ANALYTICAL TECHNIQUES FOR THE IMPROVEMENT OF MASS SPECTROMETRY PROTEIN PROFILING

by

Richard Craig Pelikan

B.S. in Computer Science, University of Pittsburgh, 2002

M.S. in Computer Science, University of Pittsburgh, 2005

Submitted to the Graduate Faculty of

the Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2011

UNIVERSITY OF PITTSBURGH

INTELLIGENT SYSTEMS PROGRAM

This dissertation was presented

by

Richard Craig Pelikan

It was defended on

April 7, 2011

and approved by

Miloš Hauskrecht, Ph.D. Associate Professor

Department of Computer Science *and* Intelligent Systems Program

William L. Bigbee, Ph. D. Professor

Department of Pathology

Gregory F. Cooper, Ph. D. Associate Professor

Department of Biomedical Informatics *and* Intelligent Systems Program

Vanathi Gopalakrishnan, Ph. D. Associate Professor

Department of Biomedical Informatics *and* Intelligent Systems Program

Dissertation Director: Miloš Hauskrecht, Ph.D. Associate Professor

Department of Computer Science *and* Intelligent Systems Program

ANALYTICAL TECHNIQUES FOR THE IMPROVEMENT OF MASS SPECTROMETRY PROTEIN PROFILING

Richard Craig Pelikan, PhD

University of Pittsburgh, 2011

Bioinformatics is rapidly advancing through the "post-genomic" era following the sequencing of the human genome. In preparation for studying the inner workings behind genes, proteins and even smaller biological elements, several subdivisions of bioinformatics have developed. The subdivision of proteomics, concerning the structure and function of proteins, has been aided by the mass spectrometry data source. Biofluid or tissue samples are rapidly assayed for their protein composition. The resulting mass spectra are analyzed using machine learning techniques to discover reliable patterns which discriminate samples from two populations, for example, healthy or diseased, or treatment responders versus non-responders. However, this data source is imperfect and faces several challenges: unwanted variability arising from the data collection process, obtaining a robust discriminative model that generalizes well to future data, and validating a predictive pattern statistically and biologically.

This thesis presents several techniques which attempt to intelligently deal with the problems facing each stage of the analytical process. First, an automatic preprocessing method selection system is demonstrated. This system learns from data and selects a combination of preprocessing methods which is most appropriate for the task at hand. This reduces the noise affecting potential predictive patterns. Our results suggest that this method can help adapt to data from different technologies, improving downstream predictive performance. Next, the issues of feature selection and predictive modeling are revisited with respect to the unique challenges posed by proteomic profile data. Approaches to model selection through kernel learning are also investigated. Key insights are obtained for designing the feature

selection and predictive modeling portion of the analytical framework. Finally, methods for interpreting the results of predictive modeling are demonstrated. These methods are used to assure the user of various desirable properties: validation of the strength of a predictive model, validation of reproducible signal across multiple data generation sessions and generalizability of predictive models to future data. A method for labeling profile features with biological identities is also presented, which aids in the interpretation of the data. Overall, these novel techniques give the protein profiling community additional support and leverage to aid the predictive capability of the technology.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF EQUATIONS

GLOSSARY

AUC

Area under the Receiver Operating Characteristic Curve, a performance metric used to evaluate the decision-making ability of a predictor.

Biomarker

A naturally occurring biological component, such as a gene, protein or molecule, which indicates or characterizes normal biological processes. The presence or absence of a biomarker can also indicate a pathological process.

Dalton (Da, kiloDalton kDa)

The unit of measurement approximately equivalent to the weight of a proton or neutron. A molecule appearing at mass-to-charge ratio m/z generally weighs m Daltons.

MALDI-TOF-MS

Matrix-Absorbing Laser Desorption/Ionization Time-of-Flight Mass Spectrometry, a type of mass spectrometer technology.

Method

In general, I use the word Method to describe a technique used to accomplish a single task.

MS

Mass Spectrometer / Mass Spectrometry. See Section 2.3 for a detailed explanation.

Pathway

A set of genes which interact and depend on each other to achieve a biological function. A disturbance in part of the pathway, such as a mutated gene, may cause a disruption of the resulting biological function.

Procedure

In general, I use the word Procedure as a course of action consisting of individual Methods which are applied sequentially to a dataset.

Profile

A record in a dataset for a single sample which explains the molecular composition of that sample at the protein level.

QA/QC

Quality-Assurance / Quality Control data. The biofluid sample which generates these profiles comes from a common, pooled source. These data are intended to be used as references to check the consistency of the output of the mass spectrometer.

Raw

Refers to the state of data as it arrives from the mass spectrometer.

SELDI-TOF-MS

Surface-Enhanced Laser Desorption/Ionization Time-of-Flight Mass Spectrometry, a type of mass spectrometer technology.

Surrogate Biomarker

An element of the data which suggests the presence of a biomarker (see Biomarker above). Knowledge of the biological nature of the biomarker is usually incomplete.

PREFACE

In front of your eyes, you now see a document which is the result of hard work from many people. Although my name appears on the title page, there are several other names, belonging to great people, who enabled me to reach this point in my life.

I graduate with a doctoral degree at a time where public education seems to be less valued. I first want to mention that my achievements were not possible without the skills learned from my public education in Ewing Township, New Jersey. In particular, I would like to thank Mary-Lou Kramli and Merle Citron for their guidance, kindness and dedication to education. Mrs. Kramli was a trigger for my academic development and my love of foreign languages. Mrs. Citron nurtured my writing and creative expression. Without their efforts, I would have never had the willpower, courage or perception to attempt something so reckless as a doctoral degree. You have my sincere thanks.

In all, I have spent thirteen years at the University of Pittsburgh. During this time, I have only been taught extremely well. My advisor, Miloš Hauskrecht, stands as the centerpiece of my education. I was initially a student in his first undergraduate artificial intelligence course at Pitt, and though the class was probably the most difficult of my undergraduate courses, it bridged a gap between my fascinations with machinery and computers. A year later, Miloš took a huge risk in accepting me as a graduate student under his supervision. After all, I had no formal training in complex math, I was "only" a Master's student, and my academic performance in the first year of graduate school was incredibly humbling as I struggled with the shock of needing to change the way I learned and thought. Despite all this, I was able to succeed. Miloš was a great advisor to others and to myself. I am forever grateful for his advice, support and sacrifices which allowed me to pursue my doctoral degree. I now proudly carry his teaching onward.

I was lucky to have been put on a great research project. Having a latent interest in biology, I was able to grasp the early concepts of the newly emerging field of proteomics. This project was initially fostered by a collaboration between Miloš and Dr. James Lyons-Weiler, who was incredibly supportive and insightful, and provided generous financial support for the early stages of my graduate student career. The proteomics research group also included Dr. David Malehorn and Dr. William Bigbee, who later became a member of my dissertation committee. Dave deserves special mention here. He was responsible for generating, transporting and often explaining every nuance of proteomic data to me. His letters of recommendation were no doubt instrumental in my eventual acceptance to a doctoral program, and my awarding of a National Library of Medicine training fellowship. His fine taste for Belgian beers qualifies him as one of the finest men I know. This is taking a shortcut - I could easily devote an additional preliminary chapter on Dave and Jim, but in the interest of time, I depend on the words "Thank You", and hope you understand the rest. Dr. Bigbee has had an enormous amount of faith in me as a student and researcher, despite our vastly different fields of expertise. His insight and direction of this research project embodied everything that is good about science. Thank you!

The Intelligent Systems Program and Department of Biomedical Informatics have been godsends. I am thankful for the programs' support over the last seven years. I thank Dr. Janyce Wiebe for initial GSR support before beginning my work under Miloš. I thank the ISP and DBMI students for their friendship. I was so well-respected by DBMI students and faculty, that I felt as if I was among family when attending their events and conferences. In particular, I thank my other committee members, Drs. Vanathi Gopalakrishnan and Greg Cooper for their guidance throughout my doctoral studies. Dr. Cooper had a great influence on my professionalism and had a way of reminding me what was really important. His feedback often revealed my own shortsightedness, and I will always try to carry his perspective forward in my own future work. In our limited interactions, Dr. Gopalakrishnan provided amazing advice and support. As someone who truly values bioinformatics, it was a blessing to have her counsel throughout these years. I sometimes felt that I made the biggest leaps in my dissertation work after meeting with Vanathi, and she always instilled confidence in me. Thanks to both of you!

I learned a lot from my first two officemates, Drs. Branislav Kveton and Tomáš Šingliar. They had deep knowledge and understanding of many subjects which I probably should have known about, but didn't. I spent the better parts of my earlier years playing catch-up to them. It was only when they graduated, that I realized how much I valued their presence and respect. I regret that I didn't exactly fulfill the same role for my other officemates, Shuguang Wang, Michal Valko and Iyad Batal. However, even they were patient enough to endure my madness while I worked through trying times. Thanks to all of you.

I am lucky to have had the friendship of so many throughout my educational odyssey. Ian Barber, Scott Uhler and Brian Dykas were all good friends and partners in crime throughout the early years in New Jersey. I was happy to find a suitable analog in Pittsburgh. I thank Rob Shields, Chris Cole and Mike Jones for their camaraderie. I looked up to Drs. Lauren Kokai and Candace Brayfield as role models for balancing the difficulties of work and the distractions away from it. I will always fondly remember Drs. Charles Jackson, Niraj Welikala and Stefanie Hassel for our rock-climbing group and academic storytelling sessions on the roof of Doc's (only now do I see how appropriate that was!). In particular, those that I am most anxious to thank include Mihai and Diana Rotaru, Vlad and Roxana Gheorghiu, Panickos Neophytou, Socrates Demetriades, Gavin Hicks and Katrin Mascha. Wendy Bergstein and Toni Porterfield were awesome supporters from the administrative end. I thank them for their friendship and understanding when I was too lazy to meet certain deadlines. I could not have done this without any of you!

Obviously, my parents deserve much credit. I will always look up to my father Douglas for his unparalleled sense of ingenuity, and my mother Barbara for her sense of humor. Their love, advice and support was sometimes the only thing that kept me going. Beyond being fantastic siblings, I thank my brother Philip for his mechanical support and tireless sacrifices, and my sister Elizabeth for her fashion policing. I am proud of you all.

Lastly, it is most important to thank Yvonne Franke, whose unlimited patience, love and care helped assure that I would complete this most difficult journey. I will be happy to spend the rest of my life repaying my debt to you. I love you!

This thesis is dedicated to the Pelikan family of the past, present and future.

ACKNOWLEDGEMENT

Parts of this thesis were originally published elsewhere.

- Section 5.3.3 contains text and figures which are reproduced with permission from:

Richard Pelikan and Milos Hauskrecht. *Efficient Peak-Labeling Algorithms for Whole-Sample Mass Spectrometry Proteomics.* In *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Volume 7, Issue 1, pages 126–137.
Digital Object Identifier 10.1109/TCBB.2008.31, ©2010 IEEE.

- Section 5.3.2 contains text and figures which are reproduced with permission from:

Richard Pelikan, William Bigbee, David Malehorn, James Lyons-Weiler and Milos Hauskrecht. *Intersession Reproducibility of Mass Spectrometry Profiles and its Effect on Accuracy of Multivariate Classification Models.* In *Bioinformatics*, Volume 23, Issue 22, pages 3065–3072.
Digital Object Identifier 10.1093/bioinformatics/btm415, ©2007 Oxford University Press.

The author expresses his sincere gratitude for the generous funding provided through the following grants:

- Department of Defense (Washington, D.C. , USA) Grant W81XWH-05-2-006
- National Library of Medicine (Bethesda, MD, USA) Training Grant 5T15 LM007059-20
- Arts and Sciences Fellowship, University of Pittsburgh (Pittsburgh, PA, USA)

1.0 INTRODUCTION

Bioinformatics is rapidly advancing through the "post-genomic" era following the sequencing of the human genome. In preparation for studying the inner workings behind genes, several subdivisions of bioinformatics have developed. Each subdivision concentrates on a specific level of biological granularity: genomics covers the DNA sequence and mutation of genes, transcriptomics to study the expression of these genes, proteomics to characterize the proteins produced by these genes and metabolomics to study the economy of cells. Each of these subdivisions is supported by multiple similar technologies which allow for the rapid assessment of thousands of biochemical or genetic experiments, or *assays*. This process is generally referred to as *high-throughput screening* and their results are known as *high-throughput data*.

The popularity of high-throughput screening began with DNA microarrays. This technology is still currently used to perform gene expression assays, which measure how active genes are behaving under certain conditions. In an effort to find genes responsible for a particular condition, a researcher would take biological samples from several individuals exhibiting this condition, the *cases*, and several individuals who do not, the *controls*. Producing the high-throughput data for each sample, the researcher can locate genes which exhibit a robust difference between cases and controls; this result is called a *biomarker*. A biomarker can potentially serve to indicate a number of things: presence of disease, susceptibility of a condition, responsiveness to treatment or relationship among individuals.

Finding a gene biomarker does not necessarily pinpoint the cause of a condition, as is the case with many diseases. A gene is only the blueprint for proteins, which are responsible for constituting and controlling cells. The ultimate cause of a condition may be attributed to an abnormal level or state of a protein, which could be expected to be normal from simply looking at the expression level of it's producing gene. This can occur for a number of rea-

sons. Although a gene may be highly expressed, it does not mean that its associated protein product is produced correctly or at all. Furthermore, a single expressed gene may encode more than one protein. The decision about which protein is used, and how these proteins' functions may be changed, take place outside of the realm of gene expression. Therefore, a natural next step in the investigation for biomarkers was to expand high-throughput screening beyond genomics to proteomics.

Several methodologies for performing proteomic high-throughput screens exist, and are typically based on mass spectrometry technology. This type of technology is suitable for analyzing complex mixtures of proteins, such as those obtained from biofluids, e.g. blood, urine and saliva. Since these sample sources are more easily obtainable than organ tissue samples, it increases the opportunity for data collection and production. The resulting data, termed proteomic *profiles*, quantify how much, and depending on the technology, what type of proteins are present in the sample.

The proteome changes from cell to cell, and even from time to time, and the resulting complexity of proteomic profile data is naturally high. Generally, no limit is placed on the number of types of proteins that can be quantified by the instrument. A typical proteomic profile contains tens of thousands of measurements, or *features*, and the most sensitive technologies will produce profiles with hundreds of thousands of features. Each feature has the potential to be a biomarker, but many challenges stand in the way before a good claim can be made.

Lessons learned from microarray screening also apply to proteomics. The first is that no technology is perfect - stochastic noise permeates all high-throughput technologies through multiple ways. Changes in sample collection and processing, physical limitations of the sensitivity of the detecting machinery and natural variation in the biology of the samples contribute to errors and uncertainty in the data. Beyond this, thousands of features await analysis for their potential identification as biomarkers. Many of these features can be spuriously correlated with the difference between case and control samples. Perhaps only certain combinations of features can be reliable biomarkers, but many combinations exist, and it is infeasible to investigate all of them. Determining the ability of these biomarkers to discriminate future unseen samples as case or control is yet another issue. Finally, interpreting

the quality of biomarkers and their relevance to the discriminating condition in question is a necessary step in moving a discovery-minded analysis forward.

1.1 CONTRIBUTIONS OF THE THESIS

Much research in bioinformatics has been devoted to developing methods to address each of these issues. Although many ideas translate across the types of high-throughput screening techniques, every data type has its own unique quirks and caveats. The objective of this work is to develop a framework for analysis of high-throughput mass spectrometry data. The components of this framework are briefly outlined below with descriptions of the methods which contribute to those components.

- **Preprocessing**

  Preprocessing consists of a range of methods used to make high-throughput data easier to analyze. The goal is to minimize the amount of perceived imperfections in the data. These imperfections can be anything from missing or nonsensical values to unwarranted stochastic variation, systematic or otherwise. These imperfections can obscure useful information, or may cause downstream analyses to mistakenly identify spurious biomarkers. Preprocessing methods are employed to resolve the problems caused by these imperfections, with the expectation that most of the true biological information remains unaffected. I demonstrate new methods for removing noise in a modular fashion, while conserving valuable information.

  1. Metrics for the evaluation of individual preprocessing stages are proposed. Standard preprocessing in mass spectrometry typically only evaluates preprocessing holistically, after all stages have been completed. I investigate whether stagewise evaluation through these metrics will improve preprocessing as a whole.
  2. I evaluate preprocessing methods with the goal of preserving the discriminative information between case and control profiles as much as possible. Standard preprocessing techniques do not account for this. These methods will use the above-mentioned

metrics to improve further on preprocessing methods.

- **Feature Selection and Predictive Modeling**

  These are two intertwined topics which enable a computer to predict a disease state from the mass spectrometry profile data. The former involves methods which select or construct features from the proteomic profiles which appear to have diagnostic information. These features are then fed into a predictive model, which must be given training data to learn the relationship between the selected features and the actual disease state of the sample. The predictive model must be able to discover a robust relationship from the selected features, and furthermore, the features given as input must not be spuriously associated with the disease state. Thus, the combination of the methods used for selecting features and model is critical. I evaluate whether certain methods are better at automatically determining good combinations of feature selection and predictive modeling techniques.

  1. A variety of feature selection techniques are compared. This dissertation document discusses these feature selection techniques and explains why certain feature selection techniques are preferable to others.
  2. A decorrelating feature selection procedure is presented. This feature selection technique takes advantage of the naturally highly correlated data by restricting feature selection to potentially more informative features. In preliminary studies, this technique has performed comparably to other popular feature selection techniques.
  3. The Support Vector Machine (SVM) has performed admirably in preliminary studies, even with the basic linear kernel. I compare the SVM approach to additional kernel selection techniques. One alternative to the linear kernel is to learn a kernel, which is accomplished through linear combinations of existing basis kernels. Another alternative is to select a kernel from among many, based on statistical characteristics of the dataset at hand.
  4. I compare these kernel-learning approaches with a customized kernel for mass spectrum protein profile data. This kernel uses prior knowledge about gene and protein interactions to extract pathway information within the profiles. I investigate the extent to which any of these approaches have favorable qualities which are important

for mass spectrometry data analysis.

- **Interpretive Analysis**

  These are approaches for evaluating and interpreting the results of predictive models. Two major issues exist which discourage the acceptance of predictive models for routine use: (1) the lack of statistical reproducibility of the predictive model's performance in light of multiple sources of noise, and (2) uncertainty about the underlying biological reasons responsible for the predictive model's performance. I present novel methods for addressing these concerns and assisting the interpretation of preceding results.

  1. I introduce a method for assessing the significance of predictive modeling performance. This is a nonparametric method based on the permutation test, and allows the analyst to determine the strength of a predictive model's result.
  2. I introduce a set of methods for measuring reproducibility across separate data production sessions. This is particularly useful in determining whether a positive result from an analysis of mass spectrometry data can be repeated. These methods can also strengthen the confidence that the study design is robust against many types of noise resulting in differences in sample collection and data production.
  3. I introduce a method for labeling of features in mass spectrometry profiles with protein identifiers. This is a necessary task when dealing with many types of mass spectrometric protein profiles. This is especially true in the case of Time-of-Flight Mass Spectrometry data, the predominant data type used in this thesis (See section 2.3 for further clarification).
  4. I introduce methods for deriving biological interpretations from interesting patterns in mass spectrometry profile data. This is motivated by the wish to represent profiles as aggregate features which represent the biological functions ongoing in the sample.

While a similar framework for analysis may be assembled from existing pieces and methods, my framework builds upon knowledge and methods refined by analyses of multiple datasets, each with different characteristics and requirements. Thus, the methods introduced here are either new or are novel variations of existing methods. Their effectiveness will be demonstrated on datasets pertaining to a variety of conditions. For certain areas,

I provide guidelines and insight about effective analytical techniques for existing methods. Overall, the hypothesis to be tested by this research is that *the framework proposed here improves the analysis of mass spectrometry proteomic profiling data, in terms of automating the analytical process, improving predictive modeling performance and enabling more intuitive and useful interpretation of results.* The research reported here will be primarily useful to the bioinformatics community, and particularly those who use mass spectrometry proteomic technologies for disease prediction, risk assessment and treatment prediction. Validation methods incorporating prior information will be of interest to systems biology and translational researchers, who routinely seek patterns of differential biological activity as it relates to the condition of an organism.

Figure 1 is a simple depiction of the relationships between steps of analysis for proteomic profiling data. The methods in this thesis were developed with this analytical workflow in mind. Each of the steps are described in sequence throughout this document. Chapter 2 is devoted to describing the state-of-the-art techniques in mass spectrometry proteomic profiling. The data collection and production process is described in detail, to give the reader an appreciation for the nature of the data source. The types and limits of information contained in the data are also described. A list of available mass spectrometry datasets from a variety of disease conditions is presented. In addition, I also present a mass spectrometry simulator for the generation of artificial datasets. The simulator is an iterative improvement from a previously developed physical model of mass spectrometry [1] These datasets are used in the development and evaluation of the methods presented in later chapters. Finally, Section 2.5 describes mathematical notation relevant to the following chapters.

Chapter 3 is devoted to *Preprocessing*. The methods described in this chapter present heuristics that are useful for comparing and evaluating how well preprocessing techniques work. These heuristics are used to develop novel preprocessing steps which take into account the differences between case and control profiles, and adjust the profiles in a way that preserves these differences as much as possible.

Chapter 4 deals with the search for biomarkers. This process is governed by two intertwined topics, *feature selection* and *predictive modeling*. This chapter describes the basic methods for each step, and then introduces new methods which take advantage of observ-

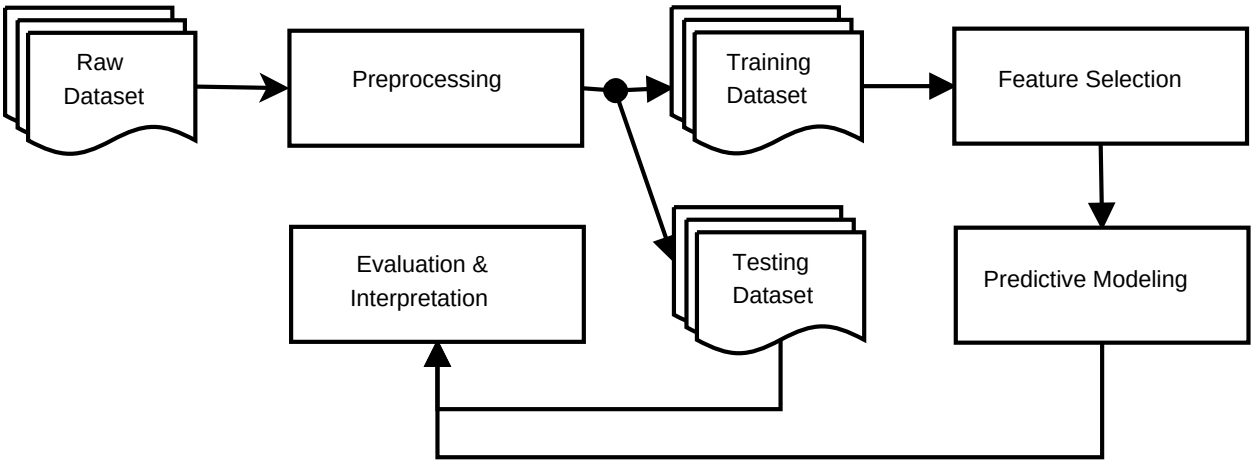

Figure 1: A flowchart demonstrating the relationship between steps of analysis of proteomic profile data.

able correlations in the data. Accounting for correlations allows flexibility and robustness in predictive modeling. An approach to automatic model selection is also investigated through approaches which try to learn the kernel of a Support Vector Machine. Additional approaches to model selection are also evaluated and should provide a basis for guidelines for model selection.

Chapter 5 discusses approaches for evaluating and interpreting the results of predictive models. As mentioned above, methods are presented to deal with the statistical and biological uncertainty inherent in mass spectrometry protein profiling data. Though a great deal of uncertainty surrounds the data source at any level, the presented methods work around this problem in order to elucidate information and bolster confidence in predictive models. This work also proposes a method which could relate elements of protein profile data to expression of biological pathways in the sample.

Chapters 3, 4 and 5 follow a similar structure (see the Table of Contents for an overview at a glance). Each chapter begins with a background section to motivate the chapter and describe the relevant challenges. This section gives enough information to understand the concepts developed in the methods section. The related work section describes the exist-

ing state-of-the-art research involved in the respective topic. The methods section details research advancements made by this thesis. Results evaluating the effectiveness of these described methods are given at the end of every chapter.

The thesis concludes with Chapter 6, which summarizes the thesis work and sheds light on future applications and extensions of the techniques presented in this thesis.

2.0 DATA

Proteomics is defined as the study of the structure and function of proteins. At a high level, one way of studying the function of proteins is to determine which are present or absent only in a group of patients with a common disease state. *Mass Spectrometry* (MS) is an analytical technique used to determine the elemental composition of a substance. In proteomic research, this substance is typically a sample biofluid consisting of, for example, blood, tissue cell lysate, urine or saliva. Many of these sample types can be obtained noninvasively, which facilitates a potential clinical screening process. A *mass spectrometer* is one of many devices which can be used to interrogate these samples to determine its constituents. A human biofluid sample is a complex mixture of proteins and other molecules. To identify and quantify each component of the mixture with certainty, any analytical technique would need to tediously separate each molecule in the sample and count it. Technologies such as gel electrophoresis, western blotting or liquid chromatography can separate proteins, but each technique has their limitations. The advantage of mass spectrometry is that a complex mixture can be analyzed with or without separation, and even small amounts of molecules will be measured. The time and cost of the analysis is also relatively low compared to other analytical techniques, which require non-reusable reagents, columns or films, larger amounts of sample and a significant amount of time for necessary chemical reactions to take place.

The process by which mass spectrometry works is described briefly and depicted in figure 2. The complex mixture sample is placed on an analytical surface. This surface may have properties which may emphasize the analysis of a particular class of proteins. The surface and attached sample are exposed to an energy source (for example, a laser beam). The surface transfers the energy source to the sample, causing individual molecules in the sample to become ionized and fly away from the surface. During this flight time, the ions (charged

molecules) are controlled and directed toward a detector plate. A more detailed description of the factors involved in the ionization process is given in the following section, which illustrates the complexity of the data production process.

2.1 BIOCHEMICAL ISSUES AFFECTING THE DATA GENERATION PROCESS

Many biochemical processes contribute to the types of peptides or proteins measured by the mass spectrometer. In particular, reparation of the analytical surface plays a critical role in the resulting data. The analytical surface captures molecules in the sample for analysis. For example, in Matrix-Enhanced Laser Desorption/Ionization (MALDI) mass spectrometry, the analytical surface is formed by a mixture of the biofluid sample and a "matrix" chemical, which crystallizes around the sample and allows laser energy to be transferred directly to molecules in the sample. The earliest and most popular method for preparing the surface is the "dried droplet" method [2]. Other methods were subsequently developed in attempts to smoothly and evenly distribute matrix-sample crystallization across the analytical surface. An investigation into the advantages of these alternative surface preparation methods revealed that two factors have the majority of the influence in the crystallization: a careful choice of the matrix chemical, as well as the amount of time allowed for the cocrystallization to occur [3].

If important molecules in the sample do not crystalize with the matrix, their presence will not be detected by the mass spectrometer, and they will be excluded from the data. Thus, it is important that molecules of interest are present in the matrix-sample crystals. The above investigations also established guidelines for matrix preparation when targeting particular types of molecules. For example, peptides greater than 3 kDa in mass crystalize better when the matrix solution includes formic acid and has a pH less than 1.8. Smaller peptides are better analyzed when the matrix solution includes no added acid [3]. Matrix solutions can be created to cover a more complete range of peptide masses, but it comes at a cost of losing potential information where a more optimized protocol is established.

If a matrix-sample crystal is hit by the laser, we must hope that the molecules in the sample become properly ionized. Only ions will fly away from the analytical surface and become measured by the mass spectrometer. Ideally, a peptide molecule will be measured intact and singly-charged. Too much energy can cause a peptide to fragment, breaking it into smaller peptides. Too much electric charge can cause the peptide to be measured at different areas of a protein profile. The electrical charge of the ion plays differing roles in various mass spectrometers. Depending on the amount of ionizing energy used, molecules will acquire or lose one or more electrons. Since the ions are repelled by their charge from the analytical surface by a constant field, adding additional charges will increase the flight speed of the ion proportionally. A doubly-charged ion will fly twice as fast as with a single charge. The ion mass and charge are combined to form the *mass-to-charge ratio*, or m/z ratio. Creating an analytical procedure which controls the amount of expected charge on a peptide would help to stabilize the analysis. Ideally, if every peptide is only singly-charged, then a peptide will be measured by the spectrometer by its mass alone. If a doubly-charged peptide is obtained, it will also be measured at half its mass.

Unfortunately, the ionization process in mass spectrometry is difficult to explain with a simple physical mechanism [4]. Ionization depends on the laser wavelength, pulse energy and pulse length, which can change from lab to lab. In addition, the relationship between these parameters and the varying sample preparation methods has yet to be extensively studied. Finally, certain classes of peptide molecules are more likely to ionize than others. In certain types of mass spectrometry experiments, the rate of successfully ionized molecules to neutral molecules can be as low as 1 out of 10000 [4]. These factors combine to make a very complex picture of what is ordinarily a small and underappreciated step in the data production and analysis of protein profiling data. However, it is important to discuss, as it shows the complexity of the data source and the need for analytical techniques which can deal with the inevitable imperfections in the data.

Prior knowledge about the chemistry of the sample and the types of molecules expected can help to guide the data production process. Primarily, the acidity or basicity of the desired molecules seems to be the most important factor influencing ionization. Matrix chemicals with appropriate proton affinities or gas-phase basicities can be selected in order to encourage

stable ionization behavior, and guidelines have been established in order to support these behaviors [4]. Furthermore, new matrix chemicals can be designed if the available chemicals are not appropriate for ionizing the desired peptide molecules. Overall, there are many characteristics which play a part in the ionization process. The most important thing is to ensure that as many ions as possible can be generated. Careful design of the matrix chemical, combined with careful selection of the spectrometer's laser settings, can help to ensure more consistent and useful data production [4].

2.2 DATA USED FOR THIS THESIS

This thesis is primarily concerned with *Time-of-Flight Mass Spectrometry* (TOF-MS), which is one method the mass spectrometer uses to calculate which ions are reaching the detector. The detector plate records a series of collision events, which consist of the electrical charge and time of impact of ions flying into the detector. The time of impact since the start of the analysis is important. Heavier, more massive molecules will fly slower than their smaller, less massive counterparts. From the length of the ions' flight path, the mass spectrometer calculates the mass of the colliding ion. The resulting data from a mass spectrometer is a list of m/z ratios and the number of ions detected with that ratio.

A *proteomic profile* is a data record produced by a mass spectrometer from a biofluid sample. Proteomic profiles can be visualized in a number of ways; a common view is displayed in figure 3. M/z ratio is presented along the x-axis, and the y-axis measures the *relative intensity* of ions present at a particular m/z ratio. Relative intensities are used instead of ion counts, due to a physical limitation on the number of ions the detector plate can sense simultaneously. There is a maximum value which the detector can sense, and all other values recorded by the machine must be normalized to this value. Figure 3 gives the illusion of a continuous line, but in reality, the measurements are individuals, and the distance between observed m/z ratios can vary. However, the continuous-line interpretation will facilitate many aspects of the data analysis, as will be seen in Chapter 3. From this point forward, a m/z ratio and its associated relative intensity will be interchangeably referred to as a *feature*

of the proteomic profile. These types of profiles typically consist of tens of thousands of features.

Mass spectrometry is an imperfect technology. It should be clear that many interpretation errors can take place. For example, a molecule may be represented by two separate features when ionized with a single and double charge. Two (or more) molecules could potentially have the same m/z ratio, causing their intensities to overlap. And amongst these uncertainties, it should be noted that very little information about the biological nature of the features is generated by the mass spectrometer. Arguably, this is the largest problem facing interpretation of MS proteomic profile data. Cutting-edge technology is being developed for the elucidation of identities of protein and peptide molecules. Briefly, ions in the flight path are subjected to another energy source, causing the molecule to be reduced to fragments. The fragments are then measured by the detector plate, and the resulting fragmentation pattern is compared to databases of known protein fragmentation patterns. Matches can often be found,however, even these methods are far from perfect. While it is possible that MS technology will improve in the future to identify all proteins, techniques can still be developed for the technology that works well now, and these techniques can be designed to incorporate additional information as the supporting technology evolves.

2.3 AVAILABLE BIOLOGICAL DATA

For the purposes of this thesis, I have assembled a collection of proteomic profile datasets for analysis. The majority of this data was produced by a *Surface-Enhanced Laser Desorption/Ionization* (SELDI) TOF mass spectrometer. Compared to other mass spectrometry techniques, SELDI uses a specialized analytical surface which can have particular affinities for certain protein types. Proteins that do not bind to the surface are washed off before ionization. This reduces the number of different protein types that can enter the mass spectrometer, and can reduce the complexity of the resulting data. The particular SELDI-TOF model used is the Ciphergen PBS-II SELDI-TOF mass spectrometer (Ciphergen Biosystems, Fremont, CA, USA). The available *Matrix-Assisted Laser Desorption/Ionization* (MALDI)

TOF data is produced by either a Voyager-Elite mass spectrometer (Applied Biosystems, Foster City, CA, USA) or Bruker Clin-Prot II mass spectrometer (Bruker Daltonics, Billerica, MA, USA).

Most datasets were produced either by the University of Pittsburgh Cancer Institute (UPCI, Pittsburgh, PA, USA) or the University of Pittsburgh Medical Center (UPMC, Pittsburgh, PA, USA). These datasets were produced with careful, strictly-controlled study designs in order to reduce the potential for bias and confounding. Each dataset contains data from samples of diseased (*case*) and healthy (*control*) individuals. Cases and controls were processed by the laboratory on the same machine. When samples could not be processed simultaneously, the order in which samples are processed by the laboratory is randomized to reduce the possibility of spurious temporally-created biomarkers. To reduce the effect of confounding between the two sample groups, the case and control samples are matched by variables such as age, gender and smoking history when applicable. This means that each case sample is matched with a control sample from an individual with similar clinical characteristics.

Several disease conditions are demonstrated in the various datasets. Table 1 lists the datasets as well as the disease condition, number of cases and controls, year produced and originating laboratory. The disease conditions cover both diseases which are similar (Ovarian, Prostate, Melanoma, Pancreatic and Lung Cancers) and dissimilar (hepatitis C, kidney necrosis, diabetes and dental caries). With the diversity of conditions studied, it will be possible to perform robust comparisons between methods, to ensure that they are not too strongly biased to one class of diseases.

2.3.1 Important dataset notes

Several datasets have special properties which allow the evaluation of certain experiments.

2.3.1.1 Prostate cancer dataset The prostate cancer dataset [5] was produced by researchers at the National Cancer Institute (NCI, Bethesda, MD, USA) Clinical Proteomics Program in 2002. This dataset suffered from unintended mistakes as a consequence of the

proteomic profiling research field being in its nascent and explosive stage. The case and control samples were processed on separate days. This had the unintended effect of introducing a biased signal artifact which essentially labeled the case and control samples perfectly [6]. While results from this dataset should not indicate great success at predicting prostate cancer, it is still a useful tool in order to demonstrate the effectiveness of certain methods in the face of overwhelming bias. The authors claim that this data was never intended to reflect the success of a potential screening platform. However, the flaws in this and another study [7] cast a pall of doubt upon the protein profiling technology [8]. Despite the rough beginning, protocols became more developed and technologies such as those presented in this work began to develop, in order to support the promise of this technology.

2.3.1.2 Vanderbilt/UPCI Lung SPORE datasets This source of data reflects a large set of samples processed at different locations, under different technologies. A larger set of lung cancer patients' serum samples were collected by the Vanderbilt University Medical Center Clinics (Nashville, TN, USA), the Nashville VA Medical Center (Nashville, TN, USA) and the UPCI. These samples were initially investigated at Vanderbilt University [9] under the MALDI technology, and then the same samples were shipped to the UPCI to be processed under SELDI IMAC (Immobilized Metal Ion Chromatography) and WCX (Weak Cation eXchange) technologies. The intersection of these three datasets were a set of 134 case and 104 control profiles. These 3 datasets give us a unique look into how a sample can be expressed across different MS platforms as well as different locations and different times.

2.3.1.3 UPCI Lung Cancer II dataset This dataset featured four data production "sessions" in which a subset of the samples were repeatedly reprocessed after spending time in a freezer. This was done in an effort to determine whether sample degradation would play a large role in observing clinically relevant patterns. In most cases, table entries corresponding to this dataset reference the initial data production session, which contained the largest set of samples out of the four sessions. In Chapter 5, all four sessions are analyzed in greater detail in the context of a reproducibility study.

2.4 SYNTHETIC DATA

In addition to the biological data summarized above, the ability to simulate data sets is also available. The simulator is based on a physical model of the TOF-MS process [1], and is able to simulate data from any TOF-MS system. While the simulator model proposed in [1] is thorough in parameterizing many aspects of the TOF-MS system, I believe there are errors in the model, so I derived new sets of equations to convert masses to times-of-flight. A detailed description of the simulator is given in Appendix From the above section, one can see that the mass spectrometer is constructed with three main components: the energy source, the flight chamber and detector plate. The simulator parameterizes each of these components in order to match the configuration of any TOF-MS system. For the purposes of simulation, these parameters are chosen to match those of the Ciphergen PBS-II SELDI-TOF-MS system. The amount of noise in the MS components can also be adjusted to demonstrate the effect of inaccuracy in the instrument on the amount of proteins in the simulated sample. Full details of the simulator and its parameters are given in Appendix A.

In biological data, we are always uncertain of the types and amounts of molecules that are present in the sample. A major advantage of simulating data is that the constituents in the sample can be controlled. Thus the simulated data can be used as a tool to study the effects of noise on individual measurements, and develop realistic expectations of how well the MS instrumentation can represent mixtures of molecules in the data. The effects of noise are described in greater detail in Chapter 3. The choice of peptides to be included in the simulated sample is arbitrary, and can fit the need of the task when a particular reference point is needed.

Mixtures of proteins and peptides can be added to the simulated sample if they are present as entries in the UniProt database [10]. The UniProt peptide identifiers are in turn used to retrieve amino acid sequences for these proteins. The expected mass of the peptide is computed as the sum of the average isotopic masses of the amino acids in the given sequence, in addition to the average isotopic mass of a single water molecule. Post-translational modifications can also affect the peptide's sequence, and these are taken into account when calculating the peptide mass. Signal peptides from complete protein sequences

are removed, and when documented in the UniProt entry, the mass of a post-translational modifying molecule is added to the peptide's expected mass. The relative abundances of each peptide in the simulated mixture is provided. Thus, the abundances and expected masses of the peptides can be used to study the ability of the mass spectrometer to detect an appropriate amount of peptide at the correct *m/z* ratio. Additionally, multiple simulations can be produced to study effects of noise on the MS system's ability to reproduce profiles. In this thesis, simulated data is used in Section 5.4.3.1 to demonstrate the effectiveness of a peak-labeling system.

2.5 MATHEMATICAL NOTATION

The following notation and terminology is useful for understanding the technical details of this thesis.

1. A *dataset* D is a collection of n proteomic *profiles*, indexed by the variable i.
2. Each proteomic profile $\mathbf{X}$ has d *features*, indexed by the variable j. It is associated with a class label, Y,which is either 0 (healthy//control) or 1 (diseased//case). The positive (or *case*) class is marked as $\mathbf{X}^+$ and the negative (or *control*) class is marked as $\mathbf{X}^-$.
3. The j^{th} feature in the i^{th} profile f_{ij} occurs at *m/z* position x_{ij} and has intensity y_{ij}.
4. The average intensity of the j^{th} feature is written as μ_j. The standard deviation of the j^{th} feature is written as σ_j. Superscripts of + or - indicate means or standard deviations within a single class.
5. The average profile, consisting of all features $\mu_1 \cdots \mu_d$ averaged over all profiles is referred to as μ, and can be computed across all positive (μ^+) or negative (μ^+) profiles.

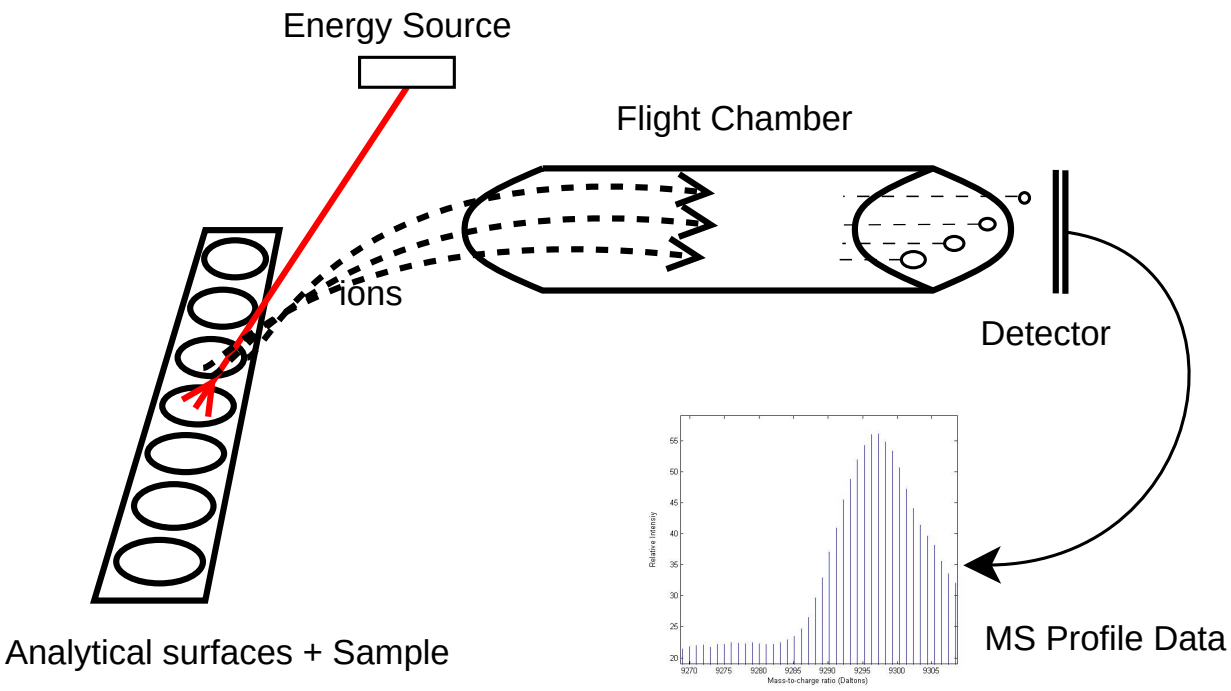

Figure 2: A simple diagram of the mass spectrometry data production process. A biofluid sample is deposited on an analytical surface. The sample is ionized by an energy source, causing protein ions to fly through a flight chamber. Lighter, smaller molecules fly faster than heavier molecules through the flight chamber. When they hit a detector plate at the end of the chamber, the mass spectrometer records the amount of detected ions. By measuring the time taken to fly through the tube, the masses of the ions are calculated and a "peak" feature at the appropriate m/z value is created in the data.

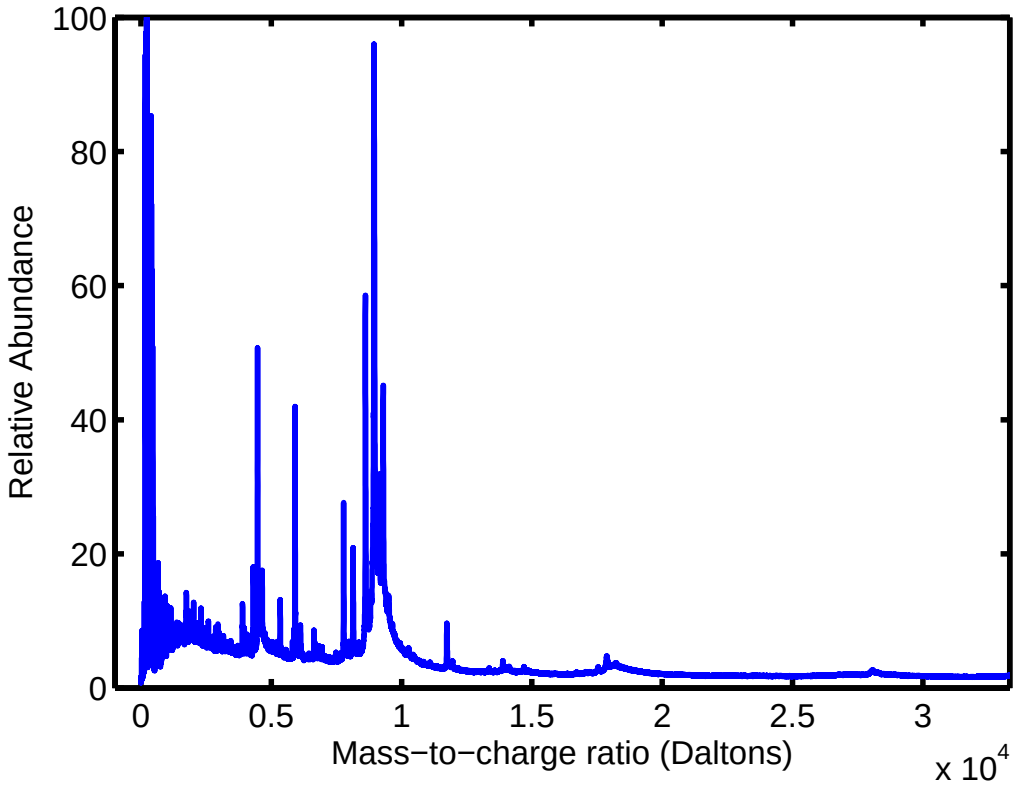

Figure 3: An example of a proteomic profile produced by SELDI-TOF mass spectrometry. The x-axis indicates the mass-to-charge ratio of molecules present in a sample for pancreatic cancer. The y-axis reflects the relative intensity, or abundance, of those types of molecules.

Table 1: Available Biological Datasets. The disease condition, originating data source, year of production, number of cases and controls, and relevant MS platform options are listed for each dataset used in this thesis.

Disease	Source	Year	Cases	Controls	Platform	Chip Affinity
Ovarian cancer	Clinical Proteomics	2003	100	100	Ciphergen SELDI-TOF-MS	IMAC
Prostate cancer	Databank	2002	63	253	Ciphergen SELDI-TOF-MS	IMAC
COPD	UPMC	2004	25	16	Ciphergen SELDI-TOF-MS	IMAC
Interstitial lung disease	UPMC	2004	19	16	Ciphergen SELDI-TOF-MS	IMAC
Hepatitis C	UPCI	2005	28	28	Ciphergen SELDI-TOF-MS	IMAC
Diabetes	UPCI, Harvard	2006	18	18	Bruker ClinPROT	MALDI
Melanoma I	UPCI	2003	25	26	Ciphergen SELDI-TOF-MS	IMAC
Melanoma II	UPCI	2004	66	72	Ciphergen SELDI-TOF-MS	IMAC
Lung cancer I	UPCI	2004	112	106	Ciphergen SELDI-TOF-MS	IMAC
Lung cancer II	UPCI	2004	366	339	Ciphergen SELDI-TOF-MS	IMAC, WCX
Lung cancer III	Vanderbilt & UPCI Lung SPORE	2005	134	104	Voyager Elite	MALDI
Lung cancer III		2005	134	104	Ciphergen SELDI-TOF-MS	IMAC
Lung cancer III		2005	134	104	Ciphergen SELDI-TOF-MS	WCX
Breast cancer	UPCI	2005	30	30	Ciphergen SELDI-TOF-MS	IMAC
Pancreatic cancer I	UPCI	2004	32	23	Ciphergen SELDI-TOF-MS	IMAC
Pancreatic cancer II	UPCI	2004	72	72	Ciphergen SELDI-TOF-MS	IMAC, WCX
Scleroderma	UPCI	2005	47	47	Ciphergen SELDI-TOF-MS	IMAC

3.0 PREPROCESSING

Preprocessing refers to a range of methods used to make high-throughput data easier to analyze. Since each high-throughput technology differs in its method of data acquisition, the techniques required differ between applications. The goal in any case is to minimize the amount of perceived imperfections in the data. These imperfections can be anything from missing or nonsensical values to unwarranted stochastic variation, systematic or otherwise. Figure 4 illustrates four MS profiles from the same sample source, but produced at four different times. Changes are apparent in the intensity and shape of the MS signal, which should ideally appear exactly the same across all four replicates. Downstream analysis techniques could identify these imperfections in a way which would discriminate between profiles which should otherwise be identical. Preprocessing methods are employed to resolve the problems caused by these imperfections, with the expectation that most of the true biological information remains unaffected. These methods are applied separately in stages to address the different types of imperfections occurring in the data. The next section briefly explains the most popular preprocessing stages and the errors that they attempt to correct.

3.1 BACKGROUND

3.1.1 Stages of Preprocessing

Calibration refers to the process of ensuring that the quantity measured by an assay is truly reflected by its feature value. In a microarray experiment, this refers to translating feature values into estimates of mRNA transcript abundance in the sample. In TOF-MS

data, calibration can refer to mapping an observed time-of-flight value to an appropriate m/z value, or to the more specific process of *alignment*. In any application, calibration helps to ensure that the reported value is not misrepresented. Alignment is more particular to proteomics, where peak features similar in shape across separate profiles may be shifted to different locations along the m/z axis. Figure 5 displays two profiles from QA/QC serum which require calibration. Their shape is nearly identical, but the peak positions appear to be shifted linearly on the x-axis. The solid line could be shifted to the left to make the shapes overlap neatly. Alignment can be used to bring these profiles into agreement.

Variance stabilization is a process used to decouple the dependance of a feature variable's variance on its mean. In many applications, features with the largest values exhibit the greatest variance. Thus, a strongly expressed assay may incur more random noise than a weaker one, detracting from its informational value. By applying a variance-stabilizing transformation (such as taking the log or cube-root of the feature values), the features tend to have a more constant variance, independent of the value of the feature. This makes adjusting for systematic noise more convenient, as the relationship of the noise process becomes similar across features. In any application, the aim is to reduce the effect of a multiplicative bias in the feature values.

Baseline correction helps to ensure that all feature values are recorded with respect to a baseline of 0 (or other suitable constant) to clearly distinguish values as being measured features versus 'default' values produced by the data collection equipment. In TOF-MS data, this is easily seen in an unpreprocessed profile, which seems to have a constant, nonzero baseline. Figure 6 displays an example of such a phenomenon in a SELDI-TOF-MS profile. In microarray data, this is referred to as *background adjustment*, and accomplishes the removal of detectable signal occurring due to reasons other than the correct transcripts binding to the probe surfaces. In any application, baseline correction assures that the scale of comparison for values has a common starting point by removing an additive bias. Common methods simply subtract a constant from every feature value. More involved methods may fit a function to the existing baseline and subtract this function to achieve a "flat" baseline. The result of the process is similar to that seen in the right panel of Figure 6.

Normalization adjusts all feature values to conform to the same scale. This process

has varying levels of granularity. Microarray users typically refer to normalization as the process that reduces variance between experiments in different datasets. In TOF-MS data, it typically refers to removing bias from an individual profile by rescaling all feature values within that profile. Figure 7 displays two profiles from QA/QC serum which should be identical. However, a difference in either the level of sample or strength of the equipment causes one profile to appear more intense than the other. Normalization would rescale the data in both profiles, so that they would appear to be on the same scale. A common approach is to normalize all features to a [0 1] range. In TOF-MS, profiles can be rescaled based on the total sum of feature values across the profile (*total ion current, or TIC, normalization*). In any application, normalization attempts to mold the data to a conformed range of variance.

Smoothing serves to eliminate a high frequency noise component in the signal. There are multiple ways that this noise can be removed. A simple solution, *moving average*, involves replacing the feature with an average of its neighboring features. Figure 8 displays an example of moving-average smoothing on a SELDI-TOF profile. The signal in the left panel carries a frequent, jagged landscape. By smoothing the signal, this variation is averaged over a small area of points along the m/z axis. The result is a smoother signal with less variation. A more intricate approach might use a kernel to give close features a particular shape. In TOF-MS profiles, an ideal peak might have a Gaussian shape, and therefore fitting close values to a Gaussian kernel might be best. Additional methods might employ a signal-to-noise filter to remove the high-frequency random component. More recent work has used wavelets to model a complex mixture of signals as a composition of simpler, smooth signals. Regardless of the method used, it is never clear where to draw the line between random noise and true biological information. A smoothing operation must be able to remove variance, but not so aggressively that the features become fuzzy and redundant.

3.1.2 The order of preprocessing stages

Most applications do not require preprocessing methods that fall outside of these steps. However, the selection and order of steps used is dependant on the type and quality of

data. In the case of TOF-MS data, performing alignment before calibrating peaks along the m/z axis is ideal. Although variance stabilization, normalization and smoothing all reduce variation, they have specialized roles which might make them more effective in particular orders. For example, variance stabilization can be seen as a type of normalization, which removes a multiplicative bias on feature values. This must be done before the additive bias is removed through baseline correction. Afterwards, additional rescaling through normalization may take place, as not every rescaling operation will reduce the dependence of feature values' variance on their means. Smoothing could be applied at any time to improve the consistency of data. However, performing it too late may cause interesting variation to be obscured and averaged out. Any preprocessing step entails removal and addition of information. Thus, any preprocessing step carries a risk, and should be kept to a minimal level of aggressiveness to preserve as much interesting information as possible. In order to adjust the harshness of each preprocessing step, they are usually controlled by a choice of parameters (such as the size of a moving-average window), or functions (such as an exponential function for modeling a shifted baseline).

3.1.3 Related Work

The list of literature on high-throughput data preprocessing is substantial. It is typical that each lab generating data will have their own preferred method, usually in addition to at least one other 'default' method suggested by the lab equipment manufacturer. Despite the differences, the individual steps remain similar. The majority of differences are brought about by how the steps are tuned to fit the quirks of the studied data. The following is a summary of the most popular methods used for preprocessing data. Each technique will be used later on as part of a comparative framework for evaluating preprocessing methods.

3.1.3.1 Preprocessing Techniques Preprocessing has been applied to data from many fields, and should not be surprising that many techniques stem from simple methods. High-throughput data is intended to demonstrate a signal, but suffers from much noise. Methods in signal refinement have been simply transplanted from their native domains and applied to

high-throughput data. Signal-to-noise filtering, originating in electrical engineering applications, has been refined for the purposes of smoothing and *variance stabilization*. The latter is commonly performed through either a fractional-power transformation [11] or more commonly, log transformation [12,13]. Applying the latter transformation will artificially inflate variance of very low feature values, as well as necessitating truncation of negative values. A generalized log-transform was later developed to address these problems by parameterizing the logarithmic transformation for individual assays [14]. Going one step further, assay-specific variance-stabilizing transformations can be created by learning an appropriate transformation function [15]. This function is derived by characterizing the relationship between the mean and variance of a feature's values. This enables a wider variety of transformation functions to be used. Another key advantage is that the method allows for the incorporation of multiple replicates without needing to perform a linear normalization (averaging) across feature replicates.

Smoothing techniques are relatively simple in nature and application. Many high-throughput data analyses use local statistics to conform a feature's value to its immediate neighbors. These include simple approaches such as moving average [13], median and geometric mean filters [16]. Techniques used to smooth time-series data have also been applied to proteomic profiles, due in part to their similarity in visualization. The Savitsky-Golay filter [17,18] is considered a very popular choice for smoothing MS data. More recent methods have been developed based on the Fourier transformation [19] and wavelet transformations [20,21]. These transformations break the proteomic signal into a combination of signals, with the intent to separate noise from the true signal. Wavelet transformations can produce multiple wavelet basis functions, which have the advantage of adapting to differences in scale and local signal structure.

Baseline correction methods generally assume that signal is constructed from a linear combination of a noise signal, true signal and baseline (zero) signal. The true signal is affected by systematic noise as a result of the imperfect machinery used to measure a feature's value. In the case of microarray data, this pertains to two separate effects; nonspecific binding of genes to an inappropriate probe artificially inflates or deflates expression values for those genes. Furthermore, the optical scanner imparts noise to the value of a probe's reading. The

original solution involved measuring mismatch probes, which are designed with a small defect, to measure the amount of non-specific binding. The background adjustment is done by subtracting this amount from the reading of the true-match probe. Although this ignored the amount of noise introduced by the scanner, later models were able to separate contributions of scanner and nonspecific binding noise. Affinities between gene targets and probes were estimated based on their nucleotide structure, which permitted models for estimating the nonspecific binding. Additional noise is modeled as a separate additive component, modeled through a distribution learned from data [22].

Baseline correction methods for proteomic spectra developed with a similar history. Early techniques concentrated on detecting 'peak' features and subtracting them from the raw signal to determine the shifted baseline [19, 23]. In these methods, the baseline is drawn piecewise by finding a local minimum or median (to reduce noise) within a sliding window of variable size. Alternative methods separated the peak-finding task from computing the baseline, either by smoothing out the peaks [24], or by ignoring them altogether, using only the local minima while ensuring a monotonically decreasing baseline [25]. The latter is a simple and popular method, because it allows the decomposition of noise and true signal at a later stage. A recent technique fits an exponential function to local medians which remain below a smoothed version of the raw signal [24]. This can help to eliminate some of the noise. However, remaining noise above the baseline may become more difficult to characterize and remove. An "orthogonal background subtraction" method was developed [16] by characterizing noise through Principal Component Analysis [26]. The top two components were used to estimate noise contributions in an area of signal which was known to consist only of baseline. These learned components were then used to remove the baseline from the entire profile.

Normalization techniques are not particularly abundant in the literature, as their purpose is simple: to enhance the appearance that data come from the same scale. The method outlined in [27] rescales all values to the [0 1] range. Similarly, the 10^{th} and 90^{th} percentiles of profiles have been mapped to 0 and 1, with linear interpolation between [28]. Quantile normalization [29] is a method which rescales data so that feature values in individual data records come from the same distribution. This method was developed for microarray analysis,

but has since been used for normalizing proteomic spectra [30]. Nevertheless, the most common method of normalizing proteomic data tends to be the *total ion current* (TIC) correction [19, 25], which normalizes all feature values of a spectra by the sum of intensities within a region of the spectra. This region can be as wide as the entire spectrum, but is typically restricted to a region which appears to have a strong signal-to-noise ratio. A similar method is global mean normalization [31], which assures the average feature intensity is the same across all profiles.

Alignment of proteomic spectra is commonly performed to ease comparison of features within and across datasets. The dynamic time-warping algorithm [32] was initially used to measure and increase the similarity between two proteomic signals over time-of-flight. However, the large size of proteomic profiles ($\gg$ 10000 positions) is often prohibitive for the memory and computational requirements of the underlying dynamic programming mechanism. Restricting the maximum distance of edits between signals [33] alleviates this problem somewhat. As an alternative, the dynamic programming approaches can be done away with completely by using parametric and nonparametric methods [33,34]. These methods restrict the warping function to a low-order polynomial. The parameters of the polynomial are fit via regression so that an appropriate distance metric between the aligned signals is minimized. To save computational costs, a method has been developed [35] to split the signal into segments which are dealt with separately, but also forced to agree when merging them at the end. One drawback of these approaches is that they require a "template" profile to which other profiles are aligned. To avoid biasing the alignment with a poor choice of a template profile, a probabilistic model was developed [36] to perform multiple alignments simultaneously with a hidden Markov model (HMM). The template becomes a sequence of states, each of which reflect a distinct feature in the profile. This is learned from the training data in a way that maximizes the probability that the HMM can produce the training data from the template state sequence. Afterwards, a profile is aligned to the template by its probability of which state each feature is in.

3.2 METHODS

As no technology can be absolutely perfect, high-throughput data requires preprocessing to assure that the effects of internal and external noise are mitigated. Preprocessing necessarily affects the downstream analysis of high-throughput data [37]. Predictive models learned on differently preprocessed data may be substantially different in terms of which features are used for diagnosis. This in turn affects the interpretation and validation of interesting models. This may suggest that preprocessing methods consistently emphasize the same features. However, preprocessing cannot correct for all variations in laboratory protocols which may be responsible for different sets of informative features. Thus, preprocessing methods must do their best to maintain any differences between classes, if they exist, so that analysis routines can examine all reasonable possibilities. Concurrently, they must also deal with the inherent stochasticity of the data collection process by benignly removing noise. Thus, profiles belonging to the same class must remain as similar as possible, to reduce the chances of spurious features arising in the feature selection phase.

This dissertation intends to develop heuristics and methodology for the evaluation of preprocessing techniques. No means exists to objectively compare sets of preprocessing methods for mass spectrometry data. The generic heuristics given here are applicable to any type of MS data, and the methodology stresses an important aspect which is often overlooked in the development of many preprocessing methods. This aspect is that the global task of achieving good prediction must be balanced against the localized removal of noise. The evaluation of these techniques are demonstrated on real biological data. The following sections describe a methodology for automatically evaluating and applying preprocessing techniques which seem to best improve discriminative information while removing targeted sources of noise. These methods are then compared for their effect on downstream performance.

3.2.1 Evaluating Preprocessing Steps

The sections below introduce the *Standard Automatic Preprocessing* procedure, henceforth abbreviated as SAP. This procedure attempts to maximize the discriminative signal remain-

ing after a preprocessing method at a tradeoff with how well noise is removed from the signal. At each stage of the preprocessing, we use two kinds of heuristics to determine a preprocessing method's performance at retaining discriminative signal and removing noise. A third parameter acts as a security measure against methods which may cause these scores to be circumvented.

In general, we are interested in the task of using proteomic profiles to predict disease. This classification task is central to any analysis of protein profiling data. Preprocessing consists of many stages, with each stage targeting a specific source of noise. Sometimes, these sources of noise are more easily defined and their effects quantified. Other sources of noise are hard to describe accurately and are more difficult to quantify. In this case, it is increasingly important to focus on the original predictive task to assist us in determining how well a noise source has been dealt with.

The Discriminative Estimate (DE) score is a global metric which stays constant throughout all stages of the preprocessing. The goal of this heuristic is to measure how easily the case and control profiles can be discriminated after a preprocessing method has been applied. We can calculate the DE score through many means. In the experiments below, I use the 10-fold cross-validated AUC of a support vector machine, evaluated on the internally split training data. Other possible ways to calculate the DE score include the following options:

- Summing the univariate scores of the top n discriminative features, where n is an index into features sorted by their Fisher score in descending order

$$DE = \sum_{i=1}^{n} \frac{\mu_i^{(+)} - \mu_i^{(-)}(i)}{(\sigma_i^{(+)})^2 + (\sigma_i^{(-)})^2} \tag{3.1}$$

- Measuring the ratio between average Euclidean distance of profiles within class to the average Euclidean distance of profiles between classes (as in equation 3.2 below).

$$
\begin{aligned}
DE &= \text{diff}_{\text{inter}}/\text{diff}_{\text{intra}} \text{ ,where} \\
\text{diff}_{\text{inter}} &= 1/(2n^{+}) \sum_{j=1}^{d} (\mu_j^{+} - \mu_j^{-})^2 \\
\text{diff}_{\text{intra}} &= 1/(2n^{-}) \sum_{i=1}^{n_+} \sum_{j=1}^{d} (\mu_{+} - f_j)^2 + \sum_{i=1}^{n_-} \sum_{j=1}^{d} (\mu_{-} - f_j)^2
\end{aligned}
\tag{3.2}
$$

The following subsections present methodology for evaluating the best technique at every stage of the preprocessing routine. However, each stage may have a different criteria to evaluate the goodness of that particular stage's contribution to the complete preprocessing routine. I call these local criteria *stagewisescores*. There may be tradeoffs between local stagewise scores and the global DE score, which is intended to reflect the goodness of the entire preprocessing routine, rather than each individual step. Therefore, in order to choose the best preprocessing method at any stage, I create so-called *Stagewise Performance* curves (Henceforth referred to as SP-curves) akin to ROC curves. For each preprocessing stage, multiple preprocessing methods will be evaluated using a stage-specific criterion. This criterion is computed along with the resulting effect on the data's DE score. A curve is generated from the points, so that the x-axis represents the quantity 1-DE, and the y-axis represents the stage-specific criterion, which is rescaled to the range [0 1]. The "optimal" method is chosen as the one whose scores are closest to the point (0,1). This gives a similar interpretation to ROC curves, in which a "perfect" classification system's ROC curve will reach the point (0,1). This selection system has the effect of equally weighting the DE score and the stagewise score. However, a higher-quality stagewise score may be weighted higher than the DE score, and for poorly defined stagewise scores, the DE score could likewise be favored.

The sections below briefly describe the type of noise each preprocessing step tries to address, and suggests a stagewise score for each stage. Some stagewise scores are very neatly defined, such as the Heteroscedacity Retention score for variance stabilization. Other stages, such as baseline correction, may have more difficulty accurately quantifying the success of the baseline removal with the signal-to-noise ratio score. In this case, the global DE score

will serve as an assistant to an imperfect local score. Section 3.1.3 includes a discussion of all methods which are analyzed at each stage of preprocessing.

3.2.2 Variance Stabilization and Heteroscedacity

Heteroscedacity is defined as a relationship between the mean and variance of features. This phenomenon can be observed by plotting the mean of feature intensities versus their standard deviation. Ideally, we would like to see a constant amount of variance across all intensities; thus the ideal plot would appear as a thin, horizontal cloud. A rough approach to quantifying heteroscedasticity can use linear regression to fit a line to points defined by these standard deviations and means. The slope of the fitted line indicates the degree to which the heteroscedasticity remains in the data after the transformation. A flatter slope is better. Furthermore, the sum of the residuals indicates to what degree the line approximates the cloud. A smaller sum would indicate that the variance-stabilizing transformation was able to alleviate more of the heteroscedasticity. Although both the slope and residual errors are important, it is not clear at what point the residual errors begin to matter more than the slope. Figure 9 displays an example on the pancreatic cancer dataset. Although the slope increases slightly, the residual error from a linear regression fit improves greatly. This is despite a slight increase in the slope from the raw data, which appears to be due mostly to the poor choice of linear regression to fit a quadratic relationship between mean and standard deviation. In practice, the difference between residual sums, combined with the small slope, is probably enough to consider this transformation beneficial to the data.

The Heteroscedacity Retention score (HR, equation 3.3) uses both the slope and sum of residuals in evaluating the goodness of a variance stabilization procedure. The sum of residuals is the dominant term in the equation. As the slope varies, additional penalties to the score are added, up to a maximum of each residual. When the slope is exactly 0, no penalty is incurred. A larger HR score indicates that the variance stabilization procedure retains more heteroscedacity (is poorer). To evaluate a variance stabilization procedure, the HR score is calculated after applying the procedure to the training set. The HR score is then used in the SP curve to decide which variance stabilization method performs the best.

$$
\begin{aligned}
HR &= \sum_{j=1}^{d} \left(R_j + R_j * (m_{\text{reg}})\right) \text{ , where} \\
R_j &= ||m_{\text{reg}} x_j + b_{\text{reg}} - y_j|| \text{ , where} \\
& \quad m_{\text{reg}} \text{ and } b_{\text{reg}} \text{ are from linear regression on } \mathbf{x} \text{ and } \mathbf{y}
\end{aligned}
\tag{3.3}
$$

3.2.3 Baseline removal, smoothing and the Signal-to-noise ratio

Baseline removal and smoothing are two techniques for removing systematic sources of noise. Baseline removal deals with constant background noise, while smoothing deals with a stochastic fluctuation between or within measurements. In the proteomic data, a baseline subtraction procedure restores the minimum of the measurements closer to 0. A smoothing technique relieves the data of high-frequency noise which may appear as separate peaks or valleys. Since these procedures remove information from the signal, the question is, how much noise is removed in comparison to true signal?

The signal-to-noise ratio (typically calculated as the ratio of amount of true signal to the amount of noise) seems to be a fitting metric for answering this question. Since the noise sources are different, interpreting how the signal is constructed can become a different task. A baseline shift is an additive noise which has its own variation. This should be considered separately from the noise which is targeted by smoothing techniques, and comes from other sources of variation influencing the signal (natural biological variation, chemical or mechanical noise). In the case of baseline removal, we can define the Baseline Signal-to-Noise Ratio score ($bSNR$) as $\frac{1}{d}\sum_{j=1}^{d}(\mu_{vj} - \mu_{bj})/(\sigma_{vj} - \sigma_{bj})$, where μ_{vj} is the mean intensity of feature j before baseline correction, μ_{bj} is the mean intensity of the baseline at feature j, and σ_{vj} and σ_{bj} refer to the standard deviations of the feature and baseline intensities, respectively. In the case of smoothing, since we only need to worry about the variance experienced after the baseline is considered, we can drop the $_{bj}$ terms and calculate the Smoothing Signal-to-Noise ratio score $sSNR$ simply as $\frac{1}{d}\sum_{j=1}^{d}\mu_j/\sigma_j$. Both scores estimate the average signal-to-noise ratio resulting from a preprocessing method. These scores are used to determine the SP curves for baseline correction and smoothing, respectively.

The standard application of baseline correction and smoothing methods are analogous to that of variance stabilization. For either stage, a set of methods is obtained. Each one is applied to the training data at an appropriate time in the preprocessing sequence. The *bSNR* score is obtained by using the chosen baseline removal method to calculate the baselines of the profiles. Likewise the *sSNR* score is computed using the resulting smoothed profiles from a smoothing method. Methods which have an acceptable *PR* score are retained for the computation of their stage's SP curve.

3.2.4 Alignment and the coefficient of variation

Alignment methods for proteomic profiling often differ in the metrics used to evaluate the quality of the alignment. In general, an alignment is considered successful if the variation in features which should be identical is minimized. Therefore, methods have attempted to characterize this variation in many ways, for example, by measuring the percentage of variance captured by the first two principal components of a PCA decomposition, or by counting the percentage of profiles in which a peak is detected. A hybrid between these approaches is using the coefficient of variation, computed as $\sigma_j \ \mu_j$. This is a measure of the dispersion of the variable, and should be smaller for measurements which should be identical. Since many of the features in a proteomic profile will most likely not be discriminative, the variation in most of the features should be minimized. Measuring the average coefficient of variation ($ACoV = \frac{1}{d}\sum_{j=1}^{d} \sigma_j \ \mu_j$) is a simple and effective metric for comparing the vastly different alignment methods discussed above. After an alignment method is applied to the training data, the $ACoV$ score is computed. Since the $ACoV$ score uses the mean and standard deviation across the entire training data set, a single score for the method is achieved. Methods with an acceptable *PR* score are used to compute the SP curve. The mean aligned profile is used as a template, when needed, to align profiles from testing data.

Occasionally, a poorly-designed method can enter the competition between other valid methods. This method may allow the *DE* score to artificially inflate. For example, the Moving-median smoothing routine available during the smoothing stage often performed very well on the training data, but results on the test data were poor. Since smoothing occurs

after baseline correction, many values are brought to 0 or close to 0. This means that the median at many points in the profile will be 0 or very small. The moving median smoother effectively erases much of the signal and smooths out several peaks if the sliding window parameter is chosen poorly. The only part of the profile that remains similar is the start of the profile (from 0 to roughly 3 kDa). This region in a MS proteomic profile is typically referred to as the "junk" region, since many biological artifacts arise from the vaporization of the analytical surface. These can include matrix molecules that are used to hold the biofluid on the analytical surface, portions of broken proteins or other contamination. Nevertheless, this region contains thousands of features and can possess spuriously discriminative features. If the rest of the profile has been smoothed to 0, the classifier evaluating the DE score will be forced to choose one of these spuriously discriminative features. However, the classifier cannot make a guarantee about the reliability of the feature appearing in the junk region of future test cases. In this case, the SP-curve will choose a poor method, and the SAP preprocessing will suffer. This means the choice of the DE score must also be made carefully, in order to try to weight robust preprocessing methods more heavily. In an effort to require methods behave reasonably in their treatment of the data, I enforce the condition that only methods which retain a number of "peak" features in the profile are considered for application. This constrains the overall signal shape as a result of the preprocessing. A peak detection routine is used to calculate positions of local maxima in the profile. These peak features are often subselected in later stages of analysis as a first pass of feature selection, since they often vary the most, and therefore have a greater chance to be discriminative. If a preprocessing routine does not retain at least $P\%$ of the peaks in the data, the method is not allowed to be chosen as a preprocessing step. The Peak Retention score (PR) is an indicator function which is 1 when the percentage of peaks retained is $> P$, and 0 otherwise. In our experiments, $P = 50\%$.

All of the above methods were implemented in MATLAB. The computational time depends largely on the number of profiles in the dataset, since many preprocessing routines operate in a vectorized manner over the feature space, but work on individual profiles at a time. Since the automatic preprocessing procedure evaluates all preprocessing methods, the worst-case complexity is dependant on the most resource-intensive method.

3.3 EXPERIMENTS AND RESULTS

These experiments are intended to evaluate and compare the above methodologies for preprocessing and demonstrate the effectiveness of automatic selection of preprocessing methods. A flowchart for these experiments is depicted in Figure 11. Briefly, a comparison is made between three different preprocessing methodologies: no preprocessing (referred to as "Raw"), the "baseline" preprocessing procedure discussed in section 3.3.1 and the "Standard Automatic Preprocessing" method, where the SP curves are used to select methods, but no class information is used during the individual preprocessing stages. The raw dataset is divided into training and testing datasets. The Standard Automatic Preprocessing (SAP) procedure uses only the training data until the 'best' preprocessing methods are known, and then these are applied to the testing set.

A predetermined predictive model is trained using the preprocessed training data, and classifies the profiles in the preprocessed testing set. The predictive model used in all experiments for this section is a Support Vector Machine (SVM) with a linear kernel and ℓ_1-norm penalized regularization. This model was chosen primarily because of its simplicity. Additional details about this model are given in 4.2.10. This model is learned from the training set and applied to the testing set, and the resulting performance (in terms of Area under the ROC curve, or AUC) is recorded and reported in the experiments below. The AUC performance is averaged over 40 training/testing data splits. These splits are identical for each of the three evaluated preprocessing procedures. I begin by describing the results of applying the baseline and SAP procedures relative to performance on raw data.

3.3.1 Baseline Preprocessing

During the development of my research, a baseline preprocessing procedure was developed [25, 38, 39]. The baseline preprocessing procedure was originally designed to remove what was perceived at the time as the "major" noise artifacts; the baseline shift and the high-frequency noise running along the entire signal. This procedure was empirically developed primarily through visualization of the data before and after preprocessing. Sat-

isfactory elimination of the noise artifacts on multiple datasets, coupled with satisfactory downstream classification performance, suggested that this procedure was robust. The baseline preprocessing procedure (not to be confused with baseline correction, the second stage of preprocessing) is performed as follows:

- *Variance stabilization* - Cube-root transformation
- *Baseline correction* - Our baseline correction procedure uses a sliding window of 200 time-points to define local minima. These points are then linearly interpolated to define the baseline. The area underneath the baseline is then subtracted from the uncorrected signal.
- *Normalization* - Total Ion Current (TIC) normalization from 1500 to 20000 Daltons. This constant is calculated individually for each profile.
- *Smoothing* - We use Gaussian-kernel smoothing to remove random noise in the signal. The kernel affects 12 time-points at once.
- *Alignment* - Before alignment takes place, a mean reference profile is computed by averaging all profiles within a dataset. Profiles are aligned individually to this reference profile via dynamic programming within an area of 200 time-points.

In order to demonstrate the ability of the baseline preprocessing procedure alone, I also evaluated classifier performance with respect to the unprocessed, raw data. Thus, the predictive model was trained on raw data, and evaluated on the raw testing data profiles. Table 2 displays the average area under ROC curves (AUC) for the predictive models evaluated on the raw and baseline-preprocessed data.

Two important notions stand out from these results. The first is that many of the datasets possess a signal in the raw data which enables correct classification beyond randomly guessing (random guessing would give AUC = 0.5). In the case of the prostate cancer dataset, we know this is due to a spurious discriminative signal introduced during data production [6]. The same effect may be present in other datasets, but this is difficult to determine. On the other hand, it is encouraging to see signal from the raw data, as it would be overly pessimistic to assume the mass spectrometer could not reveal true biological differences in the samples. Regardless of the authenticity of the signal, it is unclear whether the

baseline preprocessing procedure is robust enough to consistently improve the performance of downstream predictive models. Four out of the 14 datasets experience an average drawback from learning with baseline-preprocessed data. Of those datasets that do experience an advantage, this advantage is greater than 0.03 AUC in only 2 cases. The prostate cancer dataset deserves an exception, as the baseline procedure effectively attacks the spuriously discriminative signal. As a result, the performance drops greatly (-0.11 AUC).

In some cases, the baseline preprocessing procedure results in a poorer average AUC than possible with the raw data. This suggests that most raw data contains a "usable" discriminative signal before any preprocessing takes place. The source of this discriminative signal can be genuine biological information, or it can signify bias introduced in the data production from lack of randomization between case and control sample processing, as was the case in the prostate cancer dataset [6]. Regardless of the authenticity of the signal, it is clear that our baseline preprocessing procedure is not suited to preprocessing the majority of data. In almost all cases, the average AUC of resulting predictive models is lower than those provided with the raw data.

The advantages imparted by the baseline procedure are small in all cases except the Hepatitis and Vanderbilt Lung Maldi dataset. The addition of an additional 1% of AUC, as well as largely overlapping confidence bounds suggests that the baseline procedure may not have much influence on the development of predictive models. Regardless of the baseline procedure's performance, we can see that it will not guarantee a positive advantage over using the raw data, and we may still seek improvement through other preprocessing procedures.

3.3.2 Scored Standard Automatic Preprocessing

We assume raw protein profiles have inherent discriminability, and the noise upon these sources differ per dataset. The Standard Automatic Preprocessing (SAP) method attempts to adjust to noise sources by choosing preprocessing methods at each stage which best improve the discriminability of the data. Table 3 displays the average AUC achieved by predictive models on the baseline-preprocessed and SAP-preprocessed data. The rightmost column indicates the "advantage" of using the SAP procedure over the baseline procedure.

This advantage is surprisingly negative for all except four datasets (COPD, ILD, Prostate Cancer and Vanderbilt Lung MALDI, where the advantage is 0). Interestingly, three of these datasets incur a disadvantage from using the baseline procedure as seen in Table 2, and the advantage given to COPD by SAP is greater than the advantage given to COPD by the baseline procedure. This suggests that the SAP procedure can indeed improve over fixed preprocessing procedures which are not robust enough for every dataset.

Unfortunately, for the rest of the datasets, the advantage from using SAP is negative, but small (less than 0.05% AUC). For those datasets where SAP imparts a large disadvantage to the predictive model, we might be concerned whether data production bias was present in the raw data despite careful data production protocols. A small disadvantage may simply be due to the limited pool of methods available to the SAP procedure. A larger pool of available methods with alternatives for parameter settings may enable SAP to perform better than the baseline procedure. These methods could be created on the fly by performing a grid-search over the parameter space, and allowing the results of that search to be used in the SP-curve selection process explained in Section 3.3.1.

3.3.3 Discussion

The SAP procedure was meant to be an intelligent method for dealing with the uncertainty of whether a static preprocessing procedure was proper for any mass spectrometry protein profiling dataset. Thus it was surprising to see that the baseline routine, created primarily through experimental evaluation on the pancreatic cancer datasets, often outperformed SAP. However, performance of the two methods is very close. There are many ways to explain this behavior. The first is that our local stagewise scores for each stage may not be very strong. The heteroscedacity score for variance stabilization is a well-defined metric which targets and quantifies a noise in our signal. However, signal-to-noise ratio based metrics such as those for baseline correction and smoothing are more general and need more of an assist from the global DE score. The second reason why SAP may have underperformed was due to the small pools of methods available. With few candidates for every stage, and each stage including the method used in the baseline procedure, it is easy for SAP to select

part or all of the baseline procedure's methods. On a good note, this reinforces the strength of our choices for the baseline preprocessing procedure we developed prior to SAP. Finally, the raw data often possesses a usable discriminative signal by itself. This suggests that data production bias may already be evident before any preprocessing begins. It is possible that the baseline routine is actually failing to remove some or all of this production bias. SAP is more aggressive in targeting the noise sources, and may remove more of these production biases, but at the cost of not outperforming the baseline method in terms of predictive model performance.

The following section discusses the performance of SAP with respect to the baseline procedure in more detail.

3.3.3.1 Value of Individual Preprocessing Stages

I re-evaluated the baseline and SAP procedures by individual stage. This means that for each stage, only that stage of preprocessing is performed, and the rest are skipped. The resulting data is then given to the predictive model for training and evaluation. SAP in this case will only make a decision for the method that will perform that stage.

Table 4 displays the AUC advantage contributed by each stage of baseline preprocessing alone. No individual stage by itself stands out as the major contributor for every datsaet. Instead, each dataset seems to be most favored by a different stage of preprocessing. In general, variance stabilization (by cube-root transformation) seems to be the most harmless.

Table 5 displays the AUC advantage contributed by applying the method learned by SAP for each stage. Similarities between this table and Table 4 would indicate that SAP tried to choose the same method as baseline. This occurs most often with intensity correction, as SAP frequently chose to use the baseline procedure's TIC-normalization routine. In other stages, there was more variance in the methods selected by SAP. For those datasets where SAP outperformed baseline (COPD, ILD, Prostate Cancer, Vanderbilt MALDI), there are patterns for the selection of methods which may be indicative of the behavior of SAP.

Sometimes, a competing method provides a good fit and it is chosen consistently. For example, SAP repeatedly chose the generalized logarithm variance stabilization procedure for the ILD dataset. The result was an improvement in the variance stabilization stage. For the

COPD and Pancreatic Cancer II datasets, SAP alternated between performing no baseline correction 50% of the time, and performing the monotone-baseline correction procedure the other 50%. In the case of COPD, there was an average improvement, but the Pancreatic Cancer II dataset suffered from the mixture. It's possible that we got lucky with the mixture with the COPD data, but the large drop in advantage for the Pancreatic Cancer dataset and frequent choice to perform no baseline correction indicates that there was a poor selection of baseline correction routines available for that dataset.

Interestingly, for only the two MALDI datasets (Diabetes and Vanderbilt Lung MALDI) SAP chose to use the Log-transformation variance stabilization routine, and the average advantage of applying this transformation improves over the baseline's performance for this stage. This may suggest that the Log-transformation is better suited to data from the MALDI technology. Provided a wide array of preprocessing methods and a variety of data types, SAP may be able to establish fixed preprocessing routines per data production technologies.

The large drops in advantage for the smoothing stage should not be alarming. Those datasets which seem to suffer end up having the Moving-median smoothing routine chosen for them. This routine is simply not a good method. Since the median is used as the replacement value within the sliding window, a large number of features are reset to what the local baseline appears to be. The remaining features which stand out are likely to be from the aforementioned "junk" region, which exhibits high variance. It becomes possible to discriminate profiles in the training set by chance (and also possibly because baseline correction wasn't performed beforehand, the baseline shift between classes may be obvious). However, these features are not guaranteed to be robust for future data, and the error rate on the test set increases. For the other datasets, SAP chooses among the other smoothing methods, and this has a more positive effect. The lack of baseline preprocessing makes a difference. For the Vanderbilt Lung WCX dataset, SAP chooses the moving median routine if no baseline correction occurs, and as shown in Table 5, there is a large performance dropoff. However, when smoothed within the context of the entire preprocessing pipeline, SAP chooses a mixture of the Savitzky-Golay smoothing method, the Fourier-transform smoothing method and No smoothing. The results as shown in Table 3 indicate that, in

fact, SAP can choose a capable method. Nevertheless, this underscores the importance of the teamwork and order between preprocessing stages.

3.3.3.2 Effect of additional training data on preprocessing procedures

I investigated how preprocessing procedures behaved as a result of learning from more or less training data. Of course, the baseline procedure doesn't "learn" anything, as its preprocessing methods are fixed. However, in the previous subsection, it was apparent that changes in the training data can cause a different selection of methods for SAP. This is evidenced by the selection of a mixture of methods to perform stages of preprocessing over the 40 separate train/test splits.

For the three "Vanderbilt Lung SPORE" datasets, I repeated the experiments from this chapter using random subsamples of the training data, starting at 20%, and increasing by steps to 35%, 50%, 65% and 80% of the training set. Figures 12, 13 and 14 display a plot of each preprocessing method's average performance (in terms of AUC and their 95% confidence intervals) as a function of the available training data percentage. In the interest of time, these experiments were measured over only the first 10 out of 40 train/test splits. The test set size remains constant (30% of original data), while the training set (70% of original data) underwent subselection for each of the thresholds. The remaining data removed from the training set is not used in any way. It can be seen that SAP performs close to the baseline routine in terms of performance.

The procedures' performance trends upward as the training set size increases. This is the expected behavior, since increasing the training set size increases the heterogeneity of the training data. This in turn improves the robustness of the predictive model, resulting in better predictive performance.

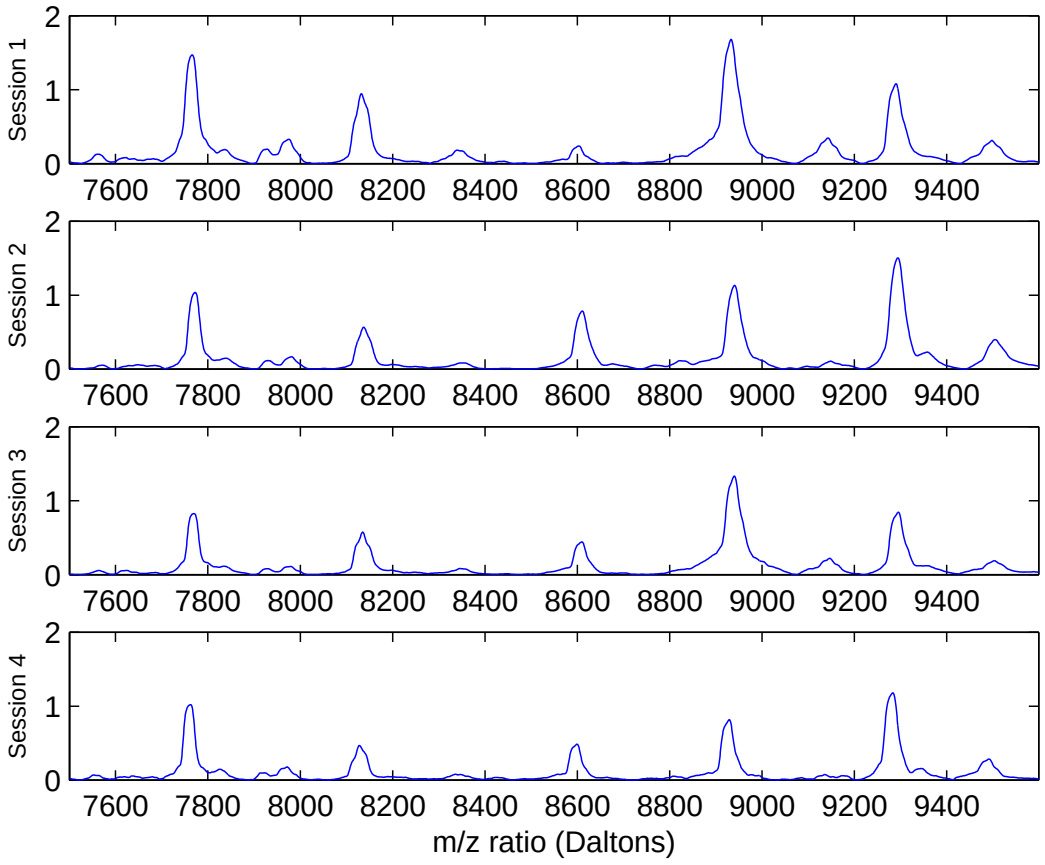

Figure 4: MS profiles for a single sample across 4 different sessions. Changes are apparent in relative intensities of peaks.

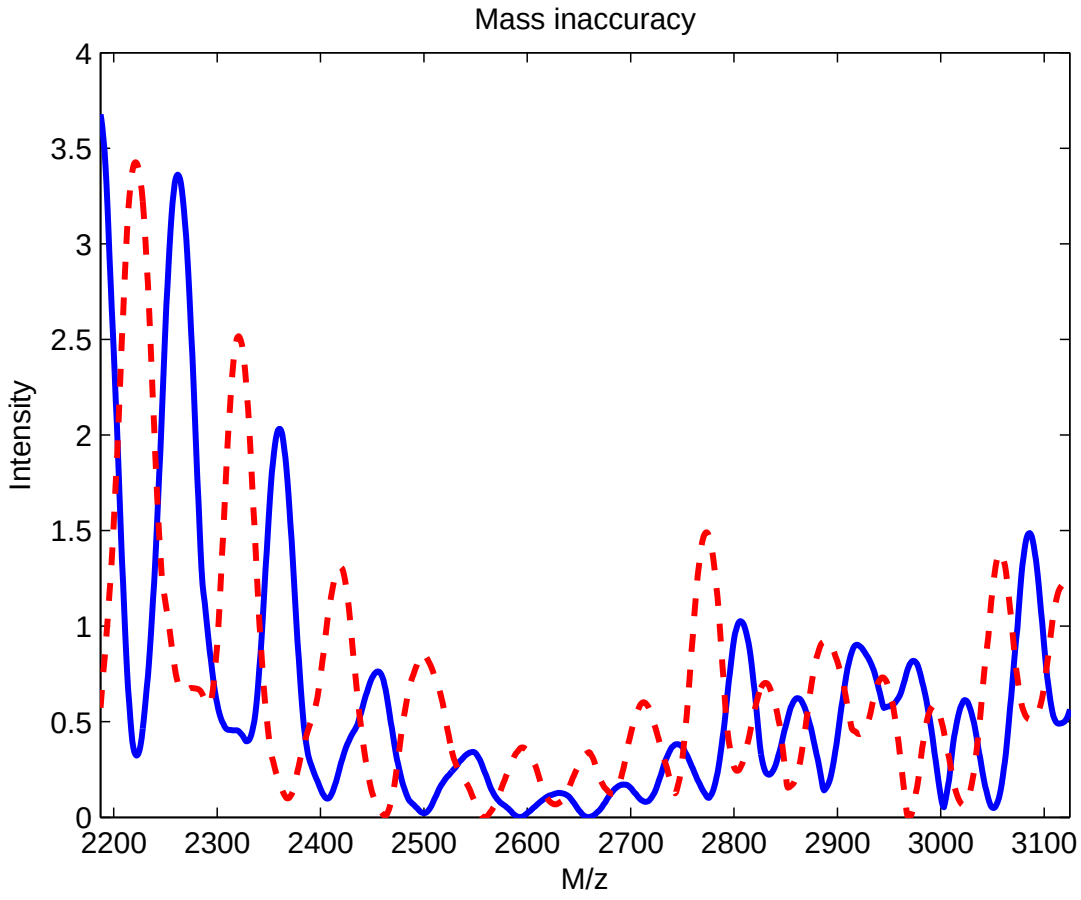

Figure 5: An example of misalignment or mass inaccuracy. Two profiles from QA/QC serum are shown. The signals are shifted so severely, the same feature (peak shape) in the two signals correspond to different m/z positions.

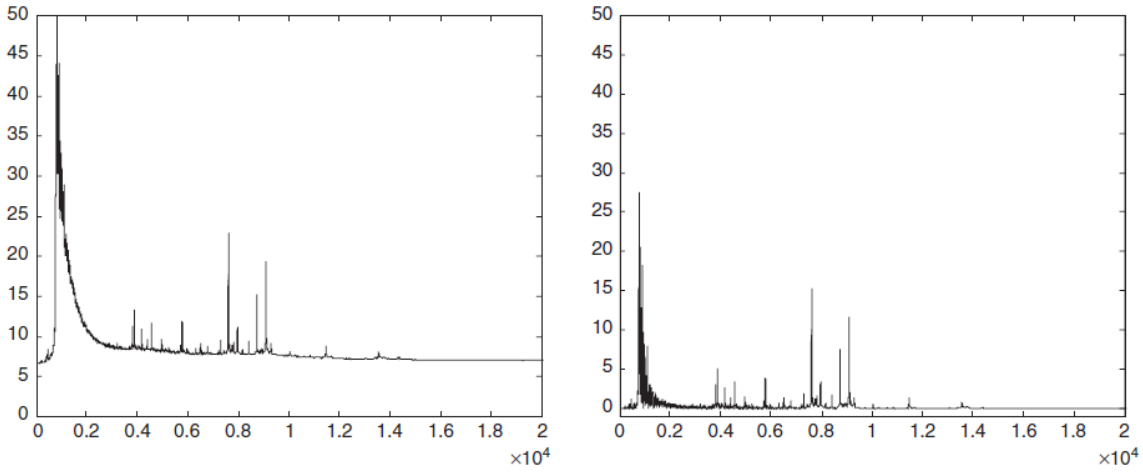

Figure 6: An example of baseline correction. Left panel: a profile with a baseline drift. Right panel: the corrected profile. The additive component in the signal is removed and the baseline is shifted to the zero intensity level.

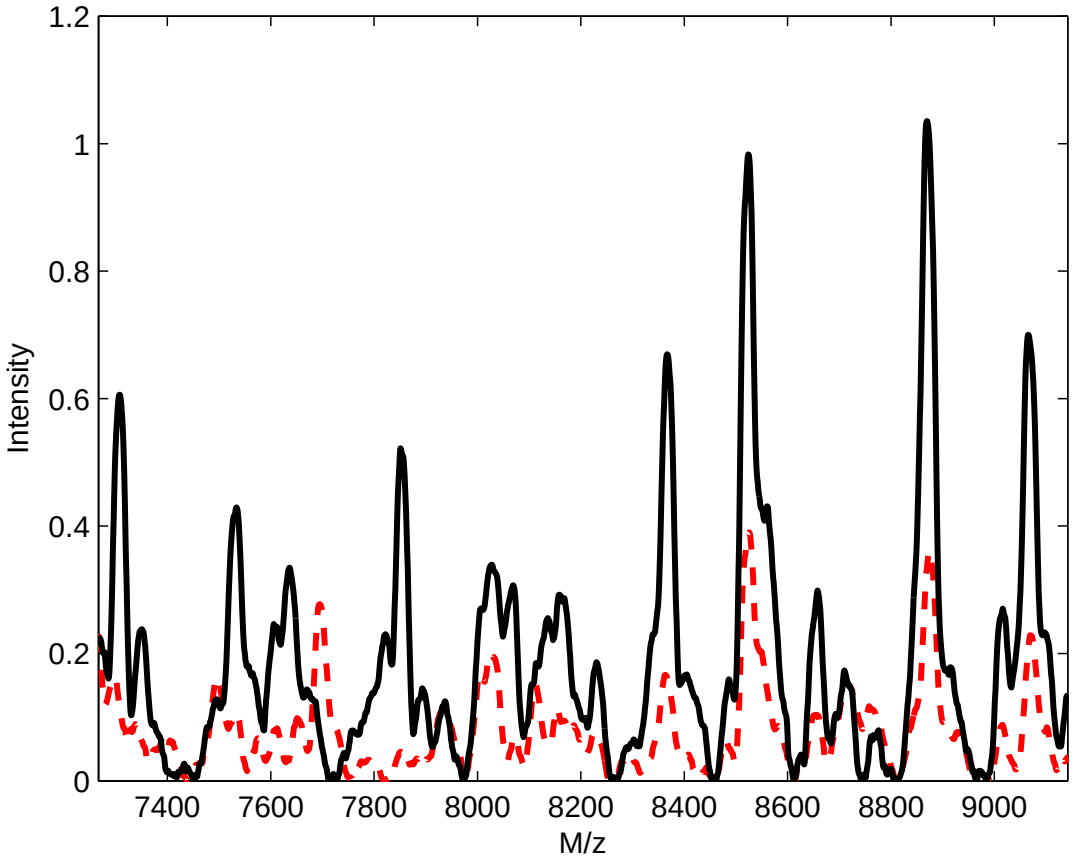

Figure 7: An example of intensity variation.

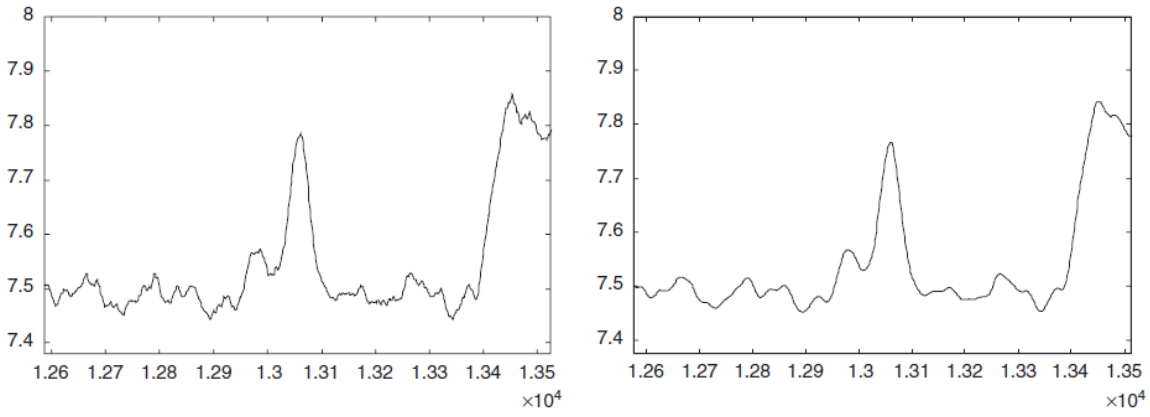

Figure 8: An example of smoothing

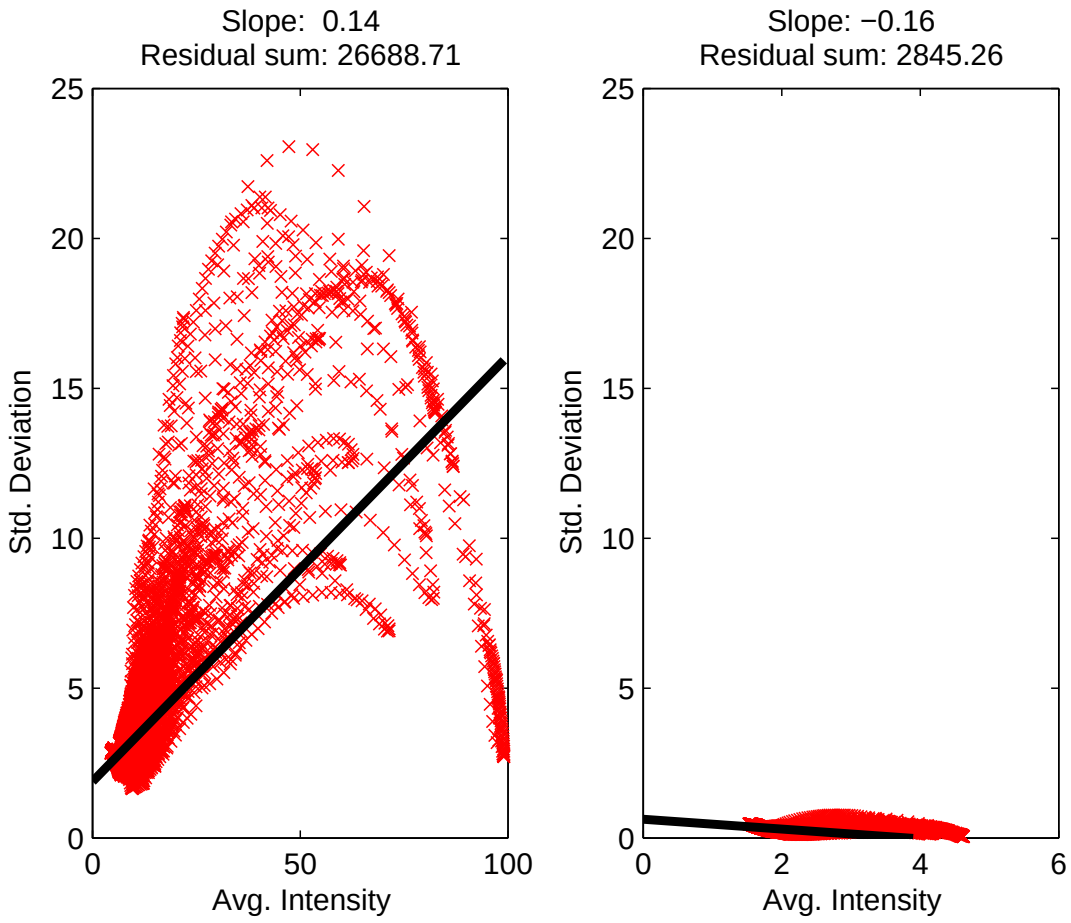

Figure 9: The effect of variance stabilization on heteroscedacity.

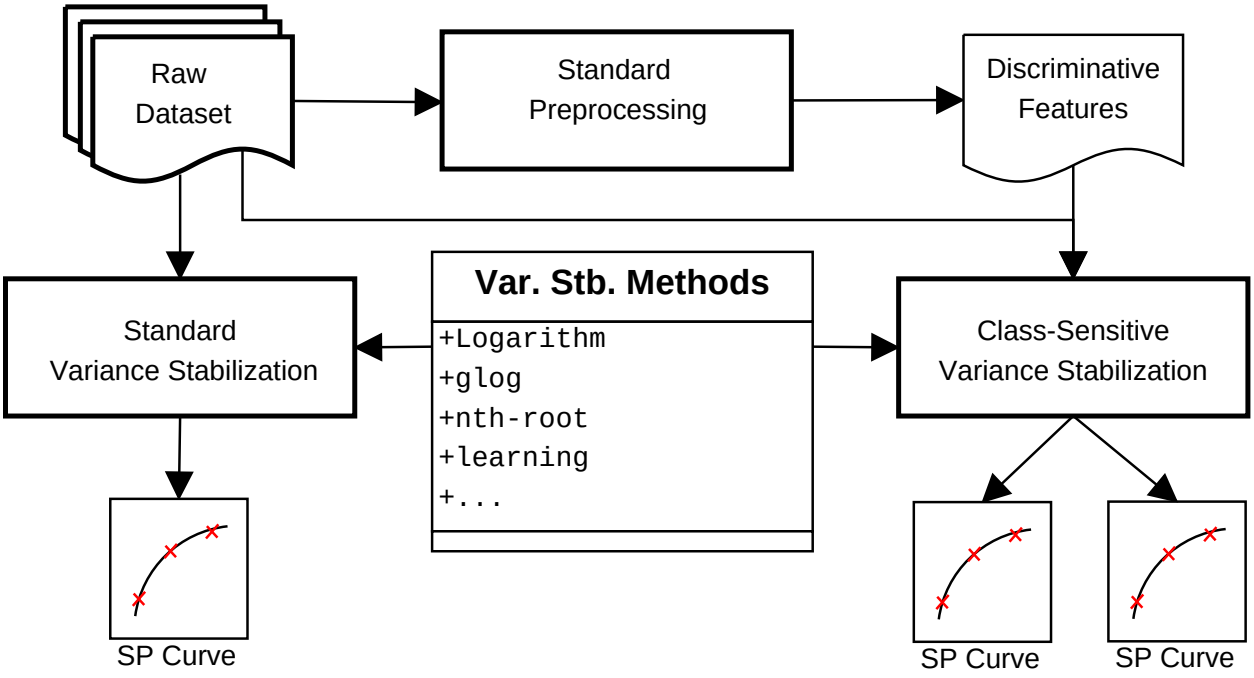

Figure 10: Flowchart for evaluation of variance stabilization techniques.

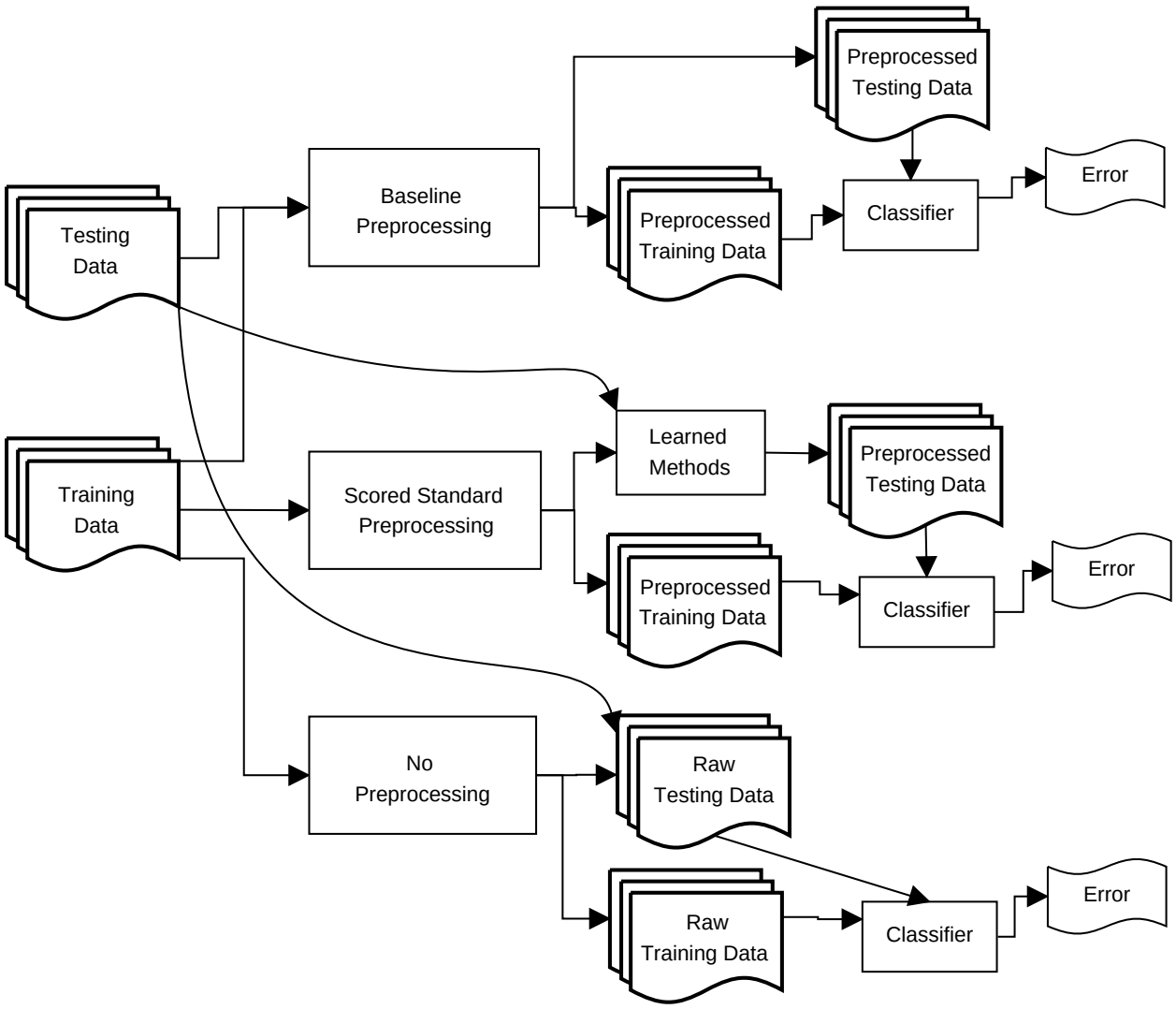

Figure 11: A flowchart for the evaluation of preprocessing procedures.

Table 2: Classifier performance on raw versus baseline-preprocessed data.

Dataset	Raw AUC	Baseline AUC	Advantage
COPD	0.5088 ± 0.2510	0.5179 ± 0.2319	0.0091
Hepatitis C	0.5933 ± 0.2098	0.7120 ± 0.1713	0.1187
ILD	0.5593 ± 0.2610	0.5108 ± 0.2591	-0.0485
Diabetes	0.6078 ± 0.2340	0.6311 ± 0.2190	0.0233
Melanoma I	0.5997 ± 0.1767	0.5875 ± 0.1916	-0.0122
Breast Cancer	0.5113 ± 0.1487	0.5140 ± 0.1366	0.0027
Pancreatic Cancer I	0.8963 ± 0.0632	0.9050 ± 0.0538	0.0088
Pancreatic Cancer II	0.8450 ± 0.0833	0.8501 ± 0.0726	0.0050
Prostate Cancer	0.9718 ± 0.0163	0.8679 ± 0.0571	-0.1039
Scleroderma	0.7164 ± 0.1135	0.7216 ± 0.1199	0.0052
UPCI Lung Cancer	0.7589 ± 0.0737	0.7803 ± 0.0713	0.0214
Vanderbilt Lung IMAC	0.8976 ± 0.0419	0.9007 ± 0.0454	0.0031
Vanderbilt Lung WCX	0.8584 ± 0.0520	0.8858 ± 0.0470	0.0274
Vanderbilt MALDI	0.8773 ± 0.0504	0.8291 ± 0.0617	-0.0481

Table 3: Classifier performance on baseline versus automatically preprocessed data

Dataset	Baseline AUC	SAP AUC	Advantage
COPD	0.5179 ± 0.2319	0.5460 ± 0.2366	0.0281
Hepatitis C	0.7120 ± 0.1713	0.6657 ± 0.1911	-0.0463
ILD	0.5108 ± 0.2591	0.5759 ± 0.2441	0.0650
Diabetes	0.6311 ± 0.2190	0.6025 ± 0.2698	-0.0287
Melanoma I	0.5875 ± 0.1916	0.5384 ± 0.1961	-0.0491
Breast Cancer	0.5140 ± 0.1366	0.4894 ± 0.1481	-0.0246
Pancreatic Cancer I	0.9050 ± 0.0538	0.8978 ± 0.0663	-0.0072
Pancreatic Cancer II	0.8501 ± 0.0726	0.8047 ± 0.0836	-0.0454
Prostate Cancer	0.8679 ± 0.0571	0.8695 ± 0.0565	0.0015
Scleroderma	0.7216 ± 0.1199	0.6630 ± 0.1144	-0.0586
UPCI Lung Cancer	0.7803 ± 0.0713	0.7307 ± 0.0792	-0.0496
Vanderbilt Lung IMAC	0.9007 ± 0.0454	0.8535 ± 0.0573	-0.0472
Vanderbilt Lung WCX	0.8858 ± 0.0470	0.8735 ± 0.0475	-0.0123
Vanderbilt MALDI	0.8291 ± 0.0617	0.8291 ± 0.0679	0.0000

Table 4: Stagewise Contributions of the Baseline Preprocessing Procedure

Dataset	Var. Stab.	Base. Corr.	Int. Corr	Smoothing
COPD	-0.0007 ± 0.0195	0.0352 ± 0.0315	0.0503 ± 0.0368	0.0876 ± 0.0237
Hepatitis C	-0.0020 ± -0.0172	0.0331 ± 0.0055	0.1565 ± 0.1112	0.0510 ± 0.0248
ILD	-0.0070 ± 0.0039	-0.0100 ± -0.0073	-0.0138 ± 0.0155	0.0320 ± 0.0386
Diabetes	-0.0038 ± -0.0009	-0.0680 ± -0.0523	-0.0318 ± -0.0203	-0.0367 ± -0.0362
Melanoma I	-0.0086 ± -0.0190	-0.0464 ± -0.0533	-0.0941 ± -0.0919	-0.0423 ± -0.0234
Breast Cancer	-0.0186 ± -0.0243	0.0188 ± 0.0007	0.0481 ± 0.0244	-0.0091 ± -0.0120
Pancreatic Cancer I	0.0052 ± 0.0032	-0.0150 ± -0.0044	-0.0284 ± -0.0182	-0.0214 ± -0.0111
Pancreatic Cancer II	0.0191 ± 0.0116	-0.0812 ± -0.0566	-0.0676 ± -0.0525	-0.0144 ± -0.0153
Prostate Cancer	-0.1505 ± -0.0788	-0.0190 ± -0.0125	0.0038 ± 0.0022	-0.0058 ± -0.0032
Scleroderma	0.0447 ± 0.0399	0.0349 ± 0.0221	0.0086 ± 0.0027	0.0285 ± 0.0354
UPCI Lung Cancer	0.0044 ± 0.0042	0.0270 ± 0.0313	0.0202 ± 0.0138	-0.0172 ± -0.0137
Vanderbilt Lung IMAC	0.0064 ± 0.0100	-0.0464 ± -0.0351	0.0093 ± 0.0127	-0.0054 ± 0.0013
Vanderbilt Lung WCX	0.0030 ± 0.0047	-0.0338 ± -0.0153	0.0287 ± 0.0255	0.0144 ± 0.0108
Vanderbilt MALDI	0.0095 ± 0.0058	-0.0399 ± -0.0291	-0.0164 ± -0.0097	-0.0019 ± -0.0028

Table 5: Stagewise Contributions of the SAP Preprocessing Procedure

Dataset	Var. Stab.	Base. Corr.	Int. Corr	Smoothing
COPD	0.0035 ± -0.0263	0.0499 ± 0.0231	0.0402 ± 0.0455	0.1005 ± 0.0893
Hepatitis C	-0.0112 ± -0.0009	0.0229 ± -0.0067	0.1565 ± 0.0816	-0.0569 ± -0.0633
ILD	0.0170 ± -0.0004	0.0101 ± 0.0055	-0.0391 ± -0.0034	-0.0467 ± -0.0191
Diabetes	-0.0261 ± 0.0340	-0.0060 ± 0.0054	-0.0072 ± 0.0103	-0.0716 ± -0.0309
Melanoma I	-0.0086 ± -0.0169	-0.0530 ± -0.0357	-0.0650 ± -0.0276	-0.0798 ± -0.0638
Breast Cancer	-0.0281 ± -0.0296	-0.0110 ± -0.0331	0.0387 ± 0.0303	0.0269 ± 0.0125
Pancreatic Cancer I	0.0052 ± 0.0083	-0.0651 ± -0.0524	-0.0284 ± -0.0108	-0.3421 ± -0.2834
Pancreatic Cancer II	0.0191 ± 0.0055	-0.1659 ± -0.1369	-0.0676 ± -0.0547	-0.2896 ± -0.2326
Prostate Cancer	-0.0166 ± -0.0066	-0.0195 ± -0.0097	0.0038 ± 0.0008	-0.2385 ± -0.1643
Scleroderma	0.0721 ± 0.0593	0.0237 ± 0.0301	0.0086 ± 0.0102	-0.1484 ± -0.1269
UPCI Lung Cancer	0.0044 ± 0.0055	0.0344 ± 0.0374	0.0202 ± 0.0289	-0.1146 ± -0.0974
Vanderbilt Lung IMAC	0.0064 ± 0.0032	-0.0803 ± -0.0599	0.0093 ± 0.0071	0.0003 ± -0.0014
Vanderbilt Lung WCX	0.0030 ± 0.0060	-0.0062 ± -0.0006	0.0287 ± 0.0268	-0.1808 ± -0.1411
Vanderbilt MALDI	0.0124 ± 0.0082	-0.0350 ± -0.0240	-0.0164 ± -0.0144	0.0000 ± -0.0017

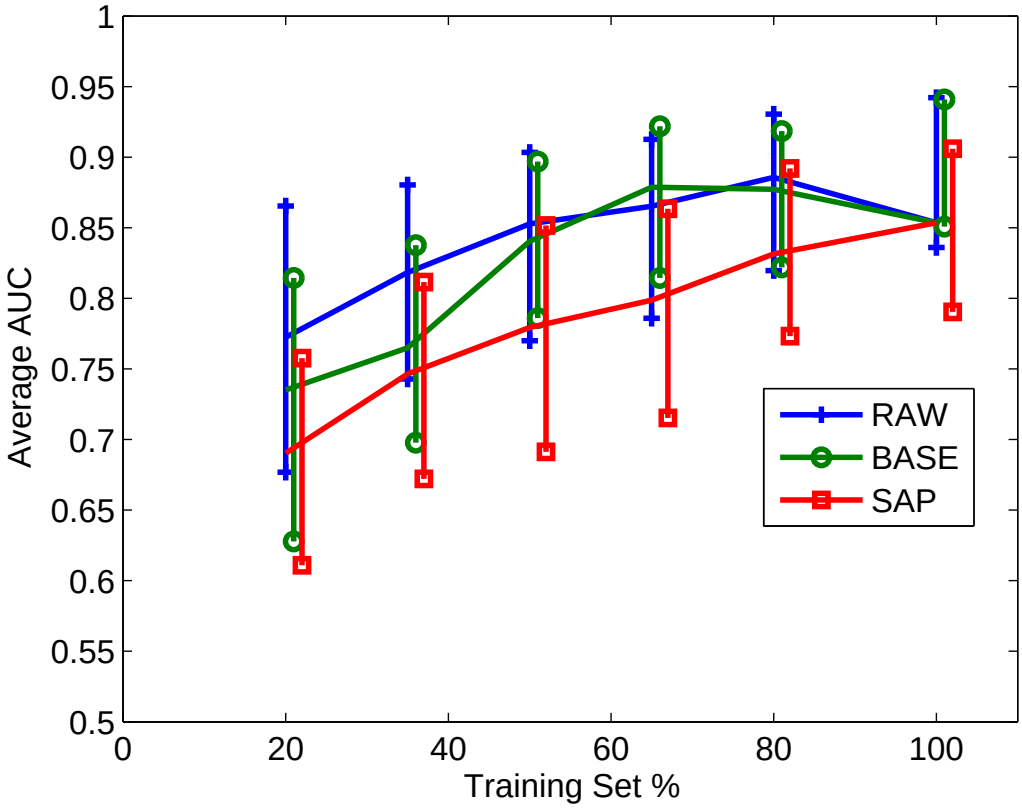

Figure 12: Performance versus varied train set size, Vanderbilt Lung SPORE IMAC data

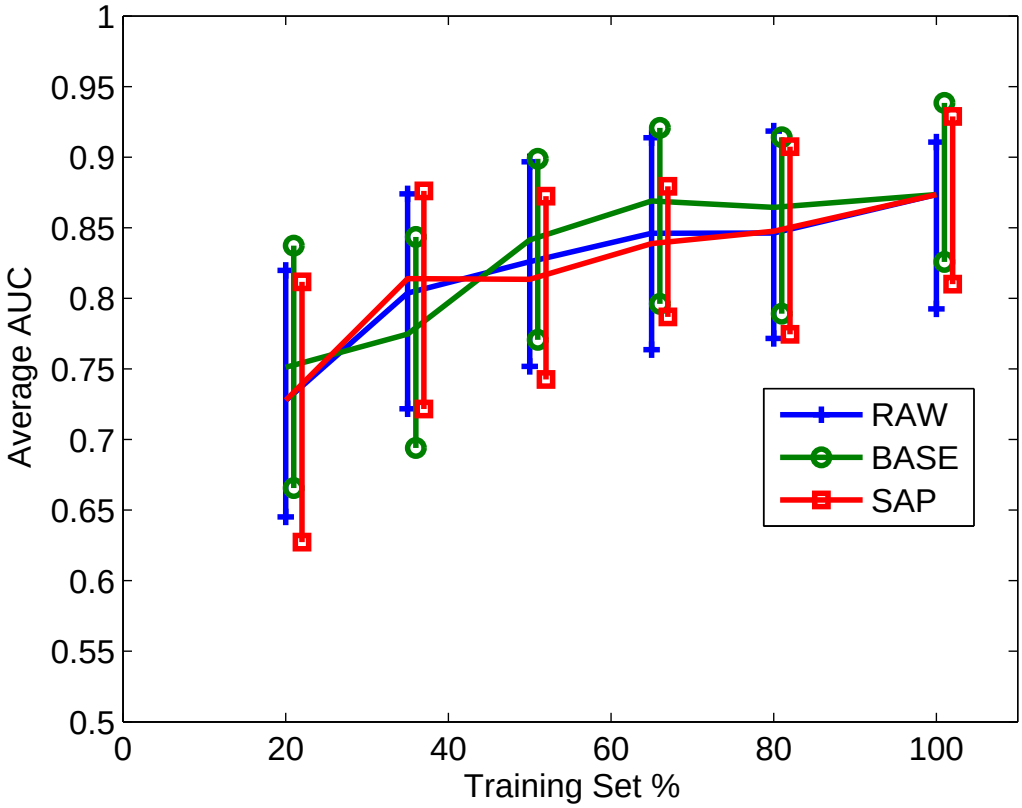

Figure 13: Performance versus varied train set size, Vanderbilt Lung SPORE WCX data

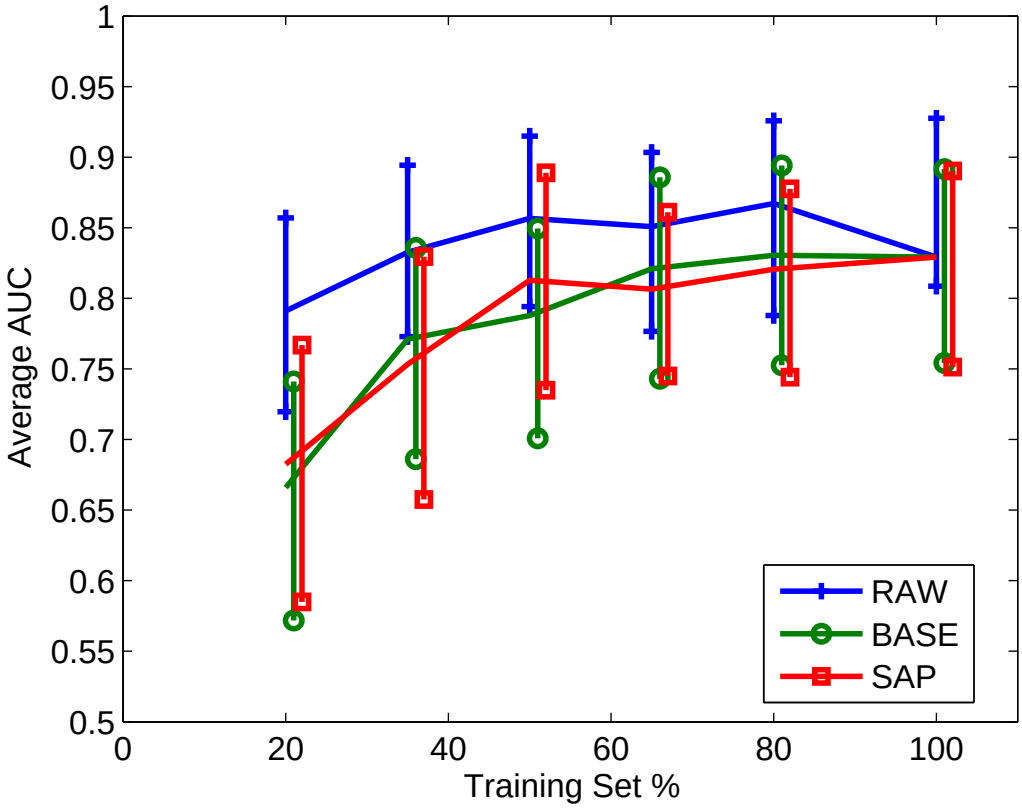

Figure 14: Performance versus varied train set size, Vanderbilt Lung SPORE MALDI data

4.0 BIOMARKER DISCOVERY AND PREDICTIVE MODELING

4.1 BACKGROUND

High-throughput data technologies such as microarray and MS profiling are producing large quantities of genomic and proteomic data relevant for our understanding of the behavior and function of an organism. This often includes the study of characteristics of disease and its dynamics. Thousands of genes are measured in a typical microarray assay; tens of thousands of measurements comprise a mass spectrometry proteomic profile. The high–dimensional nature of the data demands the development of special data analysis procedures that are able to adequately handle such data. Once these data are properly preprocessed, the central question of this process becomes the identification of those *features* (measurements, attributes) that are most relevant for characterizing the system and its behavior. We study this problem in the context of classification tasks where our goal is to build a model that lets us discriminate well among classes of samples, such as samples from people with and without a certain disease. Discovering the features and building a model that uses them are two intertwined processes. Neither task is straightforward, and both tasks have caveats which must be addressed.

4.1.1 Feature selection

Feature selection is a process that aims to identify a smaller set of features from a large number of features. Reducing the number of features is often done with the goal of simplifying the process of discriminating between classes (groups) in the data. If the number of feature candidates is small and the number of samples in the data set is large, feature selection is

only rarely an issue. However, high-throughput data suffers from the *curse of dimensionality*. Data such as MS profiles are naturally *high-dimensional*, with the number of features being in the hundred-thousands, and the number of samples in a dataset often being less than a hundred. By learning a model from data with so many features and so few samples, the estimates of parameters of the model are unreliable and may cause *overfitting*, a phenomenon in which each datum is fit so rigidly that the model lacks flexibility for future data. To avoid overfitting, feature selection is applied to balance the number of features in proportion to the number of samples.

Feature selection can be a one-shot process, but it can also include search problems where multiple sets of features are evaluated and compared. However, high-throughput data is naturally *high-dimensional*, with the number of features being in the hundred-thousands. This makes the number of possible feature subsets prohibitively large to explore exhaustively. Thus, efficient feature selection methods are typically sought. These features must also be strongly correlated with the class membership (in this case, the disease state). At the same time, the feature selection method must be correct in retrieving valid features.

Feature selection methods are typically divided into three main groups: *filter, wrapper and embedded methods*. Filter methods rank each feature according to some univariate metric, and only the highest ranking features are used; the remaining features are eliminated. Wrapper algorithms [40] search for the best subset of features. These methods use a predictive model during the selection process to evaluate feature combinations. The wrapper algorithm treats a classification algorithm as a black box, so any classification method can be combined with the wrapper. Standard optimization techniques (hill climbing, simulated annealing or genetic algorithms) can be used.

Embedded methods search among different feature subsets, but unlike wrappers, the process is tied closely to a certain classification model and takes advantage of the model's characteristics and structure. In addition to feature selection approaches, in which a subset of original features is searched, the dimensionality problem can be often resolved via *feature construction*. The process of feature construction builds a new set of features by combining multiple existing features with the expectation that their restructured form improves our chance to discriminate among the classes as compared to the original feature space.

4.1.2 Predictive modeling

Inextricably linked to the process of feature selection, *predictive modeling* refers to the utilization of selected features for the classification of data points. A predictive model is any mathematical function which maps data inputs to the class prediction. These models fall into different types of their own, depending on how they use features to arrive at the classification. Each model type offers distinct advantages and disadvantages, which therefore makes the choice of model important. Afterwards, the model must be validated to ensure that its predictions truly reflect the task at hand.

The primary objective of MS proteomic profile data analysis is to build a predictive model that is able to determine the target condition (case or control) for a given patient's profile. The predictive classification model is built from a set of SELDI-TOF- MS profiles (samples) assembled during the study. Each sample in the dataset is associated with a class label determining the target patient condition (case or control) we would like to automatically recognize. More formally, let D be a set of data pairs $\{< X_1, Y_1 >, < X_2, Y_2 >, \ldots, < X_n, X_n >\}$, where X_i denotes inputs and Y_i their designated outputs. In the case of proteomic profiles, X_i corresponds to profile readings (a vector of m/z intensity values) and Y_i to the class label: case or control (cancer or non-cancer). The objective is to build a predictive model $f : X \rightarrow Y$ that maps inputs (profiles) to outputs (labels) such that the mapping achieves high accuracy on future unseen profiles. The classification (prediction) refers to the process of applying the learned model $f : X \rightarrow Y$ to profiles and assigning the output label for them.

4.1.3 Evaluation of Classifier Methods

Our objective is to obtain models that achieve accurate predictions on future profiles. Since these examples are unobtainable, the ability of a classifier model f to generalize to such data is analyzed by splitting the available data into two subsets: a training set and a test set. The training set consists of profile samples used to pick the features and learn the model. The test set consists of profile samples withheld from the learning stage that are used to approximate the ability of the classifier to correctly predict future, yet to be seen,

data. The complete performance picture is given by the confusion matrix that represents the percentages of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) results. Secondary measures can be derived from the confusion matrix and include the following:

Error rate $(E) : FP + FN$
Sensitivity $(SN) : TP/(TP + FN)$
Specificity $(SP) : TN/(TN + FP)$
Positive predictive value $(PPV) : TP/(TP + FP)$
Negative predictive value $(NPV) : TN/(TN + FN)$

These performance measures can be computed for both the training set and test set. Test set results are more important since they testify about how the classifier generalizes to future data. However, the differences in training and testing performance statistics are still important and carry useful information. For example, a large separation between training and test errors is a sign of high variance of the estimates of the model parameters and indicates potential overfitting of the model.

The evaluation measures discussed above are appropriate indicators of a learning models performance under a 0-1 loss function that reflects the situation in which type 1 and type 2 errors (FP and FN) carry approximately the same weight. In the case where one type of error should be weighted more heavily, a binary classifier's performance can be captured and examined independent of the loss function in terms of the receiver operating characteristic (ROC). The separation of the two classes with different proportions of misclassification error types is measured and summarized using the Area Under the ROC Curve (AUC) score [41].

A number of pitfalls apply to the evaluation of predictive models. The first cautions against a perfect predictive model. Because of the intrinsic stochasticity in MS profile data, it may be impossible to obtain a model with zero expected error. Noise may simply obscure or remove important diagnostic information in features. A perfect model cannot be expected in this type of environment. Following this, even if we see a small error, we must be assured that the training and testing set were not chosen to particularly demonstrate this small error. Evaluation measures must be averaged over several, random data splits to reduce

the chance of biasing the evaluation with a lucky training/testing set split. Cross-validation techniques such as random subsampling, n-fold and leave-one-out validation can be applied to average evaluation measures over multiple data splits. And finally, the choice of which model to apply to a test set must be made solely on their performance on data in the training set. If we consider a single test set, a collection of predictive models will generate varying performance statistics. This defines a distribution of the performance statistic which is conditioned on the test set. Choosing the model that corresponds to the best value of this distribution is biased, since in most practical settings, the test data comes from the future (for example, a new patient comes into the hospital and is profiled). Since our test set changes, our distribution of performance is no longer guaranteed to be identical, and thus the best–performing model may not classify the future data well. Splitting the training set further into an internal validation set can allow for estimation of generalization performance. This process can be repeated multiple times and the results averaged in order to obtain a good estimate of generalization performance.

4.2 RELATED WORK

4.2.1 Filter Methods

Filter methods perform feature selection in two steps. In the first step, the filter method assesses each feature individually for its potential in discriminating among classes in the data. In the second step, features falling beyond some thresholding criterion are eliminated, and the smaller set of remaining features is used. This score–and–filter approach has been used in many recent publications, due to its relative simplicity. Scoring methods generally focus on measuring the differences between distributions of features. The resulting score is intended to reflect the quality of each feature in terms of its discriminative power. Many scoring criteria exist. For example, in the Fisher score [42],

$$V(i) = \frac{\mu_{(+)}(i) - \mu_{(-)}(i)}{\sigma^2_{(+)}(i) + \sigma^2_{(-)}(i)}$$

the quality of each feature is expressed in terms of the difference among the empirical means of two distributions, normalized by the sum of their variances. Table 6 displays examples of scoring criteria used in bioinformatics literature. Note that some of the scores can be applied directly to continuous quantities, while others require discretization. Scores can be limited to two classes, like the Fisher score, while others, such as the mutual information score, can be used in the presence of 3 or more classes. For the remainder of this chapter, we will assume our scoring metrics deal with binary decisions, where the data either belong to a positive (+) or negative (-) group.

Table 6: Examples of Univariate Scoring Criteria for Filter Methods

Criterion	References
Fisher Score	[43, 44]
SAM Scoring Criterion	[45, 46]
Student t–test	[47, 48]
Mutual Information	[49]
χ^2 (Chi Square)	[50, 51]
AUC	[52]
J–measure	[53]
$J5$ Score	[54]

4.2.2 Univariate feature selection

Differential scores allow us to individually rank all feature candidates. However, it is still not clear how many features should be filtered out. The task is easy if we always seek a fixed set of k features. In such a case, the top k features are selected with respect to the ordering imposed by ranking features by their score. However, the quality of these features may vary widely, so selecting the features based solely on the order may cause some poor features to be included in the set. An alternative method is to choose features by introducing the threshold on the value of the score. Unfortunately, not every scoring criterion has an interpretable meaning, so it is unclear how to select an appropriate threshold. One solution is to transform a scoring metric to a p-value. Regardless of whether the score is parametric (Fisher score, t-test) or nonparametric (wilcoxon rank-sum test), any score can be transformed into p-values through a permutation test (see section 5.1.1) For example, if the p–value threshold is 0.05 then there is a 5% chance the feature is not differentially expressed at the threshold value. Such a setting allows us to control the chance of *false positive* selections. These are features which appear discriminative by chance.

4.2.3 Multivariate feature set selection and controlling false positives

A natural step after doing univariate feature selection is to consider what combinations of features can work well. The high–throughput nature of biological data sources necessitates that many features (genes or MS–profile peaks) be tested and evaluated simultaneously. Unfortunately, this increases the chance that false positives are selected. To illustrate this, assume we measure the expression of 10,000 independent genes and none of them are differentially expressed. Despite the fact that there is no differential expression, we might expect 100 features to have their p–value smaller than 0.01. An individual feature with p–value 0.01 may appear good in isolation, but may become a suspect if it is selected from thousands of tested features. In such a case, the p–value of the combined set of the top 100 features selected out of 10,000 is quite different. Thus, adjustment of the p–value when performing multiple tests in parallel is necessary.

The *Bonferroni correction* adjusts the p–value for each individual test by dividing the

target p–value for all findings by the number of findings. This assures that the probability of falsely rejecting any null hypotheses is less than or equal to the target p. The limitation of the Bonferroni correction is that it operates under the assumption of independence and as a result it is becomes too conservative if features are correlated. Two alternatives to the Bonferroni correction are offered by: (1) the *Family–wise Error Rate method* (FWER, [55]) and (2) methods for controlling the *False Discovery Rate* (FDR, [45, 56]. FWER takes into account the dependence structure among features, which often translates to higher power. [56] suggest to control FDR instead of the p–value. The FDR is defined as the mean of the number of false rejections divided by the total number of rejections. The Significance Analysis of Microarrays (SAM) method [46] is used as an estimate of the FDR. Depending on the chosen threshold value for the test statistic T, it estimates the expected proportion of false positives on the feature list using a permutation scheme.

4.2.3.1 Multivariate filters In the experiments below, a set of four multivariate filters are evaluated.

- Leave-one-out AUC Drop Score (LOO-AUC)

  This score is calculated by evaluating a predictive model M_{all} on a complete set of features. One at a time, feature i is removed from the model and retrained. The AUC of the retrained model M_i is evaluated over 10-fold cross-validation. The score for feature i is given as:

$$\text{LOO-AUC}_i = AUC(M_{\text{all}}) - AUC(M_i) \tag{4.1}$$

- Random Forest Importance Score (RF-Import)

  This score uses Random Forests [57] to calculate the importance score for each feature. The importance score for feature i is given as:

$$\text{RF-Import}_i = \frac{\#\text{ times feature } i \text{ is selected as a splitting feature in a tree}}{\text{total } \#\text{ trees in forest}} \tag{4.2}$$

  where the selection of a splitting feature is done to optimize the Gini gain of a particular tree [58].

- ℓ_1-regularization Score (ℓ_1-Reg)

  This score uses the ℓ_1-regularized Elastic Net [59] to calculate the average regularization coefficient of each feature. The Elastic Net attempts to solve a linear regression problem in the following general form:

$$\min_{\beta \in \mathbb{R}^d} \frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - x_i^T \beta)^2 + \lambda P_\alpha(\beta) \tag{4.3}$$

  where

$$P_\alpha(\beta) = \sum_{j=1}^{d} \left[\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j| \right] \tag{4.4}$$

  is the elastic-net penalty, which involves the ℓ_1-norm. The penalty is weighted by λ over an optimization process. As this penalty changes over the schedule L of $\lambda_1 \cdots \lambda_L$, weights for features in β_l become more sparse, until only the most important features have the largest weight. For a sequence $L = \lambda_1, \cdots, \lambda_L$ and associated regression coefficients $\boldsymbol{\beta} = \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_L$ learned during the course of training an Elastic Net, the ℓ_1-regularization score is given as:

$$\ell_1\text{-Reg}_i = \sum_{l=1}^{L} \lambda_l \beta_{l,i} \tag{4.5}$$

- Adaptive Lasso ℓ_1-regularization score (Ad-Lasso)

  The adaptive lasso [60] is a technique which builds upon lasso regression [61] and is a similar approach to the Elastic Net. However, during the optimization of 4.3, we will instead weight each feature of $\mathbf{x}$ proportionally to its regression coefficients from an ordinary-least-squares regression (Although, since OLS has trouble with data having high correlation, I use ridge regression instead). The Adaptive Lasso score is given as:

$$\text{Ad-Lasso}_i = \min_{\beta} \| \| y - \sum j = 1^d x_i w_i \beta_j \| \|^2 + \lambda \sum_{j=1}^{d} |\beta_j| \tag{4.6}$$

  where $w_i = 1/\hat{\beta}_i$, $\hat{\beta}$ is obtained from ridge regression of X on Y.

4.2.4 Wrapper Methods

Wrapper methods [40] search for the best feature subset in combination with a fixed classification method. The goodness of a feature subset is determined using internal–validation methods, such as, k–fold or leave–one-out cross–validation [62]. Since the number of all combinations is exponential in the number of features, the efficiency of the search methods is often critical for its practical acceptance. Different heuristic optimization frameworks have been applied to search for the best subset. These include: forward selection, backward elimination [63], hill–climbing, beam search [64], and randomized algorithms such as genetic algorithms [65] or simulated annealing [66]. In general, these methods explore the search space (subsets of all features) starting with no features, all features, or a random selection of features. For example, the forward selection approach builds a feature set by starting from an empty feature set and incrementally adding the feature that improves the current feature set the most. The procedure stops when no improvement in the feature set quality is possible.

4.2.5 Embedded Methods

Embedded methods incorporate variable selection as part of the model building process. A classic example of an embedded method is CART (Classification and Regression Tree, [67]).

CART searches the range of each individual feature to find the split that optimally divides the observed data into homogeneous groups (with respect to the outcome variable). Beginning with the resulting subsets of the variable that produces the most homogeneous split, each variable is again searched across its range to find the next optimal split. This process is continued within each resulting subset until all data are perfectly fit by the resulting tree, or the terminal nodes have a small sample size. The group constituting the majority of the samples in each node determines the classification accuracy of the derived terminal nodes. Misclassification error from internal cross–validation can be used to backprune the decision tree and optimize its projected generalization performance on additional independent test samples.

Regularization or shrinkage methods [61, 68] offer an alternative way to learn classifi-

cations for data sets with large number of features but small number of samples. These methods trim the space of features directly during classification. In other words, regularization effectively shuts down (or zeros the influence of) unnecessary features. This is another way to effectively deal with correlations in the data.

Regularization can be incorporated either into the error criterion or directly into the model. Let $\mathbf{w}$ be a set of parameters defining a classification model (e.g., the weights of a logistic regression model), and let Error$(\mathbf{w}, \mathbf{D})$ be an error function reflecting the fit of the model to data (e.g., least–squares as likelihood–based error). A regularized error function is then defined as:

$$Error_{Reg}(\mathbf{w}, \mathbf{D}) = Error(\mathbf{w}, \mathbf{D}) + \lambda ||\mathbf{w}||,$$

where $\lambda > 0$ is a regularization constant, and $||.||$ is either the L1 or L2 norm. Intuitively, the regularization term penalizes the model for nonzero weights so the optimization of the new error function drives all unnecessary parameters to 0. Automatic Relevance Determination (ARD, [69, 70]) achieves regularization effects in a slightly different way. The relevance of an individual feature is represented explicitly via model parameters and the values of these parameters are learned through Bayesian methods. In both cases, the output of the learning is a feature–restricted classification model, so features are selected in parallel with model learning.

Regularization effects are at work also in one of the most popular classification frameworks these days: the support vector machine (SVM) [71, 72]. The SVM defines a linear decision boundary (hyperplane) that separates case and control examples. The boundary maximizes the distance (also called margin) in between the two sample groups. The effects of margin optimization are: unnecessary dimensions are penalized; only a small set of samples (support vectors) are critical for the separation. Both of these help to fight the problem of model overfitting.

4.2.6 Feature construction

Better performance can be often achieved using features constructed from the original input features. Building a new feature is an opportunity to incorporate domain specific knowledge

into the process and hence to improve the quality of features. For example, we have already seen that correlated features exist in proteomic data. Some of these sets of correlated features may relate to proteins or peptides which have similar biological function, or are activated in the same cellular signaling pathway. While incorporating this information is made difficult due to the limited information in TOF-MS data, an assembly of correlated features may still share a relationship. Machine learning methods exist which can construct new features from existing ones, in order to represent inter-feature relationships more succinctly. These methods include clustering, linear (affine) projections of the original feature space, as well as more sophisticated space transformations such as wavelet transformation. These feature construction approaches are briefly reviewed below.

4.2.7 Clustering

Clustering groups data components (data points or features) according to their similarity. Every data component is assigned to one of the groups (clusters); components falling into the same cluster are assigned the same value in the new (reduced) representation. Clustering is typically used to identify distinguished sample groups in data [73, 74]. In contrast to supervised learning techniques that rely heavily on class label information, clustering is unsupervised and the information about the target groups (classes) is not used. From the dimensionality reduction perspective, the groups identified by clustering define a new set of features and their values.

Clustering methods rely on the affinity matrix – a matrix of distances between data components. The affinity matrix can be built using one of the standard distance metrics such as Euclidean, Mahalanobis, Minkowski, etc, but more complex distances based on, for example, functional similarity of genes [75], are possible. Table 21 (see Appendix B) gives a list of some standard distance metrics one may use in clustering.

Clustering methods such as k–means clustering [76, 77], and hierarchical agglomerative clustering [78, 79]have been applied to group features in high-throughput data. When clustering features, the dimensionality reduction is achieved by selecting a representative feature (typically the feature that is closest to the cluster center, [80]), or by aggregating all features

within the cluster via averaging to build a new (mean) feature. If we assume k different feature clusters, the original feature space is reduced to a new k-dimensional space. An example method of feature clustering is to cluster features based on intra–correlation, and use the cluster center as a representative. Grouping together the most intra-correlated features removes redundancy in the data and exposes more diverse features.

4.2.8 Principal Component and Linear Discriminant Analysis

Principal Component Analysis (PCA), [26] is a widely used method for reducing the dimensionality of data. PCA finds projections of high dimensional data into a lower dimensional subspace such that the variance retained in the projected data is maximized. Equivalently, PCA gives uncorrelated linear projections of data while minimizing their least square reconstruction error. Additionally, PCA works fully unsupervised; class labels are ignored. PCA can be extended to nonlinear projections using kernel methods [81]. Dimensionality reduction methods similar to PCA that let us project high dimensional features into a lower dimensional space include multidimensional scaling (MDS) [82] used often for data visualization purposes or independent component analysis (ICA) [83]. The technique has been used extensively for classification of proteomic [84] and microarray data [85, 86], in addition to many other highly dimensional data types.

Principal component analysis identifies affine (linear) projections of data that maximize the variance observed in data. The method operates in a fully unsupervised manner; no knowledge of class labels is used to find the principal projections. The question is whether there is a way to identify linear projections of features such that they optimize the discriminability among the two classes. These techniques, termed *Discriminative projections* include Fisher's linear discriminant (FLD) [41], linear discriminant analysis [61] and more complex methods like partial least squares (PLS) [87, 88].

Take for example, the linear discriminant analysis model. The model assumes that cases and controls are generated from two Gaussian distributions with means $\mu_{(-)}$, $\mu_{(+)}$ and the same covariance matrix Σ. The parameters of the two distributions are estimated from data using the maximum likelihood methods. The decision boundary that is defined by data

points that give the same probability for both distributions is a line.

The linear projection is defined as:

$$\mathbf{w} = \mathbf{\Sigma^{-1}}(\mu_+ - \mu_-),$$

where $\mu_-, \vec{\mu}_+$ are the means of the two groups and Σ is the covariance for both groups, where $p(x|y) \sim N(\mu, \Sigma)$.

4.2.9 Wavelets

Wavelets are families of basis functions which are used as "building blocks" to approximate more complex functions. Wavelets have been used to approximate the complex MS protein profile data, since they appear as a superimposition of multiple wavy signals. The most popular wavelet transformation for MS protein profile data is the *Discrete Wavelet Transform* (DWT). In this wavelet model, the approximation for a curve $c(t)$ is given by:

$$c(t) = \sum_k s_{J,k}\phi_{J,k}(t) + \sum_{j=1}^{J}\sum_k d_{j,k}\psi_{j,k}(t) \tag{4.7}$$

The functions $\phi_{J,k}(t)$ and $\psi_{J,k}(t)$ are called *mother* and *father* functions, respectively. In the DWT method the mother and father wavelet functions are identical, so $\phi(t) = \psi(t) = 2^{-j/2}\phi(2^{-j}t - k)$. The j and k parameters control how these basis wavelet functions are translated and dilated. J controls the number of *scales*, and $k \in 1, ..., K_j$ is the number of coefficients at scale j. A scale of j means that each wavelet coefficient is spaced 2^j time units apart. The wavelet coefficients $s_{J,k}$ and $d_{J,k}$ reflect smooth and detailed behavior of the function at scale j. The DWT calculates these components through a pyramid algorithm [89], which transforms a datapoint into a vector of wavelet coefficients. Since this is a linear transformation, it can also be computed through matrix multiplication of a wavelet matrix W, which is implicitly computed during the DWT. This makes it convenient to exchange data points with wavelet coefficient representations at will.

4.2.10 Classifier Models

Many classifier models and learning approaches have been developed and are available for these classification tasks. Their common property is that they represent the mapping between inputs and outputs. For example, classifiers such as CART [67] (described briefly in section 4.2.5 and C4.5 [90] extract classification rules in terms of decision rules or trees. Some methods, including logistic regression, [61] determine the output by a learning set of parameters used to weight individual inputs. Other examples include support vector machines (SVM, [71, 72, 91], the naive Bayes classifier [92, 93] and multilayer neural networks [94–96]. In general, classification models attempt to partition a high-dimensional space of profile measurements (x), such that the case and control profiles fall into distinct regions. Many existing models, such as logistic regression or the SVM, achieve the partitioning by defining a linear decision boundary: a hyperplane that separates a high-dimensional input space x into two subspaces. Different models may use different optimization criteria.

For example, the SVM is a technique that computes a decision boundary between two classes by restricting its attention only to the samples (support vectors) that are most critical for separating the two groups. In our case, the decision boundary is a hyperplane that is maximally distant from the support vectors on either side of the hyperplane. The hyperplane is defined as:

$$w^T x + w_0 = 0 \tag{4.8}$$

with parameters w and w_0, where w_0 is the distance between the support vectors of each class, and w is the normal to the hyperplane. The parameters of the model may be learned through quadratic optimization with Lagrange parameters [72]. Then, the decision boundary is given by:

$$\hat{w}^T x + w_0 = \sum_{i \in SV} \hat{\alpha}_i y_i (x_i^T x) + w_0 \tag{4.9}$$

where α_i are Lagrange parameters obtained through the optimization process and y_i represents the class label for x_i with two possible values, -1 or 1. Note that only samples that correspond to support vectors (SV) define the hyperplane boundary, to which is normal.

The decision made by the classifier for a new input x is given by:

$$\hat{y} = sign\left[\sum_{i \in SV} \alpha_i y_i (x_i^T x) + w_0\right] \tag{4.10}$$

which corresponds to the side of the hyperplane on which the datapoint occurs, either positive or negative. The choice of the separating hyperplane in the SVM algorithm incorporates regularization effects which makes it less susceptible to overfitting [71].

4.2.11 Kernels

The dual problem formulation depends on the dot product $(\mathbf{x}_i \cdot \mathbf{x}_j)$. The dot product is a measure of similarity between the two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. The kernel trick [72] is used to map input vectors of arbitrary structure to a (potentially) higher-dimensional *feature space* through a replacement function called a *kernel*. The *kernel function* $K(\mathbf{x}_i, \mathbf{x}_j)$ replaces $(\mathbf{x}_i \cdot \mathbf{x}_j)$ with $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ where the mapping function $\Phi(\mathbf{x}_i)$ maps the vector $\mathbf{x}_i$ to a new feature-space. The "trick" is that the kernel function implicitly performs the mapping that would be done by ϕ, which reduces the complexity incurred by complex ϕ functions. Computing the kernel function instead redefines distances between points in the mapped feature space. The SVM draws a linear boundary in this new feature space, and upon returning to the original space, this boundary becomes bent into a nonlinear decision boundary. Figure 15 depicts nonlinear decision boundaries generated by nonlinear kernels. Certain kernels, such as a radial-basis kernel (bottom right), happen to suit this particular classification problem better than the standard linear kernel (top left).

This example demonstrates that different kernels may be helpful for classifying certain types of data. Special-purpose kernels have been constructed for many types of data, including strings, trees [97] and graphs [98]. These kernels take advantage of structure in the data to facilitate comparisons between data points. When such a kernel is unknown, one option is to choose from a number of popular choices. Some work has been done in automatically selecting the best kernel function from a list [99, 100]. The process works by extracting general features (meta-data) about each dataset. The meta-data is comprised of statistical ,distance-based and distribution-based measures which measure various qualities about the

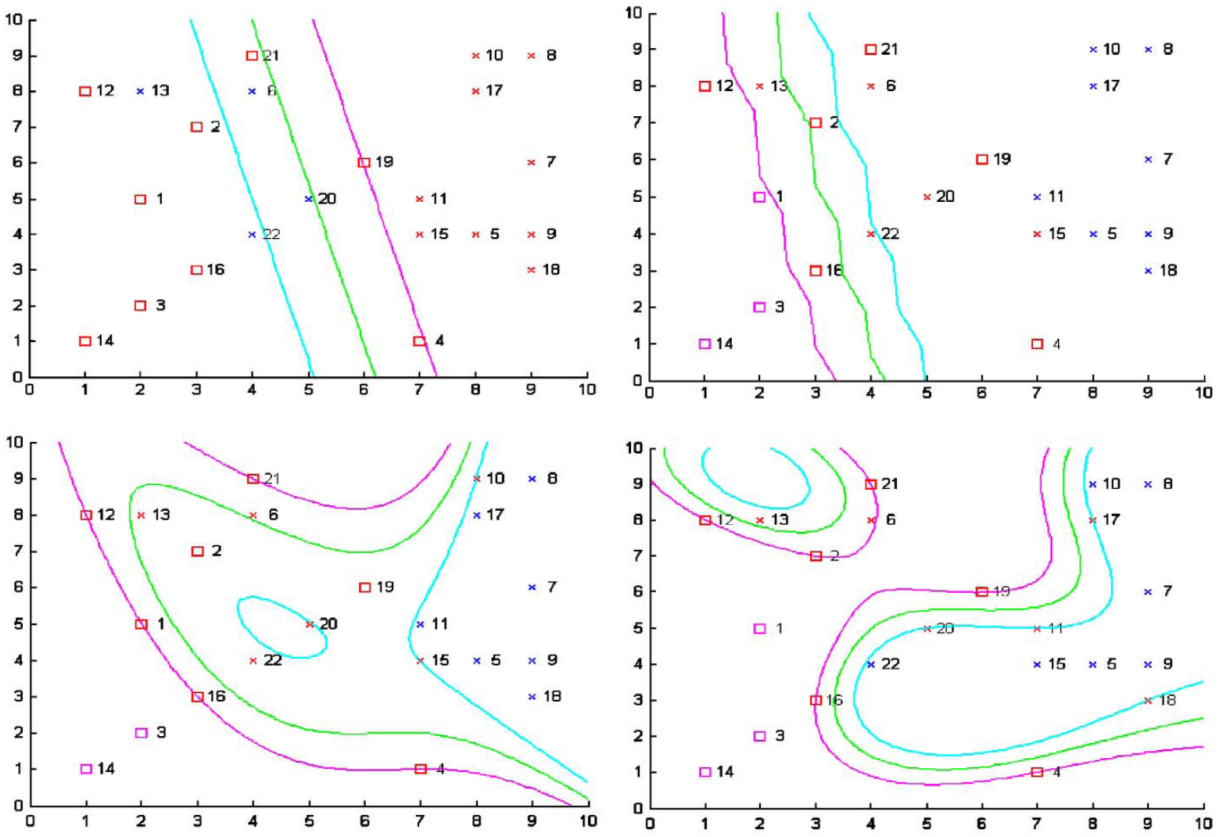

Figure 15: Example of SVM with different kernels on the same dataset.

data. A rule-learning approach is then taken to determine which qualities best reflect the applicability of a particular kernel. These rules are then applied to a test set in order to decide an appropriate kernel function.

An alternative is to learn an appropriate kernel. Some more recent work involves learning a kernel function [101, 102]. The basic approach involves learning a linear combination of kernels. Multiple basis kernels are used to project features into individual feature spaces. Weights for each kernel are used to optimize the margin as in the standard linear SVM approach. These weights can also be optimized for agreement to an ideal kernel, but the optimization method does not take misclassification error into performance, and can overfit. A quality-of-fit metric was developed [103] to resolve this issue, which further enabled a search over many kernels while penalizing complex kernels. The idea of a *hyperkernel* [103] was created to enable searches over parameters which govern the kernels. This potentially enables a kernel to be created from an infinite amount of basis kernels, but these combinations are always kept in check by a regularization quality.

The hyperkernel is enabled by a mathematical concept called the Reproducible Kernel Hilbert Space (RKHS). A *Hilbert Space* is any arbitrary-dimensional space, such as Euclidean space. However, we may want to consider spaces which are bigger or smaller, and where the idea of a "point" can carry a different meaning. Briefly, data points can be mapped to individual kernel functions, for example, Gaussians with the datapoint as the centroid. Thus, a 'point' in the space is really a continuous functional. Re-representing data points in this way complicates the process of measuring distances between them in this space, since the Gaussian functions are continuous. Instead of computing a distance, the RKHS allows computation of dissimilarities between these re-representations. Through the kernel trick, evaluating the dot product between these function re-representations ends up being as simple as evaluating the original kernel function of the two points, hence "reproducible kernel". A Hyper-RKHS (HRKHS) is a space where each point is its own RKHS. The differences between each point can be different parameterizations of the kernel functions, and computing the kernel function between two points calculates the dissimilarity between the involved RKHS spaces (namely, the difference in their parameterization of their own kernel function). Thus, similar datapoints are fit with similarly parameterized kernels, and

since the similarity metric is already computed, we need not worry about what happens in relation to the nested RKH Spaces. The result is that the decision boundary is optimized across all classes of kernels.

4.3 METHODS

High-throughput data naturally has many features, and it is important to distinguish those which are informative from the rest. Predictive models built on these features should strive to use the maximum amount of information made available. At the same time, the researcher should not be forced to do a post-hoc selection of feature selection and predictive model combinations - a certain level of confidence should be seen that good combinations can be arrived at automatically.

This dissertation intends to explore and analyze the aspects which make feature selection techniques and predictive models better at dealing with high-throughput data, with particular emphasis on MS proteomic profile data. Often times, a feature selection technique is employed for reasons that it simply improves the prediction error on a dataset, and does not often give any insight as to why this improvement is experienced. Perhaps alternative methods with similar properties exist, which can perform equally as well. The hypothesis is that *feature selection techniques which deal explicitly with correlation perform significantly better (in terms of classification performance) than those methods which do not.* The difference between correlation-aware techniques may be negligible. The evaluation of these techniques is demonstrated on both biological and simulated data.

Using a correlation-aware feature selection process is only half a step of the analysis. The other half comes from building a predictive model which is robust enough to deal with multiple classification problems, such as different diseases, yet specific enough to incorporate knowledge about the data type which may aid the classification. For example, classifiers which assume conditional independence between all features, such as Naive Bayes, may perform poorly on datasets where many features are interdependent. The experiments in section 4.4.1 show that indeed, this is the case. The SVM approach is known to work well in

spite of highly correlated features [61, 104]. Thanks to the kernel trick, I can create varied SVMs which are different predictive models, but still keep the advantage of the SVM against correlated features. Varying the kernel may result in improved classification performance, and providing a kernel which explicitly expresses valuable aspects of MS profiles may be the best option. The hypothesis is that *a kernel designed specially for proteomic data can perform equally as good as choosing among popular kernels, or learning a kernel.* The hypothesis is tested through evaluations on biological data.

4.3.1 Decorrelating Feature Selection

To keep the feature set small, the objective is to diversify the features as much as possible. The selected features should be discriminative as well as independent from each other as much as possible. The rationale is that two or more independent features will be able to discriminate the two classes better than any of them individually. Each feature may differentiate different sets of data well, and independence between the features tend to reduce the overlap of the sets. Similarly, highly dependent features tend to favor the same data and thus are less likely to help when both are included in the panel. The extreme case is when the two features are exact duplicates, in which case one feature can be eliminated. Figure 16 displays the phenomenon of correlated features in the pancreatic cancer dataset. Many of the features are correlated with any other feature by .8 or more. Filtering out highly correlated features therefore significantly reduces the amount of work needed to search for good, diverse features.

Correlation filters [25, 105] try to remove highly correlated features since these are less likely to add new discriminative information [80]. Various elimination schemes are used within these filters to reduce the chance of selected features being highly correlated. Typically, correlation filters are used in combination with other differential scoring methods. For example, features can be selected incrementally according to their p–value; the feature to be added next is checked for correlation with previously selected features. If the new feature exceeds some correlation threshold, it is eliminated [25].

The decorrelation filter method is designed to reduce the effect of correlations in selecting

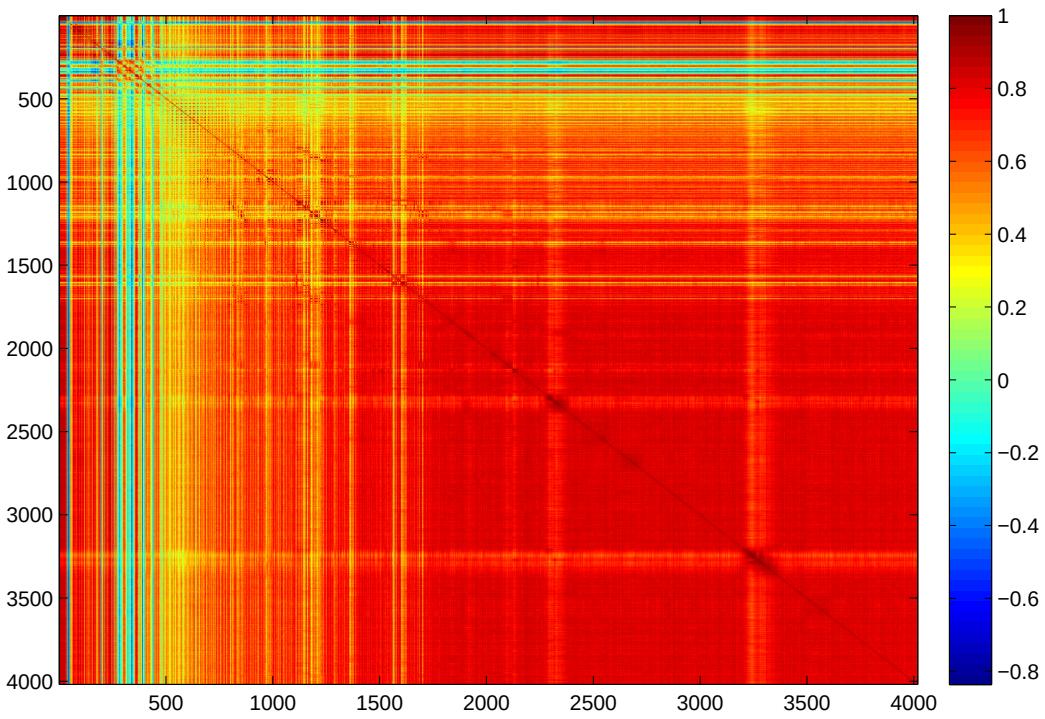

Figure 16: Correlation coefficient matrix between features

discriminative features. Multiple maximum-allowed correlation thresholds are used to create queues for features. Features in each queue can only be correlated with feature selected up to that stage by the MAC threshold. Since a correlated feature may still be able to contribute to the analysis, highly correlated features might still be selected. However, the overall diversity of the features should be improved.

Applying univariate analysis to each profile position allows us to rank the ability of each position to discriminate between case and control profiles. However, we are interested in building classifiers that utilize more than one feature and give the best possible classification performance. One simple solution to this problem would be to pick the top candidates determined by a univariate score. However, such a choice is not good for proteomic spectra in which signals at many positions, are highly correlated. This is illustrated in Figure 17 (left) that shows the heat map of the top 15 Fisher score positions for the pancreatic dataset. Note that all these positions appear to be good individual discriminants of case and control samples. However, all of them look alike, come from the same neighborhood and their signals are highly correlated (pairwise correlations are ≥ 0.97). Since such highly correlated signals are less likely to improve the discriminability of case and control samples, one may consider removing high correlates from the feature set. This goal can be achieved via correlation filtering that combines the removal of high (absolute value) correlates in feature sets rank-ordered using univariate scores. The strategy uses a maximum allowed correlation (MAC) threshold and incrementally selects the highest ranked feature such that its (absolute value) correlation with any of the previously selected features is below MAC. Figure 17 (right) shows the heat map for the top 15 Fisher score positions for the pancreatic data after correlation filtering with MAC=0.6. Profile positions differ widely giving us more opportunities to find good overall discriminants.

Experimenting with MAC thresholds on multiple cancer datasets (Pancreatic, Melanoma, Prostate, Ovarian and Lung cancer data) we have found that enforcing MAC thresholds tends to improve the quality of the feature set, but the best MAC value varies from dataset to dataset and it is different also for different univariate criteria. Thus, instead of searching for the best MAC we have developed a new feature selection procedure that combines the advantages of univariate feature scoring and de-correlation. Figure 18 gives a visual representation

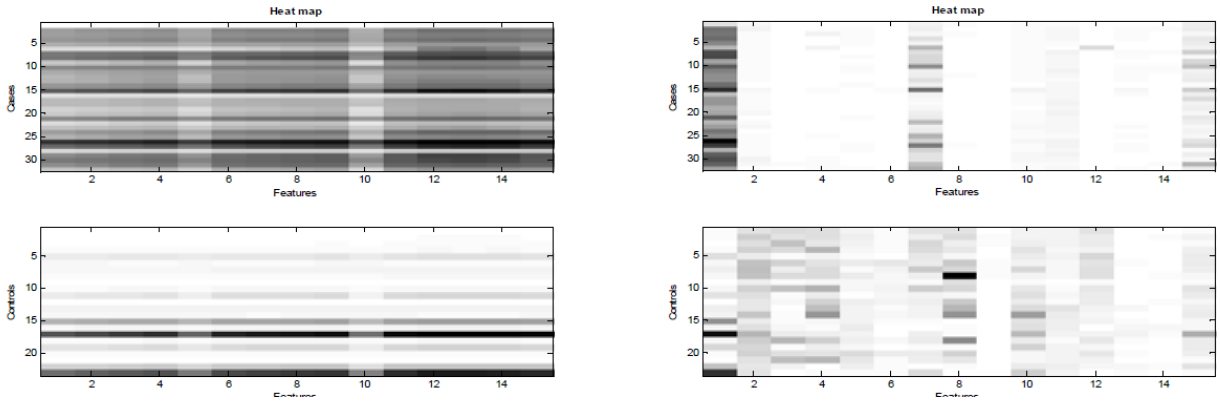

Figure 17: Left Panel: Heatmap of the top 15 profile positions (features) selected by the Fisher score, without correlation filtering. Right Panel: Heatmap of the top 15 Fisher features with correlation filtering (MAC=0.6). The top and bottom rows are case and control samples, respectively.

of the "parallel MAC" feature selection technique.

The parallel MAC feature selection procedure first rank-orders features using a given univariate differential expression score and then builds a feature set incrementally by choosing the best new feature from among multiple candidate features, each of them being the highest univariate score candidate at some fixed MAC level. The best feature is determined using a internal cross-validation scoring (10-fold is the default) based on a simple classification model such as a Nave Bayes or a linear Support Vector machine. Note that such an approach is different from the classic greedy wrapper approach that must scan and evaluate all ($\tilde{6}$0,000) possible candidate features. In contrast to this, the parallel MAC model scans and evaluates only feature candidates that correspond to highest ranked candidates at different MAC levels and the number of candidates com-pared depends on the number of MAC levels tracked. The resolution of the method may be con-trolled by increasing or decreasing the number of MAC thresholds.

This is one example of a method which directly interacts with the features to ensure that correlations are limited. PCA and wavelet transformation, discussed above, also deal explic-

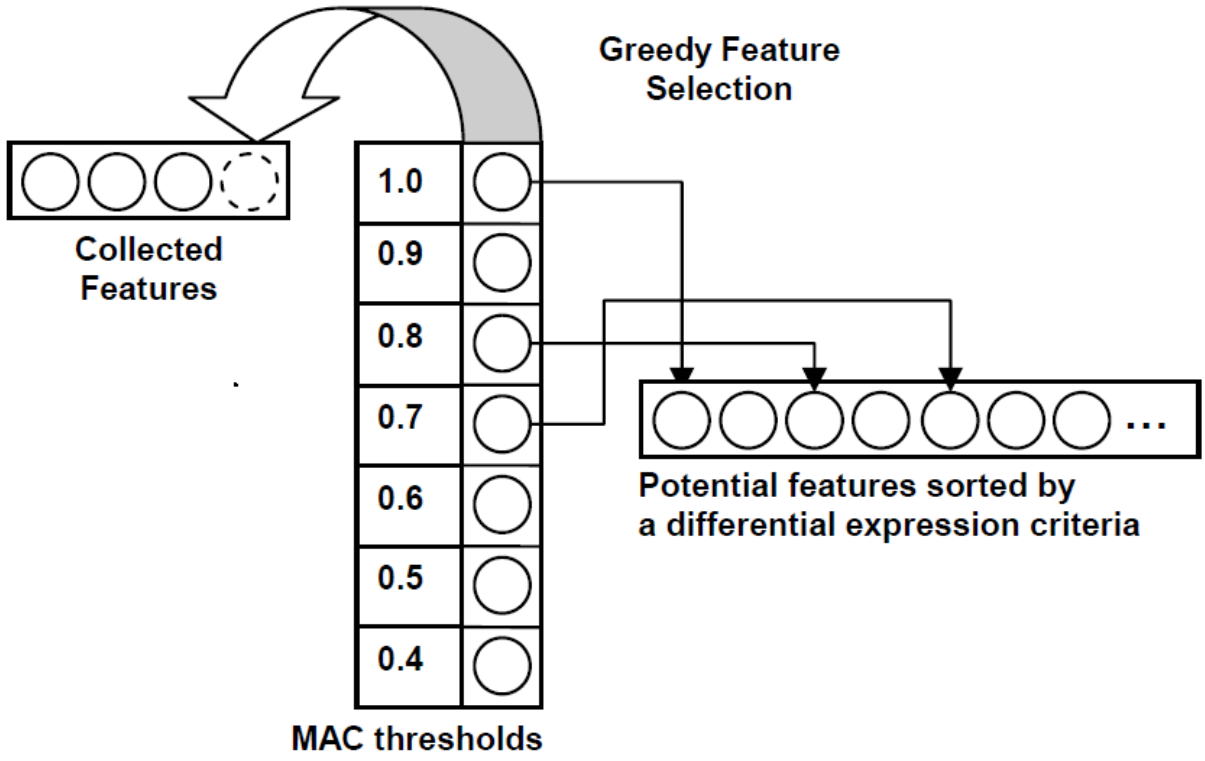

Figure 18: Schema of the parallel MAC feature selection technique.

itly with correlations by constructing features which tend to vary together. Certain feature selection techniques either assume that all features are independent, as in univariate feature selection strategies or embedded methods using classifiers which ignore feature dependencies. The fact of the matter is, discriminatory information must be found while dealing with the overwhelming amount of correlation present in raw MS profile data. However, with limited empirical knowledge about which features interact with each other, the complexity of feature selection seems limited.

With this in mind, I investigate whether any feature selection technique which accounts for correlations in the data can perform significantly better than the popular techniques which do not. Moreover, I investigated whether any technique taking advantage of correlations is statistically indistinguishable from other correlation-aware methods. In this case, the parallel MAC method would do just as well as PCA, wavelet transformations, or any other decorrelating method at selecting discriminative features.

Feature selection techniques are divided into two groups - those that handle correlations and those that do not. The techniques are taken from those discussed above throughout this chapter. 40 splits of training and testing data are performed. On each split, all feature selection techniques from either group are applied to the training data. The resulting features are fed to a linear SVM. The choice of a linear SVM in this situation is to minimize the complexity of the classifier as much as possible. The classifier is only used as a general evaluation technique for widely varied methods of feature selection. Later on, in the evaluation of classifiers, this process can be repeated to determine if either group of feature selection techniques influences the best way to decide which classifier to use. The error rate of the classifier is used to evaluate the feature selection method. Two distributions of errors are created - those resulting from feature selection ignoring correlates, and those resulting from feature selection accounting for correlates.

The Mann-Whitney U-test [106] is a nonparametric test which estimates the likelihood that two samples of data come from the same distribution. The null hypothesis in this case is that *the distributions of classification performances (errors) from the correlate-ignoring and the correlate-aware groups are identical*. The U-test determines the U-value (a p-value analog) associated with the distance between the two distributions. The nonparametric

nature of the test is necessary because the distributions of the errors are unknown and not assumed to follow any predefined distribution. A smaller U-value indicates that the null hypothesis is less likely to be true. In this case, I am expecting to reject the null hypothesis; this would suggest that the difference between error distributions of the two feature selection groups is more likely to be statistically significant.

I am also interested in determining whether any of the correlate-aware feature selection methods are statistically distinguishable from one another. For this, the Kruskal-Wallis non-parametric one-way test [107] is used. It is essentially an extension of the Mann-Whitney U-test to 3 or more groups. This test determines whether the medians of the groups (distributions of errors given by the individual correlate-aware feature selection methods) are equal. In this case, the null hypothesis is that these distributions of errors are identical, because they have equal medians and similarly scaled variances. I expect that it will be very hard to reject this null hypothesis at a reasonable significance level.

4.3.2 Kernel Comparison

The Support Vector Machine is a robust classifier framework which can be adapted to many types of data. However, several alternatives exist to the popular linear kernel which may be more effective at classifying MS protein profile data. The questions are, does a suitable kernel exist for MS protein profile data, and if so, can it be chosen automatically? The following describes a comparative study to determine whether a suitable kernel for this data can be discovered. Three separate approaches are presented.

4.3.3 Automatic Selection Among Predefined Kernels

The first approach is to learn rules about datasets which indicate the best classifier to use. This approach is similar to that taken in [99], where various statistical measures (henceforth referred to as *dataset characterization statistics*) are used to characterize all datasets available for training. The datasets are then classified using a handful of predefined kernels. Datasets are grouped based on which kernel performs best, and a rule-learning classifier (CART or C4.5) is used to generate relationships between the datasets' characterization statistics and

the best performing kernel. The testing dataset's characterization statistics are calculated and the learned rules are applied to determine the best kernel to use for classifying the data. A list of characterization statistics is given in Appendix B, Table 23. Some popular kernels are given below: The d^{th} order polynomial kernel (Equation 4.11a), radial basis kernel (Equation 4.11b), spline kernel (Equation 4.11c), multiquadratic kernel (Equation 4.11d) and Laplacian kernel (Equation 4.11e). Algorithm 1 describes the process formally.

$$K(X_i, X_j) = (\langle X_i^{\mathrm{T}} X_j \rangle + 1)^d \tag{4.11a}$$

$$K(X_i, X_j) = \exp(-\frac{||X_i - X_j||^2}{2\sigma^2}) \tag{4.11b}$$

$$K(X_i, X_j) = 1 + (X_i^{\mathrm{T}} X_j) + \frac{1}{2}(X_i^{\mathrm{T}} X_j)\min(X_i^{\mathrm{T}} X_j)^2 - \frac{1}{6}\min(X_i^{\mathrm{T}} X_j)^3 \tag{4.11c}$$

$$K(X_i, X_j) = (||X_i - X_j||^2 + \tau^2)^{1/2} \tag{4.11d}$$

$$K(X_i, X_j) = \exp\left(-\frac{|X_i - X_j|}{h}\right) \tag{4.11e}$$

The benefit of this approach is that heterogenous data can be used to enhance learning about the proper choice of kernels. Thus, with additional time and data, this approach has the possibility of becoming stronger. However, if none of the predefined kernels are able to capture any salient relationships about features of the data, then this technique will struggle. The remaining two approaches focus on constructing more appropriate kernels, in the case that one is not readily available.

4.3.4 Learning a Customized Kernel

A second approach is to learn the kernel through Hyperkernels. The hyperkernel optimizes a decision boundary of points over a class of kernels. However, the kernels selected from the list above are not in the same class (that is, governed by the same set of parameters). A suitable class of kernels would be, for example, several radial basis kernels with different

values of σ. Still, in the previous approach, we made several distinct kernels available in the hopes that their diversity would allow better classification of datasets. By sticking to a class of kernels which are only radial basis functions, we take away a lot of the diversity. Instead, we might consider the wealth of distance metrics already defined for clustering techniques in table 21. In any kernel function which incorporates Euclidean distance (for example, those in Equations 4.11b and 4.11d), we can perhaps substitute any of the clustering distance metrics and therefore create a class of kernels limited only by the number of available distance metrics. To construct a hyperkernel from this new class of metrics, a RKHS is represented using a kernel function from this class. Optimizing the decision boundary in the hyperkernel results in a decision boundary which is a linear combination of boundaries resulting from the diverse distance metrics in the constituent RKH Spaces. Algorithm 2 summarizes the evaluation procedure for evaluating a hyperkernel learned from the proteomic data.

4.3.5 Learning the Hyperkerneled SVM

The hyperkernel using linear combinations of kernels is defined as [103]:

$$k(x_p, x_q) = \sum_{i,j=1}^{m} \alpha_{ij} \sum_{l=1}^{n} c_l k_l(x_i, x_j) k_l(x_p, x_q) \tag{4.12}$$

where $k \in k_1, ..., k_n$ is a kernel function from the class of kernels considered. $c \in c_1, ..., c_n$ are the weights for the linear combination of kernels. $\alpha_{i,j}$ are variables to be optimized to minimize the Q_{reg} (Regularized Quality of fit) loss function defined in [103]:

$$Q_{reg} = \min_{\beta} \min_{\alpha} \frac{1}{m} \sum_{i=1}^{m} max(0, 1 - y_i f(x_i)) + \frac{\lambda}{2} \alpha^T K \alpha + \frac{\lambda_Q}{2} \beta^T \underline{k} \beta \tag{4.13}$$

Minimizing Q_{reg} in this formulation maximizes the hyperplane separating the training data. This quantity can be optimized as an instance of semidefinite programming (SDP, [103, 108]), which is an optimization technique that ensures that the optimized combinations of matrices (in this case, the linear combinations of RKHS kernels) remain positive semidefinite. This is important since the hyperkernel should overall be positive semidefinite to be a true kernel. For a linear SVM using a soft margin loss function [72], the SDP is formulated as follows, with notation following:

$$
\begin{array}{ll}
\underset{\beta,\gamma,\eta,\xi,\chi}{\text{minimize}} & \frac{1}{2}t_1 - \chi\nu + \xi^{\mathrm{T}}\frac{1}{m} + \frac{\lambda_Q}{2}t_2 \\
\text{subject to} & \chi \geq 0, \eta \geq 0, \xi \geq 0, \beta \geq 0 \\
& ||\underline{K}^{\frac{1}{2}}\beta|| \leq t_2 \\
& \begin{bmatrix} G(\beta) & z \\ z^{\mathrm{T}} & t_1 \end{bmatrix} \succcurlyeq 0
\end{array}
\tag{4.14}
$$

where $z = \gamma y + \chi \mathbf{1} + \eta - \xi$.

Some notation is briefly presented. The training data $X_{\text{train}} = x_1, ..., x_m$ and $Y_{\text{train}} = y_1, ..., y_m$ consist of m examples. γ and χ are Lagrange multipliers. η and ξ are vectors of Lagrange multipliers from the Wolfe dual of the SDP. β are hyperkernel coefficients. t_1 and t_2 are auxiliary variables for the SDP.

The pseudo-inverse of a matrix K is denoted $K^{\dagger}$. The hyperkernel Gram matrix $\underline{K}$ is defined by $\underline{K}_{ijpq} = \underline{k}((x_i, x_j), (x_p, x_q))$. The kernel matrix K is given by $K = \text{reshape}(\underline{K}\beta)$, which reshapes an m^2 by 1 vector $\underline{K}\beta$ to a m by m matrix. $Y = diag(y)$, is a matrix with y on the diagonal and zeroes elsewhere. $G(\beta) = YKY$. $\mathbf{I}$ is the identity matrix. $\mathbf{1}$ is a vector of ones. $y \odot z$ is the element-by-element multiplication operation.

The classification function for the optimized SVM is given by

$$
f = \text{sign}(KG(\beta)^{\dagger}(y \odot z) - \gamma) \tag{4.15}
$$

SDPs of the type represented in Equation 4.14 can be optimized using the MATLAB tools SeDuMi [109] and YALMIP [110].

4.3.6 Using a Proteomics-Specific Kernel

A third approach is to design a kernel which is well-suited to picking out the differences between MS protein profiles. The natural idea is to take advantage of ideas learned from feature selection. Certain techniques like wavelet decomposition and PCA can identify features which seem to vary on the same levels. These features should probably be selected and compressed into features which can be useful for a linear separation.

Overall, this task is more difficult than the previous one, because custom kernels typically capture the features of the data which are anticipated to differentiate dissimilar datapoints. The string kernel, for example, computes similarity measures based on the amount of matching substrings. The tree kernel also computes similarity based on the structure of the tree. It is debatable what constitutes salient structural features of proteomic profiles, but perhaps the best thing to do is consider the data source from its originally intended biological standpoint. MS profile data from diseased patients should display a deviation from the data produced from healthy individuals. Many diseases manifest themselves through disregulation of biological processes which stay within some bounds while an individual is considered "normal" or "healthy". When these processes are disturbed by a disease condition, the deviation effect can be seen, for example, as an increase in inflammatory response proteins. Even though this response is not very disease-specific, it certainly indicates that the individual is experiencing from an adverse condition. Many of these deviating responses may be present in the MS protein profile data, but they are hard to find and quantify.

Defining what "normal" means is not a straightforward task. A control population can vary over the course of study of a particular disease, and heterogenous case/control datasets are bound to sample from different portions of the true "normal" distribution. In fact, many controls for one study could be cases for another. As a result, this methodology is left somewhat open-ended. Using all available data (even if it is heterogeneously generated with respect to the target testing set) can be important for defining normalcy.

My approach for a proteomics-specific kernel involves grouping m/z positions into pathways, which individually attempt to summarize the status of biological regulatory mechanisms in the profile. The kernel-mapping function ϕ maps a protein profile to a set of pathway

features, which indicate the level of disregulation of those pathways as viewed through the profile. Thus, profiles can be classified through the kernel by their arrangement based on similarly-disregulated pathways. I refer to this approach as the "Pathway Kernel".

4.3.6.1 Defining the Pathway Kernel The advantage of this "Pathway Kernel" is that it uses prior knowledge in the form of gene pathways to guide the learning process. The method correlates these pathways either positively or negatively with previously unseen, incoming data to subtype them into one class or another. Each dataset used in the learning phase offers its own perspective on how proteomic spectra can be separated, but this method allows some backtracking to see which pathways are responsible for those divisions.

The general methodology involves learning differentially expressed patterns from heterogenous proteomic datasets. These patterns are tested for validity by a permutation test, in a fashion similar to GSEA (Gene set enrichment analysis, [111]). GSEA was a technique developed for microarray data which evaluated the strength of a multivariate feature set versus other combinations of features. Combinations of features (genes) are selected either at random or chosen by prior knowledge (i.e. genes related in a biological pathway). The strength of the feature set in distinguishing between treatment / no treatment groups is measured by an aggregate statistic over univariate t-test scores for the genes in the set. The statistic is compared to a null hypothesis distribution, estimated by permuting treatment/ no treatment labels and rescoring the gene set. In a paper by Atul Butte's group [112], GSEA was used to analyze pathway perturbations in microarray data. The pathway perturbations (in terms of correlation with a case/control label) were significantly useful in discriminating among subtypes of disease. An improvement on GSEA developed by Tibshirani and Efron (termed Gene Set Analysis, GSA [113]) was more general and robust, due to the fact that the null hypothesis distribution was computed not only with respect to randomized sample labels, but also the performance of random gene sets in the original data.

I wanted to evaluate the strength of pathways in the proteomic data, yet retain some notion of biological relevance, which would make the results more interesting in the downstream analysis. Rather than search among all combinations of m/z values for pathways, I wanted to find a way to take in prior knowledge to drive the feature selection going into the

pathway analysis. Harvard's Molecular Signature Database (MSigDB, [111]) is a database of gene sets which have defined relationships, such as proximity on the chromosome, biologically studied and curated interactions and genes with common gene ontology terms. I took these 5400 gene sets and estimated the m/z values where their produced proteins should appear in a proteomic profile. Translation of gene identifiers to protein profile features is done through the following process. Gene identifiers in MSigDB are given in the HUGO format specified by the Human Gene Nomenclature Committee [114], which associates each gene identifier with a UniProt protein identifier [10]. The UniProt identifiers are in turn used to retrieve amino acid sequences for the peptides associated with each gene. Using the amino acid sequences, the nominal masses of these peptides are calculated by summing the monoisotopic masses of their component amino acids, plus the weight of a water molecule. The resulting mass is given a single charge and aligned to a feature in the profile. This process is repeated for all genes in a MSigDB pathway. The gene sets defined by MSigDB become mapped to "protein sets" to be analyzed through the GSA technique. Pathways found to be significantly correlated by GSA are then retained and used to construct aggregate features from profiles. Profiles are then reduced to a small set of aggregate pathway features which are used by a standard kernel function, such as the linear kernel. The distance between points in this space represents the similarity of how pathways are regulated between samples. Algorithm 3 outlines the procedure for computing and evaluating the pathway kernel.

4.4 EXPERIMENTS AND RESULTS

The output of the experiments in this chapter are intended to provide guidelines for the choice of feature selection and classification models for MS profile data. I investigate whether any feature selection method which is accounting for correlation is statistically equivalent. Achieving a standard for feature selection is important to establish analytical protocols for MS profile data. The kernel selection investigation is important because many researchers only choose the model structure after seeing all methods' performances post hoc. Any clarity resulting from these experiments will give new directions into how to select models (either

automatically or by hand) based on clearly defined characteristics of data. Although feature selection and classification are not new topics, the data type is novel and the opportunity for novel insight in handling this data can be uncovered by the above described methods.

4.4.1 Feature Selection

The goal of feature selection is to find a small group of discriminative features. Our features in this data type may pertain to *surrogate biomarkers* which indicate the presence or absence of a disease. Our set of selected features should ideally be diverse. Redundant features in our selected set do not add additional information, and their selection implies the exclusion of other, potentially useful features. Some predictive modeling methods, such as Naive Bayes or logistic regression, are easily influenced by closely correlated features and may result in a poorly fit model. Unfortunately, proteomic profiling data is highly correlated by nature. A charged particle is often preceded by and followed by its lighter and heavier isotopes, respectively. This means that neighboring features on the x-axis have a high chance of being dependent, and a visible correlation results.

To demonstrate this, I used the Vanderbilt Lung Spore IMAC data to calculate the absolute value of the correlation between each unique feature pair in the dataset. This resulted in 30278^2 pairs and their correlation values, whose distribution is plotted in Figure 19. Approximately 77% of the feature pairs are correlated above 0.8. Consider a hypothetical dataset of the same size, where the features are independent of their neighbors. Such a dataset may be generated by sampling randomly from a Gaussian distribution until the equivalent number of features has been reached. The correlated structure of the true proteomic data is lost. This is seen in Figure 20 as the distribution becomes centered around 0. In this example, there is no mass in the distribution beyond correlation values of 0.5. Table 7 lists the percentage of feature pairs which are correlated greater than 0.8 to illustrate that every dataset demonstrates this bias. This experiment establishes the existence of an extreme bias towards correlates in proteomic data, and with this in mind, we should seek feature selection methods which avoid this bias.

In order to evaluate the ability of feature selection methods, I performed the following

experiments on both raw and SAP datasets. The 40 train/test splits were identical to those used in Chapter 3. First, I evaluated the performance of univariate filter methods, which rank features based on their individual relation to the class label. Next, multivariate feature filters were analyzed. Rather than evaluating the individual impact of features, the multivariate filtering approaches take into account how well features collaborate with each other to create a set which helps more together than the sum of its individual parts. Finally, I investigated techniques which take advantage of correlation in the data. These techniques either eliminate or construct new features based on their correlation, as well as reduce the dimensionality of the data.

In all cases, the predictive model used was a linear Support Vector Machine (SVM) with L2-norm regularization. The reason for choosing the L2-norm was because I intended to select only a small set of features, and the less-aggressive form of regularization would allow me to demonstrate the impact of having too many correlated features in the predictive model. ℓ_1-norm regularization would perform additional feature selection and it would be difficult to separate the effect of the predictive model on performance from the feature selection steps used.

4.4.1.1 Demonstrating the effect of correlation on filter methods

Univariate methods are the simplest feature selection method, and often the least computationally expensive. However, they can be influenced by highly correlated components in the data, and the fact that in high-dimensional, low-sample size data, some discriminative signals may arise simply by chance.

Due to the nature of proteomic data, intensity values from neighboring positions on the m/z axis are highly correlated. Thus, a large "peak" feature which appears discriminative (either genuinely or by chance) will also spread its high univariate score with its neighbors. This is a bad case in either scenario - either a good signal with lots of redundant features is selected, but a predictive model fit to them will not be robust, or a spurious signal with lots of redundant features is selected, but a predictive model will overfit to the false signal and fail to classify future data correctly.

Table 8 lists the percentage of feature pairs selected by the univariate t-test filtering

method which are highly correlated (magnitude of sample correlation above 0.8). This relationship persists even after preprocessing, as can be seen in Table 9. In the situation where future profiles are generated which do not express this feature (say, for example, a different subgroup of disease), the predictive model is poorly informed about where to look for additional information, resulting in a greater rate of error.

Although some in the protein profiling community "bin" local features together to form an aggregate peak feature [115], I do not prefer this approach. It is difficult to know for sure when separate features are indicating different molecules (in fact, ideally by the nature of the data, each feature should be a different molecule). Discriminative features can also lie along the sides of peaks or in the troughs between them, which makes evaluating peak extraction difficult. I prefer to address the problem of high correlates in univariate filters by employing a decorrelation filter. Tables 8 and 9 demonstrate the percentage of highly correlated feature pairs and AUC from a t-test filter that was applied in conjunction with correlation filtering. The maximum-allowed correlation (MAC) threshold was set to 0.6. Note that the percentage of highly correlated feature pairs drops drastically, indicating that the selected features are more diverse in the information that they carry. As a result, average AUC of models trained on these features increases (fourth column). Note that sometimes, noise in the raw data (and especially on harder datasets) causes a drop in AUC. This effect disappears with preprocessing.

Also note that applying decorrelation is not guaranteed to improve the AUC, especially if features must be selected randomly if remaining features are correlated too highly. In this case, we may even select a feature which does not collaborate well with other features in the set. This is where multivariate filtering approaches begin to play a role.

Multivariate ranking methods directly address the problem of selecting a panel of features which performs well, although not necessarily by decorrelating them. Tables 8 and 9 show (in column 5) the percentage of highly correlated feature pairs resulting from the RF-importance multivariate filter. These percentages are not necessarily lower than t-test with decorrelation, but are lower than from a plain univariate t-test filter. However, since these features are selected in a way which makes their combination more useful, rather than focusing on their individual ability, we see an average increase in AUC for models built using these features

(column 6).

4.4.1.2 Comparison of univariate versus multivariate filtering For the sake of being thorough, the following experiments were performed in order to compare univariate and multivariate filter methods, which support, and are supported by,the above analysis.

A selection of univariate filter methods were evaluated on each dataset. The Fisher-like score, Wilcoxon signed-rank score, SAM-score and t-test were applied, and the 20 highest-scored features were subselected and passed to the predictive model. See Table 22 for the definition of these scores. The choice of the scoring metrics was based on their frequency of use in the protein profiling research community. Most univariate scores can be computed in a single vectorized operation. Certain scoring metrics (e.g. the $t-$test score) are parametric while others (e.g. the Wilcoxon score used here) are nonparametric. To account for the disadvantage or advantage, I simply report the performance of the best-performing filter method in each experiment. The leftmost column in Table 10 displays the best average AUC obtained by the four performance models trained on the top 20 filtered features.

I evaluated four multivariate ranking approaches and reapplied them to the identical train/test splits as the previous experiment, on raw data. These methods were the Random Forest Importance filter, the Leave-One-Out AUC Drop filter, the ℓ_1-regularization filter and the adaptive lasso regularization filter. The scores are defined in Section 4.2.3.1. The performance of the best model out of the four multivariate filter methods is reported in the center column of Table 10. The rightmost column indicates the advantage of the best multivariate model over the best univariate model. The decisions are split, with about half the data being at a disadvantage from multivariate filtering methods. The primary reason for this is because we are working with the raw data, which contains discriminative features by chance and are biologically unrelated to the disease state. For example, consider a dataset where either cases or controls all suffer from a baseline shift. Almost all features will appear discriminative due to the baseline shift between classes. Non-neighboring features will be considered by multivariate rankers, and since it is less likely that they will be correlated, they will be added to the list of top features, on the basis of their ability to discriminate between classes. However, because of the noise in the raw data, it makes it difficult for the predictive

model to estimate the appropriate weights for these features, and the variance of the estimates for the parameters of the model will increase. Rather than deal with a "single" parameter (for the correlated feature chosen by univariate methods), a predictive model working with multivariate features from raw data will struggle with the poorly estimated parameters for multiple features. The result is a poorer model performance.

Table 11 displays the alternative to these experiments on SAP-preprocessed data. Here, the noise has been removed from the data, and in general, improvements can be seen in the performance of both univariate and multivariate models. Note however that the advantage to multivariate models is more frequently positive. Table 12 displays the percentage of feature pairs selected by each multivariate filter which are highly correlated. Compared to Table 14, the amount of correlated feature pairs is greatly reduced. Three datasets still suffer a performance disadvantage. Multivariate ranking filters do not explicitly control for correlation; the success of the multivariate filter largely depends on its ability to identify correlated features and substitute them among one another. In fact, this is a difficult operation to achieve computationally. The number of combinations of features in this extremely high-dimensional setting is prohibitive to exhaustive evaluation. However, efficient techniques such as the Parallel Decorrelation algorithm discussed in Section 4.3.1 exist which can simultaneously reduce the feature space while simultaneously accounting for substitutability.

4.4.1.3 Correlation-based feature extraction and construction Rather than expect a multivariate filter to learn the substitutability of features, we can apply techniques which account for and take advantage of their substitutability. I repeated the above experiments, using 3 correlation-aware methods to select or construct features. The Parallel Correlation, Top-K PCA Eigenvector reduction, and Wavelet Decomposition methods served this purpose. In each case, the "top 20" features are taken: the parallel decorrelation filter selects 20 features, the top-20 PCA eigenvectors are used for projecting the test data, and the coefficients from the first 20 levels of wavelet decomposition are used. The best performing models are reported in Table 13 for the raw data, and Table 14 for the SAP-preprocessed data.

On difficult datasets like COPD and Diabetes, where few good features might exist, the

noisy nature of the raw data makes itself apparent. Univariate filters will find the good feature and its immediate correlates. Good discriminative features which are uncorrelated with this initial feature are hard to come by, and likewise the correlation-aware methods suffer. This is interesting, because we would expect these uncorrelated but discriminative features (even if they are spurious) should exist just by random chance in the raw data. While the disadvantages are mostly small, it's likely that these raw datasets contain noise which correlates the most highly discriminative features (such as a baseline shift separating classes). Being forced to select a nondiscriminative feature just because all the others are correlation-substitutable can add noise to the model, decreasing performance.

4.4.2 Predictive Modeling

In order to evaluate the ability of predictive modeling methods, I performed the following experiments on both raw and SAP-preprocessed datasets. The 40 train/test splits were identical to those used in all previous experiments. Different predictive models are evaluated by changing the kernel function of an ℓ_1-norm SVM. The choice of the ℓ_1-norm is to enable aggressive, decorrelating feature selection through the predictive model itself (without needing to choose a method from Section 4.4.1).

Three different kernel learning approaches were evaluated, first on the raw data. Table 15 displays the average AUC of the ℓ_1-norm SVM when learning either a Hyperkernel, a Metadata-based kernel or the Pathway Kernel described in Section 4.3.6. Results using a standard linear kernel can be obtained from Tables 2 and 3 in Chapter 3. As can be expected, the kernel-learning approaches suffer from the noisy raw data.

Table 16 displays the results of the previous experiment when applied to SAP-preprocessed data. All three kernel-learning approaches almost always improve. The three cases where performance does not improve occur on different datasets, under different methods (Hyperkernel on ILD, Metalearning on Hepatitis C and Prior Knowledge on the Vanderbilt Lung WCX dataset). For many datasets, the effect of preprocessing gives a substantial advantage to the kernel-learning methods. The stability of the performance estimates improves for the Hyperkernel method, as evidenced by the smaller confidence intervals and higher average

AUC. The same cannot be said of the Metalearning or Prior Knowledge kernels, however.

The Pathway Kernel replaces the entire original profile data with new features. It is interesting to see this method improve with the preprocessing, especially because the features are almost totally unrelated to protein expression amounts by the time the SVM is applied. However, it is not a good kernel choice for this type of data. The pathway kernel can suffer from many sources of noise due to its design. The conversion between Gene identifier and protein is one step wrought with error [116], and furthermore, the mapping of these proteins onto profile locations is also problematic. Although the Pathway kernel method improves with preprocessing, it does not appear that the method outperforms a classifier which guesses randomly. For data types which present more information about the features' biological identities, perhaps the pathway kernel can be more useful. With TOF-MS data, however, too much is lost in translation between the gene sets and protein features to take advantage of this kind of prior knowledge.

The prior knowledge kernel has a direct biological interpretation. Helpful features reflect an overall overexpression or underexpression of a set of proteins in the profile. For the sake of interest, I listed the top 20 pathways used as features for each dataset. This list is in Appendix C, Table 24. The pathway names are taken from MSigDB [111], so the interested reader is directed to the Molecular Signatures Database [1] to browse for further information on pathways which do not have a readily interpretable name (for example, pathways in the C4 set of MSigDB, which are given a nonbiological identifier).

4.4.3 Discussion

It is difficult to make guarantees about what model will perform the best on any given dataset. However, just because an assurance of a good model doesn't exist doesn't mean we shouldn't try to achieve it. The above experiments point us in a favorable direction which should motivate our choices for feature selection and predictive modeling.

[1] http://www.broadinstitute.org/gsea/msigdb/index.jsp

4.4.3.1 Recommendations for feature selection First, our choice about feature selection is largely made due to the desire to have a parsimonious explanation for a disease pattern. Although univariate filtering methods are the extreme of sparseness, the nature of proteomic data defeats their application. High correlations and substitutability among a large percentage of the features hinder the robustness of predictive models based solely on univariately filtered features. In general, their use is not recommended for the development of a robust model.

Multivariate filters start to alleviate the issue of correlation among features. Still, searching exhaustively through many combinations of features can be computationally expensive. Additionally, multivariate filters are not explicitly aware of substitutable features, and are not guaranteed to return a robust set of orthogonal features.

The feature selection techniques which are aware of correlations (PCA, wavelet decomposition and the Parallel Decorrelation technique presented above) can deal with the substitutability of features. Interpreting the resulting features can be difficult, but not impossible.

One thing we learned from our experiments is that performing feature selection often produces a better performing model. Moreover, performing preprocessing adds an additional advantage. Preprocessing with SAP leads to the results in Table 14. Even though we saw in Section 3.3 that SAP is probably a suboptimal method, the advantage to correlation-aware techniques is largely positive for all datasets. It becomes easier to find uncorrelated features which are discriminative, and this adds to the robustness of the model, improving its performance.

I tested the significance of the advantages granted by using certain techniques. Here, "Advantages" are used to measure the difference in model performance between the application of a procedure and without the procedure applied (Advantage = Performance(after feature selection) - Performance(before feature selection)). Performance advantages must be taken into context with the difficulty of the dataset being classified. As datasets become more easily classified, it gets harder to create additional advantage, simply because there is little room for improvement. Likewise, a difficult task faces a large room for improvement, and perhaps almost anything can cause a large jump in performance. By measuring performance advantages, we somewhat limit the impact that easy or hard datasets have in the

hypothesis testing analysis.

First, I tested whether the advantages from using any of the correlation-aware methods were significantly different from one another. Columns 1 and 4 from Table 17 show the percentage of times correlation-aware methods were not significantly different from one another. The significance tests done in this table are by way of the Mann-Whitney U-test, a nonparametric rank-sum significance test for two populations. In this case, the populations consist of performance advantages in AUC obtained over the 40 splits of training and testing data. On half the datasets, none of the correlation-aware techniques are significantly better than any other. On the other half of the datasets, it's possible to select out significantly better method ($p < 0.01$). In the majority of cases where this occurs, it's due to the parallel decorrelation method being significantly better than the wavelet decomposition and PCA methods. The statistical separation of the parallel decorrelation method increases after preprocessing with SAP (Table 17, column 4).

Next, I tested whether the advantages from using correlation-aware methods were significantly higher than the advantages from using univariate and multivariate filter methods. These tests are also reflected in the additional columns in Table 17. Again, the Mann-Whitney test was used. On the raw, data, 6 of the datasets have no significant difference between correlation-aware and multivariate methods half the time. On the other 8 datasets, there are more often advantages to the correlation-aware approach. The percentage of times that correlation-aware techniques are significantly better decreases when the data is preprocessed by SAP. Univariate methods are frequently significantly poorer in the raw data - this is just a consequence of the noise in the raw data. The separation between these methods mostly remains in the SAP-processed data.

Third, I tested whether the advantages obtained by using any feature selection method after applying SAP preprocessing were significantly higher than advantages resulting from applying the same methods to the raw data. Performance advantages from applying all feature selection methods on SAP data were tested against performance advantages from applying all feature selection methods on raw data, using the Kruskal-Wallis non-parametric one-way test [107]. This is an extension of the Mann-Whitney test for 3 or more groups - in this case, the advantages for all SAP methods are grouped together, and likewise with

advantages from methods applied to the raw data. The Kruskal-Wallis test operates under the null hypothesis that the samples come from populations such that choosing random observations from both samples will only result in one observation being greater half the time.

In the final column of Table 17, the p-value resulting from the Kruskal=Wallis test is given for each dataset. In only three of the datasets (COPD, Breast cancer and nearly ILD), the p-value was less than 0.01. This suggests that, very frequently, performing feature selection on SAP-preprocessed data is not likely to give you the same performance advantage as simply performing feature selection on raw data and building a model. Rather, the advantage will be greater with the SAP-processed data.

4.4.3.2 Recommendations for predictive model choice

I analyzed different kernel learning approaches to improve the SVM classifier, and I wished to test whether any method offers a consistent advantage over another. Evaluating the kernel-learning approaches was an exploratory process. The decision about what model to use in the first place because of familiarity with the model, simplistic reasons, or personal preference. When we want to consider whether better representations of the data exists in order to achieve a better classification, it becomes necessary to evaluate other models. For example, I once used the Pathway Kernel to map proteomic profiles into their pathway features, and then considered whether a linear classifier could separate the resulting transformed feature space. Unfortunately, the feature space was perhaps more convoluted than before, but I wondered whether or not a more advanced kernel method could separate the data in this feature space. The result would be beneficial for the sake of interpretation - heavily weighted features would have a direct biological interpretation. To this end, this foray into kernel learning was developed.

From Tables 14 and 16, there are a few things to take note of. First, the Meta-learning approach learned to only use the linear kernel on SAP data. This made the method not much better than the results presented in Chapter 3 to evaluate the SAP method. On the raw data, the noise in the profiles causes the Meta-Learning kernel to switch between kernels often, but the performance was not competitive with the other kernel-learning methods (table not shown). After SAP preprocessing, the Meta-Learning approach realized that the

best kernel was the linear kernel, and applied it continuously with only a few exceptions. The COPD, Scleroderma and Melanoma datasets made use of the polynomial kernel and multilayer-perceptron kernel twice each during 4 of the training and testing splits (but each dataset on a different split, so these anomalies seem to be more about the individual datasets than the train/test split). Moreover, many of the dataset metrics become useless under such high dimensionality, and would not differentiate many datasets after averaged over more than a handful of features. In the end, it seems the Meta-learning approach relies more on the internal cross-validation to choose a kernel, and is not appropriate for high-dimensional data such as MS protein profiling.

The Hyperkernel approach does not perform very well, given its complexity. Although it produces predictive models which are better than random in some cases, the best univariate filter approaches without decorrelation can outperform this method. The computational complexity in terms of time and memory of solving the Semidefinite Program is quadratic in the number of profiles, which is quite a large overhead considering other kernel methods are linear in the number of profiles.

Finally, the Pathway kernel is not a good option at this point in time. Due to the problems involved in translating gene sets to protein location sets in the profile, the kernel cannot provide a good re-representation of the data. This makes it difficult to use the kernel to build a predictive model. However, if features could be labeled as proteins in the profile, then the Pathway Kernel should be employed to take advantage of this information.

Table 18 displays the percentage of datasets on which a performance advantage of a kernel-learning approach was statistically significantly better than another. Four kernels are evaluated: The hyperkernel, the meta-learning kernel, the pathway kernel and the linear kernel selecting 20 features by way of the Parallel Decorrelation algorithm. Performance advantages are calculated by subtracting the average AUC from the plain linear kernel on SAP-processed data from the average AUC of the kernel-learning approach's predictive model. These performance advantages undergo a Mann-Whitney U-test for significance at $\alpha = 0.5$. Since we know the Meta-learning kernel approach is acting as a plain linear SVM, we can interpret the results accordingly. The only reason why the linear kernel would outperform the Parallel Decorrelation kernel would be due to the limit of the number of

features. In the end, the linear kernel outperforms other kernels most often. Although the parallel decorrelation-mapped kernel also performs well, it may be worth saving the aggravation about how many features to pick, and simply go with a linear SVM kernel.

4.4.3.3 Using the decision-theoretic approach to recommend a predictive model

The analyses of different classification methods in terms of AUC statistics summarize how well the models are able to discriminate among cases and controls for many different diagnostic thresholds. However, in practice we want to pick just one decision threshold.

One (simple) way to pick the decision threshold is to optimize the misclassification using the zero-one cost function, where each misclassification contributes equally to the error score. This is clearly suboptimal in many real-world cases where the cost of misdiagnosing a patient with a disease when he is actually healthy is not equal to the cost of misdiagnosing a diseased patient as healthy. To find the optimal diagnostic threshold we can adopt a decision-theoretic framework where each prediction/disease contingency is assigned a utility (or a cost). Then, the optimal decision threshold is the one that optimizes the expected utility defined by these contingencies.

Let $p(y = a|x)$ be the probability the profile x is a, where $a = \{disease, healthy\}$, and let $p(y = \neg a|x) = 1 - p(y = a|x)$ denote the probability it is not a. Let $u(a', a)$ be a utility function that assigns a real number to the true disease state a' and the decision made a. The utility is the negative of the corresponding mismatch cost.

The optimal decision $a*$ for x is then defined as:

$$a*(x) = \underset{a}{\operatorname{argmax}}\, P(y = a|x)u(a, a) + P(y = \neg a|x)u(\neg a, a).$$

The optimal decision can be rewritten in terms of a simple decision threshold on $P(y = \text{disease}|x)$. Briefly, we should choose "*disease*" if:

$$P(y = \text{disease}|x) \geq \frac{u(\neg\text{disease}, \neg\text{disease}) - u(\neg\text{disease}, \text{disease})}{u(\neg\text{disease}, \neg\text{disease}) - u(\neg\text{disease}, \text{disease}) + u(\text{disease}, \text{disease}) - u(\text{disease}, \neg\text{disease})} \tag{4.16}$$

otherwise the choice should be "*healthy*".

The utility function $u(a', a)$ is typically elicited from an expert. However, the probability model is not; it is learned from data. Some classification models and methods, such as Naive Bayes, logistic regression or Linear Discriminant Analysis are probabilistic and attempt to estimate $P(y|x)$ directly. Others, such as the SVMs which were used extensively in our previous investigations, do not have an immediate probabilistic interpretation. To build a probabilistic model for the SVM, we can apply the probabilistic transformation model introduced by Platt [117]. Let $f(x)$ denote the value assigned to the profile x by the SVM model trained with labels $+1$ corresponding to the disease and -1 to the control labels. Then the probability $P(y = \text{disease}|x)$ is modeled as:

$$P(y = \text{disease}|x) = \frac{1}{1 + \exp(-wf(x) + w_0)}$$

where w, w_0 are parameters fitted on the training data.

The probability model $p(y|x)$, whether it is based on the Naive Bayes or an SVM, is an estimate of the true model that is learnt from data. Its accuracy depends on the quality of the learning algorithm and the number of examples available to learn it. Now, the question is whether we can evaluate the goodness of the model and which metric we should judge it by. One way to approach the task is to rely on the empirical likelihood measures based on the test data. Briefly, the probabilistic model M learned on the train data is applied to every datapoint (x, y) in the test set and is used to calculate the predictive likelihood $P(y|x, M)$. The likelihood of M on the data is then calculated as $\prod_{(x,y)\in D_{\text{test}}} p(y|x, M)$. A log-likelihood metric $\sum_{(x,y)\in D_{\text{test}}} logp(y|x, M)$ is more practical and prevents us from reaching small values and resulting numerical operation problems.

Table 19 displays the predictive log-likelihood of our three kernel-learning approaches as well as that of the linear kernel used to evaluate SAP in Chapter 19. A number closer to 0 is better. Most of the time, the Hyperkernel-based SVM seems to produce the best-calibrated model for converting SVM outputs to probabilities. It is interesting to note that the Vanderbilt Lung datasets, which are generated from the same set of samples, all benefit the most from the pathway kernel-based SVM. In fact, the UPCI Lung Cancer dataset and the Vanderbilt Lung IMAC dataset share a large subset of samples, so it is no coincidence

that they share similar preferences for certain methods. Given these results, the hyperkernel-based approach may have some merit when it is required to produce a probabilistic output.

input : Raw training dataset D, set of kernels $K = \{K_1, \cdots, K_p\}$

output: Selected kernel K_s

Estimate performance of each kernel via cross-validation.

for $i \leftarrow 1$**to** 10 **do**

  Split D into training data D^{train} and validation data D^{valid} by subsampling.

  train-metrics$(i) \leftarrow$ Calculated dataset characterization metrics for D^{train}

  for $j \leftarrow 1$**to** p **do**

    perf$(j) \leftarrow$ performance of SVM with kernel K_j, trained on D^{train} and tested on D^{valid}

  end

  Save the index of the best performing kernel on this dataset.

  sel-kern$(i) \leftarrow$ index$(max($perf$))$

end

Learn decision rules between data characterization metrics and best performing kernel.

rules $\leftarrow$ CART-learn(train-metrics,sel-kern)

Use learned rules to select best kernel on training data.

data-metrics $\leftarrow$ Calculated dataset characterization metrics for D

$s \leftarrow$ CART-apply(rules,data-metrics)

Algorithm 1: Procedure for Evaluating Predefined Kernel Selection

- Repeat over many splits of training and testing data
  1. Split dataset D_i into training data D_{train} and testing data D_{test}
  2. Define the class of kernels $K = k_1, ..., k_l$
  3. Optimize the hyperkernel using K, D_{train} and the SDP in Equation 4.14
  4. Use an SVM to classify D_{test}.

Algorithm 2: Procedure for Learning and Evaluating Custom Hyperkernel

1. Repeat over many splits of training and testing data
   1. Split dataset D_i into training data D^{train} and testing data D^{test}
   2. Evaluate the GSA score of each pathway
   3. For those k pathways which are found to be statistically significant w.r.t D_{train} by GSA
      a) Compute the differential expression of the features in those pathways. This defines a set of values which are related to the differences between the classes in D_{train}.
      b) Compute the correlation between these differential expression values and the intensities of features for the same pathway for each profile $i \in D_{test}$. This results in a single value P_k for that pathway.
   4. Let $\phi(D_{test_i}) = \{P_1, ..., P_k\}$
   5. Use a linear kernel $K = \langle \phi(x_i), \phi((x_i^{test})) \rangle$
   6. Use an SVM to classify D^{test}.

Algorithm 3: Procedure for learning the pathway kernel

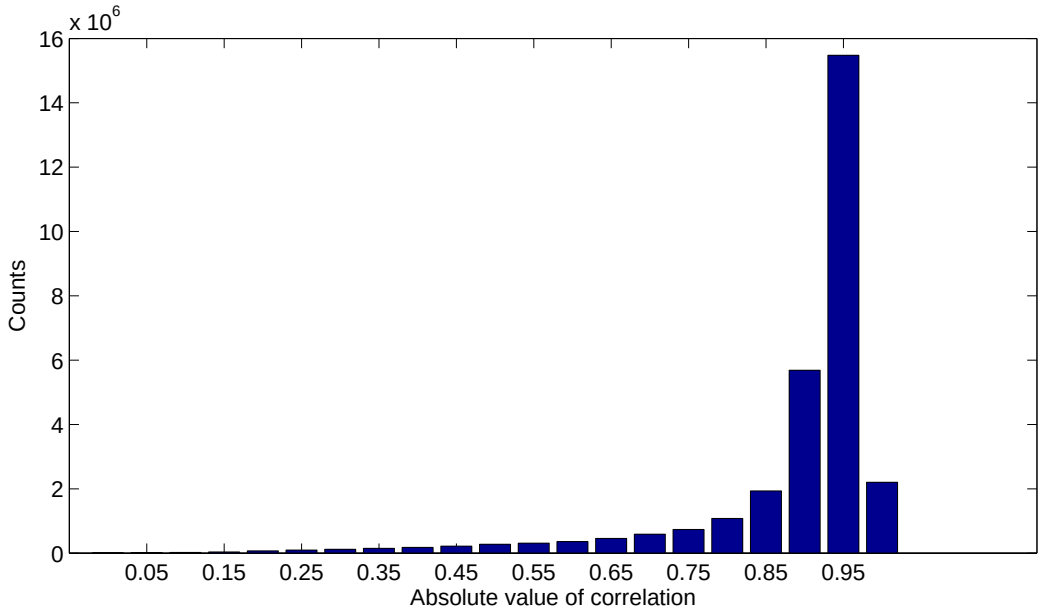

Figure 19: The distribution of absolute correlation values among all feature pairs for the Vanderbilt Lung IMAC dataset. 77% of the feature pairs are correlated higher than 0.8, demonstrating the correlation bias prevalent in proteomic MS data.

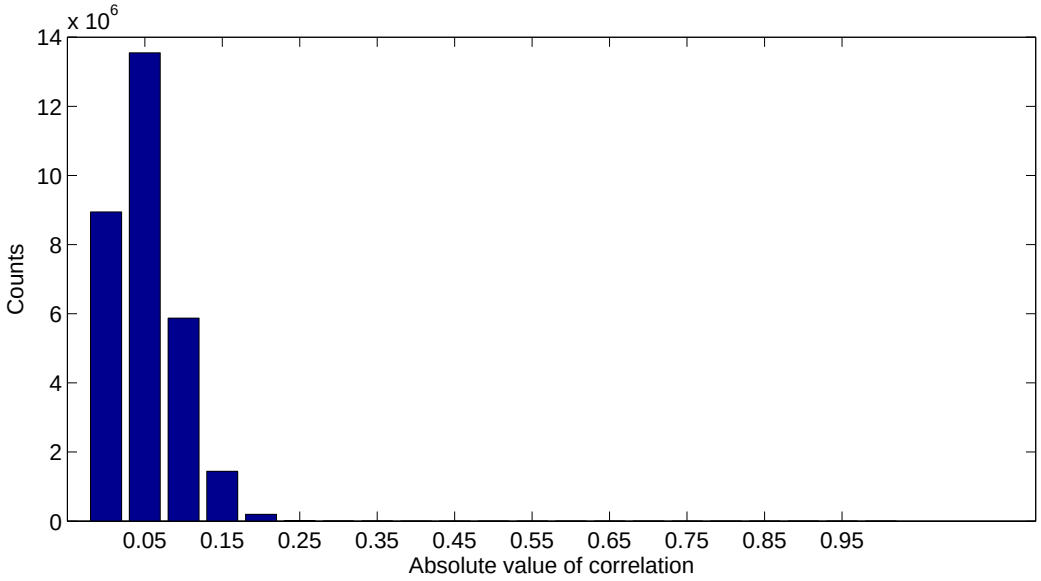

Figure 20: The distribution of absolute correlation values for a hypothetical dataset generated by randomly sampling from a Gaussian distribution. The distribution is more heavily tailed towards low values of correlation. 0% of the probability mass is correlated higher than 0.5.

Table 7: Percentage of the distribution of feature pair absolute correlation value above 0.8. For most datasets, more than two thirds of the features are highly correlated with each other. For the Vanderbilt Maldi Lung dataset, 52% should not be striking, since this dataset has roughly twice as many features as the other SELDI datasets.

Dataset	Percentage of distribution mass above 0.8
COPD	77%
Hepatitis C	77%
ILD	77%
Diabetes	95%
Melanoma I	77%
Breast Cancer	77%
Pancreatic Cancer I	77%
Pancreatic Cancer II	77%
Prostate Cancer	96%
Scleroderma	77%
UPCI Lung Cancer	78%
Vanderbilt Lung IMAC	77%
Vanderbilt Lung WCX	77%
Vanderbilt MALDI	52%

Table 8: Demonstration of the effect of high correlates among selected features in raw data. Columns indicate the percentage of feature pairs correlated above 0.8 using either a univariate t-test filter, a decorrelated t-test filter at 0.6 MAC, or the RF-Importance multivariate filter. Average AUC using each of these feature sets is also given.

Dataset	T-test	AUC	T-Test MAC=0.6	AUC	RF-Import	AUC
COPD	89%	0.6270	1%	0.5680	0%	0.5672
Hepatitis C	56%	0.5060	4%	0.6588	2%	0.5069
ILD	77%	0.5475	14%	0.5344	1%	0.6377
Diabetes	100%	0.5247	2%	0.5578	4%	0.5064
Melanoma I	81%	0.5777	19%	0.5357	1%	0.5188
Breast Cancer	96%	0.5429	25%	0.5261	10%	0.5561
Pancreatic Cancer I	50%	0.8019	1%	0.8575	8%	0.8939
Pancreatic Cancer II	37%	0.6819	2%	0.8338	5%	0.8122
Prostate Cancer	40%	0.9269	2%	0.9490	15%	0.9032
Scleroderma	41%	0.6993	2%	0.7428	1%	0.7155
UPCI Lung Cancer	50%	0.6225	2%	0.7450	9%	0.7228
Vanderbilt Lung IMAC	56%	0.7686	6%	0.9096	16%	0.8518
Vanderbilt Lung WCX	49%	0.8227	15%	0.8681	4%	0.8616
Vanderbilt MALDI	95%	0.7848	8%	0.8474	32%	0.8343

Table 9: Demonstration of the effect of high correlates among selected features in SAP data. Columns indicate the percentage of feature pairs correlated above 0.8 using either a univariate t-test filter, a decorrelated t-test filter at 0.6 MAC, or the RF-Importance multivariate filter. Average AUC using each of these feature sets is also given.

Dataset	T-test	AUC	T-Test MAC=0.6	AUC	RF-Import	AUC
COPD	52%	0.5699	2%	0.5961	1%	0.6413
Hepatitis C	56%	0.4862	8%	0.6038	8%	0.6444
ILD	100%	0.5035	1%	0.5428	3%	0.6136
Diabetes	100%	0.5996	2%	0.5597	1%	0.6334
Melanoma I	31%	0.5388	27%	0.5314	9%	0.5677
Breast Cancer	29%	0.5832	2%	0.5122	2%	0.5581
Pancreatic Cancer I	31%	0.5608	0%	0.8793	17%	0.9088
Pancreatic Cancer II	100%	0.5240	2%	0.8501	45%	0.9130
Prostate Cancer	72%	0.6711	1%	0.8107	7%	0.8716
Scleroderma	41%	0.5350	0%	0.6022	7%	0.5491
UPCI Lung Cancer	78%	0.6188	0%	0.7360	15%	0.7240
Vanderbilt Lung IMAC	47%	0.7236	2%	0.8572	25%	0.8790
Vanderbilt Lung WCX	47%	0.7379	0%	0.8857	99%	0.8912
Vanderbilt MALDI	54%	0.6328	11%	0.8562	9%	0.8426

Table 10: Comparison of Univariate vs Multivariate Feature Selection on Raw Data

Dataset	Best Univariate	Best Multivariate	Advantage
COPD	0.6270 ± 0.2210	0.5638 ± 0.2604	-0.0632
Hepatitis C	0.5484 ± 0.2212	0.5368 ± 0.2326	-0.0116
ILD	0.5549 ± 0.2784	0.6027 ± 0.2448	0.0477
Diabetes	0.5474 ± 0.2313	0.5355 ± 0.2662	-0.0119
Melanoma I	0.5825 ± 0.1653	0.5208 ± 0.1969	-0.0616
Breast Cancer	0.5698 ± 0.1343	0.5559 ± 0.1246	-0.0139
Pancreatic Cancer I	0.8310 ± 0.0828	0.8939 ± 0.0679	0.0629
Pancreatic Cancer II	0.7437 ± 0.1086	0.8433 ± 0.0779	0.0996
Prostate Cancer	0.9190 ± 0.0291	0.9047 ± 0.0330	-0.0143
Scleroderma	0.7479 ± 0.1071	0.7298 ± 0.1071	-0.0180
UPCI Lung Cancer	0.6411 ± 0.0958	0.7964 ± 0.0789	0.1553
Vanderbilt Lung IMAC	0.7686 ± 0.0682	0.8518 ± 0.0579	0.0832
Vanderbilt Lung WCX	0.8558 ± 0.0611	0.8616 ± 0.0528	0.0058
Vanderbilt MALDI	0.8107 ± 0.0632	0.8512 ± 0.0520	0.0405

Table 11: Comparison of Univariate vs Multivariate Feature Selection on SAP Data

Dataset	Best Univariate	Best Multivariate	Advantage
COPD	0.6386 ± 0.2116	0.6670 ± 0.1811	0.0284
Hepatitis C	0.6622 ± 0.1538	0.6269 ± 0.1680	-0.0353
ILD	0.6831 ± 0.1905	0.6404 ± 0.2174	-0.0427
Diabetes	0.6657 ± 0.1997	0.6090 ± 0.2158	-0.0567
Melanoma I	0.5804 ± 0.1697	0.5844 ± 0.1811	0.0040
Breast Cancer	0.5838 ± 0.1337	0.6079 ± 0.1364	0.0242
Pancreatic Cancer I	0.9009 ± 0.0516	0.9349 ± 0.0412	0.0340
Pancreatic Cancer II	0.9057 ± 0.0510	0.9130 ± 0.0525	0.0072
Prostate Cancer	0.8510 ± 0.0578	0.9574 ± 0.0214	0.1064
Scleroderma	0.5908 ± 0.1392	0.6031 ± 0.1416	0.0124
UPCI Lung Cancer	0.6917 ± 0.0802	0.7304 ± 0.0790	0.0387
Vanderbilt Lung IMAC	0.8083 ± 0.0680	0.8790 ± 0.0532	0.0708
Vanderbilt Lung WCX	0.8976 ± 0.0475	0.9206 ± 0.0373	0.0230
Vanderbilt MALDI	0.8044 ± 0.0656	0.8469 ± 0.0597	0.0425

Table 12: Percentage of feature pairs with $|r| \geq 0.8$

Dataset	LOO AUC Drop	RF Importance	L1 Regularization
COPD	0%	1%	0%
Hepatitis C	0%	8%	3%
ILD	61%	3%	21%
Diabetes	6%	1%	1%
Melanoma I	6%	9%	11%
Breast Cancer	0%	2%	1%
Pancreatic Cancer I	52%	17%	5%
Pancreatic Cancer II	33%	45%	9%
Prostate Cancer	4%	7%	2%
Scleroderma	18%	7%	3%
UPCI Lung Cancer	4%	15%	2%
Vanderbilt Lung IMAC	5%	25%	4%
Vanderbilt Lung WCX	6%	99%	11%
Vanderbilt MALDI	14%	9%	7%

Table 13: Performance of Feature Decorrelating Methods on Raw Data

Dataset	Best Univariate	Best Feat. Constructor	Advantage
COPD	0.6270 ± 0.1979	0.5839 ± 0.2316	-0.0432
Hepatitis C	0.5381 ± 0.2026	0.6486 ± 0.1855	0.1105
ILD	0.5502 ± 0.2677	0.6035 ± 0.2128	0.0533
Diabetes	0.5479 ± 0.2570	0.5155 ± 0.2731	-0.0324
Melanoma I	0.5777 ± 0.1713	0.5797 ± 0.1599	0.0021
Breast Cancer	0.5698 ± 0.1225	0.5780 ± 0.1331	0.0082
Pancreatic Cancer I	0.8304 ± 0.0776	0.8923 ± 0.0686	0.0620
Pancreatic Cancer II	0.7361 ± 0.1097	0.8207 ± 0.0897	0.0846
Prostate Cancer	0.9190 ± 0.0291	0.9698 ± 0.0188	0.0508
Scleroderma	0.7186 ± 0.1066	0.7204 ± 0.1120	0.0018
UPCI Lung Cancer	0.6411 ± 0.0958	0.8368 ± 0.0659	0.1957
Vanderbilt Lung IMAC	0.7686 ± 0.0726	0.9089 ± 0.0414	0.1402
Vanderbilt Lung WCX	0.8558 ± 0.0579	0.9001 ± 0.0453	0.0443
Vanderbilt MALDI	0.8107 ± 0.0658	0.9143 ± 0.0405	0.1036

Table 14: Performance of Feature Decorrelating Methods on SAP-Preprocessed Data

Dataset	Best Univariate	Best Feat. Constructor	Advantage
COPD	0.6233 ± 0.2202	0.6478 ± 0.2175	0.0245
Hepatitis C	0.6836 ± 0.1564	0.7102 ± 0.1398	0.0265
ILD	0.6572 ± 0.1901	0.6627 ± 0.2277	0.0055
Diabetes	0.6953 ± 0.1869	0.7426 ± 0.1571	0.0473
Melanoma I	0.5838 ± 0.1876	0.5955 ± 0.1893	0.0117
Breast Cancer	0.5806 ± 0.1105	0.6067 ± 0.1287	0.0261
Pancreatic Cancer I	0.9023 ± 0.0505	0.9599 ± 0.0238	0.0576
Pancreatic Cancer II	0.8979 ± 0.0511	0.9311 ± 0.0391	0.0333
Prostate Cancer	0.8510 ± 0.0578	0.9691 ± 0.0172	0.1181
Scleroderma	0.6112 ± 0.1314	0.6889 ± 0.1153	0.0777
UPCI Lung Cancer	0.6917 ± 0.0799	0.8276 ± 0.0639	0.1360
Vanderbilt Lung IMAC	0.7985 ± 0.0668	0.9003 ± 0.0429	0.1019
Vanderbilt Lung WCX	0.8976 ± 0.0458	0.9520 ± 0.0245	0.0545
Vanderbilt MALDI	0.8086 ± 0.0628	0.9221 ± 0.0379	0.1134

Table 15: A comparison of predictive model performance (in terms of AUC) when using various kernel learning methods on raw data. An ℓ_1-norm penalized SVM was used as the base predictive model, while the kernel was learned through varying methods. The results are shown as averages over 40 splits of training/testing data along with 95% confidence intervals.

Dataset	Hyperkernel	Metalearning	Prior Knwlg
COPD	0.5025 ± 0.2662	0.5473 ± 0.1986	0.5365 ± 0.1189
Hepatitis C	0.5666 ± 0.2230	0.6299 ± 0.1619	0.5728 ± 0.1410
ILD	0.6550 ± 0.2206	0.5505 ± 0.2090	0.5538 ± 0.2124
Diabetes	0.5557 ± 0.2287	0.4783 ± 0.2037	0.5172 ± 0.1095
Melanoma I	0.5094 ± 0.2055	0.5181 ± 0.1177	0.5045 ± 0.1513
Breast Cancer	0.5121 ± 0.1303	0.5115 ± 0.1298	0.5232 ± 0.0853
Pancreatic Cancer I	0.5442 ± 0.1405	0.7812 ± 0.0632	0.5931 ± 0.0449
Pancreatic Cancer II	0.5739 ± 0.1373	0.7246 ± 0.0840	0.5460 ± 0.0851
Prostate Cancer	0.6243 ± 0.0870	0.9405 ± 0.0226	0.6571 ± 0.0851
Scleroderma	0.5301 ± 0.1342	0.6335 ± 0.0666	0.5673 ± 0.1295
UPCI Lung Cancer	0.5860 ± 0.0947	0.5000 ± 0.0000	0.5780 ± 0.0643
Vanderbilt Lung IMAC	0.7419 ± 0.0777	0.5201 ± 0.0026	0.5907 ± 0.0454
Vanderbilt Lung WCX	0.7764 ± 0.0723	0.5448 ± 0.0108	0.6764 ± 0.0602
Vanderbilt MALDI	0.7140 ± 0.0825	0.5527 ± 0.0046	0.6038 ± 0.0592

Table 16: Performance of Kernel Learning Methods on SAP-preprocessed Data

Dataset	Hyperkernel	Metalearning	Prior Knwlg
COPD	0.5640 ± 0.2170	0.6247 ± 0.1856	0.5651 ± 0.2055
Hepatitis C	0.7608 ± 0.1363	0.6542 ± 0.1905	0.6719 ± 0.1424
ILD	0.5549 ± 0.0566	0.6984 ± 0.1877	0.6380 ± 0.2077
Diabetes	0.6372 ± 0.1893	0.6473 ± 0.2208	0.6008 ± 0.1402
Melanoma I	0.5397 ± 0.2003	0.5387 ± 0.1957	0.5647 ± 0.1587
Breast Cancer	0.5654 ± 0.1227	0.5587 ± 0.1373	0.5608 ± 0.0945
Pancreatic Cancer I	0.6179 ± 0.0906	0.9544 ± 0.0324	0.6783 ± 0.1002
Pancreatic Cancer II	0.5761 ± 0.0856	0.9201 ± 0.0473	0.6079 ± 0.1036
Prostate Cancer	0.7017 ± 0.0634	0.9369 ± 0.0316	0.6913 ± 0.0832
Scleroderma	0.6263 ± 0.1297	0.6625 ± 0.1520	0.6364 ± 0.1203
UPCI Lung Cancer	0.6937 ± 0.0897	0.7595 ± 0.0725	0.5830 ± 0.0799
Vanderbilt Lung IMAC	0.8070 ± 0.0626	0.8684 ± 0.0534	0.6174 ± 0.0793
Vanderbilt Lung WCX	0.8753 ± 0.0496	0.9360 ± 0.0334	0.5877 ± 0.0729
Vanderbilt MALDI	0.8233 ± 0.0582	0.8758 ± 0.0498	0.6765 ± 0.0627

Table 17: Percentages of insignificant ($p >= 0.01$ of Mann-Whitney U-test) comparisons between feature selection methods. Key: CxC (Correlation-aware vs. Correlation-aware, 3 pairs); CxM (Correlation-aware versus Multivariate filter, 6 pairs); CxU (Correlation-aware versus Univariate filter, 6 pairs). The tests are performed on the raw data (first three columns) as well as SAP-preprocessed data (last three columns); RxS (p-value for Kruskal-Wallis test between advantages obtained on all feature selection methods on raw data versus all feature selection method advantages on SAP data). A higher percentage means that methods from the two groups more frequently offer a statistically similar performance advantage

	RAW			SAP			
Dataset	CxC	CxM	CxU	CxC	CxM	CxU	RxS p
COPD	100%	50%	50%	100%	50%	50%	0.699
Hepatitis C	67%	17%	17%	100%	50%	50%	0.000
ILD	100%	50%	50%	100%	50%	50%	0.078
Diabetes	33%	17%	17%	100%	50%	50%	0.000
Melanoma I	100%	33%	50%	100%	50%	50%	0.000
Breast Cancer	100%	50%	33%	100%	50%	50%	0.387
Pancreatic Cancer I	100%	50%	0%	67%	50%	0%	0.000
Pancreatic Cancer II	100%	50%	33%	67%	33%	0%	0.000
Prostate Cancer	33%	50%	50%	33%	33%	0%	0.000
Scleroderma	100%	17%	50%	33%	50%	33%	0.000
UPCI Lung Cancer	33%	17%	0%	33%	17%	17%	0.000
Vanderbilt Lung IMAC	33%	0%	0%	100%	33%	0%	0.000
Vanderbilt Lung WCX	33%	0%	17%	100%	50%	0%	0.000
Vanderbilt MALDI	33%	17%	0%	33%	17%	0%	0.000

.

Table 18: The percentages reflect the proportion of datasets in which the row kernel-learning method significantly outperforms the column kernel-learning method. The Linear SVM with Parallel Decorrelated features is also added for comparison. Significance tests were done with a Mann-Whitney U-test with $\alpha = 0.5$. Interpret the Meta-Learning kernel as just a plain linear kernel.

	Hyperkernel	*Metalearning*	*Pathway*	*Parallel COR*
Hyperkernel	—	7%	35%	14%
Metalearning	57%	—	50%	35%
Pathway	0%	0%	—	7%
Parallel COR	28%	0%	50%	—

Table 19: Comparison of Log-likelihoods of probability models obtained from SVM classifiers on SAP-processed data using different kernels. Columns correspond to: HYP (Hyperkernel), META (Meta-learning), PATH (Pathway Kernel) and the Linear kernel used in Chapter 3. A greater number is better - the best result for each dataset is bolded.

Dataset	HYP	META	PATH	Linear
COPD	-3.2400	-2.4017	-4.6944	**-1.5809**
Hepatitis C	-8.8668	-15.0818	-6.3207	**-5.1058**
ILD	**-3.1548**	-3.3564	-3.4733	-4.8469
Diabetes	**-2.0065**	-2.5536	-4.3701	-4.3251
Melanoma I	**-4.2802**	-4.2343	-10.9350	-7.0104
Breast Cancer	**-18.7682**	-36.8485	-30.8202	-27.8166
Pancreatic Cancer I	**-7.1689**	-13.0232	-8.2671	-37.4862
Pancreatic Cancer II	**-7.1665**	-8.6821	-7.2155	-19.6421
Prostate Cancer	**-8.8602**	-10.4713	-10.7550	-23.9234
Scleroderma	**-5.4823**	-10.0400	-5.5808	-21.3438
UPCI Lung Cancer	-15.7557	-23.3732	**-14.8621**	-29.7677
Vanderbilt Lung IMAC	-21.9110	-29.1278	**-14.7063**	-37.0586
Vanderbilt Lung WCX	-20.1528	-34.2703	**-15.5185**	-43.2553
Vanderbilt MALDI	-14.9691	-23.0833	**-14.1232**	-30.5254

5.0 INTERPRETATION

5.1 BACKGROUND

Following predictive modeling and classification of proteomic profiles, any result must be interpreted to some extent. The mass spectrometry (and in particular, the TOF-MS) data source has two problems which can make interpretation of promising results difficult. The first problem is that the data source is inherently noisy. Although preprocessing alleviates this problem to some extent, it's impossible to account for every source of variation in the equipment and sample. The second problem is that the features in the data are difficult to associate with biological concepts (henceforth referred to interchangeably as *protein identification* or *protein labeling*. This is due to the nature of how the data is generated with MS equipment (and especially complicated with SELDI-TOF-MS technology) (See section 2.3 for a better understanding of why). Even with more complex equipment, labeling features with biological identifiers for proteins is difficult. Although the methodologies for protein labeling can differ according to equipment, as long as one exists, it enables theorizing about the data in different ways. With a large collection of data and the ability to label features, it becomes possible to search for disease-specific and general biomarkers.

To address these problems, two classes of methods are considered for the interpretation of the performance of predictive models. Statistical interpretation methods aim to alleviate concerns about the caveats normally associated with predictive modeling, such as generalization error between experiments and the reproducibility of results. Biological interpretation approaches attempt to link biological relationships to the features or samples used by the predictive model. First, the features should be attached to protein labels, so that any further reasoning about the data and results will carry additional meaning. Then, additional rela-

tionships can be made using prior knowledge and online databases which document protein interactions. Uncertainty in the biological identities of features generally discourages this type of analysis, but as MS profiling technology improves, these interpretation techniques will improve as they accommodate new information. For now, the hypothesis is that the presence of these methods will enable more interesting conclusions about MS protein profile analysis beyond the statistics presented for predictive model performance. These methods should enable the analyst to reveal interesting concepts which have been hidden up until now, due to the challenging nature of the data.

5.1.1 Statistical Interpretation Methods

To some degree the statistical validation of classifier models is taken care of through cross-validation. However we may be interested in additional statistical properties of the data which extend beyond classifier error. Rather than invent separate testing techniques for each statistical property of interest, it would be desirable to use a general framework for testing.

The permutation test [118] is such a framework, which can be used to interpret the strength of any measurable statistical property T of a classifier model or its data. Examples of statistical properties aside from classifier error include differential expression scores of features or inter-class correlation.

The permutation test is a nonparametric approach to hypothesis testing, which makes it useful when the underlying distribution of T is unknown. Figure 21 illustrates the permutation test. A distribution of T under a null hypothesis H_0 is generated by randomly permuting the class membership labels of the data. For each permutation, T is recalculated, which results in an empirical estimation of the distribution of T under H_0. The null hypothesis typically reflects the idea that no special relationship exists between T and the true assignment of class labels to the data. For example, in the case of a model with excellent testing error, we might consider H_0 to represent the hypothesis that the model achieves its testing error through chance, implying that the model's selected features are spuriously correlated with the class label.

Values in this distribution are compared to the value of T under the true class label assignment. If this value outside the distribution by a significant amount, the null hypothesis is rejected. The level of significance can be determined, for example, by p-value. The parameter α describes the level of confidence at which the null hypothesis was rejected.

5.1.2 Biological Interpretation Methods

The statistical validation of predictive models is useful for empirically determining the applicability of a predictive model. Determining the reasons why these models work beyond the mathematical properties is a natural next question. Biological information is somehow encoded into the proteomic profiles, and if a predictive model is effective at using this information to discriminate between cases and controls, it would be interesting to see how the model achieves its decisions. Moreover, if a predictive model experiences errors, we might be curious whether any biological relationship to these errors are present.

In the data available for this thesis, biological information is limited to the molecular weights of molecules measured by the mass spectrometer. More recent technology exists which can supplement MS profile data by annotating features with peptide identifications. With or without this advantage, it can be uncertain as to which proteins are being represented in the MS profiles. Although the protein identities can not be known for sure, we can make educated guesses about the identities of features. In the future, if MS proteomic profiling improves to the point where all features are accurately identified, this identification step can be skipped. However, biological interpretation techniques remain unchanged.

The Pathway Kernel from Section 4.3.6 was intended as an additional way to interpret protein profiling models. The features selected by an SVM using the Pathway Kernel would correspond to separate biological systems of interactions influencing the overall proteomic profile. If the model fits the data well, we can then look at the components of the individual processes to speculate why these components affect the outcome of a disease. Unfortunately, converting the gene interaction pathways to locations in the protein profiles proved difficult with the SELDI and MALDI ToF-MS data sources.

To deal with uncertainty of the identity of features, discriminative features and patterns

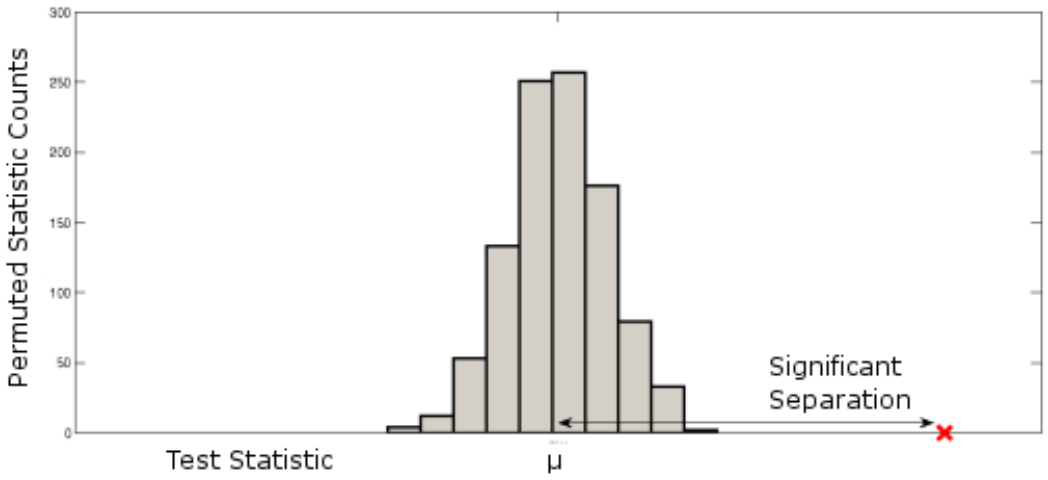

Figure 21: Example of the permutation test. The empirical distribution of the test statistic T (shaded histogram) is estimated by permuting class labels of data. If the test statistic obtained under the true class labels (red cross) falls outside of this distribution by a significant amount, the value of the test statistic is less likely to occur by chance.

are often termed "surrogate" biomarkers. These surrogate biomarkers function in discriminating diseased and healthy samples despite their abstract nature. It is possible that other abstractions can exist, which could be more easily related to biological concepts and possibly more effective for classification. This thesis intends to investigate ways of abstracting the information in proteomic profiles in order to determine whether interesting biological relationships can be discovered. This is done primarily through *latent variable models*, a set of statistical models which relate output variables to a smaller set of "hidden" variables. These latent variables are not observed directly in the data, but are modeled so that their settings produce the data through independent processes. In an appropriately designed model, these processes could be interpreted as separate biological systems of interactions influencing the overall proteomic profile. The latent variables might indicate the condition of each process, whether it is in a normal or dysfunctional state. If the model fits the data well, we can then look at the components of the individual processes to speculate why these components affect the outcome of a disease. There may be multiple ways to organize the features of a proteomic profile into biological processes. One option is to use correlations learned from the feature selection process. Another option is to label the features with their protein identities. If this organization process is good, we might expect a better fit of the model to the data.

5.2 RELATED WORK

5.2.1 Supporting Generalization Error Results

The permutation.based approach compares the error achieved on the true data to errors on randomly labeled data. It tries to show that the result for the true data is different from results on the random data, and thus it is unlikely the consequence of a random process. Note that the permutation-based method is different and thus complementary to standard hypothesis testing methods that try to determine confidence intervals on estimates of the target statistics. We also note that one may apply standard hypothesis testing methods to check if the target statistic for our classification model is statistically significantly dif-

ferent from either the fully random, trivial or any other classification model. However, the permutation framework always looks at the combination of the data label generation and classification processes and thus establishes the difference in between the performance on the true and random data. Classification error is a composite evaluation metric. Other types of performance measures for which confidence intervals have been studied so far include significance of SN at a fixed SP [119], AUC [120] and the ROC curve itself [121]. Here we briefly explain these options. Which performance measure to assess may vary according to strategy. Bootstrap-estimated or analytically determined confidence intervals around SN at a specified SP [119] requires that a desired SP be known, and this depends on its intent; for example a screening test should have very high SP to avoid resulting in too many false positives when applied to a population. Even here, however, "very high" and "too many" are rather context dependent, and should not be considered in a silo by ignoring existing or other proposed diagnostic tests. Acceptable FP values depend to a degree on the SP of existing practices, and to an extent on the prevalence of the disease. Any screen can be considered to change the prevalence of disease in the "potential patient" population, and therefore follow-up with panels of minimally invasive markers, or multivariate studies of numerous risk factors (demographic, familial, vaccination, smoking history), and longterm monitoring, might make such screening worthwhile. High-throughput proteomics highlights the need for dynamic clinical diagnostics.

The various approaches suggested by Linnet were extended and revised with a suggestion by [122] to adopt the bootstrap confidence interval method [123]. A working paper by [124] explores related approaches. One strategy is to perform bootstrapping [123] and calculate a 1-α confidence interval around a measure of interest. Bootstrapping is a subsampling scheme in which N data sets are created by subsampling the features of the original data set, with replacement. Each of the N data sets is analyzed. Confidence intervals around some measure of interest (T) can be calculated or consensus information can be gathered; in either case, variability in an estimate T is used a measure of robustness of T. Various implementations of the bootstrap are available; the least biased appears to be bias-corrected accelerated version [123].

A second strategy is to calculate confidence intervals around the AUC measure. Boot-

strapping [123] is sometimes used to estimate AUC confidence intervals. Relying on confidence in the AUC can be problematic because it reports on the entire ROC, and, in practice, only part of the ROC is considered relevant for a particular application (e.g., high SP required by screening tests. A literature on assessing the significance of partial ROC curves has been developed [125, 126]; a recent study [127] compared the features and performance of eight programs for ROC analysis. A third strategy is to calculate bootstrap confidence bands around the ROC curve itself [121]. Under this approach, bootstrapping is explored and bands are created using any of a variety of "sweeping" methods that explore the ROC curve in one (SN) or two (SN and 1-SP) dimensions.

5.2.2 Addressing concerns of Reproducibility

Earlier MS proteomic profiling studies stimulated significant enthusiasm [7], discussion [8], and controversy [128] in the general scientific community and among proteomics researchers. Potential confounding and bias in study design and analysis in initial studies [27], were recognized early on and have been addressed in subsequent research (See [129]) for an overview). Issues related to confounding and bias in study design and data analysis can be approached using appropriate principles of clinical epidemiology and laboratory research, together with careful experimental design and methods for data preprocessing and analysis.

Predictive modeling relies on the detection of potential biomarkers which may explain disease through previously understudied combinations of reproducible molecular measurements. The reproducibility of these surrogate biomarker patterns often comes into question; a pattern is not guaranteed to be replicated exactly within the same or other data generation session, or at a different laboratory. This results from the intrinsic variation introduced into the data by factors including, but not limited to, the biological nature of the samples and limitations of the MS technology.

Typical proteomic profiling studies attempted to minimize the effect of this variation by generating data in a single session. Classification results on these data sets were encouraging, but dealt only with the data variation within a single session. These data sets were produced in the 'ideal' environment where only a single instrument in a single laboratory produces

all of the available data at the same time. As a result, potential factors of inter-session and inter-site biases were ignored. Skepticism arose as to whether spectra generated during multiple sessions separated by variable intervals of time, or by a different laboratory, will be useful for predictive modeling applications. Promising inter-site reproducibility results were reported by [130, 131]. Inter-session reproducibility, however, remains a relatively open area of research.

5.2.3 Protein-Feature Association

Mass spectrometry (MS) proteomic profiling enables a parallel measurement of hundreds of proteins present in a variety of biospecimens. This technology has shown many very promising results in differentiating diseased and control samples in a variety of studies [132–135]. The studies typically report classification statistics achieved by a predictive model, such as accuracy, sensitivity, specificity or area under the ROC curve. As a supplement, features constituting the discriminatory patterns are given as a list of mass-to-charge (m/z) ratios. Unfortunately, these features were only rarely matched to proteins, which weakened the overall results and spurred some controversy surrounding the interpretation of these analyses [8]). Knowing the identities of ion species critical for the group differences could lead to understanding the biological relevance of the result and would make the results more acceptable.

An early approach to feature labeling in TOF-MS data relied solely on information about mass of the protein species, more specifically the m/z ratio of peaks in MS signals [136]. However, many different molecules can appear at the same location, and choosing among them often results in an incorrect or nonsensical identification. Improved mass spectrometry techniques dependent on sample fractionation enabled better separation of molecules along the m/z axis by separating them along another dimension, usually elution time. These MS profiling approaches typically use *Tandem Mass Spectrometry* as opposed to TOF-MS to recover amino acid compositions of molecules. Despite the additional information, feature labeling is imperfect and faces several challenges; since this thesis has only TOF-MS data available, those methods for Tandem MS/MS are beyond the scope of this thesis. For a

review of these methods and their limitations, see [137, 138].

5.2.4 Incorporating Prior Information

In an effort to incorporate more prior knowledge into the learning process, [139] created a data integration framework using Bayesian networks. The authors proposed that different data sets would necessarily require different networks to accurately represent biological processes, since different data sources will have different intentions on which processes to measure. In this work, separate data sources were linked together by various Bayesian network structures, although the performance of structured vs naive networks did not seem to differ much. Each data source was modeled as being conditioned on by the functional relationship of a gene pair. The correlations between expression measurements (as in microarray data) were discretized and converted into true/false values. The network then predicts the type of functional relationship responsible for the data. It is largely up to the learning algorithm to utilize the "prior knowledge", which comes in the form of additional data types such as wet-lab conformational assays or genetic associations.

Prior knowledge is also concentrated in publicly available sources, such as the Kyoto Encyclopedia of Genes and Genomes (KEGG, [140]). A paper by [141] sought to use KEGG as a way to validate prediction of gene pair interactions. The authors use a Bayesian scoring metric which uses KEGG information as a benchmark. Interacting gene pairs are found to be linked in various types of data. The scoring metric assigns higher scores to data which display a greater frequency of links as found in the benchmark pathway information. Each separate data source provides a score for a gene pair. The authors found it difficult to create a Bayesian approach to estimate the contribution of each data source, since the data sources have varying relative independence. As a solution, scores for a gene pair are reweighted by rank-ordering the scores and dividing by a parameter chosen to optimize accuracy and coverage of the integrated score on the benchmark. Accuracy was calculated by comparing to gold-standard small-scale gene interaction assays. The work demonstrated that even weak evidence from multiple sources can be combined for strong overall evidence for gene linkage prediction. Furthermore, clustering genes in the resulting network resulted in highly coherent

clusters in terms of function. The authors suggest the approach can be extended to human genes and will improve as the amount of collected data increases. Functional analysis of the clusters must still be done by hand.

Another effort to integrate prior knowledge was undertaken in [142]. Annotated gene expression datasets in the Gene Expression Omnibus (GEO, [143]) were mapped to concepts in the Unified Medical Language System (UMLS, [144]). Datasets were partitioned by concept into those mapped to that concept, and those not mapped. Genes in these datasets are then tested for significant differences between the groups. The score of statistical significance is subjected to a random permutation test by permuting the assignments of datasets to concepts. As a result, genes are annotated with phenotypic concepts from the UMLS, which creates a phenome-genome network derived only from expression data. Validation was done two ways: by assuring a strict statistical significance threshold and by obtaining the same gene-concept links in with homologous genes in other organisms. The latter method can be restrictive since many conditions are not studied on homologous organisms. However, thresholding by statistical significance produced many manually verified relations.

5.3 METHODS

5.3.1 Permutation-Achieved Classification Error (PACE)

The objective of optimizing a classification score itself is largely uncontrolled in most genomic and proteomic high-throughput analysis studies. Researchers do not, for example, typically attempt to determine and therefore do not report the statistical significance of the sensitivity of a test, in spite of the existence of a number of approaches for performing such assessments. Here we introduce a permutation method for assessing significance on the achieved classification error (ACE) of a constructed prediction model.

Permutation test methods work by comparing the statistic of interest with the distribution of the statistic obtained under the null (random) condition. Our priority in predictive models is to critically evaluate the observed classification performance. In terms of hy-

pothesis testing the null hypothesis we want to reject is: *The performance statistic of the predictive model on the true data is consistent with the performance of the model on the data with randomly assigned class labels.*

Permutation-Achieved Classification Error (PACE, [145] is a statistical validation framework for determining whether the effect of a predictive model is obtained by chance. For this application, our statistic of interest is the achieved classification error (ACE) of the predictive model. PACE evaluates a predictive model M by estimating the distribution of ACE T_{ACE} under M. The procedure is represented in algorithm 4 . The number of permutations is arbitrary; we use 1000 by default. Each time M is evaluated on a dataset, the ACE is estimated through cross-validation over multiple train/test splits to reduce the effect of subsampling bias. If the true ACE of M falls on the tail of T beyond the threshold specified by α, the null hypothesis is rejected. This indicates that the predictive ability of M is less likely to be by chance. This property gives additional assurance that M is valid model.

input : Predictive model M, significance threshold α, number of permutations B

output: Success of hypothesis test

Compute ACE T of model M on original data

for $b \leftarrow 1$ **to** B **do**

    Permute the group labels in the data

    Compute the ACE T_b for model M on the modified data

end

Calculate the p–value of T with respect to the distribution defined by permutations b as: $p = N_{T_b \leq T}/B$, where $N_{T_b \leq T}$ is the number of permutations for which the test statistic T_b is better than T under the true labeling

if $p < \alpha$ **then**

    Reject null hypothesis, validation succeeds on M at confidence level α

    else Accept null hypothesis, validation fails on M at confidence level α

end

Algorithm 4: PACE Algorithm

5.3.2 Measures of Reproducibility

PACE evaluates a model holistically. The permutation-based framework can be specialized to the feature level to measure the stability of information found in the data. From an evaluation standpoint, these analyses are particularly useful when dealing with data produced throughout multiple sessions. An ideal model should locate generalizable, robust features to use for the discrimination process. The same feature set across datasets for the same (or possibly for similar) diseases should remain strong if they are valid. To this end, a set of reproducibility metrics is presented to measure the reproducibility of information across data [39, 146]. These techniques also attempt to quantify how much inter-session variability plays a role in the performance of predictive modeling. Specifically, these techniques demonstrate 1) the presence of intersession noise, 2) the similarity of profiles from different sessions, 3) the reproducibility of aggregate patterns and 4) the generalizability of a model's performance when using inter-session data. The individual techniques are described in more detail below. A feature set which displays strong reproducibility characteristics across multiple data sessions is likely to benefit an accompanying predictive model, which can be further evaluated with the PACE framework described above.

- It is possible to examine the differences in signals from the same sample across multiple sessions. A signal difference score is defined to measure the discrepancies between signals from the same sample. The hypothesis tested is whether the signal difference score for profiles from the same sample is significantly better than profiles from other samples. This would indicate that identical samples processed in multiple sessions experience more similarity to themselves than to other samples in the session, supporting the usage of profiles from multiple session for analysis purposes.
- The second proposed test asks whether discriminative information is affected by intersession noise. This issue is analyzed on the feature signal and multivariate levels, using differential expression and classifier accuracy metrics, respectively. The effect of intersession noise on these statistics is determined by comparing them on single-session and randomized multi-session data sets.
- The final proposed tests evaluate the predictive performance of multivariate models on

future sessions. One test evaluates by how much the performance of classification models deteriorates on future sessions with respect to their 'ideal' single-session performance. The other test asks if performance of a multivariate model on future sessions can be improved if the model is trained on mixed-session data. The hypothesis is that if inter-session variability exists, it can be learned through multi-session data, potentially leading to accuracy gains over models trained on single sessions.

The specific methods to test these objectives are outlined below.

5.3.2.1 Reproducibility of profile signals No two MS profiles are exactly the same. Profiles may differ due to instrument noise, differences in sample preparation procedures, etc. Differences in profiles for the same sample are visible even if two profile replicates are generated in the same session, and even if they are placed on the same chip. The intra-session profile variation is well known and existing methods are robust enough to cope with it. The differences in profiles for the same sample across multiple data generation session are much less understood. The differences in the sample preparation at different times, instrument settings may effect the resulting profiles and contribute to possible inter-session biases and variability.

Figure 22 displays four MS profiles from the same sample that were generated in four different sessions. Although the shape of the profile may look similar, differences in relative intensities of peaks are apparent. Are these differences significant? Are these variations too strong to overcome so that the profiles from the same sample are useless and easy to confuse with profiles generated for other samples? To answer these questions we need to define a similarity (or distance) metric that helps us assess the differences among profiles. We would like MS profiles from the same sample to differ less across sessions than profiles from other samples. To achieve this goal we measure the similarity among a set $\mathbf{S}$ of k spectra using the average Euclidean distance d_E between all pairs of spectra:

$$d_E(\mathbf{S}) = \frac{1}{\frac{k(k-1)}{2}} \sum_{\forall 1 \le p < q \le k} \sqrt{\sum_{i=1}^{d} (p_i - q_i)^2} \tag{5.1}$$

where p and q represent a pair of spectra from the subset of k replicate spectra generated

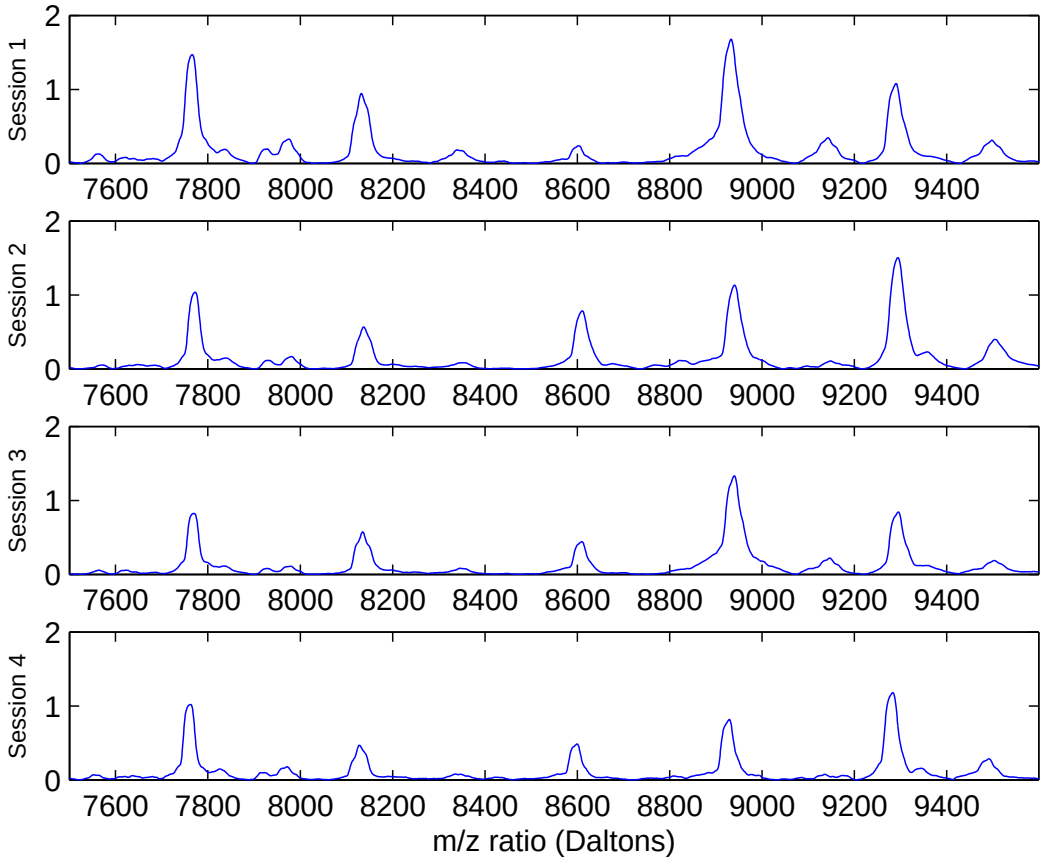

Figure 22: MS profiles for a single sample across 4 different sessions. Changes are apparent in relative intensities of peaks.

from the same sample source. Intuitively, the signal difference score measures the sum of areas between all possible superimposed pairs of k spectra; smaller values indicate better similarity.

We used the above signal difference metric first, to evaluate the similarity of spectral measurements from the same sample across multiple sessions and then, to determine that the differences from random collections of spectra from other patients are very different and thus profiles that originate from the same sample are hard to confuse with other profiles.

A random permutation test [118] was used to test the differences and their significance.

We first estimated a distribution of signal difference scores for randomly grouped spectra. Random groupings were generated by shuffling the sample identities assigned to spectra in each session. The signal difference score was recalculated for each random profile grouping, and the process was repeated 1000 times to estimate the distribution of signal difference scores for randomly grouped spectra. Next, the signal difference score for profiles belonging to the correct samples was calculated. If the score is statistically significantly different with respect to the estimated distribution, we have greater confidence that signals from the same sample are similar to each other beyond random effects. This increases our confidence that profiles generated from multiple sessions are potentially useful for analysis.

5.3.2.2 Reproducibility of Discriminatory Signals Evaluating profile similarity across sessions helps assure us of the basic consistency (reproducibility) of spectra with respect to samples their represent. However, the differences in profiles across multiple sessions are apparent (see Figure 22). This leads to a concern that information potentially useful for disease detection purposes may be lost or at least significantly compromised if data from multiple sessions were used in the analysis. To assess the effect of the potential information loss we compare data mixed from multiple sessions to data generated from individual sessions and their discriminatory power.

The information that helps us discriminate between healthy (case) and diseased (control) profiles can be drawn from a single feature (peak) of the profile, or from a combination of multiple features. We measure the quality of discriminative information for a single feature (peak) by its *differential expression score*. The score quantifies the difference observed in a profile feature between case and control groups. In this paper, we use the Fisher-like score, computed as $|\frac{\mu^{(+)}-\mu^{(-)}}{\sigma^{(+)}+\sigma^{(-)}}|$, where μ and σ represent the sample mean and variance of the feature, respectively. The signs $(+)$ and $(-)$ denote case and control samples, respectively. We note that differential expression can be measured using many other criteria [25] which would work just as well.

Testing peaks' discriminatory information loss: To determine if the differential expression information is lost across multiple sessions we assumed that feature's differential expression follows a distribution across sessions. The distribution can be empirically estimated by ran-

domly choosing each sample's spectrum from its replicate set. We generate 1000 randomized datasets and calculate a feature's differential expression score under each dataset to recover its empirical distribution.

If the profiles generated in a single session retain more discriminatory information, we expect their differential scores to be higher on average than the mixed-session distribution. We can test this by comparing the differences in between the mean score for the mixed-session distribution and the score for the single-session. We have four different sessions per sample and multiple spectra peaks. We use 100 peak regions, evaluate their single-session scores and compare their peak scores to the distributions generated for mixed session data. This process generates a distribution of score differences. If the single session spectra are 'better' we expect them to be differ on average from 0. This difference and its significance can be assessed using standard one-sided hypothesis testing framework.

Testing multivariate information loss: The reproducibility of differential information in individual features may be indicative of the reproducibility of discriminative information given by combinations of these features. However, this is not guaranteed. Are the feature combinations differently represented across sessions? If we mix data from different sessions, what is the effect on the discriminative pattern and the resulting predictive model? To answer these questions, we examine if the performance of a predictive model deteriorates on data mixed from several sessions, as opposed to data from the same data-generation session.

Performance of a predictive model is typically measured using accuracy (percentage of correct predictions), sensitivity and specificity, or area under the ROC curve statistics. In this work, we evaluate predictive models using their test set accuracy. Similarly, there are many classification models one may try to learn multivariate patterns. We use the linear Support Vector Machine model [72,91] to learn the relationship between diagnostic features and state of disease. This method has been used previously in many cancer studies [25,38,147] and is particularly favored for its 'regularized' feature selection.

To assess the reproducibility of multivariate classification patterns across sessions, we generate 1000 random datasets such that each patient (sample) receives one of the profiles from its replicate set. Our goal is to analyze differences in the performance of classifiers on: (a) models trained and tested on profiles from multiple sessions, versus (b) models trained

and tested on profiles from the same session. To measure test accuracies of models we first decide which patients (samples) will be used for training and testing purposes. Forty-six patients (samples) are split via random subsampling [148] so that 30% of the samples are in the test set. The spectra obtained for the remaining samples are used to train the predictive model. The split is always the same for both single-session and multi-session models. Test set accuracies of 1000 random models define a distribution of accuracy scores for multi-session data. This distribution can be compared to accuracy results for models trained and tested on four single sessions. However, four single sessions entries are not sufficient to make any strong conclusion. In addition, there is a chance a single train and test split may be biased. To eliminate these problems, we repeat the analysis for multiple (30) train–test splits. This lets us calculate 120 accuracy scores for single session models (30 per one session) and compare them to respective accuracy-score distributions defined by 1000 multi-session datasets. To assess the benefit or loss of multi-session data, we compare the mean of their accuracy-score distribution to accuracies achieved by single-session models. To assess the global benefit or loss, we average the results over four different sessions.

5.3.2.3 Effect of multi-session data on generalization performance In the 'ideal' analytical setup for proteomic profiling studies, a predictive model is trained and evaluated on data from the same session. It experiences only within-session noise and does not account for potential inter-session noise, should it be re-used for future prediction of profiles. However, in the practical setting of clinical screening, new samples may be processed on-the-fly, each at a different time and therefore experiencing unanticipated amounts of inter-session variability. Concerns about this inter-session reproducibility is related primarily to concerns over generalizability of predictive models that are extracted from past data sessions to profiles obtained in the future. We will analyze this aspect of the problem by learning predictive models that are tested on profiles from one target (test) session and trained on the profiles from the remaining three (training) sessions and by comparing them to the 'ideal' model trained and tested on the profiles from the same session.

We perform this analysis as follows. A target (test) session is chosen from the available four sessions. The remaining three sessions are used to train a (future) predictive model.

Next, samples are divided via the random subsampling approach to training and testing samples, such that 30% of the samples are in the test sample set. The remaining samples are represented in the training sample set. Next, we generate 1000 multi-session training datasets by assigning each patient in the training sample set a profile from one of its training sessions and learn the models for each dataset. The models are tested on the test session samples and their accuracies define the distribution of (future) test accuracy scores for mixed-session data. The mean of the distribution is then compared to the accuracy achieved by the model trained on the same session as the test session. To provide additional assurance we repeat everything using 30 different train-test sample splits and average the results. This will let us compare the average future performance of mixed-session models to the 'ideal' model for one test session. The global performance can be assessed by averaging the results for four test sessions.

In our first comparison of (a) models trained on profiles from the three training sessions, versus (b) an 'ideal' model trained on profiles from the same session as the testing set, we expect the 'ideal' models to outperform the multi-session-trained models. Inter-session variability is not present in the ideal model and is therefore expected to cause a loss of performance. Our second aim is to compare models from group (a) versus (c) models trained on profiles from a single session other than the target session. The objective is to determine if predictive models trained on multi-session data can learn to adapt to inter-session noise and hence improve their performance when compared to models learned on single sessions.

We repeat the setup in the previous experiment to estimate the distribution of accuracy scores for the 1000 models trained on multi-session data. Accuracy scores are also obtained from models trained on one of the three single sessions. The difference between the mean accuracy of the multi-session models and single-session models are kept for a total of 3 differences. This process is repeated 40 times for each of the four target sessions. We repeat the hypothesis test to determine if the mean of these differences differs significantly from 0. In the case where multi-session models have the same generalization performance as single-session models, the mean of this distribution should not differ significantly from 0.

5.3.3 Peak Labeling

A new computational approach for assigning protein labels to peaks in MS spectra is described below. This method builds upon the information about the mass of a protein's sequence and prior knowledge of the expected abundance of that protein in the biospecimen. The method starts from a list of protein species that may show up in the profile, along with their masses and expected relative abundances in the biospecimen. The method then attempts to match the proteins to peaks observed in the profile while fitting the recorded mass and intensity characteristics of the peaks. Since the measurements in the MS profile are noisy in both the mass and intensity dimension, it is not immediately clear what peaks correspond to what protein species. To model this uncertainty we rely on a probabilistic model that represents the distribution of measured masses and measured intensities for a protein in the specimen.

The distinguishing characteristic of this peak-labeling approach is the inclusion of the peak intensity aspect in the model. A previous approach to peak labeling relied solely on information about mass of the protein species, more specifically the m/z ratio of peaks in MS signals [136]. However, many different molecules can appear at the same location, and choosing among them often results in an incorrect or nonsensical identification. Our improvement is to incorporate information about the expected relative abundances of proteins in the sample. Therefore, the intensity aspect attempts to match labels to peaks based on the expected relative abundance of the label's peptide. For a reliable labeling, a label must fit the criteria of a good match to the location and intensity aspect of the peak signal simultaneously.

We believe our current development effort is particularly important for high-throughput whole-sample proteomic profiling and its post-interpretive analysis. The term "whole-sample" refers to methodologies which do not significantly deplete or fractionate samples prior to their MS analysis.

5.3.3.1 Probabilistic Model A mass spectrometry profile consists of a series of peak measurements. Our method assigns protein labels to these peaks. Each label uniquely

identifies an ion species (charged molecule of a protein or peptide) believed to be responsible for the peak. A probabilistic model and associated probabilistic score takes into account expectations about where ion species may produce peaks in the MS spectrum, and how intense they may be. The probabilistic score is optimized using a dynamic programming algorithm.

MS profiles occur as a series of measurements in two dimensions. The mass-to-charge (m/z) ratio reflects the mass of the measured ion, and its corresponding intensity reflects the number of ions measured at that m/z. Measurements in a close neighborhood may aggregate to peaks. These peaks may represent an ion species in the sample, or noise due to various sources. Typical MS profile analyses focus only on peaks selected from the profiles. Hence our analysis focuses only on peak-based profiles.

We define an **observed MS profile** S by a set of peak measurements
$S = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$

where x_i denotes the m/z location of the i^{th} peak measurement, and y_i its corresponding intensity. The peak's m/z ratio reflects the mass of the measured ion species creating that peak. The peak's intensity reflects the number of ions measured at a m/z ratio and measures (indirectly) the concentration of the species.

Our goal is to annotate peaks in the MS profiles with ion species labels. We approach this task by associating peak measurements to labels. Let $L = \{l_1, \ldots, l_d\}$ denote a set of labels for ion species we believe may be present in the profile. Each label l can be associated with either: a peak measurement $m = (x, y)$ appearing at m/z location x and intensity y, or the $m = null$ value denoting a situation in which l does not get a peak. We refer to the collection of these label-to-peak assignments as a **label-induced profile** and denote it as $S_L = \{m_1, m_2, \cdots, m_d\}$. Figure 23 illustrates the idea, note that the label-induced profile is only a subset of peaks of the original profile. Since MS profiles are noisy, one set of species labels may lead to more than just one profile S_L. We therefore define a probability distribution $P(S_L)$ that reflects how probable each of these profiles is.

To model the relation between a label-induced profile and an observed profile, we assume the observed profile S is generated from the labeled profile S_L by adding additional peaks to S_L. These peaks may correspond to species not in L or to noisy measurements. Figure

23 illustrates the relation between the two profiles. We assume the probability $P(S|S_L)$ is uniform for every profile S consistent with S_L and 0 for every inconsistent profile. The two profiles S and S_L are said to be consistent if every peak in S_L is matched in terms of the location and intensity.

5.3.3.2 Peak-labeling as an optimization problem Our goal is to find the labeling that gives the best label-to-spectrum fit. In other words, we want to identify the best possible label-induced profile $\mathbf{S}_L^*$ as supported by the observed profile S. In terms of a probabilistic model, the problem can be cast naturally as the problem of finding a label-induced profile S_L with the highest posterior probability:

$$\mathbf{S}_L^* = \underset{S_L}{\operatorname{argmax}}\, P(S_L|S) = \underset{S_L}{\operatorname{argmax}}\, \frac{P(S|S_L)P(S_L)}{P(S)} \tag{5.2}$$

Since the denominator is common for all assignments, it is sufficient to optimize its numerator:

$$\mathbf{S}_L^* = \underset{S_L}{\operatorname{argmax}}\, P(S|S_L)P(S_L) \tag{5.3}$$

The first term represents the conditional probability of observing a spectrum S given the ion species in S_L. The second term is the prior probability of S_L. For the sake of simplicity, we assume that $P(S|S_L)$ is approximately equal for all consistent matches of S and S_L. In such a case, the optimization reduces to the problem of finding the profile S_L that is consistent with S and that maximizes $P(S_L)$:

$$\mathbf{S}_L^* = \underset{\mathbf{S}_L}{\operatorname{argmax}}\, P(S_L) \text{ such that } S_L \text{ is consistent with } S \tag{5.4}$$

A label-induced profile $S_L = \{m_1, m_2, \cdots m_d\}$ assigns peaks to labels in L. If no peak is assigned to a label, a special 'null' value is used. Consequently, the model of the prior distribution for S_L, $P(S_L)$, can be defined in terms of an auxiliary label-induced-indicator profile $\hat{S}_L = \{u_1, u_2, \cdots u_d\}$ that determines whether or not each peak is assigned a label, but does not give specifics of peak measurements. Correspondingly, the values of a random

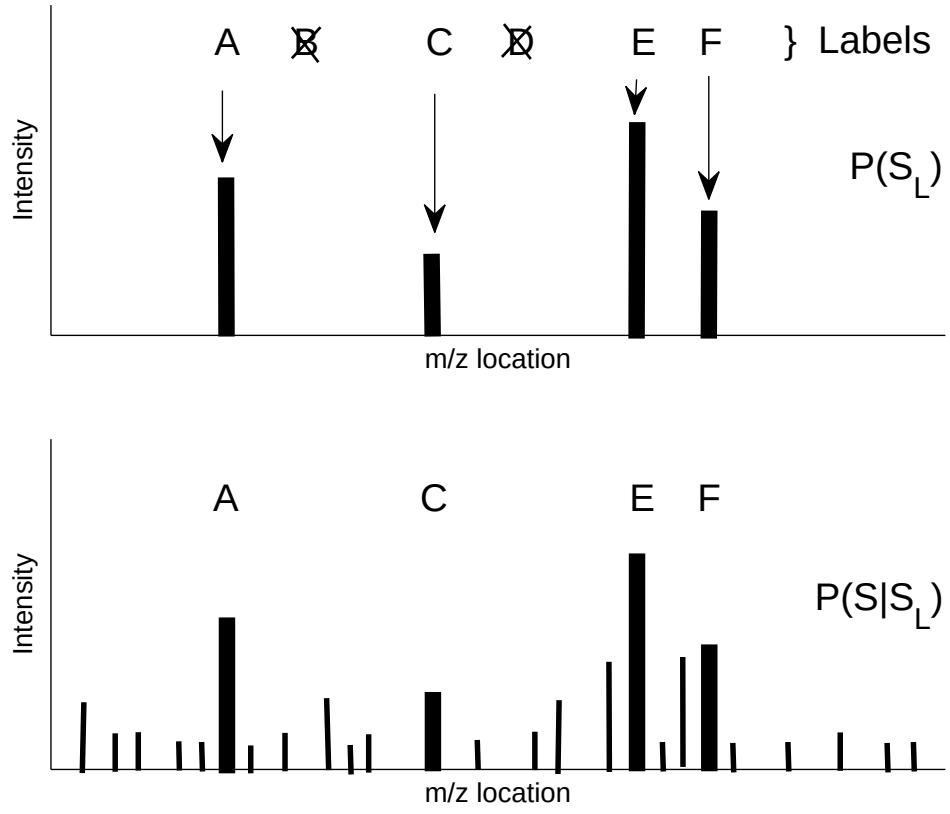

Figure 23: An interpretation of the probabilistic model. Top: the label-induced profile relates ion species labels with peak measurements (thick lines) in the profile. Some labels are not assigned to any peak (B and D are therefore struck out). Bottom: the observed mass spectrum S is given by the label-induced profile S_L plus additional noise and unlabeled measurements (thin lines). ©2010 IEEE.

variable u_i are *true* or *false* for peak and 'null' outcomes respectively. Since the label-induced-indicator profile is fully determined by S_L and does not introduce any new information[1] we can rewrite $P(S_L)$ as:

$$P(S_L) = P(S_L, \hat{S}_L) = P(S_L|\hat{S}_L)P(\hat{S}_L),$$

where $P(\hat{S}_L)$ is the probability of the labels in L being assigned to peaks or not and $P(S_L|\hat{S}_L)$ defines the probability distribution of peaks with certain location and intensity characteristics for all non-null assignments.

We define $P(\hat{S}_L)$ by assuming that the probability of an individual label receiving a peak is independent of other peaks:

$$P(\hat{S}_L) = \prod_{i=1}^{d} p(u_i) \tag{5.5}$$

where $p(u_i)$ is the probability of seeing or not seeing a peak for an i^{th} label. This distribution is defined in terms of a single parameter $p_{i,0}$ that denotes the probability of the i^{th} ion species failing to appear as a peak in the label-induced spectrum.

To define $P(S_L|\hat{S}_L)$ we propose and analyze two models that differ on the peak information incorporated into the model. The first model defines the distribution of label-induced peaks only in terms of their location information and ignores peak intensity measurements. The second model combines both the peak location and the peak intensity information to define the distribution.

5.3.3.3 Peak-location aspect First, let us consider a model for $P(S_L|\hat{S}_L)$ that incorporates only information about location of ion species in L into the model. In this case, S_L can be redefined as vector of the x components of $\{m_1, \ldots, m_d\}$. We denote this projection as S_L^x.

Briefly, if an ion species is present we expect it to be observed in the vicinity located around its m/z ratio. In other words, the closer the peak is to its expected m/z ratio, the better its peak-label fit should be. We define the probability of a label-induced spectrum as:

[1]The only reason to introduce the auxiliary profile is to simplify the definition of the distribution of S_L that uses random variables with a hybrid (null or peak measurement) outcomes.

$$P(S_L^x|\hat{S}_L) = \prod_{i=1}^{d} P(x_i|\hat{S}_L) \tag{5.6}$$

where $P(x_i|\hat{S}_L)$ is defined as:

- $P(x_i|\hat{S}_L) \sim N(x_i|\mu_i, \sigma_i)$ if the peak for the i^{th} label is among the observed peaks, that is, $u_i = true$
- $P(x_i = null|\hat{S}_L) = 1$ if the peak for the i^{th} label is not among observed peaks, that is, $u_i = false$
- $P(x_i|\hat{S}_L) = 0$ for all other cases.

The parameter μ_i is the expected time-of-flight (TOF) position[2] of the ion species corresponding to the ith label. The standard deviation σ_i is set to reflect the amount of mass inaccuracy expected for the i^{th} ion species.

Combining the model with the model for $P(\hat{S}_L)$ in equation 5.5, the probability of a label-induced profile S_L is:

$$P(S_L) = P(S_L^x|\hat{S}_L)P(\hat{S}_L) = \prod_{i=1}^{d} P(x_i|u_i)P(u_i) \tag{5.7}$$

where

$$P(x_i|u_i)P(u_i) \sim \begin{array}{l} N(x_i|\mu_i, \sigma_i)(1 - p_{i,0}) \text{ if } u_i = true \\ p_{i,0} \text{ if } u_i = false. \end{array} \tag{5.8}$$

The advantage of the above model is that it decomposes along individual labels in L. However, the decomposition comes with one limitation. It permits an out–of–order assignment of labels to peak locations, that is, there is a non-zero probability that two labels with expected m/z values $\mu_i > \mu_j$ will switch their order in the label-induced profile. We do not expect this situation to occur, although close paralogs may violate this order. To fix the problem we keep the above decomposable model, but always enforce the m/z locations of peaks to be order-consistent with the expected masses of their labels. If paralogs with very close masses are present in the label database, their order will depend on the expected masses of their labels.

[2]The TOF is approximately equal to the square root of the ion's mass. Peak locations are converted to TOF during calculations.

The other advantage of the model is that parameters μ_i, σ_i and $p_{i,0}$ restrict the range of possible peak assignments to the i^{th} label. Briefly, the i^{th} label can be assigned a peak with location x only if

$$p_{i,0} < (1 - p_{i,0}) \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i}\right)$$

If this condition is not satisfied, the null assignment is preferred to a peak at x. This is important since only peaks close to the expected m/z value should be considered as potential matches for i^{th} label.

5.3.3.4 Location-based peak-labeling algorithm The main advantage of our peak-location model is that the optimization of $\mathbf{S}_L^*$ can be carried out by a dynamic programming procedure. The decomposability of the score and the fact that possible peak matches are restricted by the parameters μ_i, σ_i and $p_{i,0}$ lets us assign a peak to a label by only considering peaks and labels in the close vicinity of its expected mass μ_i. This is a favorable environment for dynamic programming.

The proposed dynamic programming procedure works as follows. First, the set of labels are sorted according to their expected mass. Then, each label is visited in ascending order of its expected mass. The label can be associated with one of the peaks in the feasible range of the represented ion species (defined by its $p_{i,0}$ and σ_i values), or with nothing (the 'null' value). For each of these assignments, we calculate and keep its partial score and optimal partial assignments to all previously scanned labels. If no peak is assigned to label i, $p_{i,0}$ is used. The process continues till all feasible peak-label pairs are examined and their scores are assessed. The optimal peak-to-label pairing sequence is the output of the method.

5.3.3.5 Peak intensity aspect The optimization of the label-to-peak assignments above was performed using the knowledge of the protein sequence and its mass. In addition, one can obtain the information about the species expected concentrations in different types of samples. The abundances of proteins and peptides in relation to each other can inform on the proper assignments of labels to peak measurements in the profile.

We wish to incorporate information about protein and peptide relative abundance into our peak-labeling procedure. We assume that measurements along either the location or

Sort labels in ascending order based on expected mass.

for $k \leftarrow 1$ **to** d **do**

Compute a portion of $\mathbf{S}_L^*$ by aligning labels $l_a, \cdots, l_b \in L$ and peaks $m_p, \cdots, m_q \in S$ which lie in a feasible window around label l_k by maximizing $P(S_L)$ through dynamic programming:

for $i = 0$ **to** $b - 1$ **do**

$score(i, 0) \Leftarrow p_{k,0} \cdot i$

end

for $j = 0$ **to** $q - 1$ **do**

$score(0, j) \Leftarrow p_{k,0} \cdot j$

end

for $i = 1$ **to** b **do**

for $j = 1$ **to** q **do**

$P(S_L(m_k = x_j)) \Leftarrow score(i-1, j-1) \cdot P(x_j|u_k)P(u_k = true)$

$P(S_L(m_k = null)) \Leftarrow score(i-1, j) \cdot p_{k,0}$

$P(S_L(m_k = null)) \Leftarrow score(i, j-1) \cdot p_{k,0}$

$score(i, j) \Leftarrow max(P(S_L(m_k = x_j)), P(S_L(m_k = null)))$

end

end

end

return $\mathbf{S}_L^*$

Algorithm 5: Location-based peak labeling

intensity axis happens through independent mechanisms, and that the noise models for these mechanisms appear to be unrelated. This allows us to decompose the probability of observing the spectrum under a given assignment as:

$$P(S_L) = P(S_L|\hat{S}_L)P(\hat{S}_L) = \; = P(S_L^x|\hat{S}_L)P(S_L^y|\hat{S}_L)P(\hat{S}_L) \tag{5.9}$$

We reuse the definition of the peak location aspect (from equation 5.6) for the first factor. The second factor in the probability defines the peak intensity aspect. Our assumption is that relative abundance of the ion species in the sample is reflected by their measured relative intensities. Thus, the peak intensity aspect models the probability of peak intensities in S_L for all peak-to-label assignments. We define the intensity aspect of S_L as the vector of y-components of $\{m_1, \cdots, m_d\}$ and denote it as S_L^y. To model the relative abundance of species that appear as peaks, we use the Dirichlet distribution:

$$P(S_L^y|\hat{S}_L) \sim Dir(\tilde{\mathbf{y}}|\alpha_1, \ldots, \alpha_d) \sim \frac{\Gamma(\Sigma_{i=1}^d \alpha_i)}{\prod_{i=1}^d \Gamma(\alpha_i)} \prod_{i=1}^d \tilde{y}_i^{\alpha_i - 1} \tag{5.10}$$

where $\tilde{\mathbf{y}} = (\tilde{y}_1, \ldots \tilde{y}_d)$, such that $\tilde{y}_i = \frac{y_i}{\sum_{j=1}^d y_j}$ is the intensity value for the measurement assigned to the i^{th} label, renormalized according to intensities of other peak candidates in L. The parameters of the distribution $\{\alpha_1, \ldots, \alpha_d\}$ reflect the expected concentrations of d proteins and their variance.

The primary purpose of incorporating both intensity and abundance information is to resolve possible misassignment of peaks to labels. Consider two peaks that are in the region of the MS profile in which we expect a higher abundance protein to occur. Figure 24 displays an assignment of peaks to the labels for protein A and B. While A appears as the only peak in the region around its expected position μ_A, there are many peaks around the expected position μ_B of protein B. Because of spectral misalignment, the peak caused by protein B may not be the closest peak to its expected location μ_B. However, we can correct for such a problem by considering the relative abundance information. For example, if we expect protein B to be about one fifth more abundant as protein A, even if the peak with a higher intensity is further from the expected mass, the label may be reassigned to account for a proper fit to expected concentration levels.

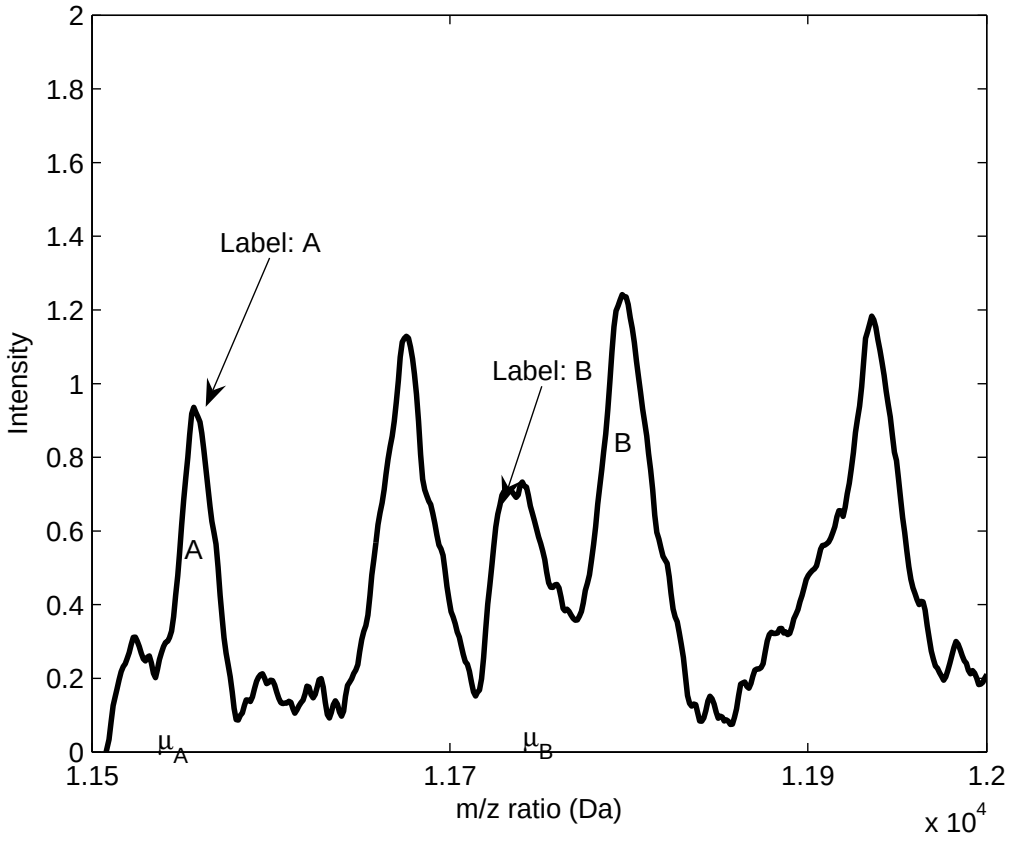

Figure 24: An illustration of misassignment which is correctable by our procedure. Expected positions of proteins are marked as μ, where the true identification of a peak is marked with a capital letter. Abundance information can help to reach the correct assignment if we know that protein B is one fifth more abundant as protein A. ©2010 IEEE.

Note that the above Dirichlet distribution model factorizes over the individual intensities if the normalizing constant $\sum_{j=1}^{d} y_j$ is known. In such a case, we would be able to couple factors from the Dirichlet with factors in the peak position model (see previous section) and optimize the two together locally within the dynamic programming scheme. Unfortunately, the normalizing constant depends on peak-to-label assignments made over the complete profile. This violates the assumption underlying the dynamic programming calculations, that the probabilistic score can be calculated using only information from a partial assignment.

5.3.3.6 Enhanced peak-labeling algorithm To overcome this problem, we approximate the components used in the calculation of the normalization constant through a greedy heuristic procedure. The procedure works as follows. First, the labels for ion species (proteins) in L we expect to see in the profile are sorted according to their abundance. Following this order, the label is assigned to the largest unlabeled peak in the region defined by its expected mass μ_i, σ_i and $p_{i,0}$. The fact that the largest peak is labeled corresponds to the situation in which an ion species with the highest concentration claims the peak. The greedy procedure does not optimize for combinations of assignments, but the hope is that it gives a good initial assignment.

The heuristic assignment gives an initial set of peak-to-label assignments and it is used to estimate the Dirichlet portion of the probabilistic score. Combining the location and intensity aspects, the probabilistic score becomes:

$$P(S_L^x|\hat{S}_L)P(S_L^y|\hat{S}_L)P(\hat{S}_L) \sim \ \sim \Gamma(\sum_{i=1}^{d} \alpha_i) \prod_{i=1}^{d} \frac{\tilde{y}_i^{\alpha_i - 1} P(x|\hat{S}_L)}{\Gamma(\alpha_i)} \tag{5.11}$$

Given the initial heuristic peak-to-label assignment, the optimization of m^* can be performed using the same dynamic programming approach as outlined above. For each label, all peak-to-label matches inside the feasible region of the current label are optimized and their partial score is calculated. To obtain the Dirichlet portion of the score for each peak-to-label match, the intensity components needed for its calculation are obtained from: (1) the partial optimal assignment of peaks to labels already computed by the algorithm, or (2) the initial heuristic peak-to-label assignment. Since some of the labels may be assigned the 'null' value,

the intensity of the null-assigned peak is set to equal the level of noise observed within its feasible region.

Because of the initial heuristic intensity assignment, the dynamic programming procedure may not lead to the optimal assignment in the first pass. To correct the heuristic assignment, we run the algorithm multiple times so that a more refined initial intensity assignment is used in the next step. The algorithm is run until no change in the global peak-to-label assignment is observed.

5.3.3.7 Applying the Model The above model is a general framework for labeling peaks. This accommodates for changes in the technology producing mass spectra, which can differ greatly. In our experiments, we test our peak-labeling method on mass spectra from both a simulated matrix-assisted and a surface-enhanced laser desorption/ionization TOF spectrometer. Thus, we set the parameters of our method to appropriate settings for time-of-flight mass spectrometry.

Measurements which contend for a label are obtained by selecting "peaks" from the MS signals. While there are many ways to measure and select a "peak", we take the following approach [25] : We begin with a set of available data $D = S_1, S_2, ..., S_k$, consisting of k spectrum signals. In the first step, a "mean" profile is created such that $S_{mean} = (x_1, (\sum_k y_{1k})/k), (x_2, (\sum_k y_{2k})/k), ..., (x_n, (\sum_k y_{nk})/k)$. This achieves a smoothing effect which ensures only highly expressed and reproducible peaks become prominent. Next, the local maxima of the mean profile are selected. Finally, a signal-to-noise filter is applied to remove those peaks that lie close to the baseline. This results in the selection of peak locations that appear to be highly reproducible.

Peak intensity correction Care should be taken when assigning intensities to peak measurements. The intensities reported should reflect the number of measured molecules for that peak to the best degree possible. If the intensities for all measurements are misrepresented unintentionally, the assumed Dirichlet model may not fit well, and few peaks will be labeled. The necessities for corrections depends on the accuracy with which intensities are reported by the MS technology. For example, our matrix-assisted laser desorption/ionization (MALDI) TOF MS simulator produces wider, shorter peaks for larger mass molecules. The

Sort labels in ascending order based on expected sample abundance.

Perform the initial heuristic alignment:

for $i = 1 \dots d$ **do**

  Select all unassigned measurements in S feasibly located around μ_i.

  Assign label l_i to the most intense of these peak measurements.

end

repeat

  for $k = 1$ **to** d **do**

    Compute a portion of $\mathbf{S}_L^*$ *by aligning labels* $l_a, \cdots, l_b \in L$ *and peaks* $m_p, \cdots, m_q \in S$ *which lie in a feasible window around label* l_k *by maximizing* $P(S_L)$ *through dynamic programming:*

    for $i = 0$ **to** $b - 1$ **do**

      $score(i, 0) \Leftarrow p_{k,0} \cdot i$

    end

    for $j = 0$ **to** $q - 1$ **do**

      $score(0, j) \Leftarrow p_{k,0} \cdot j$

    end

    for $i = 1$ **to** b **do**

      for $j = 1$ **to** q **do**

        $P(S_L(m_k = m_j)) \Leftarrow score(i-1, j-1) \cdot \Gamma(\sum_{i=1}^{d} \alpha_i) \frac{\tilde{y}_j^{\alpha_i - 1} P(x_j | \hat{S}_L)}{\Gamma(\alpha_i)}$

        $P(S_L(m_k = null)) \Leftarrow score(i-1, j) \cdot p_{k,0}$

        $P(S_L(m_k = null)) \Leftarrow score(i, j-1) \cdot p_{k,0}$

        $score(i, j) \Leftarrow max(P(S_L(m_k = m_j)), P(S_L(m_k = null)))$

      end

    end

  end

until *no change in* $\mathbf{S}_L^*$

return $\mathbf{S}_L^*$

Algorithm 6: Peak-labeling with abundance information

areas under these peaks may be quite large, but the critical points of these peaks are much lower.

Since we use the height of the critical point of a peak as its intensity, a correction is necessitated. We do this by rescaling the value y_j at the peak's critical point. We assume that peak measurement distributions follow a Gaussian distribution [1]. The standard deviation of this distribution, σ_j, is used as an estimate of the spread of a molecule's peak at m/z position x_j. We rescale the intensity y_j of the peak by a factor of y_j/c_j ,where c_j is the value at the critical point of a normal distribution with mean 0 and standard deviation σ_j. This results in a better estimate of intensities, and therefore a better fit to the empirical abundance model used by the method.

Our procedure relies on a database of information about proteins which are expected or may be found in a particular sample medium. While this data often comes from specialty laboratories, efforts are underway to characterize popular sample types such as plasma [149], urine [150] and saliva [151]. Using the available and appropriate references for the sample medium in question, we select the most highly abundant proteins and collect information about them. The result is a set of candidate labels for protein and peptide signatures in the spectra.

We begin building the set of labels by connecting to databases such as ExPAsy[3], UniProt[4], and the NCBI Entrez Protein database[5]. The amino acid sequences and molecular features regarding the abundant peptide candidates found in the literature are retrieved. As an example, figure 25 shows the relevant information obtained from Swiss-Prot about the Serum Amyloid A (SAA) precursor protein, an abundant acute-phase reactant in human serum. We use this information to compute the expected mass of a molecule of SAA. This is done by adding the average isotopic masses of the amino acids in the given sequence, in addition to the average isotopic mass of a single water molecule. A label with this expected mass is added to the label database.

The location of a protein or peptide's signature along the x-axis depends on its TOF through the mass spectrometer. This is approximately equal to the square root of the

[3]http://www.expasy.org/
[4]http://www.uniprot.org/
[5]http://www.ncbi.nlm.nih.gov/Entrez/

Serum Amyloid A Precursor (SAA_HUMAN)

MKLLTGLVFC SLVLGVSSRS FFSFLGEAFD GARDMWRAYS DMREANYIGS DKYFHARGNY
DAAKRGPGGV WAAEAISDAR ENIQRFFGHG AEDSLADQAA NEWGRSGKDP NHFRPAGLPE KY

key	from	to	length	
SIGNAL	1	18	18	Signal peptide
MOD_RES	101	101	1	N4,N4-dimethylasparagine (Probable).

Figure 25: Summary of information taken from the Swiss-Prot database on the Serum Amyloid A Precursor protein (Accession SAA_HUMAN). The amino acid sequence, as well as 2 potential post-translational modifications and their positions of occurrence, are indicated.

expected mass of the corresponding molecule. In the case of multiply-charged ions of the given molecule, the m/z position of the singly-charged molecule is divided by the number of (positive) charges expected.

In addition, information is given about post-translational modifications, which can change the amino acid structure, and subsequently the mass, of the molecule. These include phenomena such as glycosylation and phosphorylation. Potential, probable, and confirmed sites for post-translational modifications are documented in the most popular protein and peptide databases. In our example, SAA can undergo signal peptide cleavage, as well as a dimethylation at residue 101. For each possible modification, the average weight of the modifying molecule is added or subtracted as necessary to that of the original, unmodified molecule. In the case of multiply-charged ions, the expected mass is divided by the amount of charges. A new label with the modified mass is added to the label database. We avoid considering all combinations of modifications by incorporating only a single modification per label. Figure 26 displays the resulting label list after processing SAA and its modifications.

5.3.3.8 Protein abundance It is unrealistic to believe that every ion species with a specific molecular weight will be detected by the MS technology in use at its expected m/z location. Moreover, multiple proteins or peptides can share a similar expected m/z position. We augment our labeling procedure by including information about the relative abundance of proteins or peptides in the sample. This is expected to be reflected in the spectra by the

Ion Species	Expected Mass/Charge Ratio (Daltons)
Serum Amyloid A	13532.02104
SP-cleaved Serum Amyloid A	11682.70064
Dimethylated, Serum Amyloid A	13560.07484
Serum Amyloid A +2H	6766.01052

Figure 26: List of candidate protein labels resulting from information collected about Serum Amyloid A. The list includes the original precursor ion, post-signal-peptide cleavage, post-dimethylation and post-double charge forms with their estimated mass/charge ratios.

intensity of the signal along the y-axis.

In an 'ideal' measurement process, the relative abundances of molecules in the biospecimen should be evident in the intensities of peaks they produce in the spectrum [152]. For example, consider a pair of peaks, with the first peak being twice as intense as the second. In the best case, the first peak would be more likely generated by a molecule which was twice as abundant in the biospecimen as the second peak's molecule. However, many factors indicate that abundances of molecules in the biospecimen are not truly reflected by intensities in resulting mass spectra [153]. Instead, the intensity of a protein or peptide's signature along the y-axis depends on the amount of molecules successfully ionized and detected by the spectrometer.

Accurately quantifying the amount of each molecule measured in proteomic studies is a complex and ongoing field of research. The emerging methods for protein quantitation use stable isotope labeling [154] to differentiate between the proteins in two sets of sample pools. One sample pool is marked with a heavier molecular tag and the difference is noted in a shift of peaks along the x-axis equal to the weight of the tag. More recently, computational approaches have been developed for determining the relationship between the quantity of molecules in a sample and the amount of signal generated by their presence [153, 155]. These methods generate prior expectations of how well a protein or peptide can be detected based on its composition. Although these methods do not perfectly quantify the amounts of proteins measured, their computational nature negates the cost of chemical reagents. A key notion of these methods is that component peptides of proteins each have different de-

tectabilities. This is important to know in standard protein identification, which fragments proteins into their component peptides. In our alternative approach of peak labeling, proteins are generally left intact. Thus their detectability is likely an aggregate function of its component peptides. The determination of this function is not straightforward, but our model incorporates this function as a parameter α. The abundance variation parameter α is intended to represent the relationship between the content of a given ion species in a sample, and the mass spectrometer's observation of that ion species in terms of relative intensity. Accurately expressing this relationship is an ongoing process. In light of the this uncertainty, we make a number of limiting assumptions which facilitate the operation of our procedure.

We assume all molecules have an equal chance to ionize, and that their relative abundances should be ideally reflected through any measurement technique at any level. Proteins and peptides may exhibit ionization difficulty, which results in a weak or failed attempt to quantify molecules that are present in the mixture. Previous work has addressed some of the causes for this difficulty by determining the effect of a peptide's physical and chemical properties on its chance to be observed [153, 155]. Incorporating this information into a finer estimate for α is beyond the scope of this paper.

Second, we assume that proteins or peptides which experience modifications occur in the same abundance as their original, unmodified form. We make this assumption because there is little publicly available empirical evidence indicating how often modifications occur under various experimental settings. An extension of this work, in which the occurrence of modifications is estimated based on function of the protein or peptide, is under way.

Third, we assume that the abundances of proteins or peptides should be ideally reflected through any measurement technique. Any measurement technique has the possibility of favoring a particular class of molecules. We make this assumption to avoid generating our expectations of abundance based on biased measurements. We wish to avoid particularities about specific measurement technologies, each of which can be sensitive towards a particular class of molecules.

Following these assumptions, it is only necessary to collect the relative abundances of proteins from literature. For example, [149] gives a list of expected concentrations of these proteins in human plasma. We use this information to obtain the expected relative abun-

dance of the species in the sample and incorporate them to the probabilistic model. In the absence of concentration bounds, we can estimate the concentration of a molecule as equal to the weight of the sample divided by the mass of the molecule. This gives an upper bound on the abundance of the molecule.

Figures 31 and 32 show that the values parameters take on greatly influence the performance of the labeling method. Spectra produced by different machinery or experimental design will likely require different parameters for a labeling to be successful. Whenever possible, values for parameters should be chosen carefully with prior knowledge. In the absence of prior knowledge, a parameter search can be done by measuring the success of the labeling procedure on spectra from a known protein mixture, while varying the values of parameters.

Each peak position in the analyzed spectra may be assigned its own location variation parameter, σ_i. This parameter should be set to reflect the reported mass accuracy at the m/z position of the i^{th} peak for the MS instrument used to produce the data. For the Ciphergen PBSIIc MS instrument, this value was 0.2% of the expected mass. For the simulated data, the mass accuracy was unknown, but constant across all peaks. Thus, all σ_i parameters had the same value, and multiple evaluations were performed while varying this value.

Each peak position in the analyzed spectra may also be assigned its own intensity variation parameter, α_i. This parameter should be set to reflect the expected variation in the intensity of the i^{th} peak for the MS instrument used to produce the data. In both the simulated and real data, there was no way to accurately measure the variation in intensity, as the amount of ions presented for measurement is never absolutely known. In our experiments, we set α_i to be equal to the expected relative concentration of the i^{th} molecule in the sample. These parameters are scaled so that the standard deviation of each variable is 10% of their expected values. Although we did not vary these parameters during our experiments, this can be done in an attempt to achieve a better labeling.

Each peak position in the analyzed spectra may also be assigned its own reliability parameter, $p_{i,0}$. This parameter should be set to reflect the expectation that the i^{th} molecule generates a detectable peak in the MS signal. In experiments where affinity surfaces selectively bind molecules for analysis, molecules which should not bind should have their $p_{i,0}$ parameter set to reflect the probability with which they will remain in the analysis. In both

our simulated and real experiments, we set these parameters to a single value p_0. In the simulated experiments, this parameter was varied slightly over a range of values. With the SELDI-TOF data, p_0 was set to 0.05 to reflect a 5% chance that molecules in the peak database do not register as peaks in the MS signal. This value was determined by assuming a 1% chance of a peak not appearing on either tail of the distribution, and 3% to reflect imperfections in the peak detection procedure.

5.4 EXPERIMENTS AND RESULTS

The following sections deploy the methods proposed above, which aid the interpretation of MS proteomic data analysis results in varying ways. The amount of work into each method necessarily varies. PACE is a general technique which could be applied to any dataset. The next section demonstrates its application towards validating the strength of the classifiers learned on the 3 Vanderbilt Lung SPORE datasets in Chapters 3 and 4. However, for the peak labeling method, very little data exists which serves as a gold standard. Additionally, few datasets exist where the samples are generated in multiple sessions, so it is difficult to do a thorough validation of the reproducibility measures. Nevertheless, some results on the multisession UPCI Lung Cancer dataset are presented below, as well as peak-labeling using SELDI technology to identify calibrant proteins in human serum.

5.4.1 Case Study: Application of the PACE Technique to Lung SPORE datasets

Up until this point, we have performed many experiments resulting in good predictive models on the Vanderbilt Lung SPORE datasets. We can use the PACE technique to determine whether or not these results occur simply by chance. Instead of ACE, let our permutation test statistic instead be the average AUC of a predictive model on the SAP-processed Vanderbilt Lung SPORE datasets. The predictive model will be the same as that reported in Table 3, a ℓ_1-norm regularized linear SVM. The labels are permuted 1000 times for every one of the 40 train/test splits.

Figure 27 shows the estimated permutation distributions for the Vanderbilt IMAC, WCX and MALDI Lung SPORE datasets, from left to right, respectively. The AUC achieved by our predictive model under the true labeling is marked with a small red cross. Note that the cross lies well outside the distribution, visually confirming significance. Statistically, this result is significant at the $\alpha = 0.001$ threshold, meaning it would be very difficult to accept the null hypothesis that the results achieved in Chapter 3 were obtained by chance alone. This gives us assurance that our predictive model is valid.

5.4.2 Case Study: Application of Reproducibility Measures to Lung Cancer Serum Spectra

5.4.2.1 Signal reproducibility We first examined whether proteomic spectra are reproducible across multiple sessions. We used the random regrouping test described in Section 5.3.2.1 to evaluate whether the signals from the same sample were more similar than signals from randomly chosen sample sets. Since we expect to find differences between case and control samples, this score was evaluated separately on respective subgroups of case and control spectra.

The histogram in figure 28 (left) indicates the average signal difference score for the 21 cancer patients across all 4 sessions. A distribution of 1000 averages of 21 signal difference scores for randomly selected quadruplets of spectra is plotted as a reference. The score for the replicate spectra falls outside of the score distribution for randomly grouped spectra. A similar phenomenon occurs with the control samples. Furthermore, we can assess the reproducibility of signal difference over a small region of the profile. The right panel of figure 28 displays the distribution of signal difference scores for the peak region at 8228 Da. There is less difference in the peak among profiles from the same sample than from profiles randomly assigned to a sample. There is a statistically significant difference between the signal difference scores obtained from true and random replicates, at both the global and local (peak) signal level. This assures us that profiles from the same sample do not exhibit so much difference that they can be easily confused with profiles from a different sample. This encouraging result emphasizes the reproducibility of proteomic profiles at the signal level.

5.4.2.2 Reproducibility of discriminative features We use the randomization framework from section 5.3.2.2 to determine whether differential expression scores obtained from mixed session data differ on average from the differential expression mined from single session datasets. These differences may assess the benefit or loss due to mixed-session analysis.

Figure 29 (left) displays the empirical distribution of differential expression scores for multi-session data of one prominent peak in the spectra. The distribution was obtained from 1000 random datasets such that each patient was randomly assigned a profile from one of the four sessions. The four marks indicate the differential expression scores obtained for profiles in four individual sessions.

We next determined the significance of these differences. To determine the amount of noise experienced over a range of features, we similarly analyzed the top 100 differentially expressed peak regions in the profiles. The mean was calculated for every feature's differential expression score distribution, as well as the score of the feature in the four single sessions. These four scores were subtracted from the mean and kept for each feature, resulting in a distribution of 400 differences. Figure 29 (middle) displays this distribution. If single session scores were biased (that is, better scores are produced by the single session analysis) we would expect to see the mean of this distribution to differ significantly from 0. In other words, we would expect to reject (at some significance level) the null hypothesis: the mean of differences is ≥ 0. Indeed, the mean of the distribution of differences was -0.0351, giving a p-value of 5.588×10^{-8} for the one-sided t-test, which leads to the rejection of the null hypothesis. Hence the amount of differential expression in single sessions appears to be better on average than in mixed-sessions. This shows that inter-session variability affects the measured differential information.

We expect this negative result to affect the performance (accuracy) of predictive models trained on multi-session data. The question is how big the effect really is. Earlier research studies considered it most ideal to learn from and evaluate their predictive models on data from a single session. We therefore compare the differences in accuracy between models trained on multi-session data versus models trained on single-session data.

Following the methods in section 5.3.2.2, we analyzed the accuracy of multi-session models versus single-session predictive models. Figure 29 (right) displays the distribution of

differences between mean accuracies of multi-session and single-session predictive models. If better accuracies are achieved by predictive models for single-session data, we would expect to see the mean of this distribution to be below 0. Indeed, the mean of the distribution of differences was -0.0267 which once again indicates a loss that can be explained by additional inter-session variability. To confirm the difference we used a repeated resampling experiment proposed by [156], estimating the 95% confidence interval around the mean of differences to be -0.0267 ± 0.0001. This experiment confirmed that this difference is indeed significant.

On average, there is about a 2.7% drop in accuracy when using multi-session data, demonstrating a relatively small (average) loss of reproducibility of multivariate discriminative patterns across multiple sessions. One should understand that this is an average assessment; the performance of an individual classifier may vary from session to session and also depends on how profiles from multiple sessions are mixed.

5.4.2.3 Generalization Performance Finally, we want to determine the effect the multi-session training has on predictive models which must generalize well to future, unseen profiles and sessions. The previous result demonstrates that intersession noise exists, but does not seem to greatly affect the performance of predictive models on average. However, the analysis used each session and did not try to assess the performance on future sessions. We use the methods in section 5.3.2.3 to analyze whether training predictive models on multi-session data generalizes well to profiles in future sessions and compare the performance of these models to 'ideal' predictive models trained and tested on single session data.

Figure 30 (left) displays a distribution of accuracy differences between the average of 1000 predictive models built from random multi-session training data and models trained on data that came from the session on which the model was tested. The mean of the distribution is -0.0231 which quantifies an overall average generalization accuracy loss one may expect to see by training the model on the mixed session data as opposed to the accuracy of the 'ideal' model. We analyzed the difference using an additional resampling test [156] to compute the 95% confidence interval of the mean. The result of the mean falling within -0.0286 ± 0.0001 confirmed the difference is statistically significantly different. However, in terms of absolute numbers the accuracy loss with respect to the ideal model is not bad.

In a practical setting such as clinical screening, the training data will certainly not come from the same session as the testing session. This eliminates the possibility of having an 'ideal' predictive model. We repeated the previous experiment by examining the differences between the multi-session models and models trained on profiles from a single session other than the target session. The difference from the previous experiment is that the single-session models lose the advantage of the 'ideal' environment. Inter-session noise must now be accommodated by both the multi-session and single-session-trained models.

Figure 30 (right) displays a distribution of accuracy differences between the average of 1000 predictive models trained on multi-session data and models trained on the remaining single sessions. The mean and 95% confidence interval of this distribution falls above 0 ($= 0.0289 \pm 0.0001$), indicating a benefit of training on multi-session data. The confidence interval is again computed using the repeated resampling test [156], which confirmed the difference to be statistically significantly different. This result illustrates how training on multi-session data can allow the model to adapt to inter-session noise. The better a predictive model can adapt to inter-session noise, the more reproducible the performance will be on future data.

5.4.3 Case Study: Application of Peak Labeling

Experiments with our peak-labeling method were conducted in two phases. First, we tested the method on data simulated from a virtual MALDI-TOF mass spectrometer [1]. Second, we have made preliminary validation of the procedure on real biological data for human sera. Information about 94 high-abundance proteins (their expected mass and abundance) in serum was collected from online protein databases and from the literature [149, 157]. The resulting collection was used as the label database throughout our experiments.

5.4.3.1 Phase 1: Labeling simulated data A set of 100 simulated spectra was generated with 16 controlled spiked-in peptides. The relative concentrations of these peptides were chosen arbitrarily and retained as information to be used by the identification procedure. Our task is to label peaks in the spectra correctly (true positive), while avoiding labeling

peaks which may appear as a result of noise (false positive). While Receiver Operator Characteristic (ROC) curves can be used to measure this tradeoff, the skew in number of true positives (16) versus number of potential false positives (> 300) can lead to a misrepresentation of performance. Instead, we evaluate our peak labeling method using precision-recall (PR) curves [158], which are used in Information Retrieval tasks where a skew exists in the numbers of true positives and potential false positives. In our task, precision refers to the fraction of label-assigned peaks which are matched to the correct label, while recall refers to the fraction of the 16 labels which were correctly assigned to a peak. Each of the 100 spectra were labeled, and performance statistics were averaged over the dataset.

We have tested two versions of our peak-labeling method: a baseline version that relies only on the expected mass of the species and our improved version that combines the knowledge of the expected mass together with their abundance information. The objective was to show that the inclusion of the abundance information improves the identification accuracy of the procedure.

We evaluated both our baseline method and the abundance-enhanced method over a broad range of detection conditions. These were controlled by varying the location variation parameter, σ, and the reliability parameter, p_0. In the baseline method, we observed a dramatic effect of varying p_0, which uses only peak location information. A higher value of p_0 heavily discourages a label if the location of a peak is far from the expected mass of a molecule. This results in a conservative labeling beyond $p_0 \approx 0.5$. The probability of peaks not occurring begins to outweigh all but the closest location-based matches. Thus, few if any peak-to-label matches are made beyond this point, resulting in recall and precision close to 0. As the location variation parameter σ grows, there is a slight gain in recall at a cost of precision. By providing a larger window for consideration of peak labels, the method can freely choose labels which may have been further away from their expected mass locations. However, if a close location match has already been found, increasing the window size does not help our method choose the correct label. Thus, the improvement in performance is very minimal when varying σ.

We sought improvements in precision and recall from the abundance-enhanced method. For our experiments, we set the relative abundance variation parameters to reflect the con-

centration used as input to the simulator. Peptides not involved in the process were set to have an abundance attributed to "noise". This was calculated as 1/100 of the smallest spiked-in peptide abundance. The remaining parameters σ and p_0 were varied as before. In comparison to the previous case, which used only location information, the abundance-enhanced method showed a large improvement in precision. Figure 31 compares the variation in precision and recall for both the baseline and abundance-enhanced methods when varying p_0 and fixing $\sigma = 0.05$.

The importance of p_0 in the relative abundance-enhanced method is downplayed due to the need to fit labels to appropriately sized peaks. Only a handful of labels will ever match a peak by both location and relative abundance. A correct match is either hit-or-miss, and only a value of $p_0 = 1$ causes the recall and precision to drop to 0. The result is relatively stable behavior of precision and recall as p_0 is varied. As seen in the previous case, varying σ results in an improvement of recall in exchange for lost precision. However, the precision obtained at most parameter settings are much better than under any parameter configuration examined using the previous method. This results in much better parameter configurations than can be achieved without including abundance information. The abundance-enhanced method can therefore outperform the method using only peak information.

To quantify the improvement of the method, we use the F-measure [159], the harmonic mean of precision and recall. The maximum F-measure obtained by the method using only peak information was 0.5226, versus 0.6667 when including relative abundance information. As an additional performance metric, the area under the method's PR curves (AUC) can be measured [158]. Figure 32 compares the PR curves for both versions of the labeling method. The PR curve for the peak-location method is completely dominated by the PR curve for the abundance-enhanced method. The AUC for the peak location method is 0.5782, while the abundance-enhanced method achieves an AUC of 0.7387. These results show the contribution of relative abundance information greatly improves the method.

5.4.3.2 Phase 2: Labeling spiked-in human serum

We next applied the abundance-augmented procedure to labeling of whole-sample human serum profiles with and without an added protein calibration mixture. All mass spectra were produced using a Ciphergen PBS

IIc SELDI-TOF mass spectrometer (Ciphergen Biosystems, Inc). Figure 33 displays average spectra of the calibration mixture (top), human serum sample (center), and combination of the "spiked-in" calibrant and serum sample (bottom). The calibrant contains an equimolar concentration of horse proteins equine cytochrome C and equine myoglobin. We estimate that serum proteins occur in 250-1000 fold excess of the calibrant. The peaks from the calibrant are absent in the serum profile due to their non-human nature. They are, however, visible in the calibrant/serum mix.

Our objective was to correctly label peaks of equine cytochrome C and equine myoglobin in the human serum-calibrant mixture, while avoiding these labels when the calibrant is not present in the serum. The label database of 94 highly abundant serum proteins was augmented with information about equine cytochrome C (GenBank accession P00004) and equine myoglobin (GenBank accession P68082). Parameters of the procedure were set as follows. The location variation parameter σ was set according to the reported mass accuracy of the Ciphergen PBS IIc instrumentation (0.2% of the molecular mass [27]). The relative abundance variation parameters $\alpha_1 \ldots \alpha_d$ used for the abundance component were set using the expected concentrations of the proteins represented in the label database, such that the expected value is equal to the expected concentration and standard deviation of each variable is 10% of its expected value. The reliability parameter p_0 was set to 0.05 to reflect a 5% chance that a molecule does not create a peak.

The peak-labeling procedure was first applied to the "spiked-in" spectra. Figure 34 displays the successful labeling of equine myoglobin, as well as the distinction between similar compounds from different organisms. The assignment of equine myoglobin is made possible through information about its abundance and location, which properly distinguishes it from other labels. Figure 35 displays the proper identification of equine cytochrome c among other regional peaks. In this mixture of peaks, one is detected at the expected location of equine cytochrome C. Since the peaks in the area have similar abundance, the procedure uses locational information to influence the final labeling. The label displays that the cytochrome C molecule (11702 Da) exhibits an added acetyl group (42 Da) and heme group (616 Da), bringing the total weight to 12360 Da.

In the second experiment, we applied the procedure to spectra containing only human

serum. Knowing that the calibrant is not present, we could *a priori* exclude the mixture components from the set of protein labels. However, to test the robustness of the procedure, we assumed the calibrant components are present at the same concentrations as in the spiked-in spectra. Figures 36 and 37 display the result of the labeling procedure on the serum-only profile. The procedure does not label any peak as equine myoglobin or equine cytochrome C, even though the opportunities exist; a peak is detected in the vicinity of equine myoglobin's expected mass. However, the probability of assignment to the "null" value outweighs the probability of assigning it to a protein label. In particular, the intensity observed is too low to fit the expected concentration well enough to outweigh assigning the peak to the "null" value. In the case of cytochrome C, no peak is detected in the vicinity of the target. Surrounding peaks which fall into the feasible region are assigned to the "null" value, due to a lack of labels which can fit these peaks.

It is certain that these peaks are correctly identified. Few peaks are expected to appear in the calibrant serum. Due to their nonhuman nature, they appear at places which do not seem reproducible in the serum-only analysis. This is clearly observed through addition of the calibrant to whole serum. Finally, the noticeable appearance of the peaks in the "spiked-in" serum lead to the possibility of a highly-abundant peptide. Although there are competing proteins for the spiked-in peaks besides equine cytochrome C and myoglobin, our probabilistic model is able to match them correctly. These results are promising and support further exploration.

In both experiments, we were aware of which proteins needed to be labeled, and whether the labeling was correct. The selection of parameters varies the performance of the method greatly, and in this case we are able to choose the parameters which yield the best performance. In new data, where the true peak identities are not known, it is unclear how to choose the parameters to achieve the best result. While the location variation parameter σ can be set to the expected mass accuracy of the particular instrument, the optimal or close-to-the-optimal setting of the reliability parameter p_0 remains an open question. If a calibration serum is processed along with the data, a parameter search can be performed to optimize the method to the data produced by the machinery. The parameter which results in the best labeling of the calibration serum can be used in further analysis of data coming

from the same experimental setup.

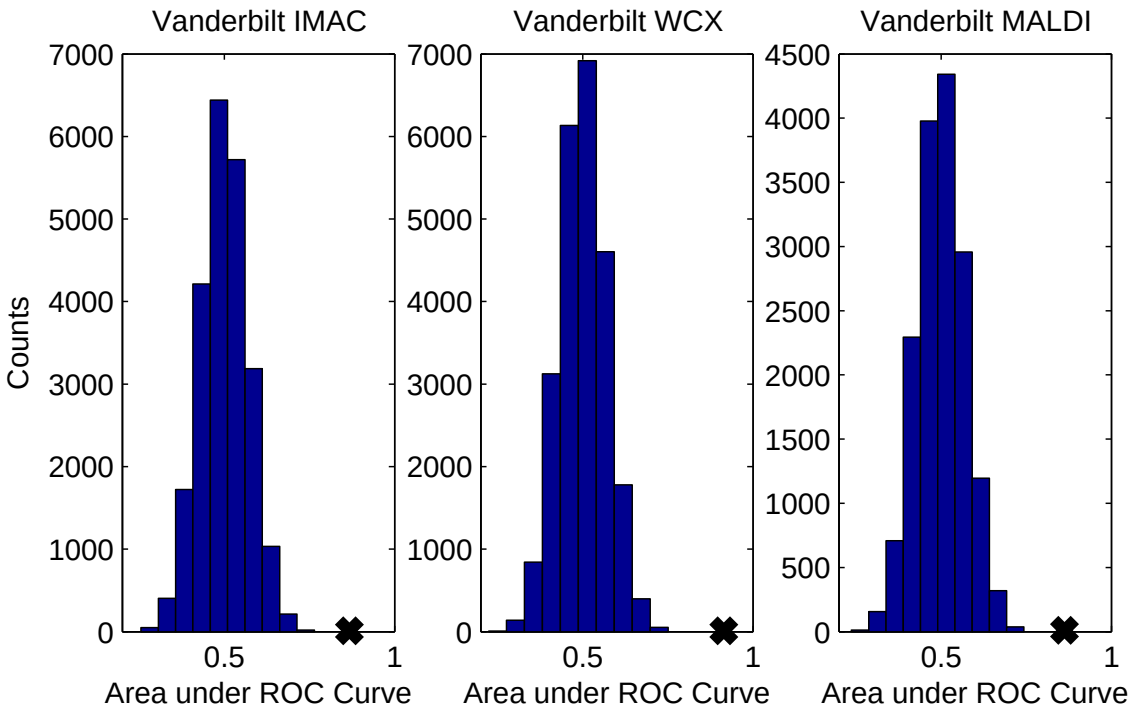

Figure 27: PACE distributions for Vanderbilt Lung SPORE data. Left panel: IMAC. Center panel: WCX. Right panel: MALDI. All three results under the true labeling (red cross) are significant at $\alpha = 0.001$ with respect to the null hypothesis distributions.

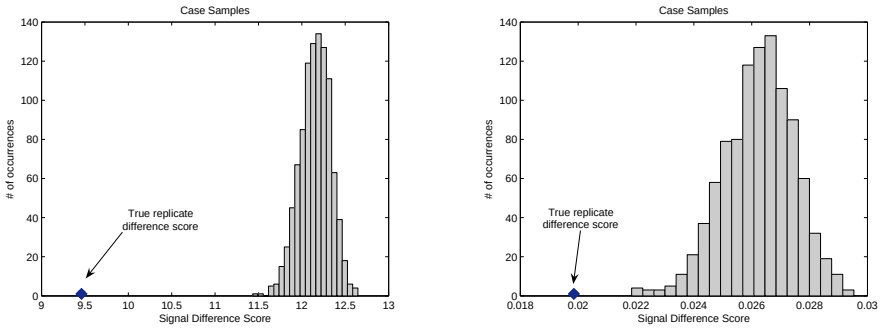

Figure 28: Distributions of signal difference scores for random groupings of profiles for case samples. The left panel displays signal difference scores taken over the entire range of the signal, while the right panel displays signal difference for a single feature at 8228 Da. The signal difference score for the true replicate spectra is plotted as a dot along the x-axis. The signal difference among the true replicates is much less than any observed signal difference among randomly grouped profiles. This indicates that the observed greater similarity between replications of the same sample is much less likely to be due to random effects.

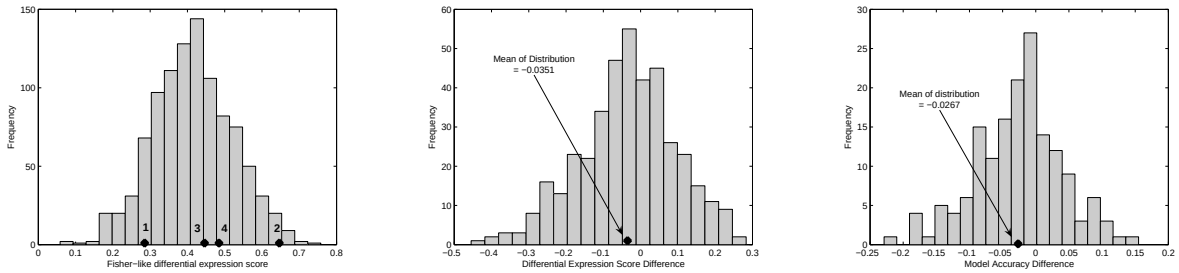

Figure 29: Left panel: Distribution of differential expression scores under random regroupings of profiles for the peak region at 12.938 kDa. The differential expression score for the peak in each of the 4 individual sessions is plotted as a dot along the x-axis. Middle panel: distribution of differences between the mean of mixed-session Fisher score distributions and single session Fisher scores for 100 peak regions. The distribution has a mean of -0.0351 and p-value of 5.588×10^{-8} for the null hypothesis: the mean is equal to 0. Right panel: distribution of differences between the mean accuracies of models trained on multi-session data and accuracies of models trained on single-session data. The mean of this distribution falls below 0 (= -0.0267), indicating an on-average benefit of training from single-session data.

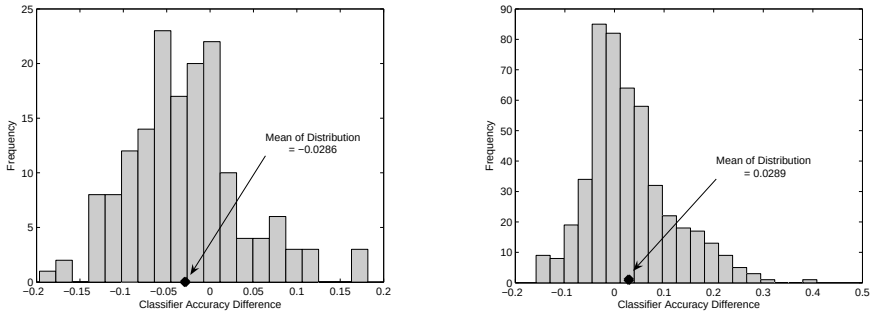

Figure 30: Left panel: distribution of accuracy differences between predictive models trained on multi-session data and models ideally trained on data from the same single session as the target test session. The mean below 0 (= -0.0286) indicates an advantage of the ideally trained single-session models. Right panel: distribution of accuracy differences between the same predictive models trained on multi-session data and models trained on single-session data from sessions other than the target testing set. The mean above 0 (= 0.0289) indicates an advantage of training on multi-session data. This illustrates the ability of predictive models trained on multi-session data to adapt to inter-session noise.

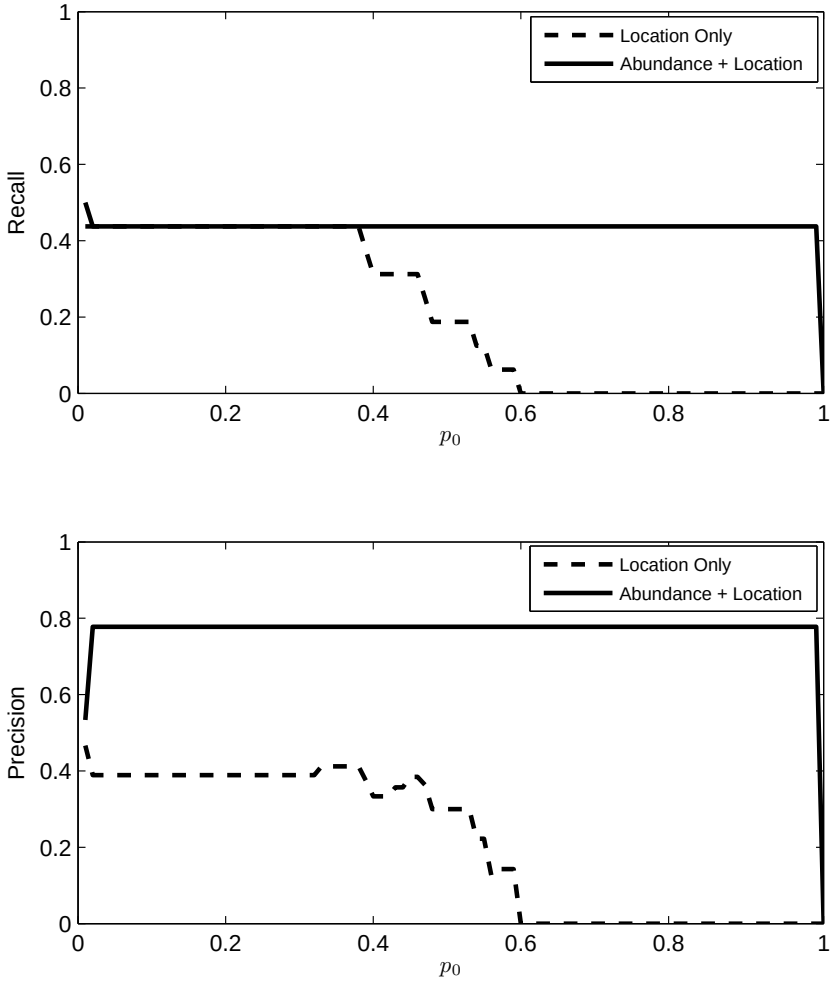

Figure 31: Performance variation at $\sigma = 0.05$ for the baseline peak-location method and the relative abundance-enhanced method. Recall (top) and precision (bottom) are plotted as a function of the reliability parameter p_0. As p_0 increases, both precision and recall of the peak-location method shrink as the requirement for close location-based matches becomes more strict. However, the importance of this parameter is noticeably downplayed in the relative abundance-enhanced method. Since fewer possibilities exist for the method to find peaks which are both appropriately positioned and sized, labelings and their performance and recall remain relatively stable. A significant gain in precision can be seen in the abundance-enhanced method, indicating fewer false positive peak-to-label matches.

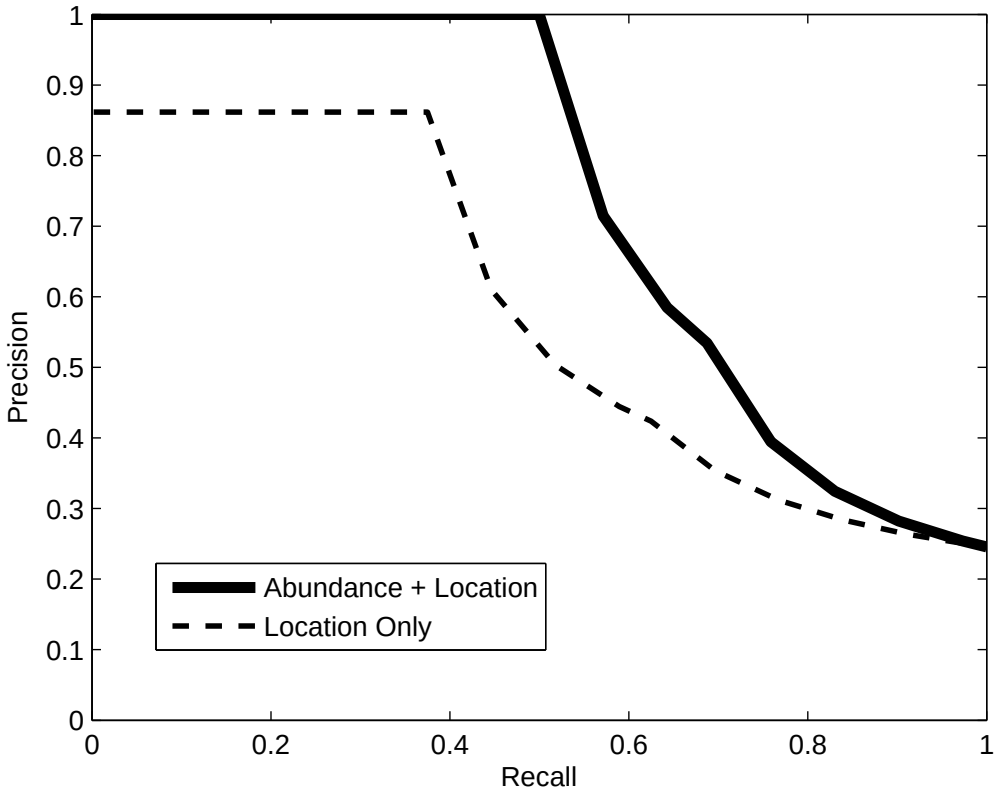

Figure 32: Comparison of Precision-Recall curves for both peak-labeling methods. The abundance-enhanced labeling method (thick solid line) improves over the method which uses only peak location information (thin dashed line). The areas under these curves were measured as 0.7387 for the abundance-enhanced method, versus 0.5782 for the basic peak location method.

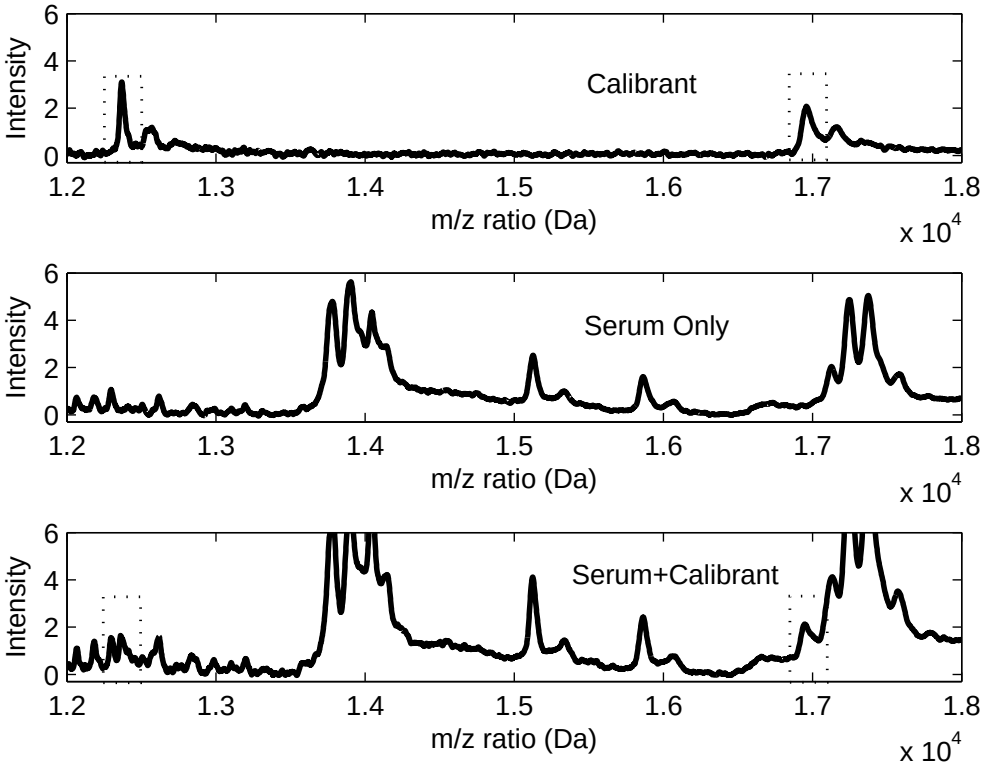

Figure 33: Calibrant spiked into whole serum sample. Top: Calibrant mixture shot alone. Center: Whole serum sample shot alone. Bottom: Whole serum sample spiked with calibrant. The spiked-in peaks are marked, and correspond to cytochrome C (12.360 kDa) and myoglobin (16.591 kDa) from the calibrant (also marked). ©2010 IEEE.

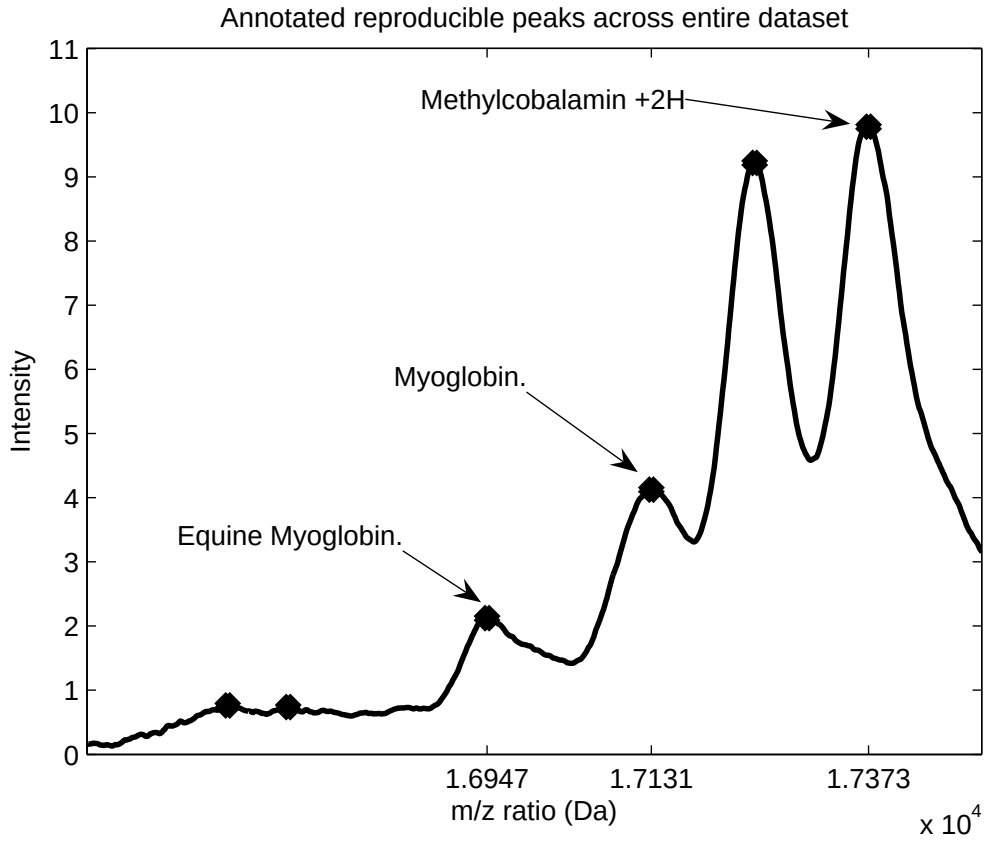

Figure 34: labeled peaks of serum-calibrant mixture in the range of [16500 17500] Da. The spiked-in peak at 16947 Da is correctly labeled as equine myoglobin. The second peak labeled myoglobin at 17131 Da corresponds to human myoglobin, which has a similar mass weight (17184 Da). The unlabeled peak positions could not be confidently assigned a label. ©2010 IEEE.

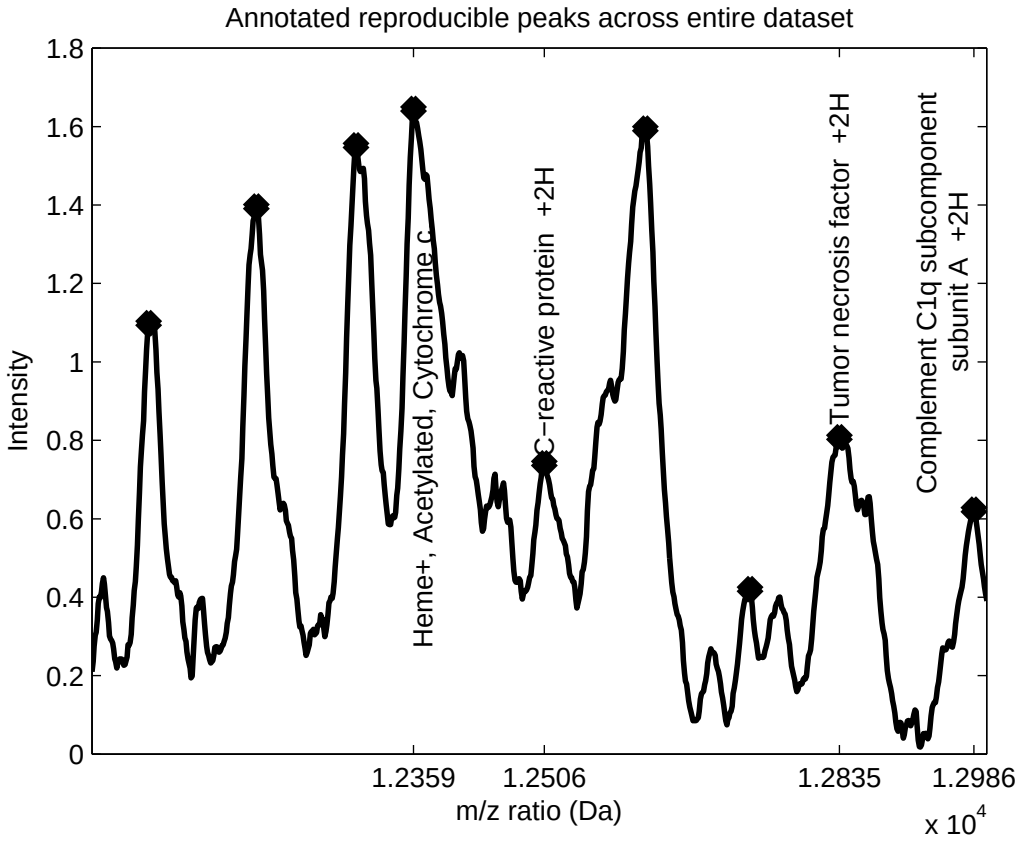

Figure 35: labeled peaks of serum-calibrant mixture in the range of [12000 13000] Da. The spiked-in peak at 12359 Da is correctly labeled as equine cytochrome C. The unlabeled peak positions could not be confidently assigned a label. ©2010 IEEE.

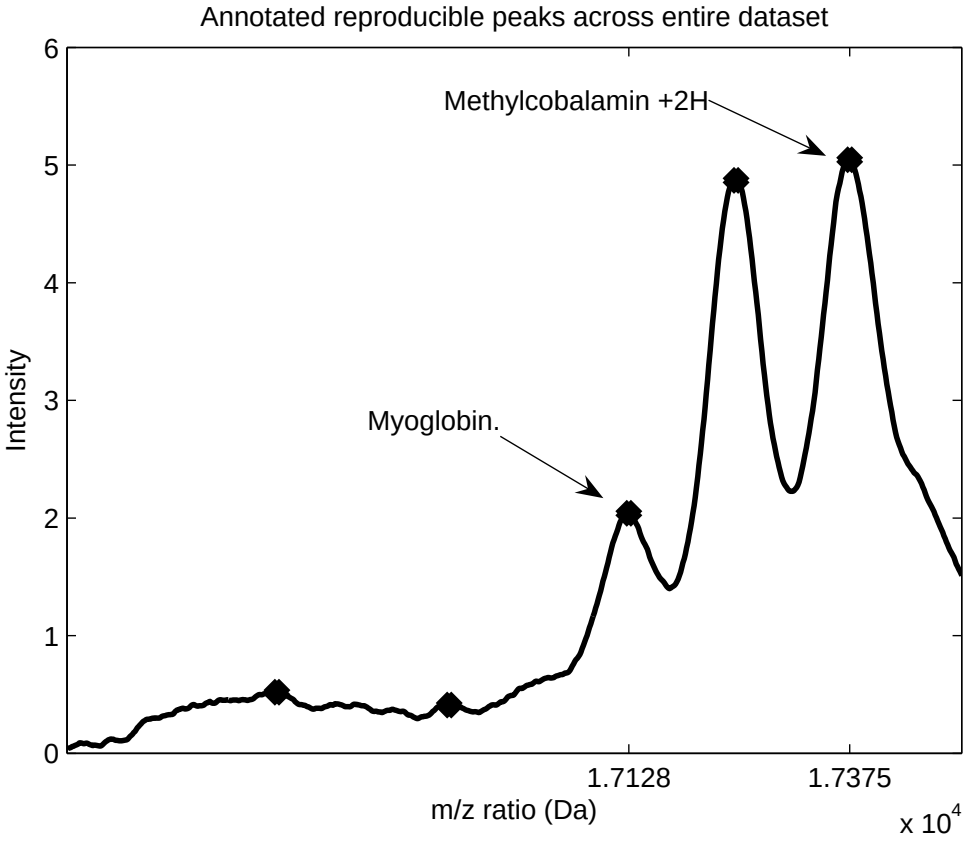

Figure 36: labeled peaks of whole human serum in the range of [16500 17500] Da. The peak at 17128 Da labeled myoglobin corresponds to human myoglobin, which has a similar mass weight (17184 Da). The unlabeled peak positions could not be confidently assigned a label. Although a peak is available at the expected mass weight of equine myoglobin (16947 Da), the intensity is too low to allow a confident decision.

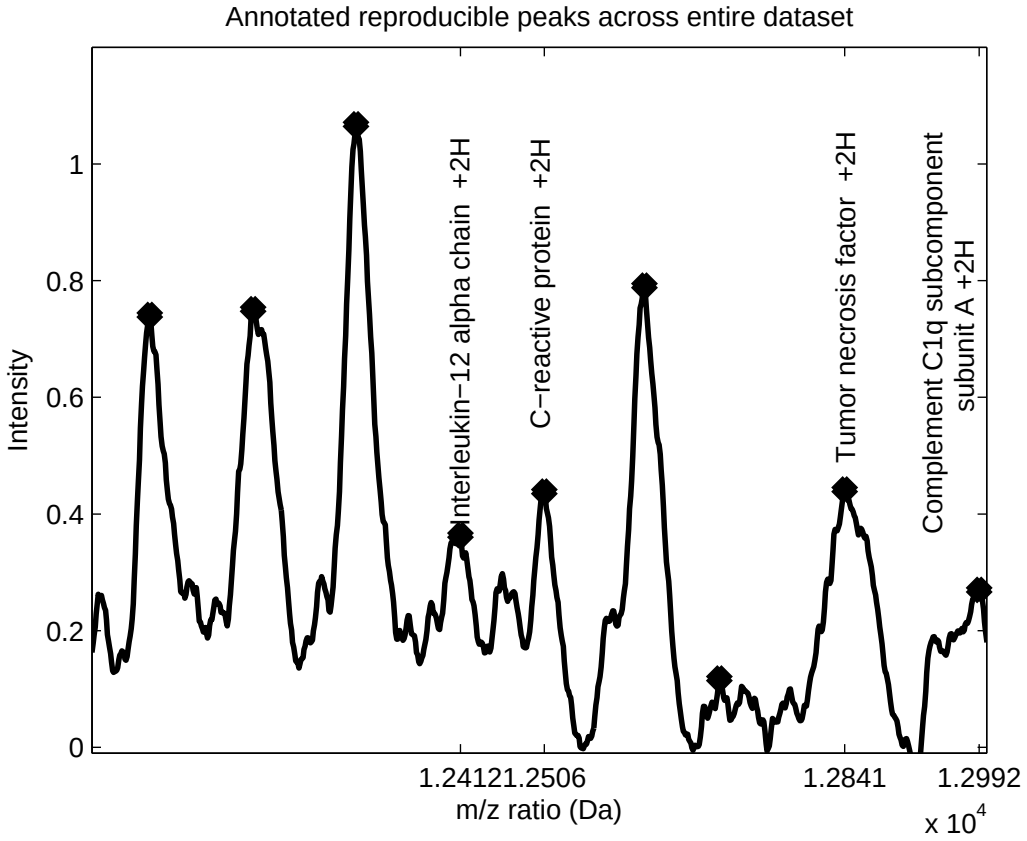

Figure 37: labeled peaks of whole human serum in the range of [12000 13000] Da. The unlabeled peak positions could not be confidently assigned a label. Although many peaks appear within the feasible range of equine cytochrome C (12360 Da), none are labeled as such due to extreme differences in location. ©2010 IEEE.

6.0 CONCLUSIONS

The ability to quickly, cheaply and noninvasively assess the health of a person is of great clinical importance. Mass spectrometry protein profiling is a method which shows promise in these directions. Discriminative signals can be found in order to predict healthy samples from a variety of complex diseases. Mass spectrometry, along with existing and emerging high-throughput technologies, faces a number of challenges which necessitate techniques that enable useful analysis and understanding of the data. Overall, the methods presented in this thesis contribute to a framework which improves the analysis of mass spectrometry protein profiling. The primary contributions of this thesis are briefly summarized below.

- Automatic Selection of Preprocessing Methods — Chapter 3 introduced and evaluated our novel approach to preprocessing MS protein profile data: the Standard Automatic Preprocessing procedure (SAP). To my knowledge, SAP is the first system which attempts to optimize the preprocessing of TOF-MS protein profile data by choosing among multiple preprocessing methods for each stage of preprocessing. SAP is competitive with a heavily-customized baseline preprocessing procedure we developed. When the baseline procedure has a negative impact on downstream predictive model performance, SAP is able to outperform the baseline procedure. The SAP procedure also demonstrates some unique characteristics which may allow it to inform a user on what preprocessing methods are best suited to a particular data source. For example, in the context of variance stabilization, SAP prefers to apply the Log-transformation to MALDI data, while the Cube-Root transformation is preferred by the SELDI data sources.
- Parallel decorrelation feature selection — Section 4.3.1 introduced the novel Parallel MAC Decorrelating feature selection method. In our experiments, we saw that the

MS protein profiling data sources are normally biased by an extremely high amount of correlations. These correlates need to be addressed directly by the feature selection or predictive modeling process, or through a combination of both. Our experiments showed that, on average, feature selection methods which take these correlations into account result in a better predictive model performance than methods which rank features in a multivariate or univariate way.

- Evaluation of kernel-learning approaches — Section 4.4.2 evaluated three kernel-learning approaches to determine whether or not kernel learning would impart a benefit to SVM-based predictive models versus the standard linear kernel. The experiments suggest that kernel learning is not preferable to a linear kernel with ℓ-1 regularization. Despite this, the calibrated probabilities of the hyperkernel are quite good, and future work could potentially tap this advantage.
- Pathway kernel — Section 4.3.6 describes the Pathway kernel, a kernel function which attempts to capture information about regulatory biological processes observable in MS protein profiles. The TOF-MS data source is not the best suited for use with the approach described for computing the kernel in Section 4.3.6. This is largely due to error in the translation process from gene interaction networks to feature mappings in the proteomic profiles. Newer technology with better annotations, and additional resources like EPO-KB [160] can help to reduce the amount of error in this translation process.
- A method for assessing the statistical significance of a predictive model's performance — Many high-throughput data sources suffer from too many dimensions and too few samples. MS protein profiling data are no different. The risk of achieving a good predictive performance just by chance increases especially with smaller numbers of samples, which is unavoidable especially with rare diseases. Section 5.4.1 demonstrates Permutation-Achieved Classification Error (PACE), a novel method for assessing the statistical significance of a predictive model's performance. PACE can be used to test whether a predictive model's power is more likely to be due to spurious patterns in the data. The experiments show that our predictive models frequently find discriminative information that is very unlikely to be due to random chance, suggesting that genuine surrogate biomarkers can be recovered from protein profiling data.

- Measures of reproducibility — The sensitivity of many high-throughput data sources, including MS protein profiling, can detect minute differences in replications of the same sample. Since proteins can change over time, a sample recovered from storage may generate a different protein profile. Section 5.3.2 introduced several measures of reproducibility which help the interpretive analysis of MS profile data generated over multiple time sessions, and possibly on different laboratory equipment. The experiments show that protein profiles retain their individuality over multiple sessions despite the effect of noise over time. We also showed that including profiles from multiple data generation sessions can improve the generalizability of predictive models on future profiles.
- Algorithms for peak labeling of molecular species in TOF-MS data — Section 5.4.3 introduced novel algorithms for labeling profile peak features with biological peptide identifiers. In experiments, we showed that a probabilistic model taking advantage of prior knowledge about protein abundance has enhanced precision and recall for labeling peaks in simulated data. The enhanced probabilistic model was also successful in selectively labeling calibrant peaks when spiked into a normal human serum sample.
- A simulator for MS protein profile data — Appendix A details a revision of a previously described physical model of a TOF Mass Spectrometer [1]. The revision includes instructions for how to simulate realistic TOF-MS spectra, given a list of UniProt peptide accession numbers. The simulator model's parameters are adjustable to reflect a variety of machinery types.

6.1 IMPACT ON BIOINFORMATICS DATA ANALYSIS

Many high-throughput data sources are very similar to MS protein profile data. Genomic and metabolomic data merely change the type of biological element being measured by the instrument. Variants of TOF-MS are currently being used to generate metabolomic data. Therefore, many of the techniques presented in this thesis can be applied to other types of bioinformatics data. In the following sections, I suggest some applications of these techniques outside of protein profiling to demonstrate their wider applicability.

6.1.1 Preprocessing of other "omic" datasets

Microarray data typically undergoes preprocessing much in the same vein as described in Chapter 3 for protein profile data. Stages of microarray preprocessing include background correction, normalization and summarization, which are analogs of MS protein profile preprocessing's baseline correction, normalization and smoothing stages. It has been shown in previous research that the choice of the most appropriate preprocessing methods is largely platform dependent [161, 162] and not easy to automate [163]. As new technologies emerge, significant effort is made to characterize noise sources and determine the optimal preprocessing procedures for each individual technology [163–165].

Repurposing the SAP preprocessing framework for microarray data preprocessing could assist the state-of-the-art efforts to automate and recommend appropriate preprocessing procedures. Large databases of microarray data from a variety of technologies exist in databases like the Stanford Microarray Database (SMD, [166]) or Gene Expression Omnibus (GEO, [143]). This data could be fed to SAP in an effort to learn preferences for preprocessing methods in relation to the originating data platform. In turn, efforts to characterize noise sources in microarray and metabolomic data can help refine the local stagewise scores of SAP to further improve the selection process.

Beyond "omic" data, image analysis is another subfield of biomedical informatics which requires significant data preprocessing. Since image data can be relatively large to store and transmit, lossless compression of the data is often required. Despite the compression, data must continue to be useful for clinical diagnosis. This tradeoff has been previously noted and studied [167], and presents yet another opportunity for the SAP framework to be applied. Global and local scores would balance the tradeoff between diagnostic information loss and storage size.

6.1.2 Feature Selection and Predictive Modeling Preferences

Feature selection remains an important topic for all high-throughput data sources, as the dimensionality of data will continue to increase with improving technology. At the time of this article, mass spectrometers like the Linear Trap Quadrupole-Orbitrap hybrid mass

spectrometer (Thermo Electron, Bremen, Germany) exist which can identify molecules at 2 parts per million [168]. Further improvements are being made to mass spectrometers, and the increasing resolution corresponds to higher data dimensionality. As isotopic distributions of molecules begin to be elucidated, the number of correlated features in the data will also increase. Correlation-aware techniques will be of utmost importance in order to perform good feature selection on these future data.

The experiments in feature selection and predictive modeling, presented in Chapter 4, have intuitive but important results. First, if our goal is to have the best classifier performance, ranking features univariately is often not the best way to do feature selection. Predictive models perform much better when their predictive features are considered in conjunction with other, or previously selected, features. Next, a simple predictive model, like the linear SVM, can frequently outperform predictive models based on more complicated kernels. The savings over computing a more complicated kernel, in terms of computational complexity, can be extremely beneficial. The choice of regularization penalty is also important. It has been previously proven that ℓ_1 regularization is effective even when the training samples are outnumbered by exponentially many irrelevant features [169]. The results in this thesis are consistent with this result.

6.1.3 Reproducibility of Results

All "omic" data sources are bound to suffer from reproducibility issues. Even those "omic" data sources which may appear in the future will face questions about reproducibility. A biological sample will degrade over time, no matter how strict and considerate the storage and processing protocols are. The reproducibility measures and evaluation framework presented in Chapter 5 are general enough to be adapted to any technology. This enables any "omic" technology user to evaluate for themselves whether that technology can generate reproducible and reliable data. This is a critical step for addressing the concerns about the applicability of a certain technology in a practical setting.

6.1.4 Next-Generation Genomic Sequencing

The advent of "next-generation sequencing" enables the rapid sequencing of genomes on a cellular basis. This allows the genomes of tumor tissue cells to be compared to genomes of cells from the tissue adjacent to, and distal from, the tumor. An emerging challenge in this new area of bioinformatics is the discovery of *driver* versus *passenger* mutations, both of which appear to be discriminative mutations between tumor and healthy tissue. However, the driver mutations are thought to be causally implicated in tumor progression, while passenger mutations are a result of chance or inconsequential errors during the cell division process. Being able to classify these two types of mutations is therefore valuable for driving further research.

The pathway kernel used in this thesis was an attempt to map genomic features to protein profile information. Although it was not successful in this application, the general technique could be applied to the analysis of driver/passenger mutation classification. Standard mutation classification approaches typically involve phenotypic, sequence-based and structure-based features [170, 171] of the resulting mutation, which are then fed to a predictive model like the SVM. Efforts to attach these features to downstream interacting gene "modules" is underway [172]. The modules differ primarily from classic pathways in that modules represent a group, of which genes may not have a direct interaction. This gives us the advantage of describing a driver mutation as being one that is present in any gene of a module, and then interacting genes of the mutated gene may exhibit a disregulated pathway. The obvious next step is to ask how these gene modules are affecting oncogenesis through protein expression. In this application, the pathway kernel could be built around these new gene modules and their resulting mutated proteins, instead of the gene pathways used in our experiments.

As these driver mutations are thought to be among the first somatic changes in the development of cancer, it may be useful to evaluate mutated gene module pathways to classify early-stage cancer profiles versus normal profiles. To identify the modules which are more likely to represent passenger mutations, the pathway kernel can be applied to classify later-stage cancers versus healthy tissue. A comparison of the most relevant features

for both of these analyses would be very revealing. Ideally, pathways and modules which predict cancer will differ between the two studies. Those pathways or modules effective for the early-versus-normal study, but less effective for the late-versus-normal study, could be more likely influenced by driver mutations. Pathways or modules which only appear relevant for the late-versus-normal study are probably more likely due to passenger mutations. In the worst case, this technique could narrow down the vast number of mutations which require driver/passenger classification.

6.2 OPEN QUESTIONS AND FUTURE WORK

The methods and results presented here raise a number of interesting open problems in proteomic profiling analysis. Several directions, as discussed below, could be undertaken in order to improve these methods further.

6.2.1 Tradeoff of the local and global scores in SAP

It is still unclear how to proportion the SP-curve's weight to the global DE score versus the local stagewise scores of each stage. In some cases, for example, the Heteroscedacity Retention score described in Section 3.2.1, it would be possible to estimate a weight by evaluating several weight combinations and determine the average benefit to multiple datasets under that weighting scheme. This could help to become a qualitative measure for the stagewise score; as the weight estimation begins to lean more on the global metric, it probably means that the local metric for that stage requires revision.

6.2.2 Class-sensitive Automatic Preprocessing

An improvement of Standard Automatic Preprocessing, which I call *Class-sensitive Automatic Preprocessing*, takes the ideas in SAP one level further. Each stage defines a pair of stagewise scores, one for features expected to be discriminative, and another for those which are superfluous. The goal in each stage is to optimize the DE score and normal stagewise

score only with respect to the discriminative features, while at the same time allowing the non-discriminative features to also select their own preprocessing method. This allows discriminative features to be preferentially treated and may improve the overall quality of the signal resulting from preprocessing.

6.2.3 Method Parameterization Search for SAP

The amount of methods available to SAP for this paper were small, because each method operated using default parameters. Ideally, SAP treats differently parameterized methods (for example, a smoothing procedure with sliding window size of 24 features, instead of 12) as competitors for a given stage. An automated grid search over a parameter space for preprocessing methods might lead to better performance in some stages of preprocessing.

6.2.4 Combining feature selection with kernel learning

One of the better performing methods in this chapter is the linear SVM with ℓ_1-regularization, combined with the parallel decorrelation algorithm. It remains to be seen whether any of the kernel-learning methods can be improved upon if they go through feature selection. We saw that the hyperkernel approach is able to give well-calibrated probabilities despite not being accurate. It would be interesting to see whether we could improve the classification accuracy by performing feature selection, but retain the benefit of better calibrated probability estimates.

APPENDIX A

SIMULATION OF TOF-MS DATA

The model proposed by Coombes et al. [1] describes a mathematical model for the simulation of TOF-MS proteomic profiles. It is available for the S-Plus programming environment from the authors' website (http://bioinformatics.mdanderson.org/cromwell.html). The input to the system is a set of parameters which govern the behavior of physical properties of molecules and how they are propagated and detected by the mass spectrometer. Molecular weights can be fed to the simulator to produce their resulting mass spectra. I was unable to reproduce realistic spectra using this package directly, but using the derivations and methods below, I was able to make an improvement on the simulator which does produce realistic spectra.

A.1 PARAMETERS OF THE MODEL

Figure 38 displays a general schematic of a mass spectrometer simulated by this model. Let the parameters of the data simulator, as described by Coombes et al. in [1], be defined as follows: L is the length of the drift tube. D_1 is the distance from the sample plate to the electric field, which is bounded at the front by the focusing grid. D_2 is the width of the electric field, which is bounded on the far end by the accelerating grid. The laser, upon striking the sample, imparts a velocity v_0 on a molecule from the distribution $N(\mu, \sigma)$. Energized ions drift for delay time δ before reaching the electric field. Voltage V_1 is then applied to the sample plate, causing ions to be propelled into the electric field. The electric

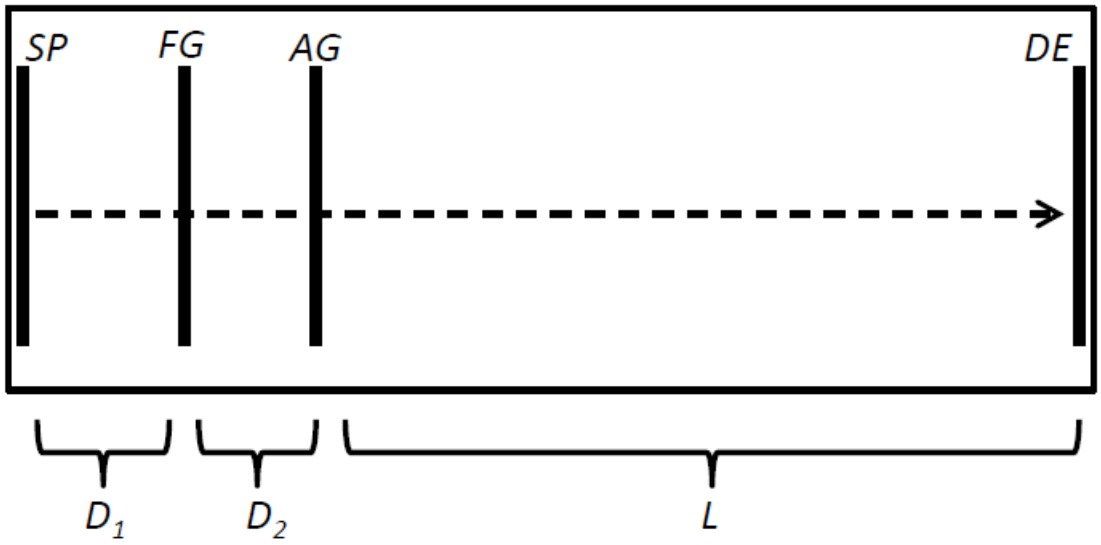

Figure 38: A general diagram of a mass spectrometer simulated by the Coombes et al. model. (SP): sample plate, (FG): focusing grid, (AG): accelerating grid, (DE): detector plate.

field is charged with voltage V, which further increases the acceleration of ions. These ions continue onward until they reach the detector. The amount of time during which the detector is able to measure incoming particles (i.e. the resolution) is given as τ.

The default values of the parameters of the model, under both the implementation in [1] and the implementation used in this thesis are given in Table 20.

A.2 DERIVATION OF TIME-OF-FLIGHT VALUES

Using the Time-of-Flight equations derived in [1], I was not able to reproduce protein profiles which looked realistic compared to genuine TOF-MS data from the available biological datasets used in this work. I do believe the description in [1] is sound, but I am unable to find the error myself, due to a lack of understanding of the physical laws used for their derivation. Instead, I derived the equations for particles using the most fundamental laws

Table 20: Simulator Parameters and Default Settings

Parameter	Coombes et al.	This work
D_1	$17e^{-3}$	$17e^{-3}$
D_2	$8e^{-3}$	$8e^{-3}$
L	1	1
V	20000	20000
V_1	2000	2000
δ	$600e^{-9}$	$600e^{-9}$
τ	$4e^{-9}$	$4e^{-9}$
μ	350	350
σ	50	50

of classical physics. To my knowledge, the derivation below is a correct procedure for estimating Time-of-Flight (TOF) values for molecules present in the sample. Please excuse my incredibly simplified explanation, for the field of physics is not my area of expertise, and if I make errors in assumption or otherwise, then I hope to facilitate their discovery by the more advanced reader. I use the SI units of measurement, where possible, in the following derivation.

Electric fields are measured in units of volts (V, electric potential) per meter (m, distance) or newtons (F, force) per coulomb (electric charge). Let E represent an electric field and $z = 1.602e^{19}$ represent the elementary charge of one electron in coulombs. Therefore,

$$E = \frac{V}{m} = \frac{F}{z} \tag{A.1}$$

From Newton's Second Law of Motion, net force on a particle is equal to the mass (g, grams, to prevent confusion with meters) of the particle times its acceleration a:

$$F = ga \tag{A.2}$$

Rewriting Equation A.2 results in the following:

$$\begin{aligned} F &= zE \\ F &= \frac{zV}{m} \end{aligned} \tag{A.3}$$

And therefore, the acceleration of a particle in a particular electric field will be given as:

$$\begin{aligned} a &= \frac{F}{g} \\ a &= \frac{zV}{mg} \end{aligned} \tag{A.4}$$

Our goal is to calculate the time it takes for an ion to travel from the sample plate to the detector. The linear distance d traveled by a particle with initial velocity v, under uniform, constant acceleration a, is given as:

$$d = vt - \frac{1}{2}at^2 \tag{A.5}$$

This equation can be rewritten in terms of time by applying the quadratic formula.

$$t = \frac{-v \pm \sqrt{v^2 - 2a(d)}}{a} \tag{A.6}$$

While negative terms can result from this equation, we are not interested in negative times (these ions would not reach the detector).

We split up the calculation of the TOF values into three stages. First, there is the time for the focusing phase, t_f, which occurs from the ion's location in the plume, x_0 to the position of the focusing grid of the electric field, D_1. Next, for the accelerating phase, t_a, we consider the time taken from entering the electric field at D_1 and leaving the electric field at D_2. Finally, the drift time t_d is calculated as the time taken to travel from D_2 to the detector, a distance of L.

The initial velocity v_0 imparted on an ion is given stochastically by the distribution $N(\mu, \sigma)$ every time an ion requires its TOF value calculated. The three flight times are calculated using Equation A.2 as follows:

$$t_f = \frac{-v_0 + \sqrt{v_0^2 - 2a_f(D_1 - x_0)}}{a_f} \qquad \texttt{where } a_f = \frac{zV_1}{D_1 g} \tag{A.7}$$

The velocity of the ion at the end of the focusing phase, v_t, will be equal to $v_t = t_f * a_f + v_0$.

$$t_a = \frac{-v_f + \sqrt{v_f^2 - 2a_f(D_1 - x_0)}}{a_f} \qquad \texttt{where } a_a = \frac{zV}{D_2 g} \tag{A.8}$$

The velocity of the ion at the end of the acceleration phase, v_a, will be equal to $v_a = t_a * a_a + v_t$.

In the drift phase, acceleration is assumed to be 0 (constant velocity). Therefore, to prevent numerical error from dividing by 0, we use the relationship between distance, velocity and time to compute t_d. Recalling that distance = rate * time:

$$t_d = \frac{L}{v_a} \tag{A.9}$$

Finally, Time-of-Flight is calculated as $\delta + t_f + t_a + t_d$. This procedure results in realistic TOF-MS spectra.

A.3 USING THE SIMULATOR TO GENERATE SPECTRA

My simulator takes as input, values for the parameters listed in Table 20, and a list of peptide accession numbers corresponding to peptides which should appear in the mass spectra.

First, a sample amount S is defined in kilograms. This will be the total amount of sample analyzed. I do not know what a reasonable setting is for this parameter. For most experiments, I set $S = 4e^{-16}$kg.

Next, each separate molecular species (peptide) to be analyzed in the sample is defined by its primary amino acid peptide sequence. In my experiments, the primary sequence is obtained from the UniProt database [10]. The expected mass of the peptide is computed as the sum of the average isotopic masses of the amino acids in the given sequence, in addition to the average isotopic mass of a single water molecule. Post-translational modifications can also affect the peptide's sequence, and these are taken into account when calculating the peptide mass. Signal peptides from complete protein sequences are removed, and when documented in the UniProt entry, the mass of a post-translational modifying molecule is added to the peptide's expected mass.

Each molecular species is assigned a relative proportion, which indicates how abundant that molecular species is relative to other molecular species. Following this, I calculate the number of possible molecules pm which can fit in the sample. This is calculated by $pm = S/m_i$, where m_i is the mass of molecular species i. The simulator is used to calculate TOF values using the equations above for each of the pm molecules. A quadratic relationship is expected between TOF values and m/z, such that the TOF value squared approximates the m/z value to a constant factor [173]. Let the i^{th} row of matrix of A be $]1, TOF_i, TOF_i^2]$ for all ions to be simulated in the spectra. Let $b_i = m_i$. The coefficients x for a least-squares solution to the system of equations $Ax = b$ are computed. Finally, TOF values are binned according to the resolution τ of the mass spectrometer. These TOF values are then converted to m/z values by multiplication with the coefficients x learned above. Naturally, the number of TOF values entering that bin reflect the intensity measured at that particular m/z value. The result is a spectrum consisting of detected m/z values and the amount of molecules detected in that time bin, which corresponds to relative abundance.

APPENDIX B

TABLES OF MATHEMATICAL FORMULAE

Table 23: Dataset Characterization Measures

Geometric mean	$= \frac{1}{d}\sum_{i=1}^{d}\left[\prod_{j=1}^{n} x_{i,j}\right]^{1/n}$
Harmonic mean	$= \frac{1}{d}\sum_{i=1}^{d}\frac{n}{\sum_{i=j}^{n}(1/X_i)}$
Trim mean	$=$
	$\frac{1}{d}\sum_{i=1}^{d}(\mu_i$with 20% highest and lowest percentiles removed from$x_i)$
Standard Deviation	$= \frac{1}{d}\sum_{i=1}^{d}\left(\frac{1}{n-1\sum_{j=1}^{n}(x_{i,j}-\frac{1}{n}\sum x_{i,j})}\right)$
Range	$= \frac{1}{d}\sum_{i=1}^{d}(\max(x_i)-\min(x_i))$
Median	$= \frac{1}{d}\sum_{i=1}^{d} median(x_i)$
Interquartile Range	$= \frac{1}{d}\sum_{i=1}^{d}$ 75$^{\text{th}}$ percentile of x_i - 25$^{\text{th}}$ percentile of x_i)
Maximum / Minimum Eigenvalue	= Computed from sample covariance matrix
Skewness	$= \frac{1}{d}\sum_{i=1}^{d}\frac{\mathbf{E_j}(x_{i,j}-\mu_i)^3}{\sigma_i^3}$
Kurtosis	$= \frac{1}{d}\sum_{i=1}^{d}\frac{\mathbf{E_j}(x_{i,j}-\mu_i)^4}{\sigma_i^4}$
Correlation Coefficient	= Avg correlation coefficient btwn all pairs
Z-score	$= \frac{1}{d}\sum_{i=1}^{d}\frac{1}{n}\sum_{j=1}^{n}\frac{x_{i,j}-\mu_i}{\sigma_i}$
Euclidean Distance	= Avg dist. btwn all pairs (See table 21)
Mahalanobis Distance	= Avg dist. btwn all pairs (See table 21)
Cityblock Distance	=Avg dist. btwn all pairs (See table 21)
Chi-Squared PDF	$=\frac{x^{(v-2)/2}e^{-x/2}}{2^{v/2}\Gamma(v/2)}$

Table 23: (continued)

Chi-Squared CDF	$= \int_0^x \frac{t^{(v-2)/2}e^{-t/2}}{2^{v/2}\Gamma(v/2)}dt$
Normal PDF	$= \frac{1}{\sigma\sqrt{2\pi}}e^{-(x-\mu)^2}/2\sigma^2$
Normal CDF	$= \frac{1}{\sigma\sqrt{2\pi}}\int_{-\inf}^{x} e^{-(x-\mu)^2}/2\sigma^2 dt$
Binomial PDF	$= \binom{n}{x}p^x q^{(n-x)}\mathbf{I}_{0,1,\ldots,n}(x)$
Discrete Uniform CDF	$= \frac{\text{floor}(x)}{N}\mathbf{I}_{0,1,\ldots,n}(x)$
Exponential PDF	$= \frac{1}{u}e^{x/u}$
F PDF	$= \frac{\Gamma\left[\frac{(\nu_1+\nu_2)}{2}\right]}{\Gamma\left(\frac{\nu_1}{2}\right)\Gamma\left(\frac{\nu_2}{2}\right)}\left(\frac{\nu_1}{\nu_2}\right)^{\frac{\nu_1}{2}}\frac{x^{\frac{\nu_1-2}{2}}}{\left[1+\left(\frac{v_1}{v_2}\right)x\right]^{\frac{\nu_1+\nu_2}{2}}}$
Gamma PDF	$= \frac{1}{b^a\Gamma(a)}x^{a-1}e^{x/b}$
Geometric CDF	$= \sum_{i=0}^{\text{floor}(x)} rq^i$
Hypergeometric CDF	$= \sum_{i=0}^{x} \frac{\binom{K}{i}\binom{M-K}{N-i}}{\binom{M}{N}}$
Lognormal PDF	$= \frac{1}{x\sigma\sqrt{2\pi}}e\frac{-(ln(x)-\mu)^2}{2\sigma^2}$
Poisson PDF	$= \frac{\lambda^x}{x!}e^{-\lambda}\mathbf{I}_{0,1,\ldots,n}(x)$
Rayleigh PDF	$= b^{x/2}e^{(-x^2/2b^2)}$
Student's t PDF	$= \frac{\Gamma((v+1)/2)}{\Gamma(v/2)}\frac{1}{\sqrt{v\pi}}\frac{1}{(1+(x^2/v))^{(v+1)/2}}$

Table 21: Examples of distance metrics for clustering.

Metric	Formula
Euclidean distance	$d(r,s) = (x_r - x_s)(x_r - x_s)'$
Standardized Euclidean distance	$d(r,s) = (x_r - x_s)trace(\Sigma)^{-1}(x_r - x_s)'$
Mahalanobis distance	$d(r,s) = (x_r - x_s)\Sigma^{-1}(x_r - x_s)'$
City Block metric	$d(r,s) = \sum_{j=1}^{n} \lvert x_{rj} - x_{sj} \rvert$
Minkowski metric	$d(r,s) = \sqrt[p]{\left(\sum_{j=1}^{n} \lvert x_{rj} - x_{sj} \rvert^p\right)}$
Cosine distance	$d(r,s) = \left(1 - \frac{x_r x_s'}{\sqrt{x_r' x_r}\sqrt{x_s' x_{s)}}}\right)$
Correlation distance	$d(r,s) = 1 - \frac{(x_r - \bar{x}_r)(x_s - \bar{x}_s)'}{\sqrt{(x_r - \bar{x}_r)(x_r - \bar{x}_r)'}\sqrt{(x_s - \bar{x}_s)(x_s - \bar{x}_s)'}}$
Hamming distance	$d(r,s) = \frac{\#(x_{rj} \neq x_{sj})}{n}$
Jaccard distance	$d(r,s) = \frac{\#[(x_{rj} \neq x_{sj}) \wedge ((x_{rj} \neq 0) \vee (x_{sj} \neq 0))]}{\#[(x_{rj} \neq 0) \vee (x_{sj} \neq 0)]}$

x and x' denote a column vector and its transpose respectively.

x_r and x_s indicate the r^{th} and s^{th} samples in the data set, respectively.

x_{rj} indicates the j^{th} feature of the r^{th} sample in the data set.

$\bar{x}_r$ indicates the mean of all features in the r^{th} sample in the data set.

Σ is the sample covariance matrix.

Table 22: Formulae for popular filter scores

Filter Name	*Formula*
Fisher Score	$score(i) = \frac{(\mu_+(i)-\mu_-(i))^2}{(\sigma_+(i))^2+(\sigma_-(i))^2}$
Student t–test	$score(i) = (\mu_+(i) - \mu_-(i))/\sqrt{\frac{\sigma_+^2}{n_+} + \frac{\sigma_-^2}{n_-}}$
Mutual Information	$score(X) = \sum_x \sum_y p(X = x, Y = y) \cdot \log \frac{p(X=x,Y=y)}{p(X=x)\cdot(Y=y)}$
(Chi–Square)χ^2	$score(X) = \sum_x \sum_y \frac{(p(X=x,Y=y)-p(X=x)\cdot p(Y=y))^2}{p(X=x)\cdot p(Y=y)}$
AUC	$score(i) = \int$ ROC Curve for feature i
J–measure	$score(X) = \sum_x p(X = x\|Y = 0) - p(X = x\|Y = 1) \cdot \log \frac{p(X=x\|Y=0)}{p(X=x\|Y=1)}$
$J5$ Score	$score(i) = \frac{\mu_+(i)-\mu_-(i)}{\frac{1}{m}\sum_{j=1}^{m}\|\mu_+(j)-\mu_-(j)\|}$

The standard SAM technique is meant to be used in a permutation setting, however, the scoring criteria can still be used for filtering methods.

$score(i) = \frac{\mu_+(i)-\mu_-(i)}{s(i)+s_0}$

The correcting constants $s(i)$ and s_0 are computed as follows:

$s(i) = \sqrt{\frac{(1/n_+)+(1/n_-)}{(n_1+n_2-2)}\left[\sum_{j=1}^{n_+}(x_j(i) - \mu_+(i))^2 + \sum_{j=1}^{n_-}(x_j(i) - \mu_-(i))^2\right]}$

$s_0 = 1$ for purposes of simplicity.

APPENDIX C

LISTS OF TOP TWENTY PATHWAYS USED BY PATHWAY KERNEL PER DATASET

Table 24: Lists of 20 most relevant MSigDB Pathway Identifiers per dataset

COPD	DCPATHWAY
	module_287
	GTGGGTGK_UNKNOWN
	CARM_ERPATHWAY
	V$ZID_01
	V$PITX2_Q2
	module_49
	AMINO_ACID_AND_DERIVATIVE_METABOLIC_PROCESS
	GH_GHRHR_KO_6HRS_DN
	MESODERM_DEVELOPMENT
	TGCCTTA,MIR-124A
	CCTGAGT,MIR-510
	HSA00480_GLUTATHIONE_METABOLISM
	HSA00604_GLYCOSPHINGOLIPID_BIOSYNTHESIS_GANGLIOSERIES
	RESPONSE_TO_WOUNDING

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
	UVB_NHEK1_UP
	V$AP2REP_01
	HDACI_COLON_BUT16HRS_DN
	RNA_CATABOLIC_PROCESS
	MORF_PPP2R5B

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
Hepatitis C	HSA00340_HISTIDINE_METABOLISM
	V$ATF1_Q6
	VESICLE_MEDIATED_TRANSPORT
	RUTELLA_HEPATGFSNDCS_UP
	ZHAN_PCS_MULTIPLE_MYELOMA_SPKD
	PYRUVATE_METABOLISM
	V$AFP1_Q6
	ESR_FIBROBLAST_UP
	RRCCGTTA_UNKNOWN
	PROTEIN_AMINO_ACID_DEPHOSPHORYLATION
	V$AR_01
	V$OCT1_B
	PROTEIN_TYROSINE_PHOSPHATASE_ACTIVITY
	V$HMEF2_Q6
	V$AREB6_04
	module_543
	GOLGI_ASSOCIATED_VESICLE
	CELL_PROJECTION
	UV_ESR_OLD_UNREG
	module_125

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
ILD	LYTIC_VACUOLE
	VACUOLE
	V$AP2REP_01
	module_15
	chr9q
	CCTGCTG,MIR-214
	INTRACELLULAR_PROTEIN_TRANSPORT
	REGULATION_OF_PROGRAMMED_CELL_DEATH
	HSA00641_3_CHLOROACRYLIC_ACID_DEGRADATION
	V$CREBP1_01
	module_155
	AGUIRRE_PANCREAS_CHR1
	RESPONSE_TO_ORGANIC_SUBSTANCE
	SA_BONE_MORPHOGENETIC
	FLECHNER_KIDNEY_TRANSPLANT_WELL_UP
	CTACTGT,MIR-199A
	BECKER_ESTROGEN_RESPONSIVE_SUBSET_2
	LEE_CIP_DN
	MORF_ERCC2
	NI2_MOUSE_UP

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
Diabetes	GTGCCAT,MIR-183
	RIBOSOMAL_SUBUNIT
	chr14q11
	V$TBP_01
	MITOCHONDRIAL_MATRIX
	MEMBRANE_FRACTION
	chr10q24
	CATTGTYY_V$SOX9_B1
	CALCIUM_ION_BINDING
	module_98
	ION_BINDING
	module_124
	GNF2_CDH11
	TATAAA_V$TATA_01
	module_5
	KANG_TERT_DN
	BRCA_BRCA1_POS
	MGGAAGTG_V$GABP_B
	PEART_HISTONE_DN
	MORF_TTN

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
Melanoma I	V$WHN_B
	module_229
	HDACI_COLON_BUT12HRS_DN
	CTCTATG,MIR-368
	V$MYOD_Q6_01
	CORTICAL_ACTIN_CYTOSKELETON
	ATGCTGG,MIR-338
	CELLULAR_RESPONSE_TO_STIMULUS
	chr15q23
	MEMBRANE
	CORTICAL_CYTOSKELETON
	N_GLYCAN_DEGRADATION
	MORF_EIF3S2
	module_212
	CALCIUM_REGULATION_IN_CARDIAC_CELLS
	V$ELK1_01
	MEMBRANE_PART
	LEE_TCELLS2_UP
	WELCSH_BRCA_DN
	V$ETS2_B

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
Breast Cancer	chr12q
	CELL_PROJECTION_BIOGENESIS
	chr10q25
	STEROID_HORMONE_RECEPTOR_SIGNALING_PATHWAY
	GNF2_NPM1
	CELL_CORTEX
	BHATTACHARYA_ESC_UP
	chr1q23
	GNF2_SPI1
	HSA00521_STREPTOMYCIN_BIOSYNTHESIS
	chr5p15
	ZHAN_PCS_MULTIPLE_MYELOMA_SPKD
	TUBE_MORPHOGENESIS
	module_533
	chr2q35
	ZHAN_MM_CD138_CD2_VS_REST
	CONDENSED_CHROMOSOME
	PEART_HISTONE_UP
	CELL_DIVISION
	module_61

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
Pancreatic Cancer I	module_545
	OLD_FIBRO_UP
	ADIP_DIFF_CLUSTER3
	STEMPATHWAY
	TCAPOPTOSISPATHWAY
	LEE_MYC_TGFA_DN
	GGCNNMSMYNTTG_UNKNOWN
	HYPOPHYSECTOMY_RAT_UP
	LEE_CIP_DN
	MEMBRANE_BOUND_VESICLE
	LU_IL4BCELL
	MORF_IL4
	V$MEF2_03
	LYMPHOCYTE_DIFFERENTIATION
	module_259
	WELCSH_BRCA_DN
	POD1_KO_MOST_DN
	NUCLEOLAR_PART
	REGULATION_OF_PH
	chr4p14

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
Pancreatic Cancer II	chr4q23
	TCA
	KREBS_TCA_CYCLE
	NEGATIVE_REGULATION_OF_MAP_KINASE_ACTIVITY
	GGCNNMSMYNTTG_UNKNOWN
	THELPERPATHWAY
	STEMPATHWAY
	CAR_IGFBP1
	GLYCEROPHOSPHOLIPID_METABOLIC_PROCESS
	V$AHR_Q5
	RYTGCNNRGNAAC_V$MIF1_01
	chr2q23
	TCAPOPTOSISPATHWAY
	ATCMNTCCGY_UNKNOWN
	GLYCEROPHOSPHOLIPID_BIOSYNTHETIC_PROCESS
	MORF_PTPRR
	WALLACE_JAK2_DIFF
	ACTACCT,MIR-196A,MIR-196B
	V$HIF1_Q3
	module_166

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
Prostate Cancer	DNA_PACKAGING
	ENZYME_LINKED_RECEPTOR_PROTEIN_SIGNALING_PATHWAY
	module_41
	V$PAX4_01
	V$NKX22_01
	module_5
	CAGGTA_V$AREB6_01
	CTTTGA_V$LEF1_Q2
	V$PAX4_03
	TGGAAA_V$NFAT_Q4_01
	TRANSITION_METAL_ION_BINDING
	CYTOPLASMIC_PART
	V$OCT1_07
	V$DR4_Q2
	SYSTEM_PROCESS
	NEUROLOGICAL_SYSTEM_PROCESS
	WGTTNNNNNAAA_UNKNOWN
	PLASMA_MEMBRANE
	STRESS_ARSENIC_SPECIFIC_UP
	NUCLEOBASE__NUCLEOSIDE__NUCLEOTIDE_AND_NUCLEIC_ACID_METABOLIC_PROCESS

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
Scleroderma	TTCYNRGAA_V$STAT5B_01
	V$AP1_Q6_01
	RESPONSE_TO_ABIOTIC_STIMULUS
	V$NRF2_Q4
	V$MYOD_Q6
	TGANTCA_V$AP1_C
	YCATTAA_UNKNOWN
	V$YY1_Q6
	module_118
	module_24
	module_334
	V$NFE2_01
	V$HFH4_01
	CTTTAAR_UNKNOWN
	module_23
	RESPONSE_TO_CHEMICAL_STIMULUS
	V$GR_01
	module_330
	CELLULAR_HOMEOSTASIS
	V$ER_Q6_01

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
UPCI Lung Cancer	module_349
	KANNAN_P53_UP
	TPA_SKIN_UP
	LAMB_CYCLIN_D1_UP
	MORF_PPP1CC
	LEE_CIP_UP
	SHIPP_FL_VS_DLBCL_DN
	LIZUKA_L1_GR_G1
	module_166
	GAMETE_GENERATION
	chr13q33
	INOSITOL_PHOSPHATE_METABOLISM
	chr18q21
	PANTOTHENATE_AND_COA_BIOSYNTHESIS
	module_540
	NKTPATHWAY
	METHOTREXATE_PROBCELL_UP
	V$TAXCREB_01
	NAKAJIMA_MCSMBP_MAST
	module_107

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
Vanderbilt Lung IMAC	HOFMANN_MDS_CD34_LOW_AND_HIGH_RISK
	module_286
	module_429
	UREA_CYCLE_AND_METABOLISM_OF_AMINO_GROUPS
	DEVELOPMENTAL_MATURATION
	ZUCCHI_EPITHELIAL_UP
	HYDROLASE_ACTIVITY__ACTING_ON_GLYCOSYL_BONDS
	IDX_TSA_UP_CLUSTER2
	module_440
	chr6q14
	NO2IL12PATHWAY
	UBIQUITIN_LIGASE_COMPLEX
	module_424
	RECEPTOR_SIGNALING_PROTEIN_ACTIVITY
	SARSPATHWAY
	CANTHARIDIN_DN
	IRITANI_ADPROX_DN
	DNA_DAMAGE_RESPONSE__SIGNAL_TRANSDUCTION_BY_P53_CLASS_MEDIATOR
	FLECHNER_KIDNEY_TRANSPLANT_REJECTION_PBL_DN
	IDX_TSA_UP_CLUSTER4

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
Vanderbilt Lung WCX	METPATHWAY
	GTPASE_REGULATOR_ACTIVITY
	AACTGAC,MIR-223
	CCR3PATHWAY
	chr2q37
	PROTEASOMEPATHWAY
	P53_SIGNALING
	GNF2_G22P1
	BRENTANI_SIGNALING
	TGAGATT,MIR-216
	PROTEASOME
	GOLUB_ALL_VS_AML_UP
	chr2p22
	TGACATY_UNKNOWN
	GCM_PSME1
	HSA00670_ONE_CARBON_POOL_BY_FOLATE
	RECEPTOR_ACTIVITY
	MORF_CASP2
	RNA_METABOLIC_PROCESS
	AGGGCAG,MIR-18A

Table 24: (continued)

Dataset	MSigDB Pathway Identifier
Vanderbilt MALDI	module_279
	ATATGCA,MIR-448
	V$OCT_Q6
	module_334
	RECEPTOR_SIGNALING_PROTEIN_SERINE_THREONINE_KINASE_ACTIVITY
	TTTNNANAGCYR_UNKNOWN
	V$MEF2_02
	KRCTCNNNNMANAGC_UNKNOWN
	STEMCELL_COMMON_DN
	V$IRF1_Q6
	AGED_MOUSE_RETINA_ANY_UP
	module_183
	PARK_RARALPHA_MOD
	module_345
	module_182
	V$COUP_DR1_Q6
	PROTEIN_KINASE_BINDING
	HOFFMANN_BIVSBII_BI_TABLE2
	ATCTTGC,MIR-31
	ACAACCT,MIR-453

APPENDIX D

STANDARD OPERATING PROCEDURES FOR BASELINE PREPROCESSING

For the sake of reproduction, our standard operationg procedure for performing baseline preprocessing is given below.

D.1 BASELINE PREPROCESSING

By default, the order of preprocessing steps continues as follows:

- Variance stabilization.
- Baseline correction.
- Profile normalization.
- Smoothing
- Alignment

D.1.0.1 Variance Stabilization Higher intensities typify higher-variance measures in TOF-MS datasets. We decouple the dependency of variance of measurements upon their intensity by applying the cube-root transformation to all profile readings (Hauskrecht et al., 2005).

- Change the intensity of every profile measurement to be the cube root of that measurement.

D.1.0.2 Baseline correction To ensure that intensities are measured with a baseline of 0, instrument noise is removed by a sliding-window method:

- Calculate a minimum over local window of width 200 data points to determine the baseline shift constant for the window's middle position.
- Subtract the constant from the intensity reading.
- Slide the window forward
- Continue to compare new baseline correction techniques.

D.1.0.3 Profile normalization Profile normalization is performed by TIC correction on a limited range.

- Sum intensities between the m/z range of 1500 and 16500 Daltons.
- Divide each intensity in the profile by this amount.
- Continue to compare new baseline correction techniques.

D.1.0.4 Smoothing Smoothing is accomplished by rounding intensities to fit a Gaussian shape.

- For each intensity i in profile p do:
  - Fit i to a Gaussian distribution based on its 10 immediate neighbors on either side. The Gaussian distribution has standard deviation equal to 2.
  - Move to the next intensity.
- Continue to compare the effects of smoothing.

D.1.0.5 Alignment

- Calculate the mean profile by averaging all samples in the study
- Identify peaks in the mean profile using Procedure Peak-Identification
- For all profiles p do:

  - Identify peaks of the profile p
  - Use dynamic time-warping procedure to adjust the m/z values associated with the profile p so they align with the peak in the mean profile
- Continue to compare alternative alignment strategies.

D.1.0.6 Handling technical replicates If more than one technical replicate exists for a sample, the replicates are averaged together to produce a single proteomic profile per patient.

- Average spectra that correspond to the same sample.
- Measure the effect of averaging on COV in search of variance inflation (undesirable outcome)

D.1.1 Standardized Peak Identification (Data Characterization)

D.1.1.1 Data characterization

- Ensure the following for each profile p:
  - Each profile consists of n intensity measurements
  - Each intensity measurement is assigned an m/z value.
  - Individual peaks or peak complexes may be analyzed.
- Each profile contains a class label corresponding to its disease state (typically 1 for case, and 0 for control).
- Consider various binarization approaches prior to modeling.
- Continue to consider additional (multivariate) data characterizations.

D.1.1.2 Peak identification

- Average all profiles in the dataset.
- Using a sliding window of width 12 (measurements), search for local maxima in this averaged profile.
- Local maxima are marked as peak positions.
- Transfer the positions of marked peaks to the original profiles.

- Continue to compare additional peak detection algorithms.

BIBLIOGRAPHY

[1] K. R. Coombes, J. M. Koomen, K. A. Baggerly, J. S. Morris, and R. Kobayashi. Understanding the characteristics of mass spectrometry data through the use of simulation. *Cancer Inform*, 1:41–52, 2005.

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

[3] S. L. Cohen and B. T. Chait. Influence of matrix solution conditions on the MALDI-MS analysis of peptides and proteins. *Anal. Chem.*, 68:31–37, Jan 1996.

[4] Renato Zenobi and Richard Knochenmuss. Ion formation in maldi mass spectrometry. *Mass Spectrometry Reviews*, 17(5):337–366, 1998.

[5] E. F. Petricoin, D. K. Ornstein, C. P. Paweletz, A. Ardekani, P. S. Hackett, B. A. Hitt, A. Velassco, C. Trucco, L. Wiegand, K. Wood, C. B. Simone, P. J. Levine, W. M. Linehan, M. R. Emmert-Buck, S. M. Steinberg, E. C. Kohn, and L. A. Liotta. Serum proteomic patterns for detection of prostate cancer. *J. Natl. Cancer Inst.*, 94:1576–1578, Oct 2002.

[6] KA Baggerly, SR Edmonson, JS Morris, and KR Coombes. High-resolution serum proteomic patterns for ovarian cancer detection. *Endocr Relat Cancer*, 11:583–4, Dec 2004.

[7] Emanuel F Petricoin, Ali M Ardekani, Ben A Hitt, Peter J Levine, Vincent A Fusaro, Seth M Steinberg, Gordon B Mills, Charles Simone, David A Fishman, Elise C Kohn, and Lance A Liotta. Use of proteomic patterns in serum to identify ovarian cancer. *Lancet*, 359(9306):572–577, Feb 2002.

[8] Eleftherios P Diamandis. Point: Proteomic patterns in biological fluids: do they represent the future of cancer diagnostics? *Clin Chem*, 49(8):1272–1275, Aug 2003.

[9] P. B. Yildiz, Y. Shyr, J. S. Rahman, N. R. Wardwell, L. J. Zimmerman, B. Shakhtour, W. H. Gray, S. Chen, M. Li, H. Roder, D. C. Liebler, W. L. Bigbee, J. M. Siegfried, J. L. Weissfeld, A. L. Gonzalez, M. Ninan, D. H. Johnson, D. P. Carbone, R. M. Caprioli, and P. P. Massion. Diagnostic accuracy of MALDI mass spectrometric analysis of unfractionated serum in lung cancer. *J Thorac Oncol*, 2:893–901, Oct 2007.

[10] C. H. Wu, R. Apweiler, A. Bairoch, D. A. Natale, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, R. Mazumder, C. O'Donovan, N. Redaschi, and B. Suzek. The Universal Protein Resource (UniProt): an expanding universe of protein information. *Nucleic Acids Res.*, 34:D187–191, Jan 2006.

[11] G.E.R. Box and D.R. Cox. An analysis of transformations. *Journal of Royal Statistics Society Series B*, 26:211–246, 1964.

[12] B.P. Durbin, J.S. Hardin, D.M. Hawkins, and D.M. Rocke. A variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics*, 18 Suppl 1:S105–110, 2002.

[13] W. Huber, A. von Heydebreck, H. Sltmann, A. Poustka, and M. Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18 Suppl 1:96–104, 2002.

[14] D.M. Rocke and B. Durbin. A model for measurement error for gene expression arrays. *J. Comput. Biol.*, 8:557–569, 2001.

[15] R Tibshirani. Estimating optimal transformations for regression via additivity and variance stabilization. *Journal of the American Statistical Association*, 83:394–405, June 1998.

[16] D. Bylund. *Chemometric Tools for Enhanced Performance in Liquid Chromatography-Mass Spectrometry*. PhD thesis, Uppsala University, Uppsala, Sweden, 2001.

[17] A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–1639, 1964.

[18] W. Wang, H. Zhou, H. Lin, S. Roy, T. A. Shaler, L. R. Hill, S. Norton, P. Kumar, M. Anderle, and C. H. Becker. Quantification of proteins and metabolites by mass spectrometry without isotopic labeling or spiked standards. *Anal. Chem.*, 75:4818–4826, Sep 2003.

[19] K.A. Baggerly, J.S. Morris, J. Wang, D. Gold, L.C. Xiao, and K.R. Coombes. A comprehensive approach to the analysis of matrix-assisted laser desorption/ionization-time of flight proteomics spectra from serum samples. *Proteomics*, 3:1667–1672, Sep 2003.

[20] Du P, S.M. Lin, W.A. Kibbe, and Haihui Wang. Application of Wavelet Transform to the MS-based Proteomics Data Preprocessing. *BIBE 2007*, pages 680–686, Oct 2007.

[21] K.R. Coombes, S. Tsavachidis, J.S. Morris, K.A. Baggerly, M.C. Hung, and H.M. Kuerer. Improved peak detection and quantification of mass spectrometry data acquired from surface-enhanced laser desorption and ionization by denoising spectra with the undecimated discrete wavelet transform. *Proteomics*, 5:4107–4117, Nov 2005.

[22] Zhijin Wu, Rafael A. Irizarry, Robert Gentleman, Francisco Martinez-Murillo, and Forrest Spencer. A model-based background adjustment for oligonucleotide expression arrays. *Journal of the American Statistical Association*, 99(468):909+, 2006.

[23] K.R. Coombes, H.A. Fritsche, C. Clarke, J.N. Chen, K.A. Baggerly, J.S. Morris, L.C. Xiao, M.C. Hung, and H.M. Kuerer. Quality control and peak finding for proteomics data collected from nipple aspirate fluid by surface-enhanced laser desorption and ionization. *Clin. Chem.*, 49:1615–1623, Oct 2003.

[24] M. Wagner, D. Naik, and A. Pothen. Protocols for disease classification from mass spectrometry data. *Proteomics*, 3:1692–1698, Sep 2003.

[25] M. Hauskrecht, R. Pelikan, D. E. Malehorn, W. L. Bigbee, M. T. Lotze, H. J. Zeh, D. C. Whitcomb, and J. Lyons-Weiler. Feature selection for classification of seldi–tof–ms proteomic profiles. *Appl Bioinformatics*, 4(4):227–246, 2005.

[26] I. T. Jolliffe. *Principal Component Analysis*. Springer, 1986.

[27] Keith A Baggerly, Jeffrey S Morris, and Kevin R Coombes. Reproducibility of SELDI-TOF protein patterns in serum: comparing datasets from different experiments. *Bioinformatics*, 20(5):777–785, Mar 2004.

[28] R. Tibshirani, T. Hastie, B. Narasimhan, S. Soltys, G. Shi, A. Koong, and Q. T. Le. Sample classification from protein mass spectrometry, by 'peak probability contrasts'. *Bioinformatics*, 20:3034–3044, Nov 2004.

[29] B.M. Bolstad, R.A. Irizarry, M. Astrand, and T.P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19:185–193, Jan 2003.

[30] S.J. Callister, R.C. Barry, J.N. Adkins, E.T. Johnson, W.J. Qian, B.J. Webb-Robertson, R.D. Smith, and M.S. Lipton. Normalization approaches for removing systematic biases associated with mass spectrometry and label-free proteomics. *J. Proteome Res.*, 5:277–286, Feb 2006.

[31] A. Sauve and T. Speed. Normalization, baseline correction and alignment of high-throughput mass spectrometry data. *Proceedings of Gensnips*, 2004.

[32] D. Sankoff and J. Kruskal, editors. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Center for the Study of Language and Inf, December 1999.

[33] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 1978.

[34] J. Ramsey and X. Li. Curve registration. *Journal of the Royal Statistics Society Series B*, 60:351–363, 1998.

[35] D. Bylund, R. Danielsson, G. Malmquist, and K. E. Markides. Chromatographic alignment by warping and dynamic programming as a pre-processing tool for PARAFAC modelling of liquid chromatography-mass spectrometry data. *J Chromatogr A*, 961:237–244, Jul 2002.

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

[37] Robert van den Berg, Huub Hoefsloot, Johan Westerhuis, Age Smilde, and Mariet van der Werf. Centering, scaling, and transformations: improving the biological information content of metabolomics data. *BMC Genomics*, 7(1):142, 2006.

[38] R. Pelikan, M. Lotze, J. Lyons-Weiler, D. Malehorn, and M. Hauskrecht. *Serum Proteomic Profiling and Analysis*. Elsevier, 2004.

[39] R. Pelikan, W.L. Bigbee, D. Malehorn, J. Lyons-Weiler, and M. Hauskrecht. Intersession reproducibility of mass spectrometry profiles and its effect on accuracy of multivariate classification models. *Bioinformatics*, 23:3065–3072, Nov 2007.

[40] R. Kohavi and G. John. The wrapper approach, 1998.

[41] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[42] Paul Pavlidis, Jason Weston, Jinsong Cai, and William Noble Grundy. Gene functional classification from heterogeneous data. In *RECOMB*, pages 249–255, 2001.

[43] T. R. Golub, D. K. Slonim, P. Tamayo, C. H. M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, , and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[44] T. S. Furey, N. Christianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Hauessler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

[45] V G Tusher, R Tibshirani, and G Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci U S A*, 98(9):5116–5121, Apr 2001.

[46] J.D. Storey and R. Tibshirani. *The analysis of gene expression data: methods and software*, chapter SAM thresholding and false discovery rates for detecting differential gene expression in DNA microarrays. Springer, New York, 2003.

[47] Long AD Baldi P. A bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. *Bioinformatics*, 17:509–519, 2001.

[48] W. S. Gosser. The probable error of a mean. *BIOMETRIKA*, 6:1–25, 1908.

[49] N.S. Tzannes and J.P. Noonan. The mutual information principle and applications. *Information and Control*, 22(1):1–12, February 1973.

[50] Lehmann E.L. Chernoff H. The use of maximum likelihood estimates in chi2 tests for goodness-of-fit. *The Annals of Mathematical Statistics*, 25:576–586, 1954.

[51] H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes, 1995.

[52] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, April 1982.

[53] P. Smyth and R. M. Goodman. An information theoretic approach to rule induction from databases. *IEEE Transactions on Knowledge and Data Engineering*, 4(4):301–316, August 1992.

[54] S Patel and J Lyons-Weiler. cageda: a web application for the integrated analysis of global gene expression patterns in cancer. *Appl Bioinformatics*, 3(1):49–62, 2004.

[55] P.H. Westfall and S.S. Young. *Resamplingbased multiple testing: examples and methods for p-value adjustment*. Wiley, 1993.

[56] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society B*, 57:289–300, 1995.

[57] L. Breiman. Random forests — random features. *Technical Report 567*, 1999.

[58] C. Strobl, A. L. Boulesteix, A. Zeileis, and T. Hothorn. Bias in random forest variable importance measures: illustrations, sources and a solution. *BMC Bioinformatics*, 8:25, 2007.

[59] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[60] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, December 2006.

[61] T. Hastie, R. Tibshirani, , and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[62] Fuller E. A. Krus D. J. Computer-assisted multicrossvalidation in regression analysis. *Educational and Psychological Measurement*, 42:187–193, 1982.

[63] Avrim Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

[64] S. Russel and P. Norvig. *Artificial Intelligence*. Prentice Hall, 1995.

[65] J. Koza. Survey of genetic algorithms and genetic programming, 1995.

[66] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing, 1983.

[67] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and regression trees*. Wadsworth International Group, Belmont, CA, 1984.

[68] Eric P. Xing, Michael I. Jordan, and Richard M. Karp. Feature selection for high-dimensional genomic microarray data. In *Proc. 18th International Conf. on Machine Learning*, pages 601–608. Morgan Kaufmann, San Francisco, CA, 2001.

[69] R. Neal. Assessing relevance determination methods using delve generalization in neural networks and machine learning, 1998.

[70] D. MacKay. *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology, 1992.

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

[72] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.

[73] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. *Journal of Computational Biology*, 7:559–584, 2000.

[74] D. K. Slonim, P. Tamayo, J. P. Mesirov, T. R. Golub, and E. S. Lander. Class prediction and discovery using gene expression data. *RECOMB*, 2000.

[75] N. Speer, C. Spieth, , and A. Zell. Spectral clustering gene ontology terms to group genes by function, 2005.

[76] J. McQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–97, 1967.

[77] G. Ball and D. Hall. A clustering technique for summarizing multivariate data. *Behavioral Science*, 12:153–155, 1967.

[78] R. M. Cormack. A review of classification. *J. Roy. Stat. Soc. A*, 134:321–367, 1971.

[79] M B Eisen, P T Spellman, P O Brown, and D Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A*, 95(25):14863–14868, Dec 1998.

[80] Isabelle Guyon and Andre Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 2003.

[81] F. Bach and M. Jordan. Kernel independent component analysis, 2001.

[82] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 1994.

[83] Christian Jutten and Jeanny Herault. Blind separation of sources, part 1: An adaptive algorithm based on neuromimetic architecture. *Signal Process.*, 24(1):1–10, 1991.

[84] K. C. Verhoeckx, S. Bijlsma, E. M. de Groene, R. F. Witkamp, J. van der Greef, and R. J. Rodenburg. A combination of proteomics, principal component analysis and transcriptomics is a powerful tool for the identification of biomarkers for macrophage maturation in the U937 cell line. *Proteomics*, 4:1014–1028, Apr 2004.

[85] C. Romualdi, A. Giuliani, C. Millino, B. Celegato, R. Benigni, and G. Lanfranchi. Correlation between Gene Expression and Clinical Data through Linear and Nonlinear Principal Components Analyses: Muscular Dystrophies as Case Studies. *OMICS*, Apr 2009.

[86] G. Z. Li, H. L. Bu, M. Q. Yang, X. Q. Zeng, and J. Y. Yang. Selecting subsets of newly extracted features from PCA and PLS in microarray data analysis. *BMC Genomics*, 9 Suppl 2:S24, 2008.

[87] Michael C. Denham. Implementing partial least squares. *Statistics and Computing*, 1994.

[88] T. Dijkstra. Some comments on maximum likelihood and partial least squares methods. *Journal of Econometrics*, 22:67–90, 1983.

[89] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7):674–693, 1989.

[90] Quinlan JR. *C4.5: programs for machine learning.* Morgan Kaufmann, San Francisco, CA, 1993.

[91] Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[92] T Bayes. An essay towards solving a problem in the doctrine of chances. *Philosopical Transactions of the Royal Society of London*, 53:370–418, 1763.

[93] S Russel and P Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 2002.

[94] G. Ball, S. Mian, F. Holding, R. O. Allibone, J. Lowe, S. Ali, G. Li, S. McCardle, I. O. Ellis, C. Creaser, and R. C. Rees. An integrated approach utilizing artificial neural

networks and SELDI mass spectrometry for the classification of human tumours and rapid identification of potential biomarkers. *Bioinformatics*, 18:395–404, Mar 2002.

[95] S Haykin. *Neural Networks.* Macmillian, New York, 1994.

[96] C Bishop. *Neural Networks for pattern recognition.* Oxford University Press, Oxford, 1995.

[97] S. V. N. Vishwanathan and A. J. Smola. Fast kernels on strings and trees. In nips15e, editor, *nips15*, 2002.

[98] T. Gartner, P. A. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. *Proceedings of the 16th Annual Conference on Computational Learning Theory and the 7th Kernel Workshop*, 2003.

[99] Shawkat Ali and Kate A. Smith-Miles. A meta-learning approach to automatic kernel selection for support vector machines. *Neurocomputing*, 70(1-3):173 – 186, 2006. Neural Networks - Selected Papers from the 7th Brazilian Symposium on Neural Networks (SBRN '04), 7th Brazilian Symposium on Neural Networks.

[100] J. Nahar, S. Ali, and Y. P. Chen. Microarray data classification using automatic SVM kernel selection. *DNA Cell Biol.*, 26:707–712, Oct 2007.

[101] Gert Lanckriet, Nello Cristianini, Peter Bartlett, and Laurent El Ghaoui. Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, 5:2004, 2002.

[102] Thore Graepel and Ralf Herbrich. The kernel gibbs sampler. In *NIPS*, pages 514–520, 2000.

[103] Cheng Soon Ong, Alexander J. Smola, and Robert C. Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research 3*, 2003.

[104] Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. In *Neural Information Processing Systems*, page 16. MIT Press, 2003.

[105] D.T. Ross, U. Scherf, and M.B. Eisen et al. Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, 24:227–235, 2000.

[106] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 1947.

[107] W H Kruskal and W A Wallis. Use of ranks in one-criterion variance analysis. *J. Am. Stat. Assoc*, 1952.

[108] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1994.

[109] Jos F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones, 1999.

[110] J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[111] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U.S.A.*, 102:15545–15550, Oct 2005.

[112] J. D. Tenenbaum, M. G. Walker, P. J. Utz, and A. J. Butte. Expression-based Pathway Signature Analysis (EPSA): mining publicly available microarray data for insight into human disease. *BMC Med Genomics*, 1:51, 2008.

[113] B "Efron and R." Tibshirani. On testing the significance of sets of genes. *Annals of Applied Statistics.*, 1:107129., 2007.

[114] H. M. Wain, M. J. Lush, F. Ducluzeau, V. K. Khodiyar, and S. Povey. Genew: the Human Gene Nomenclature Database, 2004 updates. *Nucleic Acids Res.*, 32:D255–257, Jan 2004.

[115] J. L. Lustgarten, V. Gopalakrishnan, H. Grover, and S. Visweswaran. Improving classification performance with discretization on biomedical datasets. *AMIA Annu Symp Proc*, pages 445–449, 2008.

[116] Roger Day and Kevin McDade. Critiquing bioinformatics resources when id mapping is needed to integrate genomic and proteomic studies. In *Proceedings of the 2010 Joint Statistical Meetings*, Vancouver, British Columbia, Canada, August 2010.

[117] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

[118] Philip Good. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer, 1994.

[119] K. Linnet. Comparison of quantitative diagnostic tests: type I error, power, and sample size. *Stat Med*, 6:147–158, Mar 1987.

[120] S. Vida. A computer program for non-parametric receiver operating characteristic analysis. *Comput Methods Programs Biomed*, 40:95–101, Jun 1993.

[121] Sofus A. Macskassy, Foster J. Provost, and Michael Littman. Confidence bands for roc curves. In *In CeDER Working Paper*, 2003.

[122] R. W. Platt, J. A. Hanley, and H. Yang. Bootstrap confidence intervals for the sensitivity of a quantitative diagnostic test. *Stat Med*, 19:313–322, Feb 2000.

[123] B. Efron and R. Tibshirani. *An Introduction to the Boostrap*. Chapman & Hall, 1993.

[124] Zhou X and Qin G. A new confidence interval for the difference between two binomial proportions of paired data. In *UW Biostatistics Working Paper Series, no. 205*, 2003.

[125] L. E. Dodd and M. S. Pepe. Partial AUC estimation and regression. *Biometrics*, 59:614–623, Sep 2003.

[126] S. Gefen, O. J. Tretiak, C. W. Piccoli, K. D. Donohue, A. P. Petropulu, P. M. Shankar, V. A. Dumane, L. Huang, M. A. Kutay, V. Genis, F. Forsberg, J. M. Reid, and B. B. Goldberg. ROC analysis of ultrasound tissue characterization classifiers for breast cancer diagnosis. *IEEE Trans Med Imaging*, 22:170–177, Feb 2003.

[127] C. Stephan, S. Wesseling, T. Schink, and K. Jung. Comparison of eight computer programs for receiver-operating characteristic analysis. *Clin. Chem.*, 49:433–439, Mar 2003.

[128] David F Ransohoff. Bias as a threat to the validity of cancer molecular-marker research. *Nat Rev Cancer*, 5(2):142–149, Feb 2005.

[129] WE Grizzle, OJ Semmes, WL Bigbee, L Zhu, G Malik, Oelschlager DK, Manne B, and Manne U. The need for review and understanding of seldi/maldi mass spectroscopy data prior to analysis. *Cancer Informatics*, 1, 2005.

[130] O John Semmes, Ziding Feng, Bao-Ling Adam, Lionel L Banez, William L Bigbee, David Campos, Lisa H Cazares, Daniel W Chan, William E Grizzle, Elzbieta Izbicka, Jacob Kagan, Gunjan Malik, Dale McLerran, Judd W Moul, Alan Partin, Premkala Prasanna, Jason Rosenzweig, Lori J Sokoll, Shiv Srivastava, Sudhir Srivastava, Ian Thompson, Manda J Welsh, Nicole White, Marcy Winget, Yutaka Yasui, Zhen Zhang, and Liu Zhu. Evaluation of serum protein profiling by surface-enhanced laser desorption/ionization time-of-flight mass spectrometry for the detection of prostate cancer: I. Assessment of platform reproducibility. *Clin Chem*, 51(1):102–112, Jan 2005.

[131] Zhen Zhang, Robert C Jr Bast, Yinhua Yu, Jinong Li, Lori J Sokoll, Alex J Rai, Jason M Rosenzweig, Bonnie Cameron, Young Y Wang, Xiao-Ying Meng, Andrew Berchuck, Carolien Van Haaften-Day, Neville F Hacker, Henk W A de Bruijn, Ate G J van der Zee, Ian J Jacobs, Eric T Fung, and Daniel W Chan. Three biomarkers identified from serum proteomic analysis for the detection of early stage ovarian cancer. *Cancer Res*, 64(16):5882–5890, Aug 2004.

[132] B Watkins, R Szaro, S Ball, T Knubovets, J Briggman, J Hlavaty, F Kusinitz, A Stieg, and Wu Y. Detection of early-stage cancer by serum protein analysis. *American Laboratory*, 33:32–36, 2001.

[133] C P Paweletz, B Trock, M Pennanen, T Tsangaris, C Magnant, L A Liotta, and E F 3rd Petricoin. Proteomic patterns of nipple aspirate fluids obtained by SELDI-TOF. *Dis Markers*, 17(4):301–307, 2001.

[134] Tatyana A Zhukov, Roy A Johanson, Alan B Cantor, Robert A Clark, and Melvyn S Tockman. Discovery of distinct protein profiles specific for lung tumors and pre-malignant lung lesions by SELDI mass spectrometry. *Lung Cancer*, 40(3):267–279, Jun 2003.

[135] Evelyn Zeindl-Eberhart, Sibylle Haraida, Sibylle Liebmann, Peter Roman Jungblut, Stephanie Lamer, Doris Mayer, Gundula Jager, Stephen Chung, and Hartmut Manfred Rabes. Detection and identification of tumor-associated protein variants in human hepatocellular carcinomas. *Hepatology*, 39(2):540–549, Feb 2004.

[136] N Barbarini, P Magni, and R Bellazzi. A new approach for the analysis of mass spectrometry data for biomarker discovery. *AMIA Annu Symp Proc*, pages 26–30, 2006.

[137] Leo McHugh and Jonathan W. Arthur. Computational methods for protein identification from mass spectrometry data. *PLoS computational biology*, 4(2), February 2008.

[138] Kolker E, Higdon R, and Hogan JM. Protein identification and expression analysis using mass spectrometry. *Trends Microbiol*, 14:229–35, May 2006.

[139] C. Huttenhower and O.G. Troyanskaya. Bayesian data integration: a functional perspective. *Comput Syst Bioinformatics Conf*, pages 341–351, 2006.

[140] M. Kanehisa, S. Goto, S. Kawashima, and A. Nakaya. The KEGG databases at GenomeNet. *Nucleic Acids Res.*, 30:42–46, Jan 2002.

[141] I. Lee, S.V. Date, A.T. Adai, and E.M. Marcotte. A probabilistic functional network of yeast genes. *Science*, 306:1555–1558, Nov 2004.

[142] A.J. Butte and I.S. Kohane. Creation and implications of a phenome-genome network. *Nat. Biotechnol.*, 24:55–62, Jan 2006.

[143] D.L. Wheeler, T. Barrett, D.A. Benson, S.H. Bryant, K. Canese, V. Chetvernin, D.M. Church, M. Dicuccio, R. Edgar, S. Federhen, M. Feolo, L.Y. Geer, W. Helmberg, Y. Kapustin, O. Khovayko, D. Landsman, D.J. Lipman, T.L. Madden, D.R. Maglott, V. Miller, J. Ostell, K.D. Pruitt, G.D. Schuler, M. Shumway, E. Sequeira, S.T. Sherry, K. Sirotkin, A. Souvorov, G. Starchenko, R.L. Tatusov, T.A. Tatusova, L. Wagner, and E. Yaschenko. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, 36:13–21, Jan 2008.

[144] O. Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Res*, 32(Database issue), January 2004.

[145] J. Lyons-Weiler, R. Pelikan, H.J. III Zeh, D.C. Whitcomb, D.E. Malehorn, W.L. Bigbeee, and M. Hauskrecht. Assessing the statistical significance of the achieved classification error of classifiers constructed using serum peptide profiles, and a prescription

for random sampling repeated studies for massive high-throughput genomic and proteomic studies. *Cancer Informatics*, 1:53–77, Feb 2005.

[146] Hauskrecht Milos and Pelikan R.". Inter-session reproducibility measures for high-throughput data sources. *Proceedings of the AMIA Summit on Translational Bioinformatics*, March 2008.

[147] M. Hauskrecht, R. Pelikan, M. Valko, and J. Lyons-Weiler. *Feature Selection and Dimensionality Reduction in Genomics and Proteomics*. Springer, 2007.

[148] Bradley Efron. *The Jackknife, the Bootstrap, and Other Resampling Plans*. Society for Industrial & Applied Mathematics, January 1987.

[149] N Leigh Anderson and Norman G Anderson. The human plasma proteome: history, character, and diagnostic prospects. *Mol Cell Proteomics*, 1(11):845–867, Nov 2002.

[150] H H Rasmussen, T F Orntoft, H Wolf, and J E Celis. Towards a comprehensive database of proteins from the urine of patients with bladder cancer. *J Urol*, 155(6):2113–2119, Jun 1996.

[151] Shen Hu, Yongming Xie, Prasanna Ramachandran, Rachel R Ogorzalek Loo, Yang Li, Joseph A Loo, and David T Wong. Large-scale identification of proteins in human salivary proteome by LCMS and 2DGE–MS. *Proteomics*, 5(6):1714–1728, Apr 2005.

[152] Alexey I Nesvizhskii and Ruedi Aebersold. Interpretation of shotgun proteomic data: the protein inference problem. *Mol Cell Proteomics*, 4(10):1419–1440, Oct 2005.

[153] Haixu Tang, Randy J Arnold, Pedro Alves, Zhiyin Xun, David E Clemmer, Milos V Novotny, James P Reilly, and Predrag Radivojac. A computational approach toward label-free protein quantification using predicted peptide detectability. *Bioinformatics*, 22(14):481–488, Jul 2006.

[154] M.R. Flory, T.J. Griffin, D. Martin, and R. Aebersold. Advances in quantitative proteomics using stable isotope tags. *Trends Biotechnol.*, 20:S23–29, Dec 2002.

[155] Peng Lu, Christine Vogel, Rong Wang, Xin Yao, and Edward M Marcotte. Absolute protein expression profiling estimates the relative contributions of transcriptional and translational regulation. *Nat Biotechnol*, 25(1):117–124, Jan 2007.

[156] C. Nadeau and Y. Bengio. Inference for the generalization error. *Machine Learning*, pages 239–281, 2003.

[157] Mark D. Schuchard, Christopher D. Melm, Angela S. Crawford, Holly A. Chapman, Steven L. Cockrill, Kevin B. Ray, Richard J. Mehigh, William K. Kappel, , and Graham B. I. Scott. Immunoaffinity depletion of 20 high abundance human plasma proteins. *Origins*, 21, December 2005.

[158] J Davis and M Goadrich. The relationship between precision-recall and roc curves. *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[159] CJ Van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.

[160] J. L. Lustgarten, C. Kimmel, H. Ryberg, and W. Hogan. EPO-KB: a searchable knowledge base of biomarker to protein links. *Bioinformatics*, 24:1418–1419, Jun 2008.

[161] G. C. Tseng, M. K. Oh, L. Rohlin, J. C. Liao, and W. H. Wong. Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects. *Nucleic Acids Res.*, 29:2549–2557, Jun 2001.

[162] Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res.*, 30:e15, Feb 2002.

[163] Marianna Zahurak, Giovanni Parmigiani, Wayne Yu, Robert Scharpf, David Berman, Edward Schaeffer, Shabana Shabbeer, and Leslie Cope. Pre-processing agilent microarray data. *BMC Bioinformatics*, 8(1):142, 2007.

[164] G Smyth, Y Yang, and T Speed. Statistical issues in cdna microarray data analysis. *Methods in Molecular Biology*, 224:111–136, 2003.

[165] RA Irizarry, B Hobbs, F Collin, YD Beazer-Barclay, KJ Antonellis, U Scherf, and TP Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4:249–264, 2003.

[166] J. Hubble, J. Demeter, H. Jin, M. Mao, M. Nitzberg, T. B. Reddy, F. Wymore, Z. K. Zachariah, G. Sherlock, and C. A. Ball. Implementation of GenePattern within the Stanford Microarray Database. *Nucleic Acids Res.*, 37:898–901, Jan 2009.

[167] K. J. Kim, K. H. Lee, B. Kim, T. Richter, I. D. Yun, S. U. Lee, K. T. Bae, and H. Shim. JPEG2000 2D and 3D Reversible Compressions of Thin-Section Chest CT Images: Improving Compressibility by Increasing Data Redundancy Outside the Body Region. *Radiology*, 259:271–277, Apr 2011.

[168] A. Makarov, E. Denisov, A. Kholomeev, W. Balschun, O. Lange, K. Strupat, and S. Horning. Performance evaluation of a hybrid linear ion trap/orbitrap mass spectrometer. *Anal. Chem.*, 78:2113–2120, Apr 2006.

[169] Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *In ICML*, 2004.

[170] A. Dixit, L. Yi, R. Gowthaman, A. Torkamani, N. J. Schork, and G. M. Verkhivker. Sequence and structure signatures of cancer mutation hotspots in protein kinases. *PLoS ONE*, 4:e7485, 2009.

[171] A. Torkamani and N. J. Schork. Prediction of cancer driver mutations in protein kinases. *Cancer Res.*, 68:1675–1682, Mar 2008.

[172] A. Torkamani and N. J. Schork. Identification of rare cancer driver mutations by network reconstruction. *Genome Res.*, 19:1570–1578, Sep 2009.

[173] Csar Costa Vera, Roman Zubarev, Hanno Ehring, Per Hakansson, and Bo. U. R. Sunqvist. A three-point calibration procedure for matrix-assisted laser desorption/ionization mass spectrometry utilizing multiply charged ions and their mean initial velocities. *Rapid Communications in Mass Spectrometry*, 10(12):1429–1432, 1996.