source* for mass spectrometry was invented by Michael Karas and Franz Hillenkamp (Karas and Hillenkamp, 1987) and independently by Koichi Tanaka (Tanaka *et al.*, 1988). For use with MALDI, the *analyte* has to be mixed with a matrix substance. That mixture crystalizes when dried down. The crystallized sample is shot with a laser in vacuum. The matrix absorbs the energy and bursts at some point, releasing ionized molecules in the process. It can be observed that most MALDI ions are only singly charged, which makes MALDI spectra comparably easy to interpret. The exact processes that lead to the ionization of molecules in MALDI are not fully understood. Karas *et al.* proposed the theory of "lucky survivors" according to which multiply charged ions are created initially but are neutralized by an excess of electrons. Most ions are neutralized again, but a few singly charged ones survive the process (Karas *et al.*, 2000). Knochenmuss and Zenobi (2003) present a model of their own together with a review of recent studies on the subject. Different matrix substances are used depending on the aim of the analysis. The matrix heavily influences the

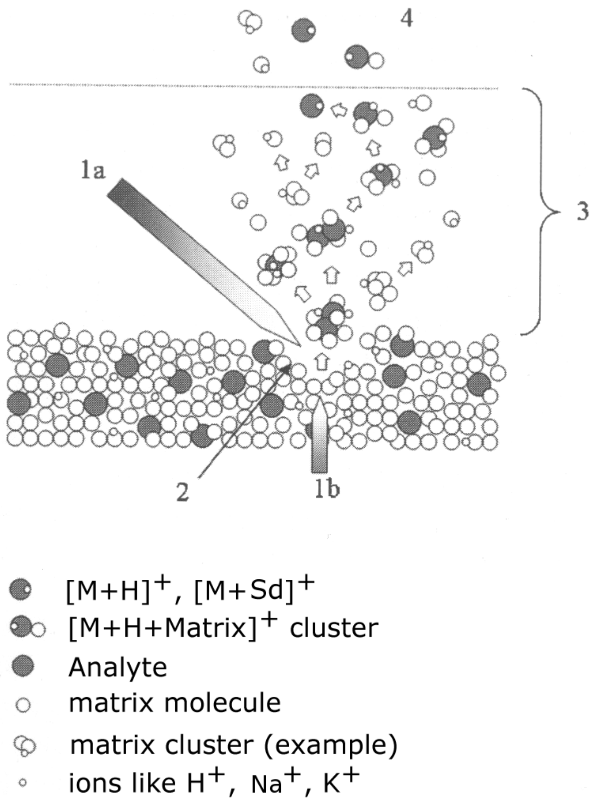


Figure 2.3: Desorption and positive ion generation in MALDI ion source: 1) High energy particles (1a photons), 2) Collision cascade and desorption of clusters, 3) *Plume*: High pressure, lots of collisions. Decomposition of clusters, generation of ions by reactions between molecules and ions, adsorption of H^+ where applicable. 4) no further collisions in high vacuum, i.e. only decay of single molecules. Image copied and translated from Budzikiewicz and Schäfer (2005)

ionization process. For proteins, positive ion mode (i.e. positive ions are created) is used. A MALDI spectrum is shown in Fig. 2.4. Further reviews on MALDI can be found in Bahr *et al.* (1994); Beavis *et al.* (1992); Karas *et al.* (1991).

Electrospray ionization (ESI)

The use of electrospray to ionize molecules was first introduced by Dole *et al.* (1968); Mack *et al.* (1970). John Fenn was the first to utilize this in combination with mass spectrometry twenty years later (Fenn *et al.*, 1989; Meng *et al.*, 1988; Yamashita and Fenn, 1984).

With ESI, the analyte is introduced in liquid form. For protein analysis, ESI is often coupled to liquid chromatography (LC) to reduce spectra complexity by separating the protein mixture before MS analysis. A typical raw spectrum is shown in Fig. 2.6. The analyte solution passes through an electrospray needle that has a high potential difference with respect to the sampling cone which is charged as counter electrode (see Fig. 2.5). This causes small charged

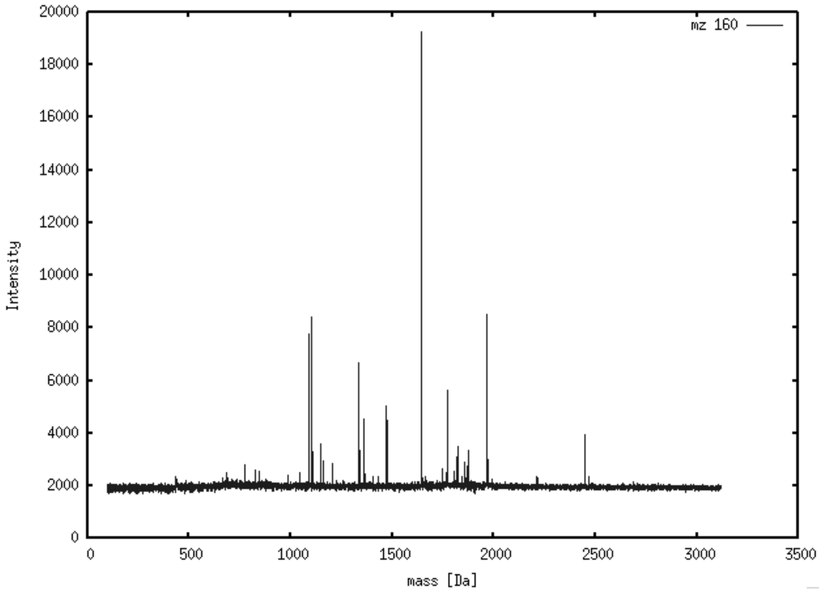
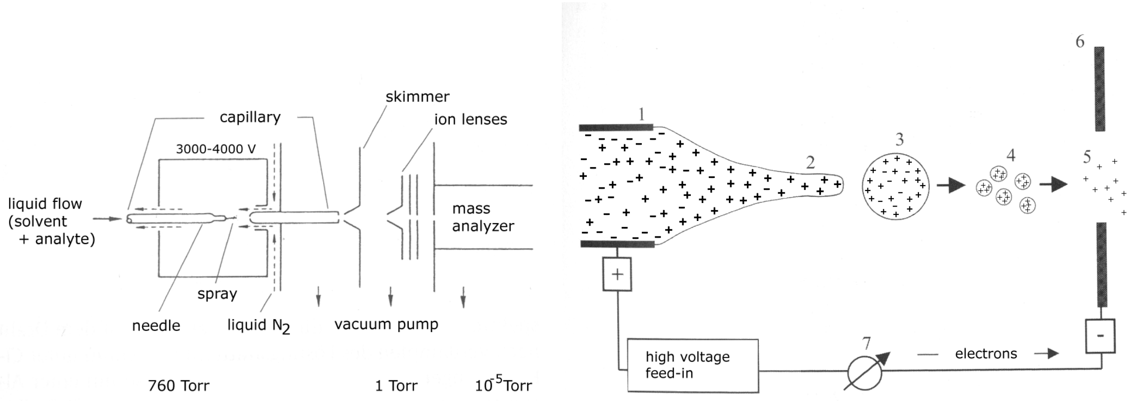


Figure 2.4: A typical MALDI spectrum: Mass-to-charge is plotted against the detector count (Intensity).



(a) Principle of ESI ion source

(b) ESI spray area enlarged

Figure 2.5: Electrospray ionization (ESI) schematics. Image source: *left*: Budzikiewicz and Schäfer (2005); *right*: Hesse *et al.* (2005).

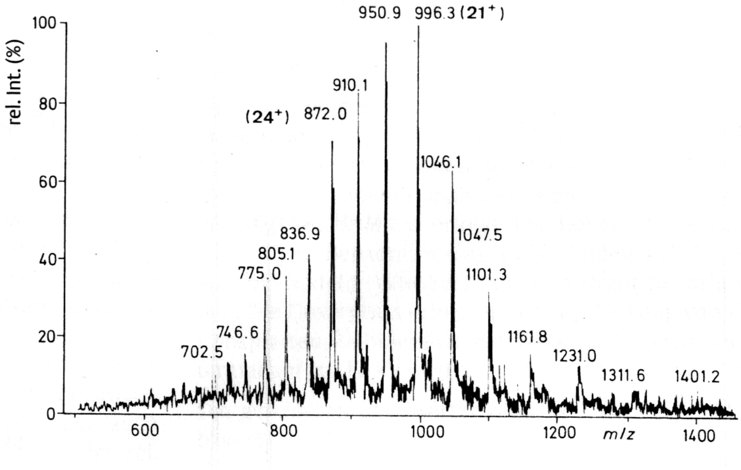


Figure 2.6: ESI mass spectrum of Interleukin 6, taken with a Finnigan MAT TSQ-700. Image source: Hesse *et al.* 2005

droplets from the needle to be repelled from the needle towards the sampling cone. These droplets have a high surface charge. Solvent evaporates during the traversal towards the sampling cone, which makes the droplet shrink while keeping the same surface charge. At some point, the charge becomes to high for the surface tension to hold the droplet together and it "explodes" into smaller droplets or charged analyte molecules. A schematic of the process is depicted in Fig. 2.5b. The process repeats for the droplets. Analyte ions can have multiple charges. With these, even larger molecules can be analyzed that would be outside the mass detection range if they were only charged once. The drawback is a more complex spectrum.

2.2.2. Mass analyzers

While the *ion source* determines what kind of samples can be analyzed with the mass spectrometer, the *mass analyzer* determines the mass range, sensitivity and accuracy of the instrument. All mass analyzers utilize the behavior of charged particles in electric and magnetic fields in vacuum. Lorentz force law (Eqn. 2.1) and Newton's second law of motion (Eqn. 2.2) apply:

$$F = q\ (E + \mathbf{v} \times \mathbf{B}) \tag{2.1}$$

$$F = m\mathbf{a} \tag{2.2}$$

Here, $\mathbf{F}$ is the force applied to the ion, q its charge, E the electric field, $\boldsymbol{v} \times \mathbf{B}$ the vector cross product of the ion velocity and the magnetic field, m its mass, and $\boldsymbol{a}$ the ion's acceleration. Remember that $\boldsymbol{a} = \dot{\boldsymbol{v}}$.

The differential equation

$$\frac{m}{q}\,\boldsymbol{a} = E + \boldsymbol{v} \times \mathbf{B}$$

then determines the particle's motion in space and time if we know the particle's initial condition. So particles with the same mass m and charge q behave exactly the same. Mass spectrometry is commonly presented with m/z (the *mass-to-charge ratio*) on the x-axis, where $z = q/e$ is the number of elementary charges e the ion carries.

There are a lot of different types of mass analyzers of which the most common ones are itemized here.

- **Sector field mass analyzer** This analyzer uses an electric or magnetic field to deflect accelerated ions. The path the ions take are bent according to their mass-to-charge ratio. Lighter, more-charged, or faster-moving ions are deflected more. Detectors at different positions detect a certain range of mass-to-charge ratios each.
- **Time-of-flight (TOF)**

  A time-of-flight analyzer (Fig. 2.7) accelerates the produced ions with a static electric field and measures the time they need to reach the detector. For particles of the same charge, their acceleration depends only on their mass. Lighter particles reach the detector earlier.

  Modern TOF analyzers work in reflectron mode, where particles are reflected at one end of the instrument with an electric field to prolong the flight area without enlarging the whole instrument. This leads to a better separation and thus a higher mass resolution. It can correct for the fact that ions do not start at exactly the same position when accelerated.
- **Linear quadrupole ion trap**

  Quadrupole mass analyzers (Fig. 2.8) use oscillating electrical fields to selectively stabilize or destabilize ions passing through a radio frequency (RF) quadrupole field. Using this, certain ions are trapped in a two-dimensional electrical field and can be selectively discarded from the trap by m/z. A set of quadrupole rods and a static electrical potential at the ends of the rods confine the ions, as shown in Fig. 2.8. See Douglas *et al.* (2005) for further reading on linear ion traps.

  Clearly, the advantage is the ability to selectively pass or discard ions which makes this type of mass analyzer ideal for a Tandem MS device (see Section 2.2.4).

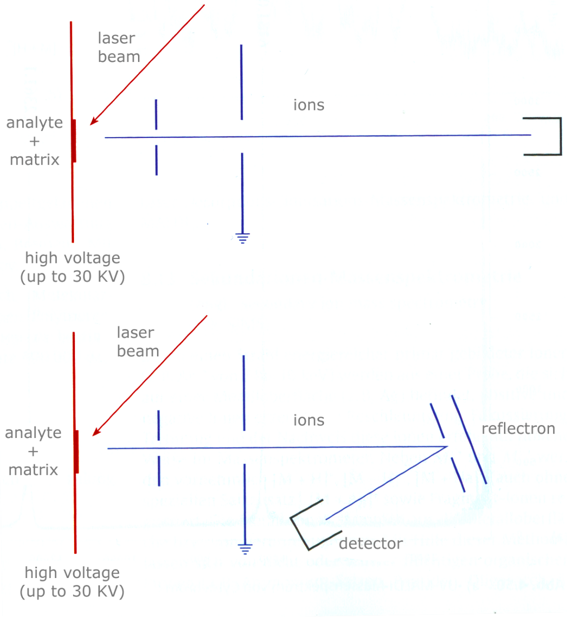


Figure 2.7: Concept of a MALDI-TOF mass spectrometer with reflectron (bottom) and without (top). Molecules are mixed with a matrix substance and crystallize when dried down. They are ionized by shooting a laser into the crystallized sample (see Fig. 2.3). A force field is applied right after the laser hits, which accelerates the ions depending on their mass and charge. They fly along a force field free region. The time between the acceleration and the detector hits is measured. The amount of hits at each time point can be plotted into a typical mass spectrum (see Fig. 2.4) after transformation of time-of-flight into mass-to-charge. Image source: Hesse *et al.* (2005).

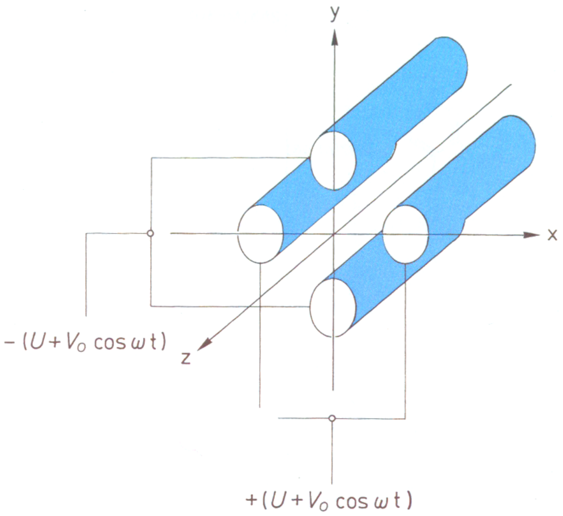


Figure 2.8: Quadrupole mass analyzer consists of four parallel rods. A radio frequency between opposing rod pairs is superimposed with a direct current voltage. Ions traversing through the rods can only pass them if their m/z ratio is appropriate for the current voltage ratio. The other ions collide with the rods.

- **Fourier Transform Ion Cyclotron Resonance (FT-ICR) mass analyzer** The FT-ICR makes ions move on a circular path with a homogenous magnetic field. The rotational frequency depends on the m/z of an ion. When an alternating electrical field is applied that matches the angular frequency of a circulating ion, cyclotron resonance occurs: The radius of the resonating ion grows by taking up energy from the alternating field. This change can be measured by detectors at fixed positions. To measure ions with different masses, the alternating field is changed and the signals for different masses retrieved via *Fourier transform* (FT). These mass analyzers have high accuracy and extremely high mass resolution. The latter depends on the applied homogeneity and field intensity. Currently applied FT-ICR have magnetic flux densities of 6 to 10 T. As a comparison, the earth's magnetic field has 31 µT, a typical refrigerator magnet 5 mT[2]. A superconductive magnet is necessary to attain such a strong field. Therefore, the device has to be cooled with liquid helium. Marshall *et al.* (1998) offer an introduction to the principles and generic applications of FT-ICR mass spectrometry.

- **Orbitrap** The orbitrap is the most recent development of ion trap mass spectrometers. Ions are shot into the instrument radial to a central electrode. They take up a circular motion (orbit) through electrostatic attraction. At the same time they oscillate along the axis of the central electrode. This oscillation causes signals in the detectors which can be mapped to mass-to-charge ratios by FT. In contrast to the FT-ICR, an orbitrap uses an electrostatic rather than a magnetic field. Therefore, no cooling is necessary. The resolution is nearly as good as that of an FT-ICR.

Manufacturers of MS devices for protein analysis are Bruker Daltonics (`http://www.bdal.de`), Applied Biosystems (`http://www.AppliedBiosystems.com`), and Thermo Fisher Scientific (`http://www.thermo.com`). Except sector field MS, which is more appropriate for smaller molecules, all mentioned mass analyzers are built into modern MS devices.

2.2.3. Detector

Two types of detectors can be distinguished. For mass analyzers that destroy the ions as they are analyzed (TOF or sector field analyzers), the detector is a device that detects a particle and multiplies its effect, since the number of detected ions is often very small. Possible detectors are a photo multiplier, secondary ion multiplier, ion-to-photon detector, Faraday cup, channel electron multiplier, or Daly detector. In the early days of MS, photo plates were used. Modern commercial MS devices have micro-channel plates (specialized photo multipliers) as detectors. Dubois *et al.* (1999) published a comparison between ion-to-photon and micro-channel plates.

[2] According to the NewScientist issue on April 12th 1997, 16T were used by researchers of the University of Nottingham and the University of Nijmegen to levitate a frog.

In FT-ICR or orbitrap mass analyzers, ions are measured by induction. They are not absorbed but only pass a pair of metal surfaces, producing a weak AC current. This is why instruments with this technique can measure ions multiple times, accounting for their high mass resolution.

2.2.4. Tandem MS (MS/MS)

For protein identification, proteins are fragmented into smaller pieces prior to MS analysis. The list of masses that occur in the spectrum is called *peptide mass fingerprint* (PMF). This list is usually compared against a database of theoretical PMFs to acquire a list of candidate proteins that is then ranked according to the further context.

However, if samples are more complex, identification of the contained proteins by their peptide masses alone is no longer feasible. Tandem MS adds at least one *collision chamber* and another mass analyzer to the setup (Fig. 2.9). Selected ions from the first analyzer called *parent ions* are passed on to a collision chamber where they are fragmented. A fragmentation method commonly used in proteomics is collision-induced dissociation (CID). The fragment ions are then passed on to another mass analyzer. From the fragment spectra or MS^2 spectra, additional information about the possible constitution of the parent ions can be drawn. If in doubt about certain fragments (for example in case of neutral loss), these can be further fragmented resulting in MS^3 spectra.

With tandem MS, different experimental setups are possible. One mode commonly used for protein analysis is multiple reaction monitoring (MRM). With MRM the instrument can be configured to scan for multiple preset mass-to-charge ratios. This allows to identify specific ions that were observed but not identified previously. Another application is the scan for known molecules, for example in pharmacokinetic studies.

2.3. Protein separation techniques

To decrease sample complexity protein separation techniques are applied prior to the MS analysis. These existed already before MS was used for protein identification in such a large scale and were adapted to optimally facilitate the MS analysis.

Two-dimensional polyacrylamide gel electrophoresis (2-D PAGE) 2-D PAGE combines isoelectric focussing (IF) and sodium dodecyl sulphate polyacrylamide gel electrophoresis (SDS-PAGE). Both are techniques for the separation of complex protein mixtures. These are carried out orthogonally to each other, resulting in a higher resolution.

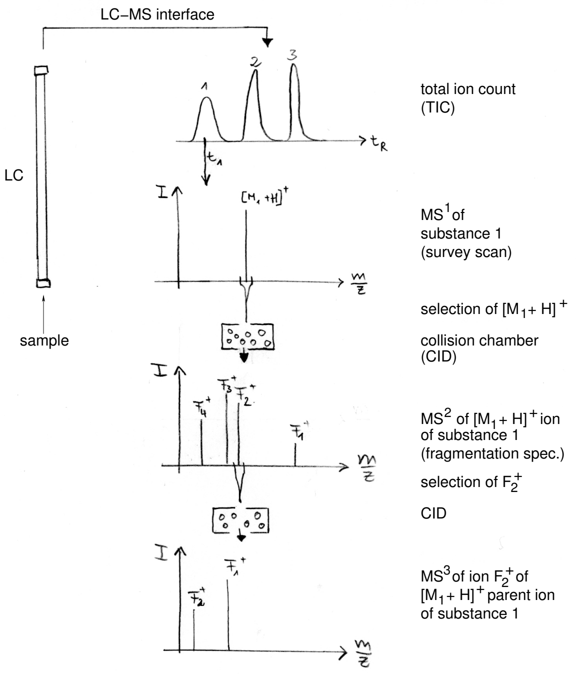


Figure 2.9: Workflow of a tandem MS instrument: A complex mixture is separated with LC, portions of analyte are accumulated. The often hydrophilic solvent is pumped off. The remaining sample is passed to a mass analyzer capable of passing or discarding selected ions, such as a quadrupole. A normal mass spectrum of the mixture is taken. Ions with a specific mass-to-charge ratio are passed on to a *collision chamber*, where they are fragmented via collision-induced dissociation (CID). The fragments are analyzed in another mass analyzer, resulting in the MS^2 spectrum. An additional fragmentation chamber and analyzer can be used to produce MS^3 spectra.

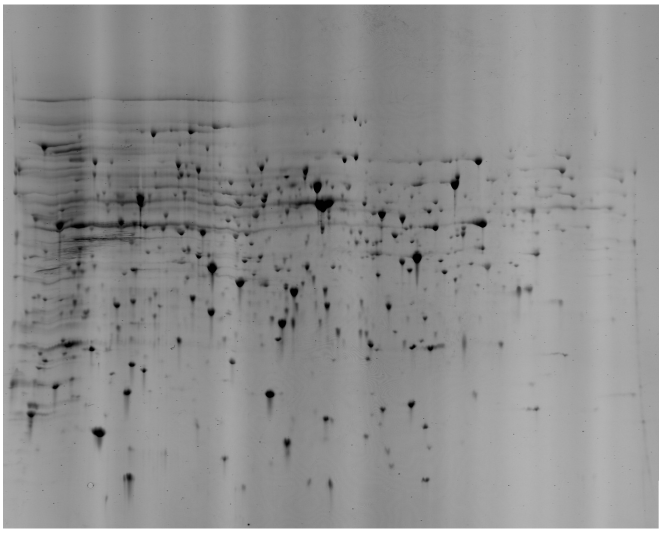


Figure 2.10: Example of an image of a two-dimensional polyacrylamide gel electrophoresis (2D-PAGE). This one originates from a cytoplasmic protein extract from cells of *Corynebacterium glutamicum* from the public ExPASy World-2DPAGE Repository (Li *et al.* 2007, `http://world-2dpage.expasy.org/`)

The IF step utilizes a pH gradient and separates proteins by their content of basic and acidic residues (separation by isoelectric point (IP)).

SDS-PAGE is used for the second dimension. It separates the proteins by their size. An electrical field is applied to the polyacrylamide gel, which draws the proteins towards the anode, because they are charged more negatively than the surrounding gels. Depending on their size they are obstructed more or less by the gel. Smaller more compact molecules move faster than bulkier ones.

The resulting spots in the gel contain similar protein species. A staining step is necessary to make the protein spots visible in the gel. Different staining solutions exist that influence the method's sensitivity and linearity with protein amount. Palagi *et al.* (2006) wrote a review that discusses gel image analysis tools.

Liquid Chromatography (LC) Chromatography is a separation technique for molecule mixtures. The sample is mixed with a mobile phase (a liquid or gas) that is moved through a stationary phase (solid). Molecules interact with the stationary phase that impedes their movement while the mobile phase carries them with it. Separation takes place because different molecules in the mixture have different affinity to the mobile and stationary phase. In Liquid Chromatography (LC), the mobile phase is a liquid.

High-performance liquid chromatography (HPLC) is often used to separate peptide mixtures prior to MS analysis. Applications for separation of other organic molecules or biopolymers in analytical chemistry and biochemistry exist as well. With HPLC, the stationary phase consists of particles packed into a column, and the sample is forced into the column under high pressure. Molecules are separated based on their polarity. Commonly, reversed-phase HPLC is used, where there is a non-polar stationary phase and a moderately polar mobile phase. Interaction of the analyte with the stationary phase does not only depend on polar moments but structural properties also play a role. The time an analyte needs to pass the column is called *retention time* (t_R).

2.4. LC coupled to ESI or MALDI – principle and differences

A complex peptide mixture resulting from tryptic digestion is injected into the LC column, usually by a sampling robot. Analyte leaving the column is collected and automatically inserted into the tandem MS when enough volume has accumulated or in certain time intervals. For each sample, many spectra are taken, introducing *retention time* as a third dimension to the acquired spectra.

When coupled to ESI, only a certain amount of time is available for the analysis of a sample portion. This time has to be divided between high resolution survey scans over the whole mass range (MS^1) and fragmentation spectra (MS^2). Fragmentation spectra facilitate identification whereas MS^1 spectra are necessary for quantification. Both modi have to be balanced depending on the aim of the study. With LC-ESI samples can be analyzed very rapidly, but cannot cover all peptides in a sample in one run. This approach is called *shotgun proteomics* because the analysis is similar to firing randomly with a shotgun: Usually, in a single tandem MS run, only a small part of the proteome is seen when applying this method, the protein mix is under-sampled. To increase the sampling rate, multiple runs with the same sample, or a protein separation by SDS-PAGE and subsequent analysis of fractions of the separated proteins can be performed.

For LC-MALDI, on the other hand, a sample separated by LC has to be spotted onto targets automatically. Matrix substance and analyte need time to crystallize before MS analysis. This forces an off-line procedure, and at the same time allows to reanalyze each sample portion as necessary, and take as much time as necessary to fragment all parent ions of interest. Thus, LC-MALDI is more time-intensive than a single LC-ESI run, but does not need to be run multiple times to analyze the sample as complete as possible. Another advantage of MALDI is that plates with droplets can be stored for several days, allowing to reanalyze samples or even a complete run in case of a machine error or difficulties with the acquired spectra. Thus, it is great for a data-dependent analysis. MALDI spectra are easier to interpret because mainly singly charged ions are observed.

During the last years, LC-ESI has become very popular for high-throughput protein analysis. It is widely accepted to be more reproducible than MALDI MS. A recent study shows that LC-MALDI can be highly reproducible, too (Neubert *et al.*, 2008). Other groups published quantitative studies analyzing complex protein mixtures using MALDI as well (Gobom *et al.*, 2000; Griffin *et al.*, 2001; Ji and Li, 2005; Krijgsveld *et al.*, 2003; Mirgorodskaya *et al.*, 2005).

2.5. Computational methods for proteomics

With the enormous amount of data produced by MS, computational tools and methods are necessary to make use of it. Computational proteomics is a large, fast-evolving area of research. Before any external method can be applied, software inside of MS devices transforms the measured physical property into mass-to-charge values according to a calibration method. The resulting data output by the instrument is called raw data. Tools for proteomics analysis exist for image analysis of two-dimensional gels, peak-detection and extraction of two- and three-dimensional MS data, retention-time alignment for LC-MS, de novo sequencing from tandem MS, identification and quantification of proteins, validation and quality control, data compression, and storage. Often, multiple of these methods are configured in a pipeline. This introduction cannot even begin to cover the existing methods. A critical overview is given by Matthiessen in a recent review about methods, algorithms and tools in computational proteomics from a practical point of view (Matthiesen, 2007). It covers spectra interpretation (preprocessing, identification, and validation), quantitation, and data storage. Other reviews about this huge field of research have been written by Palagi *et al.* (2006) and Lisacek *et al.* (2006) with a focus on tools for proteomics and comparative proteomics respectively.

This work aims at the improvement of a method for protein quantification. Therefore, the focus of this chapter is on computational methods for quantitative proteomics.

2.5.1. Identification of proteins with MS — qualitative proteomics

For now, let us consider the MS device a black box that produces a list of masses. With a protein as input, the output would be only a single mass, which is not enough to unambiguously identify a protein. Therefore, proteins are commonly digested with a proteolytic enzyme, most often trypsin, resulting in a mixture of peptides. If we understand a protein as a string over the alphabet $\mathcal{A}$ of amino acids, the resulting peptides should ideally be non-overlapping substrings of the protein's sequence. In practice, some cleavage sites are missed, resulting in longer substrings that overlap in some cases. With the peptide mix as input, the MS generates a list of corresponding masses called *peptide mass fingerprint* (PMF) which is used as a pattern to identify the protein or proteins in the sample from a database of known proteins. Database search engines specialized to this task simulate the digestion process and generate theoretical PMFs from a given database. These are compared to the query PMF and return a rated list of candidate proteins. Nowadays, search engines take common post-translational modifications and incomplete tryptic digestion into account. The best-matching proteins, the list of modified or unmodified peptides and what mass they have been matched to, together with scores per peptide and protein, and the *sequence coverage* are returned as results.

The PMF approach only works up to a certain complexity of the sample. If there are a lot of proteins in the input sample, the above-mentioned procedure is not feasible anymore. Protein separation techniques are commonly applied prior to MS analysis to decrease the samples complexity. Tandem MS (see Section 2.2.4) has been established to facilitate identification of proteins in complex biological samples. Search engines for identification via MS/MS compare the *peptide fragmentation fingerprint* (PFF)[3] to predicted fragmentation patterns.

[3] peaks extracted from the MS^2 spectrum

There are theories about the mechanism of peptide fragmentation (Dongre *et al.*, 1996) that allow prediction (Schütz *et al.*, 2003) and de novo sequencing from fragmentation spectra (Han *et al.*, 2004; Liu *et al.*, 2006; Lu and Chen, 2004; Ma *et al.*, 2005; Syka *et al.*, 2004). These sequences are used to exclude proteins from the space of possible identification results, making identification of proteins in very complex samples feasible.

Lots of effort is put into the development of meaningful scores to rate the reliability of the identification. Usually, these scores are not real probabilities for the correct identification, but merely heuristic. There is a lot of literature proposing new scoring methods based on statistics (Kaltenbach *et al.*, 2007; Nesvizhskii *et al.*, 2003; Wan *et al.*, 2006). Nonetheless, commercial identification software dominates the field because it is bundled with the MS device by the manufacturer. Often, external methods are used to estimate or limit the false positive rate (FPR), for example by using a decoy database containing scrambled entries. The best results are achieved when mixing the decoy entries with the target database entries (Elias and Gygi, 2007; Elias *et al.*, 2005; Peng *et al.*, 2003). Reidegeld *et al.* (2008) present a tool to create such a database.

I refer to Chamrad *et al.* (2004) as well as Shadforth *et al.* (2005) for an overview of commonly used search engines.

In this work, the Mascot PMF and PFF (version 2.104) search engine is used to identify proteins (Pappin *et al.*, 1993). Mascot is a widely used identification database search engine.

2.5.2. Quantitative MS

It has become obvious that it is not sufficient to know which proteins are present in a cell to describe its state, which may enable us to differentiate between normal and abnormal cells. We also have to know how abundant each protein is.

Before quantitative MS began to emerge, quantification was achieved using fluorescence tags, dyes, or radioactive markers with good sensitivity. But they cannot identify proteins at the same time and are only applicable to highly abundant proteins. With MS, it is now possible to achieve identification and quantification from the same sample in a high-throughput manner.

Often, the coefficient of variation (CV) is calculated as a reproducibility measure using the mean value μ and the standard deviation σ of a distribution:

$$CV = 100 \, \frac{\sigma}{\mu} \tag{2.3}$$

It allows comparison of the deviation within a distributions with different units or different means. However, the CV is very sensitive to small changes if the mean is near zero.

MS has been used as a high-throughput technique for protein identification for some years now. However, making it work quantitatively is not straight-forward, because the measuring sensitivity differs between different types of molecules. The reason is in its working principle: Molecules have to be ionized and brought into gas phase to be detected. The detection efficiency depends on various factors, whose influence and interdependence are mostly unknown for large molecules such as peptides. Therefore, peak intensities are not a function of only the peptides' abundances.

2.5.3. Peak intensities

Abundances of different peptides cannot be derived by simply comparing their peak intensities, not even within a single spectrum or between runs with the same mixture. In addition, sometimes peptides are not detected at all, for example because they are below the noise level of the preprocessing method. Factors that influence peak intensities are numerous.

Some of them can be controlled to a degree and influence the whole peptide mixture: the sample preparation method, the analyte concentration in the matrix or the matrix substance itself (Gusev *et al.*, 1996), and settings of the MS device (Aresta *et al.*, 2008). Physicochemical properties of individual peptides as well as other substances (other peptides or contaminants) in the analyte mixture also influence peak intensities but cannot be arranged for obvious reasons. Such properties are secondary structure (Wenschuh *et al.*, 1998), tertiary structure (Winston and Fitzgerald, 1998), hydrophobicity (Breaux *et al.*, 2000; Schaller, 2000), the amino acid composition (Baumgart *et al.*, 2004; Olumee *et al.*, 1995), and their position in the sequence (Gonzalez *et al.*, 1996). It is known that basic residues in a molecule enhance its ionization efficiency (Zhu *et al.*, 1995).

In ESI MS, multiple charge states of peptides can be observed. Thus, for sufficiently large peptides only higher charged ions are within the mass range of the spectrometer. In analog to the ionization probability for singly charged ions in MALDI, there is a distribution of charge states for a peptide in ESI. Depending on their composition, different peptides have different charge distributions (Schnier *et al.*, 1996). The charge distribution also depends on the solvent (Iavarone *et al.*, 2000) and other parameters. Nielsen *et al.* (2004) analyzed the effect of *hydrophobicity*, *pI*, and molecular mass on detection probabilities of peptides for LC-ESI.

Nonetheless, peak intensities have been shown to be reproducible under carefully controlled conditions (MALDI-TOF: Jarman *et al.* (1999), LC-ESI-TOF and -ion trap: Wang *et al.* (2003)). For ESI, Wang *et al.* showed that although the effect of other substances in the mixture on a peptide's peak intensity (ion suppression effect) is more noticeable for a more complex mixture, intensities of individual peptides are linear with their abundance. Thus, quantification based on peak intensities is feasible. Anderle *et al.* (2004) investigated the noise behavior of processed LC-MS data, and found that for high intensities, a constant coefficient of variance (CV) is dominant, while Poisson-like variations can be found for low intensities. Their results indicate that for processed LC-MS data a constant coefficient of variation is dominant for high intensities, whereas a model for low intensities explains Poisson-like variations.

Peak intensities have been used successfully to enhance protein identification reliability (Elias *et al.*, 2004; Parker, 2002; Yang *et al.*, 2008).

The next section gives a compressed overview of available quantification techniques. For a more thorough discussion, the author recommends these reviews: Bantscheff *et al.* (2007); Ong and Mann (2005). Also, Sanz-Medel *et al.* (2008) wrote a review that focus more on the chemical aspects of quantitative MS.

2.5.4. Relative and absolute quantification

To overcome the difficulty peptide-specific measuring sensitivity poses for inter-peptide comparison, and to facilitate comparative protein quantification via MS, various techniques have been developed.

Labeling techniques The working principle of these is that an isotope label causes a mass shift but no change in measuring sensitivity. In theory, an isotope-labeled molecule is chemically identical to its unlabeled counterpart. The label is added to the peptides of one of the to-be-compared samples at some step during sample processing. By knowing the mass difference the label causes, the pairs (or even multiples) of peaks allow accurate relative quantitation between identical peptides of different samples. Relative quantitation of proteins is derived by averaging or taking the median of the corresponding peptides' ratios. The use of the mean or median is based on the assumption of log-normally distributed ratios. Boehm *et al.* (2007) recommend the use of linear regression instead for small datasets. Unfortunately, labeling methods are extremely expensive and time-consuming.

- **Stable isotope labeling** by amino acids in cell culture (SILAC, Ong *et al.* 2002) allows for up to three states to be compared by combining ^{15}N and ^{13}C labels that are incorporated into the sample metabolically. The advantage is the introduction as early as possible in the preparation pipeline, such that all errors that occur during sample preparation become systematic. Therefore, SILAC is one of the most accurate quantification techniques.

- **Enzymatic labeling** with ^{18}O (Yao *et al.*, 2001) during or after digestion is another way to label peptides if metabolic label is not possible and still avoid the complications that chemical labeling may cause. One drawback here is that only a part of all peptides are labeled, and different peptides are not labeled with the same efficiency, which causes difficulties when comparing abundances between different proteins.

- By **Chemical labeling** a isotope-containing group is added chemically to certain reactive groups of a peptide after digestion. Isotope-coded affinity tag (ICAT, Gygi *et al.* 1999) targets the reactive thio group of cysteine residues. Therefore, only cysteine-containing peptides can be tagged, greatly reducing the fraction of the sample that can be quantified, since cysteine is a rare amino acid.

  Isotope tags for relative and absolute quantification (iTRAQ, Ross *et al.* 2004) and a few other methods target the peptide's N-terminus and an amino group of lysine, covering a broader range of quantifiable peptides. Furthermore, iTRAQ allows to compare up to eight samples at once. A practical problem is that there might occur side reactions leading to incomplete labeling in some cases.

  Ross *et al.* (2004) presented an isobaric tagging reagent for multiplexed peptide quantitation with MALDI-MS/MS that enhances the ion signal intensities compared to ICAT. Wu *et al.* (2006) found the accuracy of ICAT and iTRAQ to be similar, but iTRAQ to be more sensitive. However, the information gained with both methods is complementary.

The major drawbacks of labeling techniques are the increased complexity, time, and cost requirements of the preparation, as well as the limited proteome coverage: Only about up to 50% of the proteome of a *monad* can be identified by MS, and much less can be quantified. Labeling further reduces the proteome coverage. Especially for large high-throughput comparative studies, labeling techniques are often too expensive.

Label-free quantification methods have the potential to achieve a better proteome coverage, have a higher linear dynamic range of measurable abundances (up to 3 orders of magnitude), and come for free: No expensive labeling reagents and additional preparation steps are necessary. Also, the number of samples that can be compared is not limited. However, due to the peptide-dependent measuring sensitivity of MS, this comes at the cost of a lower accuracy (Bantscheff *et al.*, 2007).

At the moment, there are two possibilities for label-free quantification. Either ion intensities are used directly and extracted from the spectra they appear in (extraction from the ion chromatogram (XIC)). Higher accuracy here (i.e. more MS^1 spectra) comes at the cost of less MS^2 spectra, i.e. less peptides can be identified in the same run. Or *spectral counts* (Liu *et al.*, 2004), the number of MS^2 spectra in which peptides of an analyzed protein occur in, are used instead. Spectral counting utilizes the observation that this number is correlated to the protein's abundance. Rappsilber *et al.* (2002) computed a protein abundance index (PAI) by dividing the number of observed peptides by the number of theoretically observable peptides for a given protein. They could improve the abundance estimation by using an exponentially modified PAI (emPAI) in a later study (Ishihama *et al.*, 2005). Recently, they utilized this approach to profile the abundance of *Escherichia coli* (Ishihama *et al.*, 2008). However, this approach assumes that all peptides have the same detection likelihood, which is not the case.

Peptide detectability prediction To correct for peptide-specific detection probabilities, Lu *et al.* (2007) developed absolute protein expression measurements (APEX), a spectral counting method that uses a prediction of detection probabilities based on the frequency of amino acids, length and molecular weight of peptides. The authors evaluate different classifiers for this purpose, and find that bagging with a forest of random decision trees produces the best results. The predicted values are then used to enhance the spectral counts-based, uncorrected abundance estimation by about 30 %. Old *et al.* (2005) evaluated spectral counting methods and state that "Peak intensity values useful for protein quantitation ranged from 10(7) to 10(11) counts with no obvious saturation effect, and proteins in replicate samples showed variations of less than 2-fold within the 95% range (+/-2sigma) when >or=3 peptides/protein were shared between samples. Protein ratios were determined with high confidence from spectral counts when maximum spectral counts were >or=4 spectra/protein, and replicates showed equivalent measurements well within 95% confidence limits."

Tang *et al.* (2006) introduced the concept of *peptide detectability*: the probability to observe a peptide in a standard sample analyzed by a standard proteomics routine. They classify peptides into detectable and undetectable ones with a *neural network* approach using the peptide's sequence and neighboring regions in the parent protein. They derive a minimum acceptable detectability for identified proteins (MDIP), a cutoff value that maximizes the sum of true positives and true negatives. The MDIP is shown to increase as protein abundance decreases, which according to the authors could be utilized to improve quantification.

Others have studied the prediction of peptide detectability values. Mallick *et al.* (2007) introduced the term *proteotypic* for peptides that can be detected in more than of the expected occurrence. They predict proteotypic peptides with an accuracy of up to 90% using a *Gaussian mixture model*. Features were selected using a *hierarchical hill climbing* algorithm out of a large variety of physicochemical properties derived from amino acid indices (Kawashima *et al.*, 1999). The authors classify peptides from four different MS setups (PAGE-MALDI, PAGE-ESI, LC-ESI and LC-ESI with ICAT labeling) with cumulative accuracies of up to 90%.

A few years earlier, Gay *et al.* (2002) already pursued a similar goal. They used amino acid frequencies, and some easily accessible properties such as *pI* and *hydrophobicity* on which they tested a number of different algorithms for classification into observed and unobserved peptides as well as regression of actual peak intensities. They used the same peptides in training and test set of the classifiers, therefor measuring the ability of the learning algorithms to reproduce intensities or detectability of peptides. However, their main focus was on the derivation of rules from a decision tree model.

Internal standards The above-mentioned quantification methods allow relative quantification between samples. To derive absolute quantification, the use of synthetic internal standards has been reported in the 90s already (Desiderio and Kai, 1983). Today, it is often applied for MS as a method called absolute quantification of proteins (AQUA, Gerber *et al.* (2003)).

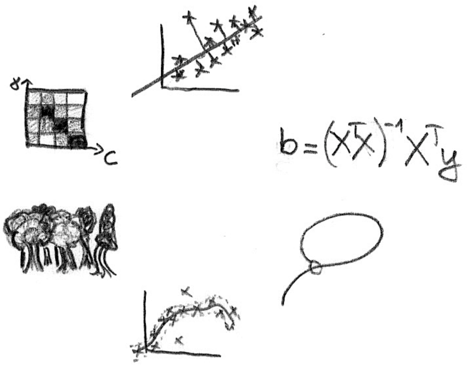
Today, peptides can be synthesized quite fast by designing a DNA sequence that is expressed into a target protein and digest it into peptides. Nonetheless, it is not easy to find a peptide as internal standard that is always detected, because it is infeasible to synthesize and screen every possible peptide. With twenty possible amino acids, even for short peptides the search space is much too large, especially since not only the frequency but also the order of peptides seems to play a role for the peptide-specific sensitivities. The prediction of proteotypic peptides by Mallick *et al.* (2007) helps to determine suitable peptides.

Disentanglement of terms: detectability, flyability, ionizability, ... There is no accepted term for the peptide-specific sensitivity of MS measurements. Tang *et al.* (2006) define "detectabilty" or "peptide detectability" as the probability that a certain peptide is observed at all. Another term in this context is "flyability". This refers to how prone a peptide is to be detected in the detector of a TOF device, i.e. proneness to fly in this mass analyzer. Another related term is "ionizability" which is the probability for a peptide to become ionized. This is a prerequisite to be detected in the first place. Both flyability and ionizability do not account for loss of peptides outside the MS itself. Flyability or ionizability are difficult to determine because we cannot run an MS without the preparation steps earlier in the workflow that also influence the final detectability.

Detectability in the sense Tang *et al.* use it, on the other hand, is not the same as the probability for a peptide to be detected in the detector of an instrument, as might be suspected. It denotes the probability to detect the peptide at all, including wet lab preprocessing of the sample and in silico post processing of the raw spectra, such as the parameter-dependent extraction of a list of peaks (*peak picking*). In other words, it denotes the probability that a peptide's peak is detected by a given peak detection software and can be identified.

3. Machine learning - methods and validation

Machine learning is a field of research that deals with algorithms for automatic knowledge discovery from data. Based on methods from classical statistics, it has developed its own advanced algorithms and tools. *Data mining* is a related field, which comprises any statistical or mathematical method for pattern recognition in data. As such, it also includes machine learning methods. In machine learning, a distinction is drawn between *unsupervised learning* and *supervised learning*.



Unsupervised learning methods build a model that describes the input data. It allows prediction, visualization of high-dimensional data, and structure discovery. Most unsupervised learning algorithms are *clustering methods*. These categorize data according to similarities and dissimilarities. Others, such as *principle component analysis* transform the data into another representation that allows to explore the structure within the data.

Supervised learning algorithms aim at learning a function that produces a correct output for a given input after a training phase. During training, a "teacher" or "supervisor" presents correct input-output pairs. Classification (e.g. hand writing recognition) as well as the estimation of a function with regression methods are supervised learning applications.

In this chapter, I give an introduction to supervised learning. Methods used in this work are presented and an introduction to feature selection is given. Common pitfalls are explained in Section 3.9. These might be helpful to understand the choice of input representations presented in Chapter 7. To understand the core parts of this chapter, knowledge of linear algebra is mandatory. There are a lot of text books dedicated to this topic, and also free resources on the internet. For a more thorough discussion of statistical learning, and more learning algorithms, see Vapnik (2000), Hastie *et al.* (2001), Bishop (2007), or Duda *et al.* (2000). The notation used here is explained in Section A.1 of the appendix.

3.1. Supervised learning

Often, we want to predict the response or output of a system to a given input. If we do not have enough previous knowledge about how this system works, we can observe its input and output and derive prediction rules from our observations by *learning from examples*. A whole class of learning algorithms are designed for this task. They can be *trained* by presenting a *training set* of N example observations consisting of pairs $\{x_i, y_i\}, i = 1, \ldots, N$ of inputs x_i and output or *target* values y_i. For each example, the learning algorithm calculates an estimated output value $\widehat{y}$, compares it to the target value and adapts itself to the differences $\widehat{y} - y$. In statistical learning theory this is called *supervised learning*. In classical statistics, the same problem would be formulated as a function approximation where a function $f(\mathbf{x})$ is fitted to d-dimensional points $(\mathbf{x}_i, y_i), i = 1, \ldots, n$ in Euclidean space. The aim of a learning algorithm is usually to either minimize a prediction error E for data not included in the training set or maximize the likelihood that the predicted data is emitted by the system under consideration. The exact definition of E differs from case to case.

When the training is completed, the algorithm should be able to predict not only examples from the training set, but also previously unseen examples from the same source, an ability called *generalization*. As in life, it is not enough to learn the training examples by heart (i.e. fit a function that follows every single training point) to be able to predict the output for new ones. If a function follows the training points too closely, leading to bad generalization, this is called *overfitting*. Overfitting can be controlled by inflicting a complexity restriction upon the prediction model. For example, support vector regression (see Section 3.3.1) directly enforces the smoothness of the model via a complexity term, for a polynomial fit, the degree of the polynomial controls complexity, the lasso (see Section 3.5) and other shrinkage methods have a parameter to constrain coefficients of their model, thus decreasing complexity. We will discuss model selection in Section 3.8. First, here is an introduction to the regression methods used in this work.

3.2. Linear regression

An old, yet still often useful regression model is the linear model. It makes the very restrictive assumption that the response of the system is at least approximately linear with respect to its input space.

A linear model (LM) has the following structure:

$$\widehat{\mathbf{y}} = \widehat{b}_0 + \sum_{j=1}^{d} \mathbf{x}_j \, \widehat{b}_j \tag{3.1}$$

with $\widehat{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$ and $j = 1, \dots, d$. The b_j are called *coefficients*, b_0 is called *intercept* or *bias*. The x_j are called *features* or *predictors* for the system output.

This can also be written as matrix equation:

$$\widehat{\mathbf{y}} = \mathbf{X}^T \mathbf{b}. \tag{3.2}$$

Here, $\mathbf{X}$ is a $N \times d$ matrix consisting of the columns $\mathbf{x}_j = (1, x_{1j}, \dots, x_{Nj})^T, j = 0, \dots, d$, where $\mathbf{x}_0$ is the unit vector for the intercept, and rows $\mathbf{x}_i = (x_{i1}, \dots, x_{id}), i = 1, \dots, N$. A column of $\mathbf{X}$ corresponds to a feature of the input of the system, while each row corresponds to an example point in $\mathbb{R}^d$.

A well-known algorithm to find the coefficients is *least squares*. It minimizes the error represented by *residual sums of squares* (RSS) – a sum of squared differences between estimated output and the response of the system over the training examples $i = 1, \dots, n$:

$$\text{RSS}(\mathbf{b}) = \sum_{i=1}^{N} (y_i - \widehat{y}_i)^2 = \sum_{i=1}^{N} \left(y_i - \widehat{b}_0 - \sum_{j=1}^{d} x_{ij} \widehat{b}_j \right)^2 \tag{3.3}$$

We can compute $\widehat{\mathbf{b}}$ by solving:

$$\widehat{\mathbf{b}} = \underset{\mathbf{b}}{\text{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \widehat{y}_i)^2 \right\} = \underset{\mathbf{b}}{\text{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \widehat{b}_0 - \sum_{j=1}^{d} x_{ij} b_j \right)^2 \right\} \tag{3.4}$$

RSS($\mathbf{b}$) is a convex function, hence the global minimum can be found by setting the first derivative to zero and solving with respect to $\mathbf{b} = (b_0, b_1, \dots, b_d)^T$.

Writing Equation (3.3) in matrix notation

$$\text{RSS}(\mathbf{b}) = (\mathbf{y} - \mathbf{X}b)^T (\mathbf{y} - \mathbf{X}b)$$

and solving

$$\frac{\partial \text{RSS}(\mathbf{b})}{\partial \mathbf{b}} = 0$$

allows us to express $\widehat{\mathbf{b}}$ as

$$\widehat{b} = \left(X^T X\right)^{-1} X^T y \tag{3.5}$$

This solution is linear in y. If the columns of X are not linearly independent, i.e. there are redundancies in the columns of X, $(X^T X)^{-1}$ is singular and the solution to Equation (3.5) is not uniquely defined. Most implementations of regression algorithms detect and resolves these redundancies automatically.

In the final model, the absolute values of the coefficients indicate the importance of their corresponding features. Strictly speaking, this is only the case if all example observations are independent, if there are no correlations between any of the features, and if the linear model is appropriate for the system's response behavior in the first place. However, in reality, systems often are not linear. Therefore, the coefficients give only a rough indication of the feature's importance in these cases.

3.2.1. Properties

Least squares is fast and easy as well as easily interpretable via the coefficients. No parameters have to be chosen. However, this model assumes linearity of the data, which is a very restrictive assumption. If this assumption does not hold for the system under consideration, the LM has a large prediction error. It is sensitive to outliers and does not perform well in high dimensions.

3.2.2. Implementations

Pretty much any statistical toolkit contains the least squares algorithm. In this work, the function `lm` of the statistical toolkit R (R Development Core Team 2006) is used.

3.3. Support vector machines

Support vector machines (SVM) are a group of learning algorithms for classification that are designed to minimize the generalization error by finding a separating hyper-plane $\boldsymbol{wx} - b = 0$ that maximizes the margin between both classes (see Fig. 3.1). Here, $\boldsymbol{w}$ is a vector of coefficients and b the intercept that define the hyper-plane. With final model, examples are classified according to which side of the separating hyper-plane they are on.

For this purpose, two planes $\boldsymbol{wx} - b = 1$ and $\boldsymbol{wx} - b = -1$ (dashed lines in Fig. 3.1) have to be found. The margin's width is $\gamma = 2/\left\|\boldsymbol{w}\right\|_2$.

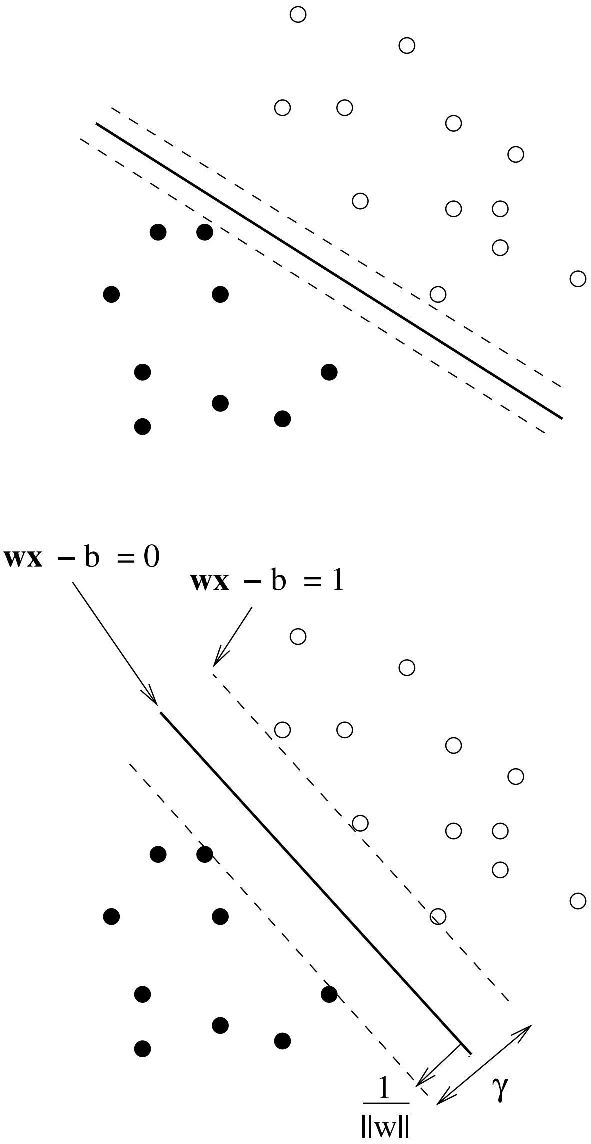


Figure 3.1: Separation of two classes using different separation planes. The black line denotes the linear separation hyper-plane $\boldsymbol{w}\boldsymbol{x} - b = 0$, which is a line in the two-dimensional case. The dotted lines denote the planes $\boldsymbol{w}\boldsymbol{x} - b = 1$ and $\boldsymbol{w}\boldsymbol{x} - b = -1$ that are parallel to the separating hyper-plane and delimit the margin that separates one class from the other. The dots depict example data points from two classes (black and white).
In both cases, the hyper-plane separates both classes perfectly. But the hyper-plane in the example above would have an increased risk of misclassifying new points of data because it is very near to points of both classes. The misclassification risk is lower for the lower example. By maximizing the margin $\gamma = \frac{2}{\|\boldsymbol{w}\|_2}$, the classification error for data not in the training set (generalization error) is minimized. The data points lying at the border of the margin are called *support vectors*.

This leads to a minimization problem

$$\left(\widehat{w}, \widehat{b}\right) = \underset{w,b}{\operatorname{argmin}} \left(\frac{1}{2}\|w\|_2^2\right), \tag{3.6}$$

subject to

$$y_i\left(wx_i - b\right) \geq 1, \tag{3.7}$$

whose solution depends solely on the support vectors, i.e the data points, that reside at the border of the margin.

SVM classification details if classes are not separable by a linear function, the constraints have to be relaxed a little. If data points are inside the margin they are penalized with an error. So the SVM has to maximize the margin and at the same time minimizing the error. For this purpose, slack variables ξ_i are introduced which measure how far into the "wrong" side of the margin data points x_i lie. So Equation (3.6) is extended to

$$\left(\widehat{w}, \widehat{b}, \widehat{\xi}\right) = \underset{w,b,\xi}{\operatorname{argmin}} \left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i\right), \tag{3.8}$$

subject to

$$\begin{aligned} y_i\left(wx_i - b\right) + \xi_i &\geq 1, \\ \xi_i &\geq 0, \end{aligned} \tag{3.9}$$

where C has to be chosen to regulate the tradeoff between the minimization of the training error and the maximization of the margin.

To solve such an optimization problem, usually algorithms use its dual problem:

In the general case, an optimization problem can be written

$$\min_{\mathbf{u}} f(\mathbf{u}),$$

subject to constraints

$$g_i(\mathbf{u}) \leq 0 \quad \text{for } i \in \{1, \ldots, k\},$$

$$h_i(\mathbf{u}) = 0 \quad \text{for } i \in \{1, \ldots, m\}.$$

The generalized Lagrangian function of this is

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{u}) + \sum_{i=1}^{k} \alpha_i g_i(\mathbf{u}) + \sum_{i=1}^{m} \beta_i h_i(\mathbf{u})$$

Here, the dual function is defined as $F(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \inf_{\mathbf{u}} L(\mathbf{u}, \boldsymbol{\alpha}, \boldsymbol{\beta})$. In some cases, F can be retrieved by solving

$$\frac{\partial \mathcal{L}(\mathbf{u}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial u_i} = 0$$

for u_i, and substituting it in $L(\mathbf{u}, \boldsymbol{\alpha}, \boldsymbol{\beta})$. Then the dual problem is solved by optimizing

$$\max_{\alpha, \beta} F(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \inf_{\mathbf{u}} L(\mathbf{u}, \boldsymbol{\alpha}, \boldsymbol{\beta}),$$

$$\text{subject to} \qquad \alpha_i \geq 0 \quad \text{for } i \in \{1, \ldots, N\}.$$

This is applied to the minimization problem at hand for the linearly separable case, yielding

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} + \sum_{i=1}^{N} \alpha_i \left[1 - y_i \left(\boldsymbol{w}^T \mathbf{x}_i + b\right)\right].$$

The dual optimization problem after derivation and substitution can be formulated as

$$\max_{\alpha_i, \alpha_j} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \, y_i y_j \, \mathbf{x}_i^T \mathbf{x}_j,$$

subject to

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, i \in \{1, \ldots, N\}.$$

The addition of the regularization term $C \sum_{i=1}^{N} \xi_i$ extends the above dual form only by constraining the α_i from growing too large: $C \geq \alpha_i \geq 0$.

The dual problem is usually solved by Quadratic Programming. The intercept b can be found using the support vectors. The classification can then be done by calculating

$$f(x) = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i \mathbf{x}\mathbf{x}_i - b\right). \tag{3.10}$$

In most cases classes cannot be separated in the vector space that is spanned by the feature vectors. Transforming the data into a even higher dimensional data space with a function φ makes it possible to execute the separation in that data space: $\varphi : \mathbb{R}^n \to \mathbb{R}^m$, $m > n$. A separation is possible for almost all cases, if the dimension of the vector space is high enough. Now the scalar product of $\mathbf{x}\mathbf{x}_i$ is considered in the $\mathbb{R}^m$ instead of the $\mathbb{R}^n$. Of course, it is not feasible to really calculate $\varphi(\mathbf{x})$. Therefore, the inner product is evaluated by a *kernel function*, while φ itself is not even known:

$$\varphi(u) \cdot \varphi(v) \equiv K(u, v) \tag{3.11}$$

Classification is then done by calculating

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) - b\right). \tag{3.12}$$

A commonly used kernel function is the radial basis function[1] $e^{-\gamma \|x_i - x_j\|^2}$, but the great advantage of kernel methods, such as SVM, is that they are modular: Kernels can be replaced without changing the whole method.

There are also multi-class SVMs that separate multiple classes (Hsu and Lin 2002). Hastie *et al.* (2004) propose an algorithm to solve the whole regularization path for SVM, which is implemented in the `svmpath` packet for the statistical toolkit R (R Development Core Team, 2006).

3.3.1. SVM for regression

To get from SVM to SVM for *regression* (SVR), the ε-insensitive loss function $|y - f(\mathbf{x})|_\varepsilon = \max\{0, |y - f(\mathbf{x})| - \varepsilon\}$ was introduced by Vapnik (1995). The basic idea is to put a tube with radius ε around the regression function. Data points are allowed to lie inside this tube without producing an error, i.e. errors only if they are higher than an $\varepsilon > 0$ chosen a priori.

[1] This is a different γ than the margin size but used here for traditional reasons.

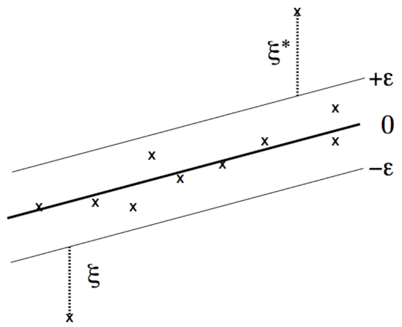


Figure 3.2: Principle of support vector regression, shown for a linear example. Data points that lie inside a tube defined by ε do not count towards the total of the error term in the optimization function. The other datapoints are accounted for with their distance ξ or ξ^* from the tube borders. Image from Schölkopf *et al.* (1999).

Since the choice of ε can be difficult, the ν-SVR introduced by Schölkopf *et al.* finds the best ε automatically by minimizing a cost function. Only ν, an upper bound of the number of errors allowed and a lower bound to the number of support vectors, has to be chosen *a priori*. The ν-SVR generalizes an estimator for the mean of a random variable which throws away the largest and smallest examples (a fraction of at most $\nu/2$ of either category), and estimates the mean by taking the average of the two extremal ones of the remaining examples (Schölkopf *et al.*, 1999). Because of this property, ν-SVR show good robustness against outliers.

Given a data set with training examples $\{x_i, y_i\}$, $i \in \{1, \dots, N\}$ and $x_i \in \mathbb{R}^n$, the primary optimization problem for a regression ν-support vector machine is

$$\left(\widehat{\boldsymbol{w}}, \widehat{b}, \widehat{\xi}, \widehat{\xi}^*\right) = \underset{\boldsymbol{w}, b, \xi, \xi^*}{\operatorname{argmin}} \left(\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\left(\nu\varepsilon + \frac{1}{N}\sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right)\right)\right) \tag{3.13}$$

subject to

$$\left(\boldsymbol{w}^T\varphi\left(\boldsymbol{x}_i\right) + b\right) - y_i \leq \varepsilon + \xi_i, \tag{3.14}$$

$$y_i - \left(\boldsymbol{w}^T\varphi\left(\boldsymbol{x}_i\right) + b\right) \leq \varepsilon + \xi_i^*, \tag{3.15}$$

$$\xi_i, \xi_i^* \geq 0,\ i \in \{1, \dots, N\},\ \varepsilon \geq 0. \tag{3.16}$$

Here, $\nu \in [0, 1]$, C is the regularization parameter again, and $\boldsymbol{x}_i$ are mapped into higher dimensional space by φ. The parameters ξ and ξ^* denote errors to both sides of the regression function respectively. If $\boldsymbol{w}^T\varphi\left(\boldsymbol{x}\right)$ is in the range of $y \pm \varepsilon$, thus being inside a tube in a distance of ε around the function, it does not count towards the total error. The principle is shown in Fig. 3.2.

3.3.2. Properties of SVR

The unique design endows it with good generalization characteristics. It finds global optima independently of the optimization algorithm used, and has only few parameters to tune. Because the error is determined by the ε-insensitive loss function, SVR are robust to noise. The kernel trick allows it to deal with very high-dimensional data. They can pick out meaningful descriptors from the data and ignore redundant or irrelevant ones to some degree.

On the other hand, parameter search can take quite long, training times depend on the specific problem and cannot be predicted easily. The size of the model grows with the size and dimensionality of the training data. The SVR model is a black box: It is very inconvenient to get insight into how the input data influences the model.

3.3.3. Further reading and implementations

For a practical guide, see Burges (1998) or Hsu *et al.* (2003). Gunn (1998) wrote a compact report on Support Vector machines in general that gives a good overview. SVMs is a kernel method. A whole, relatively young field of research deals with kernel methods. A good starting point to learn about literature and the basics is `http://www.kernel-machines.org`, a website managed by an editorial board.

A widely used and well-maintained SVM library is `libsvm` (Chang and Lin, 2001). In this work, the `libsvm` interface of the e1071 package available for R is used (Dimitriadou *et al.*, 2006; R Development Core Team, 2006). Other interfaces are listed at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

3.4. Random forests

Random forests are a group of methods for classification and regression that use *bagging*: They build a number n of trees, each of which delivers a prediction or classification. Single trees are often unstable, because a slight change in the data might lead to a totally different tree. Bagging (Breiman, 1996) overcomes this issue by averaging many trees, and grow the trees from random subsets of the training data (bootstrap sample). Random forests consist of a collection of trees that use independent identically distributed random vectors θ_t of integers for each tree $t \in \{1, \ldots, n\}$, where the integers correspond to the choice of a feature subset of size m from the full set. The predicted output is calculated by taking a majority vote (classification) or the mean value over the output (regression) of a number n of the trees.

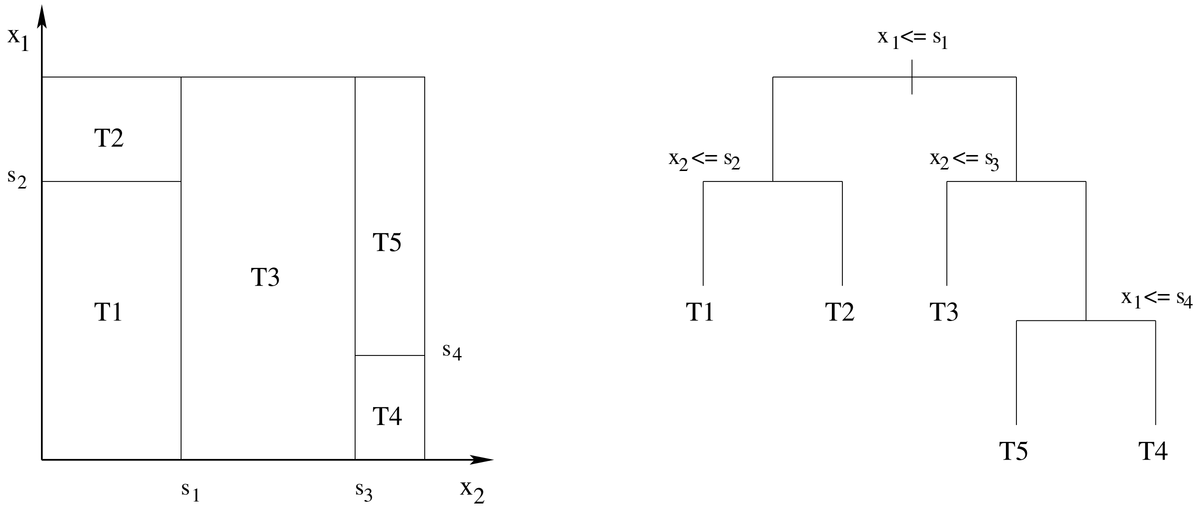


Figure 3.3: Feature space partition by binary recursive splitting for the two-dimensional case. (*left*: feature space visualization, *right*: tree visualization)

For most of these methods, single trees are constructed by binary recursive splitting of the feature space into partitions (see Fig. 3.3). There exist various methods that use different ways to inject some randomness into each tree, as discussed in Breiman (2001). According to Breiman's discussion, there is a type of forests that performs well compared to other bagging methods while being faster and simple to use. It selects a set θ_k of features randomly at each node k to grow each tree. The training set and θ_k is used to determine the splitting point s_k for the current node. The best splitting point s_k is determined by searching through all variables determined by θ_k. Splitting commences as long as the node size is at least z, and no pruning (removal of nodes) is done afterwards. The prediction for each partition is done separately with a simple model (for example the average).

Although this method is quite simple, it has a few desirable properties. Apart from typical properties of bagging methods, namely robustness to outliers and noise, the discussed type of random forest is fast, simple to use, and easily parallelized. For bagging methods, the generalization error can be estimated in a special way: For each training example $\kappa_i = (x_i, y_i)$, only the votes of those trees for which the bootstrap sample does not contain κ_i are used. The remaining trees are called *out-of-bag* classifiers. The out-of-bag estimate for the generalization error is the error of the out-of-bag trees measured on the training set. According to Breiman (2001), the out-of-bag estimates are unbiased, i.e. this type of random forest does not overfit. Although random forests have been shown to work well in practical applications, the statistical mechanism is still subject of active research.

3.4.1. Further reading and implementations

The "Manual On Setting Up, Using, And Understanding Random Forests" describes training and data mining with random forests (Breiman, 2002a,b). The newer guide (V4.0) contains more additional information about dealing with missing values, while V3.1 contains information about variable importance assessment, which are missing in V4.0. There is also a corresponding paper published in the Machine Learning journal (Breiman, 2001). Basic theory and other tree-based methods can be found in Hastie *et al.* (2001). Lin and Jeon analyze the properties of random forests and adaptive nearest neighbor in a technical report (Lin and Jeon, 2002).

The original random forest implementation by Breiman and Cutler is written in `Fortran77`. The R package `randomForest` implementation is based on their code. Liaw and Wiener (2002) explain the use of the `randomForest` package. Variable importance assessment is implemented in this package. The mean squared error of the out-of-bag data is computed for each tree, then the same is computed after permuting the variable whose importance is determined. The differences are averaged and normalized by the *standard error*. In addition, proximity between input vectors can be used for unsupervised learning and outlier detection.

3.5. Shrinkage methods

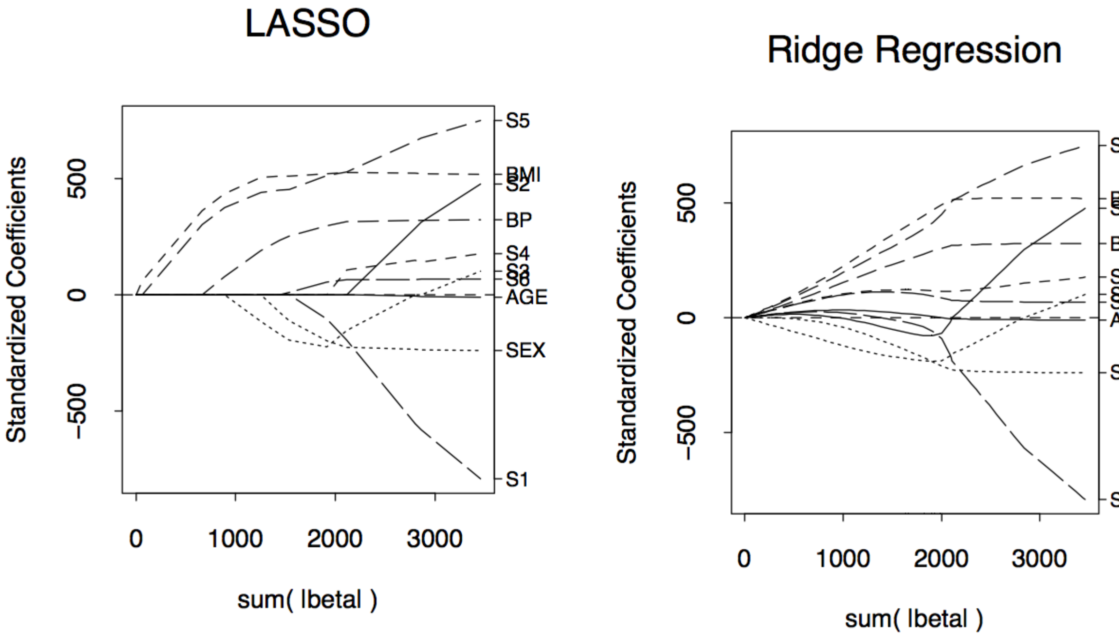
One possibility to avoid overfitting is to constrain the complexity of the approximation function. In equation 3.4, we wrote the least squares method as an optimization problem. If we impose a restriction $f(\mathbf{b}) <= t$ on the coefficients $\mathbf{b}$ that constrain the overall size of the coefficients, this forces a lower complexity of the resulting model, while at the same time minimizing the residual sum of squares:

$$\widehat{\mathbf{b}} = \underset{\mathbf{b}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \widehat{y}_i)^2 \right\} = \underset{\mathbf{b}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \widehat{b}_0 - \sum_{j=1}^{d} x_{ij} b_j \right)^2 \right\} \tag{3.17}$$

$$\text{subject to} \quad f(\mathbf{b}) <= t \tag{3.18}$$

This problem is equivalent to

$$\widehat{\mathbf{b}} = \underset{\mathbf{b}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \widehat{y}_i)^2 = \sum_{i=1}^{N} \left(y_i - \widehat{b}_0 - \sum_{j=1}^{d} x_{ij} b_j \right)^2 \right\} + \lambda \cdot f(\mathbf{b}), \tag{3.19}$$



(a) Lasso

(b) Ridge Regression

Figure 3.4: Comparison between lasso and ridge regression coefficients. Lasso adapts the nonzero coefficients while ridge regression shrinks coefficients proportional to their absolute value. Image from Hesterberg *et al.* (2008).

where λ controls the impact of the additive *regularization* term on the model. *Ridge regression* (Draper and Smith, 1998; Miller, 2002) for example imposes an L_2 penalty term $f(\mathbf{b}) = \|\mathbf{b}\|_2 = \sum_{i=1}^{d} (b_i)^2$. *The lasso* (Tishirani, 1996) uses an L_1 penalty $f(\mathbf{b}) = \|\mathbf{b}\|_1 = \sum_{i=1}^{d} |b_i|$.

Both of these problems have the same solution for the extreme values of λ: If $\lambda = 0$, the solution corresponds to the least squares solution, because the penalty term disappears. If λ is very large, the coefficients are shrunken to zero. The regularization parameter λ can be increased step by step from zero to a large value that sets all coefficients to zero. The resulting solutions $\widehat{\mathbf{b}}_\lambda$ are called the *regularization path* of the method. This path differs between ridge regression and LASSO, as shown in Fig. 3.4. While ridge regression shrinks coefficients in proportion to their magnitude, LASSO shrinks them to zero one by one, such that a set of active (i.e. non-zero) coefficients is determined for a given λ. Since the coefficients give a rough indication of feature importance, the active set of lasso coefficients constitute a feature selection.

3.6. Feature subset selection

Why is feature selection important? In most cases, it is not exactly known which features are important for modeling of a system's response. Additional or redundant features in a feature set might introduce noise and additional dimensions which worsens prediction results. By choosing only the most important features, accuracy is improved. Another reason is transparency: Even if the learning algorithm could ignore unimportant features to some degree, we often want to know which features mainly influence the output of the system under consideration. Lastly, it saves computational time if a smaller set of features is used.

Generally, the training error $E = \frac{1}{N}\sum_{i=1}^{n} L\left(y_i, \widehat{f}(x_i)\right)$, where L is a loss function that measures the error for individual training examples, decreases more and more as we add features or increase the model's complexity. At some point, the model begins to overfit, and the prediction or generalization error increases. This is visualized and explained more closely later in this chapter. Thus, the training error is not a good criterion to choose the best subset. Instead, the generalization error, i.e. prediction error on test data from the same distribution as the training data, has to be minimized.

There might be combinations of features that can replace a single feature or the other way round. Thus, it is not enough to test each single feature for its relevance to the output, but we have to look at combinations of features. Also, unimportant features might worsen the prediction performance. The naive approach to subset selection would be to determine the performance of every possible subset and take the best one. *Best subset regression* finds subsets of size $k \in \{1, 2, \ldots, d\}$ with the smallest residual sum of squares. However, with a lot of features, it becomes infeasible to go through all possible subsets. Also, the best subset on the training data usually is not the best one on new data from the same source whose true distribution is unknown.

Feature selection for regression problems should be accurate, interpretable, stable, and the selected features should generalize well to new data. Methods such as stepwise selection, best subset regression, and ridge regression cannot fulfill all of these criteria. More recent methods like forward stagewise regression (Hastie *et al.*, 2001) and the lasso are more stable and give better prediction accuracy, but can be slow. Least angle regression (LARS, Efron *et al.* 2004) potentially solves all these issues and is fast.

In the following, forward stepwise selection, and LARS which is related to both stagewise regression and the lasso, are explained.

3.6.1. Forward stepwise selection

Forward stepwise selection searches for a suboptimal subset of the entire feature set by successively adding features to an empty set which lead to an increase in performance. The method starts with the intercept and iteratively adds features which decrease the residual sums of squares the most, if the change is significant. To assess significance, an F statistics-based measure can is used:

$$F_t = \frac{RSS_t - RSS_{(t+1)}}{RSS_{(t+1)}/(N-k-2)}, \tag{3.20}$$

where t denotes the current step with a model with k predictors. When we add a predictor at step $t+1$, the above function is evaluated for every possible remaining predictor. The one with the highest value is added if it is above the 90th or 95th percentile of the $F_{1,N-k-2}$ distribution. This procedure is also described in Hastie *et al.* (2001).

Forward stepwise selection often does not find the best subset but a suboptimal one, since each step minimizes the error only locally. The choice of the predictor to be added depends on predictors already in the set, which might not be optimal. *Backward stepwise selection* iteratively deletes predictors from the model containing all possible predictors, using a similar criterion and the F statistics, but can only be applied if $N > d$. There are hybrid techniques that choose to drop or add a predictor in each step.

3.6.2. Shrinkage methods for feature selection

Shrinkage methods with L_1 penalty such as the lasso shrink coefficients to zero. The active feature set for a given regularization parameter λ constitute a discrete feature selection. The coefficient values give a rough indication of feature importance.

These methods can be set into relation to forward stepwise selection: While in forward stepwise selection, selecting a feature makes its coefficient jump towards the least squares solution, *forward stagewise regression* changes the coefficient of the feature with the highest correlation to the target value only by a small amount with each step.

Least-angle regression (LARS) A relatively new method, LARS works similar to forward stagewise selection, but instead of making many small steps, one large step is computed that jumps towards the inclusion of the next feature in one step (Efron *et al.*, 2004). The first feature LARS chooses is the one with the smallest angle (i.e. correlation) between that feature and the response variable. This is illustrated in Fig. 3.5. The algorithm proceeds in that feature's

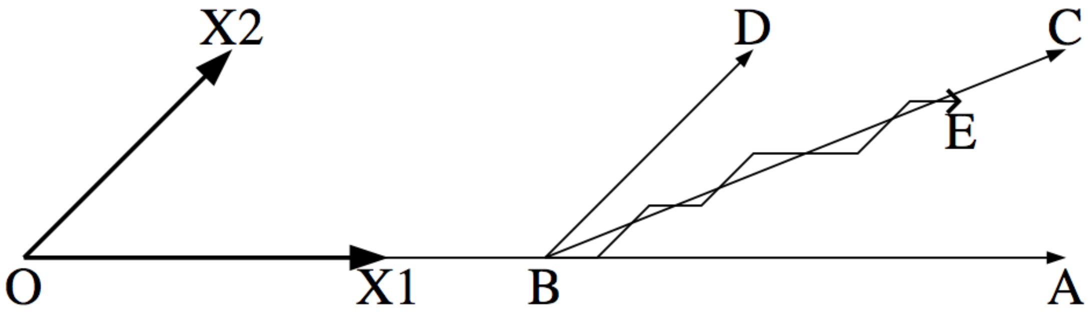


Figure 3.5: Comparison between least-angle regression (LARS, *left*) and forward stagewise regression (*right*) steps. This is a two-dimensional example with two features X1 and X2. The axes span the coefficient space of these two features. First, both algorithms go from 0 to B. Forward stagewise then includes X2 in the model and approaches the least squares fit C in many small steps, increasing the coefficients for X1 and X2 alternating. LARS jumps directly to C in one step. Image from Hesterberg *et al.* (2008).

direction in coefficient space until the correlation of another feature equals that of the first feature. Then it follows the direction of the least-squares fit based on both angles. A comparison between LARS and forward stagewise is shown in Fig. 3.5.

The following explanation of the LARS algorithm is a short summary of the basics in a very well-written review paper (that is still under review as of now) on least angle and L_1 regression (Hesterberg *et al.*, 2008). According to Hesterberg *et al.* (2008), LARS is remarkably fast. "The entire sequence of LARS steps with $p < n$ variables requires $O(p^3 + np^2)$ computations - the cost of a least squares fit on variables." (Efron *et al.*, 2004). With certain modifications, LARS can be used for a fast fit of the whole regularization path of the lasso and stagewise models. A C_p-type statistic constitutes a model selection criterion for this method which allows to choose from all the calculated models. It is based on a theorem in Efron *et al.* (2004) that states that the number of steps k is approximately the number of degrees of freedom. C_p is an unbiased estimator of the true generalization risk.

$$C_p = \frac{1}{\hat{\sigma}^2} \text{RSS} - n - 2k \tag{3.21}$$

where $\hat{\sigma}^2$ is the estimated residual variance, assuming that $n > p$. There have been discussions whether C_p is an appropriate model selection criterium for LARS (Ishwaran, 2004; Loubes and Massart, 2004; Stine, 2004).

Interestingly, all three methods (LARS, lasso and stagewise regression) have similar but in the general case not identical solutions, although they are based on different concepts: lasso is an L_1-regularized least-squares optimization, whereas stagewise regression is close to boosting algorithms. LARS, although derived from stagewise, has similarity to Newton's method (Hesterberg *et al.*, 2008).

L_1 **regularization path for generalized linear models (glmpath)** This method carries the idea of L_1-regularization over to generalized linear models (GLM). GLMs have been developed to unify multiple existing statistical models into one (Nelder and Wedderburn, 1972).

Therefore, the coefficients for this method are obtained by solving a set of non-linear equations with complex optimization techniques. The non-linear equations satisfy the maximum likelihood criterion

$$\widehat{\mathbf{b}} = \underset{\mathbf{b}}{\operatorname{argmax}} \left\{ L(\mathbf{y}|\mathbf{b}) \right\},$$

where L denotes the likelihood function with respect to the given data $(\mathbf{X}, \mathbf{y})$. In the L_1-penalized case, an additive penalty term $\lambda \cdot \|\mathbf{b}\|_1$ is added to the above optimization problem in analogue to the lasso (see Section 3.5). Unlike the lasso and the support vector regularization path, the GLM paths are not piecewise linear. The coefficients have to be computed exactly for values of λ that have to be chosen. Park and Hastie (2007) propose a way to compute the exact coefficients at the values of λ at which the set of non-zero coefficients changes, using the *predictor-corrector* algorithm (Algower and Georg, 1990; Garcia and Zangwill, 1981).

3.6.3. Implementations

The R package `lars` provides an interface to LARS. The beta version of the generalized least-angle regression `glars` is still under development to provide a numerically more stable version of LARS. It has been provided to the author by Tim Hesterberg for testing and also contains the `glmpath` method.

3.7. The two-sample t-test

A two-sample t-test is a method to find out if the mean of values drawn from two sources are significantly different. Of course, the difference between the mean of two samples is almost always non-zero. This does not necessary mean that this is the case for the source of the two samples. There are different versions of this test, depending on the assumptions imposed on the samples. It always assumes that the samples are normally distributed. This can be tested for example with a Kolmogorov-Smirnov test. Other assumptions are identical variance and independent samples. There are also variants which do not impose the latter two assumptions, but these are not discussed here. If samples are not normally distributed, there are non-parametric tests instead, such as Mann-Whitney U test for independent samples, or the binomial test or the Wilcoxon signed-rank test for dependent samples.

The test statistic of the two-sample t-test follows a Student's t-distribution. The H_0 hypothesis for this test is that the means differ. From the Student's t-distribution, a cutoff p-value can be found up to which the H_0 hypothesis is accepted. This p-value represents the probability that t could be that large by chance. For independent samples with identical variance and different sample sizes, t is calculated as

$$t = \frac{\overline{x_1} - \overline{x_2}}{s_{\overline{x_1} - \overline{x_2}}}, \qquad \text{where} \tag{3.22}$$

$$s_{\overline{x_1} - \overline{x_2}} = \sqrt{\frac{(n_1 - 1)\ s_1^2 + (n_2 - 1)\ s_2^2}{n_1 + n_2 - 2}} \tag{3.23}$$

Here, $\overline{x}$ is the estimated mean, and s_2 the variance of the samples, n the sample size, and subscripts 1 and 2 denote the two groups.

3.8. Model evaluation

Most learning algorithms have parameters to be set by the user, and often we do not know how to set these. Therefore, a number of possible parameters or parameter sets have to be evaluated. Various methods exist to estimate the generalization error for a model in the presence of only a fraction of the possible data our system emits, and without the knowledge of the true data distribution.

In a perfect world, there would be a huge amount of representative data from the data source whose output is to be predicted. Ideally, the choice of the best parameters then commences on the training set, a representative portion of these data. Another representative portion called *validation set* that is independent from the training set would be used to choose the optimal model, and another portion called *test set* is used to predict output values with the chosen model. The prediction error achieved on the *test set* is a good estimation of the generalization error. It is important that this test set did not influence the training procedure or choice of the model in any way, else we would underestimate the generalization error. Unfortunately,

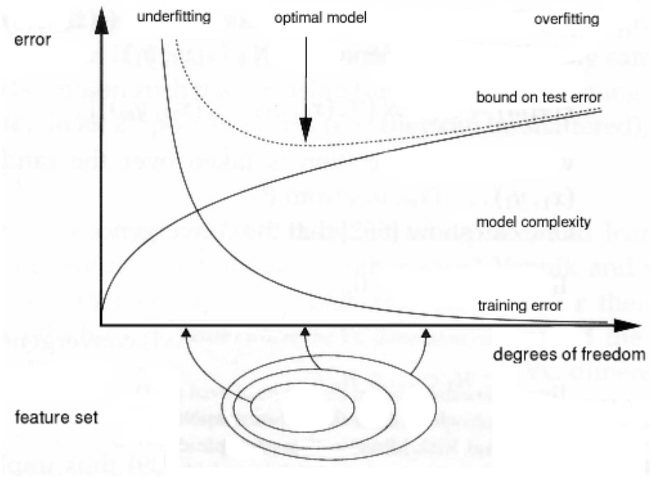


Figure 3.6: This illustrates the trade-off between bias and variance in statistical learning. A more complex function can fit the training data more closely, and thus has a low bias, but a high variance in regard to test data from the same distribution. The optimal model yields the lowest prediction error on the test data while bias is traded off against variance.

data is often limited, which implies that there is not enough data for three representative datasets.

To determine the optimal model, there are ways to reuse data (cross-validation, bootstrap) or assess an estimate of the generalization error analytically. Structural risk minimization (SRM, Vapnik 1982), the C_p statistics (Hastie *et al.*, 2001), the Bayesian information criterion (BIC, Schwarz 1978), or the Akaike information criterion (AIC, Akaike 1973) are examples for analytical solutions to this problem.

3.8.1. Structural risk minimization

(SRM, Vapnik 1982) is a principle for model selection that optimizes the trade-off between the quality of the fit to the training data (training error) and the model complexity. The underlying motivation is that a very complex model can achieve a very small training error but will generalize poorly to new data. The theoretical foundation for this observation is called *bias-variance trade-off*: The expected mean squared error (MSE) for application of a model to arbitrary many points from an unknown function can be shown to be constituted of additive terms for the variance of the noise, the bias (mean deviation from the training data), and the variance of the predicted values. The variance of the noise is a data-dependent constant. Thus, bias and variance of the predicted values have to be minimized at the same time to minimize the expected MSE. We can set the variance to zero by always predicting a constant value, which results in a high bias. Or we can fit the model perfectly to the data, which results in a low bias, but a variance equal to that of the training data. The SRM principle finds the optimal trade-off by minimizing the complexity of the model while at the same time minimizing the empirical error (see Fig. 3.6). SVMs follow this principle. It can be used for feature selection as well if there is a measure for the model complexity.

3.8.2. Cross-validation

If there is not enough data available to apply an independent test set, cross-validation is a way to make best use of the available data in training *and* to estimate the generalization error. In cross-validation, the data set is divided into n portions. The learning algorithm is trained n times, with each of the portions left out in one of the runs to be used as test set. The best choice of n here depends on the behavior of the learning algorithm with respect to training set size. If the size of the data set used for training becomes too small (n small), we overestimate the generalization error. If n is identical to the number of training samples, the training samples are very similar to each other, which might lead to high variance. Also, the computational resources increase with larger n. Usually, 5- or 10-fold cross-validation is used. After assessing the optimal parameters, the whole training data is used to build the final model.

Validation measures are necessary to pick the best model using prediction results obtained in cross-validation:

The squared Pearson's correlation (r^2) measures the strength and direction of a linear relationship between two random variables x and y.

$$r^2_{xy} = 1 - \frac{s^2_{y|x}}{s^2_y}, \tag{3.24}$$

where $s^2_{y|x}$ is the square error of a linear regression $y = a + bx$:

$$s^2_{y|x} = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - a - bx_i)^2 \tag{3.25}$$

and s^2_y the variance of y:

$$s^2_y = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})^2 \tag{3.26}$$

The root mean squared error ($RMSE$) measures the cumulative deviation of the data points (x_i, y_i) from the diagonal in a scatterplot between x and y.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.27}$$

3.8.3. The Bayesian and Akaike information criteria

The AIC and BIC are statistical criteria for model selection. They maximize the likelihood for the evaluated model to explain the true data source distribution with only a sample being available.

The AIC is similar to the C_p but can be applied more generally.

$$\mathrm{AIC} = -\frac{2}{n}\ \mathrm{loglik} + 2\ \frac{D_f}{n} = \mathrm{RSS} + 2\ \frac{k}{n}\ \widehat{\sigma}^2 \tag{3.28}$$

for a sample of size n and D_f the number degrees of freedom. This is often k: the number of predictors plus one for the intercept. For more complex models, it has to be replaced with an estimate of the model complexity. The log-likelihood loglik is a term to describe the likelihood that the model explains the data. For a Gaussian distribution, -2 loglik matches the residual sum of squares RSS. Another loss function summed over the training data can be used where it is appropriate. The AIC for different models provides an estimate of the generalization error curve. The optimal model according to this criterion is the one with the minimum AIC value.

The BIC is also known as the Schwartz criterion. For $n > e^2 \approx 7.4$, it penalizes the addition of new features more than the AIC, prefering simpler models. It can be written as

$$\mathrm{BIC} = -2\ \mathrm{loglik} + (\log n)\ D_f = \frac{n}{\widehat{\sigma}^2}\left(\mathrm{RSS} + (\log n)\ \frac{k}{n}\widehat{\sigma}^2\right) \tag{3.29}$$

Although AIC and BIC are similar, the asymptotic behavior of both criteria is very different. According to Rudolf Beran's comment on Shao (1997), "BIC-like criteria would perform better if the true model has a simple structure (finite-dimension) and AIC-like criteria would do better if the true model is a complex one (infinite-dimension)". He argues in favor of the BIC, that in praxis, simplicity is often chosen over correctness, because the aim is to extract information and not find the true model.

3.9. Pitfalls in statistical learning

Missing values If there are data points (training examples) where values for features or the target value are missing, these have to be treated in a special way. Either we discard them, thereby decreasing the training set size. However, for small training sets, the training set size has an impact on the prediction accuracy: The more examples there are from a data source, the better we can estimate the true function. Alternatively, we can come up with a replacement of the missing values according to context. The mean $\overline{x_j}$ for a missing feature value or zero could be possible choices.

Variance and importance of features Features with a higher variance have a higher impact on the regression function. Often this is not desired. It implies that these features are more important. As a countermeasure, features should always be centered and normalized by variance:

$$x_j^{\text{norm}} = \frac{x_j - \overline{x_j}}{\sigma(x_j)}, \tag{3.30}$$

where $\overline{x_j}$ is the estimated mean and $\sigma(x_j)$ the standard deviation of x_j

4. Scope of this work

The aim of this work has been to assess if and how peak intensities of peptides analyzed with mass spectrometry (MS) can be modeled. Label-free quantification accuracy suffers from the fact that the measuring sensitivity of MS is peptide-specific (see Section 2.5.4). As of yet, there is no model to predict the signal intensities for new peptides since the chemistry leading to the differences in sensitivity is very complex.

We propose to use predicted peak intensities to correct for the peptide-specific deviation of the peak intensities for constant peptide abundance. This is to be achieved by modeling the whole workflow of an MS experiment that peptides undergo. By predicting peak intensities accurately for peptides with the obtained model, the above-mentioned deviations could then be calculated easily. Before this can be tested in an actual biological study, we concentrate on finding out if peak intensities can be predicted at all, and which is the best way to do so. The approach is described in more detail in Chapter 5.

A model defined by known signal intensities of all possible peptides in the way described by Jarman *et al.* is prohibitive: It is infeasible to measure all possible peptides to acquire and store their intensity. Thus, our model has to be more general and applicable to peptides that were not identified and measured before, so it can be used for quantitative analysis of new proteomes.

This constitutes an important step towards accurate label-free quantification via peptide peak intensities.

4.1. Related work

In the introduction to proteomics (Section 2.5.4), I point out studies that deal with the prediction of peptide detectability or intensity. This work constitutes a new method: While peptide detectability prediction has been presented by multiple parties (Gay *et al.*, 2002; Lu *et al.*, 2007; Mallick *et al.*, 2007; Tang *et al.*, 2006), peak intensities have only been analyzed by Gay *et al.* (2002) and Jarman *et al.* (1999) so far. Both did not directly aim at peak intensity *prediction* though: Jarman *et al.* (1999) analyzed the mean and standard deviation of peptide peak intensities to be able to recognize known protein mass fingerprints (PMFs). In the work of Gay *et al.* (2002), the majority of the distinct peptides were present in both training and test set. Thus, they mainly measured the reproducibility of intensities by the applied algorithms. Their main focus was on the discovery of rules for peak intensities. In contrast, this work aims to actually predict peptide-specific sensitivity of *new* peptides using peak intensities.

4.2. MS setups relevant to this work

The main work concentrates on MALDI-TOF datasets with prior protein separation using 2D-PAGE. Because these spectra very often contain only one protein, all peptides belonging to this protein in the spectrum can be extracted easily and have the same abundance. Thus, the peak intensities of these peptides correspond to the MS device's peptide-specific sensitivity. Data from LC/ESI MS was acquired in a later phase of the project. Therefore, only basic analyses were carried out on these data. We introduce the relevant setups for both kinds of data here.

There are a lot of highly specialized mass spectrometry setups that comprise an ion source, one or multiple mass analyzers, and a detector. These can be coupled to an analyte feed. In addition, different protein separation techniques exist to decrease the complexity of the analyte mixture before the analysis. All of these parts are freely configurable, depending on the type of analysis that is to be done. Often, setups between laboratories have common main components, but there are slight differences in the setup details. As research is proceeding, new setups are developed and adapted to the current needs. In the following, two setups for protein analysis are described whose basic assemblies are relevant to this work.

MALDI-TOF One possible setup is the separation of a complex biological sample by two-dimensional gel electrophoresis (2-DE) and analysis with MALDI-TOF – a MALDI ion source coupled with a TOF mass analyzer, as shown in Fig. 2.7. The typical procedure is shown in Fig. 4.1. Here, each protein leads to a single spectrum.

LC-ESI Another widely used setup is to digest a protein mixture first, leading to a very complex peptide mixture, then separate it by HPLC and analyze it with ESI-MS.

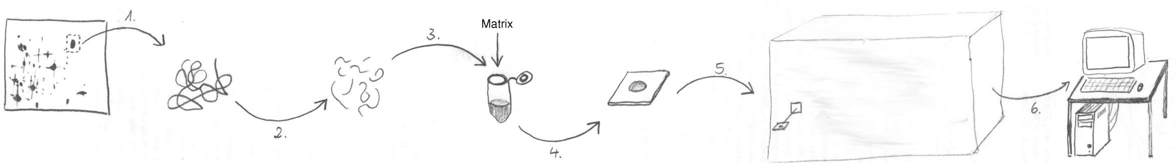


Figure 4.1: Workflow for protein analysis with MALDI-TOF after protein separation by 2D-PAGE: Single spots are cut from the gel (step 1.), manually or automatically. Usually, one spot contains only one protein or sometimes multiple isoforms of the same protein. These are digested proteolytically using *trypsin* or another *protease*, resulting in a mixture of shorter peptides (2.). After being mixed with a matrix substance (3.) and left to crystallize on an object plate (4.) the sample is inserted into the MALDI-TOF MS device (5.). The resulting output data is directly fed into a PC (6.) where it is stored, and data analysis is performed.

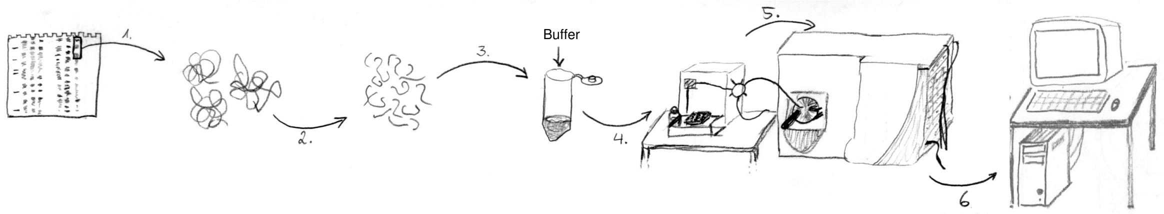


Figure 4.2: Workflow for protein analysis with LC-ESI and prior separation with SDS-PAGE: Fractions are cut from the SDS gel (step 1.) manually. A fraction contains many proteins. These are digested proteolytically using *trypsin* or another *protease*, resulting in a mixture of peptides (2.). After solution in running buffer (3.), the sample is injected into a liquid chromatography (LC) device (4.), portions eluting from the LC are fed into the MS device (5.). The resulting output data is directly fed into a PC (6.) where it is stored, and data analysis is performed.

5. Modeling the mass spectrometry process

This chapter deals with the following problems:

1. What is the aim of the modeling?
2. Which steps have to be modeled?
3. What techniques can be applied to model these steps?
4. Each step introduces sources of error and/or noise that our model does not consider. How are these constituted? How do they influence the overall process?

5.1. The aim of the model

The constructed model should enable us to correct for the deviation from a linear relationship between peptide abundance and a corresponding peak intensity value obtained from an MS experiment. It should model the whole workflow of an MS experiment that peptides undergo. By predicting peak intensities for peptides with this model after performing normalization to account for differences in protein abundance, the mentioned deviations could then be calculated easily.

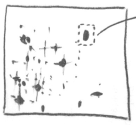
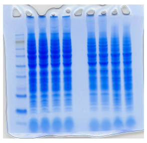
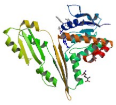
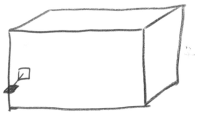
For the purpose of peptide-specific sensitivity prediction, it makes sense to include pre- and post processing steps, although it complicates the process to model. It is infeasible to gain data of the separate processes on a large enough scale, and in a typical application these steps are always included.

Before studying the usefulness of this model in an application, we have to study different methods to determine how to build such a model. At this point, it is unclear how to model the peptide-specific sensitivity from peptide strings. Existing models have predicted spectra for molecules as large as a few atoms. Although these models may be accurate under certain conditions it does not allow us to predict peak intensities of large molecules such as peptides in a full-scale experiment. Additional factors specific to protein MS that influence the detectability of peptides (such as tryptic digestion efficiency or PTMs) have to be accounted for.

5.2. Steps and techniques

Here, analogies between the MALDI-TOF and LC-ESI MS pipelines are shown and generalized into a pipeline from which the model is derived. Details listed in the table are explained in the next chapter. Numbers denote states in the whole process, transitions between these are marked with characters.

We are interested in the input (proteins) and the output (normalized peptide-specific sensitivity values as target values, and their corresponding peptide sequences). This is what this model is to capture.

step	2D-PAGE MALDI	LC-ESI	general description
1			separated protein mixture
$\mathcal{A}$	cut out a spot (single protein)	cut out a fraction (multiple proteins)	select protein(s)
2			selected protein(s)
$\mathcal{B}$			tryptic digestion
3			peptide mixture
$\mathcal{C}$	add matrix, crystallize	add running buffer	method-specific treatment
$\mathcal{D}$			MS analysis
5	one MS^1 spectrum per protein	one MS^1 spectrum per time point + MS^2 spectra of the highest peaks in MS^1 + MS^3 spectra	acquired raw data

step	2D-PAGE MALDI	LC-ESI	general description
$\mathcal{E}$	peak extraction + ID by PMF (see 2.5.1)	peak extraction + ID by peptide fragmentation fingerprint (PFF)	peak extraction and identification (ID)
$\mathcal{F}$	extraction of peak heights (Section 6.1.2)	extraction of area under curve (AUC) values (Section 6.2.2)	intensity extraction
$\mathcal{G}$	normalization between spectra (Section 6.1.4)	merge data from different fractions, normalization by protein abundance (Section 6.2.4)	normalization
$\mathcal{H}$			feature extraction, calculation of target value for each peptide
6			extracted data points as pairs $d = (p, y)$ with p a peptide sequence and y a target value

Transition $\mathcal{B}$ between state 2 (proteins / long strings) to 3 (peptides / non-overlapping substrings) is a tryptic digestion. This is simulated: In the wet lab, the enzyme trypsin cuts proteins after lysine (K) or arginine (R) if not followed by proline (P). There are exceptions to this rule though, and errors in the sample treatment may lead to more so-called missed cleavages, resulting in longer peptide fragments that contain lysine or arginine. The digestion is simulated by a function $\mathrm{phi} : s \mapsto \Omega$ where s is a protein sequence and Ω a set of non-overlapping substrings of s.

Example The string s = SLLNIDPHSSDYLI**R**LSPPDL**K**HEFAL**KP**QSFTSIA**R**YWGILSNE is mapped to Ω = {SLLNIDPHSSDYLR, LSPPDLK, HEFAL**KP**QSFTSIAR, YWGILSNE}

Under carefully controlled conditions tryptic digestions works quantitatively, so we do not model missed cleavages, i.e. overlapping substrings. Knowledge about the protein sequence s is gained from identification via database search (see Section 2.5.1). These identifications are taken as a gold standard , i.e. the model assumes correct input strings.

Transitions $\mathcal{C}$ to $\mathcal{H}$ involve multiple nonlinear transformations, which are highly complex and interdependent. We do not use a physical/chemical simulation here, neither do we model these steps separately. Even if all the factors involved were known, their individual influence and how they are related to each other is not (see Section 2.5.3). Instead, we use a machine learning approach in combination with prior transformation of peptide strings to high-dimensional numerical vectors. Our first choice is support vector regression (SVR, Section 3.3.1), because it is robust and is known to generalize well to new data. In principle, any regression model can be used. The ν-SVR has three parameters that have to be set. With these, the regression function itself is adapted to the data by the algorithm automatically. The parameters are found using a grid search in combination with ten-fold cross-validation (see Section 3.8). The parameter grid is layed out as $\nu \in [0.2, 0.8]$ in steps of 0.1, the regularization parameter $C \in \left[e^{-3}, e^{9}\right]$ and Gaussian kernel bandwidth $\gamma \in \left[e^{-5}, e^{7}\right]$, both in steps of e^2. The mapping of peptide strings p to numerical vectors $\mathbf{x}$ as input for the ν-SVR training constitutes the core of the model and this work. Chapter 7 and 8 deal with these mappings.

5.3. Sources of noise and errors

Despite standardized protocols, there are a lot of factors in MS which account for systematic and non-systematic errors (noise).

Modifications Additive PTMs (see 2.1.1) change the molecular weight of the affected molecule. Fixed modifications can easily be integrated in the model because they affect 100% of the molecules that contain the affected amino acid. Variable modifications are those that only occur on a portion of the molecules that can be affected. In this case, the mass of some of these changes. Since modifications also change the sensitivity of the MS device to the affected molecule, we cannot just add the intensity from the modified peptide's mass-to-charge ratio. We neither know how many of the peptides have been modified. Thus, the peak intensities of the unmodified peptides that also occur with (partial) modifications are lower than expected by the model.

In other terms, abundances of these peptides could be described by

$$I = f_1\, i_1 + f_2\, i_2,$$

where f_1 and f_2 are unknown, if there was only one possible modification. We would have to predict both values for each peptide. However, there are no target values for i_1 and i_2 because there is no information available about the modified fractions of a peptide. Also, these values are very difficult to obtain from a non-synthetic dataset.

Ideally, we should know for each peptide if there are modified versions of it in the same sample, and exclude it from the training. Unfortunately, we do not have this information for all of our datasets. So far, our model ignores modifications.

Suppression effect The ion suppression effect introduces an additional complication. The term has been introduced by Buhrman *et al.* (1996). Ion suppression is a secondary effect that suppresses the signal of a compound in the presence of other compounds that compete for ions during ionization. The intensity prediction would take this effect into account automatically if there were large amounts of data available in which one peptide occurred in multiple spectra with all possible combinations of other peptides, and if there were no contaminations. However, such a dataset is impossible to acquire. Knowing this, we neglect the fact that peptide peak intensities depend not only on the peptide's constitution, but also on the combination of other peptides present. This is an addition unknown noise component. Wang *et al.* showed that even in a complex mixture the intensity of a certain peptide is still linear with its concentration, although the ion suppression effect is more noticeable than in a less complex synthetic mixture Wang *et al.* (2003). Therefore, I am confident that its effect is negligible for the MALDI spectra used in this work.

Incomplete digestion As mentioned above, small errors during wet lab processing may lead to incomplete digestion during the preparation steps that generate the peptides from proteins. As a result, a certain amount of some peptides is missing. Instead, the analyzed mixture contains longer peptides that include the sequences of these peptides. Thus, the resulting peak intensities that the model is adapted to are lower than expected for these cases.

Ideally, only peptides that are cleaved to 100% or 0% (i.e. have a cleavage site that is always missed) should be used to adapt the model. Again, this information is not always available. Independent of the dataset, the model assumes each peptide string to be non-overlapping with other substrings of the corresponding protein sequence.

Analyte concentration in matrix For MALDI-TOF, the sample is mixed with a matrix substance, and left to dry and crystallize before being inserted into the device for analysis. The matrix-analyte ratio has to be carefully controlled because it influences the noise level of spectra. Matrix molecules compete for protons along with the analyte molecules. Therefore, too much matrix substance increases the noise, which leads to more unidentified peaks as well as higher variation of peak intensities between replicate runs especially for peaks with low intensity. If the concentration changes between different spectra, the inter-spectra variance is increased.

For LC-MSMS, the sample is mixed with a running buffer. Here, the concentration together with the amount introduced into the device has to be controlled. In case of too much analyte, a detector saturation may occur. This implies that at high intensities linearity with concentration for a specific peptide cannot always be guaranteed.

The model takes this effect into account implicitly by adapting itself to the presented data. However, if this parameter is changed, a new instance of this model would have to be adapted to the changed characteristics. This issue can be avoided altogether if the sample preparation is carried out with care and the resulting spectra are controlled visually for saturation effects before further analysis.

Variations between different runs (machine replica) There is some variation between runs of the same device with the same sample for various reasons. First, the involved machines are highly precise and thus susceptible to minor disturbances in the environment such as temperature and pressure changes, or magnetic fields. These factors can only be controlled to a certain degree. Secondly, most often, the sample preparation is done by human experts, so there is always some variation in the handling, even for persons in the same lab using the same protocol. For MALDI, the mixing of the matrix and the analyte is critical. Inhomogeneities lead to variations of the analyte concentration between runs with the same sample. As a result, noise is added to the target values for the learning algorithm, which effects the accuracy of the model. This effect increases if fewer replica are available. Noise can be suppressed to a degree by using a robust mean or median of all runs instead of the mean.

Identification issues Section 5.2 states that any protein identification is taken to be true. There is a small chance for a protein misidentification though. Some of these may also come in combination with a peptide misidentification. The resulting error in the model would be an erroneous numerical input vector for the regression step of the model.

Undersampling (LC-ESI) With each LC-ESI run, only a fraction of the peptides in the sample can be identified, because there is only limited time to analyze the sample coming from the LC column before a new drop has accumulated. Therefore, we miss a portion of the peptides, making normalization with known protein levels inaccurate. This can be overcome by acquiring a lot of replicate runs of the same sample, which is very time-consuming.

Charge states (LC-ESI) ESI spectra contain multiply charged ions. Therefore, there can be peaks from differently charged ions for one peptide in a spectrum.

Obviously, sources of noise and errors in MS data are numerous, and they cannot always be excluded. Often, there is not enough information available to include them in the model. Therefore, a robust learning algorithm is necessary.

5.4. Accuracy enhancement of absolute quantitation with predicted peak intensities

To quantify using predicted peak intensities $\widehat{I}_p$ for peptide p from our model, a peptide-specific correction factor $\widehat{f}_p$ has to be calculated as $\widehat{f}_p = \frac{1}{\widehat{I}_p}$. The relative abundance can be calculated from the measured intensity I_p of peptide p as

$$c_p^{rel} = I_p \, \widehat{f}_p$$

If we applied this to peptides of the same protein in a MALDI spectrum, c_p^{rel} should be 1 for all peptides. Absolute quantitation of peptides is possible by normalizing c_p^{rel} with the overall sample concentration c_{sample} as proposed by Lu *et al.* (2007):

$$c_p^{abs} = I_p \, \widehat{f}_p \, c_{sample}$$

A more precise quantitation would be possible if a known amount c_k of a peptide k (or multiple peptides) with a high value for $\widehat{I}_k$ is spiked in before the MS run. The absolute abundance would then be

$$c_p^{abs} = I_p \, \widehat{f}_p \, \frac{c_k}{\widehat{I}_k}$$

Instead of $\widehat{I}_k$ (i.e. an estimated intensity for the spiked-in peptide), a known intensity I_k from previous experiments containing peptide k could also be used to get an even higher accuracy. We are aware that this only works if the peptide k is observed at all. Careful consideration has to be directed to the choice of this peptide.

The absolute or relative quantitation $c_r^{rel/abs}$ for a given *protein* r can be derived using the quantitative values from its peptides:

$$c_r^{rel/abs} = \mu\left(c_i^{rel/abs})\right) = \frac{1}{N_r} \sum_{i=1}^{N_r} c_i^{rel/abs}$$

where N_r is the number of peptides for protein r and $i \in [1, N_r]$. For large N_r, a robust mean (α-trimmed mean) could be used instead of the normal mean.

6. Data acquisition, processing, and analysis

This chapter presents a close-up of the data used in this thesis. Because there are significant differences in the data handling between the MALDI and LC-ESI datasets, there are two sections for both types of data that are in the order of the data processing pipeline. Section numbers refer to the corresponding section for MALDI and LC-ESI data respectively.

First, the **wet-lab procedures** (Section 6.1.1 and 6.2.1) the data is obtained from in the first place.

After MS analysis, raw spectra undergo **in silico preprocessing** to derive peptide intensities (Section 6.1.2 and 6.2.2). By intensities, both peak heights or area under curve values can be denoted, and both are used as described below.

Because of the varying quality of data from this domain it is necessary to filter the available spectra. That way, the noise and number of erroneous data points in the final datasets can be at least reduced. However, it is impossible to assure 100% correct data without evaluating every single spectrum manually. The **dataset construction** sections (6.1.3, 6.2.3) describe the criteria that are used to select the final datasets.

Different types of **normalization** are necessary depending on the data context (Section 6.1.4, 6.2.4). Common to data from both domains is that intensities are expected to be approximately lognormal distributed (Listgarten and Emili, 2005). Also, errors become additive when taking the logarithm, thus stabilizing the variance (Anderle *et al.*, 2004; Bantscheff *et al.*, 2007).

MS signal intensities can be reproducible if care is taken during all preparation steps. However, errors may happen in various steps. Therefore, **statistical analysis** (Section 6.1.5, 6.2.5) of the resulting intensities is necessary for quality assurance. It allows us to determine runs where machine errors or intensity shifts have occurred. Either discarding or data-dependent normalization can be used to encounter these issues. Even if statistical analysis does not result in any measure being taken, it is important to assess the quality and reproducibility of the data to be able to judge the results in down-stream analyses.

Statistical methods Various statistical methods and visualizations are applied to get a better understanding of the data:

- Reproducibility is always an issue with MS data. To analyze this, correlations and coefficients of variation (CV) between intensities of replicate measurements of the same peptides are determined.
- Even peptides that the device is very sensitive to are not always detected in the peak picking procedure. Apart from that, a higher number of replicate measurements leads to more certainty about the peptide-specific sensitivity and a better chance to recognize outliers. Therefore, it is illustrative to get an overview of how often peptides were found through histograms.
- Intensity distributions may vary between different runs which has an impact on prediction accuracy later-on. Density estimation plots and boxplots (Tukey, 1977) are useful to assess these differences. As an additional benefit, visualization of the intensity distribution between LC-ESI runs allows us to easily spot runs where the MS device did not work correctly.
- Common statistical methods assume normal distribution. Q-Q plots are one way to visualize deviations from a normal distribution.

6.1. MALDI-TOF data

6.1.1. Wet lab procedures

Two sets of MALDI raw spectra, denoted A and B, were used to obtain data for this study. They were generated on a Bruker Ultraflex instrument (Bruker Daltonics, Bremen, Germany) during experiments on *Corynebacterium glutamicum*. The proteins were separated by 2D gel electrophoresis and digested into peptide fragments with trypsin prior to MS analysis. The corresponding peptide sequences were derived from protein identification using MASCOT peptide mass fingerprinting (Pappin *et al.*, 1993) and an in-house database containing *C. glutamicum* protein sequences. The wet lab preparation was done by Martina Mahne (dataset A) and Nicole Hansmeier (dataset B)[1].

[1] Institute for Genome Research and Systems Biology (IGS), Bielefeld University

6.1.2. In silico preprocessing

The following steps are performed by the peak extraction software developed by Matthias Steinrücken and me. It is based on the `imslib` library framework that has been developed by Sebastian Böcker, Anton Pervukhin, Henner Sudek, Marcel Martin, and Matthias Steinrücken.

De-noising and baseline correction Noise filtering is done with a Savitzky-Golay filter of length 17 and degree 4 (Savitzky and Golay, 1964). A baseline correction is applied by calculating a list of maxima and minima on the filtered spectra, controlled by parameters for the minimum and maximum width of a peak. The list of minima is used to estimate the baseline, which is subtracted from the de-noised spectrum.

Eliminate noise peaks The list of maxima is filtered for noise peaks by the following algorithm:

```
Algorithm CleanList
for each window of width $w$
    do sort list of maxima by their height
        calculate $b$ as the mean of the ordered list after trimming away the upper 50% and
        the lower 25% of the values
        set the threshold $\theta = b\xi$
        discard all peaks with height below $\theta$
```

The multiplier $\xi = 3.0$ and window width $w = 500$ Da was set after testing different values by visual inspection of the spectra. Peaks still present after this cleaning step are considered for the consecutive steps (isotopic deconvolution and peak matching).

Isotopic deconvolution Isotopic deconvolution was carried out as follows: We calculate the theoretical isotope pattern of an average protein in 500 Da steps throughout the appropriate mass range. For peaks that lie between these calculated points, the isotope pattern is linearly interpolated. The n^{th} isotope peak for a monoisotopic peak at mass m with intensity i is denoted $h(i, m, n)$. We iterate through the list of maxima in ascending order of mass-per-charge ratio.

```
Algorithm FastIsotopicDeconvolution
for each peak $P$
    do $m_1$ = mass of peak $P$ and assume that it is a monoisotopic peak
        $i_1$ = intensity of peak $P$
```

```
set $n = 2$
while there is a peak $P_n$ inside $[m_{n-1} + 0.9, m_{n-1} + 1.1]$ Da
    do $i_n$ = intensity of $P_n$, assume it to be the $n^{th}$ isotopic peak
        if ($i_n < 0.01$ i)
            then continue to next peak, at the beginning of for loop
        calculate the theoretical intensity of the $n^{\text{th}}$ isotopic peak $\hat{i}_n = h(i_1, m_1, n)$
        update $i_n$ and $i_{n-1}$ as $i_n^{new} = i_n - \hat{i}_n, \quad i_{n-1}^{new} = i_{n-1} + \hat{i}_n$
        $n = n + 1$
```

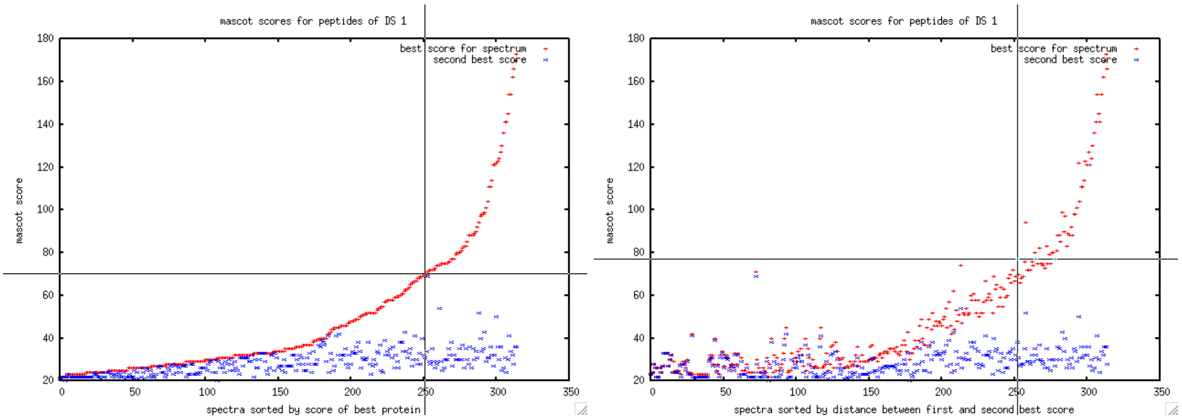
Theoretical digestion Now having a list of monoisotopic peaks with summed intensities, the protein sequence from the MASCOT peptide mass fingerprint identification is used to perform a theoretical tryptic digestion on it. As a result, a list of peptides is retrieved. The protein string is split after the character K or R if not a P is following. Only peptides that would result from a perfect digestion are calculated. We calculate the monoisotopic masses of these theoretical peaks.

Identification of proteins in the MALDI datasets To get the protein sequences, Mascot identification results handed to me by Andreas Wilke (dataset A) and Christian Rückert (dataset B) were used.

Identification parameters for dataset A include carbamidomethyl as a fixed modification, oxidation of methionine as a variable modification, and no missed cleavages. The mass tolerance is 1 Da.

Dataset B was searched with Mascot with various parameter sets: In addition to carbamidomethyl as a fixed modification, the following parameters were used a) with and without oxidation of methionine, b) tolerance within {50, 100, 150, 200, 250, 500, 750} ppm, and c) up to {0, 1, 2} missed cleavages allowed.

Peak matching Using the masses from the theoretical digestion we look for matches all over the spectrum. The matched peak's intensity is then assigned to the peptide sequence and the spectrum currently under consideration. We allow for mass errors of up to 1.0 Da to consider a peak a match. Spot checks in the resulting mass error in the matched peak lists showed that there are actually large masses, for which such a large error occurs. In almost all cases, the errors increase towards larger masses, suggesting that the matches are correct even though the calibration was not good. Mismatched peptides (i.e. mass error non-monotonic) mostly occur in the area below 800 Da where most of the matrix noise peaks are expected. In case of multiple peaks being in the allowed window, the one with the lowest error (i.e. the nearest peak) is chosen as a match. We do not choose the peak with the highest intensity since we do



(a) Ordered by first hit score

(b) By distance between first and second hit score

Figure 6.1: Mascot scores for first and second hit of dataset A. Based on this, a cutoff of 65 was chosen below which spectra were discarded. This results in a minimal distance of 39 between first and second best hit scores. The cutoff point is visualized by a grid: Only hits in the upper right corner are used in the further analysis.

not want to assume any knowledge about the intensities which we do not have. The highest peak does not necessarily have to be the correct match. For A for the cysteine mass 103.009184 Da was used, while for B, we use 160.030648 Da which is the mass of carbamidomethylated cysteine.

The mass ranges for the peak extraction are [650, 3118] Da for A and [800, 3578] Da for B.

6.1.3. Construction of data sets

Dataset A The first and second hit scores from the Mascot identification were evaluated to determine a cutoff that retrieves about 20% of the spectra. To make sure spectra only contain one protein, the distance between first and second hit scores has to be large. Based on this evaluation (Fig. 6.1), a cutoff distance of 39 between first and second hit score was used. This implies Mascot scores of above 65. As a result, 62 of 315 spectra are used for further analysis. Of 27 identified proteins, 15 were present in multiple spectra.

Dataset B was taken from a study by (Hansmeier *et al.*, 2006) and run through a fully automated MASCOT peptide mass fingerprinting search with 42 different sets of search parameters (see 6.1.2). A protein was considered identified if there were more than 6 parameter sets with the same hit. The resulting list was filtered automatically to fulfill the following

	Dataset A	Dataset B
# spectra	61	184
# proteins	27	125
# non-matches	164	971
# matches	371	1023
# duplicate proteins	15	35
duplicate peptides	50.8%	29.8%
modifications	no	fixed carbamidomethyl
mass range	650 - 3118 Da	800 - 3578 Da
selection	expert	mostly automatic

Table 6.1: Overview of MALDI dataset properties. # *matches*: number of distinct peptides for which peaks are found in the spectra, considering only peptides without missed cleavages. # *non-matches*: number of theoretical peptides for which no match was found. # *duplicate proteins*: number of proteins for which more than one spectrum is contained (detailed numbers are visualized in Fig. 6.4). *Duplicate peptides*: percentage of peptides found in more than one spectrum. *Modifications*: peptide modifications considered in the peak matching procedure. This is different from the modifications considered during protein identification.

properties: a) Protein mass within 8000 to 12000 Da, b) pI between 4 and 7 because that is the range the 2D-PAGE gel allows, c) MASCOT protein hit score above 65, and d) sequence coverage above 15%. Application of this protocol left 182 of 493 spectra for further analysis. Of 125 identified proteins, 35 were present in more than one spectrum.

From the peak extraction, 371 (out of 535 theoretically possible) data points each consisting of a distinct peptide sequence and in a number of cases multiple intensity values are retrieved for A and 1023 (out of 1994) for B. Unmatched peaks leading to unlabeled data points were not used. To summarize the differences, A can be considered a small, carefully chosen dataset while B is larger and of lesser overall quality. The number and fraction of unmatched peptides are much higher in B.

An overview of both MALDI datasets is shown in Table 6.1.

6.1.4. Normalization for MALDI datasets

For a MALDI spectrum, the protein abundance in a single spectrum is unknown and varies between spectra. Therefore, to normalize spectra, we apply two alternative formulae, followed by a logarithm.

Normalization by corrected mean ion current (mic). The intensity of a peak p is scaled by the mean ion current (i.e. the mean of all $i = 1, \ldots, N$ values in the whole spectrum) after peak extraction to yield the normalized intensity

$$I_p^{mic} = \ln \left(\frac{I_p}{\frac{1}{N} \sum_{i=1}^{N} C_i} + 1 \right) , \tag{6.1}$$

where I_p denotes the non-normalized intensity of peak p after peak extraction (for details see Section 6.1.2). The value $C_i = D_i - B_i$ is the raw value D_i at position i after de-noising and the nearest baseline value B_i subtracted, and i runs over all raw values (i.e. not only peptide peaks) of the spectrum s the peptide was found in: $\min(\frac{m}{z}, s) \leq i \leq \max(\frac{m}{z}, s)$.

Normalization by the sum of all peptide peak intensities (sum). The intensity of a peak p is scaled by the sum of all matched peptide's peak intensities $i = 1, \ldots, P$ to yield

$$I_p^{sum} = \ln \left(\frac{1000 \cdot I_p}{\sum_{i=1}^{P} I_i} + 1 \right) , \tag{6.2}$$

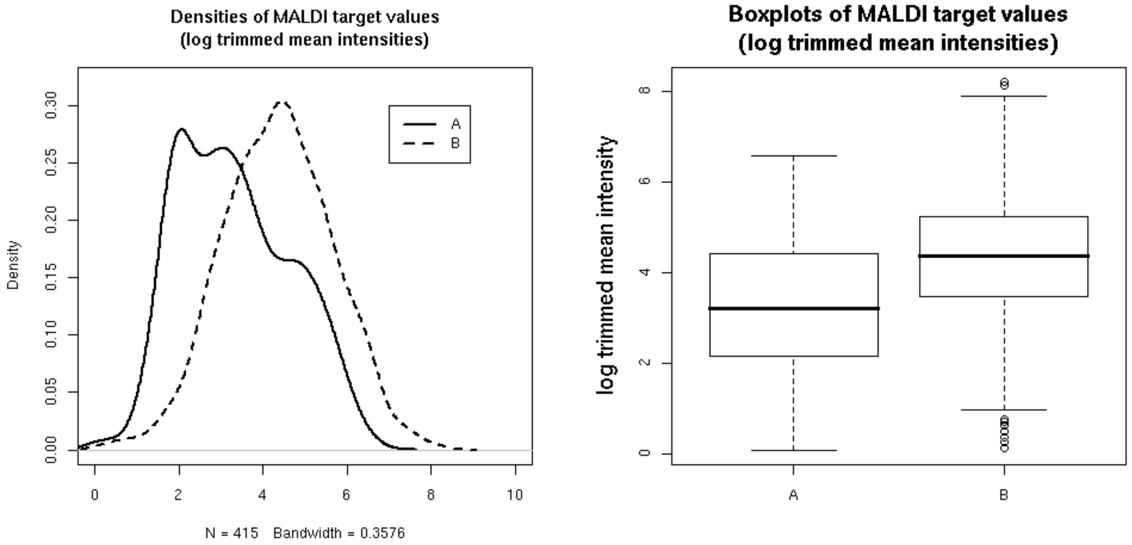
where I_i denotes the intensity of the i^{th} peptide peak after peak extraction.

Trimmed mean intensities In general, there can be more than one measurement for each distinct peptide. Hence, there are peptides that have more than one intensity value. An *α-trimmed mean* ($\alpha = 50\%$) is calculated for cases with more than three target values. Otherwise, the mean is taken. This increases the statistical certainty of target values and cancels the effect of outliers. Unfortunately, for cases with only a single measurement no such benefit can be attained.

6.1.5. Statistical analysis

An analysis of the MALDI datasets A and B reveals some differences. Dataset B contains about thrice as many distinct peptides as A. The distributions of both sets differ: The logarithmic normalized target values of B are very near to normal distributed, while there is a skew towards lower values for A (Fig. 6.3). The means of both distributions also differ slightly (see Fig. 6.2).

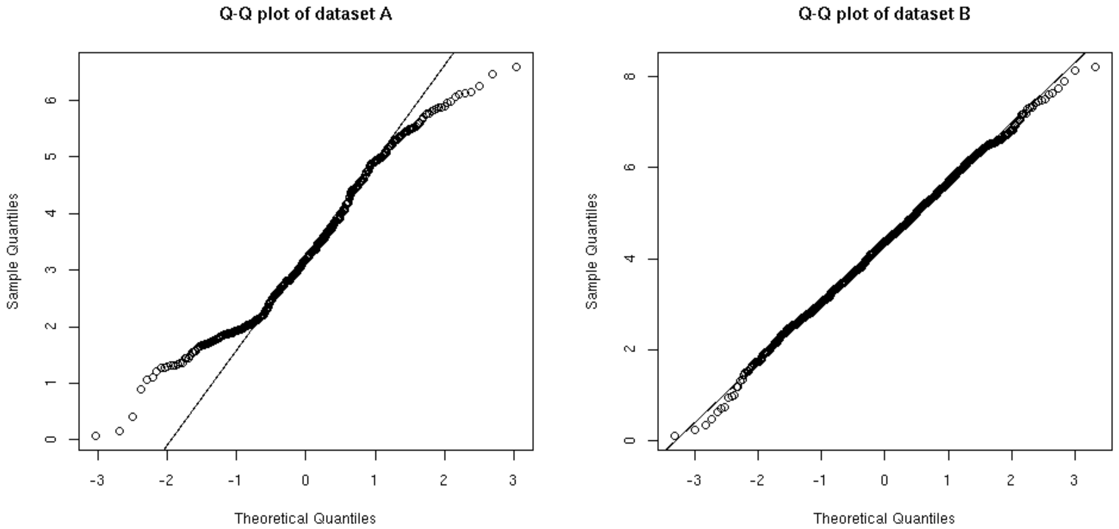
The reproducibility is better for A, which has a squared Pearson's correlation of $r^2 = 0.64$ between target values of the same peptide for replicate measurements compared to $r^2 = 0.35$ for B (illustrated in Fig. 6.6). If the intensity values are held against their respective target values (i.e. trimmed mean values), correlations of $r^2 = 0.84$ and 0.68 are recorded. These latter values can be considered an upper bound for the best achievable prediction performance if only peptides with multiple measurements were used. Visually, it is obvious that B has a



(a) Density estimation

(b) Boxplots

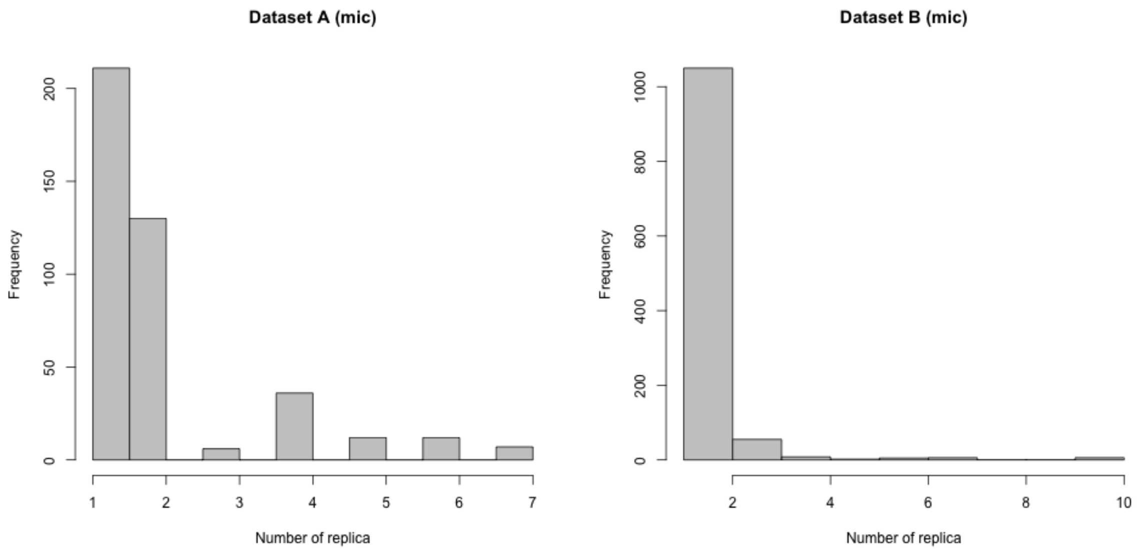
Figure 6.2: Distributions of MALDI datasets as estimated density and boxplot. The values of dataset B are higher in the mean and its distribution is more balanced but with more outliers.



(a) Dataset A

(b) Dataset B

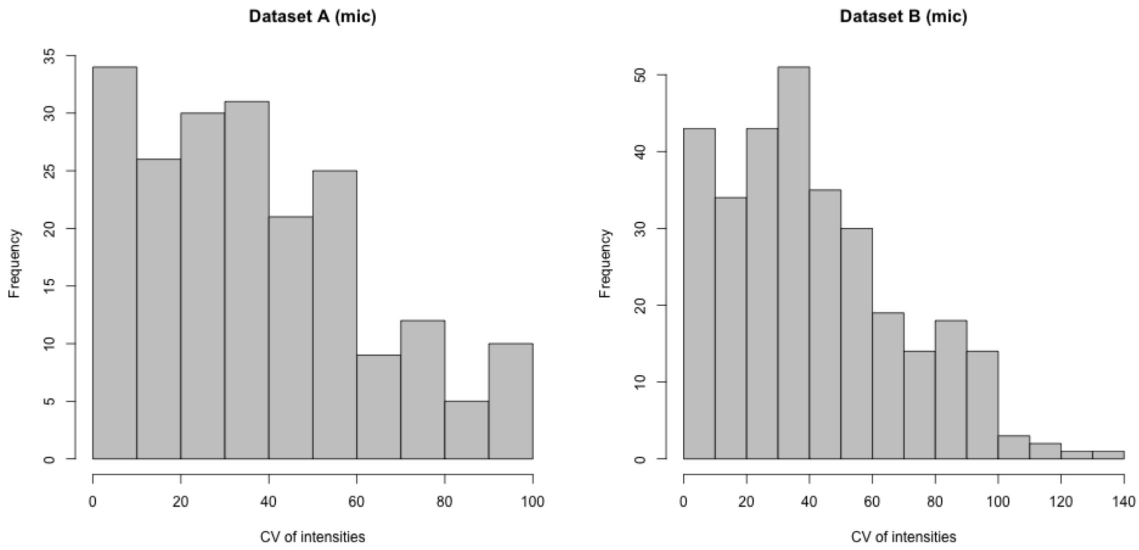
Figure 6.3: Q-Q plots for target values of both MALDI datasets.



(a) Dataset A

(b) Dataset B

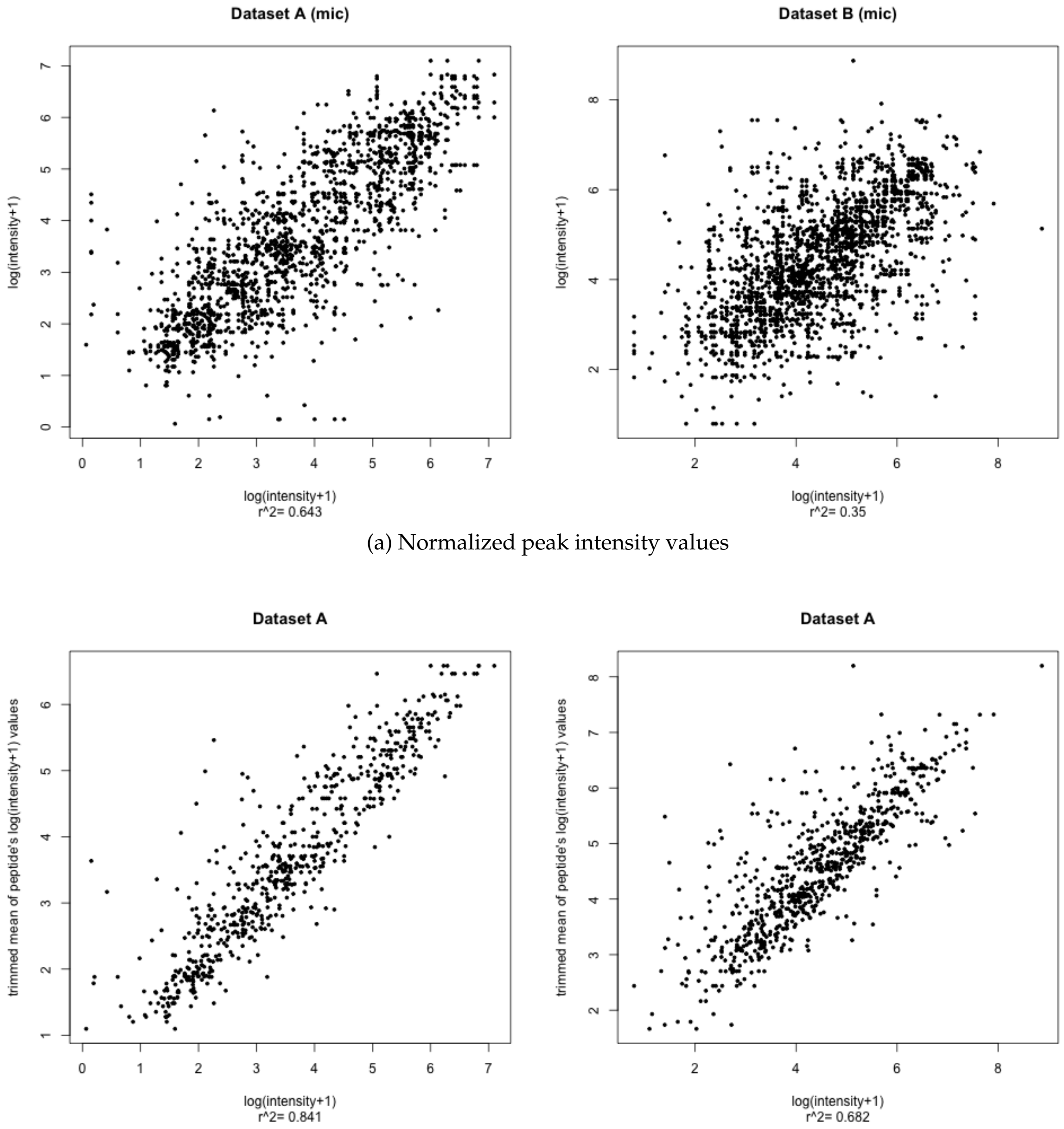
Figure 6.4: Histograms of the number of replicate peptide measurements for both MALDI datasets.



(a) Dataset A

(b) Dataset B

Figure 6.5: Histograms of coefficients of variance (CV) between intensities of replicate runs for both MALDI datasets.



(a) Normalized peak intensity values

(b) Normalized peak intensity values vs. trimmed mean target values

Figure 6.6: Scatter plots and correlation coefficients of within peptide peak intensity variance between runs for all peptides of both MALDI datasets (*top*: dataset A, *bottom*: dataset B). The recorded correlations can be considered an upper bound of the achievable prediction performance if there was exactly one measurement per peptide (single target values) or if there were multiple measurements for each peptide (trimmed mean values).

higher spread. In addition, there is a smaller portion of peptides that have been measured multiple times in B. Nonetheless, the coefficients of variation are similar (mean coefficients of variance (CV) of A: 32.6%, B: 31.2%, see Fig. 6.5). In addition, there is a smaller portion of peptides that have been measured multiple times in B (Fig. 6.4). Thus, B can be considered the noisier of both datasets. Table 6.3 shows a summary of the mentioned reproducibility measures in comparison to the LC-ESI datasets.

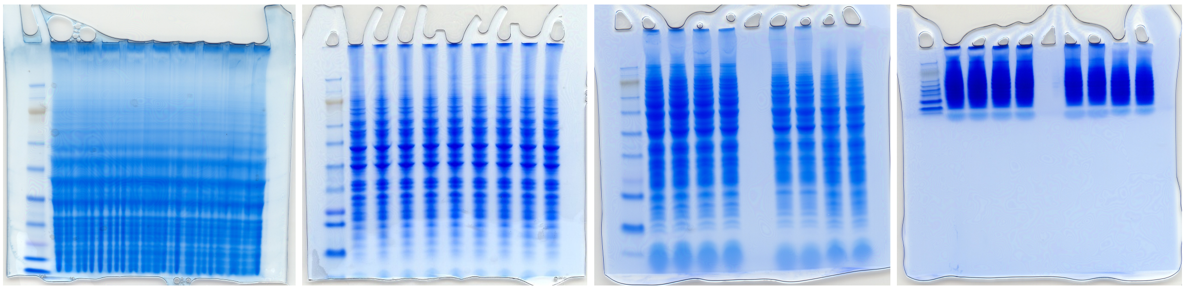


Figure 6.7: SDS-PAGE separation of the yeast samples. From left to right: Y_1, Y_2, Y_3 gel for five fractions, Y_3 gel for one fraction. For Y_3, the four columns on the left belong to the *S. cerevisiae* samples (two columns each).

6.2. LC-ESI data

6.2.1. Wet lab procedures

Yeast whole cell lysate from cultures Y_1, Y_2, and Y_3 of the same *Saccharomyces cerevisiae* strain were grown on different days. For this, the standard sequenced lab strain BY4741 from Open Biosystems[2] is used. The cultures were in mid log growth when collected. For Y_3 two growth essays denoted $Y_3(1)$ and $Y_3(2)$ were done in parallel. The whole cell lysates of Y_1 and Y_3 were sonicated, treated with thioethanol, and heated to 95 C for 5 minutes to get rid of DNA and large membrane fragments. The mixture was separated using SDS-PAGE, and $F \in \{1, 5, 10, 16\}$ fractions were preprocessed and analyzed via MS separately. Preprocessing involved denaturing with *dithiothreitol* (DTT), alkylation with *iodoacetamide*, and incubation with trypsin over night. Overall, six sample sets were generated (see Section 6.2.3 for details): $F = 16$ for Y_1, $F = 10$ for Y_2, $F \in \{1, 5\}$ for the two preparations of Y_3.

Bovine serum albumin (BSA, 1 nmol) was spiked into the five-fraction samples prior to digestion to normalize between fractions later-on. All samples were analyzed with LC coupled to an LTQ-Orbitrap[3], which was set to fractionate (and take MS^2 spectra of) the six most significant peaks of the MS spectra. In case of neutral loss, MS^3 spectra are taken. In addition, the unfractioned ($F = 1$) samples were also analyzed in an LTQ-FT.

6.2.2. In silico preprocessing

For the yeast shotgun proteomics data, raw spectra were preprocessed using the Thermo Finnigan libraries that come with the MS device software. The top 200 peaks picked from

[2] catalogue number YSC1048, derived from parent strain S228C

[3] Thermo Fisher Scientific

the MS^2 spectra are used for PFF identification via Mascot 2.104 using the *Saccharomyces* genome database (SGD, Cherry *et al.* (1997)). Search parameters included up to 2 missed cleavages, and the following modifications: carbamindomethyl on cysteine as fixed modification, as well as deamination up to two times, methionine oxidation, and N-terminal pyroglutamic acid formation as variable modifications. The sequence information the device software generates from the MS^2 spectra is discarded because the Mascot search engine does not use this information. An additional PFF identification was carried out using a database of reversed human protein sequences from IPI human[4]. By counting decoy hits at which a 1% false-positive rate is obtained, a cutoff score was determined, as proposed by Elias *et al.* (2005); Peng *et al.* (2003). Only peptides with a score above this cutoff have been retrieved.

All the data is collected in a database where the modifications found during the PFF identification, missed cleavages, peptide identification scores, and the relations within the data are readily available.

To extract intensities, the elution profile of each peptide was extracted within a 10 ppm window around its detected mass. From the profile, a Gaussian was matched with the Levenberg-Marquardt algorithm (Levenberg, 1944; Marquardt, 1963) to the elution profile peak to determine the area under curve (AUC). The algorithm is described in "Numerical Recipes in C" (Press, 1992). AUC values from different charge states and those from peptides being identified multiple times in a run of a sample are added. This results in one overall intensity per peptide per run and sample.

6.2.3. Dataset construction

If nothing else is denoted, peptides occurring in multiple fractions as well as peptides occurring anywhere in the whole dataset with modifications are excluded from the data set. If a peptide is a substring of a longer peptide with missed cleavages that was found in the same dataset, both are excluded. ESI produces multiply charged ions. Peptides with high masses lead to charged ions that may be outside the analyzed mass range. Because this data is normalized with the protein concentration *in the whole sample*, missing these portions of the peptide would be crucial. Therefore, only peptides with masses within 500 to 2000 Da are included in the final datasets. Table 6.2 shows a rough overview of the datasets.

[4] European Bioinformatics Institute (EMBL-EBI), `http://www.ebi.ac.uk/`

sample	fractions	device	replica	notes
Y_1	16	Orbitrap	1	
Y_1^+	16	Orbitrap	1	incl. peptides occurring in multiple fractions
Y_2	10	Orbitrap	5	
Y_2^+	10	Orbitrap	5	incl. peptides occurring in multiple fractions
$Y_{3(1),O}$	1	Orbitrap	4*	no or very few peptides were identified in three of the runs
$Y_{3(2),O}$	1	Orbitrap	4*	no or very few peptides were identified in three of the runs
$Y_3(1)$	1	LTQ-FT	5	
$Y_3(2)$	1	LTQ-FT	5	
$Y_3(2)F_5$	5	Orbitrap	7	
$Y_3(2)^+F_5$	5	Orbitrap	7	incl. peptides occurring in multiple fractions

Table 6.2: Overview of LC-ESI datasets.

6.2.4. Normalization for LC-ESI data

Protein abundances were measured with TAP-tagging (Ghaemmaghami *et al.*, 2003) during early exponential growth phase on the same *S. cerevisiae* yeast strain that was also used in this analysis. This technique usually has CVs of 200 to 300% (two to three fold variation). We normalize the intensities cumulated over the whole sample by dividing with these values. Peptides of proteins that could not be quantified by TAP-tagging are discarded. This enables us to normalize for different peptide abundances, thus extracting the peptide-specific sensitivities.

In analogue to the MALDI data (see Section 6.1.4), a α*-trimmed mean* ($\alpha = 50\%$) is calculated for replicate measurements of the same peptide. Unlike with the MALDI data, the median is taken even if only three measurements are available, because it is often observed that one of the values deviates from the other two.

6.2.5. Statistical analysis

A few things independent of the intensity values can be noted: The more runs of a sample, the better the proteome coverage (compare numbers for proteins and peptides in Table 6.3). This is not surprising for LC-ESI data since the analyte is undersampled with each run. In addition, the more fractions the sample is divided in prior to MS analysis, the better the coverage.

sample	fractions	device	runs	proteins	unique peptides	Mean CV	Median CV	r^2	Mean CV (log)	Median CV (log)	r^2 (log)
Y_1	16	Orbitrap	1	1235	2760						
Y_1^+			1	1351	4369						
Y_2	10	Orbitrap	5	1309	3641	8.393	15.88	0.104	1.13	2.246	0.764
Y_2^+			5	1378	5025	13.27	30.85	0.104	1.859	4.735	0.353
Y_2^C			4	1254	3486	21.44	15.44	0.09	3.469	2.158	0.757
$Y_3(1)$	1	LTQ-FT	4	454	1547	65.53	61.61	0.034	13.91	17.08	0.059
$Y_3(2)$			4	439	1558	72.82	70.15	0.071	18.16	26.04	0.027
$Y_3(1)^C$			3	439	1516	59.89	64.09	0.05	5.217	5.109	0.159
$Y_3(2)^C$			3	420	1496	62.97	67.91	0.09	16.65	15.31	0.184
$Y_3(2)F_5$	5	Orbitrap	7	842	2679	36.35	38.03	0.424	3.104	5.474	0.36
$Y_3(2)^+F_5$			7	870	3134	43.17	46.73	0.31	3.732	6.82	0.103
$Y_3(2)^CF_5$			5	814	2438	34.31	33.21	0.525	5.934	5.096	0.533
A (*mic*)		Ultraflex	n.a.	27	371	32.64	37.39	0.544	9.784	14.67	0.643
A (*sum*)		Ultraflex	n.a.	27	371	30.46	37.35	0.637	9.495	15.74	0.679
B (*mic*)		Ultraflex	n.a.	125	1023	31.17	36.05	0.091	7.352	10.55	0.350
B (*sum*)		Ultraflex	n.a.	125	1023	26.27	30.64	0.540	6.214	9.165	0.572

Table 6.3: Reproducibility statistics of all datasets. For the orbitrap runs of the unfractioned $Y_3(2)$ sample, three runs did lead to no or very few identified peptides due to bad MS^2 spectra. After analysis of intensity distributions from different runs of the same samples, bad runs were discarded, leading to cleansed datasets denoted with a superscript C. The datasets that are primarily used for the prediction of peak intensities are highlighted in gray.

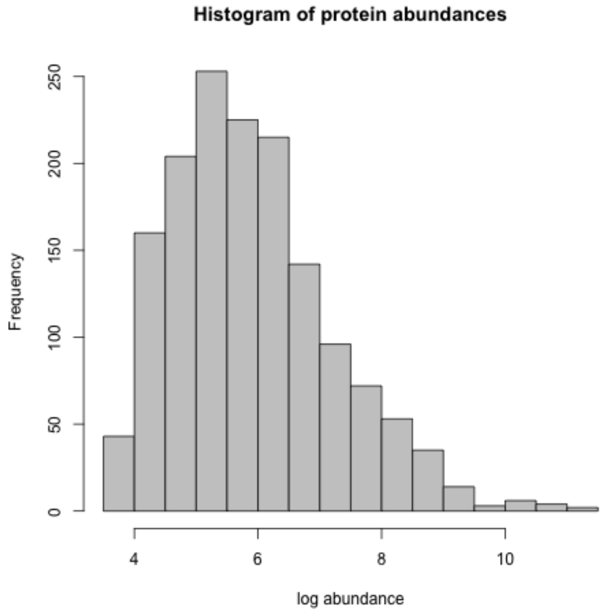


Figure 6.8: Histogram of logarithmic protein abundances that were found with TAP tagging (Western).

Analysis is only carried out on the datasets that have replicate measurements. Thus, Y_1, $Y_{3(1),O}$, and $Y_{3(2),O}$ are excluded. Of the remaining sets, Y_2 has the most peptides (3641), $Y_3(2)$ also contains quite a lot (2679), $Y_3(1)$ and its biological replicate ($Y_3(2)$) are smaller (about 1500). For this work, each peptide constitutes a data point or example for the machine learning methods. Thus for datasets of the same quality, larger is better. Not every peptide is found in all replicate runs. An overview of the numbers of replicates can be seen in Fig. 6.10. In general, about half of the peptides in a dataset have been measured only once. There is no correlation between the number of runs a peptide was found in and its mean log intensity for these datasets.

The overall distribution of mean logarithmic intensities are generally close to normally distributed (Fig. 6.14). Spectra taken on the LTQ-FT have a different (lower) range of values than those from the orbitrap (Fig. 6.11). This does not allow any inference about the devices' sensitivities, but just has to be kept in mind when doing across-machine predictions. Apart from that, the overall distributions between different samples analyzed on the same device are quite similar.

The log-intensity distribution of replicate runs is visualized in Fig. 6.9. These plots allow the visual detection of outlier runs, i.e. runs whose distribution is untypical or shows a systematic shift. This indicates a device malfunction or other error in the analysis. It can be noted that these runs also have a significantly reduced number of identified peptides compared to other runs from the same sample (Table 6.4). The distribution of intensities depends more on the time of the analysis than on the sample itself. Based on these findings, the following runs are excluded from further analysis:

original dataset	**excluded runs**	**new dataset**
Y_2, Y_2^+	10	Y_2^C, $Y_2^{+,C}$
$Y_3(1)FT$	46	$Y_3(1)^CFT$
$Y_3(2)FT$	50	$Y_3(2)^CFT$
$Y_3(2)F_5$, $Y_3(2)^+F_5$	38, 41	$Y_3(2)^CF_5$, $Y_3(2)^{+,C}F_5$

We denote the cleansed datasets with a superscript C. Analysis of the reproducibility shows that removal of untypical runs reduces the median CV and increases the correlation between replicate peptide intensities (overview Table 6.3) except for Y_2. Thus, for Y_2 the original dataset is kept.

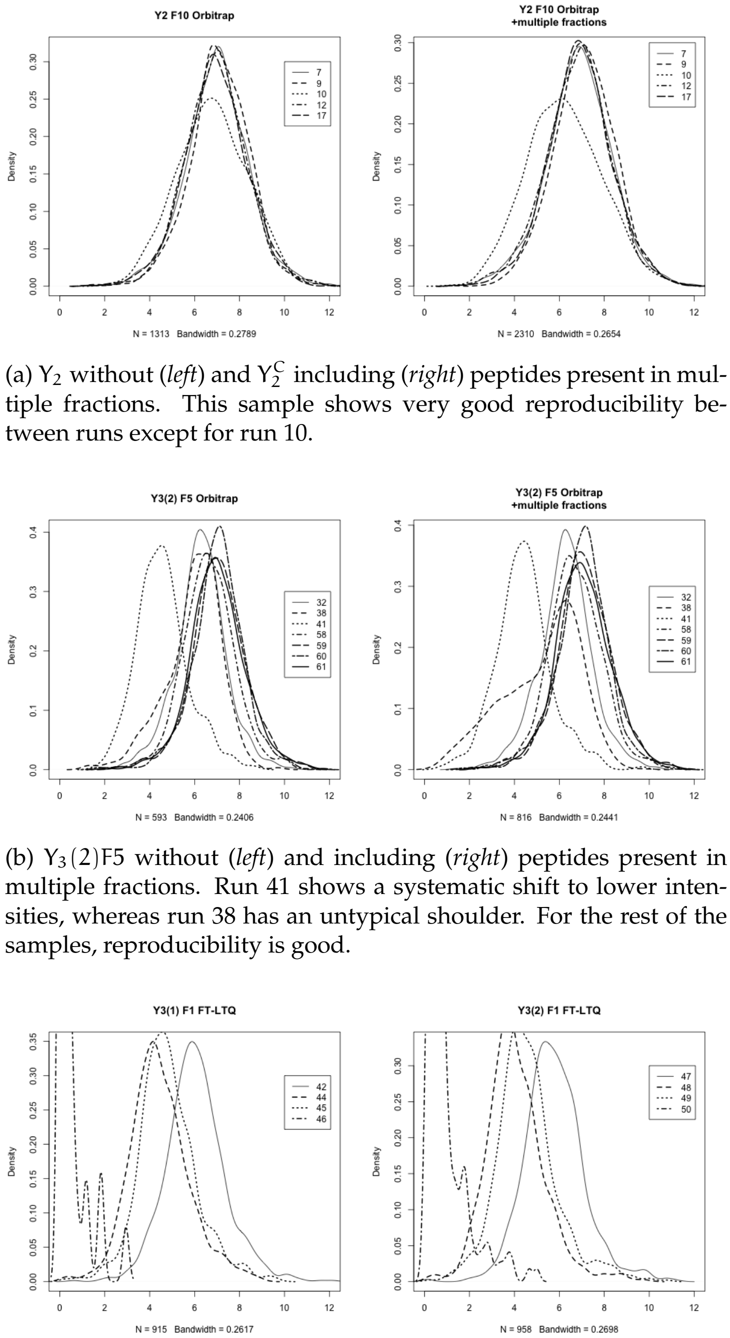
Squared Pearson's correlations between replicate runs of the cleansed datsets are between $r^2 = 0.16$ and $r^2 = 0.76$. Scatter plots between log-intensities of replicate measurements per peptide are shown in Fig. 6.12. In spite of some characteristic differences between the samples, general trends can be noted: Augmentation of datasets with peptides that occur in multiple runs (denoted by a superscript +) increases the variance. These augmented datasets profit most from the cleansing, but they still show a larger spread than those that consist exclusively of peptides that were only found in one of the fractions. This indicates that a normalization between fractions might further increase the reproducibility.

Y_2 shows the best reproducibility in terms of correlation, that of $Y_3(2)^CF_5$ is intermediate and $Y_3(n)FT, n \in 1, 2$ is worst. This also reflects in the CV values (Fig. 6.13).

6.3. Summary

The MALDI datasets are really small for a machine learning application (371 and 1023 data points for dataset A and B respectively) and have high variance. A lot of peptides have been measured only once such that there is no variance-stabilizing effect for these. Dataset A has a lower variance then B and a higher portion of peptides with more than one measurement.

Of the ESI datasets, Y_2 shows very good reproducibility and is of medium size (3641 data points). Only this data set is considered further because it is largest and have the lowest variance. The others show a higher variance even after removal of untypical runs. It can be observed that within the LC-ESI datasets, reproducibility increases with the number of fractions, i.e. with decreasing sample complexity. Centering and normalization by variance of the individual runs belonging to different gel fractions could improve the variance between runs of different fractions. Another possibility is the use of standard peptides to normalize between fractions. We tried this using bovine serum albumin (BSA) that was spiked into each



(a) Y_2 without (*left*) and Y_2^C including (*right*) peptides present in multiple fractions. This sample shows very good reproducibility between runs except for run 10.

(b) $Y_3(2)F5$ without (*left*) and including (*right*) peptides present in multiple fractions. Run 41 shows a systematic shift to lower intensities, whereas run 38 has an untypical shoulder. For the rest of the samples, reproducibility is good.

(c) $Y_3(1)F1$ (*left*) and its biological replicate (*right*) analyzed with the LTQ-FT. Run 46 and 50 have very low values in general and an untypical distribution. The sample might have been depleted. The reproducibility of the other samples is fair.

Figure 6.9: Distributions of log intensities for replicate runs for all LC-ESI datasets.

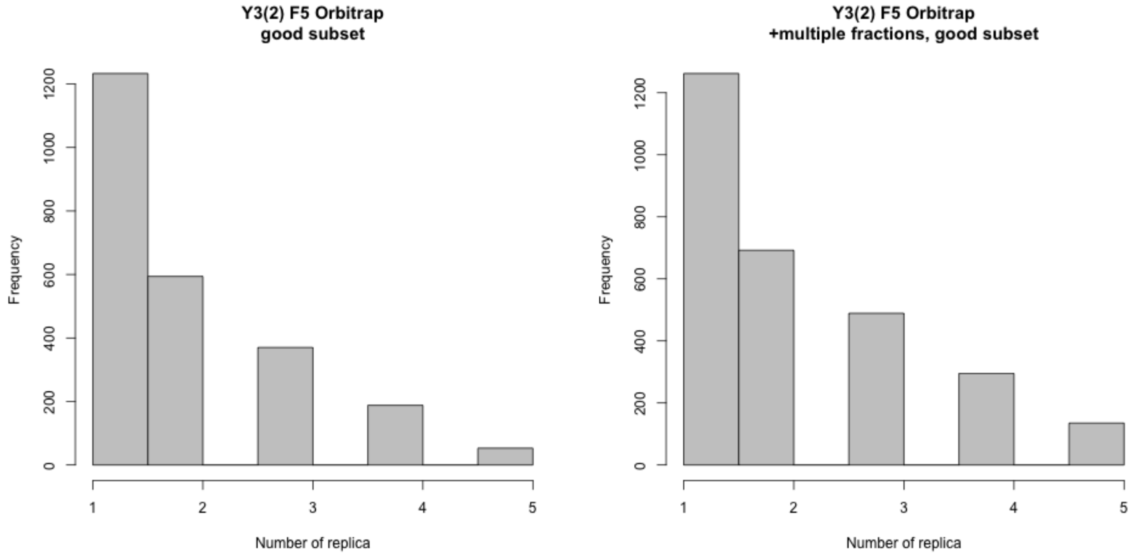


Figure 6.10: Typical histograms of the number of replicate peptide measurements for LC-ESI datasets without (*left*) and including (*right*) peptides present in multiple fractions. This example shows dataset $Y_3(2)^C$F5.

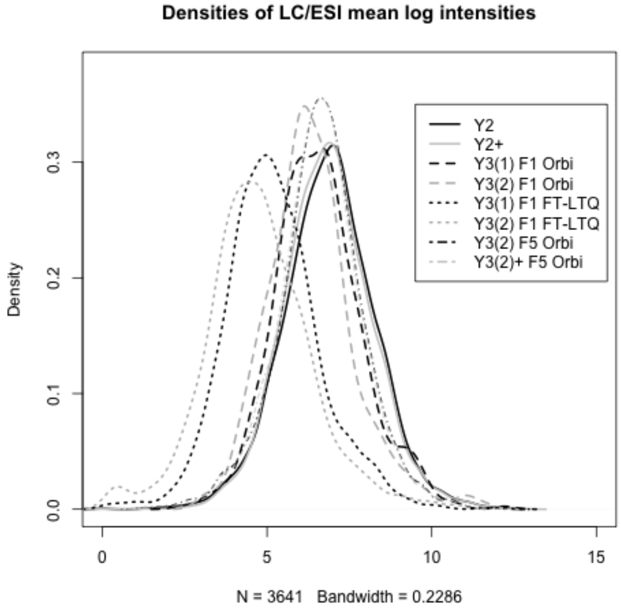
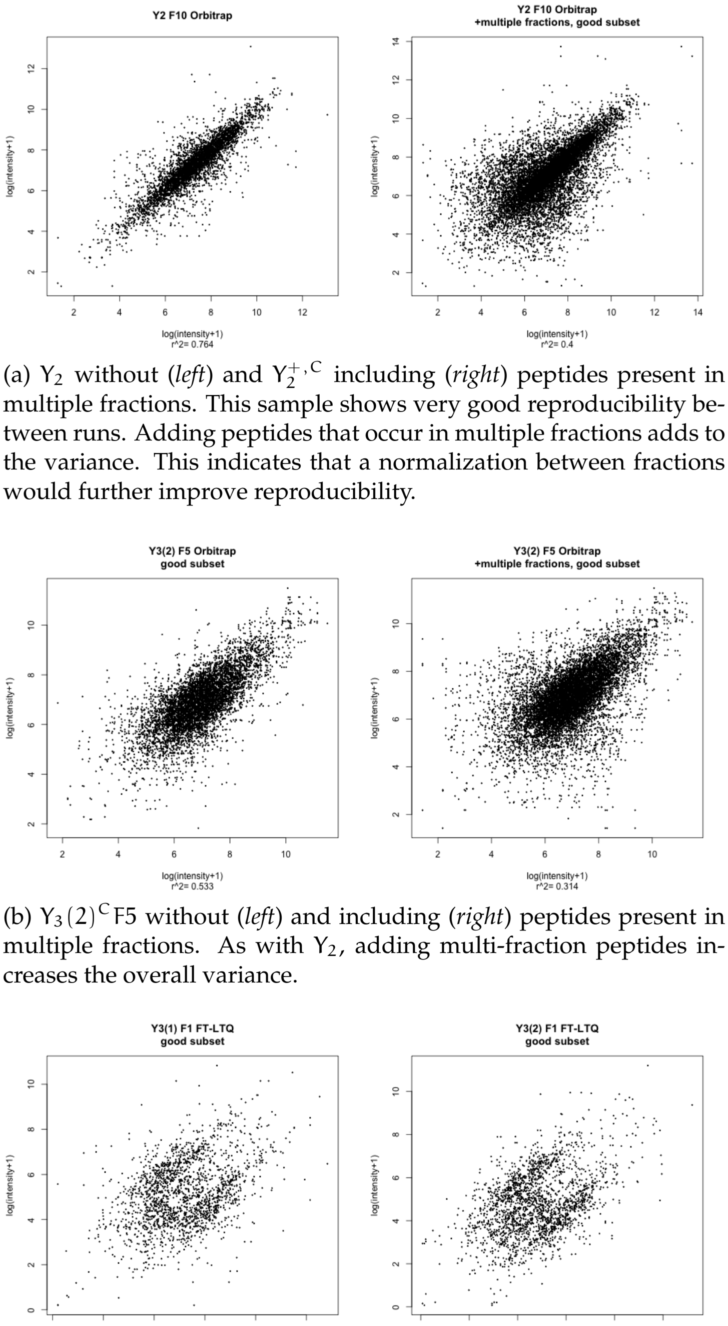


Figure 6.11: Distribution densities of mean log intensities for all peptides in all samples analyzed with LC-ESI. The x-axis shows mean log intensities, the y-axes shows the frequency as an estimated density. The intensities from the FT-LTQ are generally in a lower range. The runs from the orbitrap show good agreement of their overall intensity distributions.

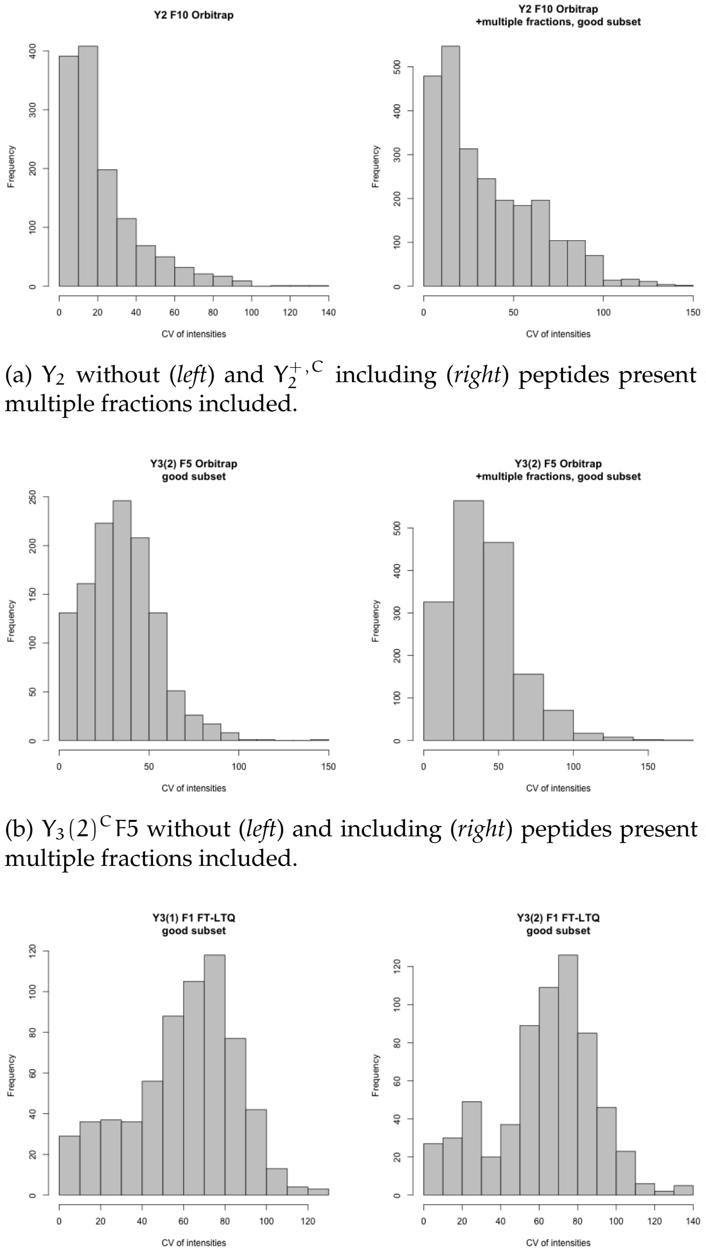


(a) Y_2 without (*left*) and $Y_2^{+,C}$ including (*right*) peptides present in multiple fractions. This sample shows very good reproducibility between runs. Adding peptides that occur in multiple fractions adds to the variance. This indicates that a normalization between fractions would further improve reproducibility.

(b) $Y_3(2)^C F5$ without (*left*) and including (*right*) peptides present in multiple fractions. As with Y_2, adding multi-fraction peptides increases the overall variance.

(c) $Y_3(1)^C F1$ (*left*) and its biological replicate (*right*) analyzed with the LTQ-FT.

Figure 6.12: Scatter plot of log intensities of the same peptides in replicate runs after removal of bad runs. The value of r^2 denotes the squared Pearson's correlation.



(a) Y_2 without (*left*) and $Y_2^{+,C}$ including (*right*) peptides present in multiple fractions included.

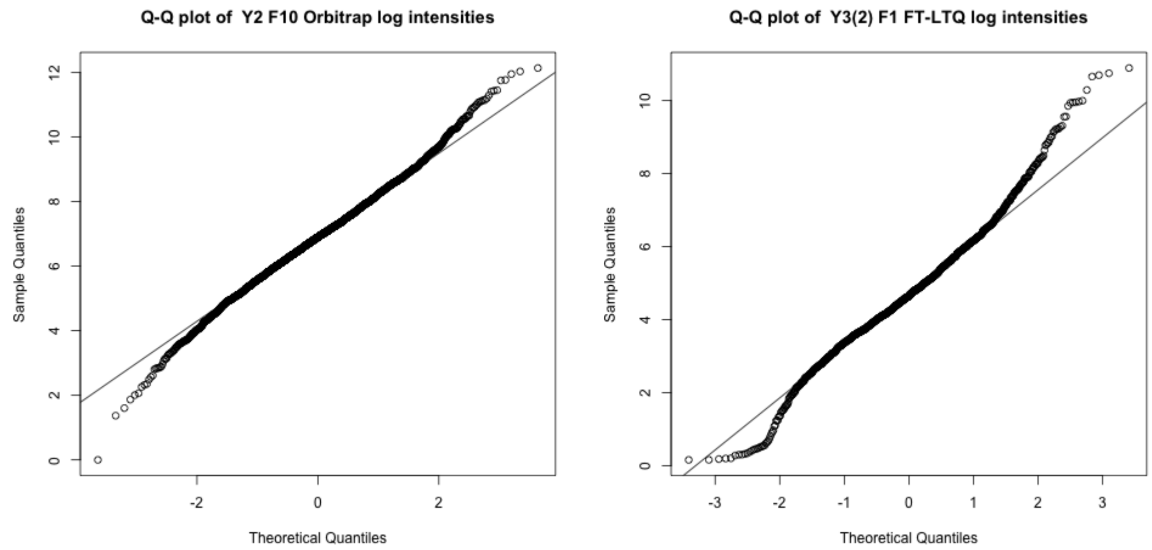
(b) $Y_3(2)^C$F5 without (*left*) and including (*right*) peptides present in multiple fractions included.

(c) $Y_3(1)^C$F1 (*left*) and its biological replicate (*right*) analyzed with the LTQ-FT.

Figure 6.13: Histograms of coefficients of variance (CV) between replicate runs for all LC-ESI datasets after removal of bad runs.

Sample	Run ID	No. of identified peptides
Y_2	7	1313
	9	1407
	10	356
	12	1080
	17	1279
Y_2^+	7	2310
	9	2328
	10	836
	12	1913
	17	2256
$Y_{3(1)_O}$ F1 (orbitrap)	19	16
	20	486
$Y_{3(2),O}$ F1 (orbitrap)	22	24
	23	291
$Y_3(1)$ F1 (FT-LTQ)	42	916
	44	644
	45	771
	46	37
$Y_3(2)$ F1 (FT-LTQ)	47	958
	48	712
	49	692
	50	174
$Y_3(2)$ F5	32	593
	38	428
	41	191
	58	997
	59	945
	60	873
	61	1141
$Y_3(2)^+$ F5	32	816
	38	712
	41	285
	58	1287
	59	1225
	60	1168
	61	1473

Table 6.4: Number of identified peptides for different runs. Runs which show a shift from the distribution of the respective other runs (see Fig. 6.9) have a significantly lower numbers identified. This applies to all runs whose numbers are in italics. These runs should be discarded or normalized to fit the other distributions.



(a) Y_2 F10 Orbitrap

(b) $Y_3(2)$ F1 LTQ-FT

Figure 6.14: Q-Q plots of mean log intensities visualize whether data follows a normal distribution. For an ideal normal distribution, data would lie on the thin line. These shapes are typical for the LC-ESI datasets produced for this work. For the Y_2^C dataset, which shows good reproducibility between runs (see Fig. 6.9), the data follows the normal distribution quite well (*left*). For datasets with more variation between runs as for example $Y_3(2)$ F1 taken on the LTQ-FT, more deviation can be noticed.

fraction's sample prior to digestion. However, no BSA peptide could be identified in *all* of the fractions.

It is widely assumed that ESI is more reproducible than MALDI. In a direct comparison, median CVs of the MALDI datasets are higher than for most of the cleansed LC-ESI datasets. However, the squared correlation between replicate measurements of dataset A is between those of the two LC-ESI datsets with the best reproducibility. The LC-ESI datasets are significantly larger than the MALDI datasets and span twice the range of log-intensities.

7. Peak intensity prediction

This chapter describes the feature extraction from peptide strings and the second part of the modeling: the prediction of the normalized peak intensity values with the supervised learning algorithm ν-SVR.

The encoding of the peptide strings used as input vectors $\mathbf{x}_i$ for the learning algorithms is critical to the success of the model. In this work a top-down approach is used: Extract many features initially and then find a good subset with feature selection methods.

First, I present four sets I designed, and describe their extraction. Then, a short analysis of these sets follows. The last part of this chapter shows prediction results achieved with ν-SVR on these feature sets.

7.1. Representation of Peptides

To extract features in the first place, two sources of information are used: The peptide sequence itself and chemical properties of single amino acids.

7.1.1. Computer scientist's paradigm: Peptides are strings

Monomers feature set (*mono*) This feature set contains only single counts of all single amino acids in a given peptide, yielding 20-dimensional feature vectors. This type of encoding has been used before as part of larger feature sets for the prediction of peptide properties (Bonner and Liu, 2004; Gay *et al.*, 2002; Li *et al.*, 2000; Vucetic *et al.*, 2001)

Sequence feature set (*seq*) To additionally incorporate information about the order of amino acids, we extract the number of single amino acids, pairs, and triples from the peptide sequence. In addition, the terminal pairs on both ends of the peptide are binary encoded and added to the vector. They have been shown to have an effect on peak-intensities (Krause *et al.*, 1999; Yang *et al.*, 2008). This yields a 9220-dimensional very sparse vector.

7.1.2. Biochemist's paradigm: Peptides are molecules

Chemical feature set (aa) For this set we use physicochemical information of the amino acids constituting the peptide. Amino acid attributes are taken from the amino acid index database (Kawashima *et al.*, 1999). Mallick *et al.* (2007) also use amino acid indices to predict proteotypic peptides. Each amino acid index $AA = (AA_1, \ldots, AA_{20})$ consists of twenty values each belonging to one specific amino acid. Let $m(s)$ be a mapping of the amino acids character s to its position in the index. For a given peptide with sequence $S = (s_1, \ldots, s_n)$ of length n, the value for one single feature f is calculated as the sum of the amino acid's values for that index: $\Sigma^{AA}(p) = \sum_{k=1}^{n} AA_{m(s_k)}$. There are 516 indices in the database, therefore, for each peptide, we calculate 516 such features. In addition, other included features are peptide length, mass, and numbers and fractions of acidic, basic, polar, aliphatic, and arginine residues. Three values that describe the gas-phase basicity are added to the vector: a) The estimated gas-phase basicity is calculated as proposed by Zhang (2004) as well as b) a sum over the residual values of the amino acids that were used for this estimation, and c) that sum scaled with the length of the peptide. Overall, this feature set is 531-dimensional.

The gas-phase basicity is defined as the negative of the Gibbs energy change (ΔG_r^o) associated with the reaction:

$B + H^+ \rightarrow BH^+$

in the gas phase. It is also called absolute or intrinsic basicity. Here, H denotes a proton, B a molecule acting as a base (i.e. taking in a proton). ΔG_r^o is a measure for the energy change of the reaction in a closed system. A negative value indicates that the reaction favors the direction where the product (to the right side of the arrow) accumulates. Therefore, the higher the gas-phase basicity, the more energetically favored the product is. In other terms, the higher the gas-phase basicity, the more of the product ion BH^+ will form. This is a simplification. To learn more about thermodynamics, Atkins and Paula (2006) can be recommended.

The gas-phase basicity constitutes an important measure for peptides in mass-spectrometry because in positive ion mode, ionized molecules are added a proton, resulting in a BH^+ species where B is the neutral peptide molecule. Harrison (1997) wrote a review about the gas-phase basicity and proton affinity of amino acids and peptides.

Expanded amino acid index features (*cac*) There are indications that the peak intensity does not only depend on amino acid frequencies but also on their order. We cannot encode the exact order of amino acids because the data space and therefore the necessary training data set size would explode. This "chem and counts" feature set (*cac*) takes the idea of the summed amino acid indices further by using descriptors in the spirit of topological descriptors (Gasteiger and Engel, 2003). These descriptors make use of properties of single atoms and their topology in a chemical compound. However, I do not use descriptors of atoms directly, because amino acids are very similar in structure and contain a lot of atoms. Instead, descriptive values (amino acid index values) for whole amino acids are used.

For each amino acid index AA, seven descriptive values are calculated per peptide p of length n, with amino acid index values $(AA_{m(1)}, AA_{m(2)}, AA_{m(n)}) := (p_1, p_2, \ldots, p_n)$, constituting $7 \cdot 516 = 3612$ features:

- The absolute differences of AA between amino acids that have a distance d in the peptide string. This is calculated for $d \in 1, 2, 3$. For $d = 1$ this corresponds to the sum of absolute first derivatives of the amino acid index values in the discrete case.

$$D_d^{AA}(p) = \sum_{j=1}^{n-1} \left(|(p_j^{AA} - p_{j+d}^{AA})| \right) := \sum_{j=1}^{n-1} f_j^{AA} \tag{7.1}$$

- The discrete absolute second derivative represents a smoothness measure for the amino acid index AA:

$$S^{AA}(p) = \sum_{j=1}^{n-2} (|f_j - f_{j+1}|) \tag{7.2}$$

- The absolute difference between lowest and highest amino acid index value:

$$M^{AA} = |\min\{p_1^{\alpha}, \ldots, p_n^{\alpha}\} - \max\{p_1^{\alpha}, \ldots, p_n^{\alpha}\}| \tag{7.3}$$

- The summed amino acid index values $\Sigma^{AA}(p)$, as described in Section 7.1.2.
- The average $\mathrm{Avg}^{AA}(p)$: The summed value divided by the peptide's length.

These values are calculated for the peptide sequence expanded by the two amino acids adjacent in the originating protein sequence.

From the *aa* feature set, the length, mass, and estimated gas-phase basicity are included in addition to the summed amino acid index values. Fourteen special counts and all twenty amino acid frequencies are added to the vector. Overall, this feature set is 3649-dimensional.

Special counts Each special count feature consists of the number of amino acids in a peptide belonging to a certain group. The groups are itemized below. The group membership of the amino acids is shown in Section A.4 of the appendix. More precise explanations can be found in a chemistry textbook (for example Vollhardt and Schore 2002).

- **aromatic** Organic compounds are divided into aromatic and aliphatic ones. Aromatic compounds are those that contain delocalized ions, which results in a special behavior.

- **aliphatic nonpolar** Aliphatic compounds are organic compounds that are not aromatic.

- **acidic** Acidic molecules are likely to donate a proton to other compounds.

- **basic** Basic molecules are likely to accept a proton from other compounds.

- **polar** Polar molecules are those that have slightly differently charged ends because of differences in the electronegativity of their atoms. More electronegative atoms draw electrons more strongly than less electronegative ones. This results in a certain chemical behavior that is analogues to the behavior of small magnets. Water molecules are polar, for example.

- **small polar** Small molecules that are polar.

- **most flexible** Molecules are not totally rigid. The bonds between atoms retain an overall structure, but groups may turn around their bonds. Some molecules are more flexible than others. This feature counts the most flexible amino acids.

- **least flexible** Count of the most rigid (= lest flexible) amino acids.

- **sheet formers** Proteins form a secondary structure (see 2.1). Sheet formers are those amino acids that are more likely to be present in the beta sheet secondary structure.

- **sheet breakers** In analogue to sheet formers, sheet breakers are those amino acids that are less likely to be present in a beta sheet.

- **PEVK** (P+E+V+K) This count has been one of the features used for the prediction of disordered proteins (Romero *et al.*, 2001).

- **charge count** This counts the charges of the whole peptide. There are different ways to count the overall charge of a peptide: a) (K+R+H+D+E)/2, b) K+R+H-E-D, c) K+R - (E+D). All three are included in the feature vector.

7.2. Statistical properties of the feature spaces

7.2.1. Correlated features

An analysis of the correlation between features for dataset A reveals that the *aa* and *cac* feature sets contain a lot of highly correlated features ($r^2 > 0.8$). Specifically, a lot of the amino acid index features from *aa* are highly correlated with length and mass, because values are summed over all amino acids. Also, a number of features are correlated to the number of aliphatic residues. The number of arginine residues (R) is correlated to the estimated gas-phase basicity (GB500) ($r^2 = 0.94$). Other correlated features in aa are listed in the original amino acid index database (Kawashima *et al.*, 1999). It is important to know that there are a lot of highly correlated features within *aa* and *cac* because these will affect feature selection in these sets.

7.2.2. Frequency of dimers and trimers

The *seq* feature set is designed to partly capture the order of the peptide sequence. This results in its high dimensionality and sparseness, since a single tryptic peptide can only contain a fraction of the possible dimers or trimers. In a small dataset, such as the MALDI datasets used in this work, there are dimers and trimers that do not occur at all or only once. Figure 7.1 shows a histogram of the frequencies in dataset A. Most trimers and even dimers do not occur at all. The LC-ESI dataset Y_2 is almost ten times larger than A. In Y_2, almost all dimers occur at least once, but about half of the trimers (2857 of 8000) do not occur in any of the peptides (Fig. 7.2).

A much larger dataset would be needed for this feature set to develop its potential, that is, to actually capture information about the order of amino acids.

7.3. Prediction

7.3.1. Methods

I apply ν-SVR with a Gaussian kernel, using the extracted feature sets as they are. In this work, features are always centered and normalized by variance before application of any machine learning method. A rough grid search in parameter space is performed: ν is sampled in steps of 0.1 within $[0.2, 0.8]$, $C \in \left[e^{-5}, e^{9}\right]$ and $\gamma \in \left[e^{-7}, e^{3}\right]$, both in steps of e^2.

The prediction performance is measured by the squared Pearson's correlation (r^2), the root mean squared error ($RMSE$), and also inspected visually in scatter plots. These plots show the

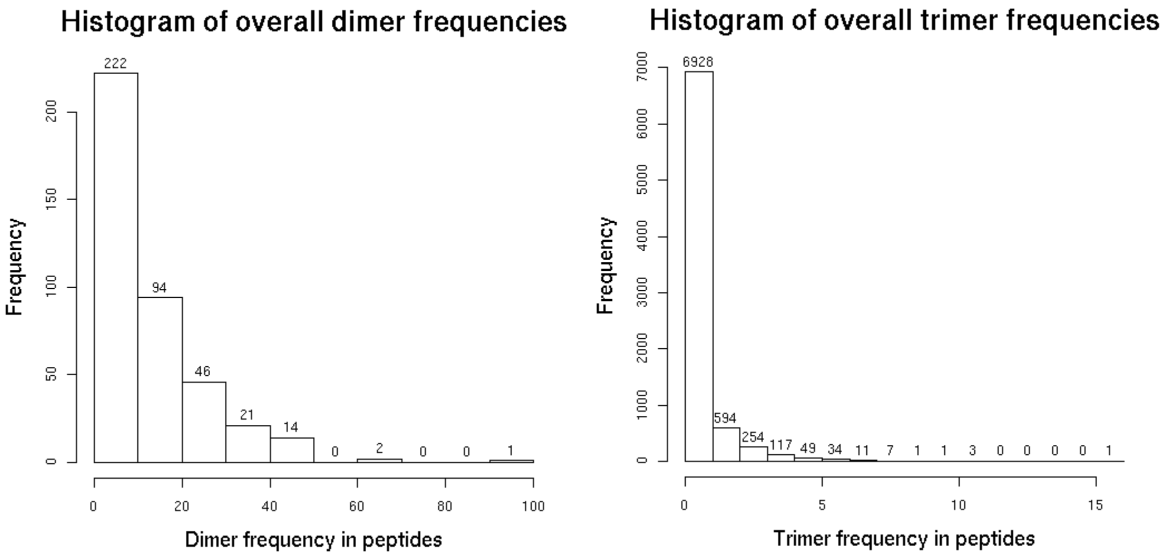


Figure 7.1: Number of times dimers *(left)* or trimers *(right)* occur in the sequence feature set of A. While a good portion of the dimers occurs more often then ten times in the whole dataset, most of the trimers do not show up at all or just once. In principle, the sequence feature set captures some of the order of amino acids in the peptide. However, considerably more data is necessary for this to be useful for the prediction.

relationship between target and predicted values. For a perfect prediction, the relation should be linear, thus the scatter plot should show dots on the diagonal. The strength of the linear relation is measured by r^2, while the tilt from the diagonal increase the RMSE.

Since it is infeasible to inspect all possible models visually, the best parameter set is chosen by the highest mean r^2 over all test sets in ten-fold cross-validation. If a model had a perfectly linear (i.e. $r^2 = 1$) relationship between predicted and target values, and does not show a diagonal line in the scatter plot, the point cloud could still be tilted by appropriate normalization, and thereby lower the RMSE. In most cases, the model with the highest r^2 also had the lowest RMSE.

For across dataset validation, the model trained and parameter-tuned on one of the datasets is used to predict the target values of the other dataset respectively. For the strict validation, peptides that occur in both A and B are omitted from the validation set.

First, results on the MALDI datasets are presented, then the results of first experiments with the LC-ESI datasets are shown.

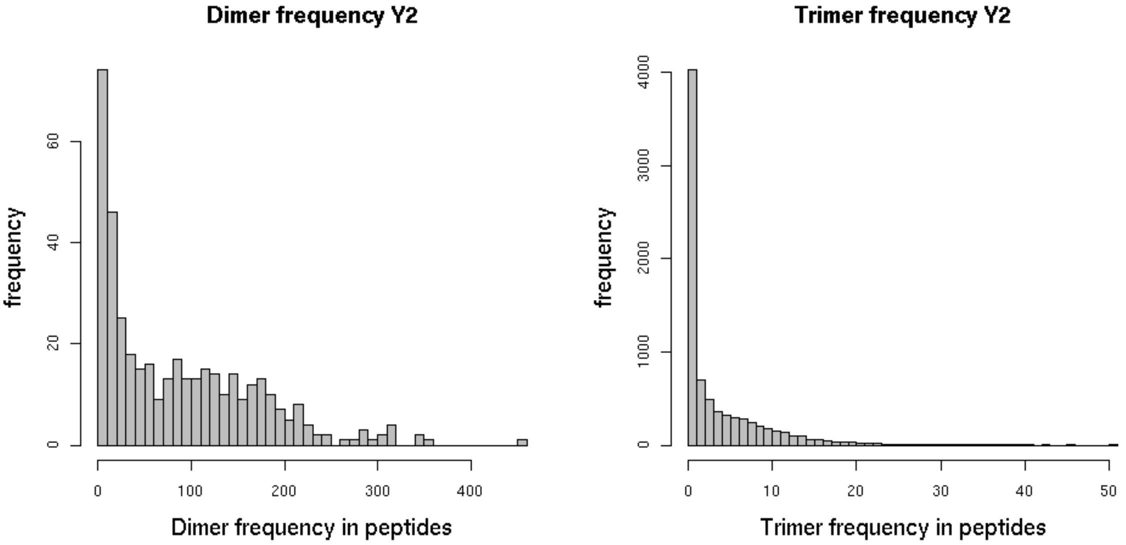


Figure 7.2: Number of times dimers *(left)* or trimers *(right)* occur in the sequence feature set of Y_2. Almost all dimers occur at least once, but about half of the trimers (2857 of 8000) do not occur in any of the peptides.

7.3.2. MALDI dataset prediction results

The *mic* normalization generally has a slight advantage over *sum* normalization while the other trends are identical for both. Therefore, only the *mic* normalization is used in the following extended analysis and feature selection.

Among all combinations, the **best result** of the initially extracted feature sets is achieved with the *mono* feature set on dataset A. Here, 10-fold cross-validation yields an overall squared Pearson's correlation of $r^2 = 0.46$. In the across dataset validation, the correlation coefficient is only slightly reduced to $r^2 = 0.44$, or $r^2 = 0.37$ for the strict validation case.

Comparison of feature sets. The chemical (*aa*) feature set works almost as good. The *seq* feature set shows the worst result ($r^2 = 0.32$ in the 10-fold cross-validation), and *cac* is in between. Standard deviations within the ten test sets during cross-validation are between $\sigma_{r^2} = 0.058$ and 0.126 for the squared correlation coefficient and between $\sigma_{RMSE} = 0.072$ and 0.147 for the $RMSE$ (*mono* and *aa* feature sets). **These correlations are significant and show that peak intensities can be predicted with statistical learning methods.**

The bad performance of the sparse 9220-dimensional *seq* feature set is probably due to the high dimensionality compared to the much lower number of data points. While in principle this feature set captures partial information about amino acid order, it seems inappro-

mic normalization				
validation dataset	**feature set**	**prediction performance** [$r^2(\mathrm{RMSE})$]		
		cross-validation	across dataset validation	across dataset strict validation
A	*mono*	0.46 (0.989)	0.44 (1.251)	0.37
	seq	0.32 (1.124)	0.21 (1.548)	0.14
	aa	0.44 (1.013)	0.42 (1.274)	0.34
	cac	0.37 (1.057)	0.36 (1.287)	0.23
B	*mono*	0.22 (1.178)	0.20 (1.366)	0.21
	seq	0.19 (1.194)	0.10 (1.430)	0.10
	aa	0.27 (1.114)	0.20 (1.543)	0.20
	cac	0.27 (1.112)	0.22 (1.343)	0.20

sum normalization				
validation dataset	**feature set**	**prediction performance** [$r^2(\mathrm{RMSE})$]		
		cross-validation	across dataset validation	across dataset strict validation
A	*mono*	0.38 (1.048)	0.43 (1.090)	–
	seq	0.27 (1.143)	0.26 (1.349)	–
	aa	0.41 (1.029)	0.38 (1.126)	–
B	*mono*	0.21 (1.148)	0.17 (1.214)	–
	seq	0.18 (1.181)	0.10 (1.280)	–
	aa	0.27 (1.095)	0.18 (1.368)	–

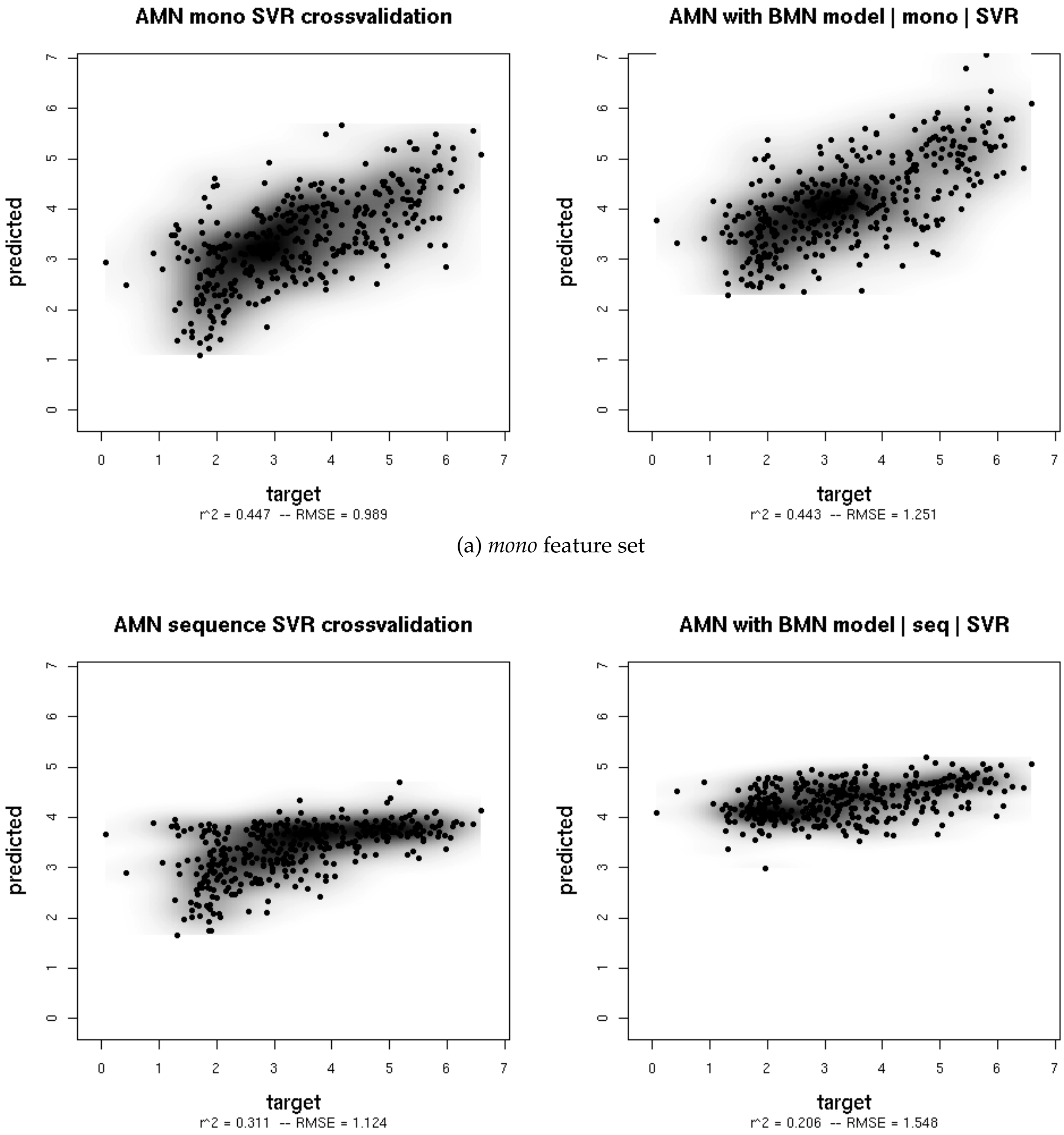
Table 7.1: Prediction accuracy results for the extracted features sets on both MALDI datasets with the ν-SVR. r^2 denotes the squared Pearson's correlation, RMSE is the root of the mean squared error. "–" denotes that the corresponding result has not been assessed. In general, the correlation values are slightly better for the mic normalization. The *mono* feature set on dataset A yields the best prediction accuracy.

priate for the small MALDI training datasets. This is a general problem: There are indications that not only the amino acid frequencies but also their order determines peak intensities. However, the information necessary to comprise this relationship explodes. Even if the amino acid order is encoded only partially, as in this case, much more training data is needed.

A vector consisting only of gas-phase basicity values calculated for eight different temperatures was tested as well, but the prediction results are very poor because the gas-phase basicity values at different temperatures are highly correlated.

Scatter plots. The scatter plots of the *mono* and *aa* feature sets are very similar to each other (figures 7.3, 7.4 for A, 7.5, and 7.6 for B). The cross-validation plots show a point cloud that is almost diagonal and shows considerable spread especially for low values. The across dataset prediction plot shows that A is predicted systematically too high if the model trained and parameter-tuned on dataset B is used. This is not surprising, since B has a distribution around slightly higher values than A. An additional centering and normalization by variance of the target values improves this: It lowers the RMSE to that of the cross-validation while the correlation coefficient is unchanged.

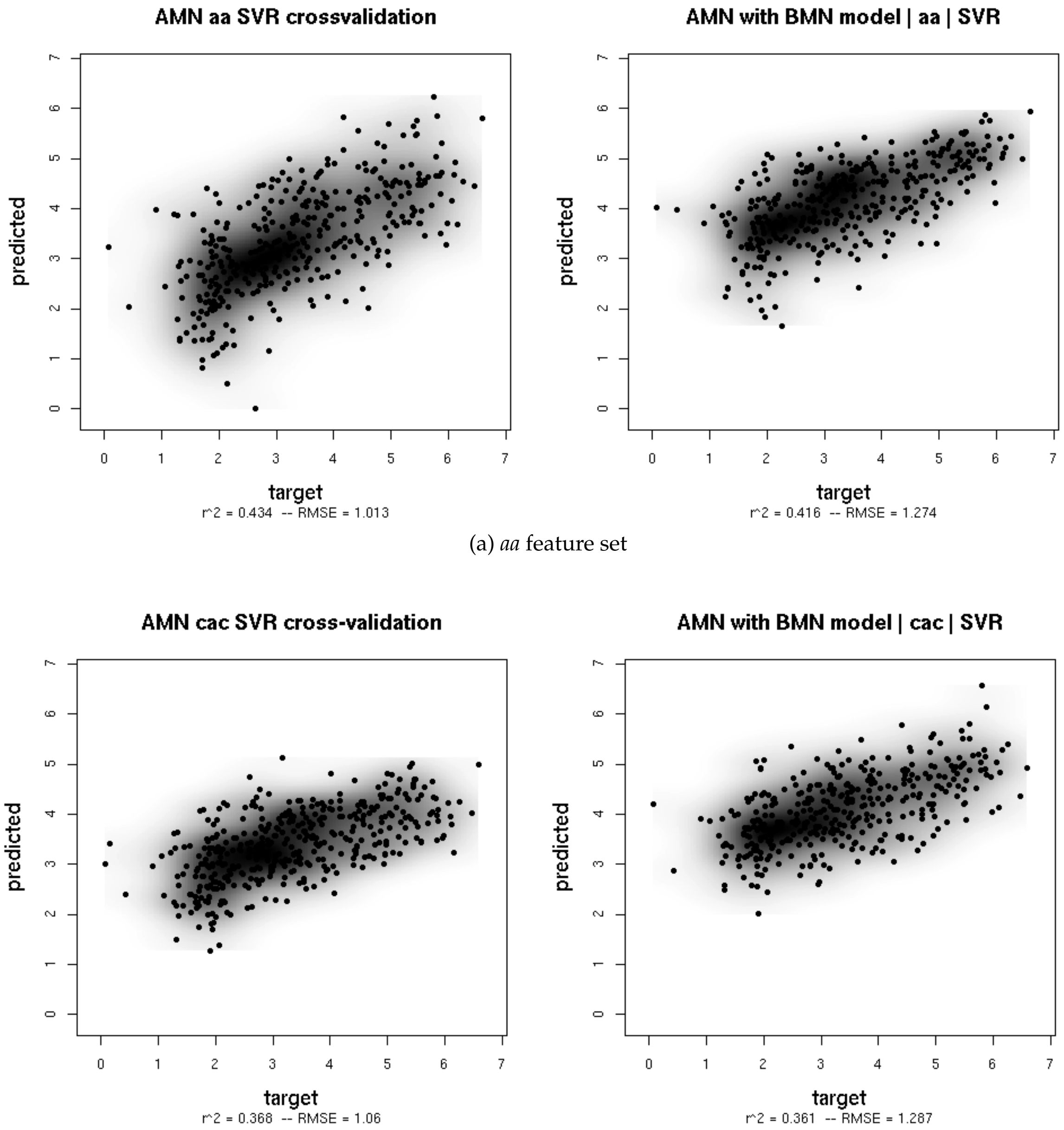
Comparison of datasets. Generally, dataset A gives much better results than B, although the latter is larger. The obvious reason is the higher within-peptide variance of normalized intensities and the higher fraction of peptides without replicate measurements in dataset B. Nonetheless, the ν-SVR is able to draw the main trends from B, since A can be predicted with a model trained on B even better than B itself.



(a) *mono* feature set

(b) *seq* feature set

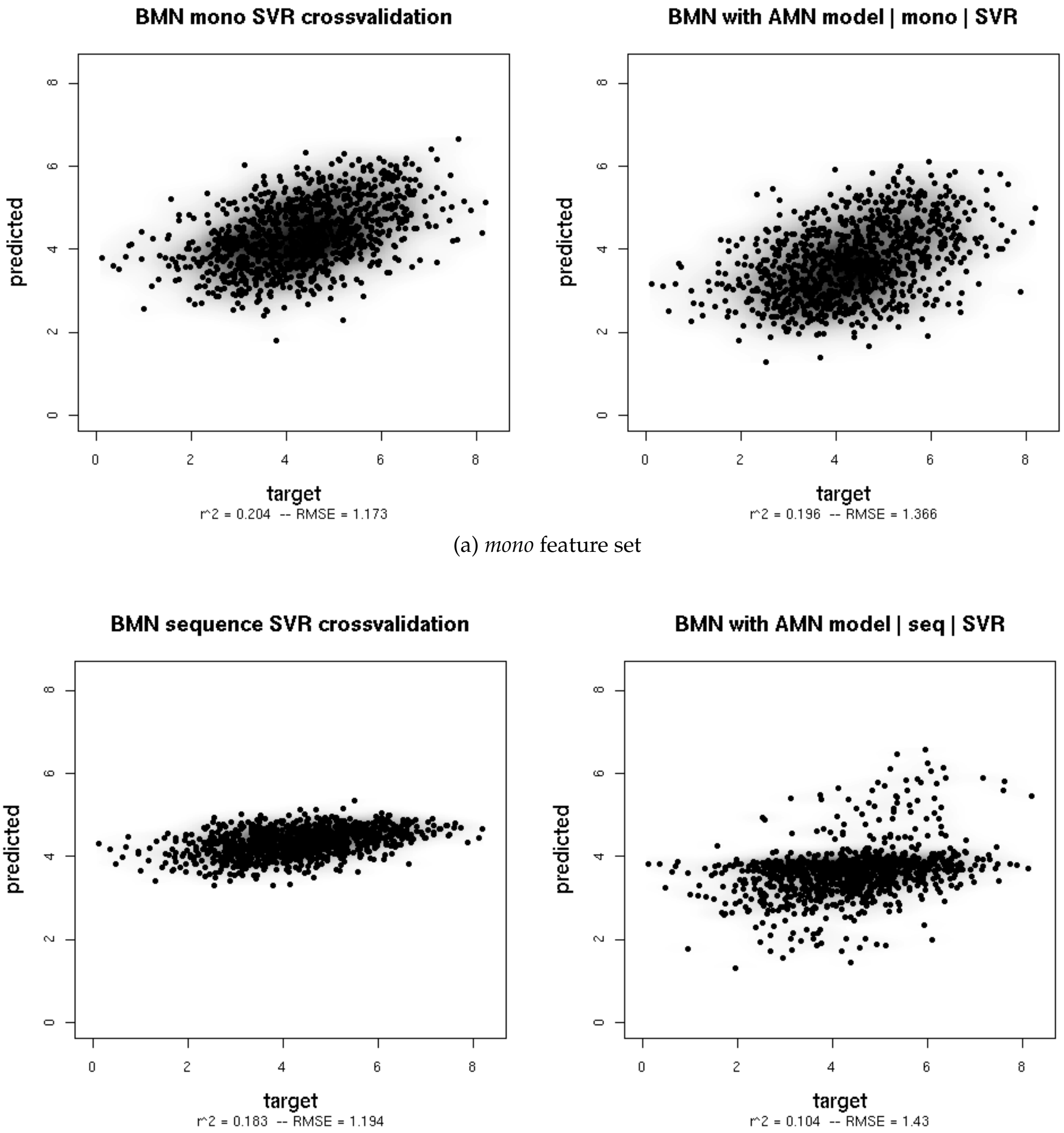
Figure 7.3: Scatter plots of the prediction results for string-based feature sets of dataset A (*mic* normalization) predicted with ν-SVR. Target values are plotted against predicted values (*left*: cross-validation, *right*: across dataset prediction.) The scatter plots have been overlayed by a two-dimensional density estimation. Dark gray values denote a high density. r^2 denotes the squared Pearson's correlation, RMSE is the root mean squared error. Ideally, the points would lie on a diagonal line.



(a) *aa* feature set

(b) *cac* feature set

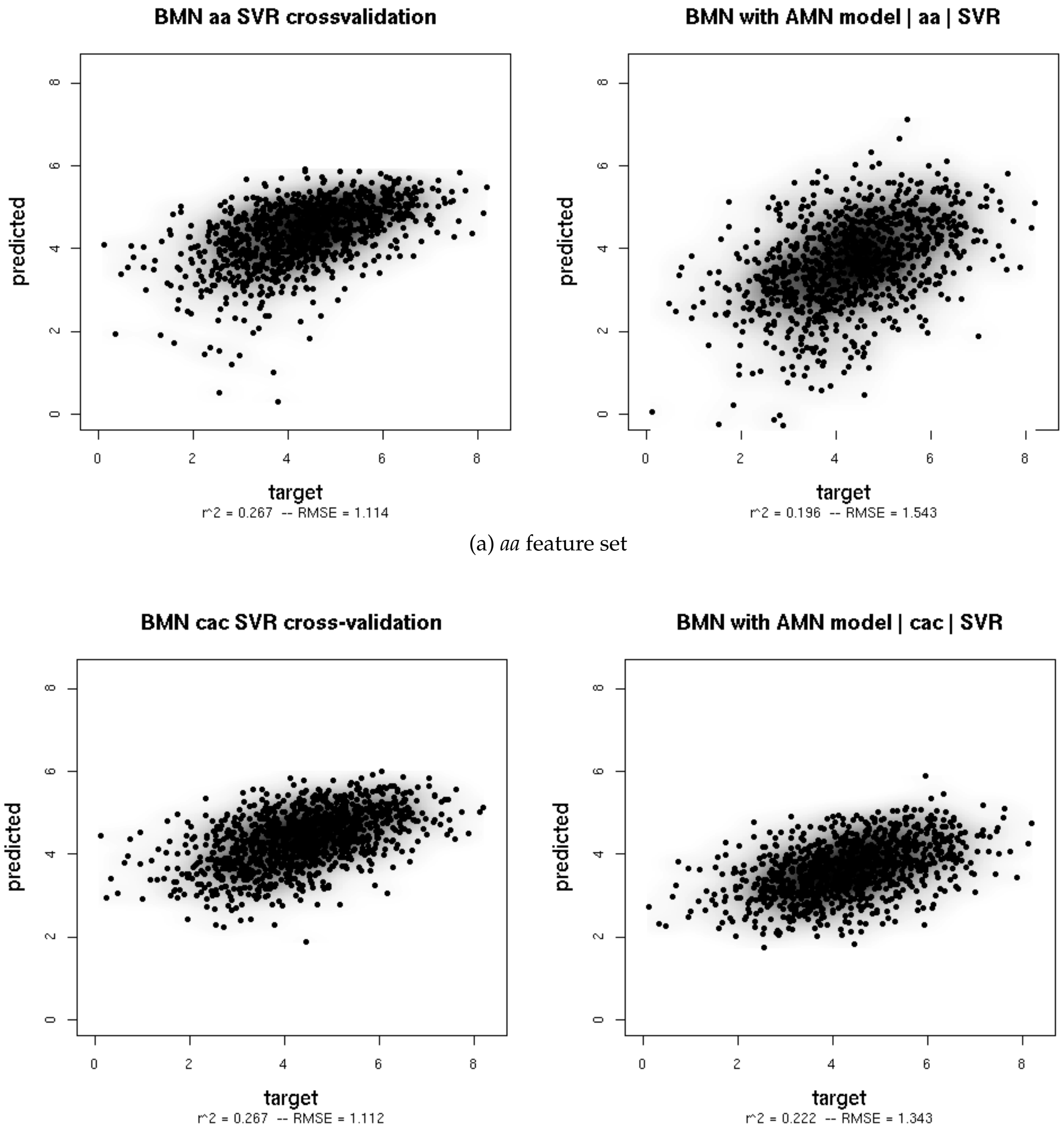
Figure 7.4: Scatter plots of prediction results for the physicochemical property feature sets of dataset *A* (*mic* normalization) predicted with ν-SVR (*left*: cross-validation, *right*: across dataset prediction). A detailed explanation of this type of plot is found in Fig. 7.3.



(a) *mono* feature set

(b) *seq* feature set

Figure 7.5: Scatter plots of prediction results for the string-based feature sets of dataset B (*mic* normalization) predicted with ν-SVR (*left*: cross-validation, *right*: across dataset prediction). A detailed explanation of these type of plot is found in Fig. 7.3.



(a) *aa* feature set

(b) *cac* feature set

Figure 7.6: Scatter plots of prediction results for the physicochemical property feature sets of dataset B (*mic* normalization) predicted with ν-SVR (*left*: cross-validation, *right*: across dataset prediction). A detailed explanation of these type of plot is found in Fig. 7.3.

7.3.3. Results for LC-ESI data

The LC-ESI data have been acquired in a late phase of this work. Therefore, only few machine learning applications have been tested. The results on trimmed-mean target values are presented here. The random forest regression method (see Section 3.4) is used because it is much faster to find its parameters and perform the training and validation. Since the random forest's prediction results (presented later in Chapter 8) are generally only a little worse than the ν-SVR results, random forest regression can be applied to acquire first prediction results for the LC-ESI data.

Results With these datasets, the prediction results are quite bad. The prediction of Y_2 using only the amino acid index features of the *aa* feature set with a random forest reaches a squared correlation between target and predicted values of only $r^2 = 0.09$, and $r^2 = 0.07$ with the *mono*, $r^2 = 0.04$ with the *seq* feature set. This is much worse than the predictions on the MALDI datasets, although the ESI datasets have a lower variance between replicates.

Discussion While the used approach gives reasonable results for MALDI data, it is not straightforward to apply it to ESI data. There are various possible reasons for the poor results on the ESI data.

With ESI, multiply charged ions are created in the ionization process. Here, intensities of different ions are added, but the majority of the peptides shows only one charge state anyway. Peptides with different charge states are mixed in the dataset. This might cause difficulties for the learning algorithm. A possible cure would be to divide peptides with different charge states into different training sets and predict them separately, discarding all peptides that are found with multiple charge states.

Another possible reason is the high CV of the measurement of protein abundances that are used to normalize the peak intensities, so that they reflect peptide-specific sensitivities (Section 6.2.4). Instead of these, more precise measurements by Ghaemmaghami *et al.* (2003) could be used, but only a fraction of the proteins have been measured more precisely (with a CV of up to 20% instead of 200 to 300%), so this would decrease the size of the datasets.

7.3.4. Summary

In this work, I want to assess whether the prediction of peptide-specific sensitivities in mass spectrometry from peptide sequences is feasible, and if so, how it can be done. The focus is on MALDI data. For these, only one protein is measured per spectrum, such that the peptide's peak intensities in a spectrum directly reflect the peptide-specific sensitivity of the MS device. Peptide sequences have been converted into four alternative numerical encodings (feature sets). The extracted feature sets use either only the peptide sequence (*mono*, *seq*) or additional information about chemical properties (*aa*, *cac*).

The results achieved with these four initial feature sets presented in this section already show that sensitivity prediction with machine learning methods is feasible. A squared Pearson's correlation of $r^2 = 0.46$ is recorded in ten-fold cross-validation, and $r^2 = 0.37$ in a prediction for totally new peptides. This is recorded on the MALDI dataset with the lower variance between replicate measurements (dataset A), using a feature set consisting only of the absolute numbers of single amino acids in the peptide sequence (*mono* feature set). The smaller of the feature sets with physicochemical properties (*aa*) leads to predictions that are almost as good. The prediction does not work as well on the second MALDI dataset (B), which has a higher variance between replicates. However, ν-SVR can draw the main trends from B and predict A with B's model. Generally, the lower-dimensional ones of the extracted feature sets generalize better to new data.

On the ESI datasets, the prediction results are really poor: $r^2 = 0.09$ is the best correlation recorded in cross-validation with random forest regression. Possible reasons are the more complicated normalization or the more complex spectra compared to MALDI.

The next steps are to find out if these results can be improved by reducing the feature sets, and to assess which peptide properties are especially important for peptide-specific sensitivities. These goals are pursued in the next chapter. Extended analyses of the prediction are postponed to Chapter 9.

8. Feature selection

Most of the extracted feature sets are very high-dimensional and redundant. It is known, that high dimensionality decreases the generalization capability of a model: Addition of unimportant information increases the overall noise, which make overfitting more likely. I want to increase the generalization performance and prediction speed by discarding unimportant features from the large extracted feature sets.

The other, maybe even more important, motivation for feature selection is to obtain knowledge about the system: Which peptide properties are important for the peptide-specific sensitivity? Are there differences between the datasets? In this chapter, I evaluate these questions.

The main problem with the physicochemical feature sets is that features are highly correlated. It can be assumed that different methods will choose completely different sets of features which contain the same or related properties, because these methods generally use different criteria to select the final model. For this reason, it is inappropriate to trust only a single feature selection method. Instead, I will apply various methods and try to integrate their results to come up with a list of relevant properties. A direct comparison is not trivial: Not all methods are equally trustworthy. The prediction performance of all methods will be assessed and kept in mind as a measure of trustworthiness of the features. In a large table, a visual overview will be presented from which conclusion can be drawn.

Another difficulty is that the results of the last chapter showed that both MALDI datasets are quite different from each other. In this chapter, I will focus on dataset A, which can be predicted with better accuracy using the initially extracted feature sets. All results in this chapter are for dataset A with the *mic* normalization unless stated otherwise.

Subsequently, the applied feature selection and importance assessment methods are explained.

8.1. Methods

Three feature selection methods have been applied to the large physicochemical feature sets: A heuristic derived from forward stepwise selection, least-angle regression, and path following algorithm for generalized linear models. Random forest regression is used for feature importance assessment. Other more simple methods are assessed for comparison only.

Boosted forward stepwise selection has been applied to the *aa* and *seq* feature sets (Section 8.3.1). Because forward stepwise selection is a greedy method, the selection is repeated twenty times to determine features that are chosen often (more than 5 out of 20), which constitutes a kind of majority vote of all twenty models, similar to *boosting*. Because the features selected by multiple forward stepwise selection runs cannot outperform the initially extracted sets, a few features are added manually for a more complete description of the peptide. These are not considered selected by this method in the comparison. The whole feature set including these added feature is called "selected subset" (sss).

Random forest feature importance assessment is applied to the *sss* feature set (Section 8.3.2). The *aa* feature set contains only summed amino acid index features. As a result, some features contained in the selected subset are highly correlated with the peptide's mass. To get rid of this, the features correlated with more than $r^2 = 0.8$ are scaled by mass prior to the importance assessment. The feature importance is then assessed on the *sss* containing the scaled features. The method is described in more detail in Section 3.4.

Least-angle regression (LARS) is applied to all features in the *cac* feature set (Section 8.3.3). In addition to the amino acid index the features are derived from, the algorithm can choose between different representations of that amino acid (see 7.1.2). LARS uses the C_p statistics to choose the optimal model.

L_1-regularization path of generalized linear model (glmpath). As with LARS, this shrinkage method is applied to all features in the *cac* feature set (Section 8.3.3). The applied `glmpath` implementation provides the BIC and AIC to choose the optimal model.

To assess feature importance in the string-derived feature sets *mono* and *seq*, a two-sample t-test between peptides containing a given substring and those not containing it has been carried out. In addition, variable importance has been assessed with random forest regression.

Comparison of feature importance weights

Feature ID	Forward stepwise	Random forest	LARS	glmpath
Basicity or acidity				
GB500	█	█	-	-
arginine count	▓	▒	-	-
JOND750102	-	x	░	-
FAUJ880111	▒	░	-	-
Hydrophobicity				
NADH010106		░	-	-
NADH010107	▒	░		-
WILM950102	▒	▒	-	-
PONP800106	-	x	░	-
EISD840101	-	x	░	░
RADA880107	-	x	░	-
SWER830101	-	x	░	░
NAKH900106	-	x	░	-
ZIMJ680101	░	x		-
Conformation				
PRAM820103	-	x	█	█
VASM830103	▓	▒	▒	▒
ROBB760107	░	░	-	-
FAUJ880106	-	x	░	

Comparison of feature importance weights

Feature ID	forward selection	random forest	LARS	glmpath
Secondary or tertiary structure				
FINA770101	-	░	-	-
ARGP820102	░	░	-	-
RICJ880113	-	x	░	-
RICJ880114	░	x	░	░
AURR980102	░	x	-	-
QIAN880123	░	x	-	-
QIAN880126	-	x	░	-
QIAN880139	-	x	░	-
OOBM850105	-	x	░	-
Others				
mass	-		-	-
KHAG800101	-	░	░	
KUMS000101	-	x	░	░
OOBM850104	░	░	-	-

Figure 8.1: Visual overview of physicochemical features selected by different feature selection algorithms on dataset A. The darker a cell, the higher the importance weight. "-" denotes the feature has not been selected with the corresponding method. 'x' marks features that have not been presented to the corresponding method. See text for discussion.

8.2. Integration of selected features from different methods

For each method, the features are assigned weights that reflect their individual importance. However, the methods provide different means of acquiring a weight. Figure 8.1 shows a broad overview of the weights of the amino acid indices that the selected features are derived from. Figure 8.2 shows a corresponding view for the importance of single amino acids. Gray values denote the importance of a feature: the darker, the higher the weight. The feature with the maximum weight of each method is black (i.e. its gray value is 0). Crossed-out rows indicate that the corresponding feature or amino acid index has not been considered for that method.

Weights have been determined within each method separately. This implies a feature with the same gray value in different methods does not generally have the same importance in both methods. This is how the weights are calculated:

Boosted forward stepwise selection and random forest importance For random forest regression, this set outperforms any of the other sets in regard to generalization performance. Therefore, this selection is considered with high confidence. The weight $w_i^{fs} = m/20$ with m the number of times the feature has been chosen.

Random forest The weights are $w_i^{rf} = \frac{a_i}{a_{max}}$, where a_i is the accuracy decrease of feature i if it is permuted. The maximum value is $a_{max} = 0.361$.

LARS Only 20 of 3651 *cac* features have been selected by this method (Section 8.3.3). The generalization prediction performance of this feature set (*larf*, $r^2 = 0.31$, evaluated with random forest regression) is similar to the *sss* set. The weight for a feature i is calculated as $w_i^{lars} = \frac{b_i}{b_{max}}$ where b_i is the coefficient of feature i.

Glmpath The minimum BIC for models with low complexity was determined visually, because both BIC and AIC have their global minimum at the least squares solution (i.e. the full model) for this method. Nine features have non-zero coefficients at this point. The prediction accuracy with random forest regression is slightly lower than for the forward stepwise selection heuristic and LARS on one of the datasets, but much lower on the other one. However, the lesser prediction performance, indicates that the selected model is too constraint (i.e. does not consider enough features). The weight is determined in analogue to LARS, with $b_{max} = 3.37$.

The two-sample t-test on the *seq* feature set emits p-values p_j to measure the probability that the determined difference in the mean intensity of the two groups for substring j is not by chance. The weights are calculated as $w_j = \frac{\ln p_j}{\ln p_{\min}}$. Here, the smallest value $p_{\min}$ is mapped to a gray value of 0.0 (black). For this test, no prediction performance can be assessed, and the single substrings are tested independently from each other.

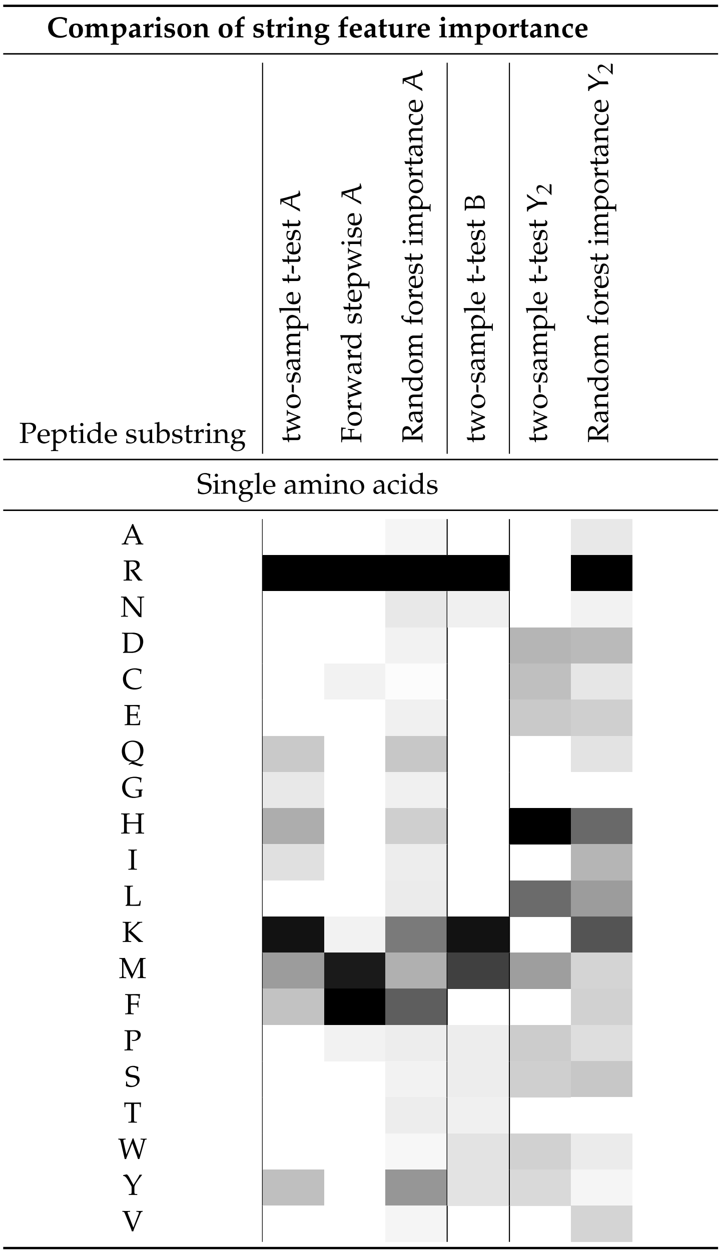


Figure 8.2: Visual overview of importance of single amino acid features when applying different methods to the various datasets. Darker cells indicate a higher importance. See text for discussion.

8.2.1. Evaluation of the feature selection comparison for MALDI datasets

Next, the selected features are interpreted:

Physicochemical features Of the features derived from amino acid indices, only the forward stepwise selection, the LARS and the glmpath methods constitute real feature selection methods in the sense of resulting in an actual subset from a larger feature set. All three selection methods led to subsets containing multiple features describing mostly basicity, hydrophobicity, and conformation. Secondary structure related indices are also selected, but with low importance. The random forest importance can be used to rank the features within the *sss* feature set and does not select features. Of the features in the *sss* feature set, that have been added in manually, only the peptide's theoretical mass attained a high importance value.

The glmpath selection is a subset of the features selected by LARS. In general, forward stepwise selection and LARS chose features derived from different amino acid indices. Where the forward stepwise selection could only choose within summed amino acid index features, LARS and glmpath have been presented with additional features derived from the indices by other rules (Section 7.1.2), and they almost always choose the difference between lowest and highest amino acid index value over the peptide (M), the other chosen features are mostly distance-based features ($D_x, x \in \{2, 3\}$).

String feature selection Of the feature selection and importance assessment in string-based features, only the single amino acid character numbers are evaluated, because no consistent pattern can be found in the selected dipeptides. This is most likely due to the small size of the MALDI datasets. Most dipeptides and tripeptides do not occur often enough or at all. The prediction performance of the *mono* feature set (i.e. the one containing the number of single amino acids in the peptide only) results in the overall best prediction accuracy using ν-SVR. For other evaluated prediction methods, selected feature sets work better than the *mono* feature set, and the feature selection methods never select a single amino acid count except for the arginine residue (R). The weights given to the *mono* features are analyzed in the following.

Of the single amino acid counts, all methods agree that the number of arginine residues (R) and methionine residues (M) is important for the intensities of both MALDI datasets. The presence or absence of lysin residues (K) causes a significant difference in the intensity according to a two-sample t-test, but is only of intermediate importance according to the forward stepwise selection and random forest assessment. This is not surprising, because only the latter two methods consider other features in the same set, and K and R are complementary: Most tryptic peptides either end in R or K and seldom contain these within the string.

Thus it is important if the string ends in K or R. In the order of their relative importance, the numbers of phenylalanine (F), tyrosine (Y), histidine (H), and glutamine (Q), are important in dataset A.

On the LC-ESI data, the two-sample t-test and random forest importance assessment do not agree very well: R and K are rated highest by random forest importance and not at all in the t-test. This means that the groups of peptides containing or not containing R or K do not have different means but yet give and important contribution to the prediction function. Both methods agree that H is very important.

The emerging overall importance pattern is similar between MALDI dataset A and the studied ESI dataset.

8.3. Detailed results of the different methods

8.3.1. Forward stepwise selection

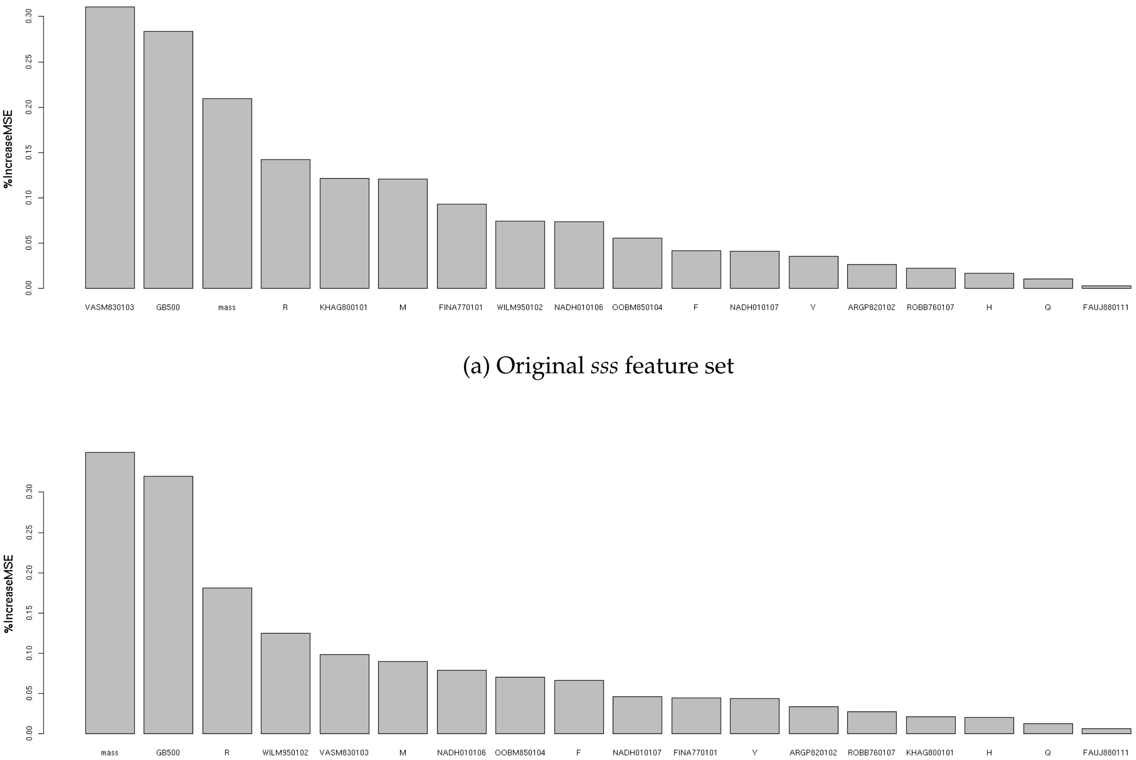
A forward stepwise selection was applied twenty times as described in Section 8.1. Instead of the RSS required for this method, the prediction errors of a 10-fold cross-validation with the ν-SVR are used. The models are built with the corresponding parameters chosen by the grid search on the full feature set, since it is infeasible to repeat a grid search for each selection step. The constituting features of the final *sss* feature set are shown in Table 8.1.

Analysis of the selected features The features selected most often in the feature selection on dataset A were the estimated gas-phase basicity at 500 K (GB500), the absolute number of arginine residues (R), the relative population of conformational state E (VASM830103 Vásquez *et al.* 1983), the hydropathy scale based on self-information values in the two-state model at 36% accessibility (NADH010106 Naderi-Manesh *et al.* 2001), the hydrophobicity coefficient in RP-HPLC, C8 with 0.1%TFA/MeCN/H_2O (WILM950102 Wilce *et al.* 1995, and the number of positive charges (FAUJ880111 Fauchére *et al.* 1988) of the *aa* feature set. From the *seq* feature set, the numbers of arginine (R), phenylalanine (F), and methionine (M) residues were selected most often.

Forward stepwise selection is a greedy method and does not find an optimal solution. None of the selected sets from each single run of the method leads to a better performance than that of the original set. Thus, other features that round out the description of the peptide are added in. After prediction, the importance of the single features that constitute the *sss* feature set are assessed, using random forests for regression Breiman (2001, 2002a). The prediction results for the random forest prediction are discussed below (Section 8.3.2). Figure 8.3a visualizes feature importance measured on the test sets. According to this, VASM830103 is the most

Feature ID	Explanation	Selected
GB500	estimated gas-phase basicity at 500 K (Zhang *et al.*, 2004)	20
VASM830103	Relative population of conformational state E (Vasquez *et al.*, 1983)	11
NADH010106	Hydropathy scale (36% accessibility) (Naderi-Manesh *et al.*, 2001)	9
FAUJ880111	positive charge (Fauchere *et al.*, 1988)	6
WILM950102	Hydrophobicity coefficient in RP-HPLC, C8 with 0.1%TFA/MeCN/H_2O (Wilce *et al.* 1995)	6
OOBM850104	Optimized average non-bonded energy per atom (Oobatake *et al.*, 1985)	2
mass	Molecular mass of the peptide	–
KHAG800101	The Kerr-constant increments (Khanarian-Moore, 1980)	–
NADH010107	Hydropathy scale (50% accessibility) (Naderi-Manesh *et al.*, 2001)	–
ROBB760107	Information measure for extended without H-bond	–
FINA770101	Helix-coil equilibrium constant (Finkelstein-Ptitsyn, 1977)	–
ARGP820102	Signal sequence helical potential (Argos *et al.*, 1982)	–
R	No. of arginine residues	20
F	No. of phenylalanine residues	20
M	No. of methionine residues	17
Q	No. of glutamine residues	5
Y	No. of tyrosine residues	4
H	No. of histidine residues	–

Table 8.1: Selected feature subset. Features from both the *aa* and *seq* feature sets that were selected most often in a forward selection and which can be assumed to be relevant for the ionization process in MALDI MS are included in this set. The literature that describe the indices can be found with Kawashima *et al.* (1999). Gray-shaded features have been added manually.



(a) Original *sss* feature set

(b) *sss* feature set with scaled features

Figure 8.3: Importance of features in dataset *A* with random forest regression on the *sss* feature set. They y-axis shows the percentaged decrease of accuracy if the values of the corresponding feature are randomly permuted. KHAG800101, VASM830103, FINA770101, and ARGP820102 are highly correlated to the mass feature in *sss*. These have been scaled by mass, constituting another importance ranking (shown in the lower plot).

important feature, followed by GB500 and the peptide's theoretical mass. Of the added in features, the mass and Kerr-constant increments (KHAG800101, Khanarian and Moore 1980) are most important.

Some of the constituting features are correlated to the peptide mass. To get rid of this, these features (KHAG800101, VASM830103, FINA770101, and ARGP820102) have been scaled by the mass. Afterwards, none of these normalized features is correlated to mass any more. Another importance assessment is applied to the *sss* feature set containing the scaled features instead of the original ones. The feature importance is shown in Fig. 8.3b). In comparison to the original *sss* feature set, all of the scaled features except VASM830103 become unimportant. Mass and GB500 are the most important features. The number of arginine residues (R) and WILM950102, a hydrophobicity index, rank next. VASM830103 leads the features that are of intermediate importance.

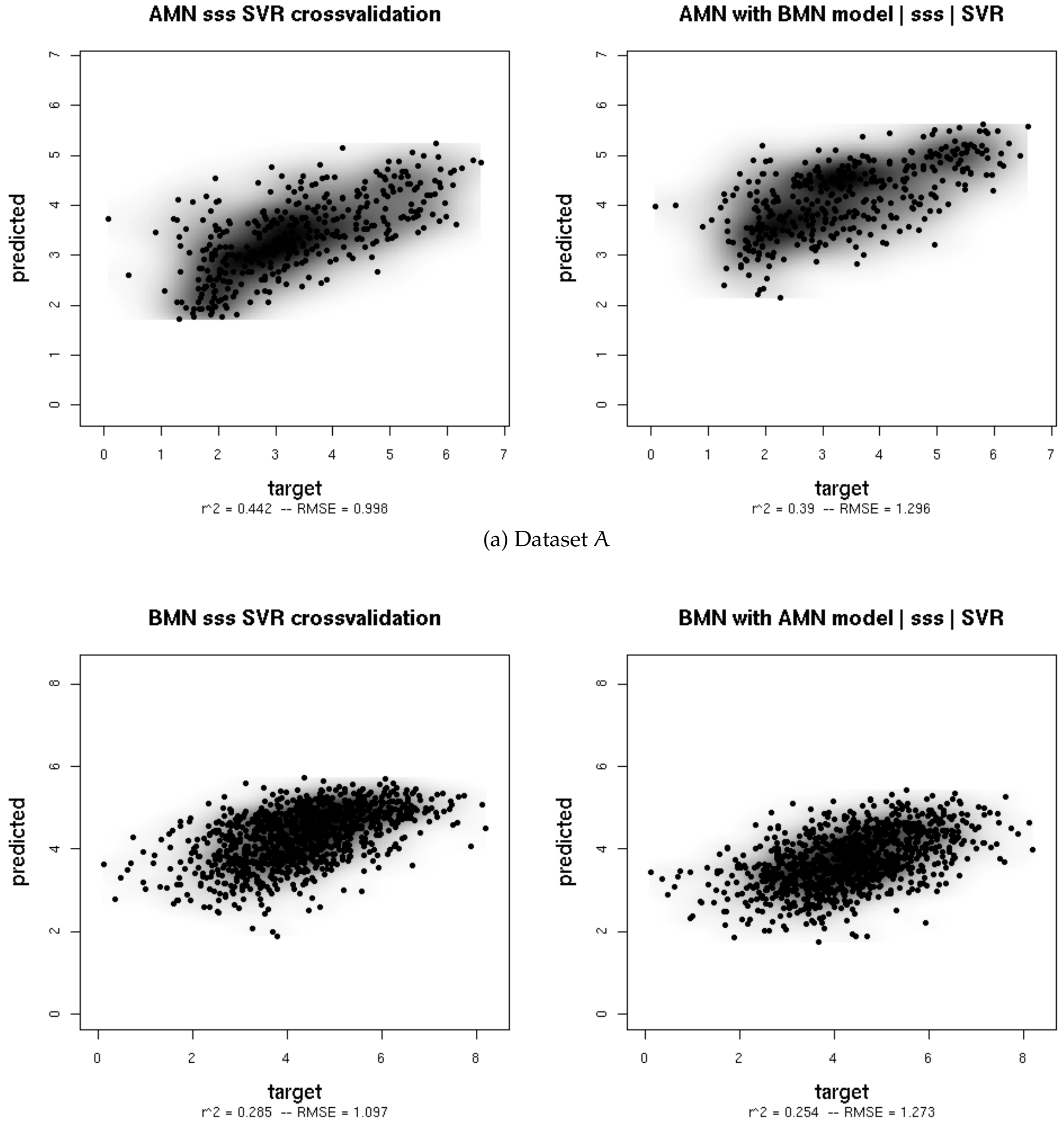
This indicates that following mass and gas-phase basicity, hydrophobicity and conformation play a role for peptide-specific sensitivity. However, not much is known about the conformation of peptides after tryptic digestion, when crystallized within the matrix, or as ions in gas-phase. Looking at the chemistry, it makes sense that the gas-phase basicity influences ionization efficiency. The number of positive charges have been reported by Mallick *et al.* (2007) to be relevant for the probability to observe a peptide ion. It has often been chosen in the feature selection but is the lest important one in the *sss* feature set according to our feature importance accession.

The number of histidine residues (H) has been found to be correlated with detection probabilities in MALDI MS by Mallick *et al.*. It is the only, if weakly, basic residue except K and R, presumably making a difference in basicity for tryptic, i.e. already quite basic peptides. However, it is one of the three lest important features according to the random forest method in our dataset. We can exclude the correlation ($r = 0.922$) between H and FAUJ880111 as the cause for their low importance: The importance ranking is the same if one of both is left out completely, so the other features already cover the information of FAUJ880111 and H.

ν-SVR Prediction results on *sss* features

For the ν-SVR, the *sss* feature set shows a slightly lower accuracy compared to the *mono* feature set (Table 8.2). The scatter plots (Fig. 8.4) between target and predicted values are qualitatively similar to those of the *mono* feature set (Fig. 7.3).

However, for dataset B, the *sss* feature set leads to a better prediction than any of the other feature sets.



(a) Dataset A

(b) Dataset B

Figure 8.4: Scatter plots of the prediction result for the *sss* feature set on both MALDI datasets with ν-SVR (*left*: cross-validation; *right*: across dataset prediction). A detailed explanation of this type of plot is found in Fig. 7.3.

ν-SVR *sss* feature set results				
validation dataset	**feature set**	**prediction performance** [r^2(RMSE)]		
		cross-validation	across dataset validation	across dataset strict validation
A	*mono*	0.46 (0.989)	0.44 (1.251)	0.37
	sss	0.44 (0.990)	0.39 (1.298)	0.34
B	*mono*	0.22 (1.178)	0.20 (1.366)	0.21
	sss	0.29 (1.097)	0.25 (1.237)	0.27

Table 8.2: Prediction accuracy results for the selected subset features on both MALDI datasets with the ν-SVR compared to the *mono* feature set, which is the best of the initially extracted feature sets. r^2 denotes the squared Pearson's correlation, RMSE is the root of the mean squared error.

8.3.2. Random forests for feature importance assessment

For random forests, the number n of trees is sampled between 250 and 3000 in steps of 50, and the best model in terms of n is selected using the best r^2 on the training set (see Section 7.3.1 for a motivation of this approach).

For random forests, the *sss* is the most successful of all feature sets with $r^2 = 0.39$ for cross-validation and $r^2 = 0.30$ in the strict across-dataset validation. Thus, the assessed feature importance using random forest regression can be assumed to be valid. All other feature sets are similar to each other in the strict generalization case. Surprisingly, that applies to the *seq* features, too, on which ν-SVR performs much worse.

Dataset B gives much better prediction results ($r^2 = 0.31$ in the strict across-dataset validation) with a random forest regression than with ν-SVR when using the *sss* feature set. With random forests, the generalization performance of B is comparable to that of dataset A.

Breiman (2001) states that random forests do not overfit. We observe slight overfitting for dataset A, but in fact there is no overfitting at all for dataset B.

Scatter plots The prediction vs. target value scatter plots for the *sss* feature set (Fig. 8.5) look similar to that of the ν-SVR, but the across dataset validation shows a wider range of predicted values. Specifically, the higher values are predicted better in the mean but also have a slightly higher spread than with ν-SVR.

For the feature sets *aa*, *mono*, and *cac*, the random forest plots are less diagonal than that of the ν-SVR and predicted values span a smaller range (not shown). The *seq* feature set scatter plots show a similar structure to the aforementioned feature sets, which is much better than their performance with ν-SVR, where the point cloud is almost a horizontal line.

validation dataset	feature set	forest size	prediction performance [r^2]		
			cross-validation	across dataset validation	across dataset strict validation
A	*mono* ν-SVR	(n.a.)	0.46	0.44	0.37
	seq ν-SVR	(n.a.)	0.32	0.21	0.14
	mono	2650	0.39	0.35	0.25
	seq	250	0.35	0.27	0.26
	aa	600	0.33	0.34	0.26
	cac	2750	0.40	0.32	0.24
	sss	750	0.39	0.37	0.30
	spec	800	0.30	0.02	0.02
B	*mono* ν-SVR	(n.a.)	0.22	0.20	0.21
	seq ν-SVR	(n.a.)	0.19	0.10	0.10
	mono	300	0.24	0.24	0.23
	seq	700	0.21	0.22	0.21
	aa	2750	0.23	0.23	0.24
	cac	1200	0.26	0.26	0.27
	sss	1150	0.29	0.29	0.31
	spec	1500	0.29	0.01	0.01

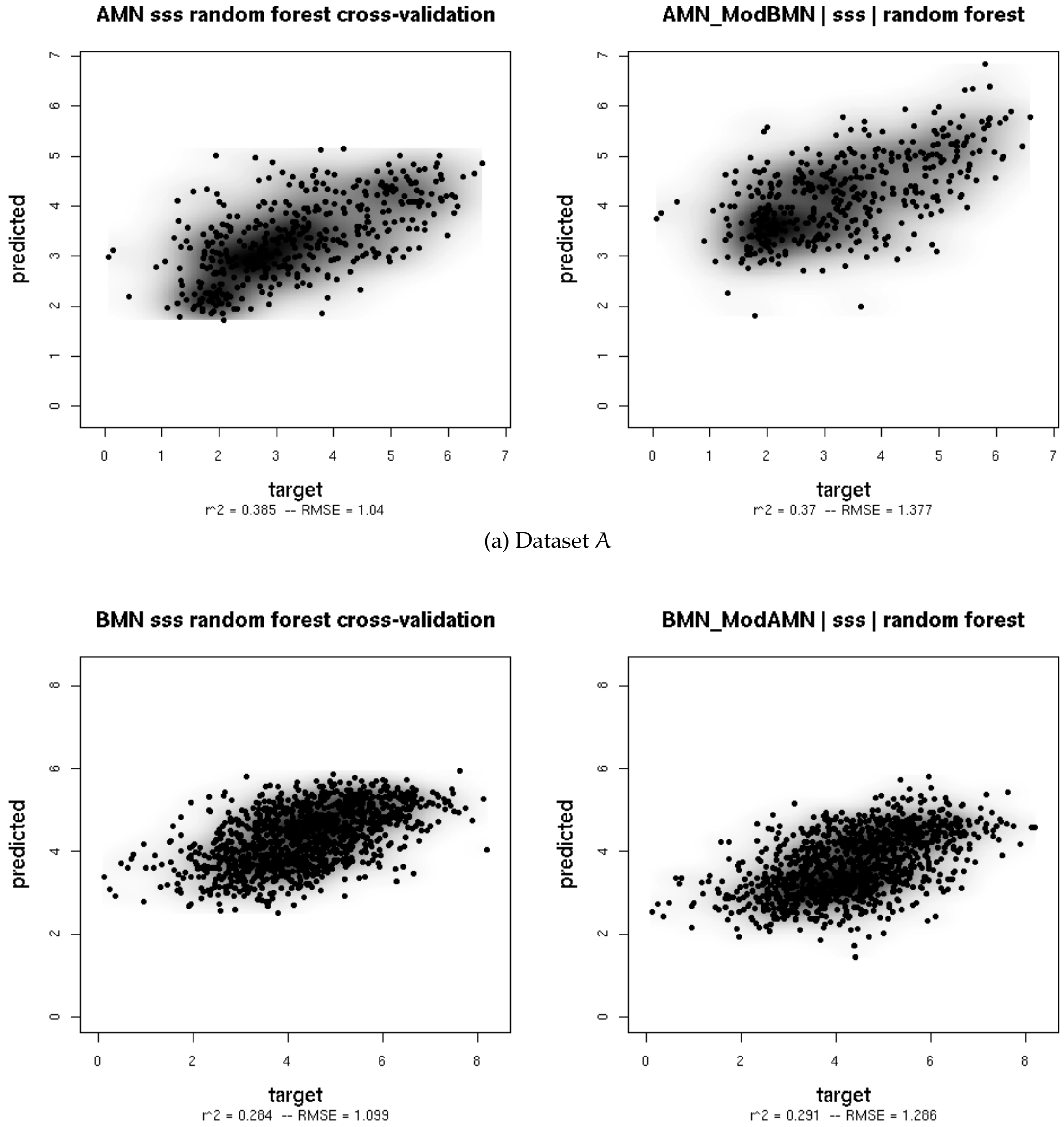
Random forest results

Table 8.3: Random forest prediction accuracies. The random forest regression works well with the *sss* set. The generalization performance with the *sss* features on dataset B is better than for any ν-SVR prediction. Surprisingly, it also leads to better results with the *seq* feature set than ν-SVR (shown in gray for comparison).

For dataset B, all scatter plots are similar to those of the ν-SVR, except for the *seq* feature set: For this set, the random forest prediction shows a separation of the point cloud between two levels of predicted values. A hint of this structure can be observed for the other scatter plots of dataset B, which supposedly gives the random forest a small advantage over the ν-SVR prediction of dataset B.

8.3.3. L_1-penalized methods for feature selection

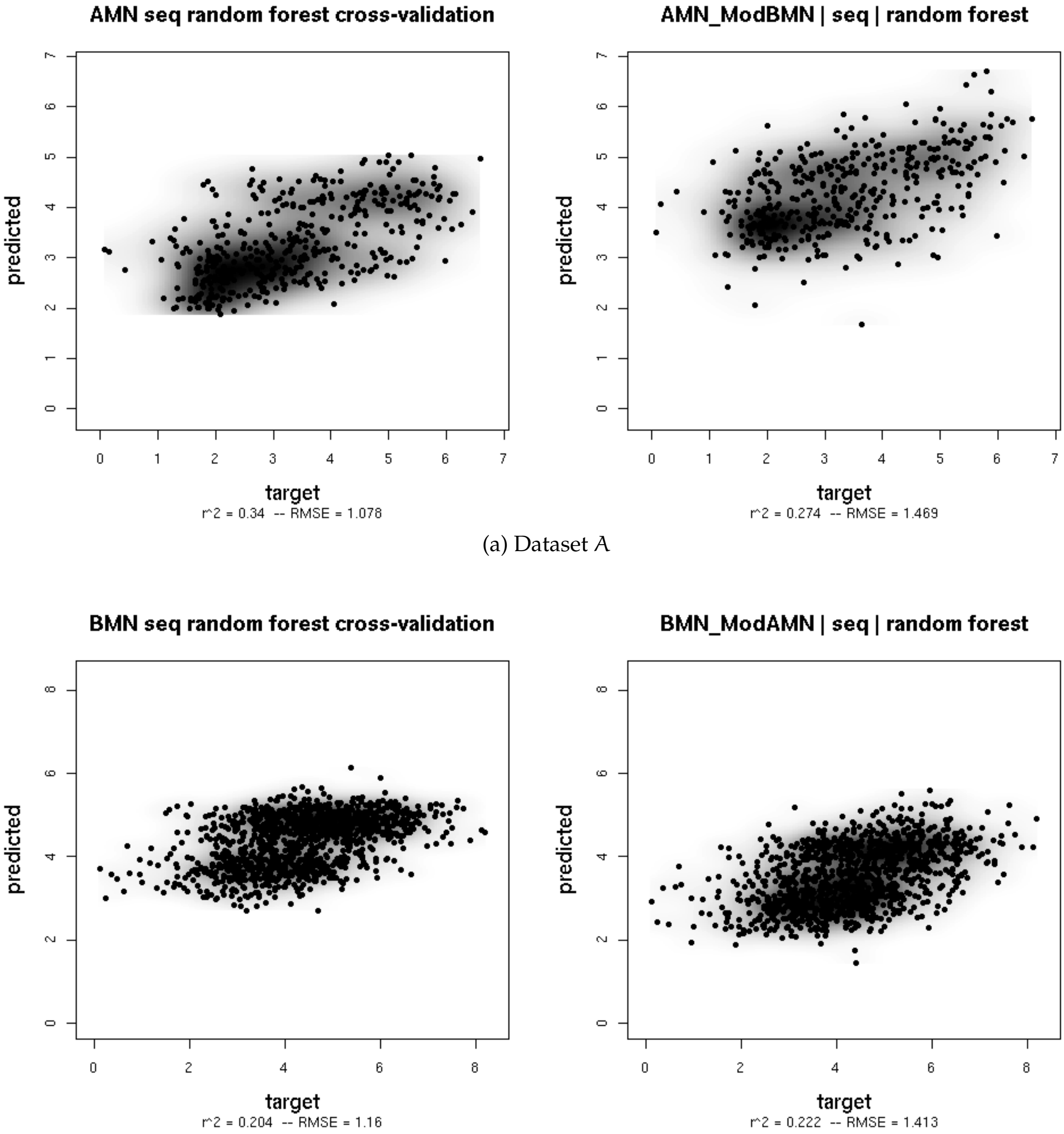
Shrinkage methods (Section 3.5 and 3.6.2) that use an L_1 constraint on the coefficients have a special property: They gradually shrink coefficients of unimportant features to zero. The



(a) Dataset A

(b) Dataset B

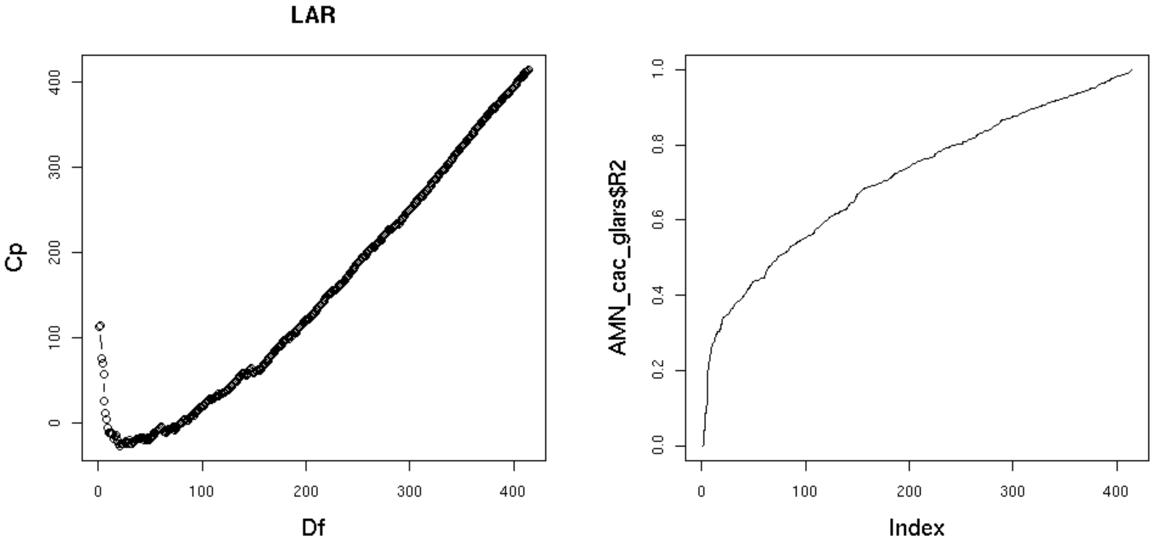
Figure 8.5: Scatter plots of the prediction results on MALDI datasets with the random forest regression (*sss* feature set, *mic* normalization, *left*: cross-validation, *right*: across dataset prediction). A detailed explanation of this type of plot is found in Fig. 7.3.



(a) Dataset A

(b) Dataset B

Figure 8.6: Scatter plots of the MALDI datasets with the a random forest regression (*seq* feature set, *mic* normalization, *left*: cross-validation, *right*: across dataset prediction). A detailed explanation of this type of plot is found in Fig. 7.3.



(a) Cp of the least-angle regression path.

(b) r^2 of least-angle regression path.

Figure 8.7: Visualization of least-angle regression path on dataset A (*mic* normalization). The x-axes show the steps of the method. The rightmost model has the most degrees of freedom and corresponds to the least-squares solution. Increasing the size of the model (from left to right) generally increases the correlation between predicted and target values and decreases the estimated generalization risk. At the point of minimum Cp, the generalization is assumed to be best. There is a kink in the r^2 curve at this point. From there, r^2 increases more slowly than before.

non-zero coefficients constitute a feature selection. I apply the lasso and a path-following algorithm for L_1-penalized generalized linear models (glmpath) on the *cac* feature set to exploit this possibility.

Feature selection by least-angle regression If LARS is applied to the (*cac*) feature set, twenty coefficients are non-zero at the step with minimum C_p. About a third of these features have to do with structure or conformation (8 of 20), another third is hydrophobicity-related (7 of 20).

The LARS path is visualized in Fig. 8.7. Increasing the complexity of the model (from left to right) generally increases the correlation between predicted and target values. For small λ or step numbers, the estimated generalization risk decreases. At the minimum C_p, r^2 increases more slowly. At this point, the squared correlation coefficient is $r^2 = 0.34$. Further releasing the constraint on the model, thereby adding more features, increases r^2 on the training set but would lead to overfitting. As a comparison, the best ν-SVR prediction for the strict across

Features chosen by least-angle regression			
AAIndex	description	feature type	coefficient value
PRAM820103	Correlation coefficient in regression analysis (Prabhakaran-Ponnuswamy, 1982)	M	-4.16
VASM830103	Relative population of conformational state E (Vasquez *et al.*, 1983)	M	1.28
QIAN880126	Weights for beta-sheet at the window position of 6 (Qian-Sejnowski, 1988)	D_2	-0.34
EISD840101	Consensus normalized hydrophobicity scale (Eisenberg, 1984)	M	-0.23
JOND750102	pK (-COOH) (Jones, 1975)	M	0.22
RADA880107	Energy transfer from out to in (95% buried) (Radzicka-Wolfenden, 1988)	M	0.20
QIAN880139	Weights for coil at the window position of 6 (Qian-Sejnowski, 1988)	M	0.10
SWER830101	Optimal matching hydrophobicity (Sweet-Eisenberg, 1983)	M	0.089
PONP800106	Surrounding hydrophobicity in turn (Ponnuswamy et al., 1980)	D_3	-0.066
RICJ880113	Relative preference value at C_2 (Richardson-Richardson, 1988)	M	0.052
KUMS000101	Distribution of amino acid residues in the 18 non-redundant families of thermophilic proteins (Kumar et al., 2000)	M	0.045
OOBM850105	Optimized side chain interaction parameter (Oobatake et al., 1985)	D_2	-0.029
NAKH900106	Normalized composition from animal (Nakashima et al., 1990)	D_3	-0.023
RICJ880114	Relative preference value at C_1 (Richardson-Richardson, 1988)	M	0.022
FAUJ880106	STERIMOL maximum width of the side chain (Fauchere et al., 1988)	M	0.020
RICJ880114	(see above)	Σ	0.018
ZIMJ680101	Hydrophobicity (Zimmerman et al., 1968)	S	0.011
OOBM850105	(see above)	D_3	-0.011
KHAG800101	The Kerr-constant increments (Khanarian-Moore, 1980)	M	0.003
NADH010107	Hydropathy scale (50% accessibility) (Naderi-Manesh *et al.*, 2001)	Avg	-0.0006

Table 8.4: All Features with non-zero coefficients determined by minimum Cp (estimated generalization risk) with a least-angle regression (LARS) method. See Section 7.1.2 for an explanation of the feature types. The literature that describe the indices can be found with Kawashima *et al.* (1999).

dataset validation case on the full feature set is $r^2 = 0.23$. Clearly, the ν-SVR fails to extract only the relevant information from this 3649-dimensional feature set. On the contrary, LARS is able to extract relevant features and is transparent at the same time. The generalization of this subset (*larf*) chosen using dataset A is still fair on B.

For dataset B, the feature selection via LARS fails: The minimum C_p is at the full model (i.e. the least squares solution), which totally overfits: It achieves perfect correlation on the training data but $r^2 = 0.01$ if used to predict new peptides (strict across dataset validation on A).

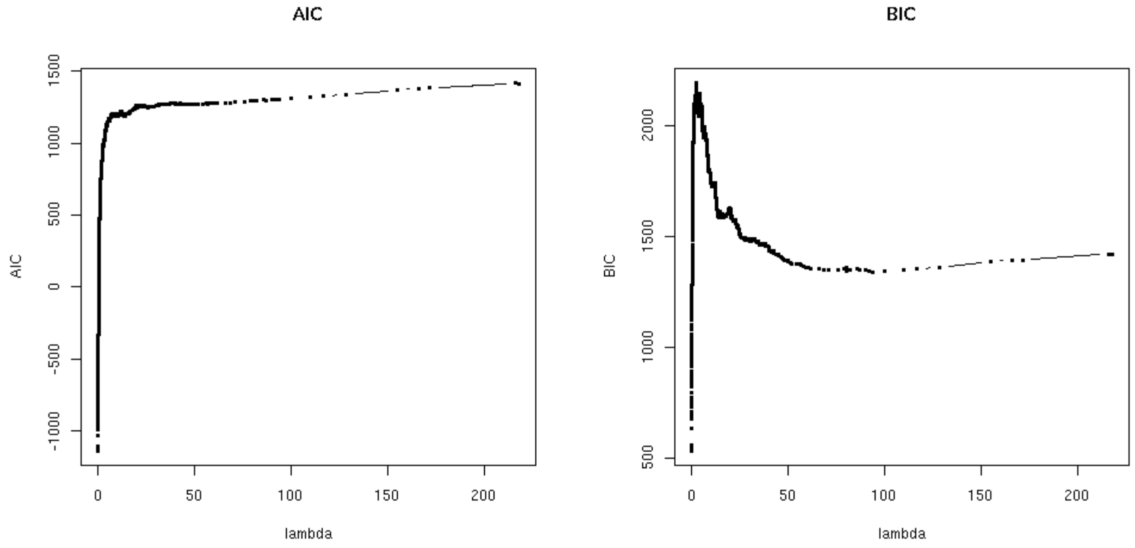
These are the prediction results for the feature sets select by LARS (*larf* and *larfB*):

Least-angle regression prediction results

validation dataset	**feature set**	**step**	**prediction performance** $[r^2]$		
			cross-validation	across dataset validation	across dataset strict validation
	sss (RF)	n.a.	0.39	0.37	0.30
A	*larf* (LARS)	20	0.34	0.02 (*larfB*)	0.01 (*larfB*)
	larf (RF)	n.a.	0.44	0.38	0.31
	sss (RF)	n.a.	0.29	0.29	0.31
B	*larfB* (LARS)	1128	0.9996	0.17 (*larf*)	0.16 (*larf*)
	larf (RF)	n.a.	0.23	0.22	0.20

Feature selection by L_1-penalized generalized linear models To choose a feature set with the glmpath method, the BIC and AIC are provided by the used implementation. However, both criteria are minimal at the model with the most degrees of freedom, which corresponds to the least squares solution where all features are in the model (Figure 8.8). However, least squares does not generalize at all on our data. The detailed results are shown in the next chapter (Section 9.1.4). Also, that way, there is no feature selection.

Because the chosen model should be interpretable, a simple model would be preferable. For a small number of degrees of freedom (for $\lambda > 50$), there is a minimum in the BIC curve at step 10, $\lambda = 94.16$. The nine features with non-zero coefficients at this step are evaluated. Their coefficients and the BIC curve for $\lambda > 50$ are shown in Figure 8.9. The selected features (*glmf*) are a subset of the features selected by LARS, and the absolute coefficient values are in the same order as the LARS results: The most important features are related to conformation, followed by hydrophobicity features. As with the LARS selection, almost all features are of type M (see Section 7.1.2), meaning that they are calculated as the difference between maximal and minimal value over all amino acids in a given peptide.



(a) AIC

(b) BIC

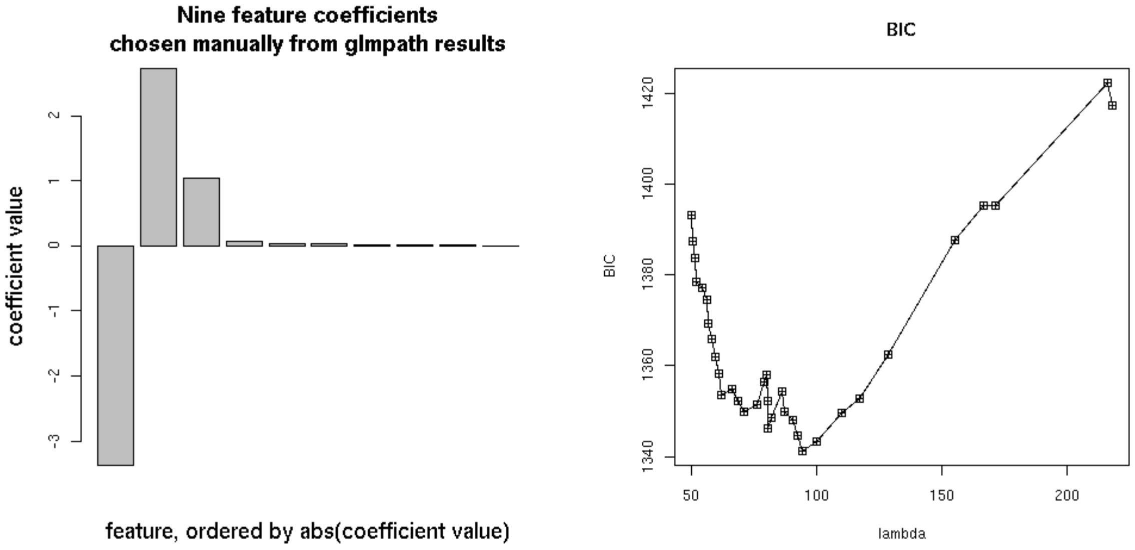
Figure 8.8: AIC and BIC plotted against λ, the L_1 regularization parameter. The minimum is at the full model for both indices, which corresponds to the least squares solution.

To assess the prediction performance of the chosen features, random forest regression is used with the feature set chosen by the glmpath method (*glmf*). These are the results:

Prediction results of glmpath features with random forests

validation dataset	**feature set**	**forest size**	**prediction performance** [r^2]		
			cross-validation	across dataset validation	across dataset strict validation
A	*sss* (RF)	3000	0.39	0.37	0.30
	larf (RF)	500	0.44	0.38	0.31
	glmf (RF)	500	0.40	0.38	0.29
B	*sss* (RF)	1150	0.29	0.29	0.31
	larf (RF)	500	0.23	0.22	0.20
	glmf (RF)	500	0.18	0.18	0.16

The *glmf* feature set is comparable to *sss* and *larf* on dataset A, but much worse when used for prediction of dataset B. It can be suspected that the chosen model is too constraint, i.e. does not consider enough features.



(a) Coefficients for $\lambda = 94.16$

(b) BIC for large λ

Figure 8.9: Coefficient values of the nine non-zero coefficients determined by minimum BIC at $\lambda > 50$ in the glmpath method (*left*). BIC curve for $\lambda > 50$ (*right*). For large values of λ there is a minimum at $\lambda = 94.16$.

8.3.4. t-test in the *seq* feature set

Data points were separated into two lists: One list with peptides that contains a specific corresponding mono-, di- or trimer, and another list with those that do not. A two-sample t-test was committed on both lists for each feature of the *seq* feature set of various datasets. The results are shown in Table 8.6 and 8.7. For both MALDI datasets, the peptides with the monomers R, K, M, and the terminal dimer GK show a significant difference in their mean target value from those peptides that do not contain these substrings. H and F seem to be relevant for dataset A but not for B. R and H containing peptides have higher intensities than the respective other groups, while K and M have lower intensities. This result is in accordance with a prediction using random monomer five-tuples. Those five-tuples that contain R or K, F, and H or M, are the most successful in a cross-validation.

The two LC-ESI datasets with the lowest within-peptide variance, Y_2 and $Y_3(2)^CF_5$, behave similar to each other in the two-sample t-test. A significant difference is observed for M and H, however, in contrast to the MALDI datasets, H-containing peptides have lower intensities. In addition, L is found in peptides with higher intensities. Although these findings are significant (i.e. have a low p-value), the difference between the means is not very large in most cases (0.96 at best), considering that logarithmic intensities from the LC-ESI datasets have a much

Features chosen by glmpath			
AAIndex	description	feature type	coefficient value
PRAM820103	Correlation coefficient in regression analysis (Prabhakaran-Ponnuswamy, 1982)	M	−3.373
VASM830103	Relative population of conformational state E (Vasquez *et al.*, 1983)	M	1.033
JOND750102	pK (-COOH) (Jones, 1975)	M	0.069
EISD840101	Consensus normalized hydrophobicity scale (Eisenberg, 1984)	M	0.037
SWER830101	Optimal matching hydrophobicity (Sweet-Eisenberg, 1983)	M	0.026
RICJ880114	Relative preference value at C_1 (Richardson-Richardson, 1988)	Σ	0.014
KUMS000101	Distribution of amino acid residues in the 18 non-redundant families of thermophilic proteins (Kumar et al., 2000)	M	0.010
FAUJ880106	STERIMOL maximum width of the side chain (Fauchere et al., 1988)	M	0.009
KHAG800101	The Kerr-constant increments (Khanarian-Moore, 1980)	M	0.004

Table 8.5: The nine features with non-zero coefficients determined by minimum BIC for models with $\lambda > 50$ in glmpath method. The overall minimum of both BIC and AIC is at the full model (i.e. the least squares solution). See Section 7.1.2 for an explanation of the feature types. The literature that describe the indices can be found with Kawashima *et al.* (1999).

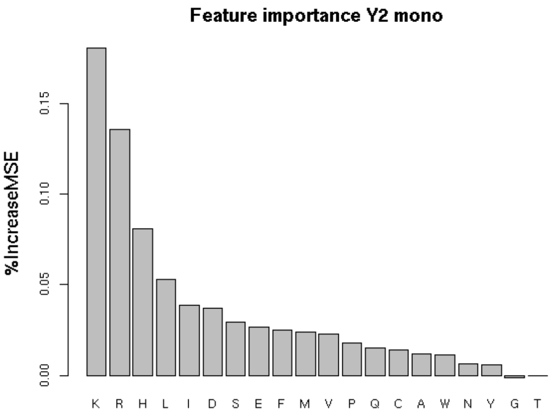


Figure 8.10: Importance of features in dataset Y_2 with a random forest on the *mono* feature set. They y-axis shows the mean decrease in accuracy if the values of the corresponding feature are randomly permuted.

substring s	p value	$\mu(s_A^-)$	$\mu(s_A^+)$	size(s_A^-) s	size(s_A^+)
R	2.41e-15	2.76	3.73	171	244
K	2.85e-14	3.72	2.78	243	172
M	1.69e-06	3.44	2.53	366	49
H	2.25e-05	3.20	3.90	338	77
VKe	4.64e-05	3.36	2.31	403	12
VK	8.92e-05	3.36	2.39	402	13
VF	1.25e-04	3.29	4.65	403	12
Y	2.11e-04	3.18	3.71	296	119
F	2.99e-04	3.15	3.67	272	143
GF	5.83e-04	3.29	4.48	400	15
Q	8.97e-04	3.16	3.61	260	155
GKe	0.001	3.37	2.62	392	23
TKe	0.001	3.36	2.45	401	14
SV	0.001	3.28	4.15	393	22

substring s	p value	$\mu(s_B^-)$	$\mu(s_B^+)$	size(s_B^-) s	size(s_B^+)
R	4.12e-33	3.69	4.64	339	795
K	6.76e-31	4.65	3.74	770	364
M	3.72e-25	4.52	3.41	966	168
DK	3.80e-07	4.38	3.35	1112	22
DKe	1.18e-06	4.37	3.38	1113	21
GM	1.67e-05	4.38	3.23	1112	22
AKe	2.27e-05	4.39	3.64	1085	49
NKe	5.82e-05	4.38	3.42	1111	23
GRe	9.18e-05	4.31	4.93	1054	80
QRe	1.37e-04	4.33	5.10	1100	34
W	2.63e-04	4.40	3.93	1034	100
AK	2.75e-04	4.39	3.73	1083	51
NK	3.03e-04	4.37	3.51	1110	24
GR	6.16e-04	4.32	4.86	1051	83
QR	7.46e-04	4.34	5.01	1098	36
FRe	0.001	4.34	5.28	1111	23
IK	0.001	4.37	3.47	1113	21
IKe	0.001	4.37	3.47	1113	21
GKe	0.001	4.37	3.74	1104	30

Table 8.6: Results of two-sample t-tests of set s^+ (normalized intensities of peptides containing a substring s) against the set s^- of those not containing it in the corresponding dataset (MALDI, A and B). Only substrings with a p-value ≤ 0.001 are shown. Only substrings are shown that occur in more than 10 (A) / 20 (B) peptides. An "a" as a prefix denotes that the substring is located at the beginning of the string, "e" as a suffix means it is located at the end of the string. Else the substring occurs anywhere in the peptide. Gray shaded lines mark substrings that are present in the lists of both MALDI datasets.

substring s	p value	$\mu(s^-_{Y_2})$	$\mu(s^+_{Y_2})$	size$(s^-_{Y_2})$ s	size$(s^+_{Y_2})$
H	1.27E−11	6.95	6.52	3112	529
L	4.60E−07	6.70	6.96	1019	2622
aTP	2.59E−05	6.88	7.84	3619	22
aVL	3.62E−05	6.88	7.66	3596	45
M	7.92E−05	6.91	6.36	3508	133
EE	1.70E−04	6.91	6.57	3391	250
DE	4.22E−04	6.91	6.55	3423	218
QD	6.13E−04	6.91	6.29	3547	94
D	7.56E−04	6.97	6.81	1786	1855
HS	7.65E−04	6.90	6.16	3598	43
aII	9.81E−04	6.89	7.52	3614	27
DD	1.17E−03	6.91	6.56	3473	168

substring s	p value	$\mu(s^-_{Y_3})$	$\mu(s^+_{Y_3})$	size$(s^-_{Y_3})$ s	size$(s^+_{Y_3})$
H	1.23E−06	6.84	6.49	2098	340
L	9.75E−05	6.60	6.85	551	1887
M	1.03E−03	6.81	6.30	2345	93
aLV	1.13E−03	6.78	7.67	2417	21
PR	1.26E−03	6.80	6.33	2379	59
PRe	1.26E−03	6.80	6.33	2379	59
P	1.32E−03	6.72	6.88	1419	1019

Table 8.7: Results of two-sample t-tests of set s^+ (log mean intensities of peptides containing a substring s) against the set s^- of those not containing it in the corresponding dataset (ESI, Y_2 and $Y_3(2)^C F_5$). Only substrings with a p-value <= 0.001 are shown. Capital letters denote amino acids. An "a" as a prefix denotes that the substring is located at the N-terminal end (i.e. beginning of the string), "e" as a suffix means it is located at the C-terminal end (i.e. end of the string). Else the substring can occur anywhere in the peptide.

larger range than those of the MALDI datasets. The largest difference is found for peptides beginning with "VL" in both ESI datasets. Surprisingly, the presence or absence of R or K does not make a significant difference for the peak intensities. To investigate this further, feature importance in the *mono* feature set is tested with a random forest on dataset Y_2. Section 8.3.2) describes how accuracy decrease is calculated. In contradiction to the t-test results, K and R are rated most important, followed by H and L (see Fig. 8.10).

We can now compare the results between MALDI and LC-ESI datasets. The importance of single amino acid count features is similar between A and Y_2: For the MALDI datasets, dimers on the C-terminus (those at the end of peptide string) show a significant difference between the intensities, but non of the N-terminal dimers (beginning of the string). For the LC-ESI datasets, it is the other way around.

8.4. Summary

In this chapter, multiple feature selection methods have been applied and compared to find lower-dimensional reduced feature sets to reduce noise and enhance accuracy, and also extract knowledge about properties relevant to peptide-specific sensitivities from the datasets.

It can be observed that both datasets behave quite differently: Reducing the dimensionality of the high-dimensional initial feature sets with feature selection methods leads to a slight increase in the generalization capability on the more noisy and more difficult to predict MALDI dataset B with ν-support vector regression (ν-SVR), but a slight decrease on the dataset that is predicted more accurately (dataset A). For random forest regression though, a clear increase over initially extracted feature sets is observed.

On dataset A, no prediction method using a selected feature set is better than ν-SVR with a simple set of single amino acid counts (*mono*). However, both the features selected by a forward stepwise selection heuristic (*sss*) and least-angle regression (*larf*) lead to reasonable prediction results. On the more noisy and more difficult to predict **dataset** B, on the other hand, features selected from A achieve an increase of prediction accuracy. A feature set that has been selected by multiple forward stepwise selections and filled up manually (*sss*), beats the best result on the noisier dataset B using the initially extracted feature sets. A feature set selected with least-angle regression (*larf*) leads to results similar to those of the *sss* feature set. So both LARS and the forward stepwise selection heuristic succeed in selecting relevant features from very high dimensional feature sets.

Therefore, **knowledge can be extracted from the selected features**. I can conclude that apart from hydrophobicity and basicity, properties already known to influence peptide-specific sensitivities (represented by peak intensities), properties related to the amino acids' conformation also play a role in MALDI MS. Nonetheless, secondary structure based features are only of low importance. Therefore, I hypothesize that not secondary structure but conformation in general is more relevant. The specified features of relevance can be derived from application of three different feature selection methods. Although, as assumed, different methods choose different sets of features from the redundant chemical property feature sets, they agree on the properties these are derived from. None of these properties alone suffices for a prediction. The physicochemical features are preferred over single amino acid counts.

The three most important single amino acids were found to be arginine (R), methionine (M), and phenylalanine (F). Most important single amino acids in the peptide string reflect the physicochemical properties: arginine (R) and lysine (K) determine the basicity of tryptic peptides to a large degree, and histidine (H) is the only other amino acid that are basic. Phenylalanine (F) and tyrosine (Y) are both hydrophobic and aromatic, and comparably large. As such they might influence the conformation of peptides. Tryptophan (W) is of the same type but is quite rare, so it probably occurs too seldom in these small datasets to be of importance.

It is widely assumed that amino acid order, and not only their frequencies are important for peptide-specific sensitivity. If given the choice between features summed over amino acid index values of the components of a peptide and structure descriptors such as the differences of these values between amino acids of the peptide (Section 7.1.2), the feature selection methods almost always choose the structural features over the summed ones. The difference between the lowest and highest value for amino acids within a peptide is chosen most often, indicating that often the property of one of the amino acids dominates. The second often chosen structural features are based on the differences between amino acids in a distance of two or three in the peptide sequence. This is an indication that the order indeed is important, too.

Gay *et al.* (2002) also extracted knowledge from MALDI spectra, using an M5′ decision tree. The authors studied only very few physicochemical properties, but mainly used the fractions of certain amino acids per peptide. They were concerned that the small size of their dataset does not allow knowledge discovery but their findings might rather be attributed to that specific dataset. However, they also found phenylalanine (F), arginine (R), and methionine (M) to be of importance. Additionally, they found that asparagine (N), glutamic acid (E), and isoleucine (I) are not used in any rule of the decision tree. The results of this work acknowledge at least N and E to be of low importance. Gay *et al.* state that for their dataset the isoelectric point (pI) is not important. The corresponding amino acid index has not been chosen by any of the feature selection methods applied in this work. The only attribute that can not be acknowledged by the feature selection methods used here is that Gay *et al.* found the aliphatic index to be involved in the decision tree result.

From a computer scientist's perspective, it would be interesting to know which method works most reliably, or if multiple methods have to be applied every time, when this kind of redundancy is found in the data. The best selected set is the *sss* feature set, which has not been selected fully automatic. However, the set chosen by LARS comes close to it. LARS (glmpath) itself did not show good generalization accuracy, but the selected features lead to good (acceptable) generalization results with random forest regression.

LARS is the only method that made a good selection fully automatic. It is fast, easy interpretable, and the selected feature generalize well. There have been discussions about the C_p statistics as a model selection criterion for LARS (Ishwaran, 2004; Loubes and Massart, 2004; Stine, 2004). There has not been a direct comparison to other criteria with LARS in this work, but we can compare it to the selection with the closely related glmpath method that uses the BIC and AIC. In this application on data with more parameters than data points ($p > n$) and many correlated predictors, the C_p with LARS leads to a much better selection than the BIC or AIC with the glmpath method. The BIC and AIC were minimal for the full model with glmpath. However, on the second dataset B, it also selects the full model (and overfits). It would be interesting to use the S_p statistics with LARS for model selection instead: In an evaluation of Stine (2004), S_p leads to smaller models and $RMSE$ than C_p with LARS.

Additionally, random forest prediction results were presented and compared to ν-SVR results to validate the importance assessment. In most cases, the random forest prediction is less accurate than ν-SVR.

9. Extended analysis

This chapter presents the outcomes of some extended analyses that answer questions about the prediction. First, arising questions and the found answers are discussed. The second part shows the details of the applied analysis methods.

For which target values are the predictions best? (Section 9.1.1) The analyses show that low intensities are more difficult to predict than others. A possible reason is the noise behavior: Noise in mass spectra is additive and, hence, will have a stronger effect for low intensities. Also, noise in regions of lower intensities behaves differently from that of higher intensities (Anderle *et al.*, 2004). The use of two or more different models specialized to different intensity ranges might overcome this problem.

Do the learning methods predict the true signal? (Section 9.1.2) Sometimes, machine learning methods do not learn what we intend them to. This is not always obvious since the outcome of a prediction may be due to patterns in the data that have nothing to do with those that we aimed to find in the first place. With a shuffling experiment, this can be answered positively: Indeed, the true signal is predicted, i.e. the predicted intensities are correlated to the peptide sequences.

How do the prediction accuracies compare to Gay *et al.*? (Section 9.1.3) In their study, Gay *et al.* (2002) use multiple measurements of the same peptide in the regression, and allow peptides to occur in both the training and test set during cross-validation. This does not measure the ability of the applied models to *predict* but rather to *reproduce* the target values (logarithmic intensities). It must be mentioned that their work focusses on knowledge discovery rather than achieving the best possible prediction performance. However, in my work, peptides are used exclusively in training *or* test set, because I want to assess if and to what degree peak intensity prediction on new peptides is feasible. As a consequence, the presented results are not comparable to those of Gay *et al.* right away.

By allowing peptides to occur in the training and test set, and using duplicate values instead of one trimmed mean value per peptide, an evaluation corresponding to that of Gay *et al.* is set up for comparison purposes. With this setup, the more reproducible of the MALDI datasets is clearly "predicted" more accurately than the numbers reported by Gay *et al.* This suggests that my approach performs better on peak intensity prediction in MALDI, yet this cannot be stated with certainty without a comparison on the same dataset, which I could not acquire.

Is the relation between the peptide representations and the target values really non-linear? (Section 9.1.4) A linear model is unable to predict this relation for new peptides, whereas the applied non-linear methods achieve a significant positive correlation between predicted and target values. This indicates that the relation between the applied peptide representations and the target values is non-linear.

Can we incorporate unlabeled peptides in the dataset? (Section 9.1.5) A considerable portion of the theoretical peptides are not found by the peak extraction. Hence, they cannot be assigned a target value. Different ways of incorporating these peptides into the datasets are studied. The results show that it is of advantage to include unlabeled peptides using the peak intensities of noise peaks at the suspected m/z value as target value. A considerable improvement of the overall prediction accuracy can be achieved on the more noisy of the MALDI datasets (dataset B). For dataset A, a slight improvement can be observed.

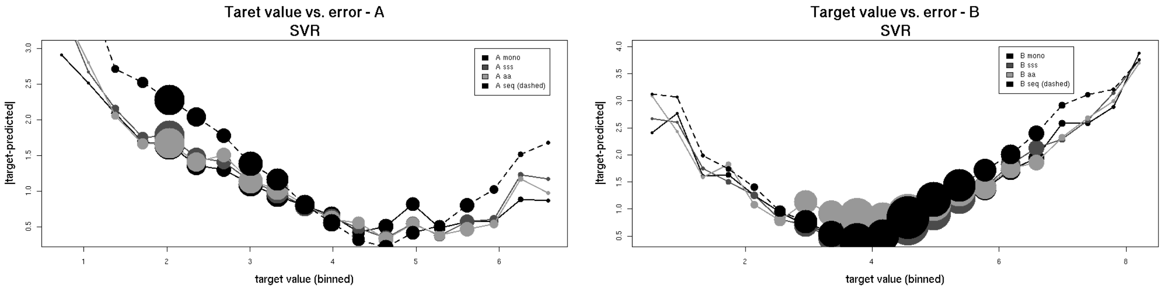
9.1. Detailed results

9.1.1. Analysis of error behavior

The most reliable prediction for both MALDI datasets is achieved for slightly above intermediate intensities. Low target values generally have a high prediction error (see Fig. 9.1). Often, prediction performance is better for areas in target value space where more samples are available. Here, this is not the case: Areas with higher errors do not agree with areas having a low number of examples, suggesting that low intensities are more difficult to predict. For Random Forest regression, these figures look similar (not shown).

9.1.2. Are the predicted intensities the true signal?

Concerned about wether a lot of peptide sequences might be wrong, this experiment is to ensure the presented results are due to the actual intensity signal. All peptide sequences of dataset A were shuffled randomly. Then the cross-validation was repeated with this scrambled dataset, the *mic* normalized target values, and the *sss* feature set.



(a) Dataset A

(b) Dataset B

Figure 9.1: Prediction error vs. target value for different feature sets with the MALDI datasets. Data was pooled into 20 bins according to the target value. For each bin, the mean absolute prediction error is plotted. The size of the dots shows the number of values falling into the corresponding bin.
The lowest error is achieved for intermediate target values, the highest error occurs for low ones. The region with minimum error differs slightly for both datasets. Dataset A has the minimum prediction error at slightly above intermediate target values. Unlike dataset A, the error increases again towards higher target values for B.

Result This leads to a correlation coefficient of $r^2 = 0.02$ in the mean with a standard deviation of 0.012 (100 runs). Fig. 9.2 shows a typical scatter plot. This is a clear indication that the learning algorithms pick up true signal, that is, the predicted intensities are correlated to the peptide sequence.

9.1.3. Duplicate peptides in training and test set

To compare our results to Gay *et al.* (2002), the logarithmic duplicate measurements of the peptides of the MALDI datasets are used instead of the logarithmic trimmed mean values, and peptides are allowed to occur in both the training and test set during cross-validation. This comparison still does not take into account how many peptides were present multiple times, how often they occur, or the between-peptide variance of the datasets.

Result Gay *et al.* measure a correlation coefficient of $r = 0.59$ using the M5′ algorithm, a decision tree method. This corresponds to $r^2 = 0.35$. Dataset A shows a noticeably larger correlation coefficient between target and predicted values than the result of Gay *et al.*, whereas dataset B is predicted a little worse (Table 9.1). This indicates that the approach chosen in this work performs quite well in comparison. However, this can only be stated with certainty, if the evaluation had been done with the same dataset. A comparison to the original dataset of Gay *et al.* would have been desirable but it could not be made available to me.

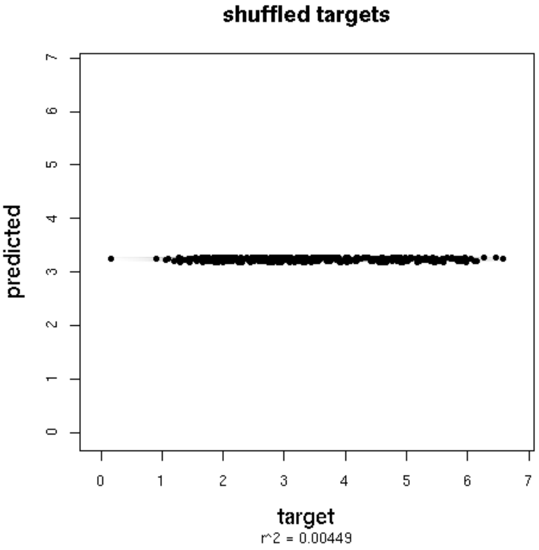


Figure 9.2: Prediction result when shuffling the peptide sequences of dataset A randomly. This clearly shows that we predict an actual signal that is related to the peptide's sequences.

Comparison to Gay *et al.*	
dataset and feature set	**prediction performance** [r^2 (r)] **in a 10-fold cross-validation**
A dupl. with *mono* features (ν-SVR)	0.61 (0.78)
B dupl. with *mono* features (ν-SVR)	0.28 (0.53)
Gay *et al.* (M5′)	0.35 (0.59)

Table 9.1: Comparison between prediction of duplicate measurements in this work and that of Gay *et al.* (2002) using ν-SVR and the *mono* feature set. Unlike with the rest of this work, in this test, duplicate measurements of the same peptide are allowed in training and test set to be able to compare the results to those of Gay *et al.*. Dataset B is comparable to their results, whereas dataset A shows a noticeably larger correlation coefficient between target and predicted values.

Linear model results				
validation dataset	**feature set**	**prediction performance** [r^2]		
		cross-validation	across dataset validation	across dataset strict validation
A	*mono* ν-SVR	0.46	0.44	0.37
	sss RF	0.39	0.37	0.30
	mono	0.27	0.20	0.15
	aa	0.36	0.02	0.00
	sss	0.26	0.22	0.17
B	*mono*	0.23	0.10	0.10
	aa	0.24	0.00	0.00
	sss	0.24	0.15	0.16

Table 9.2: Linear model results compared to best ν-SVR and random forest (RF) results.

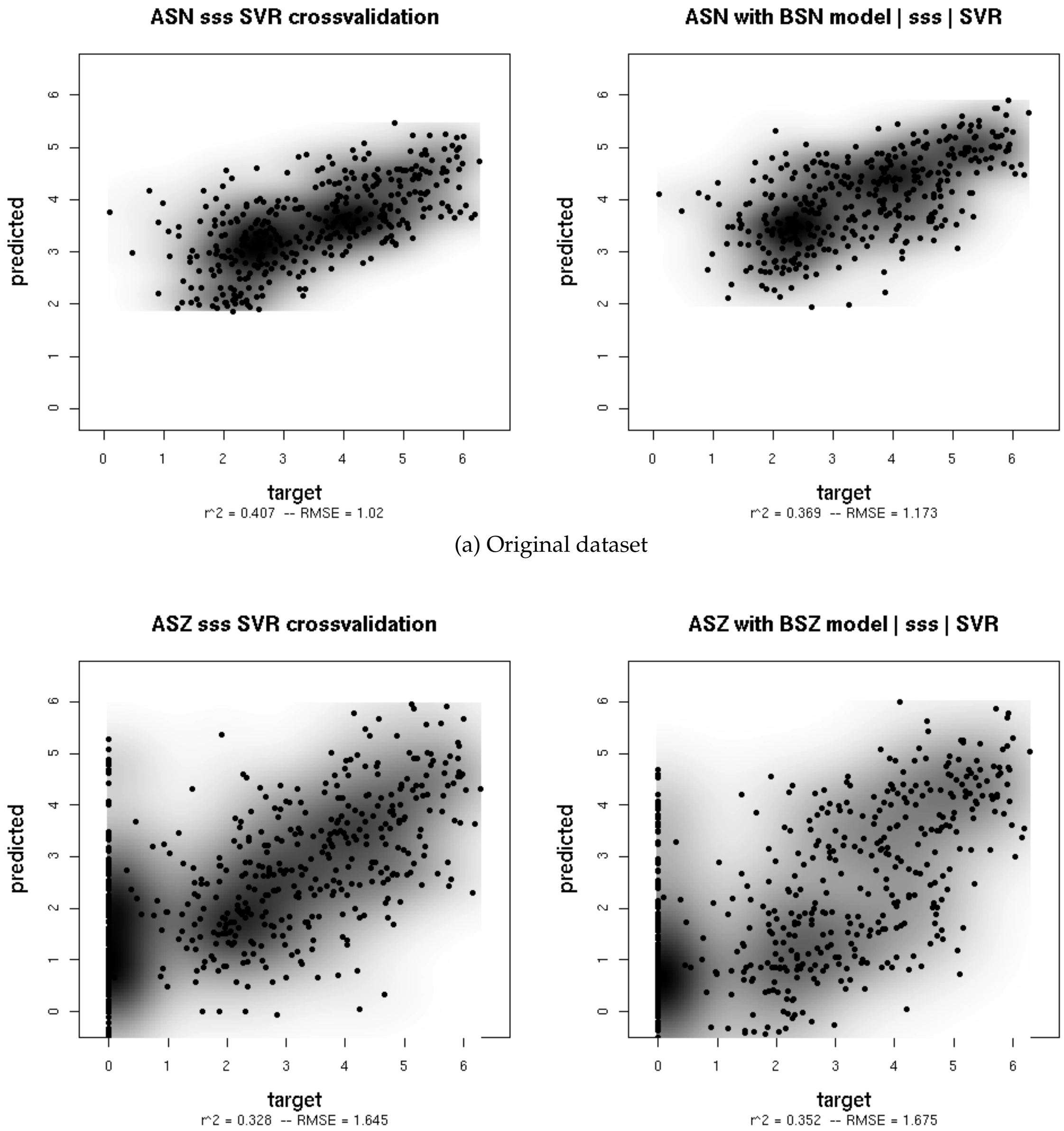
9.1.4. Linear model

Sometimes a linear model (LM) can outperform fancier methods. Therefore, an LM is applied to the *mono*, *aa*, and *sss* feature sets. While it can keep up with random forests in the cross-validation, the generalization performance is really bad (Table 9.2). This suggests a non-linear relationship between the feature sets and the normalized peak intensities.

9.1.5. Unlabeled data

A certain fraction of peptide strings that are generated in the simulated digestion step, no matching peak can be found during peak extraction. Hence, no target value exists for these peptides. This section deals with ways to incorporate these peptides to make use of any additional information that might be beneficial for the overall model.

Using zero as target value for unlabeled peptides One approach is to incorporate unlabeled peptides (those without target value) with a fixed target value of zero. As a result, the orientation of the point cloud in a target vs. predicted values scatter plot becomes more diagonal, which is good (Fig. 9.3). However, the spread also increases. Overall, the prediction of the labeled peptides gets worse. The unlabeled peptides are predicted with values in the whole range of target values with a mean above zero.



(a) Original dataset

(b) Dataset including unlabeled data with zero as target value

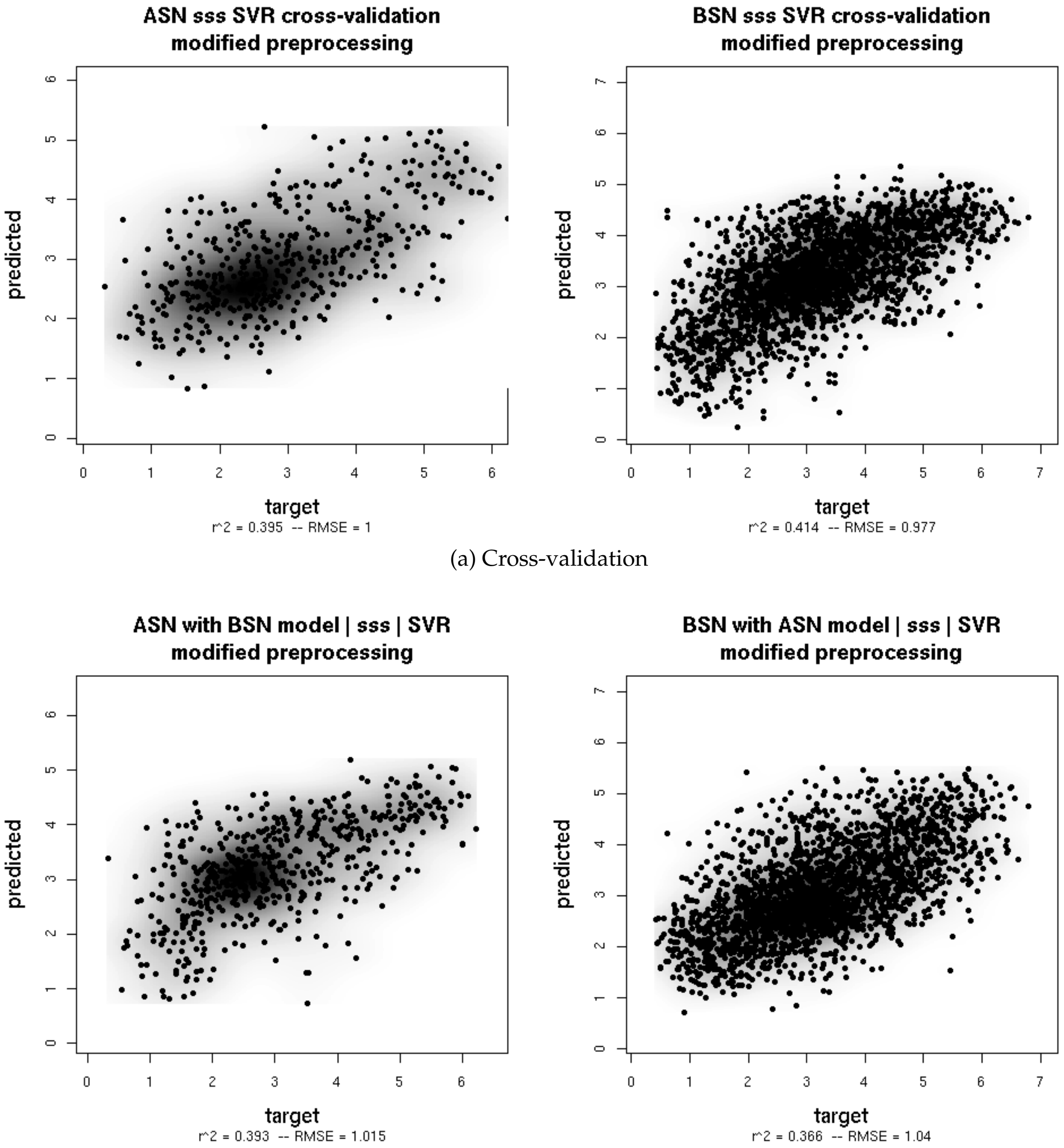
Figure 9.3: Comparison of prediction plots between a dataset without (*left*) and with (*right*) unlabeled data points (unobserved peptides) that have been incorporated into the dataset with target value of zero. Including unlabeled data with an artificial target value of zero leads to a higher variance in the non-zero predicted values. Qualitatively, a more diagonal orientation of the point cloud of labeled data can be observed. The predicted values for unlabeled data points cover almost the whole range of possible values and are a little too high in average.

Prediction results with unlabeled data included		
	10-fold cross-validation r^2(RMSE)	across dataset validation r^2(RMSE)
A	0.41 (1.014)	0.37 (1.173)
A modified preprocessing	0.40 (1.000)	0.39 (1.015)
A with unmatched peptides	0.33 (1.645)	0.35 (1.675)
B	0.29 (1.085)	0.18 (1.368)
B modified preprocessing	0.41 (0.977)	0.37 (1.040)
B with unlabeled peptides	0.34 (1.845)	0.30 (1.975)

Table 9.3: Comparison between datasets without unlabeled peptides to datasets from the same source (*sum* normalization), extracted with a modified preprocessing method that allows peaks below noise threshold to enter the dataset, and to the standard datasets (A, B) with unlabeled peptides added with a target value of zero.

Using noise peaks as target values For dataset B a considerable improvement is achieved when the modified preprocessing is applied (Table 9.3). For A there is a slight improvement in the across dataset validation. It can be suspected that this is due to B containing a larger fraction of unlabeled peptides. This shows that it is sensible to use the noise level instead of zero as a value for unlabeled peptides.

From the scatter plots (Fig. 9.4) it can be observed that the incorporation of unmatched peptides tilts the point cloud towards the diagonal. By adding values in the lower range, additional information is added in this range for the regression function.



(a) Cross-validation

(b) Across dataset validation

Figure 9.4: Results from prediction of datasets that have been extracted from the raw spectra using a modified preprocessing which extracts noise peak intensities for peptides that do not have a peak above noise level. This improves the prediction results of dataset B.

10. Conclusion

Summary

The aim of this thesis was to model peptide-specific sensitivities in mass spectra to enhance label-free protein quantification by mass spectrometry. Prior to this work, it was unknown if this is feasible at all. A combination of simulation and unsupervised learning methods for non-linear regression was chosen to map protein sequences to peptide sequences and these to peptide-specific sensitivities. As a model system, spectra from matrix-assisted laser desorption ionization (MALDI) of proteins separated with a two-dimensional gel (2D-PAGE) are used. In these spectra, often only one protein is found. This implies that extracted peak intensities in these spectra can directly be interpreted as peptide-specific sensitivities.

The first focus here was on the determination of an appropriate encoding of peptides as input for the learning algorithm. An evaluation of feature selection methods to extract knowledge from the spectra constitutes the second part of this work. Three different learning algorithms have been applied and compared: ν-support vector regression (ν-SVR), Random Forest regression, both non-linear methods, and least squares, a linear method. As input for the regression methods, two alternative peptide encodings have been evaluated: purely sequence-based encodings, and others that use additional chemical information. For feature selection, a forward stepwise selection heuristic and two L_1-penalized shrinkage methods were evaluated. Two small MALDI datasets have been studied, and first tests to transfer this approach to electrospray ionization (ESI) spectra have been committed.

Conclusion

This work is the first to evaluate the prediction of peptide-specific sensitivities on new peptides. The results presented in this thesis show that the prediction of peptide-specific sensitivities is indeed feasible. With the presented approach, significant correlations between target and predicted values are achieved. With ν-support vector regression (ν-SVR) and a low-dimensional purely sequence-based peptide representation, a squared Pearson's correlation of $r^2 = 0.46$ is achieved in ten-fold cross-validation, and $r^2 = 0.37$ for the prediction with a model trained on a different dataset. The supervised non-linear regression method ν-SVR outperforms the other methods in most cases, again underlining the great prediction capabilities of this method. This setup also outperforms the results of Gay *et al.* (2002), who allowed the same peptides to occur in training and test set.

The **feature selection methods rediscover properties already known to influence peptide-specific sensitivities**: The selected features are related to basicity, hydrophobicity, secondary structure. The higher sensitivity towards arginine-containing peptides is acknowledged. This shows that the selected features make sense. In addition, the feature selection results indicate that **conformation in general** and not necessarily certain secondary structure elements **might be important**. Unfortunately, not much is known about the conformation of peptides ions in gas phase. Additionally, **Methionine, phenylalanine, tyrosine, and histidine residues are also found to influence peptide-specific sensitivities**. To gain a more detailed insight, I suggest that a biochemist ought to look into the details of the most important chosen features.

It is a valid question whether knowledge extracted from such small datasets can be trusted. Gay *et al.* (2002) asked the same question when they extracted knowledge from a MALDI dataset with M5′ decision trees. Our results regarding feature importance assessment for single amino acid counts on two MALDI datasets from *C. glutamicum* proteins agree with the results from Gay *et al.* on a larger dataset containing proteins from various species: They also found that arginine, methionine, and phenylalanine are important residues. The agreement of knowledge extracted from datasets stemming from different instruments and species suggests that **the datasets used in both studies are large enough for knowledge discovery**.

Feature selection for regression is one of the most important problems in statistics. Especially in biological applications, there are often a lot of candidate features to choose from. Of the feature selection methods applied in this work, least-angle regression (LARS) (Efron *et al.*, 2004) and a manually augmented set from a forward stepwise selection heuristic both result in the feature subsets with the best generalization accuracy. However, only LARS runs fully automatically. Its prediction accuracy is not very good in comparison with ν-SVR, but the selected feature set can be predicted with good accuracy with other non-linear regression methods than LARS.

A lot of extensions and discussions have been published since LARS has been published (see the review by Hesterberg *et al.* (2008)). The model selection criterion C_p used for LARS has been subject to various studies (Ishwaran, 2004; Loubes and Massart, 2004; Stine, 2004). Ishwaran 2004 conclude that C_p often selects too large models. Also, according to Khan *et al.* (2007), LARS is sensitive to outliers. In this work, LARS constitutes a good fully automatic selection on the more reproducible MALDI dataset, but overfits on the noisier dataset. In comparison, the path following L_1-penalized general linear model overfits with both the Bayesian and the Akaike information criterion for model selection. There is need to study and compare model selection criteria for LARS as well as related architectures and expansions on difficult datasets such as mass spectrometry data more closely.

Preliminary tests to transfer this approach to ESI data are not successful so far. Possible reasons are the uncertain abundance of proteins in the analyzed sample, as well as the more complex nature of ESI spectra, containing differently charged ions instead of only singly charged ones.

To sum up, this work is the first to evaluate the prediction of peptide-specific sensitivities on new peptides. The achieved prediction accuracies are promising and show that peptide-specific sensitivity prediction is feasible. Knowledge extraction with feature selection methods leads to the rediscovery of known as well as new properties that are relevant for this problem. The modern feature selection method least-angle regression allows fully automatic feature selection on MALDI data, but the model selection criterion it uses should be subject to further analysis. The performance of the presented prediction approach in a quantitative analysis has to be assessed in wet lab studies designed specifically for this goal, as outlined below. Another very important result is that based on its studies and new proposed feature computation approaches, we propose the first integrated pipeline for automated peak intensity prediction. This is a vital step towards the improvement of label-free quantitative proteomics. Now, researches can focus on the next step to solve this specific problem. In the following, I present potential directions for further improving the presented approach.

Future outlook

It was shown in this thesis that peptide-specific intensities can be predicted, and the results are promising, but we can not expect to achieve a perfect prediction, because the data itself is not reproducible enough. Still, the achieved correlations are lower than the correlations between peak intensities from replicate runs. These can be considered upper bounds to the maximum achievable correlation possible, so there is room for improvement here. To some extent, this discrepancy can be accounted to the small size of the datasets. Inaccuracies in the peptide encoding are another reason:

The amino acid indices that are used as a basis for peptide encodings with additional chemical information are mostly measured properties of single amino acids. Peptides are molecules: In general, their chemical behavior does not only depend on the constituting amino acids but also their order and conformation. Hence, the values derived from single amino acid for the whole peptide are only a rough approximation. To increase prediction accuracy further, whole peptide properties in gas phase should be calculated more precisely. Unfortunately, this is computationally intensive.

Estimated gas-phase basicity values as whole-peptide estimates have already been used in this work. A first step towards more accurate peptide encodings could be to calculate topological descriptors for properties of single atoms instead of whole amino acids. Quantum mechanics calculations could be used to estimate peptide ion conformation in gas phase. This type of calculations can be really slow. Thus, an evaluation of the computational costs has to show if this is feasible in a practical application. With an estimation of peptide conformation, more advanced features such as three-dimensional descriptors instead of topological ones, as well as estimation of the enthalpy of formation for ionization would be possible.

The lack of really large datasets is one of the main problems encountered in this work. Ranking and binning approaches that balance the target value distribution as presented in Timm *et al.* (2006) are promising but suffer from the small dataset sizes. The number of data points falling into a single bin or rank is too low for these approaches to be of use in this case. Another possible improvement regards the available information: If a data pipeline had been available for the MALDI spectra, additional information such as modifications and peptides with missed cleavages would have been easily accessible, and could be used to lower the noise in these datasets.

For an LC-MS setup such as the LC-ESI data available for this work, knowledge about the abundances of proteins is necessary to train the models. This relates to one of the principle problems with the development of computational proteomics methods: Often there is no ground truth. In this special case, there are more exact protein abundance measurements available from Ghaemmaghami *et al.* (2003), but only for a subset of the identified proteins. Thus, dataset size can be traded off for more exact target values. The application of these measurements to scale peak intensities to more exact peptide-specific sensitivities should be the next step to transfer this approach to LC-MS data. If proteins are separated with a gel and fractions analyzed in different runs, variance is observed between these runs. This makes the use of peptides from proteins occurring in multiple fractions prohibitive. Additionally normalizing between these runs could help to solve this problem. Possible approaches are a simple normalization by variance or to spike a known amount of a standard protein into the analyzed protein mixtures. For the latter, a known amount of the protein bovine serum albumin (BSA) was spiked into the sample tube of each fraction prior to tryptic digestion. Not surprisingly, none of the BSA peptides could be identified in *all* of the fractions due to the undersampling typical for LC-ESI. The consequential step would be to connect different fractions via the detected BSA peptides. With a linear programming approach after an idea of Gunnar Rätsch (explanation in Timm (2005)), normalization coefficients that consider multiple peptides per fraction could be calculated. This would make normalization between almost all fractions possible, as an analysis of the detected BSA peptides shows. Finally, the application and evaluation of this approach in a quantitative analysis with MALDI is the most important next step. I propose two different studies: As a proof of concept, a synthetic mixture with known quantities should be analyzed with MALDI and prior 2D-PAGE separation. A correction of the constituting peak intensities by their specific sensitivities predicted with a trained model should lead to a mean abundance of 1, ideally with low variance. As a next step, a synthetic mixture of a small number of proteins in known quantities could be analyzed in the spectrum, using predicted sensitivities to estimate the abundance of these proteins. The main problem with the evaluation of this method on a more complex protein mixture is the absence of knowledge about the true abundances. Because there is no exact method to measure protein abundance, we can only correlate protein abundances predicted by this method to those estimated by other inaccurate methods such as TAP tagging (Ghaemmaghami *et al.*, 2003), and compare these correlations to those achieved by other methods for label-free quantitation (e.g. spectral counting methods, Ishihama *et al.* 2005; Rappsilber *et al.* 2002).

To acquire larger datasets, a high-throughput method such as LC-MS and a mixture of many proteins is necessary. Both are available, however, as of yet, exact knowledge about all the protein abundances in such a sample is infeasible to obtain with available technologies. We cannot derive an exact model from a large mixture of proteins where abundances are only known vaguely. It would be very helpful for the development of computational proteomics methods if there was a large standard mixture of proteins available.

Appendix A.

Additional information

A.1. Notations

$\mathbf{A}$ bold capital letters denote matrices

$\mathbf{A}^T$ a transposed matrix

$\mathbf{x}$ bold small letters denote vectors

$\mathbf{x}_j$ bold small letters with indices denote rows or columns of a matrix: j is used for columns, i for rows.

y_i small (non-bold) letters with an index denote scalar values, usually the value at the indexed position within a vector.

x_{ij} doubly indexed small letters are scalar values at row i and column j of a matrix

y small letters denote scalar values

$\mathbb{R}^n$ an n-dimensional Euclidian space

$\|\mathbf{x}\|_2$ the L_2 norm: $\|\mathbf{x}\|_2 = \sum_{i=1}^{N} (x_i)^2$

$\|\mathbf{x}\|_1$ the L_1 norm: $\|\mathbf{x}\|_1 = \sum_{i=1}^{N} |x_i|$

italic words in italic can be found in the glossary if they are not explained near where they occur

bold headings and important passages are printed in bold face

A.2. Abbreviations used for dataset variants:

A, B Data set A (A), Data set B (B)

M, S corrected mean ion current normalization (*mic*, M),
normalization by sum of peptide peak intensities (*sum*, S)

N, Z, X without unlabeled data (N),
including unlabeled data (Z),
including unlabeled data with noise values (X)

A.3. Implementation details

For application of the least-angle regression and path following method for L_1-penalized general linear models, the `glars 0.1.2.` package for the statistics toolkit R have been used. This is a beta package provided by Tim Hesterberg that is in development to overcome difficulties that still existed in `lars 0.9-6` at the time of this work (Hesterberg and Fraley, 2006). With this dataset (A, *mic* normalization, *cac* feature set) there were numerical problems when using the method from the `lars 0.9-6` package, probably because of the highly redundant dataset with much more features than data points. Used options are `method='s', type='lar', use.Gram=FALSE` for `glars`. For `glmpath`, the option `family='gaussian'` was set. Different criteria for the choice of the best contraint are implemented in the functions `lars` and `glmpath`: For `lars`, a C_p-type statistic is calculated, whereas for `glmpath`, the Bayesian (BIC) and Aikaike (AIC) information criteria are output.

The forward selection procedure was implemented in R by the author herself.

For ν-SVR, the `e1071 1.5-8` package for R with was used. The set parameters for all runs were `type='nu-regression', kernel='radial'.`

The package `randomForest 4.5-19` was used for Random Forest regression. For importance assessment, `importance=TRUE` was set.

For the linear model, built-in R function `lm` was used.

A.4. The official *IUPAC* amino acid codes.

The following table shows an overview of the twenty naturally occurring amino acids and their IUPAC codes. In addition, the amino acids can be sorted into different groups according to their properties. A black cells denote that the corresponding amino acid is in the corresponding group. The groups are explained in Section 7.1.2.

Amino acid	**one-letter code**	**three-letter code**	**aromatic**	**aliphatic nonpolar**	**acidic**	**basic**	**polar**	**small polar**	**most flexible**	**least flexible**	**sheet former**	**sheet breaker**
Alanine	A	Ala		■					■			
Cysteine	C	Cys					■			■		
Aspartic acid	D	Asp			■				■			■
Glutamic acid	E	Glu			■				■			■
Phenylalanine	F	Phe	■							■	■	
Glycine	G	Gly							■			■
Histidine	H	His				■	■	■		■		
Isoleucine	I	Ile		■						■	■	
Lysine	K	Lys				■						■
Leucine	L	Leu		■						■		
Methionine	M	Met		■						■		
Asparagine	N	Asp					■	■	■			
Proline	P	Pro		■					■			■
Glutamine	Q	Gln					■	■	■			
Arginine	R	Arg				■			■			
Serine	S	Ser					■	■	■			■
Threonine	T	Thr					■	■	■			
Valine	V	Val		■						■	■	
Tryptophan	W	Trp	■			■	■			■	■	
Tyrosine	Y	Tyr	■				■			■	■	

Table A.1: The twenty naturally occurring amino acids and their properties.

A.5. Overview of prediction results obtained in this work

dataset	feature set	ν-SVR		RF		LM		LLM*	
		10-fold cross-validation	strict across dataset validation	10-fold cross-validation	strict across dataset validation	10-fold cross-validation	strict across dataset validation	10-fold cross-validation	strict across dataset validation
A	*mono*	**0.46**	**0.37**	0.39	0.25	0.27	0.15	0.40	0.27
	seq	0.32	0.14	0.35	0.16	0.12	0.05	0.12	0.04
	aa	0.44	0.34	0.33	0.26	0.36	0.00	0.27	0.22
	cac	0.37	0.23	0.40	0.24	–	–	–	–
	spec	0.38	0.29	0.30	0.02	–	–	–	–
	sss	0.44	0.34	0.39	0.30	0.26	0.16	0.45	0.30
	larf	–	–	0.44	0.31	–	–	–	–
B	*mono*	0.22	0.21	0.24	0.23	0.23	0.10	0.26	0.20
	seq	0.19	0.10	0.21	0.21	0.17	0.00	0.07	0.00
	aa	0.28	0.20	0.23	0.24	0.24	0.00	0.21	0.18
	cac	0.27	0.20	0.26	0.27	–	–	–	–
	spec	0.18	0.12	0.29	0.01	–	–	–	–
	sss	**0.30**	0.28	0.29	**0.31**	0.24	0.16	0.28	0.21
	larf	–	–	0.23	0.20	–	–	–	–

Table A.2: Overview of prediction results on MALDI datasets with *mic* normalization, comparison between the learning architectures ν-support vector regression (ν-SVR), random forest regression (RF), least squares linear regression (LM), and local linear maps (LLM). The LLM training and evaluation have been carried out by Alexandra Scherbart. In strict across dataset validation, the model is trained and parameters are tuned on the respective other dataset, while the resulting correlation is calculated only for peptides in the evaluated dataset that do not occur in the other dataset. The numbers denote the squared Pearson's correlation (r^2) between predicted and target values.

Appendix B.

Glossary

α-trimmed mean A robust mean that trims away outliers. A list of values is sorted, and values both high and low are discarded such that α% of the list remains. The α-trimmed mean is then the mean of the remaining list of values.

Analyte A substance that is the object of interest in a laboratory analysis.

Bagging (Bootstrap aggregating) Procedure to improve the stability and accuracy of a machine learning (classification or regression) model. The new model is averaged over a sample drawn uniformly and with replacement from a training data set.

Boosting Algorithm for combining multiple classifiers into a single good one.

Bootstrapping is a resampling method from statistics that samples randomly with replacement from an original dataset, for example for variance or distribution estimation.

Chromatography Family of techniques for the separation of a mixture.

Collision chamber Device for ion fragmentation, often used as part of a tandem MS setup. Here, ions are fragmented into smaller pieces by collision-induced dissociation (CID). This involves the collision of an ion with a large neutral molecule in the gas phase.

Detector Part of a mass spectrometer that counts ions for each mass. See Section 2.2.

Dipolar Dipoles are chemical compounds that have unequally distributed electric charges, such as water. Dipolar molecules attract each other.

Dithiothreitol (DTT) is a reducing reagent that is used in protein biochemistry to reduce thio groups (SH-), for example to prevent them from disulfide bridges. Many proteins are stabilized by disulfide bridges. Treated in this way, they can not fold into their tertiary structure. As an example, DTT is used prior to proteolytic digestion. The proteolytic enzyme can cleave proteins more easily if they are denatured.

DNA Desoxyribonucleic acid. A macromolecule residing in each living cell that stores the genetic information of the individual living being.

Electrospray ionization A soft ionization technique used in mass spectrometry. See Section 2.2.

ESI *electrospray ionization*

Fourier transform Procedure to convert data between the time and frequency domain. With this technique it is possible to determine the different frequency components form a signal.

FT *fourier transform*

Gaussian mixture model Statistical method for clustering and density estimation. The model assumes a distribution to be constituted of overlayed Gaussian densities.

Hierarchical hill climbing Hill climbing algorithms are optimization methods that use local search, leading to a suboptimal solution (not the best but a good one). This is useful for problems where the search for the optimal solution is too slow, as for example the well-known traveling salesman problem.

Hydrophobicity Physicochemical property of a molecule that is repelled by water or other *dipolar* molecules. Hydrophobic molecules are not soluble in water and other dipolar solvents.

IF *isoelectric focussing*

Iodoacetamide An alkylating reagent that binds covalently with cysteine, preventing it from forming disulfide bonds within a protein. It is highly toxic, acts as a human carcinogen, and may cause reproductive damage.

Ion source Part of a mass spectrometer that produces ions. See Section 2.2.

pI *isoelectric point*

Isoelectric point (pI) The pH value of a solution at which the net charge of the (macro)molecules is zero. At this point, there is no motion of the particle in an electric field, which is useful for chromatography by electrophoresis.

Isoelectric focusing Electrophoresis with a pH gradient in a gel medium. Molecules stop to travel in the gel when they reach the location where the pH is equal to their isoelectric point (pI).

IUPAC The *International Union of Pure and Applied Chemistry* (IUPAC) is the recognized world authority on chemical nomenclature, terminology, standardized methods for measurement, atomic weights and many other critically evaluated data.

LC *Liquid chromatography*

Liquid chromatography Separation technique for organic molecules or biopolymers. See Section 2.3.

MALDI *Matrix-assisted laser desorption ionization* – Section 2.2.

Mass analyzer Part of a mass spectrometer that separates ions by their mass and charge. See Section 2.2.

Mass spectrometry "Analytical technique for measuring the mass-to-charge ratio of ions that can be used to identify unknown compounds, determine the isotopic composition of elements in a compound, and quantify the amount of a compound in a sample." (http://jilawww.colorado.edu/research/glossary/glossary_m.html)

Matrix-assisted laser desorption ionization A soft ionization technique used mass spectrometry. See Section 2.2.

Monad a single-celled microorganism (especially a flagellate protozoan)

MS *Mass spectrometry*

Neural network Often used when "artificial neural network" is meant. A machine-learning technique that simulates a network of communicating nerve cells.

Parent ion An ion formed in a mass spectrum, on which no fragmentation has occurred yet. When this ion is fragmented, it is called the parent ion or precursor ion of the contents of the fragmentation spectrum. The ions resulting from the fragmentation are called product ions.

Peptide fragmentation fingerprint (PFF) List of peaks (masses and intensities) in a fragmentation spectra – the spectrum of a parent ion after its fragmentation.

Peptide mass fingerprint (PMF) List peaks (often only masses) in an MS^1 mass spectrum.

PFF *peptide fragmentation fingerprint*

Plume A discharging gas cloud.

PMF *Peptide mass fingerprint*

principle component analysis

Protease A proteolytic enzyme. A protease cuts peptide bonds that link amino acids together in proteins or polypeptides.

Proteomics Field of research that deals with the exploration of the proteome, i.e. all the proteins that are present in a cell or organism at a point of time under defined conditions. The proteome is dynamic and its protein composition can change qualitatively and quantitatively due to changing outside conditions.

PTM *Post-translational modification*, see Section 2.1.1

Retention time Time that a compound needs to pass through a chromatographic column.

Sequence coverage The fraction of a protein that can be explained by the peptide masses found in an MS analysis. For example, if the masses of 3 tryptic peptides corresponding to 30 residues (= amino acid characters) were found by an MS analysis of a protein containing 100 residues, the sequence coverage would be 30%.

Standard error Standard deviation divided by the square root of the number of observations.

TOF Time-of-flight – a *mass analyzer*.

Translation Synthesis of proteins from mRNA molecules in living cells. The mRNA is built from DNA during *transcription*.

Trypsin A proteolytic enzyme or *protease*.

Acknowledgements

I have to thank a lot of people for supporting me during my PhD project. First of all, I'd like to thank my supervisor Jun. Prof. Tim W. Nattkemper for very helpful discussions about this work, science in general, and the perpetual reminder that the ring has to be cast into Mount Doom, while nobody is interested in Tom Bombadil. I also appreciate the help of Prof. Sebastian Böcker who supervised this thesis in its earlier phase, as well as Prof. Oliver Kohlbacher for his suggestions. Thanks to my workmate Alexandra Scherbart for lots of helpful discussion and crucial proof-reading, to her and Julia Herold for the "Mädelsabend" and other fun meetings, as well as to Julia Zakotnik for the backup while I wrote up this thesis and for the new hair cut.

Furthermore, I am thankful to Dr. Hanno Steen and his wife, Judith Jebanathirajah Steen, for their support and the great opportunity to work at the Proteomics Center of Children's Hospital Boston, MA, USA for a quarter year. The whole team of Steen lab was very pleasant and helpful, (and I loved how they love chocolate), and I learned a lot. Special thanks to Flavio Monigatti for discussions about computational issues and proof-reading. Thanks to Dalila Bensaddek, Sven Liffers, Yin-Yin Lin, Dr. Phil, and Zachary Waldon for help and instructions regarding wet lab work. I am grateful to all the friends I made in Boston, and who took part in making that visit rememberable: Julianne Gendrano, Lisa K, Matt Margeloni, Philip "Fil" Milnarik, Bailey Walker, Alex Piekarz, and Joe Puccio.

I also thank all the other proof-readers who were not mentioned yet: Epameinondas Fritzilas, Arne Hauenschild, Katharina Jahn, Jan Stallkamp, and Steve Wolter. Additional thanks go to Jan Stallkamp for providing the LaTeX template for this work and for being fussy about layout.

Thanks to the Genome Informatics group and their chair, Prof. Jens Stoye, helpful meta discussions, picking me up for lunch, field trips, and all the fun we had. Also, thanks to the Applied Neuroinformatics group for fun at the barbeques and Tim's ANI parties. I'd like to also thank the "Fachschaftsveteranen" (the people from the student council) for the weekly "Fachschaftsveteranenmensen" (lunch): Angelika Dierker, Katja Heinig, Christian "Paul" Mertes, Lars Schilling, and Markus Wiesecke, as well as the people from my other pen and paper roleplay group the enjoyable evenings (...and thanks for all the tentacles and pilzmoppeds): Jan Baumbach,Michael "Mibo" Bongenberg, Hendrik Hasenbein, Christian "Stolli" Stollenwerk, Tobias Wittkop, and Lasse Wittland.

The MALDI data has been provided by Nicole Hansmeier, Jörn Kalinowski, and Martina Mahne. Christian Rückert and Andreas Wilke helped with the protein identification. Thanks to all of these people. Thanks to Matthias Steinrücken for "needful functions" and Marcel Martin for advice with implementation details. I am grateful for the financial support I received from the NRW Graduate School for Bioinformatics and Genome Research as well as the Boehringer Ingelheim Fonds. Special thanks to Silke Kölsch for help with the bureaucracy.

Last but not least, I'd like to thank my family for their support, and Marcus Grieger, my companion in life.

Bibliography

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. Petrov and F. Csaki, editors, *2nd international symposium on information theory*. Akademiai Kiado.

Algower, E. and Georg, K. (1990). *Numerical Continuation Methods*. Springer.

Anderle, M., Roy, S., Lin, H., Becker, C., and Joho, K. (2004). Quantifying reproducibility for differential proteomics: noise analysis for protein liquid chromatography-mass spectrometry of human serum. *Bioinformatics*, **20**(18), 3575–3582.

Aresta, A., Calvano, C. D., Palmisano, F., Zambonin, C. G., Monaco, A., Tommasi, S., Pilato, B., and Paradiso, A. (2008). Impact of sample preparation in peptide/protein profiling in human serum by MALDI-TOF mass spectrometry. *J Pharm Biomed Anal*, **46**(1), 157–164.

Atkins, P. and Paula, J. D. (2006). *Physical Chemistry, Volume 1: Thermodynamics And Kinetics*. W. H. Freeman & Company, 8th edition edition.

Bahr, U., Karas, M., and Hillenkamp, F. (1994). *Fresenius' Journal of Analytical Chemistry*, **348**, 783.

Bantscheff, M., Schirle, M., Sweetman, G., Rick, J., and Kuster, B. (2007). Quantitative mass spectrometry in proteomics: a critical review. *Anal Bioanal Chem*.

Baumgart, S., Lindner, Y., Kühne, R., Oberemm, A., Wenschuh, H., and Krause, E. (2004). The contributions of specific amino acid side chains to signal intensities of peptides in matrix-assisted laser desorption/ionization mass spectrometry. *Rapid Commun Mass Spectrom*, **18**(8), 863–868.

Beavis, R. C., Chaudhary, T., and Chait, B. (1992). α-cyano-4-hydroxycinnamic acid as a matrix for matrix-assisted laser desorption mass spectrometry. *Organic mass spectrometry*, **27**, 156 – 158.

Berg, J. M., Tymoczko, J. L., and Stryer, L. (2006). *Biochemistry*. W. H. Freeman, 6th edition edition.

Berman, H., Henrick, K., and Nakamura, H. (2003). Announcing the worldwide Protein Data Bank. *Nat Struct Biol*, **10**(12), 980.

Berman, H., Henrick, K., Nakamura, H., and Markley, J. L. (2007). The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data. *Nucleic Acids Res*, **35**(Database issue), D301–D303.

Bishop, C. M. (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer.

Boehm, A. M., Pütz, S., Altenhöfer, D., Sickmann, A., and Falk, M. (2007). Precise protein quantification based on peptide quantification using itraq. *BMC Bioinformatics*, **8**, 214.

Bonner, A. and Liu, H. (2004). Comparison of Discrimination Methods for Peptide Classification in Tandem Mass Spectrometry. In *Proceedings of the IEEE 2004 Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB'04)*.

Breaux, G. A., Green-Church, K. B., France, A., and Limbach, P. A. (2000). Surfactant-aided, matrix-assisted laser desorption/ionization mass spectrometry of hydrophobic and hydrophilic peptides. *Anal Chem*, **72**(6), 1169–1174.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, **26(2)**, 123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, **45 (1)**, 5 – 32.

Breiman, L. (2002a). *Manual On Setting Up, Using, And Understanding Random Forests V3.1*.

Breiman, L. (2002b). *Manual On Setting Up, Using, And Understanding Random Forests V4.0*.

Budzikiewicz, H. and Schäfer, M. (2005). *Massenspektrometrie*. WILEY-VCH, 5th edition edition.

Buhrman, D., Price, P., and Rudewicz, P. (1996). Quantitation of sr 27417 in human plasma using electrospray liquid chromatography-tandem mass spectrometry: A study of ion suppression. *J. Amer. Soc. Mass Spectrom.*, **7**, 1099 – 1105.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**, 121–167.

Chamrad, D. C., Körting, G., Stühler, K., Meyer, H. E., Klose, J., and Blüggel, M. (2004). Evaluation of algorithms for protein identification from sequence databases using mass spectrometry data. *Proteomics*, **4**(3), 619–628.

Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Cherry, J. M., Ball, C., Weng, S., Juvik, G., Schmidt, R., Adler, C., Dunn, B., Dwight, S., Riles, L., Mortimer, R. K., and Botstein, D. (1997). Genetic and physical maps of Saccharomyces cerevisiae. *Nature*, **387**(6632 Suppl), 67–73.

Chiarugi, P. and Buricchi, F. (2007). Protein tyrosine phosphorylation and reversible oxidation: two cross-talking posttranslation modifications. *Antioxid Redox Signal*, **9**(1), 1–24.

Desiderio, D. M. and Kai, M. (1983). Preparation of stable isotope-incorporated peptide internal standards for field desorption mass spectrometry quantification of peptides in biologic tissue. *Biomed Mass Spectrom*, **10**(8), 471–479.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., and Weingessel, A. (2006). *The e1071 Package*. Department of Statistics (e1071), TU Wien, Friedrich.Leisch@ci.tuwien.ac.at. manual for R package e1071.

Dole, M., Mack, L. L., and Hines, R. L. (1968). Molecular beams of macroions. *J Chem Phys*, **49**, 2240.

Dongre, A. R., Jones, J. L., Somogyi, A., and Wysocki, V. H. (1996). Influence of Peptide Composition, Gas–Phase Basicity, and Chemical Modification on Fragmentation Efficiency: Evidence for the Mobile Proton Model. *J Am Chem Soc*, **118**, 8365–8374.

Douglas, D. J., Frank, A. J., and Mao, D. (2005). Linear ion traps in mass spectrometry. *Mass Spectrom Rev*, **24**, 1 – 29.

Draper, N. R. and Smith, H. (1998). *Applied regression analysis*. Wiley, 3rd ed. edition.

Dubois, F., Knochenmuss, R., Zenobi, R., Brunelle, A., Deprun, C., and Beyec, Y. L. (1999). A comparison between ion-to-photon and microchannel plate detectors. *Rapid Communications in Mass Spectrometry*, **13**(9), 786 – 791.

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*. Wiley and Sons, 2nd edition edition.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, **32**, 407 – 451.

Elias, J. E. and Gygi, S. P. (2007). Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat Methods*, **4**(3), 207–214.

Elias, J. E., Gibbons, F. D., King, O. D., Roth, F. P., and Gygi, S. P. (2004). Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nat Biotechnol*, **22**(2), 214–219.

Elias, J. E., Haas, W., Faherty, B. K., and Gygi, S. P. (2005). Comparative evaluation of mass spectrometry platforms used in large-scale proteomics investigations. *Nat Methods*, **2**(9), 667–675.

Fauchére, J. L., Charton, M., Kier, L. B., Verloop, A., and Pliska, V. (1988). Amino acid side chain parameters for correlation studies in biology and pharmacology. *Int J Pept Protein Res*, **32**(4), 269–278.

Fenn, J. B., Mann, M., Meng, C. K., Wong, S. F., and Whitehouse, C. M. (1989). Electrospray ionization for mass spectrometry of large biomolecules. *Science*, **246**, 64 – 71.

Foster, L. J., de Hoog, C. L., Zhang, Y., Zhang, Y., Xie, X., Mootha, V. K., and Mann, M. (2006). A mammalian organelle map by protein correlation profiling. *Cell*, **125**(1), 187–199.

Garcia, C. and Zangwill, W. (1981). *Pathways to Solutions, Fixed Points and Equilibria*. Englewood Cliffs: Prentice Hall.

Gasteiger, J. and Engel, T., editors (2003). *Chemoinformatics: A textbook*. WILEY-VCH.

Gay, S., Binz, P.-A., Hochstrasser, D. F., and Appel, R. D. (2002). Peptide mass fingerprinting peak intensity prediction: extracting knowledge from spectra. *Proteomics*, **2**(10), 1374–1391.

Gerber, S. A., Rush, J., Stemman, O., Kirschner, M. W., and Gygi, S. P. (2003). Absolute quantification of proteins and phosphoproteins from cell lysates by tandem MS. *Proc Natl Acad Sci U S A*, **100**(12), 6940–6945.

Ghaemmaghami, S., Huh, W.-K., Bower, K., Howson, R. W., Belle, A., Dephoure, N., O'Shea, E. K., and Weissman, J. S. (2003). Global analysis of protein expression in yeast. *Nature*, **425**(6959), 737–741.

Gobom, J., Kraeuter, K. O., Persson, R., Steen, H., Roepstorff, P., and Ekman, R. (2000). Detection and quantification of neurotensin in human brain tissue by matrix-assisted laser desorption/ionization time-of-flight mass spectrometry. *Anal Chem*, **72**(14), 3320–3326.

Gonzalez, J., Besada, V., Garay, H., Reyes, O., Padron, G., Tambara, Y., Takao, T., and Shimonishi, Y. (1996). Effect of the position of a basic amino acid on C-terminal rearrangement of protonated peptides upon collision-induced dissociation. *J Mass Spectrom*, **31**(2), 150–158.

Griffin, T. J., Gygi, S. P., Rist, B., Aebersold, R., Loboda, A., Jilkine, A., Ens, W., and Standing, K. G. (2001). Quantitative proteomic analysis using a MALDI quadrupole time-of-flight mass spectrometer. *Anal Chem*, **73**(5), 978–986.

Gunn, S. R. (1998). Support Vector Machines for Classification and Regression. Technical report, UNIVERSITY OF SOUTHAMPTON, Faculty of Engineering, Science and Mathematics, School of Electronics and Computer Science.

Gusev, A. I., Wilkinson, W. R., Proctor, A., and Hercules, D. M. (1996). Direct quantitative analysis of peptides using matrix assisted laser desorption ionization. *Anal Bioanal Chem*, **354**(4), 455–463.

Gygi, S. P., Rist, B., Gerber, S. A., Turecek, F., Gelb, M. H., and Aebersold, R. (1999). Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nat Biotechnol*, **17**(10), 994–999.

Han, Y., Ma, B., and Zhang, K. (2004). SPIDER: software for protein identification from sequence tags with de novo sequencing error. *Proc IEEE Comput Syst Bioinform Conf*, pages 206–215.

Hansmeier, N., Chao, T.-C., Pühler, A., Tauch, A., and Kalinowski, J. (2006). The cytosolic, cell surface and extracellular proteomes of the biotechnologically important soil bacterium Corynebacterium efficiens YS-314 in comparison to those of Corynebacterium glutamicum ATCC 13032. *Proteomics*, **6**(1), 233–250.

Harrison, A. G. (1997). The gas-phase basicities and proton affinities of amino acids and peptides. *Mass Spec Reviews*, **16**, 201–?217.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. (2004). The entire regularization path for the support vector machine. Technical report, Statistics Department, Stanford University.

Hesse, M., Meier, H., and Zeeh, B. (2005). *Spektroskopische Methoden in der organischen Chemie*. Thieme, 7th edition edition.

Hesterberg, T. and Fraley, C. (2006). S-plus and r package for least angle regression. In *Proceedings of the Joint Statistical Meetings 2006*. Insightful Corp. Proceedings published as CD ROM only.

Hesterberg, T. C., Choi, N. H., Meier, L., and Fraley, C. (2008). Least angle and l1 regression: A review. Technical report, Insightful Corp., University of Michigan, ETH Zürich. under review.

Hsu, C.-W. and Lin, C.-J. (Mar 2002). A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, **13**(2), 415–425.

Hsu, C.-W., Chang, C.-C., and Lin, C.-J. (2003). A Practical Guide to Support Vector Classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University.

Iavarone, A. T., Jurchen, J. C., and Williams, E. R. (2000). Effects of solvent on the maximum charge state and charge state distribution of protein ions produced by electrospray ionization. *J Am Soc Mass Spectrom*, **11**(11), 976–985.

Ishihama, Y., Oda, Y., Tabata, T., Sato, T., Nagasu, T., Rappsilber, J., and Mann, M. (2005). Exponentially modified protein abundance index (emPAI) for estimation of absolute protein amount in proteomics by the number of sequenced peptides per protein. *Mol Cell Proteomics*, **4**(9), 1265–1272.

Ishihama, Y., Schmidt, T., Rappsilber, J., Mann, M., Hartl, F. U., Kerner, M., and Frishman, D. (2008). Protein abundance profiling of the Escherichia coli cytosol. *BMC Genomics*, **9**(1), 102.

Ishwaran, H. (2004). Discussion of "least angle regression" by efron et al. *The Annals of Statistics*, **32(2)**, 452–458.

Jarman, K. H., Daly, D. S., Petersen, C. E., Saenz, A. J., Valentine, N. B., and Wahl, K. L. (1999). Extracting and visualizing matrix-assisted laser desorption/ionization time-of-flight mass spectral fingerprints. *Rapid Commun Mass Spectrom*, **13**(15), 1586–1594.

Ji, C. and Li, L. (2005). Quantitative proteome analysis using differential stable isotopic labeling and microbore lc-maldi ms and ms/ms. *Journal of Proteome Research*, **4**(3), 734–742.

Kaltenbach, H.-M., Wilke, A., and Böcker, S. (2007). SAMPI: protein identification with mass spectra alignments. *BMC Bioinformatics*, **8**, 102.

Karas, M. and Hillenkamp, F. (1987). *International Journal of Mass Spectrometry and Ion Processes*, **78**, 53.

Karas, M., Bahr, U., and Giessmann, U. (1991). Matrix-assisted laser desorption ionization mass spectrometry. *Mass spectrometry reviews*, **10**, 335 – 357.

Karas, M., Glückmann, M., and Schäfer, J. (2000). Ionization in matrix-assisted laser desorption/ionization: singly charged molecular ions are the lucky survivors. *J Mass Spectrom*, **35**(1), 1–12.

Kawashima, S., Ogata, H., and Kanehisa, M. (1999). AAindex: Amino Acid Index Database. *Nucleic Acids Res*, **27**(1), 368–369.

Khan, J. A., Aelst, S. V., and Zamar, R. H. (2007). Robust linear model selection based on least angle regression. *Journal of the American Statistical Association*, **102**, 1289–1299.

Khanarian, G. and Moore, W. J. (1980). The Kerr Effect of Amino Acids in Water. *Aust J Chem*, **33**, 1727–1741.

Knochenmuss, R. and Zenobi, R. (2003). MALDI ionization: the role of in-plume processes. *Chem Rev*, **103**(2), 441–452.

Krause, E., Wenschuh, H., and Jungblut, P. R. (1999). The dominance of arginine-containing peptides in MALDI-derived tryptic mass fingerprints of proteins. *Anal Chem*, **71**(19), 4160–4165.

Krijgsveld, J., Ketting, R. F., Mahmoudi, T., Johansen, J., Artal-Sanz, M., Verrijzer, C. P., Plasterk, R. H. A., and Heck, A. J. R. (2003). Metabolic labeling of C. elegans and D. melanogaster for quantitative proteomics. *Nat Biotechnol*, **21**(8), 927–931.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*, **2**, 164–168.

Li, L., Wada, M., and Yokota, A. (2007). Cytoplasmic proteome reference map for a glutamic acid-producing Corynebacterium glutamicum ATCC 14067. *Proteomics*, **7**(23), 4317–4322.

Li, X., Obradovic, Z., Brown, C. J., Garner, E. C., and Dunker, A. K. (2000). Comparing predictors of disordered protein. In *Genome Inform Ser Workshop*.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *The Newsletter of the R Project*, **2/3**, 18 – 22.

Lin, Y. and Jeon, Y. (2002). Random forests and adaptive nearest neighbors. Technical report, Department of Statisticis, University of Wisconsin.

Lisacek, F., Cohen-Boulakia, S., and Appel, R. D. (2006). Proteome informatics ii: Bioinformatics for comparative proteomics. *Proteomics*, **6**, 5445 – 5466.

Listgarten, J. and Emili, A. (2005). Statistical and computational methods for comparative proteomic profiling using liquid chromatography-tandem mass spectrometry. *Mol Cell Proteomics*, **4**(4), 419–434.

Liu, C., Song, Y., Yan, B., Xu, Y., and Cai, L. (2006). Fast de novo peptide sequencing and spectral alignment via tree decomposition. In *Pacific Symposium on Biocomputing*.

Liu, H., Sadygov, R. G., and Yates, J. R. (2004). A model for random sampling and estimation of relative protein abundance in shotgun proteomics. *Anal Chem*, **76**(14), 4193–4201.

Loubes, J.-M. and Massart, P. (2004). Discussion of "least angle regression" by efron et al. *The Annals of Statistics*, **32(2)**, 460–465.

Lu, B. and Chen, T. (2004). Algorithms for de novo peptide sequencing using tandem mass spectrometry. *DDT: BIOSILICO*, **2**(2), 85–90.

Lu, P., Vogel, C., Wang, R., Yao, X., and Marcotte, E. M. (2007). Absolute protein expression profiling estimates the relative contributions of transcriptional and translational regulation. *Nat Biotechnol*, **25**(1), 117–124.

Ma, B., Zhanga, K., and Liang, C. (2005). An effective algorithm for peptide de novo sequencing from MS/MS spectra. *J Comp Sys Sci*, **70**, 418–?430.

Mack, L. L., Kralik, P., Rheude, A., and Dole, M. (1970). Molecular beams of macroions. ii. *J Chem Phys*, **52**, 4977.

Mallick, P., Schirle, M., Chen, S. S., Flory, M. R., Lee, H., Martin, D., Ranish, J., Raught, B., Schmitt, R., Werner, T., Kuster, B., and Aebersold, R. (2007). Computational prediction of proteotypic peptides for quantitative proteomics. *Nat Biotechnol*, **25**(1), 125–131.

Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, **11**, 431–441.

Marshall, A. G., Hendrickson, C. L., and Jackson, G. S. (1998). Fourier transform ion cyclotron resonance mass spectrometry: a primer. *Mass Spectrom Rev*, **17**(1), 1–35.

Matthiesen, R. (2007). Methods, algorithms and tools in computational proteomics: A practical point of view. *Proteomics*, **7**(16), 2815–2832.

Mayr, B. M., Kohlbacher, O., Reinert, K., Sturm, M., Gröpl, C., Lange, E., Klein, C., and Huber, C. G. (2006). Absolute myoglobin quantitation in serum by combining two-dimensional liquid chromatography-electrospray ionization mass spectrometry and novel data analysis algorithms. *J Proteome Res*, **5**(2), 414–421.

Meng, C. K., Mann, M., and Fenn, J. B. (1988). Of protons or proteins. *Zeitschrift für Physik D Atoms, Molecules and Clusters*, **10**, 361 – 368.

Miller, A. J. (2002). *Subset Selection in Regression*. CRC Press, 2nd ed. edition.

Mirgorodskaya, E., Braeuer, C., Fucini, P., Lehrach, H., and Gobom, J. (2005). Nanoflow liquid chromatography coupled to matrix-assisted laser desorption/ionization mass spectrometry: sample preparation, data analysis, and application to the analysis of complex peptide mixtures. *Proteomics*, **5**(2), 399–408.

Naderi-Manesh, H., Sadeghi, M., Arab, S., and Movahedi, A. A. M. (2001). Prediction of protein surface accessibility with information theory. *Proteins*, **42**(4), 452–459.

Nelder, J. and Wedderburn, R. (1972). Generalized linear models. *Journal of the Royal Statistical Society, Series A*, **135**, 370–384.

Nesvizhskii, A. I., Keller, A., Kolker, E., and Aebersold, R. (2003). A statistical model for identifying proteins by tandem mass spectrometry. *Anal Chem*, **75**(17), 4646–4658.

Neubert, H., Bonnert, T., Rumpel, K., Hunt, B., Henle, E., and James, I. (2008). Label-Free Detection of Differential Protein Expression by LC/MALDI Mass Spectrometry. *J Proteome Res*.

Nielsen, M. L., Savitski, M. M., Kjeldsen, F., and Zubarev, R. A. (2004). Physicochemical properties determining the detection probability of tryptic peptides in fourier transform mass spectrometry. a correlation study. *Anal Chem*, **76**(19), 5872–5877.

Old, W. M., Meyer-Arendt, K., Aveline-Wolf, L., Pierce, K. G., Mendoza, A., Sevinsky, J. R., Resing, K. A., and Ahn, N. G. (2005). Comparison of label-free methods for quantifying human proteins by shotgun proteomics. *Mol Cell Proteomics*, **4**(10), 1487–1502.

Olumee, Z., Sadeghi, M., Tang, X., and Vertes, A. (1995). Amino acid composition and wavelength effects in matrix-assisted laser desorption/ionization. *Rapid Communications in Mass Spectrometry*, **9 (9)**, 744 – 752.

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

Ong, S.-E., Blagoev, B., Kratchmarova, I., Kristensen, D. B., Steen, H., Pandey, A., and Mann, M. (2002). Stable isotope labeling by amino acids in cell culture, SILAC, as a simple and accurate approach to expression proteomics. *Mol Cell Proteomics*, **1**(5), 376–386.

Palagi, P. M., Hernandez, P., Walther, D., and Appel, R. D. (2006). Proteome informatics i: Bioinformatics tools for processing experimental data. *Proteomics*, **6**, 5435 – 5444.

Pappin, D. J., Hojrup, P., and Bleasby, A. J. (1993). Rapid identification of proteins by peptide-mass fingerprinting. *Curr Biol*, **3**(6), 327–332.

Park, M. Y. and Hastie, T. (2007). l_1 regularizatino path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B*, **69(4)**, 659 – 677.

Parker, K. C. (2002). Scoring methods in MALDI peptide mass fingerprinting: ChemScore, and the ChemApplex program. *J Am Soc Mass Spectrom*, **13**(1), 22–39.

Peng, J., Elias, J. E., Thoreen, C. C., Licklider, L. J., and Gygi, S. P. (2003). Evaluation of multidimensional chromatography coupled with tandem mass spectrometry (LC/LC-MS/MS) for large-scale protein analysis: the yeast proteome. *J Proteome Res*, **2**(1), 43–50.

Press, W. H. (1992). *Numerical recipes in C: The art of scientific computing*, chapter 14.8, pages 650–655. Cambridge University Press. sampling page savitzky-golay filters.

R Development Core Team (2006). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Rappsilber, J., Ryder, U., Lamond, A. I., and Mann, M. (2002). Large-scale proteomic analysis of the human spliceosome. *Genome Res*, **12**(8), 1231–1245.

Reidegeld, K. A., Eisenacher, M., Kohl, M., Chamrad, D., Körting, G., Blüggel, M., Meyer, H. E., and Stephan, C. (2008). An easy-to-use Decoy Database Builder software tool, implementing different decoy strategies for false discovery rate calculation in automated MS/MS protein identifications. *Proteomics*, **8**(6), 1129–1137.

Romero, P., Obradovic, Z., Li, X., Garner, E. C., Brown, C. J., and Dunker, A. K. (2001). Sequence complexity of disordered protein. *Proteins*, **42**(1), 38–48.

Ross, P. L., Huang, Y. N., Marchese, J. N., Williamson, B., Parker, K., Hattan, S., Khainovski, N., Pillai, S., Dey, S., Daniels, S., Purkayastha, S., Juhasz, P., Martin, S., Bartlet-Jones, M., He, F., Jacobson, A., and Pappin, D. J. (2004). Multiplexed protein quantitation in Saccharomyces cerevisiae using amine-reactive isobaric tagging reagents. *Mol Cell Proteomics*, **3**(12), 1154–1169.

Sanz-Medel, A., Montes-Bayón, M., del Rosario Fernández de la Campa, M., Encinar, J. R., and Bettmer, J. (2008). Elemental mass spectrometry for quantitative proteomics. *Anal Bioanal Chem*, **390**(1), 3–16.

Savitzky, A. and Golay, J. E. M. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.*, **36**, 1627 – 1639.

Schaller, J. (2000). Analysis of hydrophobic proteins and peptides by mass spectrometry. *Methods Mol Biol*, **146**, 425–437.

Schnier, P. D., Price, W. D., and Williams, E. R. (1996). Modeling the maximum charge state of arginine-containing Peptide ions formed by electrospray ionization. *J Am Soc Mass Spectrom*, **7**(9), 972–976.

Schölkopf, B., Bartlett, P., Smola, A., and Williamson, R. (1999). Shrinking the Tube: A New Support Vector Regression Algorithm. In *Advances in Neural Information Processing Systems*.

Schwarz, G. E. (1978). Estimating the dimension of a model. *Annals of Statistics*, **6(2)**, 461 – 464.

Schütz, F., Kapp, E. A., Simpson, R. J., and Speed, T. P. (2003). Deriving statistical models for predicting peptide tandem MS product ion intensities. *Biochem Soc Trans*, **31**(Pt 6), 1479–1483.

Shadforth, I., Crowther, D., and Bessant, C. (2005). Protein and peptide identification algorithms using MS for use in high-throughput, automated pipelines. *Proteomics*, **5**(16), 4082–4095.

Shao, J. (1997). An asymptotic theory for linear model selection. *Statistica Sinica*, **7**, 221–264.

States, D. J., Omenn, G. S., Blackwell, T. W., Fermin, D., Eng, J., Speicher, D. W., and Hanash, S. M. (2006). Challenges in deriving high-confidence protein identifications from data gathered by a hupo plasma proteome collaborative study. *Nat Biotechnol*, **24**(3), 333–338.

Steen, H., Jebanathirajah, J. A., Springer, M., and Kirschner, M. W. (2005). Stable isotope-free relative and absolute quantitation of protein phosphorylation stoichiometry by ms. *Proc Natl Acad Sci U S A*, **102**(11), 3948–3953.

Steen, H., Jebanathirajah, J. A., Rush, J., Morrice, N., and Kirschner, M. W. (2006). Phosphorylation analysis by mass spectrometry: myths, facts, and the consequences for qualitative and quantitative measurements. *Mol Cell Proteomics*, **5**(1), 172–181.

Stine, R. A. (2004). Discussion of "least angle regression" by efron et al. *The Annals of Statistics*, **32(2)**, 475–481.

Syka, J. E. P., Coon, J. J., Schroeder, M. J., Shabanowitz, J., and Hunt, D. F. (2004). Peptide and protein sequence analysis by electron transfer dissociation mass spectrometry. *Proc Natl Acad Sci U S A*, **101**(26), 9528–9533.

Tanaka, K., Waki, H., Ido, Y., Akita, S., Yoshida, Y., Yoshida, T., and Matsuo, T. (1988). *Rapid Communications in Mass Spectrometry*, **2**, 151.

Tang, H., Arnold, R. J., Alves, P., Xun, Z., Clemmer, D. E., Novotny, M. V., Reilly, J. P., and Radivojac, P. (2006). A computational approach toward label-free protein quantification using predicted peptide detectability. *Bioinformatics*, **22**(14), e481–e488.

Timm, W. (2005). *Peak Intensity Prediction in DNA Mass Spectra using Machine Learning Methods*. Diploma thesis, Bielefeld University, Germany. Supervisors: S. Böcker and T. W. Nattkemper.

Timm, W., Böcker, S., and Nattkemper, T. W. (2006). Peak intensity prediction for pmf mass spectra using support vector regression. In *Applied Artificial Intelligence - Proceedings of the 7th International FLINS Conference*, pages 565–572. World Scientific.

Tishirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, **58**, 267 – 288.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

Vapnik, V. (1982). *Estimation of Dependences Based on Empirical Data*. Springer-Verlag.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer, 1st ed. edition.

Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory*. Springer, 2nd ed. edition.

Vásquez, M., Némethy, G., and Scheraga, H. A. (1983). Computed conformational states of the 20 naturally occuring amino acid residues and of the prototype residue α-aminobutyric acid. *Macromolecules*, **16**, 1043 – 1049.

Vollhardt, K. P. C. and Schore, N. E. (2002). *Organic chemistry: Structure and function*. W. H. Freeman, 4th ed. edition.

Vucetic, S., Radivojac, P., Obradovic, Z., Brown, C. J., and Dunker, A. K. (2001). Methods for improving protein disorder prediction. In *Proceedings of the International Joint Conference on Neural Networks 2001 (IJCNN '01)*.

Wan, Y., Yang, A., and Chen, T. (2006). PepHMM: a hidden Markov model based scoring function for mass spectrometry database search. *Anal Chem*, **78**(2), 432–437.

Wang, W., Zhou, H., Lin, H., Roy, S., Shaler, T. A., Hill, L. R., Norton, S., Kumar, P., Anderle, M., and Becker, C. H. (2003). Quantification of proteins and metabolites by mass spectrometry without isotopic labeling or spiked standards. *Anal Chem*, **75**(18), 4818–4826.

Wenschuh, H., Halada, P., Lamer, S., Jungblut, P., and Krause, E. (1998). The ease of peptide detection by matrix-assisted laser desorption/ionization mass spectrometry: the effect of secondary structure on signal intensity. *Rapid Commun Mass Spectrom*, **12**(3), 115–119.

Wilce, M. C. J., Aguilar, M.-I., , and Hearn, M. T. W. (1995). Physicochemical basis of amino acid hydrophobicity scales: Evaluation of four new scales of amino acid hydrophobicity coefficients derived from rp-hplc of peptides. *Analytical chemistry*, **67**(7), 1210 – 1219.

Winston, R. L. and Fitzgerald, M. C. (1998). Concentration and desalting of protein samples for mass spectrometry analysis. *Anal Biochem*, **262**(1), 83–85.

Wu, W., Wang, G., Baek, S., and Shen, R.-F. (2006). Comparative study of three proteomic quantitative methods, dige, cicat, and itraq, using 2d gel- or lc-maldi tof/tof. *Journal of Proteome Research*, **5**(3), 651–658.

Wuhrer, M., Deelder, A. M., and Hokke, C. H. (2005). Protein glycosylation analysis by liquid chromatography-mass spectrometry. *J Chromatogr B Analyt Technol Biomed Life Sci*, **825**(2), 124–133.

Yamashita, M. and Fenn, J. B. (1984). Electrospray ion source. another variation on the free-jet theme. *J Phys Chem*, **88**, 4451 – 4459.

Yang, D., Ramkissoon, K., Hamlett, E., and Giddings, M. C. (2008). High-accuracy peptide mass fingerprinting using peak intensity data with machine learning. *J Proteome Res*, **7**(1), 62–69.

Yao, X., Freas, A., Ramirez, J., Demirev, P. A., and Fenselau, C. (2001). Proteolytic 18O labeling for comparative proteomics: model studies with two serotypes of adenovirus. *Anal Chem*, **73**(13), 2836–2842.

Zhang, Z. (2004). Prediction of low-energy collision-induced dissociation spectra of peptides. *Anal Chem*, **76**(14), 3908–3922.

Zhu, Y. F., Lee, K. L., Tang, K., Allman, S. L., Taranenko, N. I., and Chen, C. H. (1995). Revisit of MALDI for small proteins. *Rapid Commun Mass Spectrom*, **9**(13), 1315–1320.