Combinatorial optimization for affinity proteomics

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Dipl.-Inform. Hannes Planatscher
aus Bozen/Italien

Tübingen
2016

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

Tag der mündlichen Qualifikation:	01.07.2016
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Andreas Zell
2. Berichterstatterin:	Apl. Prof. Dr. Kay Nieselt

to my family

Abstract

Biochemical test development can significantly benefit from combinatorial optimization. Multiplex assays do require complex planning decisions during implementation and subsequent validation. Due to the increasing complexity of setups and the limited resources, the need to work efficiently is a key element for the success of biochemical research and test development.

The first approached problem was to systemically pool samples in order to create a multi-positive control sample. We could show that pooled samples exhibit a predictable serological profile and by using this prediction a pooled sample with the desired property.

For serological assay validation it must be shown that the low, medium, and high levels can be reliably measured. It is shown how to optimally choose a few samples to achieve this requirements. Finally the latter methods were merged to validate multiplexed assays using a set of pooled samples. A novel algorithm combining fast enumeration and a set cover formulation has been introduced.

The major part of the thesis deals with optimization and data analysis for Triple X Proteomics - immunoaffinity assays using antibodies binding short linear, terminal epitopes of peptides. It has been shown that the problem of choosing a minimal set of epitopes for TXP setups, which combine mass spectrometry with immunoaffinity enrichment, is equivalent to the well-known set cover problem.

TXP Sandwich immunoassays capture and detect peptides by combining the C-terminal and N-terminal binders. A greedy heuristic and a meta-heuristic using local search is presented, which proves to be more efficient than pure ILP formulations.

All models were implemented in the novel Java framework SCPSolver, which is applicable to many problems that can be formulated as integer programs. While the main design goal of the software was usability, it also provides a basic modelling language, easy deployment and platform independence.

One question arising when analyzing TXP data was: How likely is it to observe multiple peptides sharing the same terminus? The algorithms TXP-TEA and MATERICS were able to identify binding characteristics of TXP antibodies from data obtained in immunoaffinity MS experiments, reducing the cost of such analyses.

A multinomial statistical model explains the distributions of short sequences observed in protein databases. This allows deducing the average optimal length of the targeted epitope. Further a closed-from scoring function for epitope enrichment in sequence lists is derived.

Kurzfassung

Biochemische Testentwicklung kann signifikant von kombinatorischer Optimierung profitieren. Parallele Multiplex-Assays benötigen komplexe Planungsentscheidungen bei Implementierung und nachfolgender Validierung. Die Komplexität der Aufgaben und begrenzte Ressourcen machen effizientes Arbeiten wichtig für den Erfolg der Testentwicklung.

In dieser Arbeit wurde zunächst gezeigt wie Proben systematisch gemischt werden können um eine multi-positive Kontrollprobe zu erstellen. Probenmischungen haben ein vorhersagbares serologisches Profil. Somit kann ein Mischungs-Vorschlag erstellt werden, um in einem Schritt möglichst viele Analyten zu kontrollieren. Ausserdem müssen für die Validierung niedrige, mittlere und hohe Konzentrationen reproduzierbar nachgewiesen werden. Es wird gezeigt, wie einzelne Proben für Validierungsexperimente ausgewählt werden können, um dieses Ziel zu erreichen. Schlussendlich werden beide Vorgehensweisen kombiniert um Probenmischungen zur Testvalidierung zu berechnen. Der Algorithmus nutzt schnelle Aufzählungstechniken in Verbindung mit einem ganzzahlig linearen Programm zur Lösung des Problems.

Ein großer Teil dieser Arbeit beschäftigt sich mit der Optimierung und Datenanalyse für Triple X Proteomics Immunoaffinitäts-Assays, die an kurze lineare terminale Sequenzen von Peptiden binden. Das Problem, die kleinstmögliche Menge an Bindern für eine gegebene Menge an Proteine zu wählen, entspricht dem bekannten Mengenüberdeckungsproblem. TXP Sandwichimmunoassays kombinieren c- und n-terminale Binder um Peptide zu identifizieren. Es wird eine Metaheuristik vorgestellt, die das Problem der minimalen Binderselektion für diesen Assaytyp besser löst, als reine gemischtganzzahlige Ansätze. Alle Algorithmen wurden in der neuen Java-Bibliothek SCPSolver implementiert. Die Bibliothek ist entwicklerfreundlich, enthält eine Modellierungssprache, ist einfach einzubinden und ist multi-plattform-fähig.

Bei der Analyse von TXP Daten ist es wichtig zu wissen, mit welcher Wahrscheinlichkeit sich eine terminale Sequenz in einem Experiment zufällig wiederholt. Der Algorithmus TXP-TEA berechnet dies und das MATERICS-Verfahren leitet das Bindungsmuster eines Antikörpers aus Immunoaffinitäts-Massenspektrometrie-Daten ab.

Abschliessend wurde die Verteilung von kurzen Sequenzen in Proteomen durch ein statistisches Modell erklärt, aus dem sich auch die optimale Durchnittslänge für Zielepitope analytisch ableiten lässt. Des Weiteren wird eine geschlossene Bewertungsfunktion für Epitopanreicherung in Sequenzlisten vorgestellt.

Die Ergebnisse dieser Arbeit steigern die Effizienz des Ressourceneinsatzes und ergänzen die Möglichkeiten zur Datenauswertung von Immunoaffinitätsexperimenten.

Acknowledgments

I would like to thank Prof. Dr. Andreas Zell for advice and financial support, and Apl. Prof. Dr. Kay Nieselt for her helpful comments on the manuscript. Thanks to Dr. Oliver Pötz, Dr. Thomas Joos and Calvin Wiese for giving me the opportunity to contribute to the Triple X proteomics project from the early stages. I would like to thank Dr. Nicole Schneiderhan-Marra, Prof. Dr. Dieter Stoll, Dr. Christopher Pynn, Dr. Sonja Schneider, Dr. David Eisen, Dr. Thomas Schreiber, Dr. Jens Göpfert, Dr. Frederik Weiss, Dr. Stefanie Rimmele, Dr. Yvonne Beiter, Dr. Angela Filomena, Dr. Nadia Baur, Dr. Nico Weber, Helen Hammer, Benedikt Lang, Bart van den Berg, Cornelia Sommersdorf, Ragna Häusler, Nicola Groll, Andreas Steinhilber, Berthold Gierke, Elise Ross and Anna Glukhova for their support and datasets from the labs at the NMI.

I thank my first mentors when I started scientific work Dr. Holger Ulmer and Dr. Felix Streichert and my former colleagues at the ZBIT Dr. Jochen Supper, Dr. Andreas Dräger, Dr. Nikolas Fechner, Dr. Georg Hinselmann, Dr. Andreas Jahn, Karsten Bohlmann, Florian Mittag and Klaus Beyreuther for their friendship and enlightening discussions during all these years at the Sand. I would like to thank Michael Schober for his excellent contributions in the development of *SCPSolver*. Also I would like to thank my former students Anne Bonin and Niklas Kasenburg.

I thank my parents Anna and Karl Planatscher for their love and support. Finally I thank my beloved wife Verena and son Benjamin for, well, just everything.

Contributions

As common in modern science this work would not have been possible without the contributions of many others. The bioinformatics scientist depends on the colleagues in the wet lab providing high quality datasets to analyze and computing problems to solve. The author is deeply thankful to all contributors.

While some parts of this dissertation have been published by the author in scientific journals or presented at conferences as talks or posters, many sections further elaborate on these findings or are yet unpublished and provide novel approaches.

This declaration explains chapter-by-chapter which parts of the thesis are original contributions of the author, and which contributions were provided by others.

1. Introduction (pages 1-22)

   Fig 1.4 (page 9) showing the fragmentation of a peptide in an ion series has been published by Hannes Röst under the creative commons license on Wikipedia. The sub-chapter 1.3.1 (pages 10-13) on MS-based immunoassays is a shortened version of Weiß *et al.* (2014), with kind permission of Frederik Weiß. Also Fig. 1.5 is part of this article. Although this review article has been co-authored by the author, the main contributors to the article are Frederik Weiß and Oliver Pötz. The full review is certainly one of the best overviews on MS-based immunoassays.

2. The mathematical programming framework SCPSolver (pages 23-31)

   The software library SCPSolver and its architecture has been conceived, designed and implemented by the author. Student assistant Michael Schober masterly programmed and compiled some parts of the software, mostly the solver-packs and the initial version of the SolverFactory, under the direct supervision of the author. All figures, texts, code examples and mathematical models in this chapter are original contributions of the author.

3. Applications of combinatorial optimization for immunoassays (pages 32-48)

   3.1) (pages 32-37) The author and Stefanie Rimmele contributed equally to the article Planatscher *et al.* (2013b). Stefanie Rimmele carried out the experiments in the lab and analyzed the data, and created Fig. 3.2 and Fig. 3.3. The author conceived the algorithm and mathematical models, formulas, implemented the software, run the calculations on the input data and set up the web service. The author wrote the text of the article with input from Stefanie Rimmele, Thomas Joos and Nicole Schneiderhan.

   3.2) - 3.3) (pages 37-43) Nicole Schneiderhan and Angela Filomena provided the question to be solved and the input data required for the analysis shown in Fig. 3.6. The content in this chapter is original work of the author and has been presented as a poster at HUPO 2014.

3.4) (pages 44-48) Berthold Gierke provided the initial question, by walking into the authors office some day and asking how to place three replicates of eight different samples on a 5x5-array without inference. The author then formulated and analyzed the more general problem and wrote ProChOpt, as described in this subchapter.

4. Optimization for immunoaffinity MS (pages 49-60)

   Thomas Joos, Oliver Pötz and Calvin Wiese posed the question answered in this chapter at the beginnings of the TXP proteomics project. Jochen Supper was involved in the initial discussions and implemented the filter pipeline. All described algorithms, figures, tables, study design, calculations and the text are original works of the author. This content of this chapter has been published in Planatscher *et al.* (2010), except for the fixed-cost model.

5. Optimization for TXP-Sandwich-Immunoassays (pages 61-70)

   Oliver Pötz and Thomas Joos designed the idea for TXP-sandwich immunoassays (Joos *et al.*, 2007). The method to select the minimum set of binders has been conceived and analyzed by the author. These findings were subject of a patent application (Joos *et al.*, 2010), and have been presented at the 26th conference on Operational Research in Rome (Planatscher *et al.*, 2013a).

6. Identification of short terminal motifs using peptide mass fingerprinting (pages 71-96)

   The idea to identify short terminal motifs from complex digests has been conceived by the author. Mass spectrometric measurement data has been provided by Frederik Weiß and David Eisen, which conducted the experiments in the lab. The author has created data analysis, mathematical models, texts, figures and tables. This work has been published in Planatscher *et al.* (2014).

7. A model for the distribution of short epitopes in proteomes (pages 97-106)

   All content of this chapter is original work of the author.

8. Summary and concluding remarks (pages 106-109)

   All content of this chapter is original work of the author.

Reuse licenses have been obtained by the respective publishers for all articles listed above.

If not explicitly stated in this declaration all other content is either original, or else sources are of course marked and cited accordingly in the text. Also publications outlined in this declaration are again explicitly introduced and cited in the text itself.

Furthermore all trademarks, product names, trade names, and logos are the property of their respective holders.

Contents

Chapter 1

Introduction

Protein biomarker discovery and quantification ranks among the most important challenges in modern biomedical research. Many diseases, organ malfunctions, injuries, and treatment side effects can be potentially diagnosed by looking at the right protein profile. Contrary to the analysis of mRNA profiles, the screening of protein expression profiles allows for direct conclusions to be made about the molecular mechanisms involved in a certain condition, as many cellular processes are directly related to protein functions.

Unfortunately, many proteins are very difficult to measure and cannot be reliably quantified because they only occur in very low concentrations. E.g. albumin is up to *1,000,000,0000* times more abundant than some cytokines (Omenn, 2004).

Mass spectrometry (MS)-based protein profiling has become one of the key technologies in biomedical research and biomarker discovery. It allows for the parallel detection of a mixture containing a limited number of peptides. For qualitative and quantitative protein profiling of a complex sample, time-consuming sample fractionation steps, such as 2D gel electrophoresis or multidimensional chromatography, are necessary. In this way, small subsets of the sample are analyzed fraction by fraction. These fractionation methods are the limiting factor in MS-based protein analysis.

This bottleneck led to the development of modern techniques that combine known methods for better accuracy and sensitivity. Immunoaffinity-Mass Spectrometry-based approaches combine techniques based on antibodies with mass spectrometry, thereby increasing sample throughput and detection sensitivity by capturing proteins or peptides from the sample using protein- or peptide-specific antibodies (Anderson *et al.*, 2004a,b; Nicol *et al.*, 2008; Warren *et al.*, 2004; Weiß *et al.*, 2014). However, the drawback is the large number of antibodies needed - one antibody per protein. In mRNA-profiling, cDNA molecules bind to synthetic probes that are both easy to postulate and synthesize. This allows for the comparatively cheap production of high-density microarrays that cover a large portion of the known genome. Unfortunately, this is not applicable in the protein world, since protein-binding molecules cannot be easily synthesized. Nevertheless, efforts are ongoing to generate antibodies for the analysis of the plasma proteome by an immunoaffinity- MS-based approach (Whiteaker *et al.*, 2007).

TXP-antibodies are a new technique for this problem. These antibodies recognize shorter parts of peptides and are therefore reusable for many assays. The task to select

the right targets for TXP-antibodies can be supported by bioinformatics (Planatscher *et al.*, 2010, 2013a) and is one main topic of this thesis.

It will be shown that it is possible to translate this task into known optimization problems, which can be solved using different heuristic and exact methods. These were implemented in the novel Java software library named *SCPSolver*, which will be discussed in Chapter 3. This modeling framework itself is not restricted to the optimization problems outlined in this work, and is applicable to many problems that can be formulated as integer programs. Some example applications, outside the realm of TXP-antibodies, will also be discussed, including the optimal pooling of a sample for positive control (Planatscher *et al.*, 2013b), the selection of samples for assay validation with a minimum number of experiments, and the placement of samples on a planar array.

After the selection, immunization, and purification, the TXP-antibodies are analyzed for their binding specificity. Heretofore, this entails a substantial effort in the lab, as the analysis requires synthetic peptide libraries and numerous mass spectrometry experiments. The second part of this thesis will discuss algorithms for inferring the antibody-binding motif from a mass spectrum obtained from the digest of a common cell line after immunoprecipitation. The epitope prediction reveals the most enriched terminal epitopes. Three different algorithms provide scores for potential epitopes and/or motifs. TXP-TEA estimates the score by sampling random spectra from a peptide database. The algorithm MATERICS combines the predicted sequences into more complex binding motifs (Planatscher *et al.*, 2014). A third approach, an alternative to the sampling-based TXP-TEA, calculates the score from a statistical model that was specifically developed for this purpose. A comparison with library screenings shows that the predictions made by the novel methods are reliable and reproducible indicators of the binding properties of an antibody.

This introductory section gives a short overview of the biochemical and mathematical principles used and referred to in the following chapters.

1.1 Proteomics

Major achievements have been made in the large-scale study of biological systems in the last decades. Sequencing entire genomes enabled the development of new screening methods. mRNA-microarrays enable simultaneous expression-level measurement of thousands of genes. Next generation sequencing (NGS) promises even more insights and an even larger amount of data.

Despite these advances in genomics, many biological and medical facts cannot be explained solely at the genome or expression level. Organisms and cells are dynamic, remarkably adaptive, and complex systems that, even if their DNA is the same, can develop various amounts of phenotypic realizations. Measuring protein abundance in the cell helps to explain the biological development and processes better. Proteomics is the study of the *proteome*, the entire set of all expressed proteins, of a species. This includes

functions, structures, sub-cellular location, interactions, and possible modifications of the proteins. The qualitative and quantitative analyses of proteins under controlled conditions require highly sensitive analytical methods.

The detection and quantification of proteins and peptides by mass spectrometry is widely used and well established in proteome analysis and biomarker discovery (Steen and Mann, 2004). MS enables the identification of the molecular weight of a compound up to 500 kDA, even if amounts only in the femto or attomolar scale are found in the sample. In the 'top-down' approach (Madsen *et al.*, 2009), proteins are identified by the spectrum of their ion fragments. When using the 'bottom-up' approach (Aebersold and Mann, 2003), proteins are proteolytically digested and then identified by the detection of the resulting peptides. Tandem mass spectrometry (MS/MS) enables the sequencing of the proteolytic peptides by matching fragmentation patterns to spectra predicted from sequence databases, or by analyzing the mass differences in the spectrum.

If a complex protein mixture is analyzed, the sample is fractioned in one or more dimensions in order to divide the sample up into smaller portions according to a gradient. The most important fractionation techniques are liquid chromatography, 2D-PAGE Gel electrophoresis, and affinity chromatography. The digestion of the unprocessed protein extract, followed by the separation of the peptides using liquid chromatography and a read-out from a mass spectrometer is known as *shotgun proteomics* (Washburn *et al.*, 2001). Another important, and in certain cases complementary, family of screening methods are immunoassays. Immunoassays apply specifically produced antibodies for the detection of proteins. These methods are also called targeted proteomics because they are very selective and sensitive, and not conducted on a large scale. If the targets are very low in abundance, these methods still generally deliver very good results compared to mass spectrometry approaches.

An important biomedical application of proteomics is the identification of biomarkers. It has been shown that some proteins are indicators for severe diseases, including cancer, heart, vascular, and neurological conditions. Large-scale proteomics approaches can be used for the discovery of new biomarkers (Veenstra *et al.*, 2005) if a large number of peptides and proteins can be screened in a sample.

1.2 Use of antibodies for bioanalytics and diagnostics

Antibodies are complex and highly variable biomolecules that have evolved to recognize the structure of other biomolecules. While these highly specific binders identify harmful substances and antigens in the mammalian body, antibodies also became highly relevant as diagnostic tools in the 20th century. The most important use of antibodies is in the identification, and if possible, quantification, of the specific antigen in patient samples. Also, the antibodies present in a patient can be used for diagnosis. In that case, immunological traces, which are antibodies produced by the body after an infection, can be used to detect the infection and estimate the onset of symptoms. (Raem and Rauch, 2007)

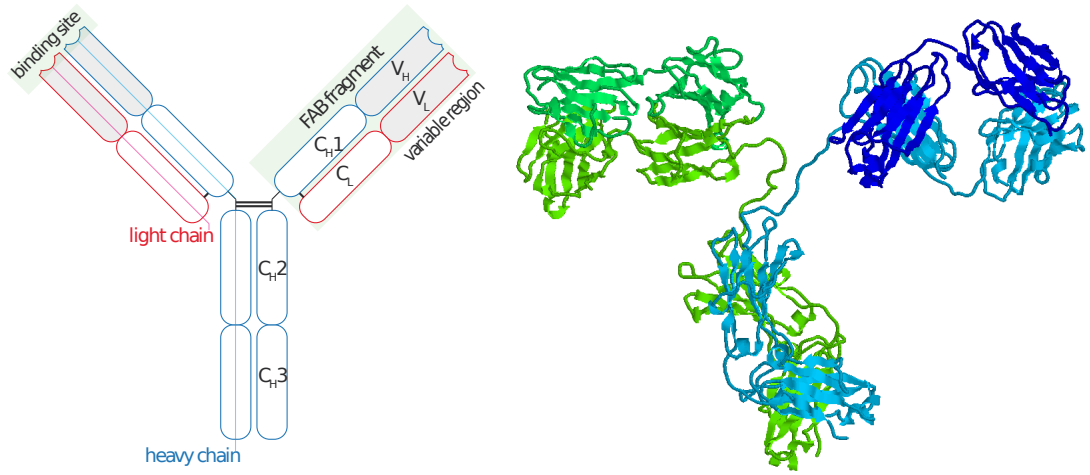


Figure 1.1: Domain structure of an immunoglobulin / rendering of a crystal structue (Protein Data Bank structure 1IGT visualized with Rasmol)

Antibodies specific for almost any biomolecule are conventionally obtained by immunization. The antigen is first injected in the body of an animal; the most commonly used species are rabbits, chickens, rats, mice, and goats. During this process, factors like size and dissimilarity to host-proteins will influence the probability of obtaining a good binder. While almost any structure can act as an antigen, only proteins will induce a full adaptive immune reaction. Adjuvants are added to the antigen to enhance its immunogenic properties by aggregating the antigen to particulates and delaying its release to prolong exposure to the organism.

After immunization, blood is extracted from the host organism. By removing all solid components and fibrinogen, the antiserum is isolated, which contains a variety of different antibodies. Some cross-reactive antibodies bind structures that are similar to the target epitope. These antibodies originate from different clones of maturated B-cells. The heterogeneous mixture of antibodies in polyclonal sera can be reduced by depletion steps. Still, due to the lifetime limits of the host animals and the different immune reaction of each organism, it is impossible to produce sera with the exact same properties (Raem and Rauch, 2007). Monoclonal antibodies are obtained by fusing murine spleen cells with murine myeloma cells. The resulting hybridoma cells have the properties of unlimited growth and the production of a mono-specific antibody. Cells can be selected by the specificity of the antibody to an antigen and be bred in cell lines. These cells are an unlimited source of antibodies with stable binding properties. Antibodies used for diagnostic kits or therapeutic treatments are generally monoclonal.

A method to produce antibodies in prokaryotic organisms is phage-display systems. By cloning human genes V_L and V_H into bacteriophages, the antibodies are displayed

on phage surfaces. Specific binders are isolated from large phage display libraries by selection and enrichment. These phages can then be used to introduce the genome in *E. coli* to produce large amounts of mono-specific antibodies.

ELISA immunoassays (**E**nzyme **L**inked **i**mmuno**s**orbent **a**ssay) can detect the direct binding of antibody and analyte. An antibody marked with an enzyme is used for detection. The antibody-enzyme complex specifically binds to the analyte bound by a captured antibody fixed on a solid phase. Alternatively, in the indirect assay, a primary antigen-specific antibody is detected by the marker antibody specific to the type of primary antibody. The same detection antibody can be used for all assays using the same kinds of antibodies, e.g. rabbit immunoglobulins.

1.3 Mass spectrometry for Proteomics

Mass spectrometry is an experimental method to determine the masses of molecule ions from a sample in a vacuum. The main components of a mass spectrometer are an ion source to produce a gas stream of ions from the sample, a mass analyzer to separate the ions according to their mass-to-charge (mz) ratio, and a detector to measure the ion stream. The resulting mass spectrum is a list of the relative amounts of detected ion mz-ratios (Lottspeich, 2006).

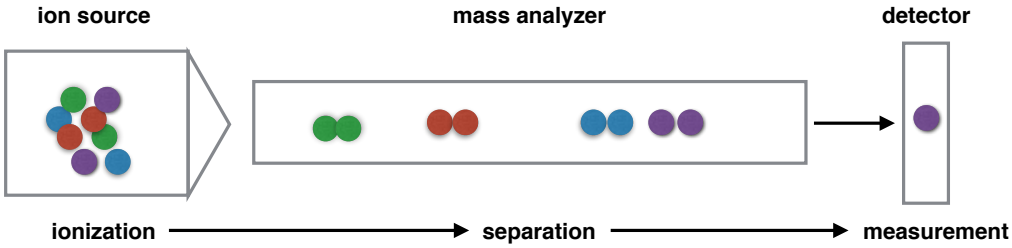


Figure 1.2: General structure of a mass spectrometer: ion source, mass analyzer, detector

Based on this mass spectrum, the chemical structure of complex molecules, e.g. the sequences of peptides, can be identified. Quantitative measurements are also possible, using either labeled standards or label-free approaches.

Ionization

In the ion source of the mass spectrometer, the analytes, which are brought into the gaseous phase, are ionized. Ionization is achieved through either the addition or loss of an electron. Methods for ionization of non-volatile compounds include electron and chemical ionization for gases and vapors. Electro-spray (ESI) (Yamashita and Fenn, 1984) and matrix-assisted laser ionization (MALDI) (Karas and Hillenkamp, 1988) are used for solid and liquid samples.

In MALDI, the ionization is performed by a nitrogen laser beam. A matrix of crystallized molecules protects the biological sample. When the laser hits the matrix, the energy is absorbed and ions are produced in the matrix. Part of the ion charge is transferred to the analytes. The ion species produced by the charge transfer are mostly ionized by one proton $[M+H]^+$. The resulting mz-values are $\frac{m+1u}{1}$, and are thus equal in value (but not in dimension) to the original molecular mass increased by the mass of a proton.

The electro-spray ionization method uses a strong electrical field for the generation of an aerosol of unipolar loaded molecules. The analytes are mixed with a liquid solvent and then forced through a metal capillary. A high voltage is applied to the tip of the capillary. The molecules in the solvent are positively charged and diffused into an evaporation chamber to the negatively charged inlet of the mass spectrometer. The molecules are mostly ionized by two protons and described as $[M+2H]^{2+}$, which leads to mz-values of $\frac{m}{2}+1u$.

Separation

After the ionization, the ions are transported to the mass analyzer by a magnetic or electrical field. The mass analyzer is a component for separating the ions according to the mass-to-charge ratio. A commonly used technique is to measure the time-of-flight (TOF) of an ion through a tube to a detector. The potential energy E_p of a particle in an electrical field with a potential difference U_f is proportional to its charge q:

$$E_p = U_f q \; . \tag{1.1}$$

The potential energy is transformed to kinetic energy $E_k = \frac{mv^2}{2}$, when the charged particle accelerates in the tube. Since the potential energy has to be the same as the kinetic energy, it follows that

$$U_f q = \frac{mv^2}{2} \tag{1.2}$$

which enables us to relate mass and charge to velocity v. The velocity is distance over time, which, in the case of the mass analyzer, is given by the length of the tube d and the time t passed until the impact of the particle on the detector. By inserting $\frac{d}{t}$ for v in the equation, and then solving for t :

$$t = \frac{d}{\sqrt{2U_f}}\sqrt{\frac{m}{q}} \tag{1.3}$$

Because $\frac{d}{\sqrt{2U_f}}$ can be considered constant, it is evident that the time-of-flight is proportional to the square root of the mass-to-charge ratio (Siuzdak, 2006).

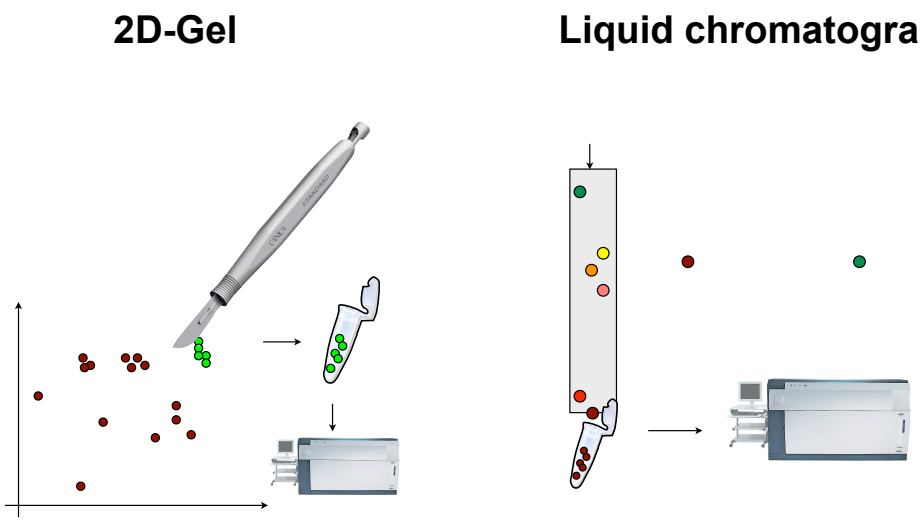


Figure 1.3: Classical fractionation techniques

Detection

After passing the mass analyzer, the ions reach the ion detector. This highly sensitive instrument is able to detect a minimum number of charged molecules. A common type of detector is an electron multiplier, which can be described as a linear array of high-voltage electrodes. Other designs do not use separated electrodes, but rather a single curve-shaped continuous electrode, but the principle of operation is the same: The ion stream impacts the first electrode, causing an electron emission. The emitted electrons induce a higher emission on the next electrode, and so on. This process amplifies the initial signal until it reaches a final collection anode. The signal is then further enhanced in a preamplifier before reaching a transient recorder, an array of high-speed analog digital converters to achieve maximum resolution (Siuzdak, 2006).

Sample Prefractionation

Samples containing complex mixtures of compounds are divided into smaller fractions before being inserted into the ion source. A common fractionation method is liquid chromatography.

Here, a mobile phase is forced to pass through a column containing the stationary phase. As different particles in the mobile phase exhibit different interactions with the stationary phase, some compounds pass the column faster than others. Another fractionation method applied to protein mixtures is sodium dodecyl sulfate polyacrylamide gel electrophoresis (SDS-PAGE). This method separates the proteins according to their electrophoretic mobility. So-called 2D Gels combine this approach with IEF (isoeletric

focussing), which separates the compounds by the relative content of their acidic and alkaline amino acid residues.

Peptide mass fingerprinting

Peptide mass fingerprinting is a well-established method in proteomics to identify proteins from the masses of their peptides (Yates *et al.*, 1993; Mann *et al.*, 1993; Henzel *et al.*, 1993; Pappin *et al.*, 1993; James *et al.*, 1993). A protein is first isolated using SDS-PAGE before being digested using a protease, such as trypsin, Glu-C, Lys-C. The digest is then measured in a mass spectrometer. The resulting mass spectrum is then compared to a database of protein sequences. Since the specificity of the protease and the masses of amino acids are known, it is possible to calculate their theoretical peptide fingerprints in silico. The more such a database pattern overlaps with an observed mass spectrum, the higher the resulting score. Usually, the software provides a ranking of matching proteins, along with scores, indicating the confidence of the protein identification. It has been shown that it is possible to elucidate the linear binding epitopes of antibodies using similar statistical approaches (Planatscher *et al.*, 2014).

The need to isolate the protein first is a major disadvantage if a sample contains many proteins that have to be identified. If the isolation does not work and the mass spectrum contains signals from 2 or more proteins, then this method will be prone to misidentifications.

Tandem mass spectrometry

In tandem mass spectrometry, two steps of mass spectrometric measurements and/or mass selection occur. In general, the first measurement determines the mass of an intact molecule, or more specifically, a peptide. If certain criteria are met, the molecule is fragmented and the fragment masses are determined. As the resulting pattern is often characteristic of the compound, as opposed to just the mass, it can be identified more easily. For example, the peptide SYFPHEIT has the exact same mass as EFYPHTIS, and 8!-2 other permutations, but only a few of these will have identical fragmentation patterns. Fragmentation is the central physical process. The most common techniques for fragmentation are collision-induced damage (CID), electron transfer dissociation (ETD), and post-source decay. In CID, the ions collide with an inert gas, such as a noble gas or purified nitrogen, in a collision cell (Siuzdak, 2006). This method is used with triple quadrupoles, quadrupole ion traps, Fourier Transform MS, and time-of-flight mass analyzers. ETD uses radical anions to induce the precursors fragmentation. ETD is used to fragment longer amino acid sequences (Syka *et al.*, 2004). Post-source Decay is a fragmentation method used in MALDI mass spectrometers.

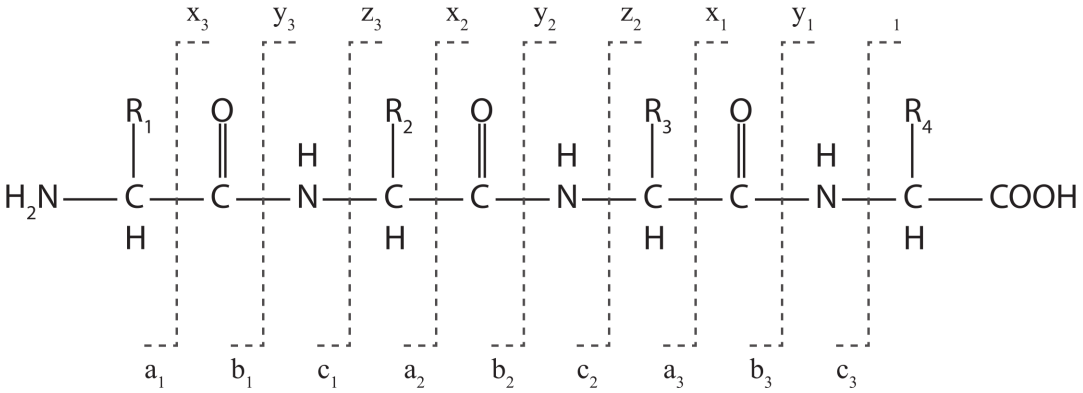


Figure 1.4: Peptide fragmentation leads to different types of fragments, some of which are detectable in mass spectrometers due to their charge. Fragments are denominated as a-, b-, and c-ions if the fragment is from the N terminal side, or as x-, y-, and z-ions if the fragments originate from the C terminal side. Image by Hannes Roest (Wikipedia) Creative Commons License by-sa

Peptide sequence identification

As described, the fragmentation patterns obtained from peptides in tandem mass spectrometry can be used for sequencing. The most common way to do this for known proteins is a database search. As with peptides, mass fingerprinting fragmentation patterns are compared to theoretical patterns that have been calculated in silico. If one or more spectra can be matched, the protein is considered as identified. The most popular tools for this method are MASCOT (Perkins *et al.*, 1999), Sequest (Eng *et al.*, 1994), XTandem (Craig and Beavis, 2004), and OMSSA (Geer *et al.*, 2004).

An alternative method to identify sequences that are not known and stored in a database a-priori is de-novo sequencing. These approaches analyze the fragment mass spectra by looking at the absolute differences of the mz-values. As fragmentation occurs at specific bonds, these mass differences are equivalent to masses of the amino acid at this position. By assembling all observed differences, the full sequence may be inferred. In practice, this is quite error-prone, but this might improve with advances in mass spectrometer resolution.

Multiple/Single Reaction Monitoring

If the mass of a peptide (precursor) and a characteristic fragment (product) ion is known, certain types of mass spectrometers (Triple-Quadrupoles) enable a specific search for and isolation of these compounds. This results in very sensitive and specific measurements with almost no background noise. It has been shown that the product ions signal intensities correlate with the parent peptides/proteins abundance in the sample. Alterna-

tively, a heavy-labeled ion is spiked in the samples. By comparing the intensity of this internal standard, absolute quantification of the endogenous peptide species is possible. In Multiple Reaction Monitoring (MRM), not just one many product ions are monitored, (Kondrat *et al.*, 1978).

1.3.1 MS-based Immunoassays

The following section on mass spectrometry-based immunoassays is a shortened version of Weiß *et al.* (2014). This review article has been co-authored by the author of this dissertation and explores the currently available mass spectrometry-based immunoassays shown in figure 1.5 in depth.

A popular approach taken for targeted protein assessment over the last decade involves combining an immunoaffinity enrichment step with mass spectrometric detection of two types: in (a) Mass Spectrometric ImmunoAssays (MSIA), the enrichment is performed at the protein level, whereas in (b) the Stable Isotope Standards and Capture by Anti-Peptide Antibodies assays (SISCAPA assays), it is done, as the name denotes, at the peptide level. With the latest improvements in mass spectrometry technology, plasma proteins in the pg/ml range could be detected. However, absolute sensitivities of MS-based immunoassays are still inferior to highly developed sandwich immunoassays, which are capable of detecting proteins in the fg/ml range (Ekins, 1998; Fredriksson *et al.*, 2002; Niemeyer *et al.*, 2005; Rissin *et al.*, 2010). The advantage of MS-based immunoassays is that the method is less error-prone than sandwich immunoassays, because the mass spectrometric read-out unambiguously confirms the identity of the analyte. Furthermore, only one capture molecule is required when an MS-based immunoassay is used. Additionally, the mass spectrometric read-out includes data that allows for discrimination between different protein isoforms.

Even though considerable effort has been made by large projects to develop a proteome-wide set of antibodies (Stoevesandt and Taussig, 2007, 2012), the unavailability of extensive antibody collections hinders the wider application of these assays by the scientific community. A prerequisite for the quantification of a tryptic peptide by means of a SISCAPA, referred to later, or iMALDI assay would be an antibody that was generated with a peptide-protein conjugate. It is common practice to develop a polyclonal or monoclonal antibody in order to obtain a suitable capture reagent (Stoevesandt and Taussig, 2012). The amount of polyclonal antibody that can be isolated from an immunized animal is sufficient to perform 20,000 MS-based immunoassays. On the other hand, a monoclonal antibody serves as an endless resource and can be produced on demand. The amount of required antigen for the antibody generation process is quite comparable. Commonly, two animals, typically rabbits or goats, are immunized to raise a polyclonal antibody.

Prior to the generation of such antibodies, it should be carefully evaluated whether the immunoprecipitation should be performed on the protein or peptide level. In the first case, the antibody has to be capable of binding the protein in its native intact form. In the second case, the protein is denatured and fragmented into peptides. For developing an

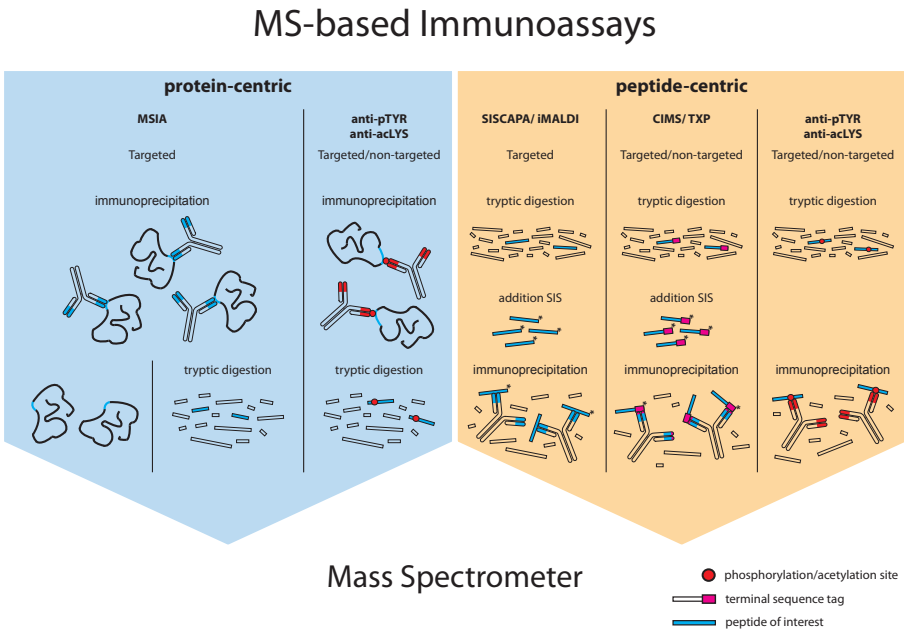


Figure 1.5: Different approaches to immunoaffinity mass spectrometry (image from Weiß *et al.* (2014))

MSIA, the protein target itself or a larger protein fragment should be used for immunization, presuming that the pure protein is sufficiently available. The recombinant synthesis of a full-length protein or protein fragment is expensive; however, for peptide-centric assays like SISCAPA or iMALDI, peptide synthesis can provide peptides for a fraction of the cost. Commonly, the peptide should contain an additional non-proteinogenic spacer and a cysteine, which are not part of the target peptide sequence, to achieve successful immunization.

Targeted MS-based immunoassays

Multiple Reaction Monitoring (MRM) mass spectrometry has evolved as a widely accepted multiplex method for quantifying analytes. However, LC-MRM assays lag behind proven immunoassays with sensitivities in the fg/mL range. The combination of two-dimensional LC setups with MRM detection will probably lead to higher sensitivities, but also to longer analytical separation times. The incorporation of a targeted enrichment step using antibodies led to further improvements in sensitivity and the development of mass spectrometry-based immunoassays. Here, the analyte the protein itself or a tryptic peptide thereof is enriched by an antibody prior to the mass spectrometric read-out. Anderson and colleagues published a method they called SISCAPA, in which protein quantification is achieved by the simultaneous immunoprecipitation of a tryptic fragment, derived from the target protein, and a stable isotopically-labeled syn-

thetic reference peptide. Subsequent detection and quantification of the protein is carried out using reversed phase liquid chromatography coupled with a triple quadrupole MS (Anderson *et al.*, 2004b). This rapid immunoaffinity enrichment step leads to analyte enrichment factors of more than 8,000 (Pagans *et al.*, 2009). Sensitivities for plasma proteins of these types of assays are typically in the lower ng/ml range (Pagans *et al.*, 2009; Neubert *et al.*, 2010; Hoofnagle *et al.*, 2008).

Moreover, the employment of magnetic particles allows for the use of large sample volumes (Anderson *et al.*, 2009) and enables samples to be re-analyzed (Whiteaker and Paulovich, 2011). By combining several antibodies in a single immunoprecipitation step, quantitative multiplexed assays for up to 50 analytes (Whiteaker and Paulovich, 2011; Whiteaker *et al.*, 2011) can be set up. The capacity to generate high throughput SISCAPA workflows was recently demonstrated (Razavi *et al.*, 2012). Here, the authors used a commercially available robotic interface to carry out simple, solid-phase extraction after the immunoprecipitation step and to inject the enriched peptide analyte directly into the mass spectrometer. As a result, the MS cycle-time could be reduced from approximately 20 minutes down to seven seconds, which permitted the analysis of 96 immunoprecipitates in just 15 min.

Mass Spectrometric Immunoassays (MSIA), as described by Randy Nelson and his colleagues, enrich intact proteins from plasma during immunoprecipitation in pipette tips and detect the protein via MALDI-MS (Nelson *et al.*, 1995; Tubbs *et al.*, 2006) or, alternatively, after digestion in MRM-MS (Krastins *et al.*, 2013; Lopez *et al.*, 2010). The advantage of the capture step at the protein level is that the information of the entire sequence is retained. Therefore, protein isoforms can be detected (Lopez *et al.*, 2010; Nedelkov *et al.*, 2006; Tubbs *et al.*, 2005) and post-translational modifications can be revealed. Nevertheless, protein degradation, solubility problems, and changes in the 3D structure can constrain the quantitative power of the MSIA approach.

The iMALDI method differs from the MSIA in that the analyte is directly eluted from the capture antibody onto the MALDI sample target by the application of an organic acid, which also serves as the MALDI matrix.

Novel MS-based immunoaffinity strategies are currently being developed for biomarker discovery projects. Pattern-based or peptide group-specific immunoaffinity enrichment enables the identification of peptide classes and can be applied to identify differentially and post-translationally modified proteins. Applications for phosphorylation, acetylation, methylation, ubiquitinylation, nitrosylation, and nitration are well described in the literature. The antibodies used in these strategies recognize distinct patterns in peptide classes, rather than targeting a single peptide.

This immunoaffinity enrichment strategy, applied as a pre-fractionation step before or after proteolytic digestion and MS detection, allows for the discovery-driven, system-wide screening of protein/peptide classes. One type of pattern exploited for immunoaffinity enrichment involves phosphorylation sites. These antibodies are capable of enriching specific phosphopeptide sequences (Mandell, 2003), phosphoserine or phosphothreoninepeptide motifs (Zhang *et al.*, 2002), or anti-phosphotyrosine (Ross *et al.*, 1981).

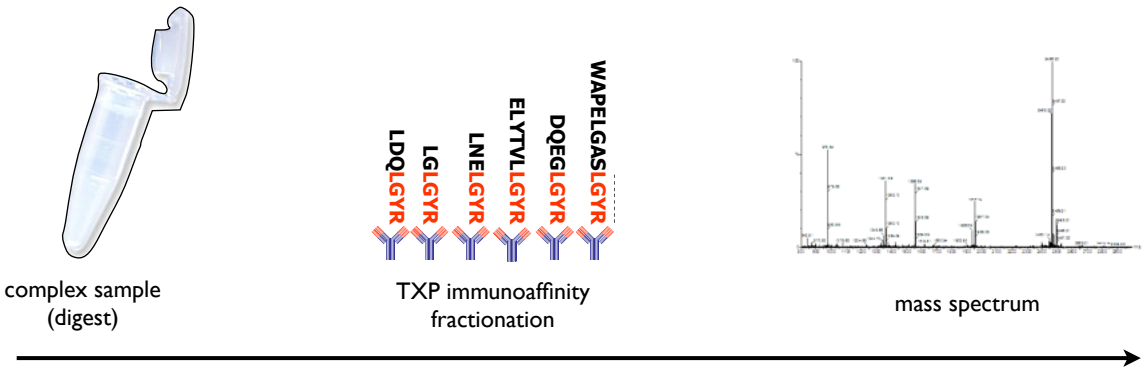


Figure 1.6: Motif-specific fractionation in the TXP workflow

Unlike anti-phosphotyrosine antibodies, anti-phosphoserine and threonine antibodies are sensitive to the amino acids surrounding the phosphorylation site (Alpha-Bazin and Quemeneur, 2012; Fíla *et al.*, 2012).

Motif-specific fractionation

Another generic pattern targeted by multi-specific binders is short sequence-based motifs. Sequences in each proteome are related through their evolutionary history. These protein families originated via the mechanism of gene duplication, where an existing gene is copied within the same organism. The second copy, while initially sharing the exact same sequence, can mutate in following generations. This may lead to a gene product performing a different task (neo- or subfunctionalization), but sharing a common domain. This leads to protein families, such as the Cytochrome P450 family, TG-Proteins, MAP Kinases, etc. With the TXP approach, it is possible to exploit sequence conservation in protein families by enriching peptides that share a conserved terminus.

Two approaches, the Global Proteome Survey (GPS) (Wingren *et al.*, 2009) and Triple X Proteomics (TXP) (Poetz *et al.*, 2009), employ binders capable of capturing short peptide motifs for affinity-based proteomic approaches. Wingren *et al.* (2009) described Context-Independent Motif-Specific (CIMS) antibodies, which they selected from a recombinant phage display library with C-terminal peptide motifs. The binders were then used for motif-specific pre-fractionation of peptides from tryptic digests, combined with mass spectrometric read-outs. A closer examination of the CIMS antibodies by *Olsson et al.* revealed relatively broad, off-target binding effects and dissociation constants in the M-range (Olsson *et al.*, 2012a). However, in a SILAC experiment, Olsson and colleagues demonstrated that the quantification of yeast proteins over 3 orders of magnitude is possible when CIMS antibodies are used in a precipitation step prior to LC-MS detection (Olsson *et al.*, 2011).

For the other motif-specific peptide capture technique, termed Triple X Proteomics

(TXP) (Poetz *et al.*, 2009), polyclonal sera were raised towards short-terminal peptide epitopes and purified using the target. The resulting antibodies were used to enrich and identify proteins from tryptically digested plasma (Volk *et al.*, 2012) or cell culture samples (Hoeppe *et al.*, 2011). Closer analyses of the antibody epitopes, by means of positional peptide libraries, revealed low off-target binding (Hoeppe *et al.*, 2011) and affinities in the two-digit nM range (Volk *et al.*, 2012). Recently, those antibodies were applied for the targeted analysis and the quantification of G-protein coupled receptors (Eisen *et al.*, 2013). Even very hydrophobic target peptides, which are normally not found in empirical mass spectrometry databases, could be identified and quantified after undergoing enrichment with TXP-antibodies. The antigens for the antibody generation are specifically selected *in silico* to cover proteotypic peptides from proteins of interest with a minimal set of TXP antibodies (Planatscher *et al.*, 2010).

1.4 Amino acid sequence databases

The bases of most calculations presented in this work are databases of primary amino acid sequences. While resources such as the PDB (Protein Data Bank) concern themselves with structural information, these amino acid databases focus either on complete protein sequences or on peptide sequences identified in tandem mass spectrometry experiments. Sequence databases can be classified by various criteria, including curation, redundancy, species, focus on a specific organ tissue, or body fluid.

1.4.1 UniProt

Combining the data in the databases Swiss-Prot, TrEMBL, and PIR-PSD resulted in the creation of the universal protein resource. The UniProt Consortium, formed in 2002 by the European Bioinformatics Institute (EBI), the Swiss Institute for Bioinformatics (SIB), and the Protein information Resource (PIR), takes care of the maintenance and annotation of the core databases: UniProtKB/SwissProt, UniProtKB/TrEMBL, UniParc, and UniRef.

UniProtKB/SwissProt contains only manually curated, reviewed, annotated, non-redundant, cross-referenced entries. Due to these criteria, the amount of protein sequences with scarce evidence for existence is low. Only 2.7 % of all sequences in the Release 2010_10 are based on prediction and 0.3 % have uncertain evidence. The remaining 97 % are based on evidence at the protein level, transcript level, or inferred from homology.

UniProtKB/TrEMBL is a repository for automatically annotated amino acid sequences, which are obtained by the translation of annotated coding sequences contained in EMBL-Bank, GenBank and the DDBJ sequence database, the Protein Data Bank or gene prediction (Ensembl, RefSeq, CCDS). While this ensures that all available sequence information is systematically collected and processed coherently, the lack of manual curation

may lead to redundancy. The **UniParc** database contains all available sequence information from the most important publicly available databases. To avoid redundancy, UniParc stores each sequence only once, even if the sequence originates from different species; each sequence gets a unique identifier. **UniRef** consists of a cluster of sequences from different organisms with varying levels of sequence identity (100 %, 90 % and 50%). The resulting entries are linked to the UniProtKB and UniProtParc entries of the sequences, which are contained in the cluster.

1.4.2 Peptide Databases for proteomics research

The Peptide Atlas is a database of peptides that were identified in tandem mass spectrometry experiments (Desiere *et al.*, 2006). The sequences of the peptides are identified using the search engines SEQUEST and X!Tandem. The identifications are then stored in the database along with the raw spectra. The aim of the Peptide Atlas is to annotate the sequenced genome of various species (*Homo sapiens*, *Mus musculus*, *Caenorhabditis elegans*, *Drosophila melanogaster*, etc.) with the sequences of peptides that are actually found by experiment. There are data packages in the Peptide Atlas that bundle the information for a single species.

The Global Proteome Machine Organization provides the GPMDB, which is also a central repository for mass spectrometry-based proteomics data (Fenyo *et al.*, 2010). Mass spectrometry data is stored along with sequence and experimental metadata. The organisms stored in the GPMDB are *Arabidopsis thaliana*, *Felis catus*, *Gallus gallus*, *Ovis aries*, *Drosophila melanogaster*, *Canis lupus familiaris*, *Cavia porcellus*, *Equus ferus caballus*, *Homo sapiens*, *Culicidae*, *Mus musculus*, *Oryctolagus cuniculus*, *Rattus norvegicus*, *Oryza sativa*, *Schizosaccharomyces pombe*, *Saccharomyces cerevisiae*, *Xenopus tropicalis* and *Danio rerio*.

The PRIDE database developed at the NCBI allows for the collection of mass spectra, peptide, and protein identifications, along with metadata (Martens *et al.*, 2005). In contrast to the other databases mentioned, there is no editorial control or a unified method for identification. Special features of PRIDE are the inclusion and mapping of biological ontologies (OLS) and an approach to unify protein accession numbers (PICR).

The ProteomExchange consortium envisions a unified approach in the form of a central mass spectrometry data repository by combining PRIDE, the Peptide Atlas, and Tranche (Hermjakob and Apweiler, 2006).

ProteomicsDB is the latest effort to map the human proteome and catalogue identified peptides by mass spectrometry (Wilhelm *et al.*, 2014). The proteomic data resulted from the identification of proteins corresponding to over 18,000 human genes. As of 2014, the database covers 93% of the known human proteome. This resource gained significant attention because it was developed using HANA, a remarkably fast in-memory database system provided by SAP.

1.5 Combinatorial Optimization

Combinatorial optimization problems are found in hundreds of different real world and academic applications. Finding the shortest routes in traffic, scheduling tasks, and the optimal allocation of resources are only a few examples. All of these problems deal with the curse of dimensionality, meaning that the number of feasible solutions increases rapidly with the size of input data.

Many problems can be solved using $O(n^k)$ steps for an input of length n. These algorithms are called polynomial, efficient, or good (Korte and Vygen, 2008). Fast polynomial time algorithms have been developed for special cases, such as the shortest path problem, the Euclidean minimum spanning tree, and the minimal assignment problem.

Some combinatorial problems are known to be NP complete. The interesting property of this class of problems is that every NP-complete problem can be reformulated to any other problem from that class. This means that if a fast algorithm is found to solve one problem of this class, then every other problem in this class can be efficiently solved.

It is important to note that for the applications described in this thesis, 0-1 integer linear programming has been shown to be NP-complete by Karp in 1972, among 21 combinatorial optimization problems (Karp, 1972). This particular technique is used for several applications in this thesis. Heuristics, such as a scatter search, taboo search, genetic algorithms, and other bio-inspired methods can find near-optimal solutions to NP-complete problems.

1.5.1 Linear Programming

Linear programming is the most important optimization method used in operations research today. It deals with the subject of solving problems of the form

$$max\, c^T x \tag{1.4}$$

subject to the constraints

$$Ax \leqslant b \text{ and } x \geqslant 0 \tag{1.5}$$

where x represents the vector of unknown variables, c and b are vectors of weights, and A is a matrix. Each inequality describes a half-space in an n-dimensional space. The intersection of all inequalities combined forms a convex polytope. While this form seems very restrictive and inflexible, it has been shown that many relevant problems can be formulated as linear programs. What is most intriguing about linear programming is that there exist very efficient algorithms to solve very large problems.

Each linear program can be transformed to a slack form by transforming all inequalities to equalities via the introduction of slack variables, e.g.

$$a_{1j}x_1 + \cdots + a_{ij}x_i + \cdots + a_{ij}x_n \leqslant b_j \tag{1.6}$$

is converted to

$$a_{1j}x_1 + \cdots + a_{ij}x_i + \cdots + a_{ij}x_n + s_j = b_j \tag{1.7}$$

In 1947, Georg Dantzig published the simplex algorithm. At present, this algorithm, or improvements of it, is the most widely used algorithm to solve linear programs. In graphical terms, the algorithm traverses the edges of the convex polytope, P, by moving from the edge to a better neighboring edge. Since the polytope is convex, an optimal solution has been reached when no adjacent edge represents a better solution than the current one.

Simplex Phase I The first phase of the algorithm searches for a feasible starting solution. To achieve this, another linear program is solved first:

$$min \sum_i z_i \tag{1.8}$$

subject to the constraints

$$Ax + z = b\, x, z \geqslant 0\ . \tag{1.9}$$

If this problem has a minimal solution with $z = 0$, then there must be at least one solution x, that satisfies all constraints. This x is then the starting point for the second phase. The minimal solution z to this helper problem is actually found by using Phase II. This is not a paradox, as due to its construction, the problem always has a trivial starting solution $(x,z) = (0,b)$. In case no solution $z = 0$ is found, the initial problem cannot be solved, because there is no vector x that would satisfy all initial constraints.

Simplex Phase II The second phase of the simplex algorithm essentially consists of the iterative solution of a system of linear equations using Gaussian elimination.

For this, a simplex tableau of the form

$$\begin{bmatrix} -\mathbf{c}^T & 0 \\ \mathbf{A} & \mathbf{b} \end{bmatrix}$$

is constructed. This structure is annotated as follows

		nonbasis variables			
		x_1	x_2	$\ldots$	
	$-c$	$-c_1$	$-c_2$		0
basis	s_a	a_{11}	a_{12}		b_1
	s_b	a_{21}	a_{22}		b_2
	s_c	a_{31}	a_{31}		b_3
	$\ldots$				

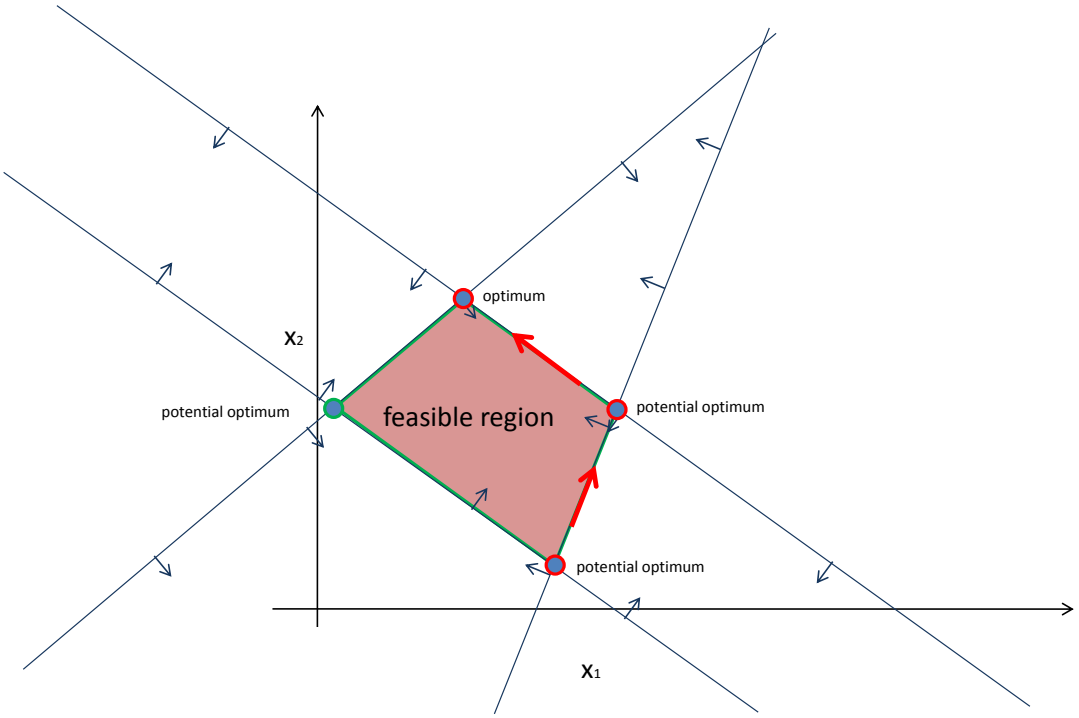


Figure 1.7: Illustration of the walk on the edges of polytope defined by the constraints in phase II of the Simplex algorithm

Non-basis variables have value 0, whereas basis variables are non-zero. Each simplex iteration consists of a pivot step, where a non-basis variable is introduced into the basis and vice-versa. The numbers in the matrix are updated according to the following rules.

The pivot element is set to its reciprocal:

$$a_{rs} = \frac{1}{a_{rs}} .$$

The other elements in the pivot row are normalized accordingly:

$$a_{rj} = \frac{a_{rj}}{a_{rs}}, \quad b_r = \frac{b_r}{a_{rs}},$$

and the pivot column is divided by the pivot elements value:

$$a_{is} = -\frac{a_{is}}{a_{rs}} \quad c_s = -\frac{c_s}{a_{rs}} .$$

All other elements are updated according to these rules:

$$a_{ij} = a_{ij} - \frac{a_{is}a_{rj}}{a_{rs}} \quad c_j = c_j - \frac{c_s a_{rj}}{a_{rs}} \quad b_i = b_i - \frac{b_r a_{is}}{a_{rs}}$$

This leads to a new basis solution. While positive coefficients are found in c, the solution can be improved.

Integer solutions Many combinatorial optimization problems are modeled using linear programs by introducing constraints requiring that some or all variables must take integer or binary values. This technique is named MILP (Mixed Integer Linear Programming).

By solving the 'relaxation' of the problem, omitting the integer constraints, the simplex algorithm finds an optimal integer solution only if the matrix A is totally unimodular. A matrix is totally unimodular if each quadratic submatrix has determinant ± 1. Surprisingly, this applies to many A-matrices arising in combinatorial optimization problems (Papadimitriou and Steiglitz, 1982), such as the incidence matrix of a bipartite graph (Heller and Tompkins, 1956).

If A is not totally unimodular, then cutting-plane algorithms are applied. These methods iteratively add constraints to the linear program, which do not restrict any integer feasible points, but they do sharpen the bounds of the solution space. This is repeated until an integer solution is reached. The first proposed method to generate such constraints are Gomory Cuts (Gomory, 1958, 1963).

If a linear program with integer variables is solved, the final tableau will contain the equations

$$x_i + \sum a_{ij}s_j = b_i \ , \tag{1.10}$$

which include the non-basic slack variables s_j, as introduced above.

Gomory proposed to construct a set of additional constraints using the fractional parts of the coefficients a_{ij} and b_i.

$$x_i + \sum \lfloor a_{ij} \rfloor + (a_{ij} - \lfloor a_{ij} \rfloor)s_j = \lfloor b_i \rfloor + (b_i - \lfloor b_i \rfloor) \tag{1.11}$$

can be rearranged to this equation

$$x_i + \sum \lfloor a_{ij} \rfloor s_j - \lfloor b_i \rfloor = +(b_i - \lfloor b_i \rfloor) - \sum (a_{ij} - \lfloor a_{ij} \rfloor)s_j \tag{1.12}$$

with all integer parts on the left, and all fractional parts on the right side. The right-hand side

$$(b_i - \lfloor b_i \rfloor) - \sum (a_{ij} - \lfloor a_{ij} \rfloor)s_j \tag{1.13}$$

is less than 1 for all integer points of the solution. Because the left-hand side of (1.12) is the integer, the inequality

$$(b_i - \lfloor b_i \rfloor) - \sum (a_{ij} - \lfloor a_{ij} \rfloor)s_j \leq 0 \tag{1.14}$$

holds for all integer points in the feasible region. By introducing this new inequality, no feasible integer point is excluded. In addition, the new constraint excludes the former solution from the feasible region. As it is numerically instable to add lots of these cuts, this technique is mostly used in a special type of branch-and-bound algorithm: Branch-and-cut.

The branch-and-cut method starts from a relaxed version of the problem. If a solution is found and it contains fractional values, which should be integers, cutting planes are added. Then, the problem is branched into two subproblems, on a fractional variable x_i. One subproblem restricts $x_i \leq \lfloor x_i' \rfloor$ and the other $x_i \geq \lceil x_i' \rceil$. This is recursively repeated on the subproblems. Recursion is stopped if the solutions to the subproblems are infeasible or if the objective value is worse than an objective value of a previously observed integer solution. Branch-and-cut methods have been proven to be very efficient and are implemented in most present-day state-of-the-art solvers.

1.5.2 Greedy Algorithms

Greedy algorithms are heuristic methods that solve optimization problems by iteratively constructing a solution. In each iteration, the solution is extended by the best local choice. However, most greedy approaches fail to find the best solution (e.g. knapsack problem, Travelling Salesman Problem (TSP[1]), etc.). The strategy is short-sighted and uses only fractions of the available data to make its decision. That being said, in many cases, greedy methods find fairly useful solutions very quickly and are therefore often used in time-critical situations, such as network routing.

Well-known examples are Dijkstra's algorithm for finding shortest paths and Kruskal's algorithm for the MSP on weighted graphs.

The class of problems for which greedy algorithms exist is best characterized by matroid theory. A matroid is a mathematical structure defined as a set of sets with the following properties.

- Property 1: The empty set is always in the matroid.
- Property 2: If a set is an element of the matroid, all subsets of the set also are.
- Property 3: If sets A and B are in the matroid and A has more elements than B, for each element x in A but not B, there is also a set consisting of all elements in B and x.

The union of all sets in the matroid is called the ground set. A basis is a set of maximum size, and is itself not a subset of any other element of the matroid.

The set of all feasible solutions and sub-solutions form a matroid. It can be shown that the greedy algorithm always returns an optimal solution for this kind of problem in

[1] Travelling Salesman Problem: The optimization problem to find the shortest round-trip visiting all cities in a list.

a matroid (maximum-weight basis). This proof has been extended to the more general class of greedoids, which do not satisfy matroid property 2. This allow feasible solutions to contain non-feasible subsets. Finally *Helman et al.* the *matroid embedding*, a structure which encompasses all problems solved by the greedy algorithm to optimality (Helman *et al.*, 1993).

However, many problems cannot be optimally solved by greedy algorithms, because good solutions to local subproblems are often not optimal on the whole. The degree to which this is true for specific problems determines how well the greedy approach is suited as a heuristic. For many well-known optimization problems, approximation bounds are known, which describe the effectiveness of greedy algorithms on them. For the set covering problem, which will be introduced later in Chapter 4, the approximation bound is

$$H(n) = \sum_{k=1}^{n} \frac{1}{k} \leq \ln n + 1 \ , \tag{1.15}$$

where n is the size of the largest set (Lund *et al.*, 2005). This is the best runtime / approximation ratio for the set cover problem (Feige, 1998) in polynomial runtime. This makes the greedy procedure a possible choice for an approximation in this case.

Large instances of the set covering have to be solved when selecting TXP antibodies for proteome-wide coverage (Planatscher *et al.*, 2010). Detailed comparisons and numerical experiments using the greedy algorithm with integer programming formulations can be found in Chapter 4.

1.5.3 Local Search Hybrid Approaches

Local search algorithms try to improve an existing feasible solution by changing one or more elements of the solution. Given a valid, but not optimal tour for a TSP, a local search could always switch two edges. If a shorter tour is found, then the solution is kept. This method, called 2-opt, is iterated until no further improvement is achieved. The Lin-Kernighan heuristic is a generalization of this procedure and is still considered to be one of the most effective ways to produce near-optimal results for the TSP (Lin and Kernighan, 1973).

It is in the nature of local search to get stuck in local optima, potentially leading to very low-quality solutions. Meta-heuristic approaches, such as memetic algorithms, iterated local search, variable neighborhood search, GRASP (Feo and Resende, 1995), and simulated annealing, combine global with local optimization algorithms to overcome this problem. Each heuristic differs at which stage a local search is applied, if a problem subset is optimized, and when the local search is stopped. While ILS simply initiates multiple searches from different starting points, methods like simulated annealing alternate to improve and distort the solution to escape local optima. The performance of these approaches is limited by the construction of a smart neighborhood function that

leads the search to good local optima. This is the case if the neighborhood function has a consistent casual relationship with fitness improvement, such that the well-defined local optima are described (Aarts and Lenstra, 2003). Other swarm-based heuristics have been proven to be quite effective for some combinatorial optimization problems, e.g. ant colony optimization for TSP (Dorigo and Gambardella, 1997).

In Chapter 5 a metaheuristic combining a greedy algorithm with an integer programming based local search is used to solve very large instances of a linear Boolean optimization with quadratic constraints (Planatscher *et al.*, 2013a).

Chapter 2

The mathematical programming framework SCPSolver

The SCPSolver library is a Java framework for solving various optimization problems with a focus on linear and mixed-integer programming which has been developed by the author. Although the library does not implement its own linear programming solver, it offers a common interface for existing solvers, thus alleviating the complexity of an individual solver implementation for the authors of linear optimization problems. This makes it very easy to try different solvers on the same problem or to update solver compatibility with version changes.

On the other hand, a common interface for linear program solvers makes it easy for solver programmers to provide the users with new implementations. All that is needed is one wrapper class that translates between the interface requirements and the underlying implementation structure.

The SCPSolver library is capable of automatically detecting available solvers in the classpath and providing the user dynamically with requested solvers. Solvers thus can be distributed in individual modules called solverpacks. If a developer wants to try another solver, the required steps are reduced to downloading the solverpack and copying it into the classpath. Solver developers can distribute solvers in a single module containing all the necessary class files and libraries.

2.1 Framework design goals and decisions

Many solvers already support Java by providing their own Java interface. However, most of these APIs are quite cumbersome to set up for the developer. Precompiled binaries, if available, have to be placed in a given directory, other libraries have to already be installed, and so on. Even worse, this is also required when the application is delivered to the user or customer. We have identified the following shortcomings of existing Java APIs for linear programming: problematic setup, APIs are not object-oriented or "Java-like", and missing platform- independence. Not all the shortcomings are common to all solvers. For example, CPLEX offers quite a good API and its deployment is fairly easy. However, SCPSolver is not an approach that yields a unified Java Middleware for linear

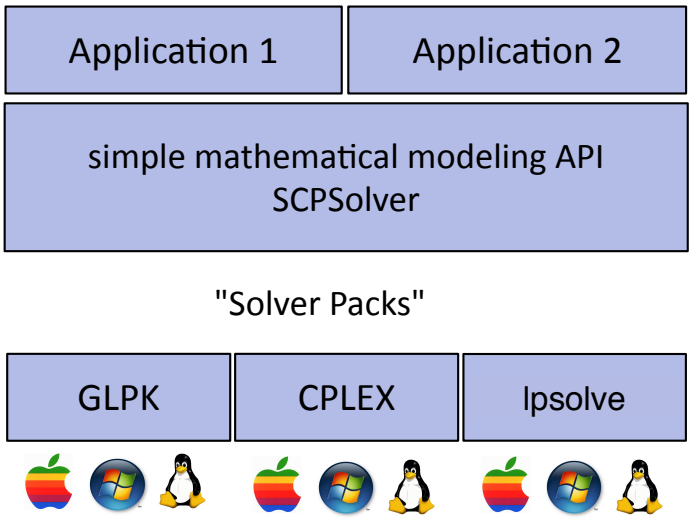


Figure 2.1: SCPSolver layer schema

programming. The software Java ILP (http://javailp.sourceforge.net/) has similar aims but does not tackle the deployment problems.

Consequently the design goals of SCPSolver are to provide a familiar API to a Java developer, the automatic deployment of binaries on multiple platforms, the ability to define large models, separation of the model from the solvers and to provide access to multiple solvers by using a plugin concept.

2.1.1 Ease of use

Usability was the main design goal during the development of SCPSolver. Many commonly used solver interfaces are not object-oriented since they just expose native C functions through the Java Native Interface. This makes the learning curve rather steep for a Java developer who is used to a certain level of abstraction. Despite the fact that SCPSolver uses object orientation and defines several interfaces, the API has a very moderate level of complexity. A developer with basic knowledge of mathematical optimization will find terms directly in the class name (e.g., Solver, LinearProgram, LinearBiggerThanEqualsConstraint) and methods where they are expected (for example the addConstraint-method in the LinearProgram class).

The LinearProgram class is a basic description for a linear program. In ordinary production usage, this class usually gets instanced first. Then all constraints and boundaries are added and finally, a solver is instanced to solve the linear program.

2.1.2 Platform independence

Since most available solvers are implemented in C/C++ or Fortran, the compiled binaries are specific to a platform (e.g., Windows, Linux, etc.) and an architecture (e.g., x86). Java, however, is platform-independent, and Java Virtual Machines are available on all major operating systems. The Java Native Interface already supports a certain level of platform independence. The Java classes that call native code do not have to be recompiled as the platform changes. For instance, a Windows Dynamic Link Library (DLL) and a Linux shared library can be distributed with the Java class, and the Java Virtual Machine (JVM) will take care of loading the correct binary.

In order to bring the efficiency of existing solvers and the platform independence of Java together, the concept of solverpacks was developed. A solverpack is a Jar-File consisting of all Java class files needed by the solver and the native libraries for multiple platforms.

The factory class SolverFactory uses the Service Provider Interface to find and load available solver classes. Additionally, this class ensures that native libraries are available on runtime and extracts the libraries from the solverpack if necessary.

2.1.3 Multi-solver platform

All linear program solvers in the SCPSolver library share a common interface: LinearProgramSolver. After creating a new linear optimization problem, a user typically requests a new solver from the solver factory and uses this solver to compute the solution. So a common call would look as follows:

```
LinearProgram lp; ...
LinearProgramSolver solver = SolverFactory.newDefault();
solver.solve(lp);
```

An alternative method could be that the user specifically asks for a certain solver:

```
LinearProgramSolver solver = SolverFactory.getSolver('GLPK');
```

Solverpacks for the following solver are available, including binaries for Windows, Linux, and Mac Os X (see the compatibility matrix in table 2.1).

GLPK The GNU Linear Programming toolkit is a solver developed under the GNU Public License. The solver implements the revised simplex and the primal-dual interior point method for non-integer problems. Gomory's integer cuts are used for the solution of integer and mixed-integer problems.

CPLEX CPLEX is a commercial high-performance solver originally developed by Robert E. Bixby. Since 2009, the software is owned and distributed by IBM. It is considered to be one of the fastest solvers on the market (http://plato.asu.edu/bench.html).

Table 2.1: Solverpack compatibility matrix for x86 architectures. While the solver libraries should also work on other distributions/editions, the matrix contains only the tested versions.

Platform/Solver	GLPK	LPSOLVE	CPLEX
Windows XP, 7	32-bit	32-bit	32-/64-bit
Mac Os X 10.6	64-bit	64-bit	64-bit
Ubuntu 10.4 Linux, Scientific Linux 5	32-/64-bit	32-/64-bit	32-/64-bit

lpsolve lpsolve is an open-source solver originally developed at the Eindhoven University of Technology. This solver is based on the revised simplex method for linear programs and branch-and-bound for mixed-integer linear programs.

The multi-solver platforms enable an open-source developer to use commercial solvers, like CPLEX, keeping the possibility to publish the source of the model under a free license viable. Users who do not have a license can still solve a given model using an open-source solver.

2.2 Problem modeling

SCPSolver supports problem modeling at two levels. On a lower level of abstraction, the user employs common double arrays or a sparse vector data type in order to define a model. It is also possible to use a higher level of abstraction that allows easier definition of optimization problems.

2.2.1 Basic interface

If a developer wants to use mathematical programming in an existing project, it is important to make the model definition interface as easy as possible. If the programmer has a basic understanding of linear optimization and Java, the API should reflect the construction of a linear program in an object-oriented manner using familiar terminology. In the low-level SCPSolver-API, the developer has to understand three basic classes and interfaces: LinearProgram, LinearConstraint, and LinearProgramSolver.

The LinearProgram object is the representation of a linear program. Consider the following example:

$$min 5.0x_1 + 10x_2 \tag{2.1}$$

under constraints

$$3.0x_1 + 1.0x_2 \geq 8.0 \tag{2.2}$$
$$4.0x_2 \geq 4.0 \tag{2.3}$$
$$2.0x_1 \geq 2.0 \tag{2.4}$$

In the SCPSolver modeling API, this could be expressed as:

```
LinearProgram lp = new LinearProgram(new double[]{5.0,10.0});
lp.setMinProblem(true);
lp.addConstraint(
   new LinearBiggerThanEqualsConstraint(new double[]{3.0,1.0}, 8.0, "c1")
);
lp.addConstraint(
   new LinearBiggerThanEqualsConstraint(new double[]{0.0,4.0}, 4.0, "c2")
);
lp.addConstraint(
   new LinearSmallerThanEqualsConstraint(new double[]{2.0,0.0}, 2.0, "c3")
);
```

Note that there are three different classes to express the constraints. This piece of code is entirely transparent to a Java developer.

If large numbers of variables are needed and the constraints are sparse, meaning that the majority of coefficients are equal to zero, a sparse matrix type can also be applied to make better use of the available memory. The LinearProgram object holds methods to set the data type of a variable to integer or Boolean, and can also set its optimization direction. In addition, it is possible to export a model to the CPLEX format, which is a de facto standard and can be read by most command-line solvers.

2.2.2 High-level interface

Like other modeling toolkits, SCPSolver supports a higher-level representation of mathematical programs. The key object for this modeling interface is the LPWizard.

The example problem defined in (2.1)-(2.4) can be modeled as:

```
LPWizard lpw = new LPWizard();
lpw.plus("x1",5.0).plus("x2",10.0);
lpw.addConstraint("c1",8,"<=").plus("x1",3.0).plus("x2",1.0);
lpw.addConstraint("c2",4,"<=").plus("x2",4.0);
lpw.addConstraint("c3", 2, ">=").plus("x1",2.0);
```

The high-level interface balances readability and rapid modeling. It allows the incremental definition of a model. New variables can be added in each new term and each variable has a unique identifier ('variable name'), which can be used in the modeling and results analysis.

In mixed-integer programs it often occurs that all the variables within specific constraints have to be declared as integer or binary. In the high level this is relatively straight forward:

```
lpw.addConstraint("c2",4,"<=").plus("x2",4.0).setAllVariablesInteger();
lpw.addConstraint("c3", 2, ">=").plus("x1",2.0).setAllVariablesBoolean();
```

By exposing those methods directly on the objects defining a term, they are immediately available through syntax completion in Java Integrated Development Environments (e.g., Eclipse, Netbeans, etc.). All information about specific constraints, which could be the integration of an atomic principle or a rule in a mathematical model, is kept in one place in the code.

2.2.3 Debugging of linear programs

A common problem when using linear programming is the process of finding errors in complex models that lead to infeasibility. If a linear program, which should have a solution, turns out to be infeasible, one or more constraints may be too restrictive or even flawed. When this happens the common methodology is to relax the program by deactivating the constraints. For the Java programmer this would mean commenting out code that defines constraints in order to isolate the error source. This process is repeated until the problem can be solved and the method is laborious if the model consists of many constraints. The code has to be recompiled, calculations have to be started, and the results must be evaluated. If a set of constraints leads to infeasibility, manual debugging becomes impractical.

Consider a linear program $min\ cx^t, \sum Ax < b$ with a $m \times n$ constraint matrix A and a m-vector b. Let C denote the set of resulting constraints $\{ax \leq b\}$. If there exists an n-vector x satisfying all inequalities in C, the system is feasible. If no such vector exists, the system is infeasible. If there is a subset of constraints where $S \subset C$ is feasible, it is called a feasible subsystem. If the expansion of a feasible subsystem $S' = S \bigcup A_i x \leq b \in C$ by only one constraint makes it infeasible, S is called a maximal feasible subsystem (MFS). A maximum cardinality feasible subsystem is defined as:

$$S_{MCF} = argmax_{S \subseteq C}\{|S| \mid S \text{ is feasible}\} \tag{2.5}$$

The identification of S_{MCF} is itself an NP-hard optimization problem.

In mathematical programming, several techniques are known that can be used to detect sets of infeasible constraints. A method called elastic programming adds slack variables to all the constraints of an unsolvable system. The system is then solved by minimizing the sum of all the slack variables.

$$min \sum_{j=1} s_j \tag{2.6}$$

$$\sum a_{ij} x_i + s_j \leq b_j \ \ \forall 1 \leq j \geq m \tag{2.7}$$

$$\sum s_j \geq 0 \ \ \forall 1 \leq j \geq m \tag{2.8}$$

This minimal solution to this system can be interpreted as the smallest change needed to be made to the boundaries in order make the constraint system feasible by only allowing subtraction from the constraint vector b. Without a lower boundary, the problem itself would be infeasible. The system

$$min \sum_{j=1} s_j^+ + s_j^- \tag{2.9}$$

$$\sum a_{ij}x_i + s_j^+ - s_j^- \leq b_j \ \forall 1 \leq j \geq m \tag{2.10}$$

$$\sum s+_j, s-_j \geq 0 \forall 1 \ \ \leq j \geq m \tag{2.11}$$

allows the adaptation of b in the negative and positive directions by using a trick that is applied if absolute values are present in a linear program. A positive slack variable s_i^+ and a negative slack variable s_i^- are added to each constraint. While this could lead to a smaller sum of modifications to the b-vector, it would be more useful for a developer who is seeking the smallest number of constraints to be modified.

This can be achieved through the following extension:

$$min \sum_{j=1} r_j \tag{2.12}$$

$$\sum a_{ij}x_i + s_j \leq b_j \ \forall \ 1 \leq j \leq m \tag{2.13}$$

$$\sum a_{ij}x_i - s_j \leq b_j \ \forall \ 1 \leq j \leq m \tag{2.14}$$

$$s_j + r_j min_j \geq 0 \ \forall \ 1 \leq j \leq m \tag{2.15}$$

$$s_j - r_j max_j \leq 0 \ \forall \ 1 \leq j \leq m \tag{2.16}$$

$$r_j \in \{0,1\} \ \forall \ 1 \leq j \leq m \tag{2.17}$$

Here an additional binary variable r_j is introduced for every constraint. min_j and max_j represent the minimum and maximum change to be applied to b_j, respectively. The complete deactivation of a constraint would be equivalent to setting $b_j = \pm\infty$. Because infinite values are not usable in a linear program, and in order to keep some control over the constraint violations, finite boundaries were introduced. The constraints (2.15) and (2.16) are satisfied only in the cases where: $s_j = 0$, meaning constraint j remains unchanged, or $min_j \leq s_j \leq max_j$ and $r_j = 1$, meaning constraint j is changed.

r_j indicates whether a constraint has been relaxed or not. The objective is set to minimize the sum of r_j, or phrased in another way, to minimize the number of elastic constraints in the constraint set, in order to achieve feasibility.

This method is implemented in SCPSolver using a simple graphical user interface called the LPDebugger. By inserting

```
LPDebugger lpd = new LPDebugger(lp),
```

the LPDebugger graphical user interface will open when the statement is reached during execution. The LPDebugger graphical user interface consists of a table listing all the constraints in a status window. The developer can select which constraints should be deactivated and then try to solve the linear program by clicking “solve”. A trial-and-error protocol is recorded in the lower text box, logging which constraints have been deactivated, the feasibility of the reduced program, and if applicable, the objective value.

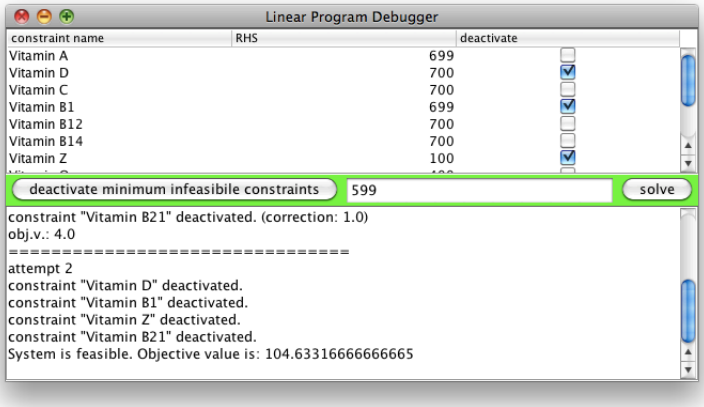


Figure 2.2: LPDebugger screenshot. The developer can use the “deactivate ”minimum infeasible constraints feature in order to find the maximal feasible subsystem. In this case, the alternative formulation 2.12 - 2.17 is generated and solved. The bound is set in the text field next to the button.

2.3 Conclusion

It this thesis alone SCPSolver has been used to create and solve optimization models for at least six different applications in assay development. It has been made available to the public 2013 on scpsolver.org. SCPSolver has been downloaded 2000 times since then. Users from Germany, UK, USA, China, India and other countries have been in contact with the author. Recently it has been used in a study by Yahoo Research to determine tournament payout structures for daily fantasy sports (Musco *et al.*, 2016).

It appears that the software has a small but solid user base, and it fulfills its initial purpose: giving Java programmers an easy to use integer programming library with some platform and solver independence.

Chapter 3

Applications of combinatorial optimization for immunoassays

During the development of biochemical assays several steps and processes need to be planned and later implemented in the lab. This chapter focuses particularly on multiplex serological assays for the simultaneous measurement of antibody concentrations in serum. We identified two problems that can be addressed using combinatorial optimization. Both problems, the systematic reference sample generation and the selection of samples for assay validation, originate from the lack of a synthetic reference sample in the realm of serological assays. In the third section both approaches are combined to find sample pools for validation experiments. The fourth problem deals with the ideal placement of a sample on a planar assay chip, with the focus being to reduce bias. All four problems have been solved using the SCPSolver library described in the previous chapter. Please note: *If not explicitly stated otherwise, the variables described in this chapter assume binary values.*

The first section following this introduction is based on the article 'Systematic reference sample generation for multiplexed serological assays' (Planatscher *et al.*, 2013b), which was written by the author of this dissertation and his colleagues. Dr. Stefanie Rimmele conducted the lab experiments and contributed equally to this work.

Multiplexed serological assays have been applied to characterize the response against a variety of pathogens including *Hepatitis B virus*, *Hepatitis C virus*, *Helicobacter pylori*, *Mycobacterium tuberculosis* (Mtb) and *influenza* (Boni *et al.*, 2013; Opalka *et al.*, 2010; Tong *et al.*, 2005; Waterboer *et al.*, 2005). In addition, proteome arrays have enabled the definition of reactive antigen sets for a variety of pathogens like Mtb (Kunnath-Velayudhan *et al.*, 2010), *Plasmodium falciparum* (Doolan *et al.*, 2008), *Human papillomavirus* (Luevano *et al.*, 2010), *Burkholderia pseudomallei* (Felgner *et al.*, 2009), and *Coxiella burnetii* (Beare *et al.*, 2008; Vigil *et al.*, 2010, 2011). Arrays containing thousands of recombinant human proteins were used in the discovery of antibodies directed against self-antigens (Gnjatic *et al.*, 2009; Hudson *et al.*, 2007; Vizoso Pinto *et al.*, 2010). Serological assays are well established within the field of autoimmune diseases (Auger *et al.*, 2009; Kattah *et al.*, 2006; Robinson *et al.*, 2002). All of these serological assays require quality-controlled sample testing procedures.

3.1 Systematic reference sample generation for multiplexed serological assays

Prior to implementation into diagnostics, appropriate assay validation has to be achieved. FDA guidelines for the development of immunoassays state that sufficient quality control samples should be used to ensure control of the assay (FDA, 2001). As a consequence, such quality-controlled samples should be available for assay validation as well as for large-scale screening and diagnostic purposes (Cummings *et al.*, 2008). Quality control samples are necessary within every assay to ensure that it performs within specifications and the samples should be reviewed before interpretation of the results of individual serum samples. The purpose of a quality control is to report that all experimental steps were executed correctly in an assay experiment and to be able to compare data over a longer period.

Reference samples for sandwich immunoassays targeting serum proteins can be easily generated by spiking the target analytes into a plasma or serum matrix. However, any serological assay is based on the presence of human antibodies specific for the selected antigens. For singleplex assays it is usually sufficient to select a serum with strong reactivity towards its respective antigen. However, identifying single sera with appropriate reactivity against a multitude of different antigens, as is required for antigen arrays, has been very difficult if not impossible in many cases. Moreover, using a single serum as a quality control to cover all targeted antigens would mean that it would only be available in limited amounts and may thus confine test development, validation, and clinical evaluation. A common, but surprisingly little-documented approach is to create pools from multiple sera in order to warrant reactivity towards all target antigens and generate a sufficiently large quality control stock (Cooley *et al.* (2008); Wong *et al.* (2004)). Here we present a mathematical approach towards a sample pooling strategy, where the composition of such a pool was calculated from an available data set with the aim that this pooled sample shows a positive response for each analyte. The threshold for a positive signal is defined by a multiple of the negative control population. In our case, we chose four times the negative control population. If a signal in the pool exceeds this threshold, the analyte is covered.

The serological response of 142 serum samples obtained from patients with active tuberculosis (TB) was analyzed using a bead array consisting of 71 TB proteins. The serological response of these sera was heterogeneous, ranging from 2-69 TB-associated proteins per serum sample. Out of the 142 sera, we found no serum reactive to all 71 antigens under investigation. Our mathematical approach identified sets of positive sera, which could be pooled to generate a quality control serum to react with all 71 TB proteins. This strategy allowed us to create defined reference samples, revealing a simultaneous serological response against all TB antigens employed in the assay. Appropriate data sets of the serological response pattern against the targeted antigens for a set of available samples provided the basis for our calculation. A mathematical model was de-

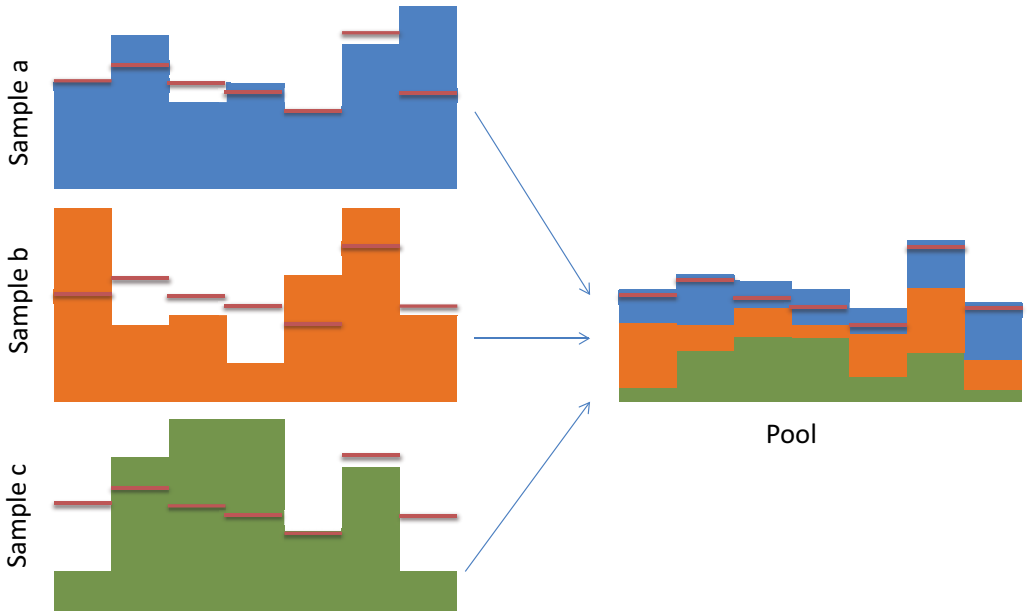


Figure 3.1: The red lines represent an antigen-specific threshold. Samples A-C show individual responses and none exceed all the thresholds. Through combination, a new pooled sample with a positive response to all antigens is created.

veloped to predict the reactivity characteristics of a given sample pool. We hypothesized that those values could be estimated from the quantitative serological response measured for the individual samples. Our first assumption was that if samples are combined, their assay signals would add up (see Figure 3.1). The second assumption was, that on average assays show a dilution linearity with a slope of 1.0. A linear integer program was constructed from the model, screening data, and the threshold vector. The objective function was to maximize the number of serum reactivities given a fixed number of serum samples from which the pool should be generated. The relative dilution of the sample pool was kept identical to the dilution of the individual serum. Results of the optimization approach revealed suggestions for the generation of optimal sample pools, differing in composition and size.

The screening of a set of samples $S_1, S_2, \ldots, S_n$ generated a data set M and $m_j(S_i)$ designates the MFI[1] -signal for target j in sample S_i . An important premise is that the assays have a predominantly linear characteristic within the range of interest. If a pooled sample P was created from samples S_k and S_l, the MFI would approximately add up

$$m_j(P) \approx m_j(S_k) + m_j(S_l) \tag{3.1}$$

[1]Medium Fluorescence Intensity

If a sample is diluted using factor $\alpha \leq 1$, the MFI will exhibit a linear change:

$$\alpha m_j(P) \approx m_j(\alpha\text{-}dilutedP) \tag{3.2}$$

We have created pools by subsequently adding samples $(S_1, S_1+S_2, S_1+S_2+S_3, \ldots)$. The threshold for a positive signal is usually defined by a multiple of the mean intensities measured in a negative control population. The threshold for target j is designated t_j. Furthermore, the decision variable x_i describes whether the sample S_i is included in the pool or not. In algorithm 1 the number of samples is fixed to a maximum, and the number of analytes covered by the resulting positive control PC pool is the target to be optimized. By allowing integer values for x_i, the constraint

$$\sum_{i=0}^{n} x_i = X_{max} \tag{3.3}$$

fixes the number of parts a pool consists of to X_{max}. E.g., a pool with $X_{max} = 5$ could consist of three parts S_3 and two parts S_7. The set of decision variables $a_1, a_2, \ldots, a_m$ indicate whether an analyte should be covered by the pool or not. By defining the coverage as

$$\sum_{i=0}^{n} x_i m_j(S_i) \geq a_j t_j,\ 1 \leq j \leq m \tag{3.4}$$

it is ensured that only if $a_j = 1$ will the sum of MFIs have to exceed the threshold t_j. The term to maximize in this case is the number of covered analytes

$$max \sum_{j=0}^{m} a_j\ . \tag{3.5}$$

Another constraint is that the resulting pool should have the same matrix dilution as was used in the normal sample preparation. If the input samples have been measured in a 1:n dilution resulting in values $m_j(S_l)$, the values need to be scaled accordingly to *1:n* X_{max}. The upper bound for X_{max} is defined by the limit of dilutional linearity. E.g., if the limit is 1:2000 and the original dilution was 1:200, the maximum for X_{max} would be 10.

For the verification of our theoretical results, the following experiments were performed. In the first experiment the assumption of the additivity of the individual signal values of the pooled samples was tested. A suspension bead array displaying the different tuberculosis antigens was incubated with human serum samples.

Bound human immunoglobuline G (IgG) antibodies were detected with an R-PE-labeled anti-human IgG. The read-out was performed on a fluorescence-based bead array reader (Luminex FlexMAP3D). Sample pools were created by subsequently pooling samples in the scheme S1, S1 + S2, S1 + S2 + S3, up to a pool consisting of six samples.

As shown in Figure 3.2 A-E, a strong correlation ($R \geq 0.957$) was observed between the values predicted from single sample screenings and the signal generated by the sample pool. The slope of the linear regression was 1.03 for the least complex pool and

Algorithm 1: Sample pool optimization

Input: $M, X_{maxupperbound}$
Output: list of protein pool recipes
while $X_max \leq X_{maxupperbound}$ **do**

$$\hat{m_j}(S_l) = \frac{m_j(S_l)}{X_{max}}$$

solve ILP:

$$\max \sum_{j=0}^{m} a_j$$

subject to

$$\sum_{i=0}^{n} (x_i \hat{m}_j(S_i) \geq a_j t_j | 1 \leq j \leq m)$$

$$\sum_{i=0}^{n} x_i = X_{max}$$

$$a_j \in \{0,1\} \forall j \in \{1,,m\}$$

$$x_i \in N_0$$

$$X_{max} = X_{max} + 1$$

end
return L

decreased to 0.7 for the pool containing up to six samples. This data supports our hypotheses about signal additivity. The observation that the signals generated by the pools for a given antigen get stronger when the number of sera in the pool increased is notable (Figure 3.2). A larger number of different paratopes for the same antigen originating from individual sera could also explain this observation.

In a subsequent experiment, our algorithm was applied to find optimal pools for the total panel of 71 TB antigens. Here a data set derived from 142 previously tested serum samples was used as the input. The allowed range for dilution of a single sample within the pool was 1:200 to 1:2000. Interestingly, we found that we had to consider the possibility that the signal intensities of each individual serum added to the pool are "' diluted"' with the other sera during the pooling process (Figure 3.2). The final serum dilution of the pool was set to 1:200, according to the standard dilution of our serological TB assay. An artificial cutoff for each TB antigen was calculated from the quadruple of the values measured in the negative control sample. The algorithm suggested ten solutions consisting of up to ten parts of up to four different samples. Thus, as expected, it was not possible to cover all analytes by using a single sample or by pooling two individual samples. Our algorithm suggested that pools consisting of at least three samples would reveal a serological response to all TB antigens. The signals of the pool for all analytes were higher than the defined threshold. The measured values correlate with the predicted values with a correlation coefficient of 0.98 (see Figure 3.3 A). The correlation between the pool and the three single samples is comparatively low, as shown in Figures 3.3 B-D.

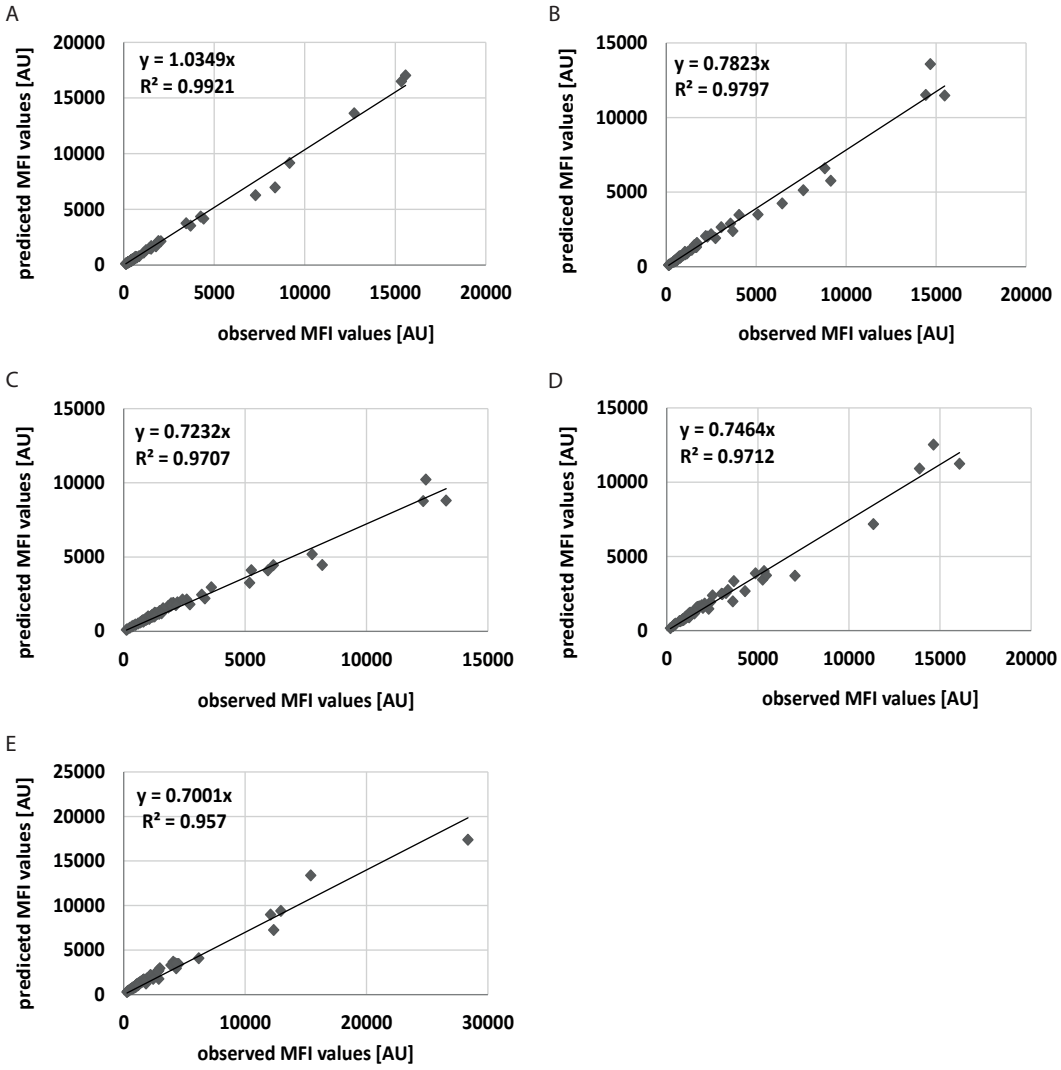


Figure 3.2: The graphs show the strong correlation between the predicted and measured results, although the linear slope is 0.7 for the most complex pool (E). (A) Correlation of prediction and measurement for pool S1 + S2 (B) pool S1 + S2 + S3 (C) pool S1 + S2 + S3 + S4 (D) pool S1 + S2 + S3 + S4 + S5 (E) pool S1 + S2 + S3 + S4 + S5 + S6.

This shows that no sample stands out in the pool and that the signal pattern is the result of the composition of all three samples.

We have created a technical quality control for multiplexed antigen assays to make sure that all antigens used in the assay have not lost their antigenicity and that all technical steps are executed correctly. With this mathematical model, we can create quality control samples for roughly 60,000 samples from only 1.5 mL of three pooled serum samples.

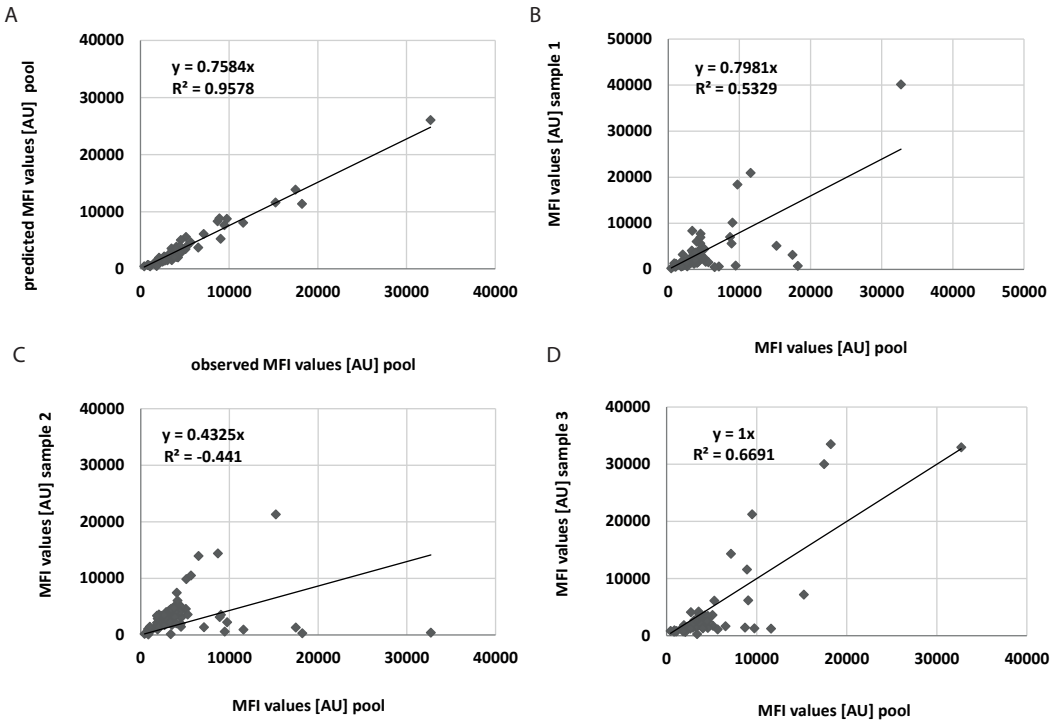


Figure 3.3: While (A) the correlation between the sample pool and the predicted values (a weighted sum of single sample values) is high, the single samples (BD) show weaker correlation with the pool. This data shows that the unique coverage characteristic of the serum pool is due to the combination of the three samples.

We also created a second pool consisting of four samples (10 parts; 5 parts sample 1, 3 parts sample 2, 1 part sample 3 and 4) with a correlation coefficient of 0.9 between the predicted and observed MFI values (data not shown). Once the first pool is running out, one can easily create a second pool consisting of different samples. Our results demonstrate that our mathematical model for sample pools makes adequate predictions. We demonstrated that quality controls for multiplex antigen assays can be created through the systematic selection and pooling of samples. Our systematic approach is scalable and can be easily adapted to other assay platforms. We believe that our method provides an important tool for diagnostic assay development and test evaluation.

3.2 Selection of samples for validation experiments

For serological assay validation it must be shown that the low, medium, and high levels can be reliably measured. Reference samples cannot be easily obtained in the realm of serological assays, because the target analyte cannot be synthesized. The only correct

way to validate such an assay is to measure a patient sample with low, a second sample with medium, and yet another one with a high level of the antibody. Of course each data point needs to be taken in multiple replicates on different days, and even with a varying number of freeze-thaw cycles in order to prove assay stability and real-world conditions. For multiplex serological assays, a whole assay panel needs to be validated. Every

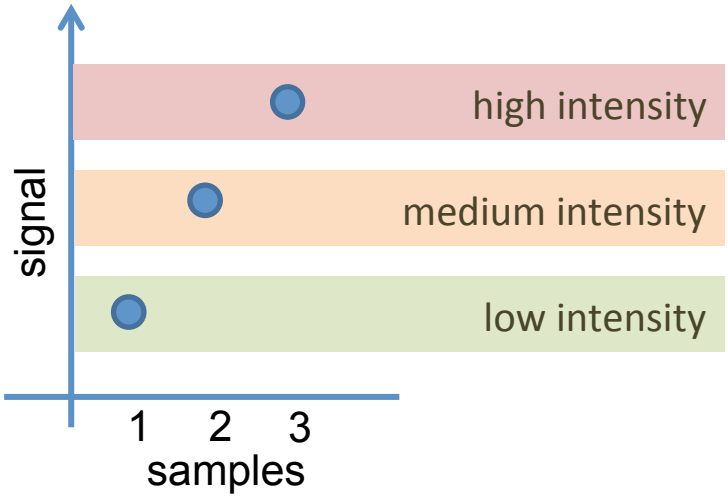


Figure 3.4: Validation requires proof that low, medium, and high levels can be reliably measured

biomarker needs to be validated on its own, so the validation step would include running the multiplex assay on three suitable samples for each analyte in the panel. Considering the replicate, interday, and stability check as multiplicators, the number of required experiments can quickly exceed a manageable size. If it is possible to choose a few samples such that the combination of their profiles would cover low, medium, and high for most analytes, validation cost and effort can be reduced.

The method described in this section has been presented as a poster at HUPO 2014 Madrid[2].

From the perspective of optimization, this is a set-covering problem. Select a **minimum** number of sera such that for each analyte a low, medium, and high resolution data point can be measured.

$$min \sum s_i \tag{3.6}$$

subject to

$$\sum s_i a_{ijk} \geq 1 \tag{3.7}$$

[2]Optimal selection of samples for multiplex serological assay validation. Authors: Hannes Planatscher, Angela Filomena, Oliver Poetz, Thomas O. Joos, Nicole Schneiderhan-Marra

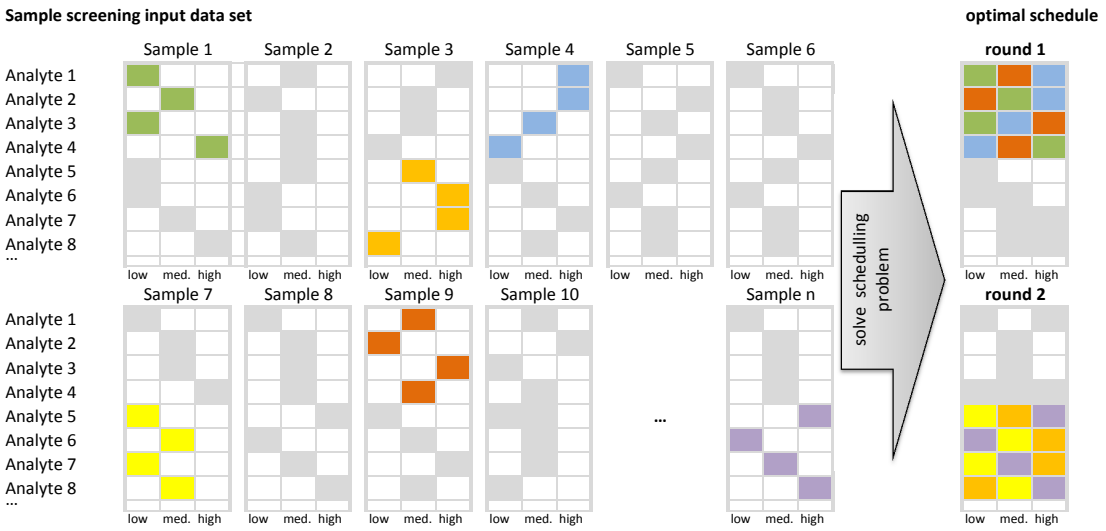


Figure 3.5: Selection of samples to cover a maximum of different ranges in validation experiments

with

$$a_{ijk} = \begin{cases} 1 & lb_{jk} \leq S_{ij} \leq hb_{jk} \\ 0 & otherwise \end{cases} \tag{3.8}$$

If there is a budget constraint, the problem can be stated as: “Choosing a fixed number of samples n, **maximize** the number of low, medium, and high level ranges for different analytes selected for validation. ”By introducing new decision variables c_{jk}, which indicate whether an analyte range is selected ($c_{jk} = 1$) or not ($c_{jk} = 0$), we can write the objective function as follows:

$$max \sum_j \sum_k c_{jk} \tag{3.9}$$

subject to the constraint that if range k of analyte j is subject to validation ($c_{jk} = 1$), at least one suitable sample s_i should be selected:

$$\sum_i s_i a_{ijk} \geq c_{jk} \tag{3.10}$$

and the budget constraint to limit the number of samples to n is

$$\sum_i s_i \leq n \tag{3.11}$$

This formulation, however, leads to solutions that maximize the number of ranges but not the number of fully validated analytes (the low, medium, and high ranges covered). This is achieved through the following formulation. Maximize the number of covered targets

$$max \sum_j t_j \tag{3.12}$$

subject to the constraint to validate all l ranges for each selected target ($t_j = 1$)

$$\sum_k c_{jk} = l t_j \tag{3.13}$$

and subject to the constraints (3.10) and (3.11).

This method can be further refined by allowing the scheduling of multiple validation rounds, as illustrated in figure 3.5. In each round, a maximum of n samples can be screened. Each round validates all ranges for a number of analytes. It does not count if the midrange is validated in one round and the high range in another. The total number of analytes treated over all rounds is maximized.

Maximize the number of targets:

$$max \sum_j t_j \tag{3.14}$$

If a target analyte is selected ($t_j = 1$), it must be scheduled for validation in at least one round

$$\sum_j r_{jl} \geq t_j \tag{3.15}$$

In each round a maximum of s_{max} samples is analyzed:

$$\sum_i s_{ij} \leq s_{max} \tag{3.16}$$

If an analyte is selected to be validated in a specific round ($r_{ij} = 1$), all ranges of this analyte need to be covered:

$$\sum_k c_{ikl} = l r_{il} \tag{3.17}$$

If a range k of a specific analyte j needs to be covered in round l, at least one suitable sample i needs to be included in this screening round:

$$\sum_i s_{ij} a_{ijk} \geq c_{jkl} \tag{3.18}$$

3.3 Generation and selection of sample pools for validation experiments

In this approach the methods from 3.1 and 3.2 are combined to validate a multiplexed serological assay using a number of pooled samples.

First a side note on how pooled samples should be made, depending on the type of serotest: IgG or IgA. These are different types of antibodies which can be measured in serological assays. For type G immunoglobulines (IgG) the concentration of the pooled sample must remain constant, meaning that pooled sample must be diluted in exactly the same way as the single sample in the preceding experiments. This is due to the high concentrations of IgG in plasma. These high amounts of free IgG can lead to matrix effects, distorting the signal. In this case the samples signal S_{it} is diluted in the pool, and the signal contributing to the pool must be corrected accordingly: $\hat{S}_{it} = \frac{S_{it}}{q}$.

Other immunoglobuline subclasses as type A (IgA) and dimeric type A (dIgA) are far less abundant in the serum. If the assay panel is aimed at these binders such corrections are not necessary because the matrix effect is deemed negligible, thus $\hat{S}_{it} = S_{it}$.

Sample pooling increases the number of serological patterns exponentially. This makes it much more likely to find a combination of patterns that allows the validation of the observed targets. The number of different serological patterns, given that n samples are available and k different samples are pooled is $\binom{n}{k}$. E.g. 100 samples can be combined to about 75 million different pools of 5, and more than 17 trillion pools of 10 samples. If pools are allowed to include varying numbers of parts (amounts) from different samples the combinatorial richness is even greater, as it these are multisets. A multiset is defined like a set, but is allowed to contain repeated elements (Knuth, 1970), e.g. $\{a,a,a,b,b,c\}$. The multiset coefficient $\left(\binom{n}{k}\right)$ is the number of multisets of cardinality k with elements for a finite set of cardinality n. The notation bears resemblance to the more widely known binomial coefficient, which calculates the number of distinct sub-sets of a fixed size from a larger finite superset.

The multiset number is calculated as follows:

$$\left(\binom{n}{k}\right) = \binom{n+k-1}{k} = \frac{(n+k-1)!}{k!\,(n-1)!} = \frac{n(n+1)(n+2)\cdots(n+k-1)}{k!} \tag{3.19}$$

This allows combining 100 samples to 48 trillion pools consisting of a maximum of 10 parts, almost tripling the pooling potential - but also the search space. The formulation of the linear program is as follows. Maximize the number of target ranges selected for validation ($V_{tr} = 1$):

$$max \sum_{r \in R} \sum_{t \in T} V_{tr} \tag{3.20}$$

with $R = \{low, medium, high\}$ (ranges) and $T = \{t_1, t_2, \ldots, t_m\}$ (targets). If a target-range is selected for validation at least one pool must cover that range ($C_{ptr} = 1$).

$$\sum_{p \in P} C_{ptr} \geq V_{tr} \tag{3.21}$$

To cover a target-range the predicted signal in the pool must be in the defined interval $[l_{tr}, H_{tr}]$.

$$\sum_{i=0}^{n} \hat{S}_{it} a_{pi} + M(1 - C_{ptr}) \geq l_{tr} \tag{3.22}$$

$$\sum_{i=0}^{n} \hat{S}_{it} a_{pi} - M(1 - C_{ptr}) \leq h_{tr} \tag{3.23}$$

A pool contains exactly q parts:

$$\sum_{i=0}^{n} a_{pi} = q \tag{3.24}$$

The variable a_{pi} defines if sample i is part of pool number p. M is a sufficiently large number, which is added to the left hand side if the constraint should be inactive, specifically when the respective level is not selected for validation. This "Big M" trick is widely used to implement such conditional constraints in linear programs. The number of parts q and the number of pools to be composed are parametrizable, and will influence the outcome. The larger these parameters are set the more likely a good solution can be found, but also the more complicated and expensive the validation experiment is.

Unfortunately it turned out that even industrial strength solvers could not find good solutions using this formulation. This is probably due to the enormous search space. E.g. if we search for 3 pools, each composed of 10 parts chosen from 100 samples, the size of the search space is $\binom{\binom{100}{10}}{3}$ which is approximately 1.7×10^{40}. This makes a search space reduction unavoidable. The following properties of the problem could be exploited. Even if in theory the variety of different serological profiles is huge, the number of distinct 'binary' coverage patterns is limited. If the pools of given size could be enumerated efficiently, and the set of candidate pools limited to only distinct coverage patterns, the search is significantly reduced. Another logical property is that coverage patterns with low cardinality (the number of coverage ranges covered by that profile) are not interesting, if the goal is to combine them to achieve maximum coverage. Therefore a minimum cardinality parameter was introduced. This parameter is quite important as it is a way to expand or shrink the search-space easily. Algorithm 2 describes how the enumeration was implemented. It is crucial that the check if a binary profile is already a member of the result set, happens in constant time. This can be achieved using a hash-type data-structure.

Algorithm 2: Recursive multiset enumeration in the pooling algorithm

enum(d,p,d_{max},$mcard$,idx,$prof$,$recipe$,$samples$,$pset$)
$pro_{new} \leftarrow profile + p \times samples[idx]$ $recipe_{new} \leftarrow recipe \cup (p, idx)$ **if** $(d = d_{max})$ **then**
  $binaryprof \leftarrow getBinaryProfile(prof_{new}, bounds)$ **if**
  $(binaryprof \notin pset) \wedge (card(binaryprof) \geq mcard)$ **then**
    $pset \leftarrow pset \cup \{(recipe_{new}, binaryprof)\}$
  end
  else
    for $i \leftarrow idx + 1$ **to** $|samples|$ **do**
      for $p' \leftarrow 1$ **to** $d_{max} - p$ **do**
        enum($d + p'$,p',d_{max},$mcard$,i,$prof_{new}$,$recipe_{new}$,$sample$,$pset$)
      end
    end
  end
end

After enumeration the set of distinct cover profiles is transformed into a set-covering problem (see Chapter 4) which can be solved very efficiently using ILP solvers.

A sample benchmark experiment to test the algorithm was conducted on a real dataset from 39 sample for a multiplex assay for 25 analytes. At first the solution for maximum coverage using 3 un-pooled samples was calculated. Here the optimal solution covered 51 of 75 ranges (68%). If sample pooling with a maximum of 2 parts per pool was enabled, 64 of 75 (90 %) could be covered. If 3 parts per pool were allowed, 69 of 75 (92 %) ranges were covered by the solution. A further increase of the pooling parts did not improve the results. Calculations on this very small toy problem were extremely fast, and took only a few seconds using the enumeration heuristic and the SCPSolver framework with the CPLEX solver as a back-end.

3.4 Placement of samples in a planar array

In order to avoid crosstalk on reverse phase protein microarrays, it is desirable to arrange the probes so that the bias is minimal. The resulting constraint satisfaction problem is difficult to solve even for small instances.

Reverse phase protein microarrays (RPMAs) are used to measure protein expression levels in the samples. The samples are immobilized on individual spots on a chip. The chip is then incubated with a specific antibody to detect a protein in the individual samples. The detection is then performed using a luminescence read-out. A chip can hold hundreds of samples or replicates. A common issue when designing protein microarrays is the need to avoid side-effects during read-out. It is likely that the intensity measured on

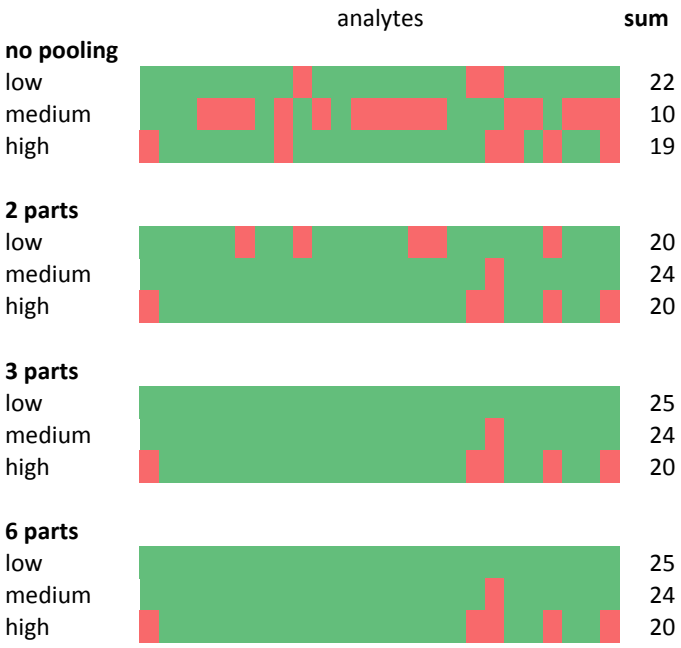


Figure 3.6: Results of the sample pool validation algorithm using different numbers of parts on a dataset with 25 analytes and 39 samples (search space size 1,798,732,635,700). Uncovered ranges are marked in red, covered ranges are marked green. This example shows that coverage improves by allowing sample pooling up to 3-part pools.

one spot is influenced by its neighboring spots. Therefore it is desirable to avoid placing replicates of the sample twice or more often in the same neighborhood if such a situation can be avoided.

The resulting combinatorial design problem is related to graph coloring. Every spot on the array is a vertex in a graph, and is connected to another vertex by an edge in that graph if (and only if), the spot corresponding to that vertex is adjacent to a common spot. A color is assigned to each sample, and an additional color (e.g. black) is defined as 'no sample'. The coloring of the resulting graph, given the additional constraints that each color must be used a specific number of times (number of replicates) - except the color designating an empty spot.

Similar problems have also been observed and studied for cDNA microarrays. However the algorithms are not directly transferable to RPMAs because some challenges resulting from the genechip on-spot synthesization do not apply.

We define the sample arrangement problem as follows: Given an array of m rows and n columns, place l replicates of k samples such that no replicate of the same sample is adjacently placed twice or more often in relation to another spot. We define the spots

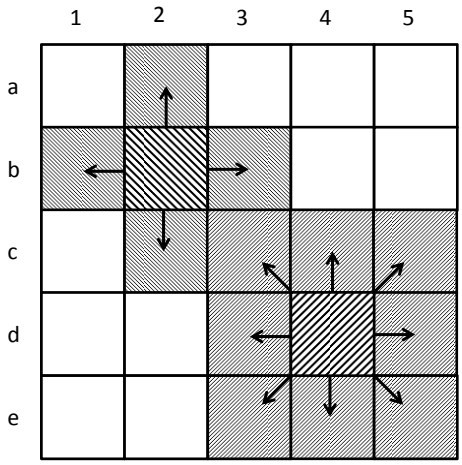


Figure 3.7: This schema depicts neighborhoods in sample placement on protein chips. Different samples must be placed in fields a2,b1,c2 and b3. If diagonal neighbourhoods are respected fields c3,c4,c5,d3,d5,e3,e4 and e5 must be spotted with different samples.

to the left, right, top, and bottom of the spot to be neighbors in the simpler variant, and include the diagonally adjacent in the tougher variant of the problem. Of course the first problem requires at least five different samples and the second at least nine, to be feasible if no empty positions are allowed.

In order to provide solutions to this practical problem in the lab, we have developed the small application ProChOpt, which can be easily used to solve the sample arrangement problem up to medium- or large-sized problems, depending on the applied Integer Linear Programming (ILP) solver package. For the SAP sample placement problem with m rows, n columns, k samples and l replicates, a corresponding ILP can be formulated. Given a binary variable x_{kij}, which indicates whether a sample of type k should be placed on the coordinates row i, column j, the following constraints are defined.

The constraint to occupy one spot with one sample type only:

$$\sum_{k} x_{kij} \leq 1 \tag{3.25}$$

The constraints that each sample should occur only once in a neighborhood:

$$x_{kij} + x_{k(i\pm1)j} + x_{ki(j\pm1)} \leq 1 \tag{3.26}$$

or the constraints that each sample should occur only once in a neighborhood including diagonals:

$$x_{kij} + x_{k(i\pm1)j} + x_{ki(j\pm1)} + x_{k(i\pm1)(j\pm1)} \leq 1 \tag{3.27}$$

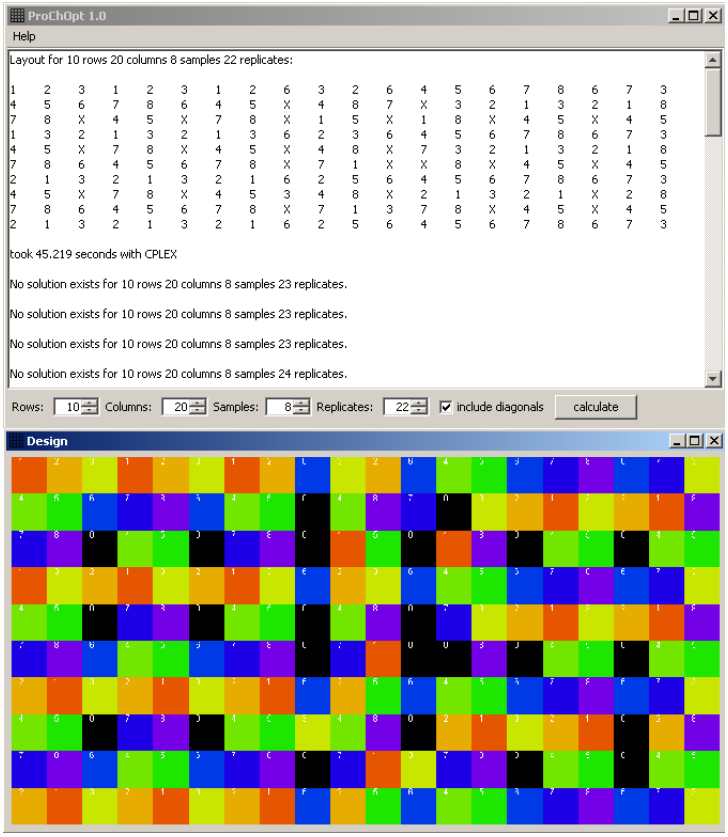


Figure 3.8: Screenshot of the ProChOpt User Interface. The user simply sets the number of columns, rows, samples, replicates and if diagonals should must be considered. The solution is presented as a table (upper part) and as a figure (lower part).

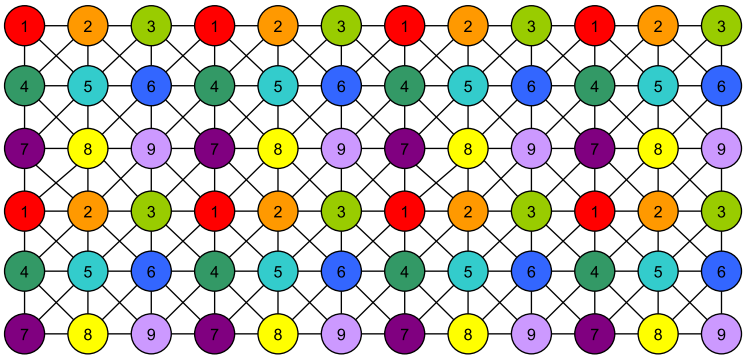
Each sample should be placed in l replicates:

$$\sum_i \sum_j x_{kij} = l \tag{3.28}$$

This formulation has mnk binary variables and $3mnk + k$ constraints. The integer linear program has been implemented using SCPSolver.

We have also developed a straightforward graphical user interface (Figure 3.8). The parameters number of rows, number of columns, number of samples, and number of replicates are set by the user. With one click the ILP is constructed, solved, and then displayed to the user. If multiple solutions are needed, e.g., for randomization, the button can be pressed repeatedly. If a solution is found, a new constraint is added to make sure subsequent solutions are different.

For quadragonal layouts and nine or more different samples, trivial solutions to this problem exist:



However, layouts with sample numbers lower than nine including empty spots are more difficult; they can be solved using the described approach. Of course problems of these type can only be solved by introducing empty spots on the quadragonal layout, because each spot has up to nine neighbors. Empty spots are placed on certain positions to avoid the, else unavoidable, multiple placement of a same sample in the direct neighborhood of one spot (9 neighbors - less than 9 samples). We have solved the problem for 5x5,5x6,... up to 10x10-grid, and tried to find layout to place a maximum number of replicates for 8 and 7 different samples. The results are shown in tables 3.1 and 3.2.

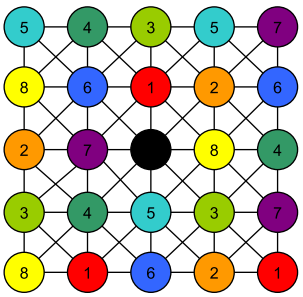
Results show that some grid sizes are comparatively inefficient. For example in a 5x7-grid only 69 % of the available spots can be used if 8 samples are spotted. The same number of replicates can be spotted on a 5x5-grid where in contrast 96 % of the cells are used. This is the resulting layout, showing a distinctive pattern, all replicates of the same sample are repeated in the shape of an asymmetrical 'T':

Table 3.1: This table shows the maximum number of replicates of 8 different samples that can be placed on a quadragonal layout of a given size. The percentage in parentheses is the share of occupied cells of the grid (e.g. 3 replicates times 8 replicates occupy 24 of 25 cells on a 5x5 layout = 96 %)

#rows/#columns	5	6	7	8	9	10
5	3 (96 %)	3 (80 %)	3 (69 %)	4 (80 %)	5 (89 %)	5 (80 %)
6		4 (89 %)	4 (76 %)	5 (83 %)	6 (89 %)	6 (80 %)
7			5 (82 %)	6 (86 %)	7 (89 %)	7 (80 %)
8				7 (87 %)	8 (89 %)	9 (90 %)
9					9 (89 %)	10 (89 %)
10						11 (88 %)

Table 3.2: This table shows the maximum number of replicates of 7 different samples that can be placed on a quadragonal layout of a given size.

#rows/#columns	5	6	7	8	9	10
5	3 (84 %)	3 (70 %)	4 (80 %)	5 (87 %)	5 (78 %)	6 (84 %)
6		4 (78 %)	4 (67 %)	5 (73 %)	6 (78 %)	6 (70 %)
7			5 (71 %)	6 (75 %)	7 (78 %)	8 (80 %)
8				7 (77 %)	8 (78 %)	9 (79 %)
9					9 (78 %)	10 (78 %)
10						11 (77 %)



This solution to this problem instance, which is probably a singularity, is non-trivial and the lab scientist may not have arrived at it very easily. Here the constraint programming formulation implemented in ProChOpt was able provide an answer, also on the question if it is at all feasible to place a number of replicates on a predefined grid, respecting the layout restrictions, or not.

Chapter 4

Optimization for immunoaffinity-MS

The contents of this chapter were published as an article titled 'Optimal selection of epitopes for TXP-immunoaffinity mass spectrometry' (Planatscher *et al.*, 2010).

As the lab proof of concept for TXP proteomics has been shown, the question arose which epitopes should be targeted to cover a large set of proteins with minimal effort based on prior knowledge of a proteome sequence. A method to select and optimize TXP-antigens, the short common terminal sequences (epitopes), is presented to cover a given set of target proteins. This leads to a substantial reduction of antibodies to be generated for a proteome wide immunoaffinity-MS approach. An in-silico digest of a fully elucidated target proteome is filtered to eliminate those peptides with undesirable properties or epitopes. The problem of selecting the minimal set of TXP-antigens is equivalent to the set cover problem. A greedy algorithm and a boolean programming approach is applied. These methods are extended to enhance the multiple coverage of the protein targets for a better experimental design.

4.1 Complexity reduction through a filter pipeline

Starting from a proteome dataset (e.g. UniProt or IPI) that is defined as the background, an in-silico tryptic digest is obtained. It is assumed that the background dataset holds information about all proteins found in the to be studied sample.

Peptides must have certain properties to be detectable by a read-out method. The mass of the peptide has to be known and, in addition, mass-spectrometers have limits in resolution and mass range. Instead of including these limitations in optimization-constraints, a filter pipeline is applied where peptides and epitopes, which do not match the criteria, are removed.

Here, the digest of a proteome P is defined by a set of pairs $D(P) = \{(P_i, p_j)\}$, where p_j is the j-th peptide in protein P_i. $p_i = a_1 a_2 \dots a_n$ is an amino acid sequence composed of the single letter amino acid alphabet. We define a peptide-antibody-combination as a quadruple labelled p_{ij}^{tl}:

$$p_{ij}^{tl} \equiv (P_i, p_j, t, l) | t \in \{n, c\}; l > 1 \, . \tag{4.1}$$

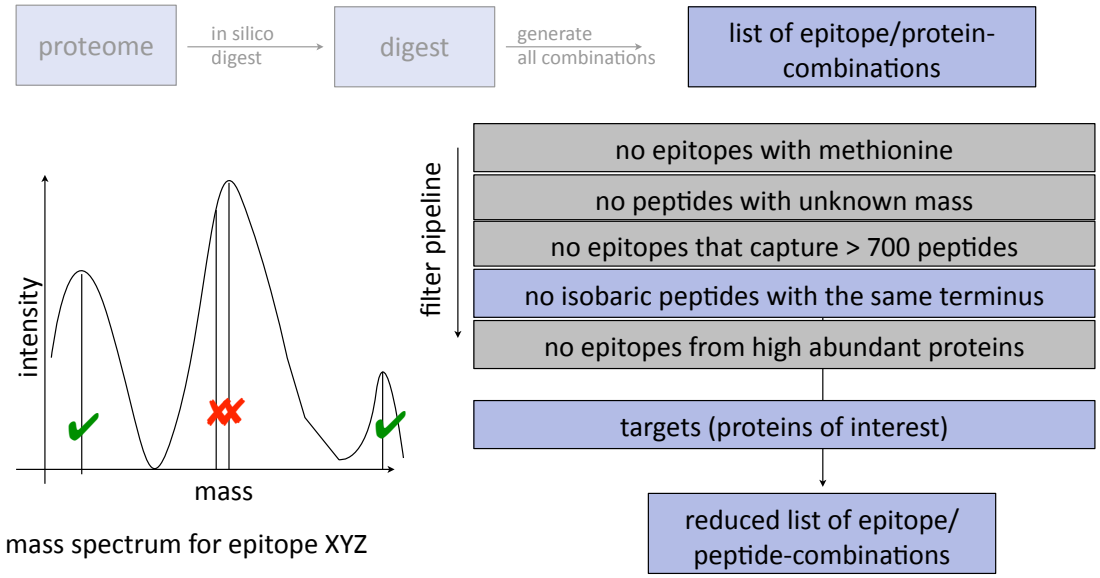


Figure 4.1: The filter pipeline removes peptides and epitopes with undesired properties, e.g. the property that two or more peptides with the same terminal sequence have a very similar mass.

Here, l defines the length of the epitope and t describes whether the terminus is n- or c-terminal. The set

$$C_{all} = C(D(P), T, l_{min}, l_{max}) = \{p_{ij}^{tl}\} \tag{4.2}$$

with

$$(P_i, p_j) \in D(P) \tag{4.3}$$

$$l \in \{l_{min}, \ldots, l_{max}\}; T \in (\{n, c\}) \tag{4.4}$$

contains all combinations for a given proteome, length range and termini. This combination set is the raw start input for the filter pipeline. The quadruple is not needed for every filter, but for reasons of formal continuity we use the definition through the whole specification of the pipeline.

Knowing the weight of captured peptides is essential for the mass spectrometry readout. Therefore, the *unknown positions filter* removes peptides containing unknown positions (symbol X), as their weight cannot be calculated.

The *methionine filter* removes combinations with epitopes containing methionine (symbol M), since chemical modifications of methionine may hamper the recognition of the target epitope by a binding molecule, especially by an antibody.

The *high abundant epitope filter* removes combinations with epitopes which would capture a large number of peptides. An antibody affine to such an epitope would be cluttered, and therefore be rather insensitive. We define a subset $C^e \subset C$ which contains all combinations $p_{ij}^{tl} \in C$ where $epitope(p_{ij}^{tl}) = e$. If $|C^e|$ is bigger than 600, the epitope e would not be considered for optimization.

The *weight filter* removes combinations which share the same terminus and have almost the same weight. These peptides can not be measured with standard mass spectrometry read-out, because the resulting peaks would overlap in the spectrum. A reasonable value for Δ_{min} is 2-10 Da for MALDI-TOF-spectrometers. In this filter, rather than excluding the terminus from the optimization only the almost isobaric peptides are not counted as identifiable by combining the specific epitope and mass information. For example the peptides AYEQLGYR and HLEILGYR could not be discriminated in a mass spectrum of a probe enriched with an antibody affine to the epitope LGYR, because the masses only differ by 1.068 Da, if the resolution of the mass spectrometer does not provide the adequate resolution.

The *length filter* removes combinations which do not fit in the detection range of the mass spectrometer. The detection range depends on the technical specifications of the mass-spectrometer, but a range from 8-30 amino acids is a good rule of thumb.

Some proteins occur with great abundance in the sample, such as actin or tubulin. Terminal epitopes of peptides from these proteins are unsuitable as epitopes for immunoaffinity experiments for the same reasons explained in the high abundant epitope filter. In this last filter step an epitope stop list, generated from a hand curated list of high-abundant proteins, is used to remove those from the list of combinations.

As shown in figure 4.1, filters are usually applied in a specific order. While the methionine, unknown positions, high abundant protein filters can be applied at any position in the pipeline, other filters are order-dependent. This is the case if a filter evaluates the expected peptide distribution C^e of an epitope e. These filters cannot be preceded by filters that change those distributions. The high abundant epitope filter must precede the weight filter, which must precede the length filter.

Through the application of this filter pipeline the preselection of epitopes is adjusted to the experimental setup and the problem dimension is significantly reduced.

The influence of the filters is shown in Table 4.1. While the unknown-positions-filter and the methionine-filter have a relatively small impact, the high abundant epitope filter and the weight filter remove a large number of combinations. The weight filters reduce the number of combinations by about 43 %, while the number of epitopes is only reduced by 3 %. The filter removes combinations from the set, which cannot contribute to the coverage (overlapping peaks). Still the corresponding antibody can capture peptides that are detectable by the mass spectrometer.

Some antibodies ('robinson antibodies') capture only one peptide from one protein. If there is an antibody that captures more peptides from the same protein and others, it is always better to choose this one over the 'robinson antibody'. Therefore all robinson antibodies are removed before the optimization starts.

Table 4.1: Impact of the different filters applied to the in-silico tryptic digest of the human proteome (UniProt taxon id 9606), N-C-terminal epitopes of length 4 and 5, $\Delta_{min} = 4$, $l_{min}^{filter} = 6$, $l_{max}^{filter} = 30$

Filter	# epitopes	# proteins	# combinations
unfiltered	671,427	20,333	4,196,636
unknown positions filter	671,253	20,333	4,195,788
methionine filter	569,365	20,332	3,839,772
high abundant epitope filter	569,354	20,332	3,312,617
weight filter	559,323	20,178	1,962,034
length filter	530,863	20,020	1,662,437
high abundant protein filter	527,164	20,010	1,598,289

4.2 Protein set cover problem formulation

The bipartite graph $G = (P \cup A, E)$ is constructed by adding proteins and epitopes as vertices, and by connecting a protein node from the protein set P and an epitope node from the epitope set A if a combination appears in the filtered set:

$$G = (P \cup A, E) \tag{4.5}$$

$$E = \{(e, P_i) | p_{ij}^{tl} \in C \wedge epitope(p_{ij}^{tl}) = e\} \tag{4.6}$$

The problem is to select a minimal set of antibodies $A_{min} \subset A$ so that every protein in P is covered by at least one epitope. The minimum set cover is a classical problem in computer science and complexity theory. The set cover can be formulated as a decision problem, where the question is asked, if a covering set of size k or less exists. This problem was shown to be NP-complete and achieving approximation ratios is no easier than computing optimal solutions (Arora, 1998). The optimization version where the smallest covering set has to be found is NP-hard. It was shown that a greedy algorithm has an approximation ratio of

$$H(n) = \sum_{k=1}^{n} \frac{1}{k} \leq \ln n + 1, \tag{4.7}$$

where n is the size of the largest set (Lund and Yannakakis, 1994).

This the best approximation ratio for the set cover problem (Feige, 1998). In this algorithm (see Algorithm 3) in each step the epitope in A covering the most yet uncovered proteins in P, is added to the solution set L, until all proteins are covered.

Another approach to the solution of the set cover problem is to formulate it as a binary linear program. The binary decision variables s_a reflect the inclusion of an epitope a to

Algorithm 3: The greedy set cover algorithm

Input: bipartite epitope-protein graph $G(P \cup A, E)$
Output: set of epitopes L
$P_{cov} = \emptyset$;
$L = \emptyset$;
while $P \setminus P^{cov} \neq \emptyset$ **do**
  foreach $a \in A \setminus L$ **do**
    //calculate how many new proteins are covered by the epitope a
    $score(a) = |\{(a,p) \in E | p \notin P_{cov}\}|$;
  end
  //select the epitope a with the highest score
  $a_s = \arg\max_a score(a)$;
  $L = L \cup \{a_s\}$;
  $P_{cov} = P_{cov} \cup \{p \mid (a_s, p) \in E\}$;
  //remove the covered proteins from the graph
  $G = G((A \cup P) \setminus (P_{cov} \cup L), E \setminus \{(a_s, P) \in E\})$;
end
return L

the solution set. The number of the selected epitopes forms the objective function:

$$min \sum_{a \in A} s_a \tag{4.8}$$

$$A = \{a | \exists a = epitope(p_{ij}^{tl}); p_{ij}^{tl} \in C\} \tag{4.9}$$

$$s_a \in \{0,1\} \tag{4.10}$$

The linear program is subject to the constraint that every protein P has to be covered by one or more epitopes in the solution:

$$\sum_{a \in A} cov(p_{ij}^{tl}, a) s_a \geq 1 \forall P_i \in P \tag{4.11}$$

$$cov(p_{ij}^{tl}, a) = \begin{cases} 1 & a = epitope(p_{ij}^{tl}) \\ 0 & otherwise \end{cases} \tag{4.12}$$

This program can be solved with available solvers such as CPLEX or GLPK. This will lead to optimal solutions, if the problem dimension is small.

To enhance the accuracy of the proteomics experiments, it would be beneficial to capture the same or multiple peptides from a protein by different binders. In addition it is

beneficial to include alternative binders in the experimental planning, in case the generation of a binder affine to a specific epitope fails.

The multicovering problem (MCP) is a generalization of the set covering problem. Several algorithms have been proposed by Dobson (1982) , Nicholas G. Hall (1992) and Rajagopalan and Vazirani (1993). Those heuristics would solve the problem of covering each protein twice or more. As it would be cost-prohibitive to double the number of binders, it is not possible to cover all target proteins more than once. This is the case at least for proteins that are covered by a very specific epitope. The following approach solves the pragmatic variant of the problem.

The greedy algorithm can be modified to enhance the probability of the selection of an epitope set that meets the multicoverage requirement for the target proteins. In this variant (see appendix) the scoring function combines two different optimization targets, minimality and redundancy, by summation to a one-dimensional multiobjective fitness function.

The function is a weighted sum of the number of proteins which are not yet covered

$$n_{cov}(a) = |\{(a,p) \in E | p \notin P_{cov}\}| \,, \tag{4.13}$$

and the number of proteins which are covered again by this antibody

$$n_{mcov}(a) = |\{(a,p) \in E | p \in P_{cov}\}| \,. \tag{4.14}$$

E denotes the edge set in the bipartite graph and P_{cov} the set of already covered proteins. The influence of new and already covered proteins on the overall score of an epitope is weighted by the parameters s_{mcov} and s_{cov}:

$$score(a) = n_{cov}(a) \cdot s_{cov} + n_{mcov}(a) \cdot s_{mcov} \tag{4.15}$$

As the original version, the algorithm terminates with a total number of epitopes lower or equal as the number targets, because every added epitope is required to cover at least one new target protein.

The choice of the parameters s_{mcov} and s_{cov} has a high impact on the results, and depends heavily on the size of the dataset. The number of epitopes with high capacity is considerably lower in small datasets than in large datasets. Because of this the probability that a protein can be covered more than once by different high capacity epitopes is small. In large datasets the situation is the opposite. As many epitopes have a very large capacity, and possibly cover up to a few hundred peptides from many different proteins, it is more probable that the sets of captured proteins overlap. In this configuration it is better to score innovation over redundancy. While this is intuitively clear, it would be a big effort to determine the best values analytically. For large datasets s_{mcov} should be chosen smaller than s_{cov}, for small datasets $s_{mcov} > s_{cov}$.

Multiple coverage can be integrated to the Integer Program formulation by changing the coverage constraints to

$$\sum_{a \in A} cov(p_{ij}^{tl}, a) s_a \geq 2 \tag{4.16}$$

for all proteins that can be covered at least twice. However this will lead to inclusion of elongated, already selected, epitopes (e.g. IER and EIER), to satisfy the double coverage constraints.

This formulation requires that all proteins are multiply covered by the solution. A better formulation reads as follows: Maximize the number of multiply covered proteins in a valid covering of all proteins, by using a fixed number of epitopes. The objective function maximizes the number of proteins which are multi-covered.

$$max \sum_{i=1}^{m} S_i \tag{4.17}$$

If the binary variable S_i is set to one, protein i has to be covered at least twice. This is guaranteed by using the following constraint:

$$\sum_{a \in A} cov(p_{ij}^{tl}, a) s_a - S_i \geq 1 \, \forall P_i \in P \tag{4.18}$$

If S_i is selected, at least two covering epitopes have to be selected in order to satisfy the constraint. This problem would be easily solved just by picking two epitopes randomly for each protein. In order the get an optimal usage of the epitopes their number is restricted by an additional constraint:

$$\sum_{a \in A} s_a \leq cost_{max} \tag{4.19}$$

Here $cost_{max}$ denotes the maximum number of antibodies to be chosen, and this has to be set by the user and may just depend on the available funding for antibody generation or purchase. An upper bound for $cost_{max}$ is the size of the optimal solution to the original multicover ILP, which already covers all proteins in the dataset twice or more. A lower bound is the minimal cost for the normal covering.

4.3 Optimal antibody subset selection with fixed cost

Budget constraints are very common in scientific projects. Using available funds to achieve the highest possible impact is crucial. In research projects using TXP-antibodies, it would be desirable to capture as many proteins as possible with a given number of binders. Of course this can be achieved using the greedy algorithm just by the inclusion of a termination criterion, when the maximal number of antibodies is reached.

The problem can be easily modelled as a linear optimization program:

$$max \sum_{i=0} p_i \tag{4.20}$$

$$\sum_{a \in A} cov(P_i, a) s_a \geq p_i \;\; \forall P_i \in P \tag{4.21}$$

$$\sum_{a \in A} s_a = k \tag{4.22}$$

The boolean variable p_i states whether protein i should be included or not. The sum of the selected proteins has to be maximized. If p_i is set to 1 the coverage constraints (4.21) ensure that at least one antibody has to be selected in order to cover the protein P_i. Finally the budget constraint (4.22) limits the maximal number of antibodies in the solution to k. This boolean linear program has $|A| + |P|$ binary variables and $|P| + 1$ constraints.

4.4 Results and Discussion

Proteomes of various organisms (*Homo sapiens*, *Mus musculus*, *Rattus norvergicus*, *Bos taurus*, *Saccharomyces cervisiae*) were obtained from UniProtKB (Wu *et al.*, 2006). Only reviewed sequences were included in the dataset. The proteomes were trypsin-digested in-silico, by cutting after lysine (K) or arginine (R), if no proline (P) followed. A complete digest without missed cleavages or mis-cleavages was assumed. The resulting digests were pre-processed and filtered as described. To investigate the use case of assay designs for a limited number of targets, the lists of proteins associated to the pathways for TGFβ , WNT and TLR signaling were obtained from the KEGG (Kyoto Encyclopedia of Genes and Genomes) PATHWAY database (Ogata *et al.*, 1999). The KEGG gene IDs in the pathway descriptions were mapped to UniProt IDs. The combination sets for the pathways were extracted from the filtered combination set of the human proteome. The coverage score $\frac{|L|}{|P|}$ of a solution L is the number of required epitopes relative to the number of proteins to cover.

The solutions of the integer program delivered by the industry standard ILP solver CPLEX after a limited running time of 12 hours were, not surprisingly, superior to the solutions provided by the greedy algorithm on all tested proteomes and epitope-length combinations (see Table 4.2). The inclusion of epitopes of length five increased the problem dimension considerably, because of the much larger number of potential epitope sequences ($O(20^{l_{max}})$). This increased the number of coverable proteins in the final combination set. Nevertheless the number of required epitopes was decreased in three of five proteomes (Homo sapiens, Rattus norvegicus, Bos taurus). When including terminal sequences of length four and five, the set cover will include shorter epitopes in most cases as they cover more proteins. If for a specific protein all epitopes of length four have been filtered out, longer sequences can still be used to cover it.

The solutions provided by the multicoverage integer program are significantly larger than the solutions, in which multicoverage was not enforced. The multicoverage greedy approach only favors but does not enforce multicoverage, so the solutions provided by this method are smaller, but not necessarily superior to those provided by the multicoverage integer program. As shown in Table 4.3 the number of multicovered proteins (Homo

Table 4.2: Comparison of the solution size (smaller is better) of the integer program IP (CPLEX solver, running time limited to 12 hours) and the greedy set cover algorithm on proteomes of different species, and epitope length settings. $|A|$ denotes the number of different epitopes, $|P|$ the number of target proteins, the percentage next to solution sizes is the coverage score $\frac{solution\,size}{|P|}$

proteome	length	IP	Greedy	$\|A\|$	$\|P\|$
Homo sapiens	4-5	2,020 (10.1 %)	2,292 (11.5 %)	527,164	20,010
Mus musculus	4-5	1,541 (9.6 %)	1,727 (10.8 %)	473,406	15,995
Rattus norvegicus	4-5	851 (11.7 %)	970 (13.3 %)	273,558	7,295
Bos taurus	4-5	790 (14.2 %)	903 (16.1 %)	199,735	5,584
Saccharomyces cervisiae	4-5	1,000 (15.6 %)	1,134 (17.6 %)	240,253	6,422
Homo sapiens	4	2,026 (10.1 %)	2,306 (11.5 %)	86,963	19,979
Mus musculus	4	1,529 (9.6 %)	1,737 (10.9 %)	83,073	15,974
Rattus norvegicus	4	858 (11.8 %)	975 (13.4 %)	64,058	7,294
Bos taurus	4	792(14.2 %)	896 (16.1 %)	53,751	5,576
Saccharomyces cervisiae	4	995 (15.5 %)	1,130 (17.6 %)	58,464	6,405

Table 4.3: Comparison of the solution size (smaller is better) of the IP MC, IP MMC, the greedy set cover algorithm, and the modified algorithm (Greedy MC, $s_{cov} = 100, s_{mcov} = 1$) on the in-silico tryptic digest of the human proteome (UniProt taxon id 9606), N-C-terminal epitopes of length 4, $|L|$ denotes the total size of the solution.

solver	# prot. single covered	# prot. multicovered	$\|L\|$
IP MC	223	19,756	3,895
IP MMC ($cost_{max}$=2,314)	3,665	16,314	2,314
IP	6,164	13,815	2,026
Greedy MC	4,126	15,853	2,314
Greedy	5,132	14,847	2,306

Table 4.4: Comparison of the solution size (smaller is better) of the integer program (IP), integer multicover (IP MC), the greedy set cover (Greedy) and multicover (Greedy MC, $s_{cov} = 1, s_{mcov} = 10$), integer maximization multicover (IP MMC, $cost_{max}$ was set to result of Greedy MC), algorithm on different pathways (subsets of the Homo sapiens proteome), and epitope length settings. The percentage in parentheses is the degree of multicoverage (bigger is better) on the dataset, e.g. a value 50% means that half of the proteins are multiply covered.

pathway	length	IP	IP MC	Greedy	Greedy MC	IP MMC	$\|P\|$
WNT	4-5	55 (13.5 %)	107 (96.2 %)	60 (22.6 %)	88 (54.1 %)	88 (77.4 %)	133
TGF	4-5	36 (8.9 %)	70 (93.7 %)	40 (19.0 %)	57 (49.4 %)	57 (74.7 %)	79
TLR	4-5	47 (5.3 %)	92 (94.7 %)	51 (16.0 %)	72 (46.8 %)	72 (70.2 %)	94
WNT	4	56 (16.6 %)	108 (94.0 %)	63 (15.8 %)	85 (48.8 %)	85 (70.7 %)	133
TGF	4	36 (8.9 %)	71 (89.9 %)	39 (13.9 %)	57 (48.1 %)	57 (69.6 %)	79
TLR	4	47 (3.2 %)	94 (93.6 %)	50 (11.7 %)	70 (45.7 %)	70 (62.8 %)	94

sapiens, $length = 4$) was increased from 14,847 (Greedy) to 15,853 (Greedy MC) by only eight additional epitopes in the solution, compared to the solution of the standard greedy algorithm. This was achieved with a setting of $s_{cov} = 100, s_{mcov} = 1$, which scores not yet covered proteins one hundred times higher than already covered proteins. The solution of the IP MC is 3,895 large, so the effort of multicovering all 19,756 proteins nearly doubles the number of epitopes compared to the solution of the standard IP, where only 13,815 proteins are multicovered. By using IP MMC with the $cost_{max}$ set to the solution size of the Greedy MC the number of multicovered proteins was increased from 15,853 to 16,314.

On the smaller pathway datasets it is possible to calculate the best possible solutions with CPLEX and GLPK in a very short amount of time (less than 2 seconds). Table 4.4 shows a comparison of the solution size and the multicoverage percentages on pathway datasets. On pathway datasets the solution sets are proportionally larger than on proteome datasets. This was expected, because the probability of shared terminal epitopes is smaller if the number of target proteins is reduced. Nevertheless coverage scores of 42 % (WNT, length=4-5, IP, 55 epitopes to cover 133 proteins) are a substantial improvement to the scenario of choosing peptide- or protein-specific antibodies for immunoaffinity-MS. The multicoverage integer program provided solutions with coverage scores from 81 % (WNT, length=4-5) to 100 % (TLR, length=4).

The settings of the multicoverage greedy algorithm were changed to $s_{cov} = 1$ and $s_{mcov} = 10$, because the probability of multicoverage through one epitope is proportional to the size of the datasets. In this way the multicoverage score begins to take effect earlier during the iterative optimization. Table 5 contains results of a grid search on the parameters of the greedy MC algorithm applied to the WNT pathway example. The multicoverage enhancing effect shows only if already covered proteins are scored higher

than new proteins. If s_{cov} is chosen bigger than s_{mcov} the multicoverage effect almost completely vanishes on small datasets.

Table 4.5: Grid search on the parameters s_{cov} and s_{mcov} of the modified greedy set cover algorithm (Greedy MC) on the WNT pathway, N-C-terminal epitopes of length 4: *solution size* (*multicovered proteins*)

s_{cov} \ s_{mcov}	1	2	3	4	5	6	7	8	9	10
1	73 (50)	84 (65)	86 (67)	85 (65)	85 (65)	85 (65)	85 (65)	85 (65)	85 (65)	85 (65)
2	63 (36)	73 (50)	79 (60)	84 (65)	86 (67)	86 (67)	86 (67)	85 (65)	85 (65)	85 (65)
3	61 (33)	64 (37)	73 (50)	79 (60)	80 (61)	84 (65)	86 (67)	86 (67)	86 (67)	86 (67)
4	61 (33)	63 (36)	64 (37)	73 (50)	79 (60)	79 (60)	80 (61)	84 (65)	86 (67)	86 (67)
5	61 (33)	61 (33)	64 (37)	64 (37)	73 (50)	79 (60)	79 (60)	80 (61)	80 (61)	84 (65)
6	61 (33)	61 (33)	63 (36)	64 (37)	64 (37)	73 (50)	79 (60)	79 (60)	79 (60)	80 (61)
7	61 (33)	61 (33)	61 (33)	64 (37)	64 (37)	64 (37)	73 (50)	79 (60)	79 (60)	79 (60)
8	61 (33)	61 (33)	61 (33)	63 (36)	64 (37)	64 (37)	64 (37)	73 (50)	79 (60)	79 (60)
9	61 (33)	61 (33)	61 (33)	61 (33)	64 (37)	64 (37)	64 (37)	64 (37)	73 (50)	79 (60)
10	61 (33)	61 (33)	61 (33)	61 (33)	63 (36)	64 (37)	64 (37)	64 (37)	64 (37)	73 (50)

After the calculation of the greedy multicover the resulting cost (solution size) was used as the cost limit $cost_{max}$ for the maximization multicover (IP MMC) formulation. The results were significantly better multicoverage percentages for all datasets for the same costs (Table 4.4, compare columns Greedy MC and IP MMC).

Both approaches for budget-constrained scenarios have been tried on the described datasets . On the smaller pathway datasets the greedy algorithm performed well. Near-optimal results were found for all pathways. By solving the integer program the coverage could be improved by one to two proteins for WNT and TLR. The solution of the integer program takes only a few seconds, so even if the benefit for solving the integer program is small, it is cheap from the computational perspective.

Table 4.6: Comparison of the achieved coverage size (bigger is better) of the integer program (IP), the greedy set cover (Greedy) algorithm on different pathways (subsets of the Homo sapiens proteome)

pathway	length	k	IP	Greedy	$\|A\|$	$\|P\|$
WNT	4,5	5	31	29	9.047	133
TGF	4,5	5	22	21	5.185	79
TLR	4,5	5	21	21	6.177	94
WNT	4,5	10	46	44	9.047	133
TGF	4,5	10	37	36	5.185	79
TLR	4,5	10	36	36	6.177	94

On the larger proteome datasets the greedy algorithm delivered optimal results for $k = 10$. For budgets of $k = 50$ the solution of the integer program led to small improvements. However if the budget was limited to 100 antibodies the IP approaches delivered better results for all datasets, with the restriction of wall clock time to 24 hours.

Table 4.7: Comparison of the achieved coverage size (bigger is better) of the integer program (IP), the greedy set cover (Greedy) algorithm on different proteomes on the subset cover problem with different budget constraints k

proteome	length	k	IP	Greedy	$\|A\|$	$\|P\|$
Homo sapiens	4,5	10	988	987	527,164	20,010
Mus musculus	4,5	10	923	923	473,406	15,995
Bos taurus	4,5	10	502	502	273,558	5,584
Saccharomyces cervisiae	4,5	10	639	639	240,253	6,422
Homo sapiens	4,5	50	3,974	3.966	527,164	20,010
Mus musculus	4,5	50	3,644	3,636	473,406	15,995
Bos taurus	4,5	50	1,760	1759	273,558	5,584
Saccharomyces cervisiae	4,5	50	2,036	2,034	240,253	6,422
Homo sapiens	4,5	100	6,494	6,578	527,164	20,010
Mus musculus	4,5	100	5,890	5,923	473,406	15,995
Bos taurus	4,5	100	2,685	2,674	273,558	5,584
Saccharomyces cervisiae	4,5	100	3,032	3,031	240,253	6,422

This result puts the additional effort of solving the ILP for proteome-sized problem into perspective, as the relative improvements are small. The greedy approach seems to solve k-size covering problem adequately for larger datasets.

4.5 Conclusions

Starting from the real-world lab engineering task, we have shown that the problem of choosing a minimal set of epitopes is equivalent to the well-known set cover problem. In combination with a filter pipeline that eliminates unsuitable peptide-epitope combinations, we proposed different methods for the solution of the problem.

For small datasets (a few hundred proteins) it is possible to solve the problem to optimality with minimal computational effort using commercial or free solvers. Larger datasets, like full proteomes, require the use of heuristics, or respectively a running time limitation of the branch-and-bound search in the integer program solvers. Large sets of proteins can theoretically be covered by TXP-antibodies with a fraction (down to 9.57 %, see Table 4.2) of the otherwise required peptide-specific antibodies for every protein. We further proposed methods to enforce (IP MC) or enhance (Greedy MC, IP MMC) the multiple coverage of a protein for a better experimental design.

Chapter 5

Optimization for TXP Sandwich-Immunoassays

Contents of this chapter have been presented as a talk at the '26th European Conference on Operational Research' (Planatscher *et al.*, 2013a) and have been subject of a patent application (Joos *et al.*, 2010).

In this approach two TXP-antibodies with epitopes of 3-5 amino acids length will be combined to a sandwich-immunoassay. When both antibodies concurrently bind, a unique 'split epitope' of 6-10 amino acids is identified. This is often sufficient to specifically identify a signature peptide from a protein.

In the classical setup each analyte would be recognized by two specifically made antibodies. Because TXP-antibodies can recognize the same epitope on many peptides, it is possible to reuse them in several combinations. In the best scenario a set of n C-terminal TXP-antibodies and a set of m N-terminal TXP-antibodies can be combined to $m \times n$ immunoassay kits. In comparison the same number of antibodies would only be sufficient for $\frac{n+m}{2}$ classical sandwich-assays.

The task of selecting the smallest-possible sets of C- and N-terminal epitopes for a given set of proteins is different from the immunoaffinity-MS optimization problem discussed in chapter 4. Here quadratic constraints have to be considered. These can be linearized, which leads to problem formulations of even higher dimension. Because of the huge problem dimensions these formulations are very difficult to solve. Therefore a greedy heuristic and a meta-heuristic using local search is presented.

5.1 Problem statement

Peptides - sharing the same terminal sequence - are enriched by using 3 to 5 amino acid specific terminal antibodies after a proteolytic digest. In a detection step using a second 3-5 amino acid specific terminal antibody, the peptide is identified. In this case two antibodies bind simultaneously to the peptide.

This process is called a sandwich-assay. In this method, a first set of binding molecules, the capture set, is immobilized on a support or on e.g. Luminex microspheres. A com-

plex protein sample is subjected to fragmentation by tryptic digest. Subsequently, the resulting peptide mixture is brought into contact with the support or the beads. Subsequently, unbound peptides are removed in a washing step.

Then second binding molecules, the detection set, are applied on the support or the beads, respectively. After washing, specific peptides or proteins can be detected. Thereby, it is preferred, if the binding molecules bind to the termini of the detected peptides. More specifically, it is preferred, if the first binding molecules bind to the N-terminus of the peptides while the second binding molecules bind to the C-terminus of the peptides. Of course, this order can be reversed.

5.2 Fast greedy algorithm

After a preprocessing step basic filters are applied to remove peptides with unknown positions or methionine in the terminal sequence from the peptide pool (see chapter 4).

Then a data structure called 'epitope combination graph' is built, which contains all existing epitope combinations. For each epitope a 'node' is found in the data structure. If a peptide has a certain epitope combination, the nodes corresponding to the epitopes are connected with a 'peptide edge'. These edges are directed, meaning that they have start and end epitope. The direction of an edge defines which epitope is used for capture (start) and detection (end). Only peptides (length 12 AA) which can bind two antibodies are added to the graph. Parallel peptide edges occur when a terminal combination is not unique, meaning that two or more peptides have the same n- and c-terminal sequence. If the optimizaton is done for a sandwich immunoassay, these edges must be removed, as the read-out by immunofluorescence can not distinguish the signals afterwards.

After this step, referred to as interference filter, the epitope combination graph is ready for the optimization step.

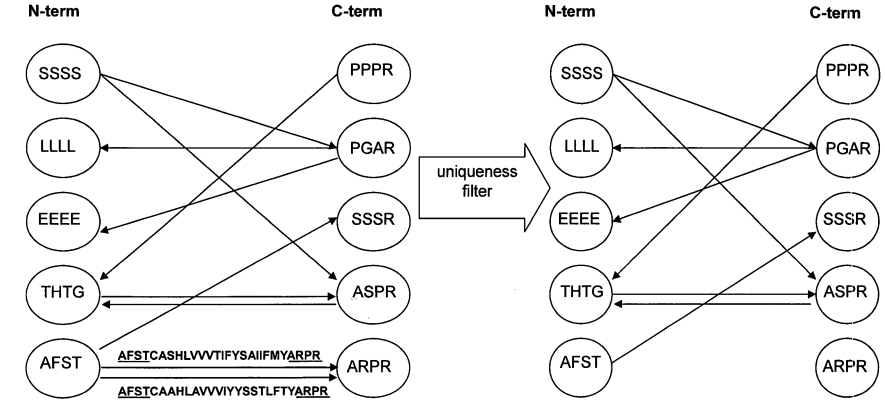
In the optimization step a score is assigned to every epitope node. The score is a weighted sum of the number of newly captured proteins and the number of newly identified proteins, by adding this binder to the solution set:

$$score(a) = |cap(a) \setminus P_{cap}|w_{cap} + |cov(a) \cap P_{cap}|w_{cov} \tag{5.1}$$

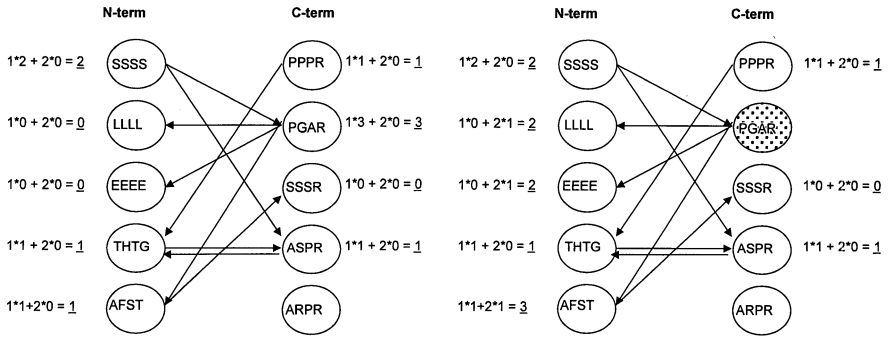
Table 5.1 shows the impact of the weights w_{cap} and w_{cov} in the scoring function on the final outcome. In this problem instance best coverage was achieved by a $w_{cap} = 5$ and $w_{cov} = 2$. In any case $w_{cap} > w_{cov}$ produces better outcomes.

After the scoring, the highest ranking node is selected. If a protein has been identified, all peptides from this protein are removed from the graph. The scores are then updated.

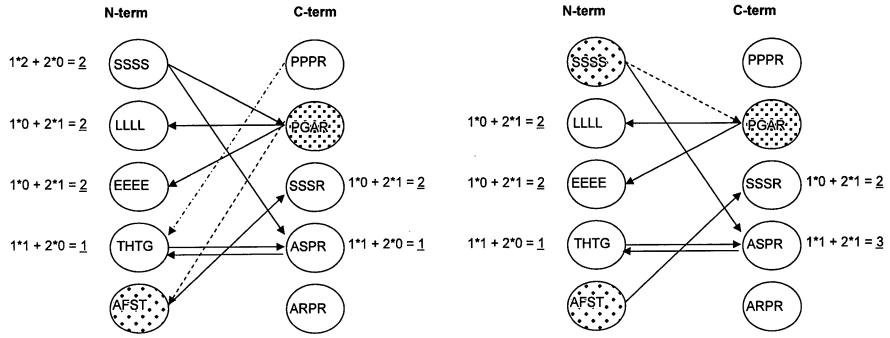
The update process can be done in constant time, because of the used graph data structure can be traversed efficiently. Each peptide-edge is a node in a peptide-protein-graph. By traversing all peptide-edges adjacent to the epitode-node, all proteins including this epitope can be visited in linear time.



(a) Uniqness filter



(b) Example iteration 1 and 2



(c) Example iteration 2 and 3

Figure 5.1: This figure shows an example for an instance of the TXP sandwich immunoassay greedy approach

Algorithm 4: The parametrized greedy algorithm

Input: bipartite epitope-protein graph $G(P \cup A, E)$
Output: set of epitopes S
$P^{cov} = \emptyset$;
foreach $a \in A \setminus L$ **do**
  //calculate how many new proteins are covered by the epitope a
  $score(a) = |cap(a) \setminus P_{cap}|w_{cap} + |cov(a) \cap P_{cap}|w_{cov}$;
end
while $P \setminus P^{cov} \neq \emptyset$ **do**
  //select the epitope a with the highest score
  $a_s = \arg\max_a score(a)$;
  $P^{cov} = P^{cov} \cup \{cov(a) \cap P_{cap}\}$;
  $P^{cap} = \{P^{cap} \cup cap(a)\} \notin P^{cov}$;
  $S = S \cup \{a_s\}$;
  //update all epitope-scores
  update();
end
return L

Each epitope node contains variables counting how many not yet captured proteins could be covered and how many captured proteins could be marked by the binder. When an epitope is selected, the algorithm updates these counters for all epitopes adjacent to the newly captured/identified proteins. The update procedure can be summarized as follows: For all peptide-edges connected to the selected node, visit the adjacent protein-node, from there iterate over all peptide-edges and update the counters in the epitope nodes.

After the update step the epitope ranking must be re-sorted, to get the next top-scoring node. This would require an $O(n\,logn)$ step at each iteration. Given the very large epitope lists ($\geq 10^6$), this leads to unnecessary long computation times.

To avoid resorting all nodes according to their changed score, a queuing data structure is used. This queue allows to reorder only the nodes which have actually changed score. The data-structure can be updated in $O(m\,logk)$-time where m is the number of changed nodes and k is the number of different score values. Without going into too much detail the nodes are sorted in score slots. By the use of memory-expensive hash-tables each element can be quickly located and removed from or inserted in the correct score slot.

The current implementation of this greedy algorithm takes less than a second to provide a solution for very large proteome sized instances, after the epitope combination has been constructed. Even though the implementation trades speed for memory about 2 GB of memory are sufficient for all calculations presented in the following sections.

w/v	1	2	5	10	100	1000
1	13,670	13,652	13,624	13,612	13,612	13,612
2	13,702	13,670	13,603	13,624	13,612	13,612
5	13,828	13,762	13,670	13,652	13,612	13,612
10	13,820	13,828	13,702	13,670	13,612	13,612
100	13,887	13,919	13,871	13,820	13,670	13,612
1000	13,867	13,867	13,867	13,867	13,820	13,670

Table 5.1: Dataset: Homo sapiens, 17885 identifiable proteins, 296170 peptides, 76866 different terminal epitopes

5.3 Linear integer programming approach

The optimization problem stated above can be formulated as a quadratically constrained integer program. The objective is to minimize the number of selected antibodies out of the set A,

$$min \sum_{a \in A} s_a \tag{5.2}$$

subject to the constraint that every protein in P can be identified:

$$\sum_{a \in cap(P_i)} \sum_{b \in mark(a,P_i)} s_a s_b \geq 1 \quad \forall P_i \in P \tag{5.3}$$

$$s_a \in \{0,1\} \quad \forall a \in A \tag{5.4}$$

. The function $cap(P_i)$ returns the set of all antibodies capturing an antibody in protein P_i. The function $mark(a,P_i)$ returns the marking antibodies for a selected antibody a in protein P_i. If s_a is 1, then antibody a is selected.

This quadratically constrained boolean linear program can be reformulated as a linear boolean program by introducing new variables and constraints.

$$min \sum_{a \in A} s_a \tag{5.5}$$

subject to the constraint that for each protein in P one peptide is selected:

$$\sum_{p \in D(P_i)} s_p \geq 1 \quad \forall P_i \in \bar{P} \tag{5.6}$$

and that each selected peptide is covered by at least one antibody on the n-terminus and one at the c-terminus

$$\sum_{a \in nterm(p)} s_a \geq s_p \quad \forall s_p \tag{5.7}$$

$$\sum_{a \in cterm(p)} s_a \geq s_p \quad \forall s_p \tag{5.8}$$

$$s_a \in \{0,1\} \quad \forall a \in A \quad s_p \in \{0,1\} \quad \forall p \in D(p) \tag{5.9}$$

The functions nterm(p_i)/$cterm(p_i)$ return the n/c-terminal antibodies for a peptide p_i. Latter formulation can be solved by an advanced solver for integer linear programs as GLPK or CPLEX. The number of variables is $|D(P)| + |A|$, the number of constraints is $|D(P)| + |P|$, excluding the integer constraints. The drawback of this formulation is that the numbers of variables and proteins gets very large fast, e.g. the number of variables used in the model for the human proteome would exceed one million binary variables. This makes it difficult to solve those whole proteome instances with this classical approach. Smaller instances, as pathways or other protein groupings, however can be solved to optimality.

5.4 Metaheuristics

The naive linearized mathematical program for the solution of the TXP sandwich cover problem has a huge number of binary decisions and constraints, which makes it hard to solve directly even using state of the art MILP solvers. We propose a hybrid method, which calculates a solution using the greedy algorithm first, then removes a subset of that solution, and then solves a linear program to repair the solution in an optimal way.

Algorithm 5: The metaheuristic algorithm

Input: bipartite epitope-epitope graph $G = (A_{cap} \cup A_{mark}, E, E \rightarrow P)$,
Output: set of epitopes S
$L = Greedy(v, w, G)$;
while $it \leq it_{max}$ **do**
  $\hat{L} \subset L$ with $|\hat{L}| = s$;
  for $i = 1; i \leq s; i = i + 1$ **do**
    $j = rnd(|L \setminus \hat{L}|)$;
    $\hat{L} = \hat{L} \cup \{l_j\}$;
  end
  $L = SolveSubsetILP(G, L \setminus \hat{L})$;
end
return L

The integer linear program for conducting the local search is defined as follows. First a subset $\hat{L}$ of size m is selected from the current solution epitope set L. Determine the set of proteins $\hat{P}$ which is not covered anymore by the remaining solution set $L \setminus \hat{L}$.

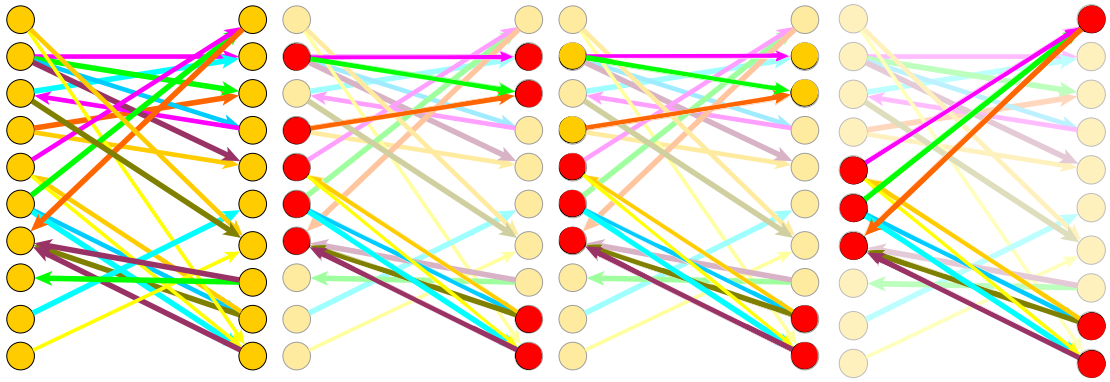


Figure 5.2: Phases of the local search algorithm: Greedy solution, random removal of solution vertices, new solution improved by local search

Now the linear program to augment the solution set such that all proteins are covered again is formulated. The objective function

$$min \sum_{a \in A} c_a s_a \tag{5.10}$$

of the linearized integer program is changed such that all epitopes, which have not been removed have zero cost:

$$c_a = \begin{cases} 0 & a \in L \setminus \hat{L} \\ 1 & otherwise \end{cases} \tag{5.11}$$

This way only epitopes that are newly included in the solution set are minimized, while epitopes already present in the solution set will not impact the objective value of the function.

The linear program is subject to the constraint that for each uncovered protein in $\hat{P}$ one peptide is selected:

$$\sum_{p \in D(P_i)} s_p \geq 1 \quad \forall P_i \in \hat{P} \tag{5.12}$$

As of before each selected peptide must be covered by at least one binder on the n-terminus and one on the c-terminus

$$\sum_{a \in nterm(p)} s_a \geq s_p \quad \forall s_p \tag{5.13}$$

$$\sum_{a \in cterm(p)} s_a \geq s_p \quad \forall s_p \tag{5.14}$$

$$s_a \in \{0,1\} \quad \forall a \in A \quad s_p \in \{0,1\} \quad \forall p \in D(p) \tag{5.15}$$

The linear program is smaller than the model to solve the full problem, and can be influenced by the subset size m. Large values of m can lead to very expensive local search steps, while small values of m are too small perturbations to leave the greedy local optimum. If the optimal solution to this problem has an objective smaller than the size of the removed subset m, the local search found a better solution to the optimization problem. This process is repeated iteratively.

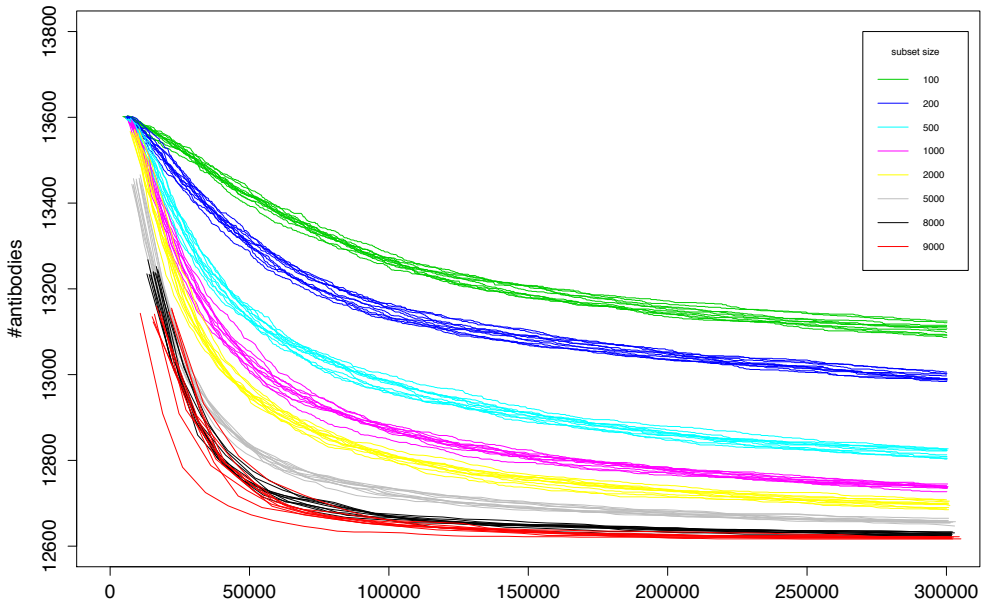


Figure 5.3: Influence of the subset size on the fitness-plot.

Eventually after a number of iterations only little or no more progress is made, because the heuristic has converged to a local optimum. The influence of the subset size on the speed of convergence is shown in figure 5.3 and figure 5.4. The figures summarize the progress of multiple runs of the heuristic applied to the same datasets which has been used for table 5.1. The runtime was restricted to 300 seconds. Results show that small subset sizes (100, 200, 500) lead to convergence of the search process in local optima far from the optimal solution. A lower bound for this problem instance is 12.558 (best non-integer node found by CPLEX after 10 minutes). The initial greedy solution size is 13.603 and the best integer solution obtained by CPLEX is 12.744. The average solution size obtained by the metaheuristic after 5 minutes with a subset size of 200 is 13.100, which is a significant improvement over the greedy solution, but still far away from the

optimal solution. Using a subset size of 1.000, solutions sizes around 12.750, similar to the ILP approach, were found. When the subset size parameter was set to larger values, final solution sizes were improved over the best solution provided by CPLEX.

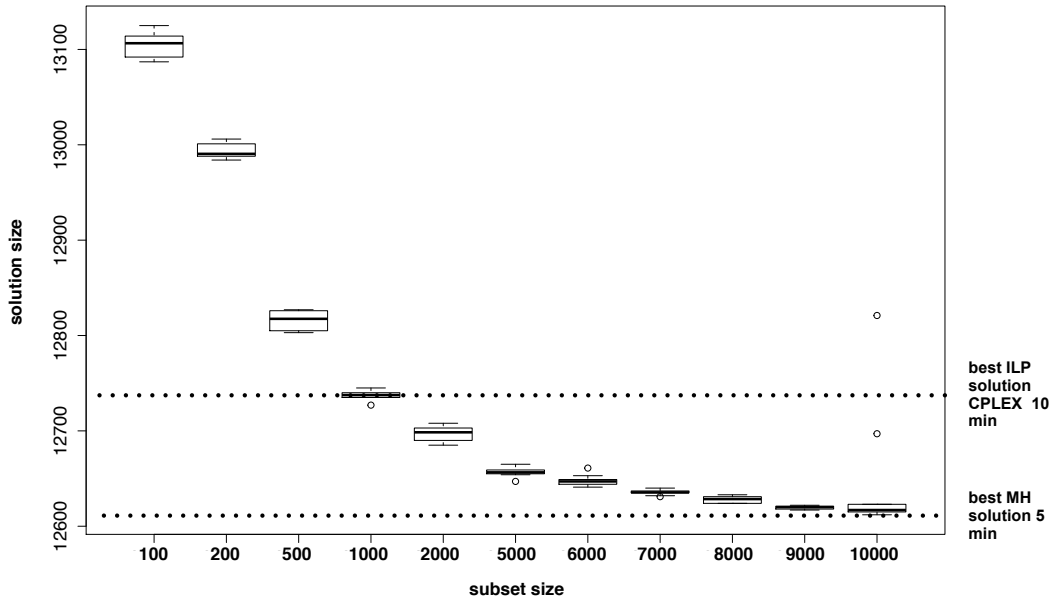


Figure 5.4: Influence of the subset size on the fitness-plot.

Surprisingly even if the perturbations to the solution set were very large, the models could still be solved quickly enough to allow multiple iterations of local search, except for subset size 10.000. In this case some of the local search problems were too large to be solved in the time constraint. This resulted in the two outliers visible in figure 5.4 at subset size 10.000. In any case the reached final solution sizes in multiple runs were stable, meaning that if the subset size parameter was not changed, results of similar quality were obtained on the same dataset in multiple runs.

5.5 Optimal antibody subset selection with fixed cost

Often budget constraints exist and the number of TXP-antibodies is restricted. In this case the number of covered proteins is maximized:

$$max \sum_{P \in \bar{P}} s_P \tag{5.16}$$

subject to constraint that for each protein in P one peptide is selected:

$$\sum_{p \in D(P_i)} s_p \geq s_P \quad \forall P_i \in \bar{P} \tag{5.17}$$

and that each selected peptide is covered by at least one peptide on the n-terminus and the c-terminus

$$\sum_{a \in nterm(p)} s_a \geq s_p \quad \forall s_p \tag{5.18}$$

$$\sum_{a \in cterm(p)} s_a \geq s_p \quad \forall s_p \tag{5.19}$$

and to the constraint that a maximum of n antibodies is selected in total

$$\sum_{a \in A} s_a \leq n \tag{5.20}$$

$$s_a \in \{0,1\} \quad \forall a \in A \quad s_p \in \{0,1\} \quad \forall p \in D(p) \tag{5.21}$$

This linear program has $|\bar{P}| + \sum_{P \in \bar{P}} |D(P)|$ constraints and $|\bar{P}| + \sum_{P \in \bar{P}} |D(P)| + |A|$ variables.

In an experiment the model was applied to the H. sapiens dataset used earlier for epitope lengths 3 and 4, and subset sizes 20, 50 and 100. In this experiment the maximum computation time has been limited to 30 minutes. For the shorter epitope size the results were of course better, due to the larger peptide set enriched by shorter epitopes. For a subset size of $k = 100$ the best solution covered 233 proteins. The same number of distinct antibodies could only cover 50 proteins in the standard setup. This is a theoretical advantage of 233 %. For the epitope length 4 the coverage was worse, however the number of assays can still be doubled using the split epitope assay compared to standard sandwich immunoassays.

Table 5.2: Solutions for the fixed cost sandwich assay cover problem for epitope lengths 3 and 4, and subset sizes 20,50 and 100 on the human proteome set. The rightmost column shows the theoretical advantage of the solution over standard sandwich immunoassays using two specific antibodies per assay.

proteome	epitope length	k	IP	advantage
Homo sapiens	3	20	26	260 %
Homo sapiens	3	50	82	328 %
Homo sapiens	3	100	233	466 %
Homo sapiens	4	20	19	190 %
Homo sapiens	4	50	60	240 %
Homo sapiens	4	100	114	228 %

Chapter 6

Identification of short terminal motifs using peptide mass fingerprinting

The contents of this chapter were published as an article titled 'Identification of short terminal motifs enriched by antibodies using peptide mass fingerprinting' in Planatscher *et al.* (2014).

In the TXP approach, antibodies bind to short linear epitopes present in multiple peptides of complex samples after protein fragmentation by trypsin. *In silico* selection of antigens reduces to minimum the set of TXP antibodies required to cover a pre-defined protein target list (Planatscher *et al.*, 2010).

Search space restriction gained from the revealed binding epitope could improve protein identification from MS and MSMS data. However, the enrichment of proteotypic peptides with TXP antibodies leads to new challenges in the analysis of mass spectrometric datasets. Results from immunoaffinity experiments using TXP-antibodies revealed the enrichment of peptides containing the targeted epitope. Some identified peptides also matched sequence variants (Hoeppe *et al.*, 2011). In that study it became clear that the terms 'specific' and 'unspecific' must be considered inappropriate in characterizing these binders. 'Specificity' generally refers to binding of one protein or peptide to an antibody. Other binding events are deemed to be unspecific, off-target or cross-reactive. In the TXP strategy, the binding of the antibody towards multiple peptides is inherent. Therefore, novel concepts for epitope identification are needed, which would enable properties of the antibody binding domain to be distinguished from interactions occurring elsewhere.

There are three main reasons why unexpected peptides can be detected in the immunoprecipitates. Firstly, this could be due to epitope variations in the polyclonal antibody. Secondly, off-target binding can occur due to sequence similarity, and unlike the immunized antigen, this involves the apparent epitope or binding motif of the antibody mixture. Thirdly, carry-over peptides may remain detectable in the sample, solely due to their persisting massive presence, despite multiple washing steps and meticulous care on the part of the lab operator. While carry-over is merely noise, it is worth quantifying the variation caused by polyclonality and the apparent epitope.

Using recombinant motif-specific antibodies (GPS/CIMS-binders) which also enrich

classes of peptides, Olsson *et al.* (2012b) observed that these binders were markedly promiscuous. They compared sequence data obtained from tandem mass spectrometry (LC-ESI-LTQ Orbitrap) with structural information. Their findings showed that the antibodies not only captured peptides containing the targeted epitope but also variations of it. Thus, the enriched peptides revealed binding motifs. Different amino acid side chains are known to interact with the antibody binding site at individual positions. However, such in-depth characterizations of epitope motifs analysis call for extensive experiments.

A common approach to elucidate a detailed linear epitope is to conduct an interaction analysis between an antibody and peptide libraries. These libraries comprise synthetic peptides which have at least one modified position (Houghten *et al.*, 1991). For instance, if an antibody is raised against the C-terminal sequence LGYR, the peptide library for an epitope would consist of XGYR, LXYR, LGXR and LGYX-peptides, with the X representing all 20 amino acids. Instead of adding a single amino acid at step n, all 20 are added in equal amounts. This leads to a library of 80 different peptides for a four-amino-acid epitope. The synthesis and subsequent measurement of these peptides are significant cost drivers in this phase of antibody development. Moreover, the quality control for such a binder would be more expensive than its generation. Our study aimed at devising a simpler, cost-effective method entailing less effort than that required in the labour-intensive peptide library approach. We present an algorithm for calculating the detailed epitope using MS data from an immunoaffinity-MS experiment.

Peptide mass fingerprinting is a well established method in proteomics (Yates *et al.*, 1993; Mann *et al.*, 1993; Henzel *et al.*, 1993; Pappin *et al.*, 1993; James *et al.*, 1993), on which we based the TXP-TEA (Terminal sequence Enrichment Analysis) algorithm. The original technique involves identifying an isolated and subsequently digested protein by the characteristic pattern in the peptide mass spectrum. TXP-TEA performs searches for patterns related to sets of peptides sharing a specific terminal amino acid sequence, instead of spectral patterns associated with a specific protein. The method compares the masses observed in the measurement with a theoretical spectrum from a database. The MATERICS algorithm we describe here merges the results of TXP-TEA to the antibody binding motif.

6.1 Terminal sequence enrichment

When an antibody enriches peptides with a common sequence, it can be anticipated that the mass spectrometer will detect more matching signals. The TXP-TEA algorithm scores signals from an observed peak list, based on a scoring table which indicates the probability of observing the same signal in a random peak under the same conditions. The less likely such a signal is, the more probable it is that the observed peak list is not random. The software reports every sequence and score that is found.

A peak list is a set of mass-to-charge/intensity pairs (mz_1, i_1) , .. , (mz_n, i_n), obtained from a mass spectrum by further signal processing (noise filtering, peak-picking, nor-

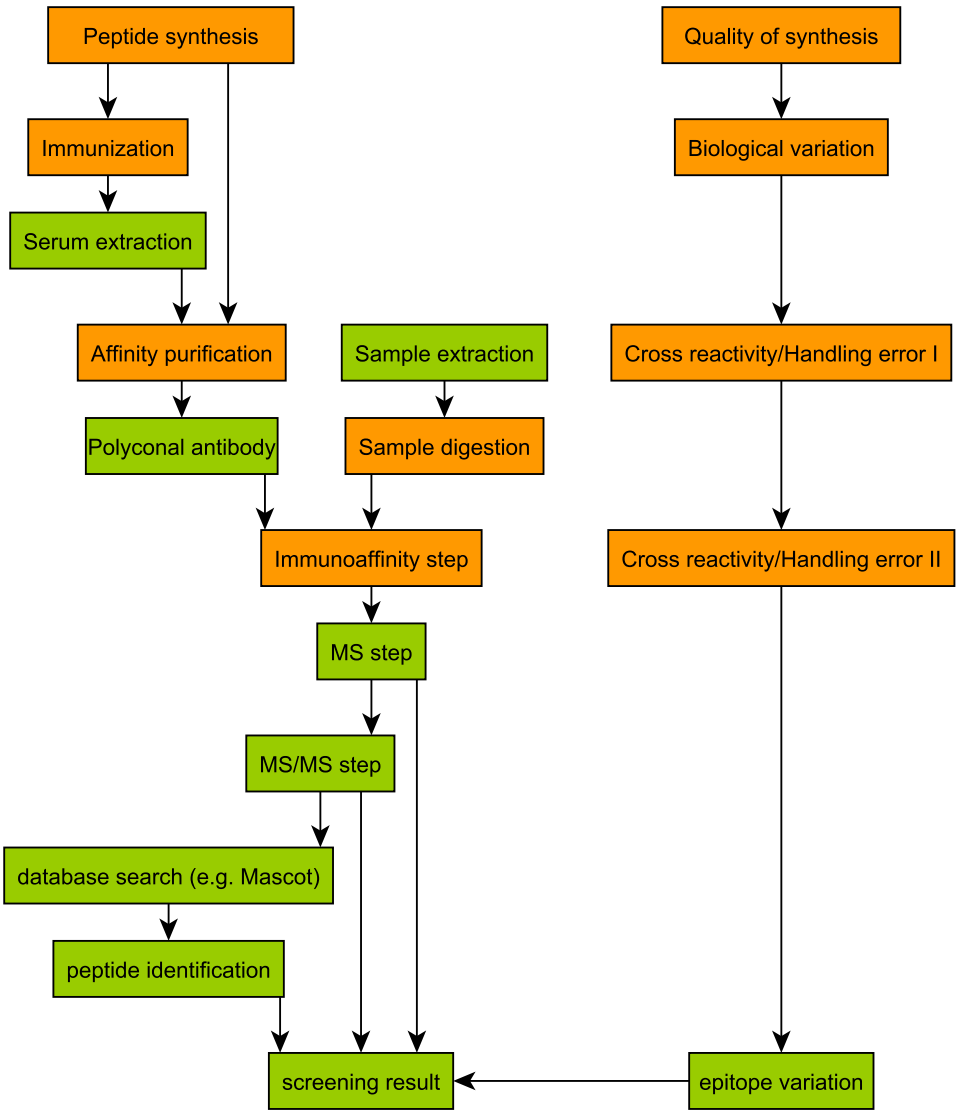


Figure 6.1: This flow diagram outlines the main source of epitope variation in the final screening result

malization, etc.). The dominant ion species produced in MALDI mass spectrometry has the charge $+1[M+H]^+$. The mz-values are $\frac{m+M_h}{1}$ and thus equal in value, but not in dimension, to the molecule plus a proton mass. By subtracting the proton mass $m_i = mz_i - M_H$, the mz-values $mz_1, ..., mz_n$ are transformed into mass values $m_1, ..., m_n$.

The sequence database contains information about peptides that can be expected in samples of a given species. Protein sequence databases, such as UniProt (Wu *et al.*,

2006) and proteotypic peptides databases such as the Global Proteome Machine (Fenyo *et al.*, 2010) or the Peptide Atlas (Deutsch *et al.*, 2008) are well-established data repositories. Proteotypic peptide databases are particularly valuable because each sequence had already been observed in mass spectrometry-based experiments. In our algorithm, a database D is a set of peptide sequences $\{p_1, ..., p_j\}$.

Using database D, TXP-TEA first makes a comprehensive list of peptides with theoretical masses, whose observed mass in M_{obs} matches a mass in the peak list by a predefined error threshold. Common MALDI mass spectrometers operate at a medium resolution and mass errors below 30 parts per million are normal. Due to the limited resolution and isobaric peptides, each signal in a spectrum can originate from different peptides. The set of selected peptides in range is:

$$S(D, M_{obs}, \varepsilon_{tol}) = \{p_i | \frac{M_{obs} - M_{p_i}}{M_{p_i}} 10^6 | \leq \varepsilon_{tol}\} \tag{6.1}$$

The result is a set of peptides

$$D_M = \bigcup_{M_{obs} \in M} S(D, M_{obs}, \varepsilon_{tol}) \tag{6.2}$$

and finally a list of possible epitope candidates is generated. Each candidate epitope can explain signals in the spectrum, provided that the peptides matching that specific terminal sequence have been enriched. The number of matching peptides found in the full database search is also relevant because it defines the background probability. For example, if 5 masses in a 73-peak spectrum match peptides sharing the c-terminal sequence LGYR and 65 peptides in the full database terminate in LGYR, this event must be rated by estimating the probability of finding 5 (or more) out of 65 (or less) peptides which share the same c-terminal sequence of length 4 in a random 73-peak spectrum.

We define an enrichment event $E_\Phi(i, j)$ as: i matching signals out of j masses from the same epitope class, by applying the parameters $\Phi = (D, \varepsilon_{tol}, t, l, k)$ in a peak list of k masses. TXP-TEA estimates the likelihood of such enrichment events by sampling from random spectra. The parameters Φ of the sampling are: the terminus (c- or n-terminal) t, sequence length l, number of peaks k, mass error tolerance ε_{tol} and peptide database D. The number of masses in the peak list k is also a sampling parameter. Each setting of these parameters requires a dedicated sampling table. Sampling generates many random peak lists by randomly selecting theoretical masses from the peptide database.

The random peak lists are then processed as described above. Algorithm 6 counts the number of repeated enrichment events and generates a sampling table. A sampling table is an $n \times m$-matrix S_Φ. m is the size of the largest epitope class in the database. n is the largest number of peaks attributable to one epitope-class, observed in a random spectrum, during the sampling process. $S_\Phi(i, j)$ is the frequency of the event that i of j expected masses match, depending on the parameters Φ.

Table 6.1 is the result of a sampling run of 25,000 iterations for mass spectra of 100 peaks and a mass tolerance of 30 ppm in a consensus data peptide database ($\Phi =$

$(ConsensusDB, \varepsilon_{tol} = 30, t = C, l = 4, k = 100)$). For example: The event of observing 4 out of 17 peptides with a common terminus occurred 127 times in 25,000 random spectra. The sampling table is not a perfect lower triangular matrix because of the occurrence of overlapping peaks. If the mass of a peptide with a unique terminal sequence is in within close range of two masses in a random spectrum, the event-counter for $E_{\Phi}(2,1)$ increases (i.e. observed 18,373 times in sampling table 6.1).

Table 6.1: Sampling table S for the consensus peptide database, number of sampled random spectra 25,000, number of peaks 100, mass tolerance 30 ppm

	number of peptides observed in spectrum				
occurrence	1	2	3	4	5
1	3205722	18373	85	0	0
2	2111151	22101	182	0	0
3	1445403	21227	235	0	0
4	1189578	23363	350	5	0
5	1074187	25731	457	6	0
6	983959	27333	536	5	0
7	900073	28868	704	14	0
8	1009570	37708	1004	13	1
9	879334	36178	1009	21	1
10	858291	38574	1187	34	1
11	918326	46414	1495	48	1
12	931052	50009	1830	49	2
13	829150	49125	2086	69	0
14	762225	47894	2035	69	2
15	790782	53216	2445	75	2 0
16	888119	64125	3099	109	1
17	858695	66359	3463	127	7
18	692556	55687	2949	128	2
19	887858	77471	4556	198	2
20	796262	72015	4481	211	9
...	...	...	...	...	...

It is better to use standard settings to limit the available choices relating to the assumed error tolerance, background database, terminus, and sequence length. This limits the computational effort to a required minimum.

However the parameter k, number of peaks, varies from spectrum to spectrum. The sampling algorithm 6 solves this difficulty by incrementally updating the sampling table. It creates the sampling table for 73-peak spectra by adding a random peak to all 72-peak spectra, from the 72-peak sampling table in the previous step. This is considerably faster

Algorithm 6: The sampling algorithms result is a table S necessary to estimate the distribution of epitope detection events in random spectra of up to n peaks.

Input: parameters $\Phi = (D, \varepsilon_{tol}, t, l, k_{max})$, iterations n
Output: sampling table S
foreach $seq \in D$ **do**
  $term = getTerminalEpitope(seq, l, t)$;
  $tcount[term]++$ `// count background distribution`
end
$\Phi' = (D, \varepsilon_{tol}, t, l, 0)$;
$S_\Phi = emptymatrix$;
$j = 1$;
while $k \leq k_{max}$ **do** `// increase peak count until maximum`
  $\Phi' = (D, \varepsilon_{tol}, t, l, k)$;
  $S_{\Phi'} = S_\Phi$;
  $i = 0$;
  while $i \leq n$ **do** `// for each random spectrum`
    `/* select random peptide and others with similar mass */`
    $randompep = D[randomInteger(|D|)]$;
    $P = getPeptides(D, mass(randompep), \varepsilon_{tol})$;
    foreach $seq \in P$ **do**
      $term = getTerminalEpitope(seq, l, t)$;
      $count[i][term]++$;
      $T = T \cup term$;
    end
    foreach $term \in T$ **do**
      if $count[i][term] > 1$ **then**
        `/* terminus term has been observed one more time for the additional mass, therefore the previously counted event must be removed from the count. */`
        $S_{\Phi'}[count[i][term] - 1, tcount[term]]--$;
      end
      `/* increase the count for the event */`
      $S_{\Phi'}[count[i][term], tcount[term]]++$;
    end
    $i = i + 1$;
  end
  $\Phi = \Phi'$;
  $k = k + 1$;
end
return S

than analyzing 73 peaks from scratch.

TXP-TEA estimates the p-value $\hat{p}(E_\Phi(i,j))$ by dividing the count of the same and more extreme events by the total number of events:

$$\hat{p}(E_\Phi(i,j)) = \frac{\sum_{k\geq i}\sum_{l\geq j} S_\Phi(k,l)}{\sum_k \sum_l S_\Phi(k,l)}. \tag{6.3}$$

In general the p-value is the probability of observing a specific or more extreme event, given that the null hypothesis is true. If a rare event occurs, the estimated p-value is sufficiently small. The null hypothesis (Pharoah, 2007) can be rejected, because observing such data under this assumption is improbable. The null hypothesis in TXP-TEA is: 'The binder does not enrich any single terminal sequence in the sample.' A simple alternative hypothesis is: 'The binder enriches the terminal epitope LGYR in the sample.'

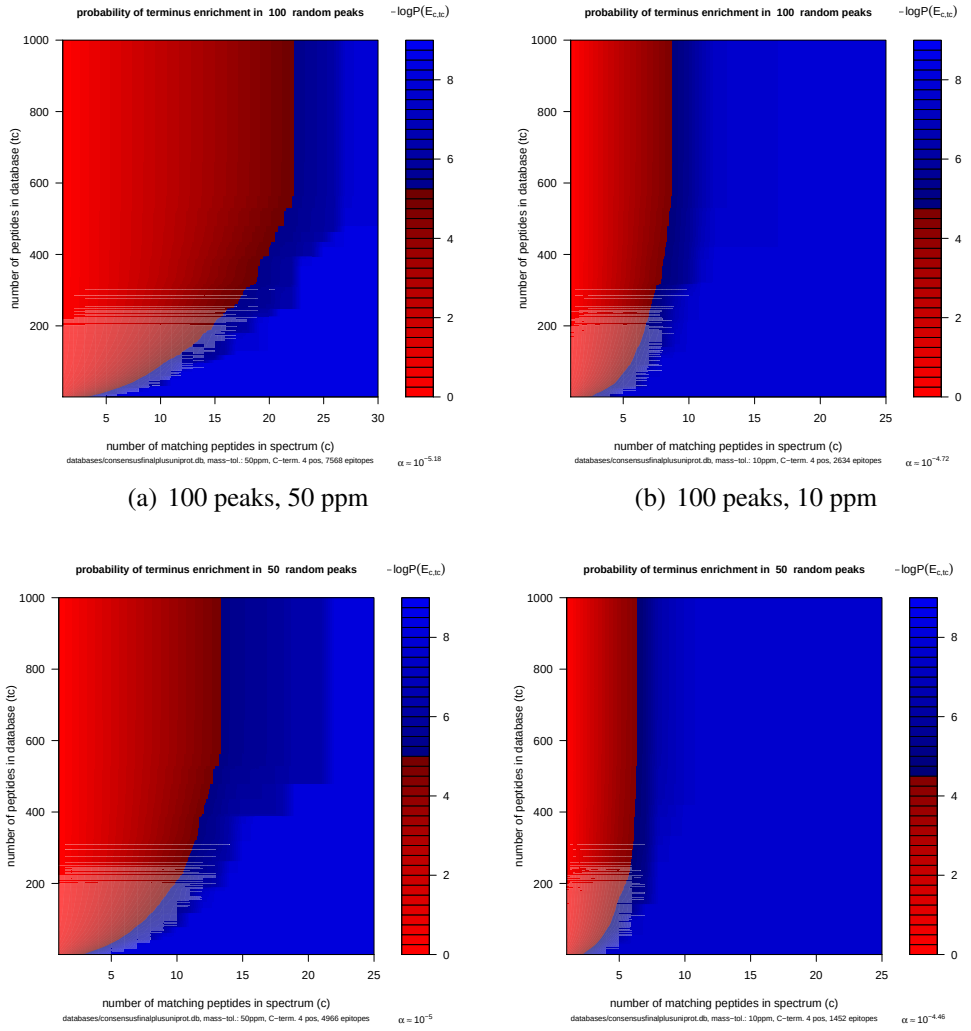
Mass accuracy and the number of peaks determine the number of different candidate epitopes to be analyzed by TXP-TEA in a search. Each candidate represents an alternative hypothesis. As these represent different hypotheses, it is important to correct for multiple testing. By assuming a significance level of $\alpha = 0.05$ and 3,000 different epitopes in a single search, $3,000 \times 0.05 = 150$ epitopes will have a significant p-value by chance, provided that the p-values follow a uniform distribution. Bonferroni correction adapts the significance level to $\alpha' = \frac{\alpha}{n}$, dividing by the number of candidate sequences. Finally TXP-TEA reports enrichment of a terminal epitope sequence if $\hat{p}(E_\Phi(i,j)) \leq \alpha'$.

Figure 6.2 visualizes p-values obtained from sampling tables generated by 25,000 random spectra from a background peptide database for *H. sapiens* (merged GPM, PeptideAtlas, *in-silico* tryptic digest UniProt). If mass accuracy is high, the number of peptides matching a mass will reduce. This also reduces matches of peptides with the same epitope to different peaks in the spectrum. This explains why higher mass accuracy allows TXP-TEA to detect significant enrichment events with low numbers of matching masses.

6.2 From sequences to complex epitopes

MS/MS experiments revealed that the enriched epitopes are more complex than the immunization antigen sequence. The antibody binds to variations of a main sequence, therefore with less affinity (Olsson *et al.*, 2012a,b). Variation often occurs mainly in one or two positions, while the other positions remain constant. These findings form the central idea of the MATERICS (mass-spectrometric analysis of terminal epitope enrichment in complex samples) algorithm, a novel approach to ascertain the motif rapidly and automatically.

Residue variation at key positions is a well-known concept which is applied by computational immunologists to MHC molecules, cell surface proteins similar to antibodies. The binding specificity of MHC-molecules is often charaterized by peptide motifs (Falk



(a) 100 peaks, 50 ppm

(b) 100 peaks, 10 ppm

(c) 50 peaks, 50 ppm

(d) 50 peaks, 10 ppm

Figure 6.2:
Visualization of scoring tables $\hat{p}(E_\phi(i,j))$ with different mass tolerances and peak numbers. The blue areas mark events that are considered to be statistically significant. The $\alpha = 0.05$ was Bonferroni-corrected by the average number of events which occurred during sampling: 50 peaks/50 ppm 4966 events, 50 peaks/10 ppm 1452 events.

and Rötzschke, 1993; Stern, 2007). A MHC class I peptide motif defines one or two internal anchor positions and an additional fixed position at the C-terminus. Each MHC allele has a characteristic motif (Sherman, 2006).

Regular expressions such as LG[AYL]R describe such sequence motifs. This expres-

sion matches LGAR, LGYR and LGLR. This form gives no information about which of the three different amino acids is more probable at the variable position. Another common way to define a peptide motif is to use a position weight matrix (PWM). For a given alignment of amino acid sequences, these matrices assign a probability p_{ij} to each amino acid A_i for a specific sequence position s_j in a sequence s.

Each motif represents a trade-off between sensitivity, specificity and model complexity. A PWM, which assigns equal probability to all amino acids at all positions, obviously matches all peaks in the observed set. Such a motif lacks any useful information, whereas a motif which assigns a probability of 1.0 to one specific amino acid at every position as well as matches a peptide for all peaks in the spectrum is a remarkable finding.

Information content, IC, derived from Shannon Entropy, is a complexity measure for PWMs (Lund *et al.*, 2005). $H(X)$ denotes a measure of uncertainty

$$H(X) = \sum_{j}^{n} p(x_i) \cdot \log(p_(x_i)) \tag{6.4}$$

to a discrete random variable X with n different outcomes. $H(X)$ is minimal if all events are equally probable and the uncertainty is thus maximal. If one event occurs, the uncertainty is minimal, and the entropy term will maximize and approach 0. The complexity measure for a PWM P

$$IC(P) = -\sum_{i}^{L}\sum_{j}^{N} p_{ij} \cdot \log(p_{ij}) \tag{6.5}$$

follows, by applying the entropy score to each position and calculating the sum.

The space of possible PWM epitopes is

$$E = \{x \in [\mathbb{R}^{20}_{[0,1]} | \sum x_i = 1.0]^l\} \tag{6.6}$$

While that set is not countable, the set of wildcards $W = \mathscr{P}(A)^l$ is enumerable. A denominates the set of amino acids. With $|\mathscr{P}(A)| = 2^{20} = 1,048,576$ the number of possible wildcards of length 4 is $1,048,576^4 \approx 1.209 \times 10^{24}$. MATERICS can appropriately limit a search from start, or abort it timely during the process.

In the first step of MATERICS, TXP-TEA generates the ranking of enriched sequences. MATERICS scores all motifs with one unspecific position (?XXX,X?XX,XX?X,XXX?) by combining the p-values of all the matching terminal sequences using Fisher's Method. This method is applied in meta-analysis statistics to accumulate evidence from different studies. The sum ρ of logarithms of p-values

$$\rho = -2\sum_{i=1}^{k} \log_e(p_i) \tag{6.7}$$

from independent tests follows a χ^2 distribution. The complementary χ^2 cumulative distribution function with $2k$ degrees of freedom is

$$pval_c(\rho,k) = 1 - \frac{\gamma(\frac{2k}{2}, \frac{\rho}{2})}{\Gamma(\frac{2k}{2})} \tag{6.8}$$

$$= 1 - \frac{\gamma(k, \frac{\rho}{2})}{\Gamma(k)} \tag{6.9}$$

$$= 1 - \frac{\gamma(k, \frac{\rho}{2})}{(k-1)!} \tag{6.10}$$

$$\tag{6.11}$$

which gives the combined p-value. $\gamma(k,\rho)$ is the lower incomplete gamma function. The list L will include epitopes for further processing, if the combined p-value is lower than 0.1, divided by the total number of motifs scored. This step removes candidate epitopes from the search, which, even in the context of similar epitopes, will not contribute to a relevant motif.

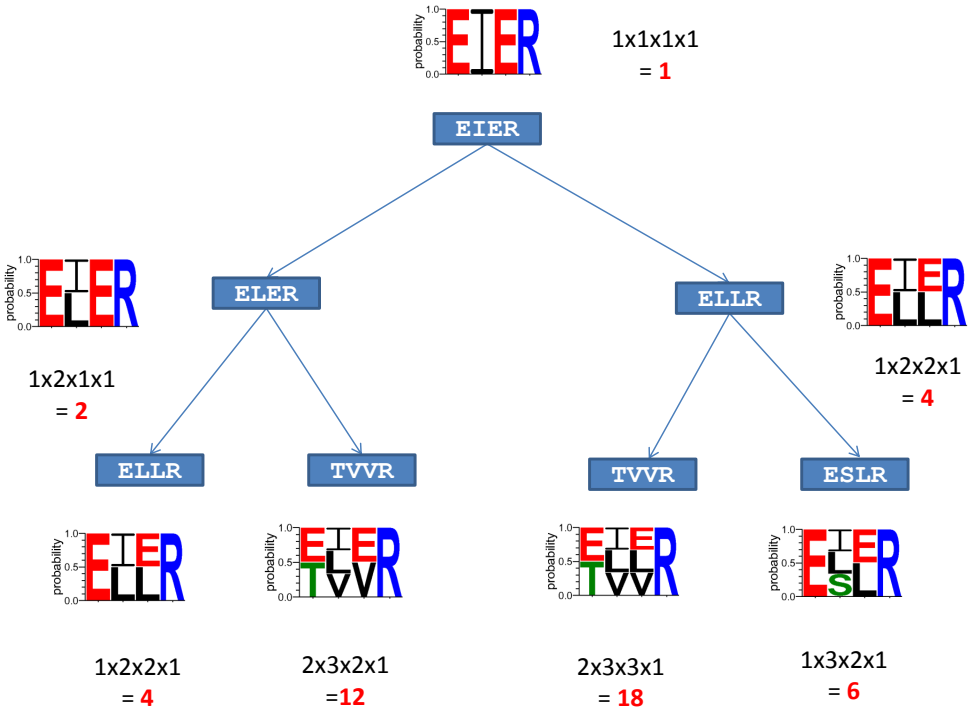


Figure 6.3: Tree representation of the recursive enumeration

In the next step, MATERICS combines sequences using recursive enumeration (see figure 6.3) and calculates the complexity of each motif. The recursion stops if the com-

plexity exceeds a certain limit. During recursion, the method updates the coverage score using Fisher's Method. The recursive loop calculates the sum of logarithms incrementally.

MATERICS uses a suitable property of Fisher's method as a recursion bound. Assuming the method is applied to a vector of sorted p-values

$$(p_1, p_2, \ldots, p_j, \ldots, p_n)$$

and that the combined p-value $pval_c(\rho_j, j)$ increases

$$pval_c(\rho_{j-1}, (j-1)) \leq pval_c(\rho_j, j)$$

at one point, it follows that by adding further tests (with p-values $> p_j$) there will be no improvement in the combined p-value. This means that $pval_c$ has a well-defined global optimum. The algorithm can terminate recursion once the combined p-value starts to increase. Other termination criteria include a complexity measure exceeding a predefined limit and sequence p-values lower than 0.05. These bounds ensure reasonably fast processing. A motif prediction takes from few seconds up to a minute on a single AMD Phenom X6 core clocked at 3.3 Ghz.

Algorithm 7: Recursive enumeration in the MATERICS algorithm: The function traverses the tree shown in figure 6.3, and combines epitopes from the result list L until the p-value combined by Fishers method starts increasing. All results are stored in a Pareto front data structure, which keeps only non-dominated (complexity and/or p-value) results.

enum(L,i,c_{max}, c', pvals,E, ρ,d)
if $((c' < c_{max}))$ **then**
  for $j \leftarrow i_l$ **to** $|L|$ **do**
    $E' \leftarrow$addToEpitope(E, L_j) ;
    $c' \leftarrow -IC(E')$; `// motif complexity`
    $\rho \leftarrow \rho + log(pvals[L_j])$;
    $pval' \leftarrow 1 - \frac{\gamma(\frac{2d}{2}, \frac{-2 sumlog pval}{2})}{\Gamma(\frac{2d}{2})}$; `// motif score`
    updateParetoFront(E',$-c'$,$score'$) ; `// add to pareto front`
    if $((pval' < old pval) \wedge (pvals[L_j] < 0.05))$ **then**
      `/* call recursively if combined pvalue decreases */`
      enum(L,j,c_{max},c',pvals,E', $pval'$,ρ,$d+1$) ;
    end
  end
end

The algorithm will include a motif in the solution set M if - and only if - no motif with lower complexity and a higher score was found. It reports motifs representing good

compromises between complexity and p-value in the final output. This concept is known as Pareto optimality in the field of multi-objective optimization. A motif M_1 with a complexity score $IC(M_1)$ and a p-value $pval(M_1)$ will dominate a second motif M_2 - if and only if - $IC(M_1) > IC(M_2)$ and $pval(M_1) < pval(M_2)$. It reflects the process of model building by the expert, which also weighs model complexity, against how much the model can explain.

The user interface presents each solution in M, enabling the user to make an informed judgement. Whereas during enumeration of the motif subspace equal probabilities were assumed at each position, the candidate PWMs are refined afterwards. The set of matching sequences denominates S and f_{aj} is the frequency of amino acid a at position j:

$$p_{aj} = \frac{f_{aj}}{|S|} \tag{6.12}$$

The Matthews correlation coefficient (Matthews, 1975) is a good indicator of the robustness of the identified models. It measures the suitability of a classification model by the number of true positive, true negative, false positive and false negative predictions. It is impossible to compress true/false positives/negatives in a single number without information loss. However the coefficient is well established, and is particularly useful in dealing with heavily imbalanced two-class classification problems.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{6.13}$$

MATERICS deals with the MCC as an alternative measure for robustness. This convention for model fitness evaluation was applied to PWM models for TXP-epitopes enrichment in mass spectra:

	presence predicted	absence predicted
observed	TP	FN
not observed	FP	TN

The number of true negatives (absence predicted, not observed) is estimated by the number of distinguishable peaks in the range from 800-2500 Da and the specified error tolerance.

Our work compares the novel algorithms to peptide libraries for motif elucidation. Experiments included measurements before and after immunoprecipitation (IP) took place. The difference between measured signal intensities is the effect of the antibody specificity at the given position.

The pre-IP measurements account for sequence-specific ionization characteristics in the mass spectrometer as well as differences in the outcome of the peptide synthesis.

The signal intensities were used to normalize the data from the post-IP experiment. The normalization coefficient for a specific peptide found in the library is:

$$F(a) = \frac{A(m_a)}{\sum_{b \in D} A(m_b)} |D| \tag{6.14}$$

where m_a is the known mass of the library peptide with amino acid a at a variable position, $A(m_a)$ the peak area at the respective position, and D the set of peptides found in the library. This 'flight factor' reflects the ionization properties: thus even if all the synthetic peptides are equally abundant in the library, the measured signal intensities will differ by orders of magnitude. Peptide prevalence $P(A)$ can be calculated by using $F(A)$ and the post-IP data

$$P(a) = \frac{A(M_a) F(a)}{\sum_{b \in D} A(M_b) F(b)} \tag{6.15}$$

For amino acid exchanges which are not distinguishable by the resolution of the mass spectrometer, the probability is shared in the inferred motif. Therefore leucine and isoleucine will always appear as equally probable in motifs constructed by the peptide library approach. MATERICS is able to discriminate isobaric exchanges, because predictions are based on sequence database information. For example if MATERICS detects only enriched peptide sequences with leucine, but none with isoleucine at the respective position, this will be reflected in the motif.

6.3 Experiments

Three different antibodies generated against the 4mer peptides AMTR, LGYR and EIER were analyzed using the MATERICS workflow. Four peptide libraries per binder were used, one for each amino acid position. Detailed lab methods can be found in Planatscher *et al.* (2010).

In our study, we used 5 μg antibody and, accordingly, 25 μL protein G-coated magnetic beads. Immunoprecipitations were carried out thrice per cell line.

The spots were analysed using an Ultraflex III MALDI-TOF/TOF mass spectrometer (Bruker Daltonics) in positive ion reflectron mode. The deflector cutoff was set up to 500 Da. Mass calibration was performed by using pre-spotted calibrants on PAC II 384 plates. The detection mass range was set from 600 to 4000 Da. The laser power was adjusted manually. The signal intensities of 2000 shots were accumulated per spot. Peaks were annotated automatically with a signal-to-noise threshold of 3 and a mass range from 750 to 4000 Da using flexAnalysis 3.0 software (Bruker Daltonics).

Each replicate was analyzed with MATERICS by means of a unified peptide database containing the peptide sequences from GPM (Beavis, 2006), PeptideAtlas (Deutsch *et al.*, 2008), and peptide identification from the Human Plasma Proteome Project (Omenn *et al.*, 2006), with 30 ppm error tolerance, and maximum complexity 2.0.

In addition benchmark experiments were performed using artificial data to test the influence of noise on the performance of MATERICS.

6.4 Results

Table 6.3 summarizes the results for the positional peptide library experiments as well as those for the MATERICS algorithm in the cell line experiments. The library results for the 'AMTR' antibody indicate minor variation at the first and third positions and high variation at the second position. MATERICS predictions detected a high degree of variation at the second position.

Table 6.2: Summary of the MCC scores observed on the highest ranked (p-value sorted) results for three binders/tissues in three IP replicates. First column is the average MCC and the number of technical replicates/spectra with a prediction result, the second column contains the standard MCC deviation.

	AMTR		LGYR		EIER	
HELA	0.202 (4)	0.018	0.16 (4)	0.011	0.107 (3)	0.018
HELA	**0.216** (4)	0.016	0.189 (4)	0.036	0.127 (4)	0.006
HELA	0.182 (3)	0.028	0.171 (4)	0.030	0.124 (2)	0.000
HEK	0.19 (4)	0.036	0.178 (4)	0.046	(0)	
HEK	0.18 (4)	0.040	0.212 (4)	0.034	0.096 (1)	0.000
HEK	0.173 (4)	0.026	**0.222** (4)	0.034	(0)	
A357	0.208 (4)	0.025	0.192 (4)	0.035	**0.167** (4)	0.018
A357	0.154 (4)	0.040	0.178 (4)	0.009	0.149 (4)	0.042
A357	0.19 (2)	0.018	0.193 (3)	0.043	0.137 (1)	0.000

For the 'LGYR'-antibody, the library experiments revealed low variations at the first, third and fourth positions and high variations at the second position. Since isoleucine (I) and leucine (L) are isobaric and therefore indistinguishable by mass spectrometry, the library method assigns equal probability to both amino acids at the first position. MATERICS predictions suggest a variation at the second position in the enriched peptide samples precipitated from the digested HELA and A397 cell line. Our results do not indicate that isoleucine is bound at the first position, the prediction only showed leucine. When applied to HEK293 immunoprecipitate mass spectra, the algorithm suggests an alternative binding of valin (V) at the first position.

The 'EIER'-antibody showed a high preferential binding to its antigen sequence. Sequences which vary at the first and third position have low binding affinity. The prediction confirmed the low variability at the first position in the HELA and A397 immunoprecipitate. MATERICS detected sequence variability at the second position in all samples.

It also predicted equal probabilities to leucine and isoleucine, and detected binding to methionine at the second position in the HEK293 and A397 cell line. The algorithm predicted variability at the third position of the motif in the HELA sample. Table 6.2 displays the average performance of the top-ranked solutions measured by MCC. Note that solutions ranked lower by p-value can have better MCC scores.

Table 6.4 compares the models obtained by the spectra of different immunoprecipitates and technical replicates of the same sample.

Table 6.3: Comparison of the peptide library results to the best (smallest combined p-value) results obtained by a run of MATERICS with 30 ppm error tolerance, maximum complexity 2.0 using the consensus peptide database

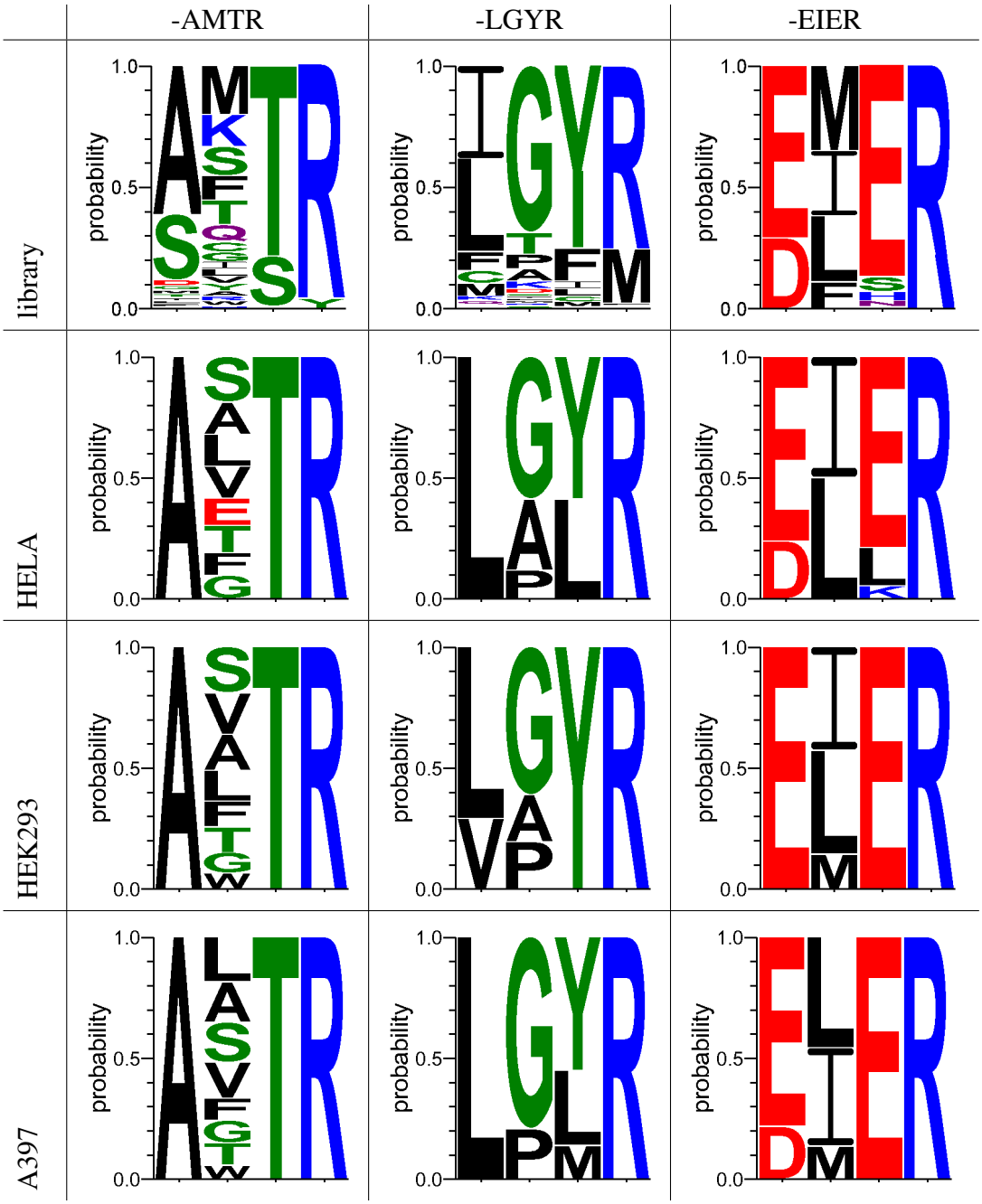
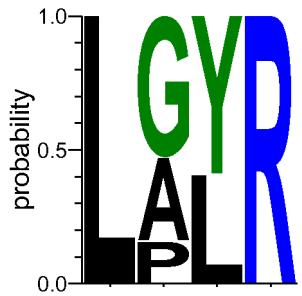
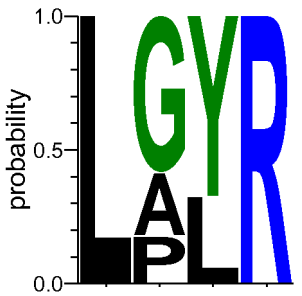
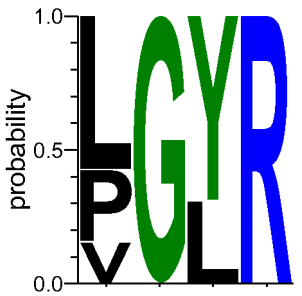
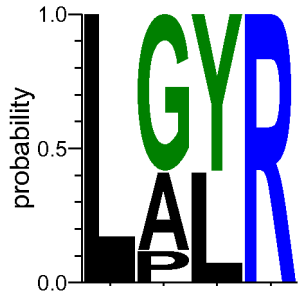
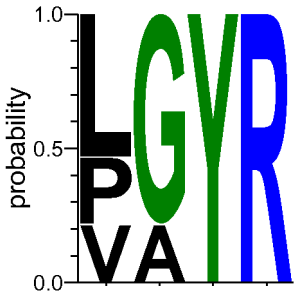
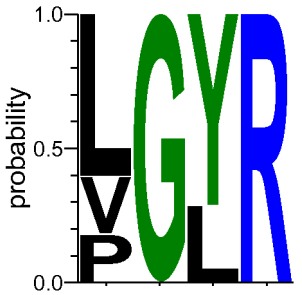
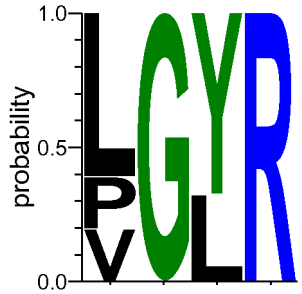
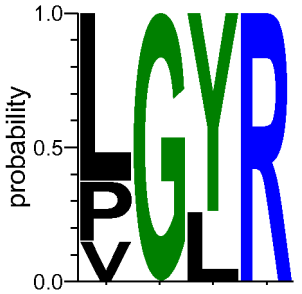
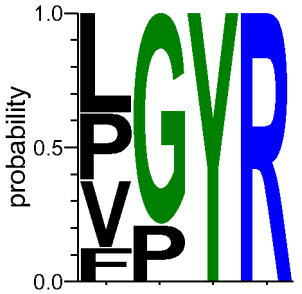
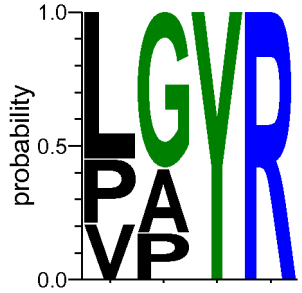
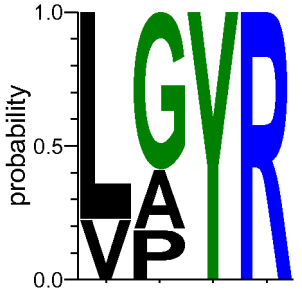
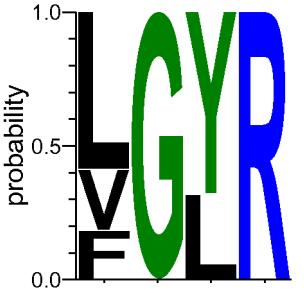


Table 6.4: Results for all biological and technical replicates for anti-LGYR serum in combination with digested HELA-cell-lysates

	IP1	IP2	IP3
rep 1			
rep 2			
rep 3			
rep 4			

6.5 In Silico Benchmarks

The benchmark tests the prediction performance of MATERICS under the influence of noise. We used artificial mass lists, generated by mixing sequences matching a predefined motif and random sequences from a database, as input. MATERICS was used to infer the respective motif back from the mass lists. By varying the size of the fraction of motif-matching sequences in the artificial list, it was possible to estimate the influence of noise on the prediction performance.

The tests were performed by generating artificial lists of 100 mass peaks. Each artificial peak list contained 4, 9, 14, 19, 24 or 29 peaks related to sequences containing a sequence matching an arbitrary 'true motif'. The masses were randomly drawn from the unified database (see article). Also Gaussian noise (10, 25 and 50 ppm) was added to the peak list. We tested if MATERICS predicted motifs which, in combination with the database, enabled to re-identify the associated 'true' peptide sequences.

1. Define true epitope
2. Select n related peptides containing true epitope from a database
3. Select (100 - n) other peptides from the database
4. Save peptide list A
5. Save peak list (protonated masses of peptides)
6. Add gaussian noise to each mass
7. Use MATERICS to predict epitope from artificial peak list
8. Find peptide sequences in the database which correspond to mass in the peaklist and the motif(s) predicted by MATERICS
9. Compare the predicted peptides to peptide list A

A benchmark script analyzes the MATERICS prediction and calculates scores using the known sequences (list A). Each predicted motif is analyzed and scored. The overall score for a MATERICS prediction is measured by the area under the curve of the specificity/sensitivity pareto plot as shown in figure 1.

An AUC of 1.0 is achieved, only if MATERICS was able predict a motif that was 100 % specific and 100 % sensitive. This signifies that one motif in the results list identified all peptides belonging to the enriched group. An AUC of 0.5 can be achieved by guess.

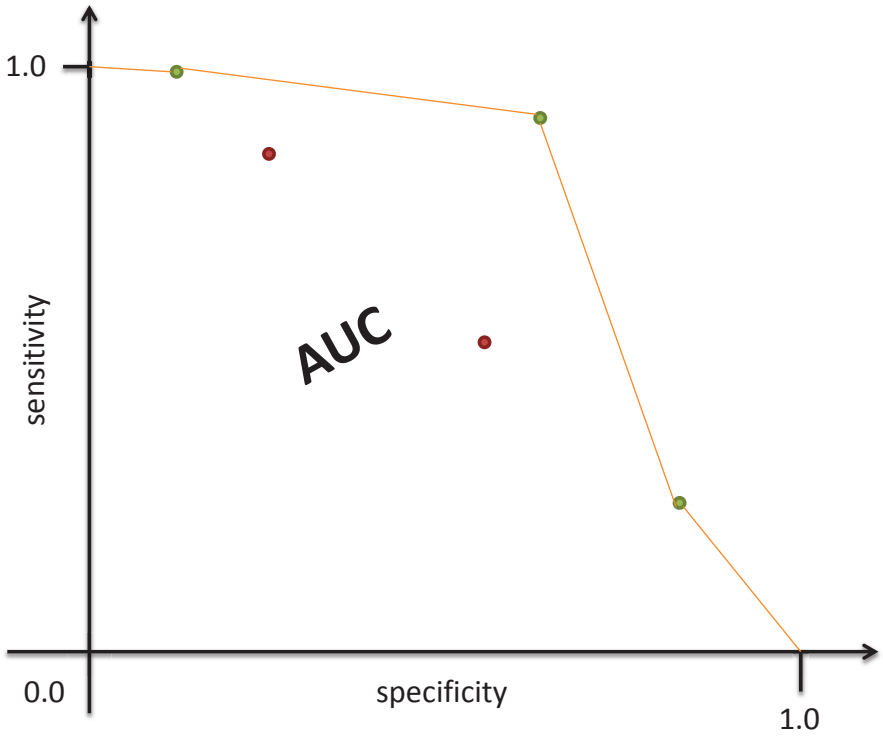


Figure 6.4: This sketch shows the pareto plot for MATERICS prediction results. The area under the pareto curve (AUC) is the overall score for the prediction.

6.5.1 Example

The following example should illustrate the scoring method.

In the example a peak list containing 14 peaks for peptide sequences containing the true motif 'LG[ALYES]R' was used. The output of the benchmark script lists true positive, false positive and false negatives sequence matches for all predicted motifs and prints the AUC score, summarizing MATERICS' overall performance:

```
Peptide matches for predicted motif LG[A|E|L|Y]R

true sequence sequence predicted by MATERICS
GMDYLGSR related to epitope, no prediction false negative
MLDNLGYR MLDNLGYR true positive
HLDFLDILLGAR HLDFLDILLGAR true positive
QCCDCCGLGLR QCCDCCGLGLR true positive
AQPWADFTLLGAR ALQGALMIYFYR false positive
SSTAMTVMADLGER SSTAMTVMADLGER true positive
```

```
GVGQADWTPDLGLR GVGQADWTPDLGLR true positive
VAAQQGFDLDLGYR VAAQQGFDLDLGYR true positive
FQNAYLELGGLGER FQNAYLELGGLGER true positive
HLMDPQVLEFLGSR related to epitope, no prediction false negative
VVFTCQATANPEILGYR LLDMELEMAFFVGPGNR false positive
PAQPESLCIVEMGGTEKQDELGER PAQPESLCIVEMGGTEKQDELGER true positive
EQPLDEELKDAFQNAYLELGGLGER EQPLDEELKDAFQNAYLELGGLGER true positive
EQPLDEEMKEAFQNAYLELGGLGER EQPLDEEMKEAFQNAYLELGGLGER true positive
GRSSLSLAKSVSTTNIAGHFNDESPLGLR GRSSLSLAKSVSTTNIAGHFNDESPLGLR true positive
LPWVCEEGAGIPTVLQGHIDCGSLLGYR LPWVCEEGAGIPTVLQGHIDCGSLLGYR true positive

TP FP TN FN specificity sensitivity
12 2 84 2 0.977 0.857

Peptide matches for predicted motif LG[E|L|Y]R

true sequence sequence predicted by MATERICS
GMDYLGSR related to epitope, no prediction false negative
MLDNLGYR MLDNLGYR true positive
HLDFLDILLGAR related to epitope, no prediction false negative
QCCDCCGLGLR QCCDCCGLGLR true positive
SSTAMTVMADLGER SSTAMTVMADLGER true positive
GVGQADWTPDLGLR GVGQADWTPDLGLR true positive
VAAQQGFDLDLGYR VAAQQGFDLDLGYR true positive
FQNAYLELGGLGER FQNAYLELGGLGER true positive
HLMDPQVLEFLGSR related to epitope, no prediction false negative
VVFTCQATANPEILGYR LLDMELEMAFFVGPGNR false positive
PAQPESLCIVEMGGTEKQDELGER PAQPESLCIVEMGGTEKQDELGER true positive
EQPLDEELKDAFQNAYLELGGLGER EQPLDEELKDAFQNAYLELGGLGER true positive
EQPLDEEMKEAFQNAYLELGGLGER EQPLDEEMKEAFQNAYLELGGLGER true positive
GRSSLSLAKSVSTTNIAGHFNDESPLGLR GRSSLSLAKSVSTTNIAGHFNDESPLGLR true positive
LPWVCEEGAGIPTVLQGHIDCGSLLGYR LPWVCEEGAGIPTVLQGHIDCGSLLGYR true positive

TP FP TN FN specificity sensitivity
11 1 85 3 0.988 0.786

Peptide matches for predicted motif LG[E|Y]R

true sequence sequence predicted by MATERICS
GMDYLGSR related to epitope, no prediction false negative
MLDNLGYR MLDNLGYR true positive
HLDFLDILLGAR related to epitope, no prediction false negative
QCCDCCGLGLR related to epitope, no prediction false negative
SSTAMTVMADLGER SSTAMTVMADLGER true positive
GVGQADWTPDLGLR related to epitope, no prediction false negative
VAAQQGFDLDLGYR VAAQQGFDLDLGYR true positive
FQNAYLELGGLGER FQNAYLELGGLGER true positive
HLMDPQVLEFLGSR related to epitope, no prediction false negative
VVFTCQATANPEILGYR LLDMELEMAFFVGPGNR false positive
PAQPESLCIVEMGGTEKQDELGER PAQPESLCIVEMGGTEKQDELGER true positive
```

```
EQPLDEELKDAFQNAYLELGGLGER EQPLDEELKDAFQNAYLELGGLGER true positive
EQPLDEEMKEAFQNAYLELGGLGER EQPLDEEMKEAFQNAYLELGGLGER true positive
GRSSLSLAKSVSTTNIAGHFNDESPLGLR related to epitope, no prediction false negative
LPWVCEEGAGIPTVLQGHIDCGSLLGYR LPWVCEEGAGIPTVLQGHIDCGSLLGYR true positive

TP FP TN FN specificity sensitivity
8 1 85 6 0.988 0.571

Peptide matches for predicted motif LGER

true sequence sequence predicted by MATERICS
GMDYLGSR related to epitope, no prediction false negative
MLDNLGYR related to epitope, no prediction false negative
HLDFLDILLGAR related to epitope, no prediction false negative
QCCDCCGLGLR related to epitope, no prediction false negative
SSTAMTVMADLGER SSTAMTVMADLGER true positive
GVGQADWTPDLGLR related to epitope, no prediction false negative
VAAQQGFDLDLGYR related to epitope, no prediction false negative
FQNAYLELGGLGER FQNAYLELGGLGER true positive
HLMDPQVLEFLGSR related to epitope, no prediction false negative
PAQPESLCIVEMGGTEKQDELGER PAQPESLCIVEMGGTEKQDELGER true positive
EQPLDEELKDAFQNAYLELGGLGER EQPLDEELKDAFQNAYLELGGLGER true positive
EQPLDEEMKEAFQNAYLELGGLGER EQPLDEEMKEAFQNAYLELGGLGER true positive
GRSSLSLAKSVSTTNIAGHFNDESPLGLR related to epitope, no prediction false negative
LPWVCEEGAGIPTVLQGHIDCGSLLGYR related to epitope, no prediction false negative

TP FP TN FN specificity sensitivity
5 0 86 9 1.000 0.357

Overall MATERICS prediction score:
epibench2_9_15_10.peaklist 0.923172757475083 AUC
```

Sensitivity and specificity are for all predicted motifs are summarized in table 6.5.

predicted motif	specificity	sensitivity
LG[AELY]R	0.977	0.857
LG[ELY]R	0.988	0.786
LG[EY]R	0.988	0.571
LGER	1.000	0.357

Table 6.5: Summary results for all predicted motifs on an artificial peak list for the true motif LG[ALYES]R

LG[AELY]R, LG[ELY]R and LGER form the pareto front.The AUC is calculated from the sensitivity and specificity values by adding rectangular and triangular areas (see figure 6.4)) :

```
1.000 * 0.357 +
0.988 * (0.786 - 0.357) +
0.977 * (0.857 - 0.786) +
(1.000 - 0.988) * 1 / 2 +
(0.988 - 0.977) * (0.786 - 0.357) / 2 +
(0.977 - 0.000) * (1.000 - 0.857) / 2 = 0.9231
```

Three c-terminal motifs were used for generating artificial data: LGYR, LG[ALYES]R and F[GT][WE]K. 10 artificial peak lists were generated for each motif, number of related peaks and noise setting. All peak lists were analyzed in MATERICS and scored using the benchmark script.

The boxplots (figure 6.5, 6.6 and 6.7) summarize the overall MATERICS benchmark results for the tested motifs. Of course prediction is easier if many peaks are attributable to the motif. However, results show that prediction is feasible even if less than 10 % of the peaks are related to motif enrichment (see figures 6.5 and 6.6). If a larger fraction (20%) of the signal is related to the motif, MATERICS almost always achieves more than 0.95 AUC. The method is more robust if signal noise is low. The prediction was most stable if only a noise level of 10 ppm was added to the artificial data. This is most visible for the F[GT][WE]K motif (figure 6.7). While prediction was robust at 10 ppm for spectra with only 9 related peaks, this was not possible at signal noise level of 25 and 50 ppm.

6.6 Discussion

The predictions by MATERICS closely reproduced the binding motifs identified by the positional peptide library experiments in different cell lines for the three antibodies. However the comparison also revealed some discrepancies between the control experiment and the MATERICS results. The motif for the anti-AMTR antibody found by the library experiment is more complex than the motifs predicted by MATERICS. This could be a consequence of complexity restrictions by the algorithm. The complexity scoring function penalizes an additional amino acid at the first position ([AS]?TR instead of A?TR) in terms of the degree of variability at the second position. The motif exceeds the complexity bound and is therefore not considered.

The binding property of the anti-EIER antibody was the most difficult to predict. MATERICS did not detect a significant enrichment in the pre-selection step for most spectra. Instead of predicting a wrong motif, it did not make any prediction at all, and instead reported that no significant enrichment was detectable. If the input data passed the pre-selection step, the predicted motifs were observable close to the results of the peptide libraries. On only one occasion did MATERICS make a wrong prediction: a comparison to the other technical replicates of the same immunoprecipitation ruled out this result. As shown in table 6.2, experiments for the anti-EIER antibody resulted in bad quality spectra for the HEK cell lysate, whereas immunoprecitipations from HELA- and A357-

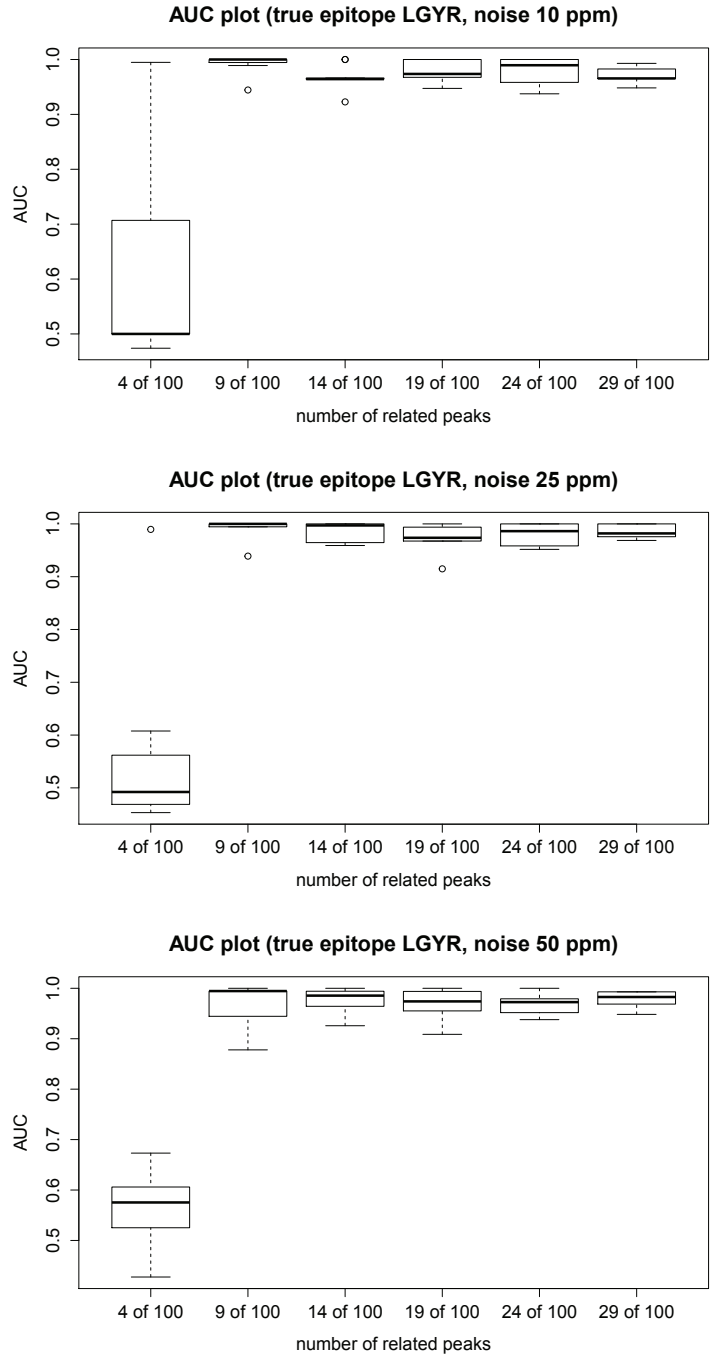


Figure 6.5: Box plots of the MATERICS AUC benchmark for artificial peaklists (n = 10) with the true epitope LGYR

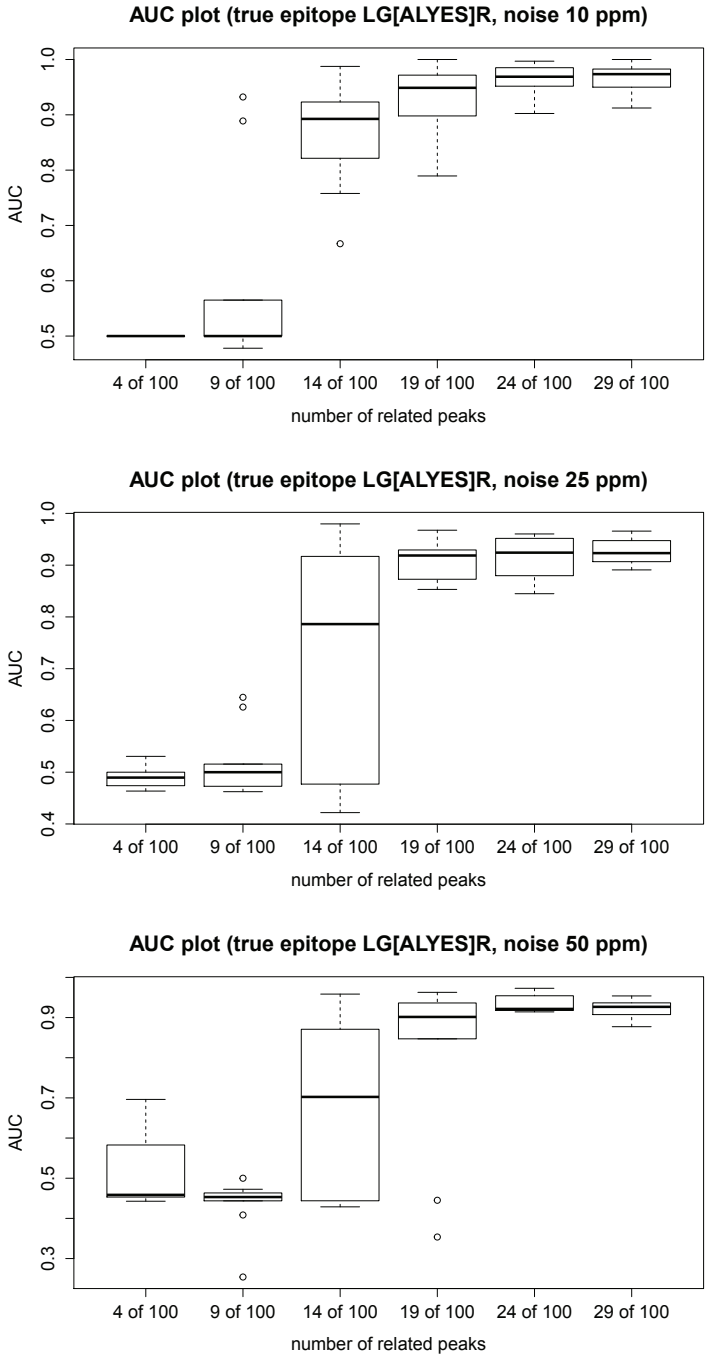


Figure 6.6: Box plots of the MATERICS AUC benchmark for artificial peaklists (n = 10) with the true epitope LG[ALYES]R

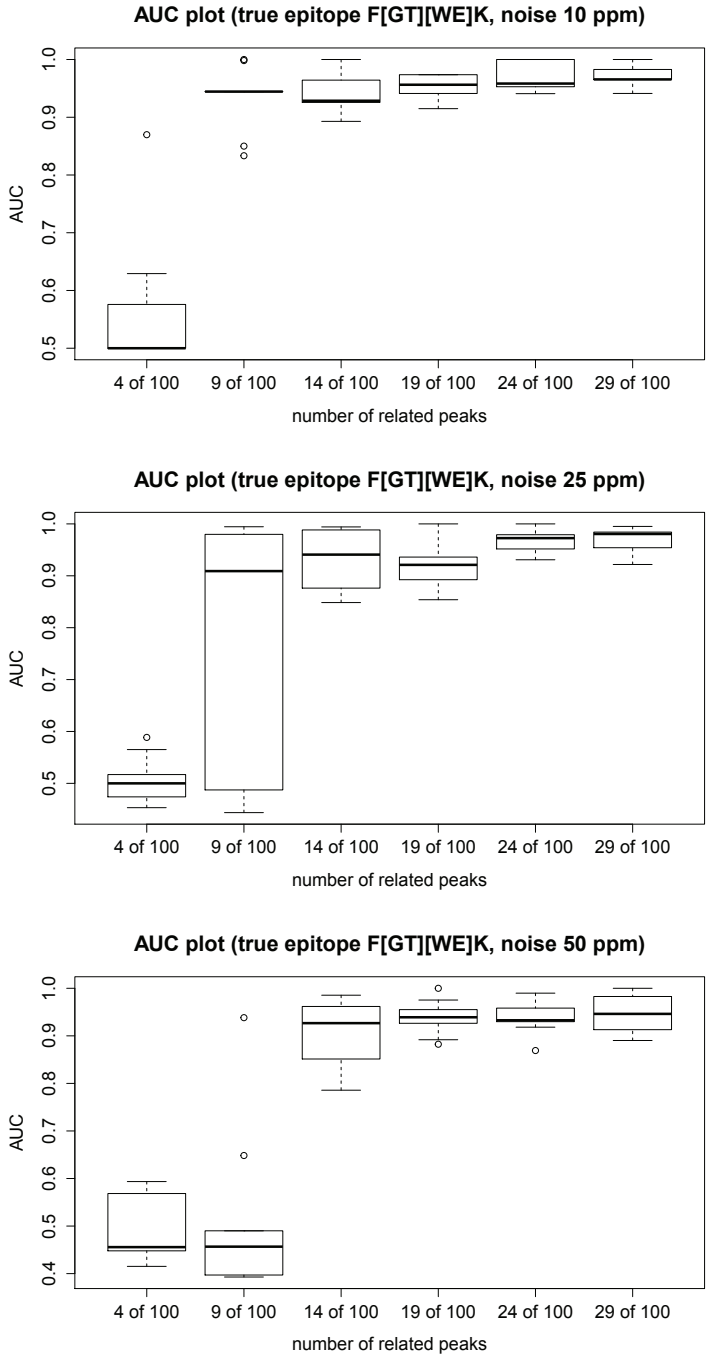


Figure 6.7: Box plots of the MATERICS AUC benchmark for artificial peaklists (n = 10) with the true epitope F[GT][WE]K

digests enabled motif prediction in many technical replicates. On the whole the HELA digest appears to be the most stable 'standard' sample for motif prediction pertaining to the three observed binders.

Although the results are obviously dependent on the observed epitope, it must be noted that using HELA cell lysate does not always produce the best prediction results. Some terminal epitopes are more abundant in some cell lines than others. Reports concluded that each cell line leads to the best MCC score for a given binder (results marked bold in table 6.2). There was adequate reproducibility within different immunoprecipitates and the technical replicates, at least for the stronger binders. The results obtained from different cell lines show a reasonable level of agreement. Apart from the minor variation described, this novel approach appears to work independently of the chosen line. This strengthens flexibility as cell lines in stock at the lab can be used to perform MATERICS experiments.

Benchmark experiments using artificial data showed, that MATERICS can make reliable predictions even if only a minor fraction of the signals found in a peak list is related to the enriched epitope.

6.7 Conclusion

Our experimental study shows that TXP-TEA and MATERICS are able to identify terminal binding motifs in immunoaffinity MS experiments. The motifs obtained closely resemble patterns found by using a peptide library approach. The described methods for motif elucidation lead to a substantial reduction in costs. In addition the novel method enables the weighting of isobaric variations in the binding motifs. These techniques might well lead to improved peptide identification algorithms, which exploit the existing data on potentially enriched sequences during the search process. Our findings are relevant to other fields of bio-medical research, such as in the identification of the binding properties of MHC molecules. Future versions of the algorithms will include options to identify internal epitopes and binding motifs as well as new ways to deal with post-translational modifications.

Chapter 7

A model for the distribution of short epitopes in proteomes

The purpose of this short chapter is to analyze the set of possible TXP epitopes in a proteome from a theoretical perspective. This makes it possible to estimate the potential of the TXP method, when applied on a set of peptides of arbitrary size, and the effect of choosing epitope length or different proteolytic agents. Also a closed-form statistical model for the enrichment of terminal sequences in peptide lists is introduced, and compared to the MATERICS approach described in Chapter 6.

7.1 Proteolytic cleavage of protein sequences

The 'bottom-up' proteomics approach always includes the enzymatic digestion of proteins prior to further analysis. Endopeptidases (e.g. trypsin, Lys-C, Arg-C,..) with a known specificity cleave proteins within the amino acid chain. These enzymes are usually found in the digestive system and, in higher mammals, are involved in blood clotting, the immune system and inflammation. The enzyme-specific cleavage pattern is mostly determined by the amino acids at the cleavage site. In order to build a reasonable hypothesis on which peptides can be expected in a digested complex protein sample derived from body-fluids or tissues, it is necessary to select a complete database of protein sequences and perform an in-silico digestion of the sequences.

The most commonly used enzyme is certainly trypsin. Trypsin catalyzes the cleavage of peptide bonds c-terminal from arginine and lysine. If proline is found in the n-terminal position the cleavage is strongly inhibited. There are several tools for the calculation of in-silico tryptic digest like Peptide Cutter and EMBOSS digest. In the further analysis we assume a full tryptic digest. Miscleavage (unexpected cleavage) and missed cleavage events are dismissed. While such events are almost always observed they are considered to be non-reproducible exceptions. Resulting peptides are therefore not suited as targets for quantitative analysis.

7.2 Epitope statistics

How many different epitopes should be expected in an average proteome? And how often does an epitope repeat? Answers to these questions will be found in this section.

The total number of possible epitopes depends on the epitope length and the used protease. Naively the number of all possible sequences of length l is 20^l, since there are 20 different amino acids. If the probability for each amino acid at all positions is higher than 0, we expect to find all possible epitopes in a large body of sequences. However if a protease is used, the set of sequences changes according to the specificity of the protease. Trypsin cleaves c-terminal of L (lysine) or K (arginine), if no P (proline) follows. Thus the number of c-terminal epitopes is $18^{l-1}2$, or $18^{l-1}2+2(l-2)18^{l-3}2$ when considering the proline as an inhibitor of trypsin cleavage.

If m epitopes are possible, how many of them are to be found in a database of N different peptides? The probability $P(k|m,N)$ that exactly k different epitopes are observed, could be calculated using the binomial distribution, if each epitope would occur with the same probability. Thus the $\mathrm{P}(k|m,N)$ could be expressed as

$$\mathrm{P}(k|m,N) = \binom{m}{k}\left(\left(1-\frac{1}{n}\right)^m\right)^k\left(1-\left(1-\frac{1}{n}\right)^m\right)^{n-k} \tag{7.1}$$

and the expected number of different terminal epitopes could be estimated by

$$\mathrm{E}(n,m) = m\left(1-\left(1-\frac{1}{n}\right)^m\right) \tag{7.2}$$

The distributions of epitope frequencies in real datasets, as the UniProt reference proteomes, however do not seem to be binomial at all (see figure 7.1). This can be explained by the oversimplified assumption that the different terminal sequences have equal occurrence probability $1/n$. It is known, that some amino acids occur less frequently than others, so this can not be the case. Studies have shown that the amino acid frequencies are related to length of the bio-synthesis pathway and the number of synonymous codons (Akashi and Gojobori, 2002).

This must reflect in the distribution of epitope frequencies as it influences the likeliness of each n-gram. Given a terminal epitope sequence $t = a_1a_2a_3a_4a_5\ldots a_n$ its probability to be found in a proteome is

$$\mathrm{P}_{\mathrm{seq}}(t) = \prod_i p_{a_i} \tag{7.3}$$

. Since each n-gram has a distinct probability a multinomial distribution function has to be used. It directly follows that in large datasets about $N\,\mathrm{P}_{\mathrm{seq}}(t)$ peptides terminating in t should be found.

In order to know how well TXP antibodies could cover a proteome of arbitrary size, the distribution of epitope frequencies is key.

Table 7.1: Amino acid frequencies from different sources and databases

		TREMBL	Swissprot	Expasy
Ala	A	8,6%	8,3%	8,3%
Arg	R	5,4%	5,5%	5,7%
Asn	N	4,1%	4,1%	4,4%
Asp	D	5,3%	5,5%	5,3%
Cys	C	1,2%	1,4%	1,7%
Gln	Q	4,0%	3,9%	4,0%
Glu	E	6,2%	6,8%	6,2%
Gly	G	7,1%	7,1%	7,2%
His	H	2,2%	2,3%	2,2%
Ile	I	6,1%	6,0%	5,2%
Leu	L	9,9%	9,7%	9,0%
Lys	K	5,3%	5,8%	5,7%
Met	M	2,5%	2,4%	2,4%
Phe	F	4,1%	3,9%	3,9%
Pro	P	4,6%	4,7%	5,1%
Ser	S	6,6%	6,6%	6,9%
Thr	T	5,6%	5,3%	5,8%
Trp	W	1,3%	1,1%	1,3%
Tyr	Y	3,1%	2,9%	3,2%
Val	V	6,8%	6,9%	6,6%

First the terms *epitope frequency class* and *epitope frequency set* should be defined. A *frequency class* t_k is defined as the set of epitopes repeating exactly k times. E.g. if the epitope -AMTR and -LGYR are the only epitopes repeating 42 times in the database t_{42} would be $\{$-AMTR , -LGYR $\}$ and $|t_{42}| = 2$. The respective *epitope frequency set* r_{42} contains the 84 peptides matching the epitopes in t_{42}. The probability set $\vec{p} = \{\mathrm{P}_{\mathrm{seq}}(t)|\forall t \in T_{l,protease}\}$ contains probabilities of all possible epitopes.

Good (1953) describes how to estimate the 'frequencies of frequencies', the size of any epitope frequency class t_n, from the vector of individual probabilities. The expected size of an epitope frequency class t_k is calculated as follows

$$N_k = \mathrm{E}(|t_k|) = \sum_{t \in T_{l,protease}} \binom{N}{k} \mathrm{P}_{\mathrm{seq}}(t)^k (1 - \mathrm{P}_{\mathrm{seq}}(t))^{N-k} \tag{7.4}$$

with database size N, set of epitopes $T_{l,protease}$ and associated probabilities. The probability of a peptide x to be part of the epitope peptide set r_k can be estimated as

$$\mathrm{P}(x \in r_k|N,p) = \frac{k\,\mathrm{E}(|t_k|)}{N} \tag{7.5}$$

Using a table of amino acid frequencies one can calculate the probabilities of all $18^3 \times 2 = 11.664$ tryptic tetra-peptides using a simplified model, neglecting the occurrence

of RP or KP within the terminus. As shown in the plot in figure 7.1 using amino acid frequencies as found in table 7.1 lead to a close fit to observed epitope repeat frequencies.

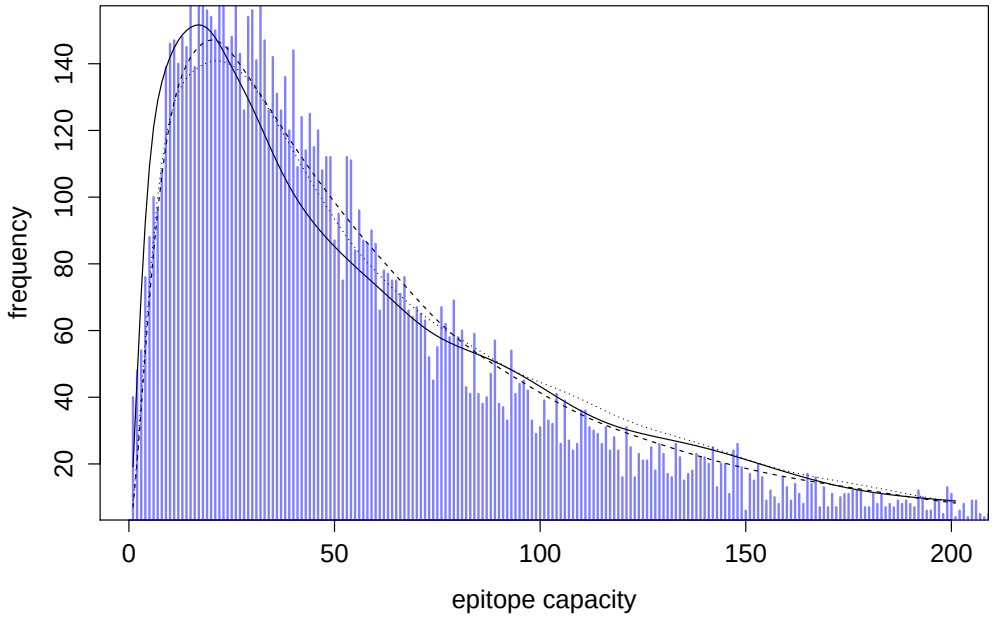


Figure 7.1: Fit of the Good model using TREMBL (solid line), Expasy (dashed line) from table 7.1 to the observed epitope frequency distribution for tryptic UniProt peptides (blue). The plot also shows a fit using an amino acid composition from the same dataset (dotted line)

The expected number of epitopes is a special case, also found in Good (1953):

$$\mathrm{E}(k|N,p) = |T| - \sum_{t \in T} (1-p_t)^N \tag{7.6}$$

with database size N, set of epitopes E and associated probabilities p. The plot if figure 7.2 shows the number of expected different epitopes depending on the database size. A database containing 100.000 tryptic peptides is expected to contain already 99% of all possible terminal sequences of length 4.

The Good model explains the observations very well, so it should be possible to use it for some deductions, for example on the length of the targeted epitope.

We shall further calculate the probabilities for peptides to belong to a given frequency class in average sized proteome digest of 10^6 peptides, depending on the epitope length. From the perspective of wishing to cover a decent, but not too large number of proteins,

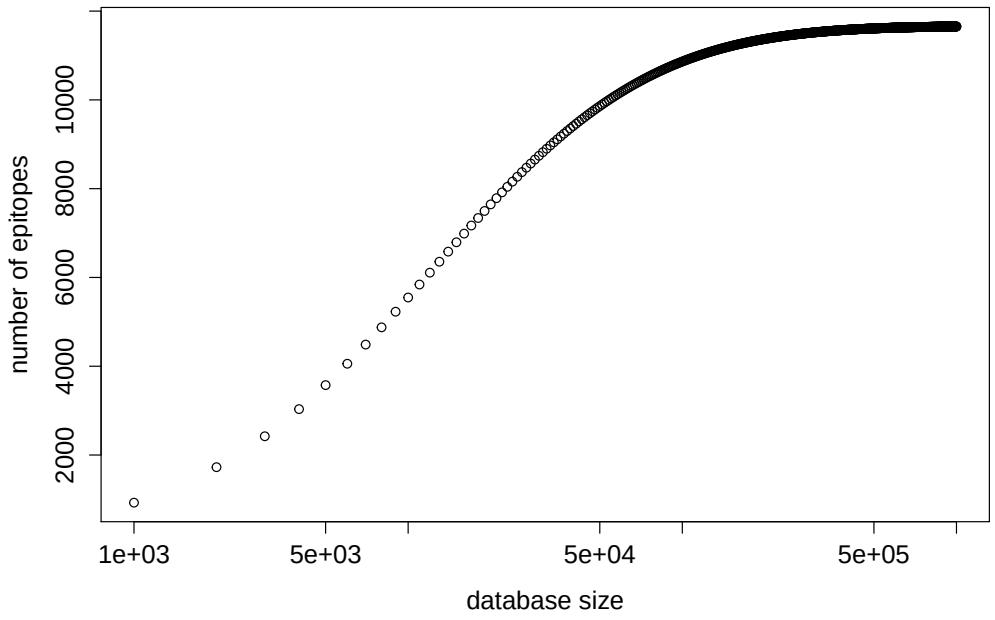


Figure 7.2: Relationship of the database size to the expected number of different tryptic tetramer epitopes usind TREMBL AA frequencies

by a single TXP antibody such a binder should theoretically bind 50 to 600 peptides. In other words ideally the binder is part of $\bigcup_{k=50}^{600} r_k$, the union of these frequency classes r_{50} to r_{600}. This is given by

$$\mathrm{P}\left(x \in \bigcup_{k=50}^{600} r_k\right) = \sum_{k=50}^{600} \frac{\mathrm{E}(|t_k|)k}{N}. \tag{7.7}$$

The results are 5.7 % for epitopes of length 3, 87.2 % for length 4 and 0.5 % for length 5. These numbers have been calculated using the respective probability vectors derived from the amino acid frequencies found in table 7.1 . This means that epitopes of length 4 in theory are best suited regarding this aspect. It is almost certain that a protein digested to approximately 40-50 different peptides contains at least one tryptic epitope which is nor extremely rare and neither very common in the proteome. When considering longer or shorter epitope lengths many epitopes will either repeat very often or be almost unique, nullifying the advantage of these binders.

7.3 Epitope enrichment in sequence lists

A central question arising when analyzing TXP data is: 'How likely is it to observe multiple peptides sharing the same terminus in a list of sequences?' This is particularly important when analyzing the capture specificity and quality of newly generated antibodies. As presented in chapter 6 one solution is to use sampling to estimate p-values for an enrichment event $E_\Phi(i,j)$, which is defined as: i matching signals out of j masses from the same epitope class, by applying the parameters $\Phi = (D, \varepsilon_{tol}, t, l, k)$ in a peak list of k masses. This approach works reasonably well (Planatscher *et al.*, 2014), however it becomes increasingly inaccurate and sampling needs to be redone for each database/search configuration. We shall deduce an analytical approach to this question in this chapter, also including distribution models introduced earlier in this section.

When introducing the TXP-TEA and MATERICS algorithms in chapter 6 we exclusively deal with enrichment events to be observed in mass lists deduced from spectra. Working with masses always includes dealing with measurement tolerances, isobaric sequences, etc. For simplicity the following probabilistic models start from sequence lists. Note, that this does not dilute the applicability of the deduction to real datasets. In the case of MS data the mass list can be easily transformed to a sequence list, and tandem MS experiments directly lead to sequence list results anyway.

In order to work with lists of sequences instead of masses or signals, the enrichment event $E_\Pi(i,j)$ will be further defined as: i matching sequences out of j sequences from the same epitope class, using the parameters $\Pi = (D, t, l, k)$, in a list of k sequences. The probability of observing $E_\Pi(i,j)$ of course depends on the size of the epitope frequency set r_j. N_j denotes the expected number of j-repeating epitopes.

We also introduce the event $\tilde{E}_\Pi(i,j)$, as the event of observing one *specific* - opposed to *any* - epitope from t_j. The probability of this event follows a hypergeometric distribution. In this case the database is the 'urn' containing N 'balls', the peptides matching the specific epitope in database j is the number of 'red' balls, the number of matching peptides in the sample i is the number of successful picks, and the sequence list length k gives the number of tries :

$$\mathrm{P}(\tilde{E}_\Pi(i,j)) = \frac{\binom{j}{i}\binom{N-j}{k-i}}{\binom{N}{k}} \tag{7.8}$$

E.g. if the database of 10^6 peptides contains 15 LGYR-peptides and we would randomly pick 200 peptides, the probability of oberving 2 LGYR-peptides would be:

$$\mathrm{P}(\tilde{E}_\Pi(2,15)) = \frac{\binom{15}{2}\binom{10^6-15}{200-2}}{\binom{10^6}{200}} = 4.16826e-06 \tag{7.9}$$

which is, of course, an extraordinarily improbable event. The odds to observe it are about 1:240000. However we are not interested in the odds of observing 2 of 15 LGYR

peptides, but the odds of observing 2 peptides of any epitope occurring 15 times in the database. This odds can be derived from the probability for the specific event. The event $E_\Pi(i,j)$ is the inclusive disjunction of the N_j possible $\tilde{E}_\Pi(i,j)$-events.

$$E_\Pi(i,j) = \bigcup^{N_j} \tilde{E}_\Pi(i,j) \tag{7.10}$$

All specific $\tilde{E}_\Pi(i,j)$-events can be considered as independent. This is a valid assumption for $i \ll k$ (only few of the k observed sequences have identical terminal epitopes), which is the relevant case in real datasets. Because the events are independent the union probability can be simplified to a closed form using DeMorgan's law:

$$\begin{aligned}
\mathrm{P}(\tilde{E}_\Pi(i,j)) = & \quad \mathrm{P}\left(\bigcup^{N_j} \tilde{E}_\Pi(i,j)\right) && (7.11)\\
= & \quad 1 - \mathrm{P}\left(\bigcap^{N_j} \neg\tilde{E}_\Pi(i,j)\right) && (7.12)\\
= & \quad 1 - \prod^{N_j} (1 - \mathrm{P}(\tilde{E}_\Pi(i,j))) && (7.13)\\
= & \quad 1 - (1 - \mathrm{P}(\tilde{E}_\Pi(i,j)))^{N_j} && (7.14)\\
= & \quad 1 - \left(1 - \frac{\binom{j}{i}\binom{N-j}{k-i}}{\binom{N}{k}}\right)^{N_j} && (7.15)
\end{aligned}$$

When ranking sequence enrichments by p-value-based scores, the score also includes events which are more extreme. A more extreme event would be either observing more than i of j sequences, or i of less than j epitopes, or both. The event of observing 'i or more of j or less' shall be formalized as $E_\Pi(\geq i, \leq j)$. The event of observing 'i or more of j' shall be formalized as $E_\Pi(\geq i,\ j)$. In analogy to $\mathrm{P}(E_\Pi(i,j))$ the probability of $E_\Pi(\geq i,\ j)$ is

$$\begin{aligned}
\mathrm{P}\left(E_{\Pi}(\geq i,j)\right) &= \mathrm{P}\left(\bigcup^{N_j}\tilde{E}_{\Pi}(\geq i,j)\right) && (7.16)\\
&= 1-\mathrm{P}\left(\bigcap^{N_j}\neg\tilde{E}_{\Pi}(\geq i,j)\right) && (7.17)\\
&= 1-\prod^{N_j}(1-\mathrm{P}(\tilde{E}_{\Pi}(\geq i,j))) && (7.18)\\
&= 1-(1-\mathrm{P}(\tilde{E}_{\Pi}(\geq i,j)))^{N_j} && (7.19)\\
&= 1-\left(1-\left(1-\sum_{a=0}^{i-1}\frac{\binom{j}{a}\binom{N-j}{k-a}}{\binom{N}{k}}\right)\right)^{N_j} && (7.20)\\
&= 1-\left(\sum_{a=0}^{i-1}\frac{\binom{j}{a}\binom{N-j}{k-a}}{\binom{N}{k}}\right)^{N_j} && (7.21)
\end{aligned}$$

Furthermore the p-value can be deduced by forming the inclusive disjunction for the epitope frequency classes $t_i, t_{i+1}, \ldots, t_j$:

$$\begin{aligned}
\mathrm{P}\left(E_{\Pi}(\geq i,\leq j)\right) &= \mathrm{P}\left(\bigcup_{m=i}^{j}E_{\Pi}(\geq i,m)\right) && (7.22)\\
&= 1-\mathrm{P}\left(\bigcap_{m=i}^{j}\neg E_{\Pi}(\geq i,m)\right) && (7.23)\\
&= 1-\prod_{m=1}^{j}(1-\mathrm{P}(E_{\Pi}(\geq i,m))) && (7.24)
\end{aligned}$$

By inserting (7.21) into (7.24), a closed form can be found:

$$\begin{aligned}
\mathrm{P}\left(E_{\Pi}(\geq i,\leq j)\right) &= 1-\prod_{m=i}^{j}\left(1-\left(1-\left(\sum_{a=0}^{i-1}\frac{\binom{m}{a}\binom{N-m}{k-a}}{\binom{N}{k}}\right)^{N_m}\right)\right) && (7.25)\\
&= 1-\prod_{m=i}^{j}\left(\sum_{a=0}^{i-1}\frac{\binom{m}{a}\binom{N-m}{k-a}}{\binom{N}{k}}\right)^{N_m} && (7.26)
\end{aligned}$$

This score can be efficiently calculated even for very large sequence lists. Since the size of the epitope frequency classes N_j can be estimated using (7.4), it is possible to return a score for databases of arbitrary size. Alternatively N_j can be tabulated for each individual database, reflecting the specific distribution characteristics.

Numerically it makes sense to calculate the logarithmic probability of the counter-event, because the $\mathrm{P}(E_\Pi(\geq i, \leq j))$ can become very small, leading to underflows.

$$\begin{aligned} EScore(\Pi, i, j)) &= log(\mathrm{P}(\neg E_\Pi(\geq i, \leq j))) && (7.27)\\ &= log\left(\prod_{m=i}^{j}\left(\sum_{a=0}^{i-1}\frac{\binom{m}{a}\binom{N-m}{k-a}}{\binom{N}{k}}\right)^{N_m}\right) && (7.28)\\ &= \sum_{m=i}^{j} log\left(\left(\sum_{a=0}^{i-1}\frac{\binom{m}{a}\binom{N-m}{k-a}}{\binom{N}{k}}\right)^{N_m}\right) && (7.29)\\ &= \sum_{m=i}^{j} N_m log\left(\sum_{a=0}^{i-1}\frac{\binom{m}{a}\binom{N-m}{k-a}}{\binom{N}{k}}\right) && (7.30) \end{aligned}$$

The score has the characteristic of a p-value and only makes sense in conjunction with a significance threshold. As observed for the TXP-TEA algorithm every score calculated is the result of a significance test, which falls under the definition of multiple testing, if strictly interpreted. Therefore the usual threshold of 5% should be divided by the number of tested epitopes m, using the Bonferroni correction for multiple comparisons. This gives an EScore-threshold $EScore_t$ of

$$EScore_t(\alpha) = -log\left(\frac{\alpha}{m}\right) \tag{7.31}$$

Only enrichment events with an $EScore \geq EScore_t(\alpha)$ should be considered as significant, and should eventually be included in a model for the true binding characteristics of the antibody. For sequence list L the set $S_\Pi(L)$ should denote the epitopes fulfilling this criterion. A simple procedure to combine all enriched sequence variants to one model is to calculate rate of enrichment, defined as the relative amount of sequences in the database which have been observered in the sample

$$re(a) = \frac{count[a]}{totalcount[a]} = \frac{i}{j} \tag{7.32}$$

for all significantly enriched sequences. The weight of this epitope in the overall binding motif for a sequence list L can then be obtained by:

$$w(a, X) = \frac{re(a)}{\sum_{b \in S_\Pi(L)} re(b)} = \frac{i}{j} \tag{7.33}$$

All of these operations can be implemented in a spreadsheet, providing lookup table for the epitope occurence ($totalcount[a]$ or j) and repeat frequencies (N_m), and integrate well to the usual workflow in the lab. The user just copies the sequence list L to the spreadsheet, and obtains the ranking of enriched epitopes and a link to the resulting motif depicted as a sequence logo.

Table 7.2: Average Spearman rank correlation coefficient when comparing E-Score ranking to TXP TEA ranking based on results from 108 different mass spectra

	A357	HEK	HELA	average
AMTR	0.928	0.931	0.930	0.930
EIER	0.908	0.893	0.910	0.904
LGYR	0.927	0.929	0.926	0.927
average	0.921	0.918	0.922	0.920

7.4 Validation of the model

In order to compare the result obtained by the TXP TEA algorithm based on sampling estimation and the enrichment model described in this chapter, screening results from the MATERICS paper (Planatscher *et al.*, 2014) have been analyzed by both methods. It has already been shown that results obtained by TXP TEA are comparable to data from expensive peptide library experiments. If the enrichment model calculates similar linear epitope rankings, the model itself and its assumptions are valid.

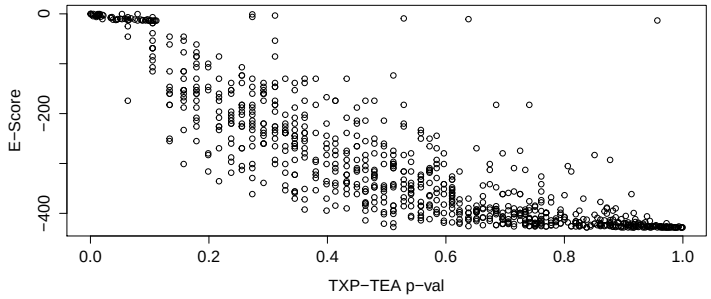


Figure 7.3: Comparison of the E-Score and the TXP-TEA p-value for a specific prediction

The methodology of comparison is described as follows. First a ranking based on the p-value of TXP-TEA and the E-score of the model are calculated. Then the rankings are compared using Spearmans Ranking Corrrelation coefficient $r_s = 1 - \frac{6\sum_i d_i^2}{n \cdot (n^2-1)}$ where $d_i = r_{tea}(e_i) - r_{model}(e_i)$ is the difference in rank.

Result of this evaluation are shown in table 7.4 and figure . On average the correlation coefficient is 0.920 for all tested rankings (108 mass spectra). This demonstrates that the ranking by E-Score is highly correlated to the results obtained by TXP TEA.

Chapter 8

Summary and concluding remarks

It has been shown that modern biochemical test development can significantly profit from combinatorial optimization.

Multiplex assays do require complex planning decision during implementation and subsequent validation. This work shows that classical integer programming approaches can help to find optimal solutions to these decision problems.

First this thesis elaborated on multiplex serological assays for the simultaneous measurement of antibody concentrations in serum. Three problems have been modelled in this application domain, which could be addressed by combinatorial optimization.

The first treated problem was to systemically pool samples in order to create a multi-positive control sample. While this has been done before, here a theoretical foundation has been proposed and proven by an actual algorithmic implementation and systematic experiments. We could show that pooled samples exhibit a predictable serological profile. A pooled sample with the desired properties can be created by using this prediction.

The next two problems dealt with multiplexed assay validation -an essential requirement imposed by health authorities. For serological assay validation it must be shown that low, medium, and high levels can be reliably measured. Reference samples cannot be easily obtained in the realm of serological assays, because the target analyte cannot be synthesized. The only feasible way to validate such an assay is to measure a patient sample with low, a second sample with medium, and yet another one with a high level of the antibody. We have shown that it is possible to choose a few samples such that the combination of their profiles would cover all ranges for most analytes, such that validation cost and effort was significantly reduced.

The last approached problem in the serological assay domain combined the latter methods to validate multiplexed assays using a set of pooled samples. By the composition of sample pools the number of serological patterns is extended exponentially. This made it much more likely to find a combination of patterns that allows the validation of the observed targets. A novel algorithm combining fast enumeration and a set cover formulation has been introduced. Using the approach it was possible to improve assay validation to optimality.

The next problem was a detour to the domain of planar assay and combinatorial designs. A common issue when designing protein microarrays is the need to avoid side-

effects during read-out. It is likely that the intensity measured on one spot is influenced by its neighboring spots. Therefore it is desirable to avoid placing replicates of the sample twice or more often in the same neighbourhood if such a situation can be avoided. In order to provide solutions to this practical problem in the lab, a small application named ProChOpt was developed, which can be easily used to solve up to medium- or large-sized sample arrangement problems. The constraint programming formulation implemented in ProChopt was able to provide answers also on the question if it is at all feasible to place a number of replicates on a predefined grid, respecting the imposed layout restrictions, or not.

The major part of the thesis dealt with optimization and data analysis for Triple X Proteomics - immunoaffinity assays using antibodies binding short linear, terminal epitopes of peptides.

It has been shown that the problem of choosing a minimal set of epitopes for TXP setups, which combine mass spectrometry with immunaffinity enrichment, is equivalent to the well-known set cover problem. In combination with a filter pipeline that eliminates unsuitable peptide-epitope combinations, we proposed different methods for the solution of the problem. For small datasets it was possible to solve the problem optimally with minimal computational effort using commercial or free solvers. Larger datasets, like full proteomes, required the use of heuristics, or respectively a running time limitation of the branch-and-bound search in the integer program solvers.

Sandwich immunoassays (SIA) use two antibodies to capture and detect a target molecule. TXP SIAs do the same with peptides as targets, by combining the respective C-terminal and N-terminal binder. The task of selecting the smallest-possible sets of C- and N-terminal epitopes for a given set of proteins was different from the immunoaffinity-MS optimization problem. Here quadratic constraints had to be considered. These models were linearized, resulting in very high dimensional problem formulations. Because of the huge dimensions these models were too difficult to solve. A greedy heuristic and a meta-heuristic using local search was presented, which proved to be more efficient than pure ILP formulations.

This concluded the optimization section of the work. All models described in this thesis were implemented in the novel Java software framework named *SCPSolver*. This modelling framework itself is not restricted to the optimization problems outlined in this work, and is applicable to many problems that can be formulated as integer programs. While the main design goal of SCPSolver was usability, it also provides a basic modelling language, easy deployment and platform independence.

The various applications of combinatorial mathematics in validation, planning and conception of immunoaffinity assay experiments which have been tackled in this work, represent merely a fraction of the potential for mathematical optimization in the realm of lab work. Due to the increasing complexity of setups, such as the degree of multiplexity, and the access to data and sample material in bio-banks, and the limited availability of lab-time - the need to make optimal decisions is a key element for the success of biochemical research and test development.

The second part of this thesis explored epitope enrichment statistics for Triple X Proteomics. One question arising when analyzing TXP data was: 'How likely is it to observe multiple peptides sharing the same terminus?' This is particularly important when analyzing the capture specificity and quality of newly generated antibodies. The algorithms TXP-TEA and MATERICS, presented in this thesis, were able to identify binding characteristics of TXP antibodies from data obtained in immunoaffinity MS experiments. The resulting motifs closely resembled patterns found by using an expensive peptide library approach. The described methods for motif elucidation lead to a substantial reduction in costs. In addition the novel method enables the weighting of isobaric variations in the binding motifs. Similar techniques might well improve peptide identification algorithms, which exploit the existing data on potentially enriched sequences during the search process. Our findings are relevant to other fields of bio-medical research, such as in the identification of the binding properties of MHC molecules.

Also the set of possible distinct TXP epitopes in a proteome has been examined from a theoretical perspective. A multinomial statistical model explains the distributions observed in sequence databases, and it was possible to use it for strategical deductions on the length of the targeted epitope. E.g. using the model, it has been shown the epitope length of 4 amino acids provides the best balance of coverage and specificity.

Further it was possible to derive an alternative analytical scoring method for epitope enrichment in sequence lists. This model can certainly be expanded to detect enrichment of internal sequences, and therefore be applied to analyze binding epitopes of any peptide-specific binding structure.

Through this work resources in the lab can be used more efficiently and it provides new tools for immunoaffinity data analysis. Hopefully, this will contribute to the development of validated, well-characterized and economically feasible new tests in research and diagnostics.

Bibliography

Aarts, E. H. L. and Lenstra, J. K. (2003). *Local Search in Combinatorial Optimization*. Princeton University Press.

Aebersold, R. and Mann, M. (2003). Mass spectrometry-based proteomics. *Nature*, **422**(6928), 198–207.

Akashi, H. and Gojobori, T. (2002). Metabolic efficiency and amino acid composition in the proteomes of Escherichia coli and Bacillus subtilis. *Proceedings of the National Academy of Sciences of the United States of America*, **99**, 3695–3700.

Alpha-Bazin, B. and Quemeneur, E. (2012). Antibody-free detection of phosphoserine/threonine containing peptides by homogeneous time-resolved fluorescence. *Analytical Chemistry*, **84**(22), 9963–9970.

Anderson, N. L., Polanski, M., Pieper, R., Gatlin, T., Tirumalai, R. S., Conrads, T. P., Veenstra, T. D., Adkins, J. N., Pounds, J. G., Fagan, R., and Lobley, A. (2004a). The human plasma proteome: a nonredundant list developed by combination of four separate sources. *Molecular & cellular proteomics : MCP*, **3**(4), 311–26.

Anderson, N. L., Jackson, A., Smith, D., Hardie, D., Borchers, C., and Pearson, T. W. (2009). SISCAPA peptide enrichment on magnetic beads using an in-line bead trap device. *Molecular & cellular proteomics : MCP*, **8**, 995–1005.

Anderson, N. L. G., Haines, L. R., Hardie, D. B., Olafson, R. W., and Pearson, T. W. (2004b). Mass spectrometric quantitation of peptides and proteins using Stable Isotope Standards and Capture by Anti-Peptide Antibodies (SISCAPA). *J Proteome Res*, **3**(2), 235–244.

Arora, S. (1998). *The approximability of NP-hard problems*. ACM New York, NY, USA.

Auger, I., Balandraud, N., Rak, J., Lambert, N., Martin, M., and Roudier, J. (2009). New autoantigens in rheumatoid arthritis (RA): screening 8268 protein arrays with sera from patients with RA. *Annals of the Rheumatic Diseases*, **68**(4), 591–594.

Beare, P. A., Chen, C., Bouman, T., Pablo, J., Unal, B., Cockrell, D. C., Brown, W. C., Barbian, K. D., Porcella, S. F., Samuel, J. E., Felgner, P. L., and Heinzen, R. a. (2008). Candidate antigens for Q fever serodiagnosis revealed by immunoscreening of a Coxiella burnetii protein microarray. *Clinical and vaccine immunology*, **15**(12), 1771–9.

Beavis, R. C. (2006). Using the global proteome machine for protein identification. *Methods In Molecular Biology Clifton Nj*, **328**(4), 217–228.

Boni, M. F., Chau, N. V. V., Dong, N., Todd, S., Nhat, N. T. D., de Bruin, E., van Beek, J., Hien, N. T., Simmons, C. P., Farrar, J., and Koopmans, M. (2013). Population-Level Antibody Estimates to Novel Influenza A/H7N9. *The Journal of infectious diseases*, **208**(4), 554–8.

Cooley, G., Etheridge, R. D., Boehlke, C., Bundy, B., Weatherly, D. B., Minning, T., Haney, M., Postan, M., Laucella, S., and Tarleton, R. L. (2008). High throughput selection of effective serodiagnostics for Trypanosoma cruzi infection. *PLoS neglected tropical diseases*, **2**(10), e316.

Craig, R. and Beavis, R. C. (2004). TANDEM: Matching proteins with tandem mass spectra. *Bioinformatics*, **20**, 1466–1467.

Cummings, J., Ward, T. H., Greystoke, a., Ranson, M., and Dive, C. (2008). Biomarker method validation in anticancer drug development. *British journal of pharmacology*, **153**(4), 646–56.

Desiere, F., Deutsch, E. W., King, N. L., Nesvizhskii, A. I., Mallick, P., Eng, J., Chen, S., Eddes, J., Loevenich, S. N., and Aebersold, R. (2006). The PeptideAtlas project. *Nucleic Acids Research*, **34**, D655–D658.

Deutsch, E. W., Lam, H., and Aebersold, R. (2008). PeptideAtlas: a resource for target selection for emerging targeted proteomics workflows. *EMBO Reports*, **9**(5), 429–434.

Dobson, G. (1982). Worst-Case Analysis of Greedy Heuristics for Integer Programming with Nonnegative Data. *Mathematics of Operations Research*, **7**(4), 515–531.

Doolan, D. L., Mu, Y., Unal, B., Sundaresh, S., Hirst, S., Valdez, C., Randall, A., Molina, D., Liang, X., Freilich, D. A., Oloo, J. A., Blair, P. L., Aguiar, J. C., Baldi, P., Davies, D. H., and Felgner, P. L. (2008). Profiling humoral immune responses to P. falciparum infection with protein microarrays. *Proteomics*, **8**(22), 4680–94.

Dorigo, M. and Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, **1**, 53–66.

Eisen, D., Planatscher, H., Hardie, D. B., Kraushaar, U., Pynn, C. J., Stoll, D., Borchers, C., Joos, T. O., and Poetz, O. (2013). G protein-coupled receptor quantification using peptide group-specific enrichment combined with internal peptide standard reporter calibration. *Journal of proteomics*, **90**, 85–95.

Ekins, R. P. (1998). Ligand assays: from electrophoresis to miniaturized microarrays. *Clin Chem*, **44**(9), 2015–2030.

Eng, J. K., McCormack, A. L., and Yates, J. R. (1994). An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry*, **5**, 976–89.

Falk, K. and Rötzschke, O. (1993). Consensus motifs and peptide ligands of MHC class I molecules. *Seminars in Immunology*, **5**(2), 81–94.

FDA (2001). FDA, Guidance for Industry Bioanalytical Method Validation.

Feige, U. (1998). A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, **45**(4), 634–652.

Felgner, P. L., Kayala, M. a., Vigil, A., Burk, C., Nakajima-Sasaki, R., Pablo, J., Molina, D. M., Hirst, S., Chew, J. S. W., Wang, D., Tan, G., Duffield, M., Yang, R., Neel, J., Chantratita, N., Bancroft, G., Lertmemongkolchai, G., Davies, D. H., Baldi, P., Peacock, S., and Titball, R. W. (2009). A Burkholderia pseudomallei protein microarray reveals serodiagnostic and cross-reactive antigens. *Proceedings of the National Academy of Sciences of the United States of America*, **106**(32), 13499–504.

Fenyo, D., Eriksson, J., and Beavis, R. (2010). Mass spectrometric protein identification using the global proteome machine. *Methods Mol Biol*, **673**, 189–202.

Feo, T. a. and Resende, M. G. C. (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, **6**(2), 109–133.

Fíla, J., Honys, D., and Fila, J. (2012). Enrichment techniques employed in phosphoproteomics. *Amino Acids*, **43**(3), 1025–1047.

Fredriksson, S., Gullberg, M., Jarvius, J., Olsson, C., Pietras, K., Gústafsdóttir, S. M., Ostman, A., Landegren, U., and Gustafsdottir, S. M. (2002). Protein detection using proximity-dependent DNA ligation assays. *Nat Biotechnol*, **20**(5), 473–477.

Geer, L. Y., Markey, S. P., Kowalak, J. a., Wagner, L., Xu, M., Maynard, D. M., Yang, X., Shi, W., and Bryant, S. H. (2004). Open mass spectrometry search algorithm. *Journal of proteome research*, **3**(5), 958–64.

Gnjatic, S., Wheeler, C., Ebner, M., Ritter, E., Murray, A., Altorki, N. K., Ferrara, C. A., Hepburne-Scott, H., Joyce, S., Koopman, J., McAndrew, M. B., Workman, N., Ritter, G., Fallon, R., and Old, L. J. (2009). Seromic analysis of antibody responses in non-small cell lung cancer patients and healthy donors using conformational protein arrays. *Journal of Immunological Methods*, **341**(1-2), 50–58.

Gomory, R. E. (1958). Outline of an algorithm for integer solutions to linear programs.

Gomory, R. E. (1963). An algorithm for integer solutions to linear programs. *Recent advances in mathematical programming*, **11**, 269–302.

Good, I. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, **40**(3), 237–264.

Heller and Tompkins (1956). An extension of a theorem of Dantzig"s. *Annals of Mathematics Studies*, **38**.

Helman, P., Moret, B. M. E., and Shapiro, H. D. (1993). An Exact Characterization of Greedy Structures. *SIAM Journal on Discrete Mathematics*, **6**(2), 274–283.

Henzel, W. J., Billeci, T. M., Stults, J. T., Wong, S. C., Grimley, C., and Watanabe, C. (1993). Identifying proteins from two-dimensional gels by molecular mass searching of peptide fragments in protein sequence databases. *Proceedings of the National Academy of Sciences of the United States of America*, **90**(11), 5011–5.

Hermjakob, H. and Apweiler, R. (2006). The Proteomics Identifications Database (PRIDE) and the ProteomExchange Consortium: making proteomics data accessible.

Hoeppe, S., Schreiber, T. D., Planatscher, H., Zell, A., Templin, M. F., Stoll, D., Joos, T. O., and Poetz, O. (2011). Targeting peptide termini, a novel immunoaffinity approach to reduce complexity in mass spectrometric protein identification. *Molecular cellular proteomics MCP*, **10**(2), M110.002857.

Hoofnagle, A. N., Becker, J. O., Wener, M. H., and Heinecke, J. W. (2008). Quantification of thyroglobulin, a low-abundance serum protein, by immunoaffinity peptide enrichment and tandem mass spectrometry. *Clin Chem*, **54**(11), 1796–1804.

Houghten, R. A., Pinilla, C., Blondelle, S. E., Appel, J. R., Dooley, C. T., and Cuervo, J. H. (1991). Generation and use of synthetic peptide combinatorial libraries for basic research and drug discovery. *Nature*, **354**(6348), 84–6.

Hudson, M. E., Pozdnyakova, I., Haines, K., Mor, G., and Snyder, M. (2007). Identification of differentially expressed proteins in ovarian cancer using high-density protein microarrays. *Proceedings of the National Academy of Sciences of the United States of America*, **104**(44), 17494–17499.

James, P., Quadroni, M., Carafoli, E., and Gonnet, G. (1993). Protein identification by mass profile fingerprinting. *Biochem Biophys Res Commun*, **195**(1), 58–64.

Joos, T., Stoll, D., Templin, M., and Poetz, O. (2007). Method for the detection and/or enrichment of analyte proteins and/or analyte peptides from a complex protein mixture. WO Patent App. PCT/EP2007/002,802.

Joos, T., Templin, M., Stoll, D., Poetz, O., Zell, A., Planatscher, H., and Supper, J. (2010). A method for determining in silico- a set of selected target epitopes. WO Patent App. PCT/EP2009/001,230.

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

Karp, R. M. (1972). Reducibility among combinatorial problems.

Kattah, M. G., Alemi, G. R., Thibault, D. L., Balboni, I., and Utz, P. J. (2006). A new two-color Fab labeling method for autoantigen protein microarrays. *Nature methods*, **3**(9), 745–51.

Knuth, D. E. (1970). The Art of Computer Programming. Volume 2: Seminumerical Algorithms.

Kondrat, R. W., McClusky, G. A., and Cooks, R. G. (1978). Multiple reaction monitoring in mass spectrometry/mass spectrometry for direct analysis of complex mixtures. *Analytical Chemistry*, **50**(14), 2017–2021.

Korte, B. and Vygen, J. (2008). *Kombinatorische Optimierung: Theorie und Algorithmen*, volume 54. Springer-Verlag, Berlin Heidelberg.

Krastins, B., Prakash, A., Sarracino, D. A., Nedelkov, D., Niederkofler, E. E., Kiernan, U. A., Nelson, R., Vogelsang, M. S., Vadali, G., Garces, A., Sutton, J. N., Peterman, S., Byram, G., Darbouret, B., Pérusse, J. R., Seidah, N. G., Coulombe, B., Gobom, J., Portelius, E., Pannee, J., Blennow, K., Kulasingam, V., Couchman, L., Moniz, C., Lopez, M. F., and Perusse, J. R. (2013). Rapid development of sensitive, high-throughput, quantitative and highly selective mass spectrometric targeted immunoassays for clinically important proteins in human plasma and serum. *Clin Biochem*, **46**(6), 399–410.

Kunnath-Velayudhan, S., Salamon, H., Wang, H.-Y., Davidow, A. L., Molina, D. M., Huynh, V. T., Cirillo, D. M., Michel, G., Talbot, E. a., Perkins, M. D., Felgner, P. L., Liang, X., and Gennaro, M. L. (2010). Dynamic antibody responses to the Mycobacterium tuberculosis proteome. *Proceedings of the National Academy of Sciences of the United States of America*, **107**(33), 14703–8.

Lin, S. and Kernighan, B. W. (1973). An Effective Heuristic Algorithm for the Traveling-Salesman Problem.

Lopez, M. F., Rezai, T., Sarracino, D. A., Prakash, A., Krastins, B., Athanas, M., Singh, R. J., Barnidge, D. R., Oran, P., Borges, C., and Nelson, R. W. (2010). Selected reaction monitoring-mass spectrometric immunoassay responsive to parathyroid hormone and related variants. *Clin Chem*, **56**(2), 281–290.

Lottspeich, F. (2006). *Bioanalytik*. Spektrum Akademischer Verlag, Heidelberg, 2. auflage edition.

Luevano, M., Bernard, H.-U., Barrera-Saldaña, H. A., Trevino, V., Garcia-Carranca, A., Villa, L. L., Monk, B. J., Tan, X., Davies, D. H., Felgner, P. L., and Kalantari, M. (2010). High-throughput profiling of the humoral immune responses against thirteen human papillomavirus types by proteome microarrays. *Virology*, **405**(1), 31–40.

Lund, C. and Yannakakis, M. (1994). On the hardness of approximating minimization problems. *J. ACM*, **41**(5), 960–981.

Lund, O., Nielsen, M., Lundegaard, C., Kesmir, C., and Brunak, S. (2005). *Immunological Bioinformatics*. The MIT Press.

Madsen, J. a., Gardner, M. W., Smith, S. I., Ledvina, A. R., Coon, J. J., Schwartz, J. C., Stafford, G. C., and Brodbelt, J. S. (2009). Top-down protein fragmentation by infrared multiphoton dissociation in a dual pressure linear ion trap. *Analytical chemistry*, **81**(21), 8677–86.

Mandell, J. W. (2003). Phosphorylation state-specific antibodies: applications in investigative and diagnostic pathology. *Am J Pathol*, **163**(5), 1687–1698.

Mann, M., Hojrup, P., and Roepstorff, P. (1993). Use of mass spectrometric molecular weight information to identify proteins in sequence databases. *Biol Mass Spectrom*, **22**(6), 338–345.

Martens, L., Hermjakob, H., Jones, P., Adamsk, M., Taylor, C., States, D., Gevaert, K., Vandekerckhove, J., and Apweiler, R. (2005). PRIDE: The proteomics identifications database. *Proteomics*, **5**, 3537–3545.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et biophysica acta*, **405**(2), 442–451.

Musco, C., Sviridenko, M., and Thaler, J. (2016). Determining Tournament Payout Structures for Daily Fantasy Sports. *CoRR*, **abs/1601.0**.

Nedelkov, D., Kiernan, U. A., Niederkofler, E. E., Tubbs, K. A., and Nelson, R. W. (2006). Population proteomics: the concept, attributes, and potential for cancer biomarker research. *Mol Cell Proteomics*, **5**(10), 1811–1818.

Nelson, R. W., Krone, J. R., Bieber, A. L., and Williams, P. (1995). Mass spectrometric immunoassay. *Anal Chem*, **67**(7), 1153–1158.

Neubert, H., Gale, J., and Muirhead, D. (2010). Online high-flow peptide immunoaffinity enrichment and nanoflow LC-MS/MS: assay development for total salivary pepsin/pepsinogen. *Clin Chem*, **56**(9), 1413–1423.

Nicholas G. Hall, D. S. H. (1992). The multicovering problem. *European Journal of Operational Research*, **62**, 323–339.

Nicol, G. R., Han, M., Kim, J., Birse, C. E., Brand, E., Nguyen, A., Mesri, M., FitzHugh, W., Kaminker, P., Moore, P. a., Ruben, S. M., and He, T. (2008). Use of an immunoaffinity-mass spectrometry-based approach for the quantification of protein biomarkers from serum samples of lung cancer patients. *Molecular & cellular proteomics : MCP*, **7**(10), 1974–82.

Niemeyer, C. M., Adler, M., and Wacker, R. (2005). Immuno-PCR: high sensitivity detection of proteins by nucleic acid amplification. *Trends Biotechnol*, **23**(4), 208–216.

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., and Kanehisa, M. (1999). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res*, **27**(1), 29–34.

Olsson, N., Wingren, C., Mattsson, M., James, P., O'Connell, D., Nilsson, F., Cahill, D. J., Borrebaeck, C. A. K., and O' Connell, D. (2011). Proteomic analysis and discovery using affinity proteomics and mass spectrometry. *Molecular & cellular proteomics : MCP*, **10**, M110.003962–M110.003962.

Olsson, N., Wallin, S., James, P., Borrebaeck, C. A. K., and Wingren, C. (2012a). Epitope-specificity of recombinant antibodies reveals promiscuous peptide-binding properties. *Protein science : a publication of the Protein Society*, **21**(12), 1897–1910.

Olsson, N., James, P., Borrebaeck, C. a. K., and Wingren, C. (2012b). Quantitative Proteomics Targeting Classes of Motif-containing Peptides Using Immunoaffinity-based Mass Spectrometry. *Molecular & cellular proteomics : MCP*, **11**(8), 342–54.

Omenn, G. S. (2004). The Human Proteome Organization Plasma Proteome Project pilot phase: reference specimens, technology platform comparisons, and standardized data submissions and analyses. *Proteomics*, **4**(5), 1235–40.

Omenn, G. S., Paik, Y.-K., and Speicher, D. (2006). The HUPO Plasma Proteome Project: a report from the Munich congress.

Opalka, D., Matys, K., Bojczuk, P., Green, T., Gesser, R., Saah, A., Haupt, R., Dutko, F., and Esser, M. T. (2010). Multiplexed serologic assay for nine anogenital human papillomavirus types. *Clinical and vaccine immunology : CVI*, **17**(5), 818–27.

Pagans, S., Sakane, N., Schnolzer, M., Ott, M., and Schnolzer, M. (2009). Developing multiplexed assays for troponin I and interleukin-33 in plasma by peptide immunoaffinity enrichment and targeted mass spectrometry. *Clinical chemistry*, **55**(6), 1108–1117.

Papadimitriou, C. H. and Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity*, volume 91. Prentice Hall.

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

Perkins, D. N., Pappin, D. J., Creasy, D. M., and Cottrell, J. S. (1999). Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, **20**, 3551–3567.

Pharoah, P. (2007). How not to interpret a P value? *Journal of the National Cancer Institute*, **99**(4), 332.

Planatscher, H., Supper, J., Poetz, O., Stoll, D., Joos, T., Templin, M. F., and Zell, A. (2010). Optimal selection of epitopes for TXP-immunoaffinity mass spectrometry. *Algorithms for molecular biology : AMB*, **5**(1), 28.

Planatscher, H., Poetz, O., Stoll, D., Templin, M. F., and Joos, T. O. (2013a). Combinatorial optimization for short-epitope immunoassays. In *EURO INFORMS 26th European Conference On Operational Research Abstract Book*, page 226.

Planatscher, H., Rimmele, S., Michel, G., Potz, O., Joos, T., Schneiderhan-Marra, N., and Pötz, O. (2013b). Systematic reference sample generation for multiplexed serological assays. *Sci. Rep.*, **3**, 3259.

Planatscher, H., Weiß, F., Eisen, D., van den Berg, B. H. J., Zell, A., Joos, T., and Poetz, O. (2014). Identification of short terminal motifs enriched by antibodies using peptide mass fingerprinting. *Bioinformatics (Oxford, England)*, **30**(9), 1–9.

Poetz, O., Hoeppe, S., Templin, M. F., Stoll, D., and Joos, T. O. (2009). Proteome wide screening using peptide affinity capture. *Proteomics*, **9**(6), 1518–1523.

Raem, A. M. and Rauch, P. (2007). *Immunoassays*. Elsevier Gmbh, München, Münster.

Rajagopalan, S. and Vazirani, V. (1993). Primal-dual RNC approximation algorithms for (multi)-set (multi)-cover and covering integer programs. *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*.

Razavi, M., Frick, L. E., Lamarr, W. A., Pope, M. E., Miller, C. A., Anderson, L., and Pearson, T. W. (2012). High throughput SISCAPA quantitation of peptides from human plasma digests by ultrafast, liquid chromatography-free mass spectrometry. *J Proteome Res*.

Rissin, D. M., Kan, C. W., Campbell, T. G., Howes, S. C., Fournier, D. R., Song, L., Piech, T., Patel, P. P., Chang, L., Rivnak, A. J., Ferrell, E. P., Randall, J. D., Provuncher, G. K., Walt, D. R., and Duffy, D. C. (2010). Single-molecule enzyme-linked immunosorbent assay detects serum proteins at subfemtomolar concentrations. *Nat Biotechnol*, **28**(6), 595–599.

Robinson, W. H., DiGennaro, C., Hueber, W., Haab, B. B., Kamachi, M., Dean, E. J., Fournel, S., Fong, D., Genovese, M. C., de Vegvar, H. E. N., Skriner, K., Hirschberg, D. L., Morris, R. I., Muller, S., Pruijn, G. J., van Venrooij, W. J., Smolen, J. S., Brown, P. O., Steinman, L., and Utz, P. J. (2002). Autoantigen microarrays for multiplex characterization of autoantibody responses. *Nature Medicine*, **8**(3), 295–301.

Ross, A. H., Baltimore, D., and Eisen, H. N. (1981). Phosphotyrosine-containing proteins isolated by affinity chromatography with antibodies to a synthetic hapten. *Nature*, **294**(5842), 654–656.

Sherman, L. A. (2006). To each (MHC molecule) its own (binding motif). *The Journal of Immunology*, **177**(5), 2739–2740.

Siuzdak, G. (2006). *The Expanding Role of Mass Spectrometry in Biotechnology*, volume 15.

Steen, H. and Mann, M. (2004). The ABC's (and XYZ's) of peptide sequencing. *Nature reviews. Molecular cell biology*, **5**(9), 699–711.

Stern, L. J. (2007). Characterizing MHC-associated peptides by mass spectrometry. *Journal of immunology (Baltimore, Md. : 1950)*, **179**(5), 2667–8.

Stoevesandt, O. and Taussig, M. J. (2007). Affinity reagent resources for human proteome detection: initiatives and perspectives. *Proteomics*, **7**(16), 2738–2750.

Stoevesandt, O. and Taussig, M. J. (2012). Affinity proteomics: the role of specific binding reagents in human proteome analysis. *Expert Rev Proteomics*, **9**(4), 401–414.

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. *Proceedings of the National Academy of Sciences of the United States of America*, **101**(26), 9528–33.

Tong, M., Jacobi, C. E., van de Rijke, F. M., Kuijper, S., van de Werken, S., Lowary, T. L., Hokke, C. H., Appelmelk, B. J., Nagelkerke, N. J. D., Tanke, H. J., van Gijlswijk, R. P. M., Veuskens, J., Kolk, A. H. J., and Raap, A. K. (2005). A multiplexed and miniaturized serological tuberculosis assay identifies antigens that discriminate maximally between TB and non-TB sera. *Journal of immunological methods*, **301**(1-2), 154–63.

Tubbs, K. A., Kiernan, U. A., Niederkofler, E. E., Nedelkov, D., Bieber, A. L., and Nelson, R. W. (2005). High-throughput MS-based protein phenotyping: application to haptoglobin. *Proteomics*, **5**(18), 5002–5007.

Tubbs, K. A., Kiernan, U. A., Niederkofler, E. E., Nedelkov, D., Bieber, A. L., and Nelson, R. W. (2006). Development of recombinant-based mass spectrometric immunoassay with application to resistin expression profiling. *Anal Chem*, **78**(10), 3271–3276.

Veenstra, T. D., Conrads, T. P., Hood, B. L., Avellino, A. M., Ellenbogen, R. G., and Morrison, R. S. (2005). Biomarkers: mining the biofluid proteome. *Molecular & cellular proteomics : MCP*, **4**(4), 409–18.

Vigil, A., Ortega, R., and Nakajima-Sasaki, R. (2010). Genome-wide profiling of humoral immune response to Coxiella burnetii infection by protein microarray. *Proteomics*, **10**(12), 2259–2269.

Vigil, A., Chen, C., Jain, A., Nakajima-Sasaki, R., Jasinskas, A., Pablo, J., Hendrix, L. R., Samuel, J. E., and Felgner, P. L. (2011). Profiling the humoral immune response of acute and chronic Q fever by protein microarray. *Molecular & cellular proteomics : MCP*, **10**(10), M110.006304.

Vizoso Pinto, M. G., Pfrepper, K.-I., Janke, T., Noelting, C., Sander, M., Lueking, A., Haas, J., Nitschko, H., Jaeger, G., and Baiker, A. (2010). A systematic approach for the identification of novel, serologically reactive recombinant Varicella-Zoster Virus (VZV) antigens. *Virology journal*, **7**, 165.

Volk, S., Schreiber, T. D., Eisen, D., Wiese, C., Planatscher, H., Pynn, C. J., Stoll, D., Templin, M. F., Joos, T. O., Potz, O., Schneider, S., and Poetz, O. (2012). Combining Ultracentrifugation and Peptide Termini Group-specific Immunoprecipitation for Multiplex Plasma Protein Analysis. *Molecular & Cellular Proteomics*, **11**(7), O111 015438.

Warren, E. N., Elms, P. J., Parker, C. E., and Borchers, C. H. (2004). Development of a protein chip: a MS-based method for quantitation of protein expression and modification levels using an immunoaffinity approach. *Anal Chem*, **76**(14), 4082–4092.

Washburn, M. P., Wolters, D., and Yates, J. R. r. (2001). Large-scale analysis of the yeast proteome by multidimensional protein identification technology. *Nat Biotechnol*, **19**(3), 242–247.

Waterboer, T., Sehr, P., Michael, K. M., Franceschi, S., Nieland, J. D., Joos, T. O., Templin, M. F., and Pawlita, M. (2005). Multiplex human papillomavirus serology based on in situ-purified glutathione s-transferase fusion proteins. *Clinical Chemistry*, **51**(10), 1845–53.

Weiß, F., van den Berg, B. H. J., Planatscher, H., Pynn, C. J., Joos, T. O., and Poetz, O. (2014). Catch and measure-mass spectrometry-based immunoassays in biomarker research. *Biochimica et biophysica acta*, **1844**, 927–932.

Whiteaker, J. R. and Paulovich, A. G. (2011). Peptide immunoaffinity enrichment coupled with mass spectrometry for Peptide and protein quantification. *Clinics in Laboratory Medicine*, **31**(3), 385–396.

Whiteaker, J. R., Zhao, L., Zhang, H. Y., Feng, L. C., Piening, B. D., Anderson, L., and Paulovich, A. G. (2007). Antibody-based enrichment of peptides on magnetic beads for mass-spectrometry-based quantification of serum biomarkers. *Anal Biochem*, **362**(1), 44–54.

Whiteaker, J. R., Lin, C., Kennedy, J., Hou, L., Trute, M., Sokal, I., Yan, P., Schoenherr, R. M., Zhao, L., Voytovich, U. J., Kelly-Spratt, K. S., Krasnoselsky, A., Gafken, P. R., Hogan, J. M., Jones, L. A., Wang, P., Amon, L., Chodosh, L. A., Nelson, P. S., McIntosh, M. W., Kemp, C. J., and Paulovich, A. G. (2011). A targeted proteomics-based pipeline for verification of biomarkers in plasma. *Nat Biotechnol*.

Wilhelm, M., Schlegl, J., Hahne, H., Gholami, A. M., Lieberenz, M., Savitski, M. M., Ziegler, E., Butzmann, L., Gessulat, S., Marx, H., Mathieson, T., Lemeer, S., Schnatbaum, K., Reimer, U., Wenschuh, H., Mollenhauer, M., Slotta-Huspenina, J., Boese, J.-H., Bantscheff, M., Gerstmair, A., Faerber, F., and Kuster, B. (2014). Mass-spectrometry-based draft of the human proteome. *Nature*, **509**(7502), 582–587.

Wingren, C., James, P., and Borrebaeck, C. A. K. (2009). Strategy for surveying the proteome using affinity proteomics and mass spectrometry. *Proteomics*, **9**(6), 1511–7.

Wong, S. J., Demarest, V. L., Boyle, R. H., Wang, T., Ledizet, M., Kar, K., Kramer, L. D., Fikrig, E., and Koski, R. A. (2004). Detection of Human Anti-Flavivirus Antibodies with a West Nile Virus Recombinant Antigen Microsphere Immunoassay. *Journal of Clinical Microbiology*, **42**(1), 65–72.

Wu, C. H., Apweiler, R., Bairoch, A., Natale, D. a., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M. J., Mazumder, R., O'Donovan, C., Redaschi, N., and Suzek, B. (2006). The Universal Protein Resource (UniProt): an expanding universe of protein information. *Nucleic acids research*, **34**(Database issue), D187–91.

Yamashita, M. and Fenn, J. B. (1984). Electrospray ion source. Another variation on the free-jet theme. *The Journal of Physical Chemistry*, **434**, 4451–4459.

Yates, J. R. r., Speicher, S., Griffin, P. R., and Hunkapiller, T. (1993). Peptide mass maps: a highly informative approach to protein identification. *Anal Biochem*, **214**(2), 397–408.

Zhang, H., Zha, X., Tan, Y., Hornbeck, P. V., Mastrangelo, A. J., Alessi, D. R., Polakiewicz, R. D., and Comb, M. J. (2002). Phosphoprotein analysis using antibodies broadly reactive against phosphorylated motifs. *J Biol Chem*, **277**(42), 39379–39387.