Systematic approaches to integrate inconsistent, noisy high-throughput data to bolster subtle relationships obscured by standard analyses

Jennifer M. Staab

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2012

Approved by:

Shawn M. Gomez

Wei Wang

Jan F. Prins

Leonard McMillan

Thomas M. O'Connell

Abstract

JENNIFER M. STAAB: Systematic approaches to integrate inconsistent, noisy high-throughput data to bolster subtle relationships obscured by standard analyses. (Under the direction of Shawn M. Gomez.)

The increasing availability and decreasing cost of high throughput technologies coupled with the availability of computational tools form a basis for a shift to a more integrated approach to analyzing biological processes. In particular, classical statistical analysis techniques are designed to analyze data characterized by a single data source and are distinguished by a much higher ratio of subjects to the number of observations. In contrast, bioinformatics and systems biology applications often involve large data sets characterized by an abundance of observations spawned from a relatively small sample of subjects. The complexity of these systems coupled with the need to integrate inconsistent (noisy) data require appropriate methodologies that address these issues.

Standard analyses can proficiently identify associations within consistent data, but these approaches are not robust at identifying relationships across data sources and/or where nontrivial amounts of inconsistency (noise) are present. Such data requires approaches that account for this increasing inconsistency within the data. One technique of accounting for such inconsistency is to limit analyses to subsets of data where the desired associations are the most prominent. Challenges for this particular approach involve the determination of subsets of interest while simultaneously establishing a metric with which to judge statistical importance.

My initial work using this approach involved providing a methodology to represent Nuclear Magnetic Resonance (NMR) Spectra as hundreds of aligned peaks as opposed to thousands of unaligned points, which allows for more sophisticated means of analysis. My later work explores the development of data mining methodologies for identifying associations that exist within subsets of inconsistent, noisy data while addressing how to sensibly target subsets of interest while establishing a metric of association that provides statistical significance. Two approaches were developed, the first of which established a p-value associated metric, while the latter allowed for multiple arbitrary metrics of interest to be used to identify statistically significant patterns. This work helps to establish methodologies for the identification of rare, but significant patterns in large noisy data sets.

Acknowledgments

My gratitude goes out to my advisor Dr. Shawn Gomez for his guidance and support throughout the course of my dissertation research. I am also grateful to Dr. Thomas O'Connell for his invaluable guidance and collaboration within the field of metabolomics and with PCANS. I am thankful to the rest of my committee, Dr. Wei Wang, Dr. Jan Prins and Dr. Leonard McMillan, for the feedback and advice they provided regarding my research and defense.

I am grateful to my longtime friend, officemate, and colleague, Kwangbom Choi, whose counsel and camaraderie throughout our graduate careers has proved to be an invaluable resource. I am thankful to the members of the Gomez Lab, Matt Berginski, Alicia Midland, Janet Doolittle and Ke Xu, whom have provided valuable friendship, feedback, and support throughout our shared lab experience. I am also thankful to my fellow bioinformatics and computational biology colleagues past and present whom have provided support and encouragement during my graduate career.

I am most grateful to my family and friends whose encouragement and support over my long graduate career have made this achievement possible. My parents for their unwavering and unconditional support, despite not fully understanding what I was working on and why it was important to me. My sister whom has always been there at the ready with advice, encouragement, and veterinary assistance at all hours of the day. My brother and nephews whom have provided the much needed extracurricular breaks to view hockey games. I am thankful to Scott and the Reidsville cycling crew whom have provided endless hours of extracurricular distraction via cycling and encouragement of my academic pursuits. I would also like to thank my longtime friends, Marisa, Jiten, and Trang, although not directly associated to academia they have provided much needed encouragement and support over the years. And a special thanks to Jim, for the much needed late night laughs, friendship, encouragement, and real world advice.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Goals

The increasing availability and decreasing cost of high-throughput (HT) technologies coupled with the availability of computational tools and data form a basis for a shift to a more integrated approach in analyzing biological processes. Classical statistical analysis techniques were designed to analyze data characterized by a single data source distinguished by a much higher ratio of subjects in comparison to the number of observations arising from each subject. In contrast, bioinformatics and systems biology often involve high-throughput data characterized by an abundance of observations spawned from a relatively small sample of subjects. Additionally, the complexity of these systems under analysis coupled with the need to integrate inconsistent (noisy) data often violates many of the assumptions of classical analytic techniques. My primary focus has been based upon the identification of relationships amongst noisy, inconsistent data within the context of providing a more integrated approach to analyzing biological processes. The approaches I developed identify subsets of data that maintain robust analytic relationships obscured by the standard methodologies.

My initial work was within the field of metabolomics and focused on providing a digital data representation through automated alignment of Nuclear Magnetic Resonance (NMR) Spectra. The longer-term goal of this work was to open up new avenues for analysis and

integration of metabolomic data and aid their incorporation into larger integrative analysis frameworks. Specifically, the transformation of the spectrum representation from points to peaks which reduces the inconsistency within a spectrum by focusing directly on the component of analysis. The algorithm reduces each spectrum from thousands of points to hundreds of consistent peaks for final analysis. Moreover, the automated alignment of the NMR spectra served as a means of further noise reduction, increasing the likelihood that the peaks within each spectrum would be fruitful with regards to the final result. The noise reduction provided by this transformation and alignment process greatly simplified data complexity and enabled further application of other means of statistical analysis of NMR spectra.

From this, my focus shifted to developing data mining methods to identify relationships that exist within subsets of inconsistent, imperfect data. My research deliberately focused upon data where traditional means of analysis proved to be futile, to identify association between response and explanatory variables as data sources are integrated over a common set of subjects. Specifically, the toxicological associations between animal study endpoints (response variables) and high-throughput/high-content bioassays (explanatory variables) as perturbed by the same potentially toxic chemicals (subjects). The methods I employed use pattern identification approaches to identify subsets of potentially toxic chemicals that perturbed sets of animal endpoints and bioassays in a consistent manner. These methods have been enhanced to allow for the incorporation of user-defined amounts of fuzziness into the results and to enable the identification of statistically significant results based upon user defined metrics (no p-value required). Furthermore, the methods can be employed upon larger, more dense datasets through targeted analysis and can be used in the integration of three or more datasets.

1.2 Brief Overview of Existing Methods

1.2.1 NMR Spectra Noise Reduction Methods

As discussed in detail in Chapter 2 within the field of metabolomics, the standard way to reduce the noise in spectra prior to analysis is through binning, a procedure that involves dividing the spectra into small windows and taking the area under the curve for each window as the final intensity (Gartland et al., 1991; Anthony et al., 1994). Ideally, these windows will be large enough to encompass peak drift and to reduce the number of points that represent a spectrum, but not so large as to include many peaks in a single bin. The latter consequence is unavoidable in crowded spectra and thus there is the potential for significant loss of information when binning, for example by including peaks belonging to multiple compounds within a single bin. Alternatives to binning typically involve some form of peak alignment procedure. Several algorithms have also been recently developed to align peaks in sets of NMR spectra Wu et al. (2006); Kim et al. (2006); Torgrip et al. (2003); Veselkov et al. (2009); Savorani et al. (2010).

Current advanced NMR alignment methods such as fuzzy warping (Wu et al., 2006), Bayesian alignment (Kim et al., 2006), Recursive Segment-Wise Peak Alignment (Veselkov et al., 2009), peak alignment by FFT (Wong et al., 2005; Savorani et al., 2010) and peak alignment using reduced set mapping without recursive target update (Torgrip et al., 2003), are based on the use of a template spectrum to help align a set of spectra. Choosing a template typically involves either selecting a single sample spectrum that appears most like the others as determined by some measure of similarity, creating an ”average” spectrum, or by choosing a reference spectrum not contained within the sample. All remaining sample spectra are then aligned to this selected template using some form of pairwise alignment algorithm. A significant problem with the template approach is that there can be a great amount of variability between any two spectra. Part of this difference arises due to the previously described chemical shift variation. In addition, significant differences arise due to the existence of disparate

groups within the data; for instance, inter-group variation between control and treated groups, subpopulation differences within these groups, etc. There may often be a priori knowledge of general subgroups, but one of the goals of metabolomics is to discover new subgroups such as different types of responders in drug or toxicity studies; by definition, templates for such groups are not known beforehand. Thus in such cases, the use of a template can significantly complicate downstream analyses. Further discussion of existing methodologies and comparison of these methodologies to our own can be found in Chapter 2.

1.2.2 Identifying Association in Inconsistent, Noisy Data

Clustering is a fundamental method of unsupervised learning that partitions data in a way as to highlight meaningful relationships by exploring how data groups based upon similarity. Given a two-dimension data matrix, 2-D hierarchical clustering can be used to consider both columns and rows of the data when looking for meaningful relationships within the data. 2-D hierarchical clustering is not ideal in inconsistent, noisy data because the methodology considers the entire record (all the data in a given row and for a given column) when partitioning the data into meaningful groups. Similarly to 2-D hierarchical clustering, biclustering is able concurrently partition data by both rows and columns. Unlike 2-D hierarchical clustering, biclustering is able to consider submatrices, or subsets of the data; thus, using biclustering is a better method than hierarchical clustering to identify meaningful relationships within inconsistent data. Computationally, biclustering works best on sparse data matrices or when heuristics are used to limit the exhaustive enumeration of all possible submatrices. This is because biclustering solutions employ algorithms with computational complexity of NP-complete, meaning they have no known polynomial time algorithms and in the worst case their runtimes are exponential. van Uitert et al. (2008) demonstrate the use of biclustering on high-throughput data when they employ their method of biclustering on sparse binary genomic data to identify interacting transcription factors. Another example is DiMaggio et al. (2010) use of biclustering on inconsistent data to explore the use of logistic regression to identify predictive association

between sets of explanatory variables and a response variables. Methods of biclustering most directly compare to our methodology because they focus upon analysis of subsets of the data.

Other methods of determining association across multiple datasets with inconsistent data typically involve a bayesian framework. Specifically, these methods tend to weight the data based on its usefulness in the underlying mathematical model of association as was demonstrated by Webb-Robertson et al. (2009) using metabolomic data. The primary motivation of the study by DiMaggio et al. (2010) was to identify relationships between explanatory and response variables that could be used in prediction; whereas, the motivation of the Webb-Robertson et al. (2009) methodology was to identify relationships that provided the most significant differences between classes based upon integrated metabolomic data. Zhang et al. has developed data mining methods to identify significant relationships that existed between sets of explanatory and response variables for categorical data (Zhang et al., 2010b,a). Similarly, van Uitert et al. (2008) developed a method that was used to determine an association between two sets of genomic data to identify clusters with novel associations between the datasets. Although not focused on the relationship between explanatory and response variables, Reif et al. (2010) developed a measure that integrates multiple sources of toxicological data together to prioritize toxicological risk. Unlike DiMaggio et al., the methodology of Zhang et al. is able to integrate together the search for relationships with significance testing of discovered relationships. DiMaggio and Webb-Robertson both use methods that are more suited for integrating data from multiple data sources where a high degree of inconsistency (noise) exists between the data sources. Additionally DiMaggio, Webb-Robertson, and Reif's methodologies are more suitable for handling numeric data as compared to the methods that Zhang et al. employ which involve pairwise association between categorical data. The methodology of van Uitert et al. addresses some degree of inconsistency within the data, but unlike the other methods, its primary goal is the discovery of novel associations identified through integration with little regard for finding all associations or assigning statistical significance to the results. Similarly to van Uitert et al., Reif's methodology does not provide statistical significance to

indicate the importance of its results. However, their methodology does provide a ranking of toxicological risk based upon multiple data sources. As discussed above there are multiple methods of integrating inconsistent data, but the biclustering methodology (like van Uitert et al. (2008)) is most similar to our methods because they both focus upon analysis of subsets of data to deal with inconsistency.

1.3 Approach and Innovations

1.3.1 Methods to Enhance NMR Spectra Analysis

Our novel approach for the alignment of NMR spectra is based on the creation of a consensus spectrum alignment through integration of pairwise spectrum comparisons (referred to as PCANS hereafter - Progressive Consensus Alignment of Nmr Spectra). To our knowledge, this is the first such consensus approach applied to the alignment of NMR spectra and the only approach that transforms spectra from points to peaks prior to alignment as opposed to using the entire spectrum. This approach has several advantages that include the ability to align spectra with significant amounts of noise in chemical shift position, peak height and peak width. By using peaks as the basis for alignment we maintain the maximally informative set of information existing within a set of spectra. As a result, the existence of subgroups within a set of spectra can be identified since group-specific peaks are maintained in the final alignment.

We characterize the performance of this approach by aligning simulated NMR spectra which have been provided with user-defined amounts of chemical shift variation as well as inter-group differences as would be observed in control-treatment applications. Moreover, we demonstrate how our method provides better performance than either a template-based alignment or binning. Finally, we further evaluate this approach in the alignment of real mouse urine spectra and demonstrate its ability to improve downstream statistical analyses such as PCA and OPLS models commonly used in metabolomics analyses.

1.3.2 Methods to Enhance Association Identification

The data mining methods implemented focus on data where traditional methods of predictive modeling failed to identify useful relationships because they considered the entire data record. In contrast our approach, similar to biclustering, identifies relationships amongst subsets of the data. Our methods differ from the biclustering and prediction scheme of DiMaggio et al. (2010) by allowing one to incorporate group identification and association in a more streamlined framework. Moreover, our methods exhaustively explore the inclusion of multiple response variables with regards to association with the explanatory variables, while the work of DiMaggio et al. considers each response variable separately. Our methods more fully explore all possible enumerations of the subsets of data that specifically support the desired association; in our case the association between response and explanatory variables. Our methods are more similar to those employed by Zhang et al. (2010b,a) with regards to incorporating association finding and significance into a streamlined framework. However, unlike Zhang, our methods focus on subsets of the data (Zhang et al., 2010b,a). While our algorithms are similar to the methodology of van Uitert et al. (2008) as in they are applied to sparse inconsistent binary data; they differ from this work in that they provide a measure of statistical significance for the results. Additionally they provide the full complement of results for a given threshold, and can be modified to integrate more than a pair of datasets. Our methods differ from all three (Zhang, DiMaggio, Webb-Robertson) by allowing one to incorporate fuzziness (allowable zeros) given specific restrictions (described later). Finally, our algorithm is able to be applied to the mining of larger datasets by constraining the search space through requiring a minimum number of pre-specified features in the output through the use of seed nodes.

1.3.3 Thesis

Classical statistical analyses are not robust in identifying relationships within data in the presence of inconsistencies or noise. By focusing analysis on subsets of data with internal

consistency, I develop methods that show improved identification of relationships as evidenced by the relevance of the generated results.

The methods I develop focus on two areas of research, NMR spectra analysis and data mining for association within inconsistent data. The contributions to improving NMR spectral analysis are discussed first. The data mining for association follows because these methods can be directly applied to NMR spectral analysis to improve the relevance of the results.

1.3.4 Contributions to Enhance NMR Spectra Analysis

To address these problems of inconsistency between NMR spectra when performing metabolomic type analysis, our methods transform and align the peaks of each spectrum. This reduces the analysis to a small subset of well-aligned aligned peaks as opposed to attempting to quantify and analyze all the unaligned points of each spectrum. Treating each spectrum as hundreds of aligned peaks as opposed to thousands of unaligned points enhances the analysis we can perform and enables us to use more sophisticated means of analysis as is discussed in detail in Chapter 2.

Innovations made with regards to NMR spectra analysis are the following:

- Spectra are transformed (subset) to a collection of peaks with properties of location, height and width instead of a collection of points
  - Reduces spectrum to relevant information
  - Reduces complexity of alignment and analysis
  - Reduction allows for more sophisticated analysis
- Alignment algorithm that employs consensus as opposed to template alignment
  - Improves quality of alignment by preventing misalignment of peaks not found within the template
  - Consensus alignment can be incorporated into any pairwise alignment scheme

  - Removes need to identify all peaks within sample spectra for template formation
  - Same amount of computation as template alignment when coupled with pairwise alignment schemes

1.3.5 Contributions to Enhance Association Identification

With the integration of datasets over a common set of observations, inconsistencies are addressed by identifying subsets of data that most strongly support the desired association between the datasets. For this problem in particular, the methods developed focus on deficiencies in current methodologies by identifying consistent relationships within noisy data. Once these subsets are identified, the methodology establishes a statistical framework under which the significance of the subsets can be ranked and their strength of association can be determined. This approach of exploring relationships within subsets of the data is meant to be used when traditional means of analysis fail to produce adequate results due to inconsistency within the data. Furthermore, this methodology is meant to be used as an exploratory tool to find underlying relationships that were obscured with traditional means of analysis.

Innovations made with regards to the discovery and prioritization of subsets:

- Determination of Subsets with Closed/Approximate Itemset Mining
  - Means to target analysis on certain relationships with use of *seed nodes*
    - Full enumeration of desired relationships based upon frequency criterion
    - Exploration of larger, more dense data through targeted analysis
    - Ability to focus analysis on multivariate associations (2+ response variables)
  - Incorporation of Fuzziness into subsets
    - For larger, more dense datasets
    - Use of statistic to provide relevance of results
- Establish Metric of Importance

- Strength of association and statistical relevance
    - Phi Coefficient used to rank and provide statistical relevance for closed / approximate itemsets applied to identify association between explanatory and response variables
    - Established techniques to enable use of bootstrap methodology on larger datasets with higher support thresholds to facilitate use of *any* metric to quantify association
- Integration of 3+ Datasets with bootstrap methodology's use of multiple metrics

1.4 Dissertation Outline

In chapter 2, I describe my initial work on PCANS in the field of metabolomics in detail. Beginning with background and motivation, describing the methodology, results on simulated and real data, our conclusions and future directions. Chapters 3 through 5 focus on my work developing data mining techniques to mine for association with inconsistent, noisy datasets. Chapter 3 describes in detail the background and related work. Chapter 4 describes using closed/approximate itemset mining in conjunction with the phi coefficient to discover significant subsets within data. Chapter 5 describes in detail mining for association with a bootstrap methodology where statistical significance is no longer dependent upon a metric with an associated p-value. Chapter 6 presents the conclusions and future directions of my thesis research.

Chapter 2

Enhancing Metabolomic Analysis with PCANS

2.1 Background

Continuing technological advances are providing rich data sets quantifying an increasingly broad range of biological processes. Obvious examples include the use of microarrays for the quantification of mRNA levels and mass spectroscopy for the identification of protein states and their interactions. Coinciding with these technological developments are computational approaches for the extraction, organization and analysis of these data. The application of improved experimental methods in combination with tailored computational approaches is providing a major driving force in the development of a more global, systems perspective of biological function and disease.

Metabolomics, also referred to as metabonomics, similarly provides a comprehensive picture of biological function by focusing on quantitative measurement of metabolites in biological fluids, cells or tissues (Nicholson et al., 1999; Robertson, 2005). The two major analytical platforms used in metabolomics are nuclear magnetic resonance (NMR) spectroscopy and mass spectrometry (MS), the latter typically being preceded by either liquid or gas chromatography (LC/MS and GC/MS respectively). The ultimate goal of these methods is to extract accurate and quantitative information as to the identity of detected metabolites. Increasingly common in metabolomic studies is the analysis of a large number of samples, where the result-

ing data is analyzed using multivariate methods such as principal components analysis (PCA). Such analyses typically require significant preprocessing of the data. In particular, it is imperative that signals for a given compound appear at the same location in all spectra. Signal locations can vary significantly, however, as in the case of LC/MS where small deviations in the chromatographic retention time can arise from variation in instrumental parameters such as flow rate, gradient slope and temperature. In NMR spectra, the peak location can vary due to differences in pH, ion content and the concentration of metabolites. For both of these methods, this variability has to be overcome in order to provide a consistent set of spectra for analysis.

The most common method of addressing variability across spectra is through binning, a procedure that involves dividing the spectra into small windows and taking the area under the curve for each window as the final intensity (Gartland et al., 1991; Anthony et al., 1994). Ideally, these windows will be large enough to encompass the peak drift, but not so large as to include many peaks in a single bin. The latter consequence is unavoidable in crowded spectra and thus there is the potential for significant loss of information when binning, for example by including peaks belonging to multiple compounds within a single bin. Alternatives to binning typically involve some form of peak alignment procedure. For LC/MS methods, a number of algorithms have been developed to align similar peaks across a set of chromatograms (e.g. (Wong et al., 2005) and recently reviewed in (America and Cordewener, 2008)). Similarly, several algorithms have also been recently developed to align peaks in sets of NMR spectra (Wu et al., 2006; Kim et al., 2006; Torgrip et al., 2003; Veselkov et al., 2009; Savorani et al., 2010). In this paper we describe a novel peak alignment method for NMR that is specifically tailored to the demands of large and disparate metabolomics datasets.

Current advanced NMR alignment methods such as fuzzy warping (Wu et al., 2006), Bayesian alignment (Kim et al., 2006), Recursive Segment-Wise Peak Alignment (Veselkov et al., 2009), peak alignment by FFT (Wong et al., 2005; Savorani et al., 2010) and peak alignment using reduced set mapping without recursive target update (Torgrip et al., 2003), are

based on the use of a template spectrum to help align a set of spectra. Choosing a template typically involves either selecting a single sample spectrum that appears most like the others as determined by some measure of similarity, creating an "average" spectrum, or by choosing a reference spectrum not contained within the sample. All remaining sample spectra are then aligned to this selected template using some form of pairwise alignment algorithm. A significant problem with the template approach is that there can be a great amount of variability between any two spectra. Part of this difference arises due to the previously described chemical shift variation. In addition, significant differences arise due to the existence of disparate groups within the data; for instance, inter-group variation between control and treated groups, subpopulation differences within these groups, etc. There may often be a priori knowledge of general subgroups, but one of the goals of metabolomics is to discover new subgroups such as different types of responders in drug or toxicity studies; by definition, templates for such groups are not known beforehand. Thus in such cases, the use of a template can significantly complicate downstream analyses.

Here we describe a novel approach for the alignment of NMR spectra that is based on the creation of a consensus spectrum alignment through integration of pairwise spectrum comparisons and referred to as PCANS hereafter (Progressive Consensus Alignment of Nmr Spectra). To our knowledge, this is the first such consensus approach applied to the alignment of NMR spectra. This approach has several advantages that include the ability to align spectra with significant amounts of noise in chemical shift position, peak height and peak width. By using peaks as the basis for alignment we maintain the maximally informative set of information existing within a set of spectra. As a result, the existence of subgroups within a set of spectra can be identified since group-specific peaks are maintained in the final alignment.

We characterize the performance of this approach by aligning simulated NMR spectra which have been provided with user-defined amounts of chemical shift variation as well as inter-group differences as would be observed in control-treatment applications. Moreover, we demonstrate how our method provides better performance than either a template-based

alignment or binning. Finally, we further evaluate this approach in the alignment of real mouse urine spectra and demonstrate its ability to improve downstream statistical analyses such as PCA and OPLS models commonly used in metabolomics analyses.

2.2 Methodology

2.2.1 Experimental NMR data collection and processing

Complete details on the urine collection and sample preparation are given in (Bradford et al., 2008). Briefly, the samples consisted of 540 μl of urine plus 60 μl of a D_2O solution containing 5mM trimethylsilylpropionate-d4 (TSP) as a concentration and chemical shift reference. The solutions were transferred to 5 mm NMR tubes and NMR spectra were acquired on a Varian Inova 400 MHz spectrometer using a 5 mm pulsed field gradient, inverse detection probe (Varian, Inc., Palo Alto, CA). The spectra were acquired with 1024 transients and a sweep width of 4650Hz digitized with 16384 points. The pulse sequence included a 4 second solvent presaturation period and a 2.6 second acquisition time. A 45 degree excitation pulse was used to provide quantitative results.

The data were processed using ACD software version 9 (Advanced Chemistry Development, Toronto, Canada). A 0.1Hz exponential line broadening was applied to the data. The spectra were phased and baseline corrected using a 6th order polynomial fitting algorithm implemented in the software. The spectra were normalized to the integral for the TSP peak. The digitized spectra were exported as text files for subsequent peak picking prior to alignment with the PCANS method. Spectral binning was carried out by dividing the spectrum into uniform 0.04 ppm bin windows and taking the integral value as the sum of the intensities of all peaks in that bin. The regions from 0.5 to 4.7 ppm and 4.9 to 9.5 were included in the integration. The regions below 0.5 and above 9.5 contained only noise and the region from 4.7 to 4.9 contained the residual solvent peak.

2.2.2 Multivariate statistical analysis

The statistical analyses were performed using SimcaP+ version 11.5 (Umetrics, Umea, Sweden). Pareto scaling was applied to both the peak picked and binned NMR data prior to principal component analysis (PCA) and orthogonal projection to latent structures discriminant analysis (OPLS-DA) (Wiklund et al., 2008; Cloarec et al., 2005).

2.2.3 Peak picking

The simulated spectra were based upon peaks manually chosen from an actual urine spectrum. Peaks were chosen such that they contained a range of features typically found within normal spectra including tall and small peaks, clusters of many closely spaced peaks, doublets, etc. These peaks and their associated chemical shift position, height and width were then used as the basic material from which simulated spectra (peak profiles) were generated. Thus, the peaks used for simulation are a representative subset of the original spectrum and the simulation program uses sets of these along with defined amounts and types of variation to generate the simulated profiles. For simulated spectra, the steps responsible for peak detection and peak attribute assignment are skipped since the simulated spectra are already defined with a set of peaks with associated attributes.

For real NMR spectra, the peak detection algorithm uses the derivative of the spectrum to detect and define potential peaks. Potential peaks must have a zero first derivative, a negative second derivative and be composed of at least 8 points, where points here refer to data values from the digitized raw spectra. In addition, we use the number of points that define a peak as well as peak height relative to neighbors to determine 'real' peaks from noise peaks within a given spectrum. Specifically, for each potential peak, we look in a region centered around this point (151 points in this work) and call this a 'true' peak if it exceeds a user defined height. This threshold height is based on the height of surrounding points within this region. In this work a true peak had to have a height greater than 70% of the surrounding points. The resulting peaks that define each spectrum are characterized by the attributes of relative intensity (height)

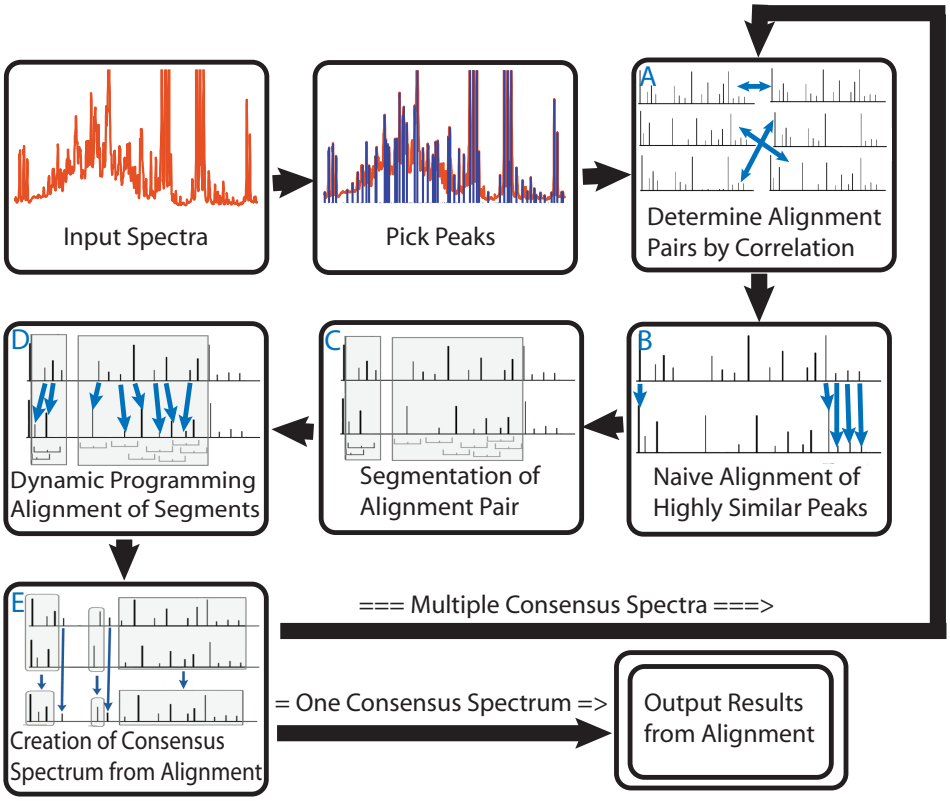


Figure 2.1: Overview of the PCANS Alignment Process. The alignment process loops through multiple iterations of pairwise alignment until achieving a single consensus profile. See text for further details.

at peak apex, chemical shift position at peak apex, and the width at half-height of the peak. The width at half-height is calculated by fitting a triangle to the peak based on the points prior and immediately after the apex point. The base of the triangle estimates the width at half height for the peak. The peaks from NMR spectra are not perfect Lorentzians because multiple compounds compose a sample; therefore, relative intensity (height) and width at half-height are not redundant information.

The overall flow of the PCANS alignment algorithm is diagrammed in Figure 2.1. Detailed algorithm pseudocode for both naive and dynamic programming alignment is provided in the next section . After peak detection, the remaining alignment steps are the same for both real and simulated spectra. The process begins with highly similar pairs of spectra being identified using synchronous sample−sample correlation (Figure 2.1A) (Sasic et al., 2000).

We note that the statistical correlation between spectra will be more influenced by the larger peaks, but this is simply a starting point in choosing which spectra to try and align first into a consensus spectrum; all peaks will ultimately be aligned. Once the alignment pairs have been identified, the pairwise alignment process begins with naive peak alignment as illustrated in Figure 2.1B. The naive peak alignment algorithm aligns corresponding peaks within the pair that have ninety percent or greater similarity across all peak attributes. Doing so also generates unaligned regions that are often bounded on both sides by regions composed solely of these highly similar (and easily alignable) peaks.

In both the naive as well as dynamic programing alignment, described next, crossover of peaks is prevented. Here, crossover is defined as shifting a peak over an adjacent peak that has already been aligned to a peak in the paired peak profile. In addition, peaks are restricted by the amount of chemical shift position movement that is allowed based upon a user defined maximum. Therefore, a pair of peaks will only align together if the amount of movement that the peaks need to make for the alignment is less than this user defined maximum. Typically, the user would define this maximum chemical shift position movement as ± 0.04 or ± 0.03 ppm, but the value is data dependent. We note that by aligning each peak within its own user defined window the notion of linear or non-linear shifting of peaks across the spectrum need not be considered.

The next step in alignment involves defining corresponding unaligned segments of the peak profile pair as depicted in panel C of Figure 2.1. Here, each spectra is segmented such that only the unaligned peaks contained within a segment will be subject to the dynamic programming alignment process. Again, these segments are paired between the two peak profiles and segments are bounded on each side either by already aligned regions or "empty" regions where it is impossible to form an alignment between a pair of peaks based upon the user-defined maximum chemical shift variation.

Both the naive and dynamic programming alignment schemes rely upon a scoring function that determines the similarity between the two corresponding peaks (Equation 2.1). Note that

this similarity score is different from correlation, despite it ranging from 0.0 to 1.0. This score indicates the proportion of similarity between two peaks, i.e. a score of 1.0 indicates the corresponding peaks are exactly the same. The similarity is determined based on the three peak attributes of height at apex, h, width at half height, w, and chemical shift position, c. While in this work each of the three peak attributes are assigned so as to contribute an equal proportion to the score, the assignment of these proportions, p_h, p_w and p_c can readily be altered as appropriate. For both height and width, the similarity is measured by difference of the two values scaled by the larger of the two subtracted from one. For the variation in chemical shift, the similarity is measure by the difference scaled by the user defined maximum amount of acceptable variation between peaks, m, subtracted from one.

$$\begin{aligned} Score = \quad & p_h * \left(1 - \left(\frac{\mathsf{h}_a - h_b}{max(h_a, h_b)}\right)\right) + \\ & p_w * \left(1 - \left(\frac{\mathsf{w}_a - w_b}{max(w_a, w_b)}\right)\right) + \\ & p_c * \left(max\left(\left(1 - \left(\frac{\mathsf{c}_a - c_b}{m}\right)\right), 0\right)\right) \end{aligned} \tag{2.1}$$

A modified dynamic programming algorithm is used to align peaks within each of the segments (see next section for pseudocode). The algorithm involves using the typical dynamic programming recursion, where the scores assigned for a given alignment between peaks are defined using the scoring function enumerated above with a gap penalty, gp, is imposed for unaligned peaks. The modification involves assigning a large penalty, the boundary penalty bp, when alignment between two peaks involves chemical shift variation greater than m or when two aligned peaks do not achieve the minimum acceptable similarity for alignment, $minScr$. The user defines both of these values, -0.10 gp and -5.0 bp in this work, with the assignment of the large penalty preventing the algorithm from violating either the maximum allowable chemical shift variation, m, or the minimum allowable similarity for alignment.

The recursive formula for aligning a pair of peak profiles with our modified dynamic

programming scheme is the following (Equation 2.2). Given a pair of spectra S and T, one defines a scores matrix c such that c has i rows equal to length(S) + 1 and j columns equal to length(T) + 1. The function $Scr(x, y)$ returns the similarity score between two peaks, x and y, computed using the formula above. The gap penalty, gp, should be greater in value than the boundary penalty, bp. The gap penalty can range from the user defined minimum similarity, $minScr$ (0.60 in this work), to a small negative number, typically -1.0. The boundary penalty should be a large negative number (we used -999 in our implementation to allow the algorithm to automatically assign its value).

$$c[i,j] = \begin{cases} j * gp + i * gp & \text{if } i = 0 \text{ or } j = 0, \\ MAX\Big\{ c[i, j-1] + gp, & \\ \quad c[i-1, j] + gp, & \\ \quad c[i-1, j-1] + & \text{if } i, j > 0 \ \& \\ \quad Scr(S[i], T[j]) \Big\} & Scr(S[i], T[j]) \geq minScr, \\ MAX\Big\{ c[i, j-1] + gp, & \\ \quad c[i-1, j] + gp, & \text{if } i, j > 0 \ \& \\ \quad c[i-1, j-1] + bp \Big\} & Scr(S[i], T[j]) < minScr. \end{cases} \tag{2.2}$$

Panel E of Figure 2.1 illustrates the final step in the process where the consensus peak profile is formed. Specifically, the consensus profile is generated by assigning a new consensus peak to each successfully aligned pair of peaks, where this consensus peak takes on the median chemical shift value and the average relative height and width of the paired aligned peaks. Peaks from either profile that fail to align are allowed to "pass-through" to the consensus profile and maintain their original attributes. Panel E of Figure 2.1 depicts successfully aligned peaks as those contained within a shaded box, those that failed alignment are unadorned.

Figure 2.2 illustrates how the entire process diagrammed in Figure 2.1 is repeated on the resulting consensus profiles until a single consensus profile is produced. The peak profiles in the top row of figure 2.2 demonstrate the initial step where the pairs of input profiles are aligned together to form consensus profiles. The next steps involve pairing these resulting consensus profiles together, aligning them and forming 'new' consensus profiles. This process

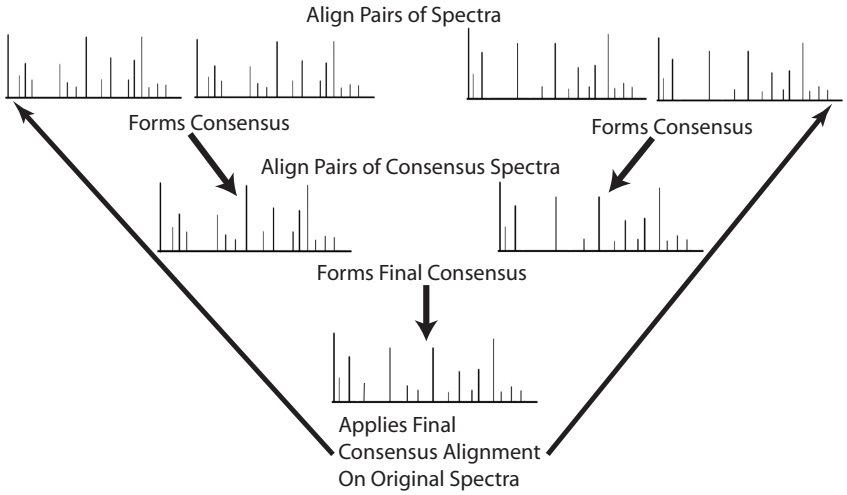


Figure 2.2: Final Consensus Profile Formation. Pairwise alignments are progressively combined together through the alignment of consensus profiles to form a final consensus profile. This profile is then used to adjust the chemical shift positions of the peaks from the original input peak profiles to their final aligned positions.

is repeated until only a single consensus profile exists as depicted at the bottom of Figure 2.2. This final consensus profile is used to adjust the chemical shift positions of the peaks from the original input profiles to their final aligned positions.

For the determination of optimal alignment parameters (p_h, p_w and p_c in the scoring function), we perturbed the peak attributes of the simulated peak profiles in a variety of different ways. Figure 2.3 depicts a representative result of one of the many simulations that were run. The results from these experiments indicate that using equal proportions is robust regardless of the perturbations introduced, as long as all three attributes experienced some amount of perturbation. If the amount of perturbation experienced by one of the three attributes is expected to be considerably less than the other two, the user might consider increasing its contribution to the score function.

2.2.4 Alignment Algorithms

For the alignment algorithms defined below, let A define the set of input peak profiles, where profile X can be defined by peaks y where $\{y \mid A[X, y], 1 \leq y \leq n_X\}$ and the value of n_X is

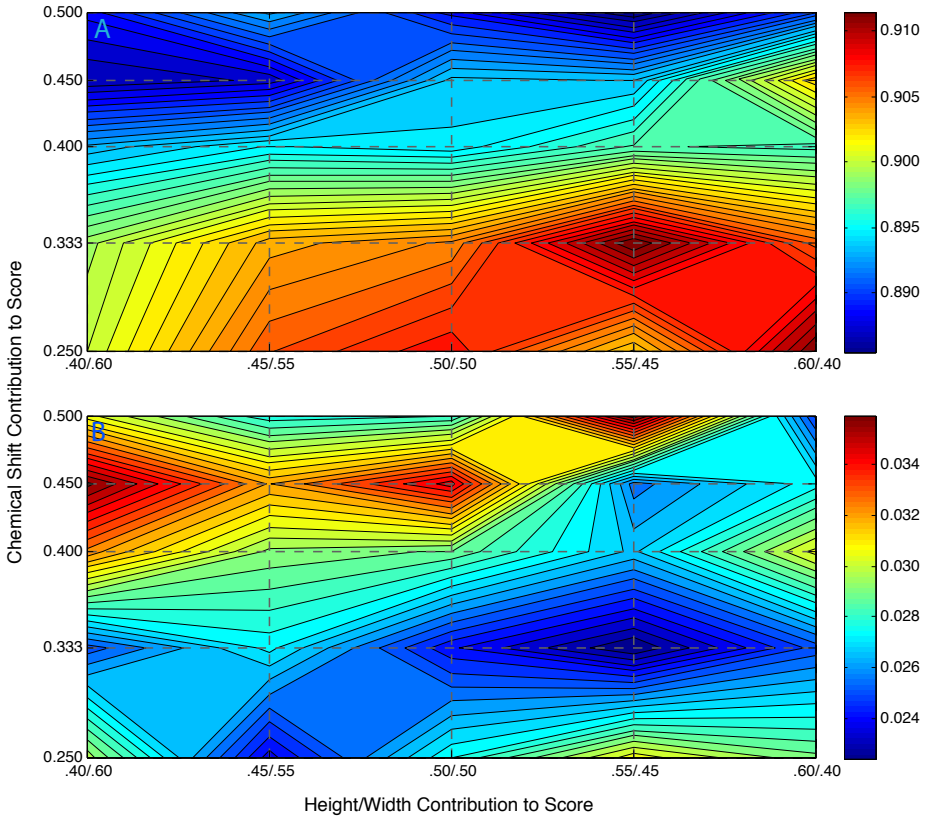


Figure 2.3: Accuracy of alignment as a function of scoring weights assigned to peak attributes (chemical shift, height, width). Simulated peak profiles were used with ± 0.03 variation in chemical shift for 50% of peaks, ± 0.10 perturbation in height for 25% of peaks, ± 0.10 perturbation in width for 25% of peaks, and 1-4 noise peaks randomly added to 50% of the profiles. Y-axis indicates the proportion of the score that is attributed to chemical shift position, x-axis indicates the proportion of the height and width that contribute to the remaining proportion of the score. Panel A depicts accuracy as indicated by the colorbar on the right and Panel B depicts the standard deviation of the accuracy measurements shown in panel A.

the number of peaks that were picked for peak profile X. The inputs for the pseudocode below involve only aligning a pair of peak profiles, S and T, such that $S, T \in A$, $\{i \mid A[S,i],1 \leq i \leq n_S\}$ and $\{j \mid A[T,j], 1 \leq j \leq n_T\}$.

2.2.5 Naive Alignment Scheme

The naive alignment incorporates a greedy algorithm that will align two nearby peaks as long as they are close in proximity (chemical shift position) to each other and achieve a high similarity score (i.e. also have high similarity in height and width). The procedure $NaiveAlign$(S,T,$maxCS$,$minScoreN$), naively aligns the pair of peak profiles S and T. This procedure inputs $maxCS$, the chemical shift value that is the maximum the user expects to have to shift a peak to obtain a match, and $minScoreN$, the minimum value of the similarity between two peaks to allow for naive alignment. Additionally, the value of $minScoreN$ is used to define the required amount of similarity in chemical shift position that two peaks must have to allow naive alignment.

The similarity between two peaks is calculated using the function $CalcScore(S[i], T[j], maxCS)$ which is based on the similarity score formula presented in the methods section of the paper. Typically, $minScoreN$ should be a high value of 0.88 or greater (0.90 for this paper) and $maxCS$ should range within 0.04 - 0.02 ppm (0.04 for this paper). Naive alignments are made using the procedure $MakeNaiveMatch$(S,T,$sIdx$,$tIdx$), which is not illustrated below due to its reliance on our algorithm's spectra data structure. The $MakeNaiveMatch$ procedure makes the naive matches given the input pair of peak profiles and the indices of their peaks that match. Nothing is returned, but the underlying peak profile data structure is changed to reflect the naive matches. Pseudocode for the algorithm can be found below as Algorithm 1 for naive alignment with the two helper functions defined in Algorithm 2 and Algorithm 3.

Algorithm 1

NaiveAlign$(S, T, maxCS, minScoreN) \equiv$
$m \leftarrow length[S]$
$ssT \leftarrow 1$
$searchCS \leftarrow maxCS * (1.0 - minScoreN)$
for $i \leftarrow 1$ to m do
    $temp \leftarrow ReturnStart(S, T, searchCS, i, ssT)$
    if $temp \neq -999$
      then $ssT \leftarrow temp$
            $matchI \leftarrow ReturnMaxMatchIdx(S, T, maxCS, searchCS, i, ssT,$
            $minScoreN)$
            if $matchI \neq -999$
              then $MakeNaiveMatch(S, T, i, matchI)$
            fi
    fi
end

Algorithm 2

ReturnStart$(S, T, searchCS, idxS, ssT)\ \equiv$
$n \leftarrow length[T]$
$z \leftarrow ssT$
if $T[z].chemShift \geq (S[idxS].chemShift - searchCS)$
  then
    while ($z > 1$ and $T[z].chemShift > (S[idxS].chemShift - searchCS)$) do
      $z \leftarrow z - 1$
    end
    if ($z \geq 1$ and $T[z].chemShift \leq (S[idxS].chemShift - searchCS)$)
      then if $T[z].chemShift = (S[idxS].chemShift - searchCS)$
        then $rssT \leftarrow z$
        else $rssT \leftarrow z + 1$
      fi
      else $rssT \leftarrow -999$
    fi
  else
    while ($z < n$ and $T[z].chemShift < (S[idxS].chemShift + searchCS)$) do
      $z \leftarrow z + 1$
    end
    if ($z \leq n$ and $T[z].chemShift \geq (S[idxS].chemShift + searchCS)$)
      then if $T[z].chemShift = (S[idxS].chemShift + searchCS)$
        then $rssT \leftarrow z$
        else $rssT \leftarrow z - 1$
      fi
      else $rssT \leftarrow -999$
    fi
fi
return $rssT$

Algorithm 3

ReturnMaxMatchIdx$(S, T, maxCS, searchCS, idxS, ssT, minScoreN)$ $\equiv$
$n \leftarrow length[T]$
$z \leftarrow ssT$
$bestV \leftarrow CalcScore(S[idxS], T[z])$
$bestI \leftarrow z$
$z \leftarrow z + 1$
while ($z \leq n$ and $T[z].chemShift \leq (S[idxS].chemShift + searchCS)$) **do**
    if $bestV < CalcScore(S[idxS], T[z], maxCS)$
      then $bestV \leftarrow CalcScore(S[idxS], T[z], maxCS)$
            $bestI \leftarrow z$
    fi
    $z \leftarrow z + 1$
end
if $bestV < minScoreN$
  then $bestI \leftarrow -999$
fi
return $bestI$

2.2.6 Dynamic Programming Alignment Scheme

To dynamically align the pair of peak profiles S and T, the procedure $DynProgAlign$(S,T, $maxCS$,*gp*,*bp*, $minScoreD$) uses the recursive formula defined in the methods section of the paper. The recursive formula from the paper defines an alignment scores matrix $c[i, j]$ and the backtrack matrix $b[i, j]$ that indicate the optimal solution. Notice that indices i and j from these matrices (scores and backtrack) are defined as i=0,...,n_S and j=0,...,n_T. The pseudocode in Algorithm 4 follows a modified dynamic programming alignment scheme as outlined by the recursive formula in the paper.

The function $CalcScore(S[i], T[j], maxCS, minScoreD)$ calculates the similarity score between two peaks using the similarity score formula as indicated in the methods section of the paper. A gap penalty, *gp*, is incurred each time two peaks fail to align. A boundary penalty, *bp*, is incurred each time two peaks are so dissimilar that they should not be allowed to align together based on the user defined minimum similarity required for dynamic align-

ment, $minScoreD$, or if two peaks' difference in chemical shift position is greater than the maximum allowable chemical shift variation, $maxCS$. The gap penalty, gp, and boundary penalty, bp, can be input by the user.

Typically, $minScoreD$ is a high value of 0.50 or greater, but this parameter should be set based upon the user's discretion (for this paper 0.60). The gap penalty (-0.10 for this paper) should be greater in value than the boundary penalty (-5.0 for this paper). The gap penalty can range from the user defined minimum similarity, $minScoreD$, to a small negative number, typically -1.0. The boundary penalty should be a large negative number or set to -999 to allow the algorithm to automatically set the value. Pseudocode for the algorithm can be found below as Algorithm 4. Notice for simplicity the $CalcScore(S[i], T[j], maxCS, minScoreD)$ function in the algorithm is implemented in a manner that returns a similarity score that is less than $minScoreD$ if the difference in chemical shift position is greater than the maximum allowable shift, $maxCS$. This allows the boundary penalty to be incurred when it is appropriate.

Algorithm 4

DynProgAlign$(S, T, maxCS, gp, bp, minScoreD)$ $\equiv$
$m \leftarrow length[S]$
$n \leftarrow length[T]$
if $bp == -999$
  then $bp = gp * n * m$
fi
for $i \leftarrow 0$ to m do
  for $j \leftarrow 0$ to n do
    if $i = 0$ or $j = 0$
      then $c[i, j] \leftarrow i * gp + j * gp$
        if $i = 0$
          then $b[i, j] \leftarrow$ " $\leftarrow$ "
          else $b[i, j] \leftarrow$ " $\uparrow$ "
        fi
      else $DiagScore \leftarrow CalcScore(S[i], T[j], maxCS, minScoreD)$
        if $DiagScore < minScoreD$
          then $DiagScore \leftarrow bp$
        fi
        if $(c[i-1, j-1] + DiagScore \geq c[i-1, j] + gp)$ and
          $(c[i-1, j-1] + DiagScore \geq c[i, j-1] + gp)$
          then $c[i, j] \leftarrow c[i-1, j-1] + DiagScore$
            $b[i, j] \leftarrow$ " $\nwarrow$ "
            if else $(c[i-1, j] + gp \geq c[i-1, j-1] + DiagScore)$ and
                $(c[i-1, j] + gp \geq c[i, j-1] + gp)$
              then $c[i, j] \leftarrow c[i-1, j] + gp$
                $b[i, j] \leftarrow$ " $\uparrow$ "
              else $c[i, j] \leftarrow c[i, j-1] + gp$
                $b[i, j] \leftarrow$ " $\leftarrow$ "
            fi
        fi
    fi
  end
end

2.2.7 Algorithm speed

In our Python implementation, alignment of the described 22 real mouse urine spectra takes approximately 2 minutes on a 2GHz laptop. Approximately 30 seconds involves the actual process of alignment, with the remaining time involving peak picking and other data process-

ing. Alignment of 150 real mouse urine spectra takes approximately 38 minutes with 1 min 54 sec being involved in alignment.

2.2.8 Simulation of NMR spectra peak profiles

To generate NMR profiles that were as realistic as possible our simulated peak profiles are based upon characteristics of urine spectra from mice. Specifically, we follow the distribution of peak locations, heights and widths as estimated from murine urine spectra using the spectrum visualization utility implemented in ACD 1D NMR Processor, version 11 (Advanced Chemistry Development, Toronto, Canada). The spectral peaks used for calculating these distributions range in chemical shift position from 2.0 ppm to 4.10 ppm. These distributions are coded into a software utility that allows the generation of simulated peak profiles. In addition, user-defined levels of noise in chemical shift position, height and width can be defined. To help simulate attributes observed with real NMR spectra, the number of peaks generated per spectrum is varied through the addition of noise peaks to the simulated profiles. To evaluate algorithm performance with profiles originating from multiple distinct classes, we generate spectra from distinctly different templates where the user defines the number of peaks common between templates.

2.3 Results

While the alignment method we propose consists of several steps which are described in detail in Methods, we provide a brief overview here. As outlined in Figure 2.1, our approach begins by first characterizing each individual spectrum by defining its peaks. The process of picking peaks can be done through a variety of methods and we have used a straightforward approach that uses the derivative of the spectrum, and other associated properties for discerning peaks. The resulting set of peaks contains the location, height and width of all peaks in a spectrum, referred to as the peak profile. These features comprise the main information content that is

used in the interpretation of NMR spectra. This approach allows each NMR spectrum to be represented by a much smaller collection of data points than if we used the full resolution of the acquired spectrum. For example, our experimental urine spectra were collected with 16384 points, but the peak picking process found that the spectrum contained less than 500 significant peaks. It must be noted that peak picking may result in loss of information if some peaks are not picked. The peak picking algorithm is under active development to ensure that peak information is not lost due to features such as low signal to noise or spectral crowding. Future work will also consider spectral features such as multiplet structure to provide more accurate peak profiles.

In the next step of the process, pairs of peak profiles are chosen for alignment, where the most similar profiles are determined through pair-wise statistical correlation (Figure 2.1A). Thus we start by aligning the most similar pairs of profiles to each other first. Each of these pairs of profiles is then aligned through a series of progressively more rigorous steps that begins with the naive alignment of the most highly similar peaks (Figure 2.1B-D). This naive alignment establishes aligned regions of high identity separated by segments that cannot be so readily aligned. These segments, bordered on either side by high-confidence aligned regions, are then aligned through a dynamic programming algorithm where the alignment score is based on chemical shift position, peak height and peak width. Note that only the peak location is altered throughout the alignment process and that peak height and width remain unaltered. Following this first pairwise alignment, a single consensus profile is created (Figure 2.1E). This process is then repeated, first for each set of pairs and then progressively for all of the generated consensus profiles. At the end of this process a single representative consensus profile is generated which defines the final alignment (see Figure 2.2). The final output consists of the set of input profiles with their respective peaks aligned to this final consensus alignment.

2.3.1 Alignment of simulated spectra

As "gold-standard" completely characterized NMR spectra for use in validation are not available, we used a simulation approach for generating peak profiles that could then be used to assess the performance of the alignment methods. In particular, the use of simulated profiles allows us to determine whether or not two or more peaks aligned through our algorithm should actually be aligned with each other, and if not, which other peaks they should be aligned to. It also allows us to introduce defined amounts of noise, either in the form of chemical shift variation, peak height, peak width, or randomly introduced "noise" peaks into each profile and measure their effect on alignment accuracy. As we wished to generate NMR profiles that were as realistic as possible, our simulated profiles were composed of a subset of peaks picked from an actual mouse urine spectrum (see Methods).

As a test of our alignment approach, we attempted to align simulated profiles under a variety of noise conditions. In these tests we generated two sets of profiles consisting of 32 profiles each, where each set was based on a different template. Each template consisted of a total of 50 peaks, 13 of which were unique to each group, allowing us to look at the effectiveness of alignment in the presence of inter-group variation. In addition to the differences derived from the peaks specific to each group, predefined amounts of chemical shift, peak height and peak width variation were also introduced before alignment. Finally, 50% of the profiles in each group had from 1 to 4 additional noise peaks inserted at random positions within each profile.

The effects of chemical shift variation on alignment accuracy are shown in Figure 2.4 where, in addition to chemical shift variation, 25% of peaks were subject to noise of $\pm 10\%$ in peak height and/or peak width at half-height. The contribution of chemical shift, peak height and width to the alignment score were kept equal in this and all other tests as this combination was found to be highly robust. Sensitivity to the choice of these weighting parameters is shown in 2.3. In Figure 2.4 we see that the accuracy of alignment is highly robust to chemical shift variation as can be seen by the slow decrease in accuracy with increasing variation. Here, alignment accuracy is calculated by dividing the number of peaks correctly aligned by the

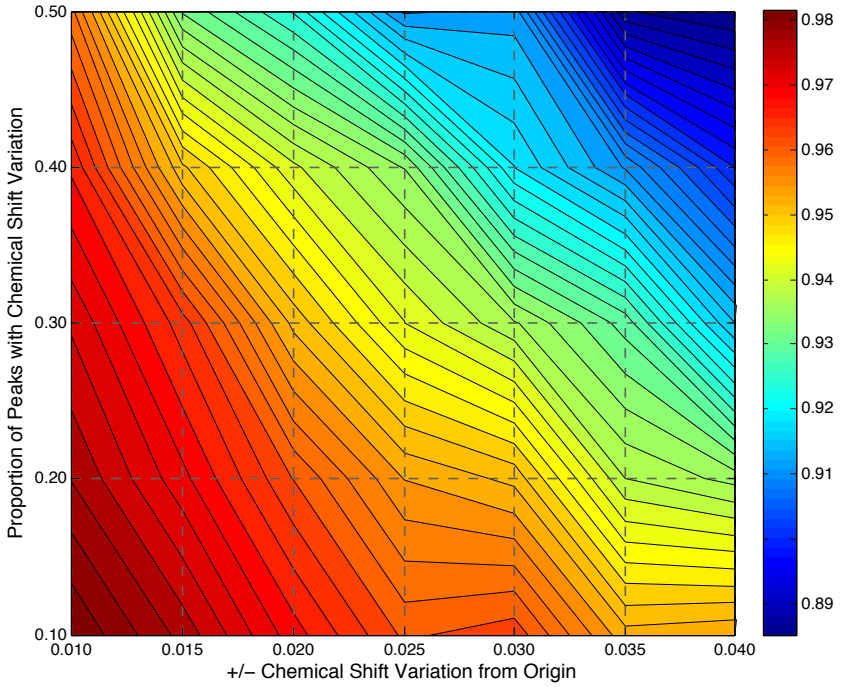


Figure 2.4: Accuracy of Alignment with Simulated Peak Profiles. The x-axis indicates $\pm$ range of chemical shift variation and the y-axis indicates proportion of peaks per profile that experienced chemical shift variation. The graph depicts accuracy as indicated by the colorbar on the right, where PCANS achieved accuracies between 98.4% and 88.5%. Besides chemical shift position, both relative intensity and width were randomly perturbed by $\pm 10\%$ of the origin for 25% of the peaks within each profile and 50% of the profiles had 1-4 noise peaks randomly added.

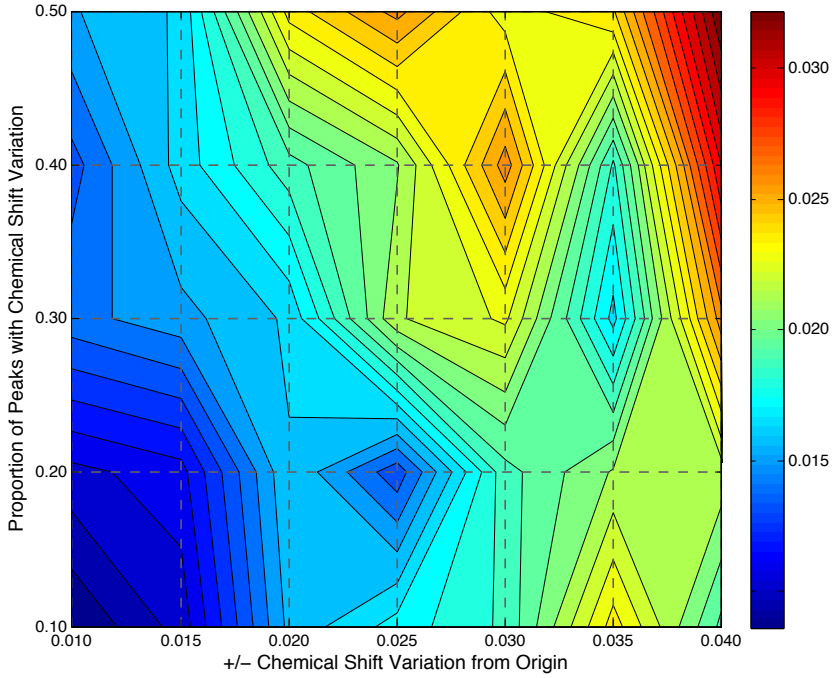


Figure 2.5: Standard deviations corresponding to the alignment accuracies shown in Figure 2.4. The x-axis indicates $\pm$ range of chemical shift variation and the y-axis indicates proportion of peaks per peak profile that experienced chemical shift variation. The graph depicts the standard deviation of the accuracy as indicated by the colorbar on the right. Besides chemical shift position, both relative intensity and width were randomly perturbed by $\pm 10\%$ of the origin for 25% of the peaks within each peak profile and 50% of the profiles had 1-4 noise peaks randomly added.

total number of peaks. Alignment is similarly robust to increases in the proportion of peaks subjected to such variation. In fact, a nearly 90% accuracy is maintained despite 50% of peaks experiencing variation of up to ± 0.04 ppm. The maximum standard deviation is ± 0.033 and the corresponding map of deviations is shown in Figure 2.5. While we used a window of ± 0.04 ppm in the alignment of individual peaks, this is a user-defined quantity that can be changed to suit the underlying data.

We also compared the accuracy of our alignment method between our consensus approach and the use of a template. Again, we started with two sets of profiles, with each set consisting of 32 profiles and 50 peaks, with 13 peaks unique to the set. Variation in chemical shift position (± 0.02 ppm) was introduced for 50% of the peaks. Peak height and width noise (25% of peaks affected with $\pm 10\%$ variation) was also independently introduced. As before, 50% of the profiles in each group had 1 to 4 noise peaks inserted at random chemical shift positions.

We iteratively chose one of the sixty-four peak profiles as the template to which all the other profiles were aligned. Thus this approach differs from the PCANS alignment method only in the fact that it uses a representative profile as a template for use in aligning the other peak profiles; all other steps are identical including the dynamic programming alignment of peaks. Over all 64 possible templates, the average accuracy using this approach was 84.4% with 99% confidence intervals of 84.02% and 84.68%. The best single template had an accuracy of 87.5%. In contrast, PCANS had a 93.9% accuracy (PCANS generates only one answer so there are no error bars in this case).

A representative region of an alignment is shown in Figure 2.6 where the template generating the highest accuracy (87.5%) was used to generate the shown template-based alignment. Differences between the template, unaligned and PCANS alignments can be readily observed. For example, three regions are highlighted that have peaks unique to Group 1. In Region 1 of the unaligned spectra (center row), it is possible to pick out by eye the existence of two likely peaks in Group 1 with no nearby corresponding peaks in Group 2. In addition, a peak unique

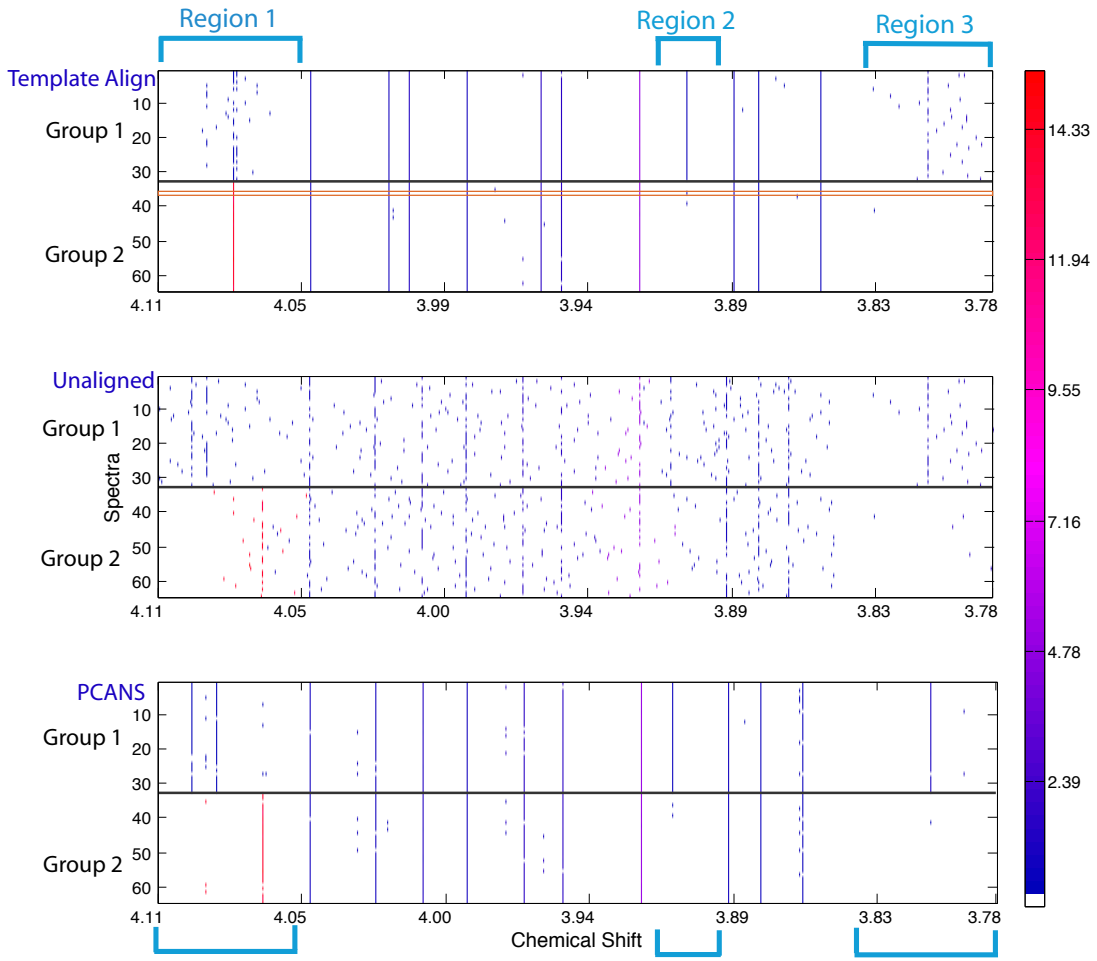


Figure 2.6: A Sample Region of Simulated Peak Profiles Before and After Alignment. Alignment is shown with either PCANS-aligned or template-aligned peak profiles. Short, individually colored bars indicate a profile's peaks. Peak profiles were simulated from two groups having group-specific peaks, with 32 profiles in each group. The colorbar on the right indicates relative height (intensity) of the simulated peak profiles. The horizontal orange rectangle in panel A indicates the best overall individual peak profile that was used as the template for this alignment. Regions indicated by vertical cyan rectangles depict cases where Group 1 has unique peaks that differentiate it from Group 2.

to Group 2 is also visible in this region. In the template alignment (top row) the two peaks of Group 1 could not be aligned, as the best overall template did not contain associated peaks in these locations. In addition, the template did have a peak in Group 2, but at the wrong location, forcing alignment of the unique Group 2 peak to a shifted location. In contrast, PCANS correctly aligned the Group 2 peak (bottom row). Furthermore, the two peaks unique to Group 1 were also successfully aligned. Note that the rightmost peak of the pair appears to be shifted to the right. This is due to the variation present within the unaligned set of peaks. Aligning noisy spectra containing peaks with varying chemical shift position with PCANS results in the alignment of peaks at their median chemical shift position. This provides a robust estimate of peak position despite potentially significant amounts of spectral noise.

In Region 2 of the template alignment, we see a well-aligned peak for Group 1. However, as we are using simulated data, we know that the position of this alignment is centered at a nearby noise peak within the template profile and inspection of the unaligned profiles also shows no obvious peak. The correct result is shown in Region 2 of the PCANS alignment. This incorrect alignment occurs because the "best" template happens to contain a nearby noise peak that is used as the basis for alignment of all other profiles.

Finally, in Region 3 (unaligned) we see strong indications of a peak in Group 1 as well as alignment of this peak with PCANS. However, in the template alignment we see no obvious change relative to the unaligned profiles. This is due to the fact that the template profile had no peaks in this region and thus none of the identified peaks could be aligned. The fact that they are present at all in the final alignment is due to the PCANS-portion of the algorithm (non-template), which allows these orphan peaks to pass through to the final alignment regardless of whether or not they are found in the template. Overall, this example demonstrates the inherent pitfalls and challenges that arise with any alignment method that is based on the concept of a template or standard spectrum.

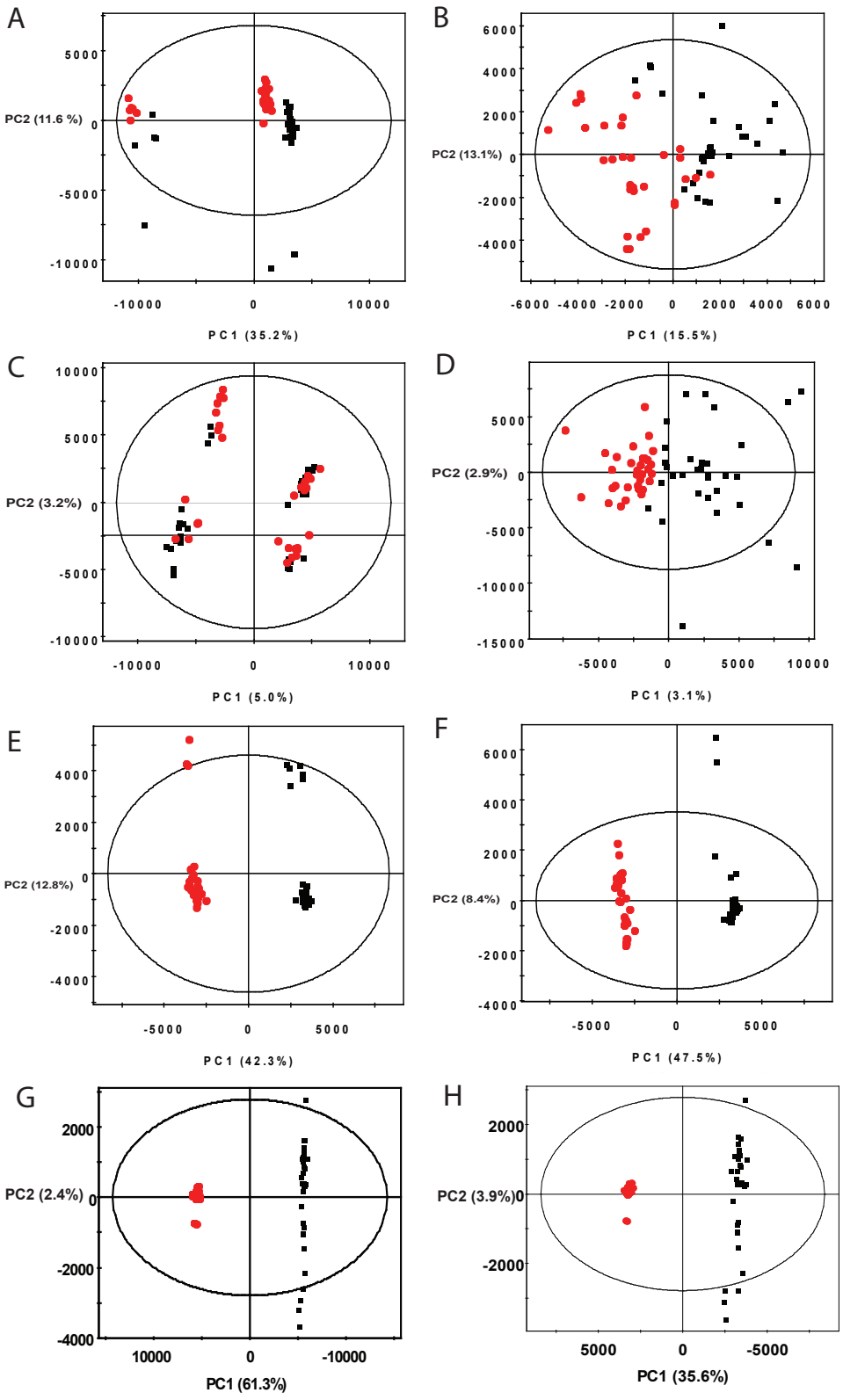


Figure 2.7: PCA Analysis of Simulated Peak Profiles. Displays binned (A & B), unaligned (C & D), PCANS aligned (E & F), and template aligned (G & H) simulated peak profiles, with (B, D, F & H) and without (A, C, E & G) outliers removed.

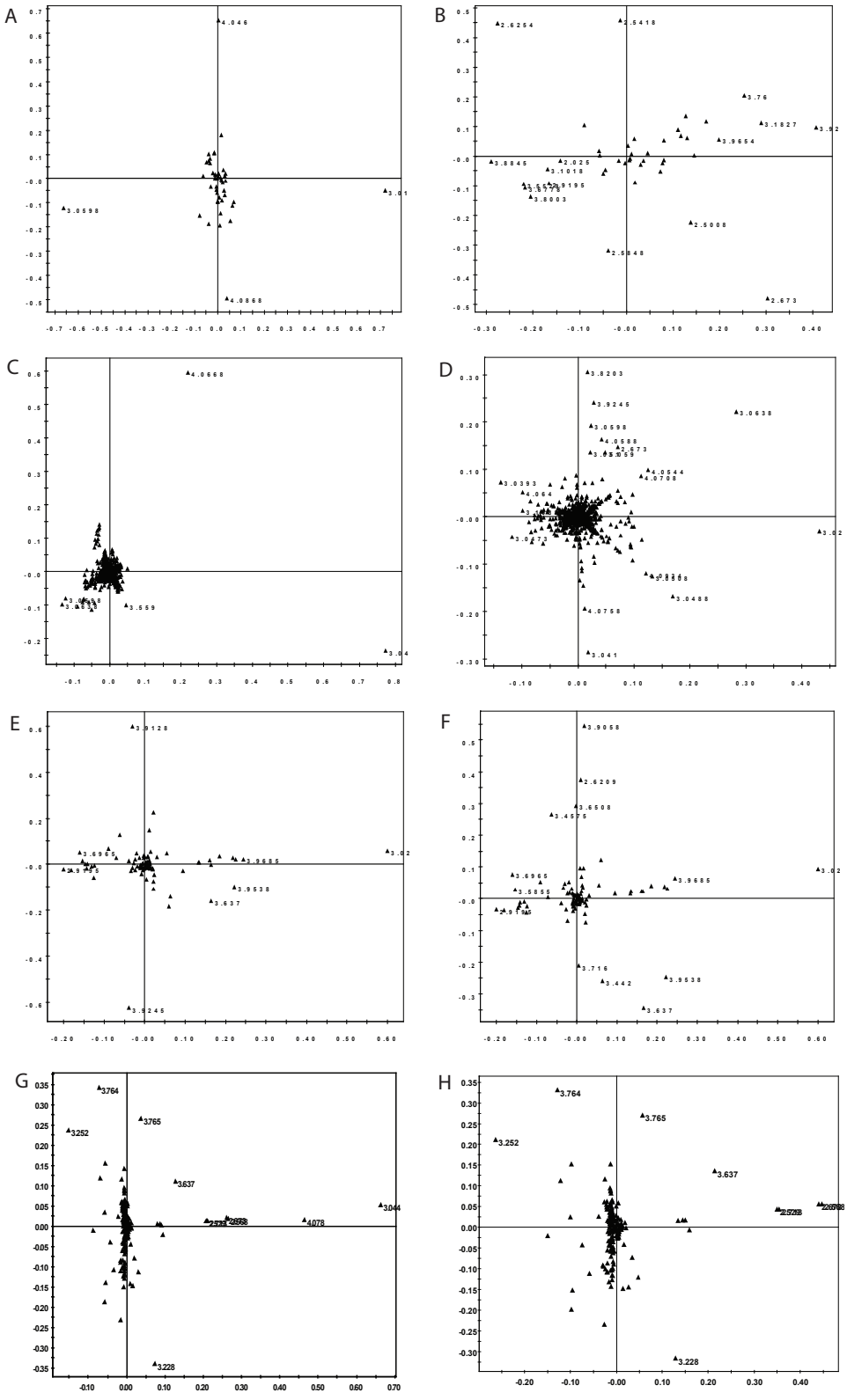


Figure 2.8: Loadings plots of binned (A & B), unaligned (C & D), PCANS aligned (E & F), and template aligned (G & H) simulated peak profiles, with (B, D, F & H) and without (A, C, E & G) outliers removed.

2.3.2 PCA analysis of simulated spectra

To further demonstrate how spectral alignment with PCANS can improve downstream data analysis, we performed a PCA analysis of unaligned, binned, PCANS-aligned and template-aligned peak profiles (see Figure 2.7). This data set consists of 1368 unique peaks when unaligned, 216 unique peaks after template alignment, 91 unique peaks after alignment with PCANS and 46 chemical shift position bins. PCA analysis of a perfect alignment of the data would show two tightly clustered groups with the only variance due to the small number of noise peaks added to each of the groups. The scores plot resulting from a standard binning procedure with uniform 0.04 ppm bin widths is shown in Figure 2.7A. This plot displays three distinct groups along with several outliers on the bottom half of the plot. The simulated peak profiles contain two large peaks that were modeled after the creatinine peaks that are found in urine. Thus the separation of the three clusters, as well as the outliers in this model, are largely due to inclusion of the creatinine peaks into four separate bins. The corresponding loadings plot for this scores plot is given in Figure 2.8. Figure 2.7B shows the scores plot that results from excluding those four bins. The separation between the groups is clearer, but the clustering is still rather diffuse.

Similarly, Figure 2.7C shows the PCA scores plot of the unaligned peak profiles. In this case the four clusters do not distinguish the groups, and are again based upon differences in the peak positions of two large creatinine peaks. As with the binning example, removal of these two peaks leads to a clearer discrimination of the groups, but again with a diffuse clustering as observed in Figure 2.7D.

In contrast, Figure 2.7E shows the results of PCA analysis after alignment with PCANS. The separation between the groups along the first principal component is complete and the clustering is very tight. The outliers from each group near the top of the plot are again due to one creatinine peak that did not get aligned with the rest. Removal of this peak from the analysis lead to the scores plot in Figure 2.7F. Although there are still a small number of outliers, the separation between the groups along the 1st principal component is excellent and

this PC explains nearly fifty percent of the variance in the data.

The final comparison is with the template aligned peak profiles. Figure 2.7G displays a very good discrimination of the groups, with the control groups being very tightly clustered. This is reasonable as the template was chosen from that group. If, as with the other plots, the creatine and creatinine peaks are removed, the separation looks quite similar but the percent variation explained by the first principal component decreases by nearly half. Compared with the PCANS plot, the control group is more tightly clustered, but the treated group is less well so. Furthermore, the first principal component of the PCANS alignment explains 46.5% as opposed to 35.6% of the variation. In this rather simple example of only two groups, the PCANS alignment does have some advantages, but the benefits would be expected to be much greater in a more complex situation which has more than two groups.

2.3.3 Alignment of Mouse Urine Spectra

To demonstrate the utility of PCANS on real data, we applied our approach to the alignment of twenty-two mouse urine spectra from a recent study of ethanol toxicity (Bradford et al., 2008). In this study, half of the samples were taken from mice receiving chronic ethanol treatment and the other half were from controls, with results from PCA analysis shown in Figure 2.9. In Figure 2.9A, the data was analyzed using standard binning with 0.04 ppm bins, resulting in 152 bins across the spectrum. As can be seen, the correct separation of the data into two groups is discovered, largely due to the presence of ethanol and ethyl glucuronide in the spectra of dosed mice (see Figure 2.10). This positive result indicates that the chemical shift drift amongst these peaks is generally smaller than the applied bin width (i.e. the bin-widths are appropriately set to capture the chemical shift variation within these samples). Figure 2.9B shows the data prior to alignment. In its unaligned form, this dataset consists of 1496 unique chemical shifts. In this case the control samples are very tightly clustered (black points) and the major variation in the data appears in the dosed spectra. Again, the corresponding loadings plot shows that the separation of the dosed group into two clusters is predominantly due to

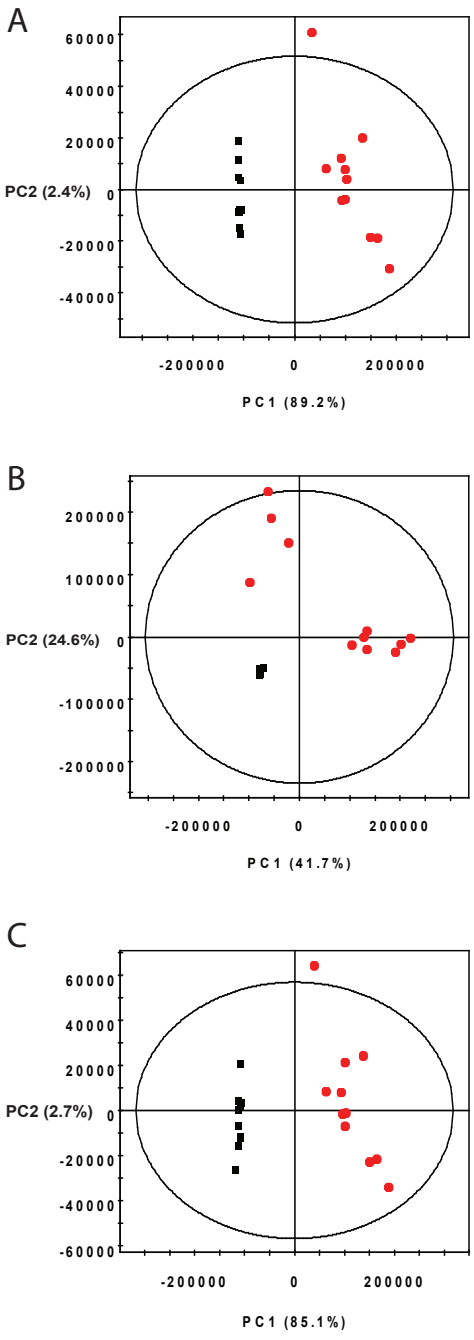


Figure 2.9: PCA Analysis of Mouse Urine Spectra. Displays binned (A), unaligned (B), and aligned (C) mouse urine peak profiles.

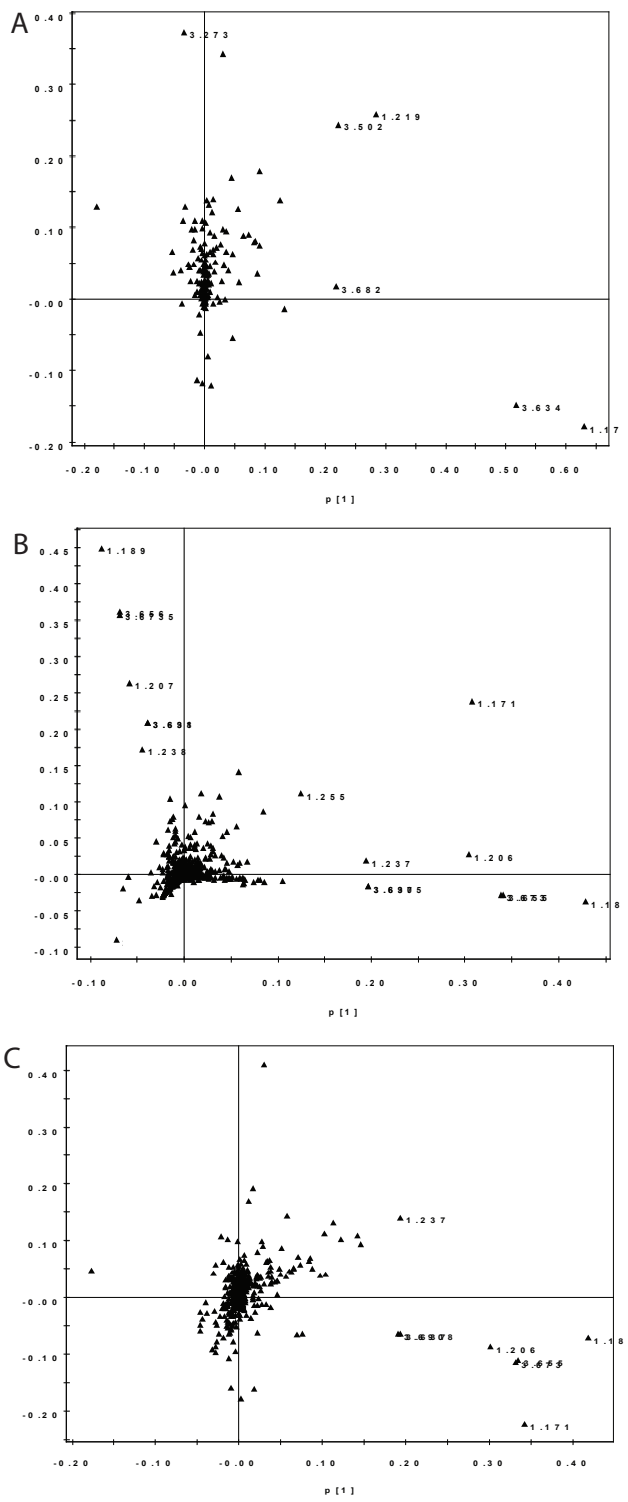


Figure 2.10: Loadings plots of binned (A), unaligned (B), and aligned (C) mouse urine peak profiles. Peaks associated with EtOH and EtOH-glucuronide are labeled with their chemical shift position.

the small differences in the ethanol and ethyl glucuronide peaks (Figure 2.10B). After peak alignment with PCANS, the number of unique chemical shifts is reduced to 483. The scores plot looks remarkably similar to that generated from the binned data and the percent variances for both of the PCs in these models is very similar (see Figure 2.9C).

Given the similarity between the binned and aligned data, the advantage of alignment with PCANS may not be obvious. However, it should be emphasized that the information content of the PCANS-aligned data is over three-fold larger than that of the binned data. More specifically, the intensity of 483 individual peaks is represented in the PCANS-aligned data, while the binned data encodes only 152 variables, many of which are influenced by (i.e. containing) multiple peaks. We can better observe the advantage that PCANS alignment provides through this added information by looking at the results of a supervised OPLS analysis.

Figure 2.11 shows the OPLS loadings coefficients plots using data from each of the three data sets, with original spectra from dosed and control samples superimposed in Figure 2.11A (Cloarec et al., 2005; Wiklund et al., 2008). Figures 2.11B-D show the back-scaled loadings coefficients such that the spectral features that are higher in the control group are positive and those that are higher in the ethanol treated group are negative. The color relates to the strength of the correlation, with red being the strongest. In Figure 2.11B, we see the OPLS coefficients prior to alignment and observe very weak correlation between peaks (blue-green colors in the Figure). In addition, several spectral features are largely missed (Regions 1 and 4 indicated in brackets at the bottom of the figure) or only weakly identified (Regions 2, 3 and 5). Application of OPLS analysis to binned data is shown in Figure 2.11C. The decrease in the data density due to peak consolidation into bins can be observed in this figure by the sparseness of data along the x-axis. While some of the correlations are higher, there are inappropriate assignments between groups as can be observed in the positively-valued peaks in Region 2. While the binned data shows the importance of the ethanol peaks in distinguishing the groups, interestingly the signals from ethyl glucuronide are very weak.

In contrast, Figure 2.11D shows the loadings coefficients after PCANS alignment. As can

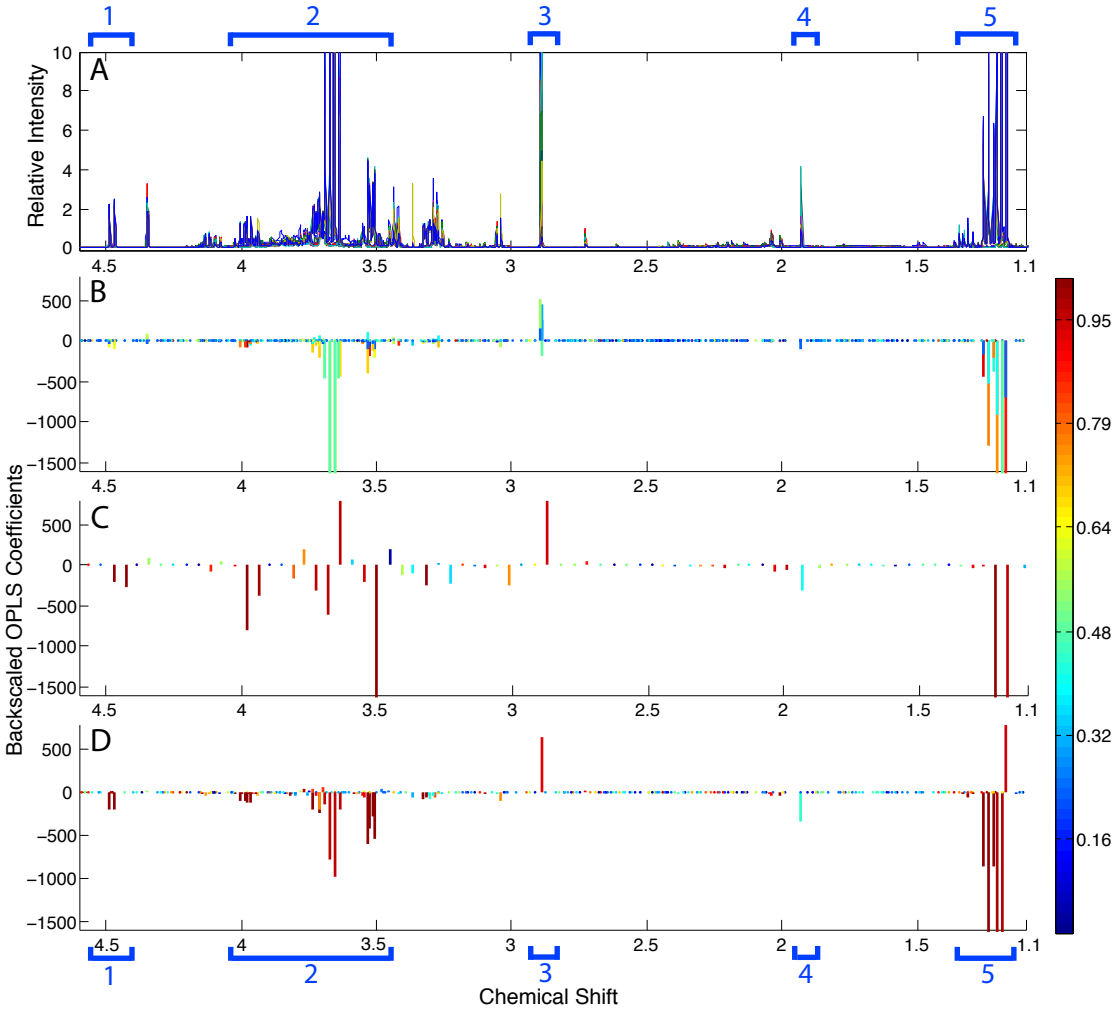


Figure 2.11: OPLS Analysis of Mouse Urine Spectra. Displays unaligned (B), binned (C), and aligned (D) OPLS coefficients from mouse urine peak profiles. Panel A depicts the unaligned peak profiles that correspond to the OPLS analysis. Peaks associated with glucose are located in bracket 1, ethanol and ethyl glucuronide in brackets 2 & 5, TMA in bracket 3, and acetate in bracket 4.

be seen along the x-axis, data density (and thus information content), is significantly higher than with binned data. In addition, the signals from ethanol and ethyl glucuronide are all very strong with colors indicating a very high confidence. Also note that while template alignment would perhaps perform similarly with regard to generating strong correlations for aligned peaks, it suffers from the earlier described difficulties that will lead to loss of peaks defining group/inter-group differences, loss of unique peaks, and alignment of different, but close peaks to nearby peaks within the template. These results demonstrate the value alignment with PCANS and its ability to enhance the information content relative to the standard binning protocol.

2.4 Conclusions

The increasing scale and complexity of metabolomics studies is driving the need for improved computational tools for data integration and analysis. We have described our PCANS approach which was developed to address the need for multiple spectrum alignment where noise in the form of chemical shift variation and deviations in peak properties is present, along with significant sample complexity.

In metabolomic analyses there are often multiple groups of spectra, such as control and treated groups, which may not be appropriately aligned with any algorithm that is primarily based upon the use of a template. For instance, the peaks from the exogenous metabolites that are present in the treated group may not be well aligned using a template from the control group. Similarly, alignment of the control spectra could be confounded by the presence of the exogenous peaks in the template. Even when a specific group (e.g. the treatment group) can be reasonably well aligned by a representative template spectrum, metabolomics is also being increasingly used to determine multiple responder phenotypes wherein the treated group may contain several subgroups characterized by distinctly different spectra. Thus a significant advantage of PCANS over the use of template-derived methods is that it is a fully unsupervised

method and can be used to align spectral data containing multiple disparate groups that may not/cannot be anticipated. Furthermore, as both aligned and unaligned peaks are incorporated into the final consensus, we minimize the amount of data lost in this process while enhancing the signal within alignable regions.

This algorithm uses peaks rather than full resolution spectra as the basis for alignment. We consider this to be a general advantage as the datasets are much smaller while having no loss of real spectral information, as peak location, height and width are all maintained. A primary goal of the PCANS process is to provide better input data for multivariate statistical analysis that will help identify significant groups in the sample set. As shown in the OPLS loadings coefficients plots in figure 2.11, the peak profiles provide an ample representation of the NMR spectra so that specific compounds can be identified. It could be argued that these peak profiles are even easier to interpret than real spectra as they are free from spectral noise and have perfectly uniform linewidths. But, should a more traditional representation of NMR spectra be desired, the information has been maintained to regenerate such a spectrum. In general, the use of peak profiles as input to PCANS allows us to maximize the amount of high-quality information available for alignment and further downstream analysis. We are further investigating the use of more sophisticated peak picking algorithms that will include recognition of peak multiplets and advanced recognition of shoulder peaks that are often present in samples such as serum that displays large, broad peaks due to the presence of macromolecules. While we have used a robust peak picking algorithm, improvements in this step will help minimize issues of inconsistent peak selections across samples. Continued incorporation of more sophisticated scoring schemes as well as more realistic handling of multiplets (rather than as separate peaks), is expected to further enhance the effectiveness of our approach. Finally, while PCANS was developed specifically with NMR data in mind, it has the potential to be applicable to the alignment of other types of data with similar properties. In particular, chromatographic data which is composed of peak positioned along a time axis would be amenable to PCANS alignment. Future work will attempt to further extend these capabilities.

2.5 Future directions

Recently the peak picking method has been revised to reflect the one used Abdo et. al.(Abdo et al., 2006). The primary difference between our method and the method Abdo et. al. described, is that ours is fully automated as to self-select a cutpoint based upon the assumption that one would expect similar numbers of peaks to be selected from all spectra. This is an improvement over the peak picking algorithm that was used by the original implementation of PCANS because one does not have to select parameters for peak picking. Additional functionality has been added that allows the reconstruct of the consensus spectrum's appearance based upon the original spectra. A webtool that performs Statistical Total Correlation Spectroscopy, STOCSY, on PCANS aligned or binned spectra has been implemented (Sasic et al., 2000). In the future, I would like to improve PCANS alignment by allowing for differing size alignment windows to be used based upon the position in the spectrum (7.5+ aromatic regions are typically more misaligned). Additionally, I would like to improve the framework to more readily align hundreds of spectra through the use of sampling to form the consensus, as to prevent an over abundance of peaks in the consensus spectrum.

As mentioned above metabolomics is being used to determine multiple responder phenotypes wherein the treated group may contain several subgroups characterized by distinctly different spectra. Thus OPLS analysis as shown in Figure 2.11 might fail to find subgroups from amongst the multiple responder phenotypes unless one can determine apriori which spectra belong to which subgroups. Methods that identify subsets of peaks and sample spectra that act in a similar fashion could identify such responder phenotypes without any apriori knowledge. This would especially be useful coupled with a measure of strength of association, which is used to determine association between different classes of spectra. The data mining methods discussed in chapters 3-5, can be used in this manner as to provide a more sophisticated way to analyze multiple responder phenotypes.

Chapter 3

Background and Related Work

3.1 Motivating Problem

Identification of associations that exist between data when the data under consideration comes from multiple data sources or where nontrivial amount of noise has been introduced into the data makes traditional means of analysis difficult. When there exists variable signal to noise ratio within a data source or across data sources, considering the entire data record (using all the data) is misleading due to the underlying noise which mask existing associations. One way of dealing with this issue is to use a Bayesian methodology to weight the reliability of the data included in the model as was illustrated by Webb-Robertson et al. (2009) using Metabolomic data. Another approach is to subset the data down to subsets that contain the strongest associations. This approach was partially explored by DiMaggio et al. (2010) when they used biclustering coupled with logistic regression to identify association between an optimal set of explanatory variables and their corresponding response variable. DiMaggio et al. used biclustering to remove redundancy amongst the explanatory variables and to determine the two most anticorrelated classes of response variables. Using biclustering in this manner allowed them to select sets of variables that would be most amenable for use with their logistic regression modeling scheme. Similar to DiMaggio et al., we subset the data to determine where the associations between the data are the strongest. Unlike DiMaggio et al., we create subsets by

considering both explanatory and response variables simultaneously; thus, producing results that are most likely to show the true underlying associations that exist within the data.

Similar to the biclustering methods, we subset the data down simultaneously by rows and columns. By using closed frequent itemset mining, our algorithm provides overlapping subsets of data limited by a minimum row threshold. Consider the case where numeric association between a set of explanatory and response variables is so complicated and contradictory that the only reliable information that can be deduced is a binary classification. Attempting to quantify the level of response to more than a binary classification just confounds the association between explanatory and response variables. The data is simplified down to a binary classification that either a response occurred or it did not occur. Because the data is binary, closed frequent itemset mining can be used to subset the data down by rows and columns simultaneously restricting the resulting subsets of data only by a minimum row threshold (known as minimum support threshold). Similar to biclustering this methodology has computational complexity of NP complete. Although high complexity, the closed frequent itemset mining approach has the benefits of considering explanatory and response variables simultaneously and will fully enumerate all subsets of data that meet the minimum row threshold. Both properties help our approach to discover the results that have the strongest associations between explanatory and response variables. Furthermore, using closed frequent itemset mining allows for a intuitive inclusion of user defined approximation/fuzziness (zeros) into the results.

Both DiMaggio and van Uitert et al. (2008) use biclustering, but van Uitert uses the resulting biclusters to identify novel associations between two integrated datasets; whereas, DiMaggio uses biclustering as means of data reduction in a much larger modeling scheme. Similar to van Uitert et al. our methodology integrates binary datasets, but our methods also statistically assess results to determine the importance of the resulting subsets. Additionally our methodology provides full enumeration of potential results for a given threshold; whereas, van Uitert's method provides the top scoring non-overlapping biclusters and it does not guarantee that the

resulting clusters show an association between the integrated datasets. We illustrate how our approach can be adjusted to integrate three or more datasets and provide results for larger, more complex datasets using a focused analysis.

3.1.1 Real World Data Example

The methods we explore are designed to identify association between sets of response and explanatory variables when the data under consideration comes from multiple data sources or nontrivial amount of inconsistency has been introduced into the data. Specifically in cases where integration of the data is so complicated and contradictory that the only reliable information deduced is binary, in the sense that either a response occurred or it did not occur. Furthermore, determining relationships between this inconsistent data is so difficult that using traditional methods of analysis fail to produce substantial results. A real world example of data that meets the above criteria is the Environmental Protection Agency's (EPA) ToxCast and Toxicity Reference Database (ToxRefDB) data programs. These datasets share a set of common potentially toxic chemicals, where ToxRefDB contains a multitude of animal study endpoints based upon exposure to these chemicals and ToxCast contains bioassay responses to those same potentially toxic chemicals (Dix et al., 2007; EPA, 2010a; Judson et al., 2009; Martin et al., 2009a; Knudsen et al., 2009; Martin et al., 2009b; EPA, 2010b). The EPA would like to integrate these data together as to determine which chemicals will cause malady as seen with the animal endpoint studies based primarily upon ToxCast bioassay responses (Dix et al., 2007; EPA, 2010a; Judson et al., 2009). Their reasoning behind this is described below along with a more detailed explanation of the data and how it qualifies as a real world example for our methods.

The efficient testing of chemicals for possible human health effects is a continually growing challenge, with over 83,000 chemicals currently within the Toxic Substances Control Act inventory and over 30,000 in widespread use(Agency, 2011; Judson et al., 2009). Complicating the demand for increased screening of chemicals having industrial and agricultural

importance is the fact that the current processes for testing chemicals is extremely complex, expensive and time intensive, with heavy reliance on animal studies that take 2-3 years and millions of dollars to complete(Judson et al., 2009). With the majority of such chemicals having undergone little to no safety testing, there is a significant need to develop complimentary or alternative approaches to help prioritize both the chemicals to be tested as well as identify the types of tests that will be most informative in the regulatory decision making process.

To help understand and address these challenges, the EPA established the ToxCast program, a high-throughput screening (HTS) effort focused on the development of methods for accurate and cost-effective chemical screening and prioritization (Dix et al., 2007; EPA, 2010a). The initial phase of the ToxCast program consisted of the testing of 309 unique chemicals against a panel of over 650 toxicity-relevant assays. While chemicals chosen for this first effort are comprised largely of food-use pesticide active components, assays vary greatly in the type of technology used, the target measured, as well as the biological context in which the assay is performed. While still in the early stages, programs such as ToxCast and Tox21 are expected to provide the methodological foundations for future sustainable efforts in chemical screening(Dix et al., 2007; EPA, 2010a; Kavlock et al., 2009).

Although providing a wealth of data across a broad spectrum of chemicals, high-throughput approaches as used in ToxCast present their own challenges with regard to data integration and downstream interpretation. There is a great deal of variation in the types of assays used for screening, with associated variation in the levels of quality, sensitivity and specificity. Furthermore, tests are performed in cells or tissues of a number of different species including rat, mouse and human. As our understanding of mechanisms of toxicity for different chemicals is far from complete, methods that can use such data to help establish more integrated pictures of the linkages between chemicals, biomolecular players and disease endpoints are of significant value. Specifically ToxCast data provided the ideal dataset to demonstrate the utility of our methods for integrating inconsistent/noisy datasets in a response and explanatory variable framework where the data is subset to consider only the strongest relationships as an

alternative to considering the entire data record.

Preliminary investigations by both the Reif et al. (2010) using the ToxPi measure and DiMaggio et al. (2010) with their biclustering and logistic regression framework indicate that the data collected from the first round of the ToxCast program is quite sparse, with highly variable amounts of inconsistency within the bioassays data. This is demonstrated by the modest subset of the data that is used in both methods and the small number of important results that are reported as discussed in greater detail below. The measure ToxPi presented by Reif et al. does integrate several sources of data into a measure that ranks a chemical's toxicity with a score (Reif et al., 2010). The difficulty of ToxPi is that it does not indicate which results are statistically significant with regards to chemical toxicity. Moreover, the paper provides scant evidence that the top ranking chemicals are toxic and the bottom ranking chemicals are benign with regards to toxicity(Reif et al., 2010). The methodology of DiMaggio et al. indicate the minimal set of bioassays that maximize the separation between 8 liver and 10 reproductive animal endpoints(DiMaggio et al., 2010). Their framework helps determine which bioassay can be uniquely associated with either liver or reproductive animal endpoints. Yet, their results fail to provide any goodness-of-fit measures for the logistic regression models that were used to determine the association between animal endpoint and bioassay. Furthermore, they are only able to determine unique association between animal endpoint and bioassay for 18 of the over 300 active animal endpoints(DiMaggio et al., 2010). Similarly the ToxPi measure only uses 90 of the over 650 bioassay to create its integrated measure of chemical toxicity(Reif et al., 2010).

The primary goal in this work is to use our methods to account for the underlying inconsistency/noise within the ToxCast data by focusing the analyses on subsets of data. We assume that the desired associations are the most prominent for subsets of the data due to this underlying inconsistency. We integrate the ToxCast and ToxRefDB data to identify association between the explanatory variables (ToxCast bioassays) and response variables (ToxRefDB animal endpoints) with regards to identifying chemical toxicity. Our methods provide statistical

measures that furnish strength of association for the subsets of data and statistical significance that accounts for multiple hypothesis testing. Additionally our methods allow for some approximation/fuzziness to be incorporated into the results in an intuitive manner.

3.1.2 ToxCast Data

The first phase of the ToxCast program involved the utilization of a multitude of *in vitro* high-throughput screening (HTS) assays to 320 chemical substances that represent a unique set of 309 potentially toxic chemicals(Dix et al., 2007). These unique chemicals were selected based upon the conclusions of EPA's previous work(Judson et al., 2009). This work surveyed the primary types of EPA regulated chemicals and compiled a set of non-redundant chemicals to be used within the ToxCast program. The majority of the ToxCast chemicals were related to food-use pesticides(Dix et al., 2007).

The over 600 *in vitro* HTS assay results that compose the ToxCast program are derived from ten different assay technologies (bioassays from ACEA, Attagene, BioSeek, Cellumen, CellzDirect, GeneAssays, Gentronix, NCGC, Novascreen, Solidus)(Dix et al., 2007). These bioassay results depict interactions between the 320 chemicals and molecular targets or cellular events that give rise to measured cellular phenotypes, gene expression, biomarker and transcription factor activity(Dix et al., 2007). For this methodology the bioassay activity is represented as a binary result such that the chemical potency (like EC_{50}, IC_{50}, AC_{50}) or lowest effective level of activity detected beyond normal functioning is considered as chemically active (indicated by one). Non-activity (indicated by zero) indicates that high chemical concentrations resulted in no detectable activity within the bioassay. The primary reasoning for using the bioassay data as a binary response is because of the added complication that integrating bioassays from many different technologies, response types (cellular, tissue, cellular phenotype, gene expression etc), and species would cause.

Current publications that provide a detailed description and analysis of specific ToxCast bioassays include those that analyze Attagene assays (Martin et al., 2010), Bioseek assays

ToxCast & ToxRefDB Data
Active Chemicals
All ToxCast 320
Unique ToxCast 309
All ToxRefDB 294
Unique ToxRefDB..... 283
Active Bioassays
All ToxCast 659
ToxCast with Gene Targets 529
Active Animal Endpoints
Original ToxRefDB 89
Expanded ToxRefDB..... 289

Table 3.1: Quantification of EPA's ToxCast and ToxRefDB data.

(Houck et al., 2009) and Gentronix, Cellumen and NCGC assays (Knight et al., 2009). Unlike these publications, we want to describe frequent chemical activity across all of the ToxCast bioassays that share chemical perturbation by the same chemical substances used in the EPA's Toxicity Reference Database, ToxRefDB, animal study endpoint studies. To simplify the results, chemical activity in bioassays that are measured at different time points are collapsed into a single value that indicates any chemical activity across all time points. Similarly, chemical activity associated to up or down regulation of genes or biomarkers are collapsed into a single value that indicates any chemical activity regardless of regulation type. Through this simplification, the number of ToxCast program HTS assay results is reduced to 659 HTS assays that were at least activated by one of the chemicals. Table 3.1 quantifies the numbers of active chemicals, HTS bioassys and animal endpoints discussed below.

Of the 659 HTS ToxCast bioassays, 529 could be further associated directly to their gene targets. These 529 bioassays represented 258 unique genes, where 202 of those genes could be associated to 175 unique KEGG molecular and cell signaling pathways or Ingenuity molecular and cell signaling pathways. Additionally, many of the gene targets were associated to multiple bioassays; thus, giving a clarifying view of chemical activity across multiple technologies, organisms and cell types. The classification of the ToxCast bioassays by activated pathways, molecular functions or biological processes allows one to get a better scientific

view of the effects of the chemical activity perturbations. The gene targets were identified by the companies responsible for that creating the 659 HTS ToxCast bioassays. These 258 unique gene targets were then used to determine which KEGG and Ingenuity pathways that the gene activation could be associated to.

3.1.3 ToxRefDB Animal Study Endpoints

Of the 309 unique ToxCast chemicals, 289 of them were used within EPA's ToxRefDB(Martin et al., 2009a; Knudsen et al., 2009; Martin et al., 2009b; EPA, 2010b). This program resulted in 86 animal study endpoints that include measurements from 2-year rat and mouse cancer or chronic bioassays studies, rat reproductive toxicity studies, and rat and rabbit developmental toxicity studies. These 86 ToxRefDB animal study endpoints were then expanded to the 289 endpoints to better describe the malady represented by these animal study endpoints. For example, mouse and rat liver lesions are expanded to describe the type/stage of lesion.

The 289 chemicals represent the union of chemical activity of chemicals that were tested by the animal study endpoints; where, the maximum number of chemicals tested on any one endpoint was 257 and the minimum was 245 out of 309 chemicals that were unique to the ToxCast bioassays. Of the 289 unique chemicals tested by the ToxRefDB animal study endpoints, 283 chemicals were chemically active for both ToxRefDB animal endpoints and the ToxCast bioassays. The relationship between bioassay and animal endpoints with regards to chemical activity will be explored through these 283 chemicals. Ideally this relationship should help identify which potentially toxic ToxCast chemicals, as indicated by chemical activity within the ToxCast bioassays, are also associated to detrimental animal endpoints.

3.2 Existing Approaches and Related Work

3.2.1 Modeling Full Data

Some methods of determining association across multiple datasets with inconsistent data commonly involve a bayesian framework. These methods weight the data based on its usefulness in the underlying mathematical model of association as was demonstrated by Webb-Robertson et al. (2009) using metabolomic data. Zhang et al. developed methods to identify significant associations that existed between set of explanatory and response variables for categorical data as in form of chi-square and anova testing of genome-wide association studies (GWAS)(Zhang et al., 2010a,b, 2009, 2008). The exhaustive enumeration and control of error rate associated with multiple testing, makes the TEAM algorithm most comparable in calculation of contingency tables statistics(Zhang et al., 2010a). The primary difference between our methodology and the methodologies used by Webb-Robertson and Zhang is that their algorithms used the entire data record; whereas, our methods focus on subsets of the data more similar to biclustering(Webb-Robertson et al., 2009; Zhang et al., 2010a). Although Webb-Robertson's method controls for inconsistency by weighting the results, the method still uses the entire record as compared to using only subsets of the data. Zhang's TEAM algorithm may work with contingency table statistics not too different than the statistic we use, but their methods only consider the association between pairs of explanatory variables (in their case SNPs) as associated with a response variable classification (in their case phenotype)(Zhang et al., 2010a). In contrast, our methods consider associations between response and explanatory variables that are more complex. Our methods also consider response variables in a multivariate sense (one or more response variables) and explanatory variables that include more than a pair.

Although there is not an association between explanatory and response variables, Reif et al. used ToxCast data to create a measure referred to as ToxPi(Reif et al., 2010). The measure integrates 90 of the over 650 ToxCast bioassays data along with chemical and pathway information for each chemical(Reif et al., 2010). ToxPi uses the entire data record because it

is calculated for all chemicals over 90 of the ToxCast bioassays(Reif et al., 2010). Another important difference is that ToxPi fails to indicate which of the final results are statistically significant regarding the measure of toxicity that it generates(Reif et al., 2010). Our methods consider partial data, and the resulting associations produced are assigned statistical significance related to strength of association that has been adjusted to account for error as associated for multiple testing.

3.2.2 Clustering and use of Partial Data

Clustering is a fundamental method of unsupervised learning that looks to partition one's data in a way so as to highlight meaningful relationships. K-means clustering is a well established form of partitioning one's data into k groups, where k is chosen a priori. Hierarchical clustering provides a potential advantage over k-means clustering because it does not require that one choose the number of clusters, k, a priori. Hierarchical clustering involves an algorithm that continues to divide or merge groups iteratively until all data is divided into single items or all data is merged into a single group. Our method differs from two-dimensional hierarchical clustering, because instead of considering all the data our method can concurrently subset the data by both rows and columns. Similarly, biclustering is able concurrently partition one's data by both rows and columns to consider submatrices, or subsets of the data. Thus, the method of biclustering of binary data is most closely related to our methodology because it also is able to concurrently subset the data by both rows and columns.

Similar to the methods we employed computationally, biclustering works best on sparse data matrices or when heuristics are used to limit exhaustive enumeration of all possible submatrices. This is because the computational complexity of considering full enumeration (all possible combinations) of the data is NP-complete, meaning they have no known polynomial time algorithms and in worst case their runtimes are exponential. This notion of considering subsets of data is explored as a means of accounting for inconsistency within the data while facilitating the discovery of the underlying associations that arise between explanatory and

response variables. When comparing our methods with others, we will reserve direct comparison to biclustering of binary data because it is most similar to primary force driving our method. That being the exploration of subsets of data as a means to deal with integration of extremely inconsistent data.

van Uitert et al. (2008) employ their method of biclustering of spare binary genomic data to identify interacting transcription factors. Van Uitert's method, BicBin, is most similar to our methodology since both consider subsets of binary data(van Uitert et al., 2008). Van Uitert's methods differ from ours because the method greedily gives the highest scoring bicluster as the result. To obtain all the non-overlapping greedily selected biclusters, one must reset the data matrix ones to zeros of the previously discovered biclusters and rerun the algorithm until the data matrix is entirely composed of zeros or can no longer produce biclusters given the input parameters. This differs from our method because our method provides a full enumeration of all subsets (overlapping or not) that meet the minimum row threshold and composition criteria. Thus, our methodology provides more relevant subsets because they meet input criteria and provide a statistic that measures strength of association between the explanatory and response variables. BicBin only provides a score as the measure as to how useful a resulting bicluster is, there is no statistic to assess quality of biclusters that result from their method. Conversely, our methods provides a measure of statistical significance that has been adjusted for error that results from multiple testing. Finally, unlike the results from our methods, there is no guarantee the biclusters produced by van Uitert's method contain an association between response and explanatory variables. These three differences make BicBin a difficult algorithm to use if one's primary purpose is to determine the most important biclusters based upon their association between response and explanatory variables.

DiMaggio et al. (2010) explore the use of logistic regression to identify association between sets of explanatory variables and a response where the variables used in the logistic modeling scheme had been reduced through the use of biclustering. This type of biclustering differed from our methods because it was used as a data reduction methodology to determine

which sets of explanatory and response variables should be used in the logistic regression models. DiMaggio did produce results that gave the most optimal association between 18 of the endpoints and a set of bioassays from the ToxCast data, but they failed to provide a measure of goodness of fit to assess the quality of their optimal models. Moreover, their resulting models only include a small portion of all possible animal endpoints (18 out of over 300) and did not consider multivariate association with regards to the response variables (animal endpoints). Conversely our methods considers all associations given a row threshold and provides clear assessment of quality of the results through the error adjusted significance test.

As stated above our methods use closed and approximate itemset mining to identify subsets of data with association between response and explanatory variables. The process will described in greater detail in Chapter 4, below is a brief discussion of how our methods differ from the standard implementations. The closed frequent itemset mining methodology used builds a depth first tree that is pruned using the minimum support threshold based upon the Apriori algorithm for frequent itemset generation (Han and Kamber, 2006). The methods employed include techniques for efficient tree traversal similar to ones described by Wang et al. (2006) in their paper that details their algorithms for discovering closed cliques. These techniques include adding response and explanatory variables of higher frequency first into the depth tree, determining potential children as only response and explanatory variables of same or less frequency, and pursuing only closed itemsets in the tree traversal Wang et al. (2006). The main distinction from Wang et al. and others is that our algorithm requires that all itemsets within the depth tree contain at least one response and explanatory variable. Thus, adding additional efficiency to the algorithm by generating only closed sets that will have an association between response and explanatory variable. This methodology gives the user the ability to target one's analysis based upon itemset composition and is unique to our algorithm, and enables the user to focus the analysis upon subsets that contain only certain response and explanatory variable combinations. Our algorithm first determines all possible response variable combinations which allows the user to then focus their analysis based upon these

initial results. Our algorithm consider closed frequent itemset as opposed to frequent itemsets because it maximizes the number of response and explanatory variables (items) that are associated with the set of observations (transactions) and it allows us to incorporate fuzziness (approximate itemsets) into the result.

To help address the challenge of identifying association between response and explanatory variable while accounting for the variability within the explanatory variables, our method allows for unbiased integration of fuzziness into the results. An approximate (fuzzy) frequent itemset can be defined as one that includes explanatory variables that might not fully be supported (be ones) for all observations (transactions) included in a particular itemset (set). In this way, approximation of the itemset is restricted to explanatory variables because the imprecision of the itemsets (in our examples) can be primarily be attributed to the variability within the explanatory variables. Allowing for approximate frequent itemsets provides a more complete view of the subsets of data in the presence of the variability (noise) within the data. Furthermore, approximate frequent itemsets provide distinguishing features associated with the discovered itemsets by allowing for a larger number explanatory variables to be included within the resulting sets.

To create the approximate frequent itemsets, our algorithm identifies the maximal frequent itemsets (leaf nodes) and in a top-down fashion collapses these sets with the closed frequent itemsets located levels above using the row and column constraints similar to those described by Cheng et al. (2008), AC-Close algorithm. This collapse consists of taking the union of a maximal frequent itemset with a closed set located levels above and accepting the resulting set if it meets the row and column constraints on allowable fuzziness. This collapse considers all combinations of sets that have connections within the lattice and the collapse continues up the levels of the lattice until row or column constraints are broken or the root set is reached. Inclusion of an approximate frequent itemset within the results concludes with the removal of its founding closed frequent itemsets from the results.

The methods we employ differ from the AC-Close algorithm in the following ways: ap-

proximate itemset creation is only initiated using maximal frequent itemsets, our algorithm considers all combinations of closed sets that meet row and column constraints during the union process, and our algorithm has core pattern factor, α, set to one instead of allowing a range. These modifications provide more appropriate approximate sets given that our results are associated with a statistic of association. Specifically, our algorithm considers approximate frequent itemset formation originating solely from maximal itemsets (leaf nodes) because these nodes will most likely result in the formation of approximate frequent itemsets given more stringent column and row constraints. Moreover, the process is likely to collapse itemsets that contain a good deal of overlap together into a single approximate itemset, which helps simplify the results. Because of the uncertainty on what provides the optimal union of closed sets when forming approximate itemsets, our algorithm allows all combinations to occur and relies upon the statistic of association to determine which of the resulting itemsets are most significant.

Liu et al. (2006) created AFI, an approximate frequent itemset mining algorithm, that uses row and column error control similar to AC-Close and our algorithm. The primary difference between AFI, and our algorithm is that is it a bottom-up as opposed to top-down algorithm. The result of this difference is that AFI does not scale on datasets that include a large number of response and explanatory variables. The efficiencies that were programmed into closed frequent itemset mining portion of our algorithm allow for hundreds of response and explanatory variables to be included dependent upon the density of the input data matrix. Our methods then create the approximate itemsets using the resulting closed frequent itemsets in a breath-first, top-down methodology that entails taking the union of the closed itemsets. Unlike AFI, this type of methodology is scalable given hundreds of explanatory and response variables. Both methods produce overlapping approximate itemsets, but our method is more efficient based upon the top down methodology and the use of closed itemsets. As described above with our comparison to the AC-Close, our approximate itemset algorithm also provides more appropriate approximate itemsets when focusing upon subsets with association between response and

explanatory variables.

3.3 Challenges

Identification of relationships that exist amongst data when the data under consideration comes from multiple data sources, experiments or where a nontrivial amount of noise has been introduced into the data can prove to be difficult using traditional methods of analysis. Typically such cases arise when there exists variable signal to noise ratio within a single data source or across data sources; thus, considering the entire data record (using all the data) is misleading due to the underlying noise which will mask relationships that exist amongst one's data. One way to deal with such an issue is to use Bayesian methods to weight the reliability (absence of noise) of the data included in the model as was demonstrated by Webb-Robertson et al. (2009) using Metabolomic data. Another way of dealing with this problem is to subset one's data down to subsets that contain the strongest relationships. This approach was partially explored by DiMaggio et al. (2010) when they used biclustering coupled with logistic regression to identify association between an optimal set of explanatory variables and their corresponding response variable. DiMaggio et al. used biclustering to remove redundancy amongst the explanatory variables and to determine two most anticorrelated classes of response variables. Using biclustering in this manner allowed them to select sets of variables that would be most amenable for use with their logistic regression models. Similar to DiMaggio et al. we look to subset the data down to subsets where the relationships between the data are the strongest. Unlike DiMaggio et al. we create subsets by considering both explanatory and response variables simultaneously; thus, producing results that are most likely to show the true underlying relationships that exist between the data.

3.3.1 Noisy/Inconsistent Data

Identification of association between response and explanatory variables can prove to be challenging when the analysis involves the integration of data from multiple data sources or where there are potentially complicated associations between data due to underlying differences in the data. Typically integration of such data proves to be rather ineffectual using classical methods of analysis because there exists variable signal to noise ratio within a single data source and/or across data sources. Therefore, considering the entire data record can be misleading due to the underlying noise which will mask relationships that exist within one's data. To deal with the challenge of integrating such inconsistent/noisy data to discover underlying relationships that exist between response and explanatory variables we explore focusing on subsets of the data instead of the entire data record. Moreover to deal with the challenge where the numeric association between ones' data is complicated and contradictory, the data was simplified to a binary response. This provided the most reliable information as it simplified the numeric information to whether or not a response occurred at any quantification. Additionally, one can use the minimum row threshold to reduce the likelihood of finding false positives and to require relationships that are true for a reasonable portion of the data. Finally, we provided a means to allow some user defined fuzziness or approximation to be incorporated into the resulting subsets of data in both the row and column dimensions.

With our real world example, the integration of the bioassays of ToxCast proved challenging because of the complicated association amongst the bioassays. As a demonstration of the variation in bioassay type, one type of ToxCast bioassays are Attagene bioassays that measure cis-acting and trans-acting transcription factors using human liver cells(Martin et al., 2010). Another example is BioSeek bioassays that detect different biomarkers and risk factors, like cytokines, chemokines, growth receptors, biological mediators and enzymes, from various human primary cells, such as endothelial, smooth muscle and fibroblasts(Houck et al., 2009). The activity of a chemical perturbation within a bioassay is expressed as a potency (like EC_{50}, IC_{50}, AC_{50}) or lowest effective level of activity detected beyond normal function-

ing. Non-activation can be interpreted as high chemical concentration resulted in no detectable chemical activity. This means even the numeric values of chemical activity have complicated association because of the variety of ways they were expressed as a potency and the variation of amount of chemical needed to cause a response. These problems of association between bioassays can be attributed to the variation of technology and organism type within bioassays and differences in how chemical perturbation was measured. Furthermore these problems with association caused variation in precision, accuracy and introduced more ambiguity into the data integration of the bioassays. Attempting to associate the bioassays with ToxRefDB animal endpoints is a challenging task because many of the bioassays utilized human tissues and cells; whereas, the endpoints were derived from rats, mice and rabbits. Although used as animal models of humans systems rats, mice and rabbits do not always respond biologically similar to humans(Mestas and Hughes, 2004; Clemencet et al., 2005; Gibbons and Spencer, 2011). Recall that the preliminary investigations by both the Reif et al. and DiMaggio et al. indicate that the ToxCast is quite sparse, with highly variable amounts of inconsistency especially amongst the bioassays as demonstrated by the modest subset of the data that is used in both analyses and the small number of important results that are reported.

3.3.2 Large Datasets

Integration of data from multiple data sources to identification of association between response and explanatory variables can prove to be challenging because it typically involves trying to analyze large datasets. With traditional analyses this may prove to be less problematic but when looking at subsets of data depending on the minimum row threshold and density of the data the number of combinations one might have to consider can become computationally prohibitive. The number of all possible subsets based upon the number of columns is given by $2^c - 1$, where c represents the number of columns in a data matrix. Given a data matrix with few columns as depicted in figure 3.1, the number of subsets is relatively small, but for even 30 columns all possible subsets will be over a billion different subset combinations. To

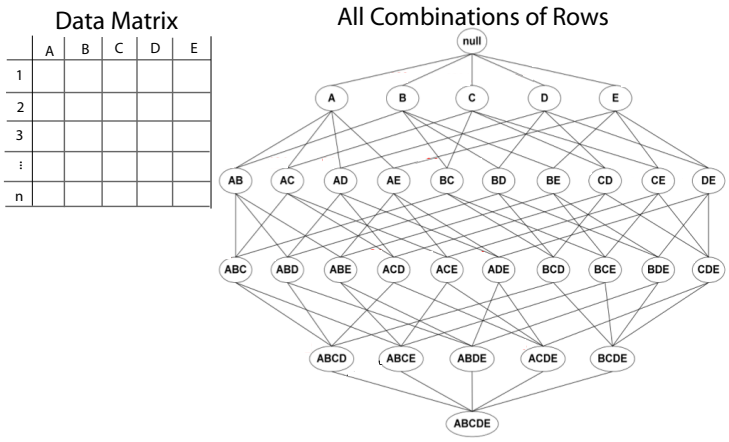


Figure 3.1: Subset Combinations. Data matrix of n rows and 5 columns depicted on left, results in 31 different subsets based upon all combinations of columns as shown on right.

help address this challenge one would want to set the minimum row threshold to consider a reasonable portion of the data as to provide relationships that are most likely true in addition to reducing the number subsets considered. Moreover one would want to require that all relationships consider are only ones that contain at least one response and one explanatory; thus, further reducing the number of combinations of rows that would need to be considered. Obviously, the more sparse the dataset the fewer number of subsets that the algorithm has to consider, but in cases of dense or large datasets one can consider further limiting subsets that are considered in the analysis. Three plausible techniques are limiting the analysis to only subsets that involve certain response variables, limiting the analysis to subsets that are multivariate in the sense that they contain two or more response variables, and perform a round of data reduction where columns that are highly correlated are combined prior to analysis. The final option for dealing with very dense data is to run the algorithm on the negative space (zeros) since they should be sparse given that the positive space is dense. Although ToxCast dataset was composed of over 900 columns it was sparse enough that the minimum row threshold was set at 40 rows, roughly a support threshold of 0.16.

3.3.3 Prioritization of Results and Multivariate Association

The number of subsets that support an association between response and explanatory variables can be quiet large; thus, making a challenge to prioritize the importance of the resulting subsets. The statistic that quantifies the association between response and explanatory variables can be used to rank the results. A better way to prioritize is to provide a p-value as a means to determine within the type-I error, the number of subsets that reach that significance level with regards to the statistic that provided the strength of association between response and explanatory variables. Recall in hypothesis testing the p-value is the probability of obtaining a test statistic at least as extreme as the one observed under the assumption that the null hypothesis is true. Moreover, the probability of committing a type-I error is the significance level of the hypothesis test, it indicates the probability of accepting false positive results. Because all results are derived from the same data sources and we test each one for significance, we must provide a means to control for the increase in type-I error that occurs when performing multiple tests. This is simply done by adjusting the p-value using the False Discovery Rate (FDR) (details about the adjustment can be found in Benjamini and Hochberg (1995) paper). There exist frequent itemset and biclustering methods, like BicBin, that will create subsets of data given a binary dataset and set of threshold criteria, and a few, like BicBin, will rank the importance of the resulting subsets. To the best of my knowledge there are not any that also provide significance adjusted for multiple testing as well as a statistic that indicates the strength of the relationship between response and explanatory variables.

A multivariate response is defined by an association that includes more than one response variable. Identification of association between response and explanatory variables where there exists at least two or more response variables can prove to be challenging using traditional means of analysis. Typically one would need to know in advance which response variables might be associated with each other or attempt to determine this prior to modeling. By reducing the problem to one of binary data where one searches for all patterns of association that involve response and explanatory variables, one can easily identify all patterns that involve a

multivariate response as long as the minimum row constraint is met.

Chapter 4

Mining for Association

4.1 Approach

Identification of associations that exist between response and explanatory variables when the data under consideration comes from multiple sources or where a nontrivial amount of noise has been introduced into the data makes using classical methods of analysis difficult. When there exists a variable signal to noise ratio within a single data source or across data sources, considering the entire data record (using all the data) is misleading due to the underlying noise (inconsistency) which masks true associations that exist within one's data. One way of dealing with this problem is to find subsets of data that contain the strongest associations. As mentioned in Chapter 3, biclustering methods enable one to subset the data simultaneously by rows and columns. Consider the case where the numeric association between the data is so complicated that attempting to quantify the level of response beyond a binary sense (event occurred or it did not) just confounds the association between explanatory and response variables. If dealing with binary data, closed frequent itemset mining can be used to subset the data down by rows and columns simultaneously to discover all patterns of association between the data given a minimum row/support threshold. A measure of the strength of association is applied to the resulting subsets to determine which are most important with regards to the relationships between response and explanatory variables. Adjustments to the data mining

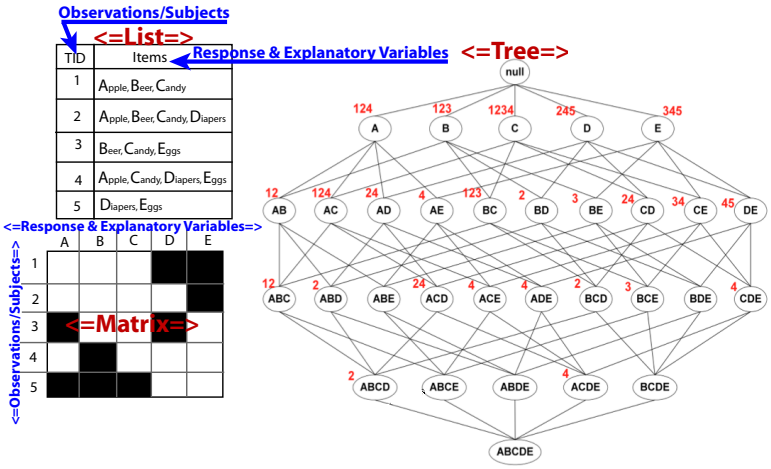


Figure 4.1: Subsetting Binary Data with Frequent Itemset Mining. The market basket list on the left contains transaction ids (TID) and their corresponding list of grocery items purchased for each transaction. This list is transformed into observations/subjects and their corresponding list of response and explanatory variables that have a response(one) for that observation/subject. The market basket list corresponds to the data in the binary matrix below it, such that column D(Diapers) has a value(is listed) for rows(transactions) 2, 4 and 5. To find all subsets, this data is transformed into the lattice on the right such that each node in the lattice is labeled by a column(s) (Item) and each node contains an associated list of rows(transactions) where that column(item) has a value of one(white cell). This lattice displays all 31 possible subsets, even though only 25 contain data. The data structures can hold response and explanatory variables and observations/subjects instead of market basket data.

methods allow user defined amounts of fuzziness/approximation to be incorporated into the results to enable them to be more tolerant of the underlying noise within the data.

Figure 4.1 shows how the typical market basket example of frequent itemset mining can be associated with a binary matrix of data such that the columns of the matrix represent the 'items' and the columns represent the 'transactions' of that market basket. The grocery items that exist for a transaction correspond directly to a cell in the binary matrix that contains a one (white cell). On the right is a lattice that provides all possible combinations of the columns of the binary matrix; thus, providing all possible subsets of the binary data matrix. Notice each node in the lattice is labeled by one or more columns (items) and that is associated with a list of rows (transactions - red numbers beside the node) for which this column(s) contain a one. As stated in the previous chapter the number of all possible subsets of the binary data matrix can be calculated by $2^c - 1$, where c represents the number of columns in the matrix. The

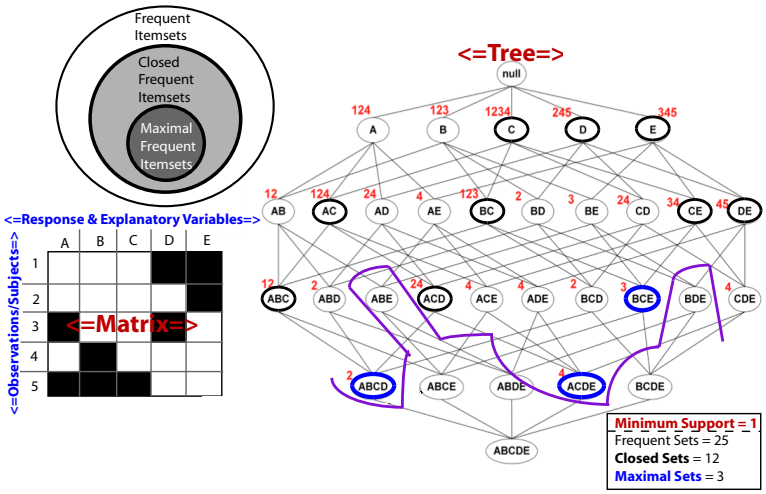


Figure 4.2: Itemset Mining Definitions. Frequent Items sets are those that meet minimum support threshold (minimum number of transactions/rows). The 25 nodes above the purple line represent frequent itemsets when the minimum support threshold is set to 1 transaction. Closed Frequent Itemsets are a subset of all frequent itemsets; thus, frequent Itemsets where none of the immediate supsersets (children) contain the exact same support transaction/row list are closed frequent itemsets. The 12 shown are nodes enclosed by a bold black oval. Maximal Frequent itemsets are a subset of closed frequent itemsets; such that, any closed itemset that has no frequent supersets (children) is a maximal frequent itemset. The 3 shown are the nodes enclosed by bold blue oval.

number of subsets that can be created for any given data matrix depends upon the density of the data; notice now in figure 4.1 only 25 of the 31 nodes contain data. The market basket data can be easily replaced by data that represents explanatory and response variables as the columns/items and the observations/subjects as the rows/transactions.

The explanatory and response variables were chosen to represent the items (columns) such that all possible combinations of these variables could be explored for relationships that exist between them. Moreover in frequent item mining the number of subsets explored is limited by a row/transaction threshold. This is a desired property because it limits the subsets considered to the number of observations that support that subset of data. We feel that a reasonably sized threshold will help guarantee the results are more likely to be true relationships within the data and not due to random chance. Specifically, only subsets (nodes in the lattice) meeting the minimum support threshold (minimum number of transactions/rows) will be explored.

To aid the explanation of the methods we used a few concepts of data mining diagramed in

figure 4.1. Notice the tree representation in figure 4.1 begins with a *null*(empty set) root node with the nodes directly connected below defined as its children nodes. Thus, the root node is the parent node of each of its children. Excluding the root node, child nodes are always supersets of their parent node; thus, they will have at most the same transaction(row) support if not less support than their parent node. This means that if a node does not meet the minimum support threshold then none of its children will meet the support criteria either. Thus, traversal of the tree down that branch can be stopped (pruned) since it will not result in more subsets being added to the results. This concept of pruning based upon a downward closed property is known as the Apriori algorithm for frequent itemset generation in data mining(Han and Kamber, 2006).

This concept applied to the tree in figure 4.1 produces the 25 frequent itemsets generated when minimum support threshold is set to one transaction, as depicted as the nodes above the purple line. Closed frequent itemsets are a subset of frequent itemsets; such that, a frequent itemset where none of its immediate supersets (children) contains the exact same support transaction list is a closed frequent itemset. The 12 closed frequent itemsets in figure 4.2 are encircled by a black oval. Additionally, maximal frequent itemsets are a subset of closed frequent itemsets; thus, a closed frequent itemset who has no frequent supersets (children) is a maximal frequent itemset. Notice in the tree graphic in figure 4.2 the 3 maximal frequent itemsets encircled by a blue oval can be viewed as the leaf nodes of the tree.

The methods that we employed to find subsets of data to explore the relationship between explanatory and response variables involved finding all closed frequent itemsets given a minimum support threshold. Our methods focused upon closed itemsets because closed itemsets have the property that none of their immediate supsersets (children) contain the exact same support transaction list; thus, maximizing the information (number of explanatory and response variables) given for a set of observations/subjects. This provides more information that can help validate the resulting subsets using outside literature. Furthermore using closed frequent itemsets allows for user defined amounts of of fuzziness/approximation to be in-

corporated into the results, which enables the results to be more tolerant of the underlying inconsistency within the data.

4.1.1 Closed Frequent Itemset Mining for Association

The methods we employed to find all closed frequent itemsets for a given minimum support threshold involved using a depth first tree to along with a few modifications to improve efficiency. To begin with the items (here response and explanatory variables) in the binary data matrix are sorted and the most frequent are added into the tree first. In addition, potential children of a newly added item can only be derived from items that occur after that newly added item in the sorted list of items. Looking at the tree in figure 4.3 this means that item A would have the highest support as compared to items B-E to adhere to the rule that most frequent items are added first to the tree. Notice that this is not the case in our cartoon example, item C is the most frequent and would have been listed as the first node if this tree had adhered to the methodology that we used. Although not true for the example tree in figure 4.3, imagine that the frequency of items A-E resulted in a list that was in alphabetical order when sorted from most frequent to least frequent item (item A was most frequent and E least). Thus using the next rule for efficiency item A's potential children are items B-E; whereas, item B's potential children are items C-E. Using such a rule results in visiting each of all possible subsets (nodes) only once during tree traversal. In figure 4.3 the blue lines (dark blue solid and light blue dashed) indicate the tree traversal that would visit each of the 25 frequent itemsets once based upon this rule. These two techniques for efficient tree traversal are similar to ones described by Wang et al. (2006) their paper that details their algorithm for discovering closed cliques.

The next efficiencies on tree traversal apply only to algorithms that find closed frequent itemsets because they prune certain branches of the tree once it is determined that an itemset (node) is known to be **not** closed. Pursuing only closed itemsets (nodes) in the tree traversal is similar to the methods described by Wang et al. (2006) in their paper that detailed their

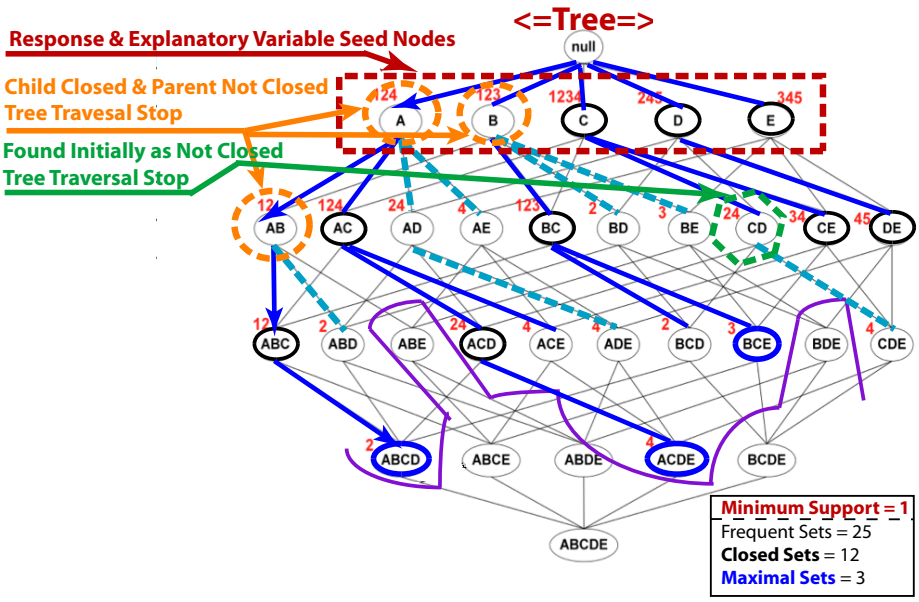


Figure 4.3: Efficiencies of Closed Frequent Itemset Mining. The red rectangle around the first level nodes highlights the seed nodes that are required to contain at least one response and one explanatory variable. The orange ovals around 3 of the nodes shows where traversal was terminated once a child node determines its parent node is not closed. The green hexagon depicts where traversal was terminated once a node is found to be not closed based upon previously defined closed itemsets.

algorithm for discovering closed cliques. Putting the most frequent items first in the tree increases the likelihood of finding most of the closed itemsets earlier in the process. Closed itemsets can not be determined until all of a node's children have been processed; thus, if an itemset is determined to **not** be closed then traversal can stop because its remaining children will also **not** be closed.

Looking at figure 4.3, the light blue dashed lines indicate where a branch was pruned (tree traversal was halted) because the parent node was determined to not be closed by one of its children. The three cases of this in figure 4.3 occur where the parent nodes are encircled by an orange dashed oval. Normal depth first tree traversal using figure 4.3 would follow the dark blue arrows beginning at the root node and proceed to the node labeled A, next to node labeled AB until reaching node labeled ABCD. Since node ABCD has no children but meets the minimum support threshold (1 transaction) it is the first closed set added to a hash table that contains all closed sets. Next processing back up the branch it is determined that

ABC is also a closed set since it doesn't share the same transaction set with the only child it has. Moreover, since the next node AB shares same transaction set with its child ABC it is determined to **not** be closed thus further traversal of its children is halted.

The next efficiency is also related to checking if an itemset is **not** closed in order to prune branches early. The check involves determining an itemset's closed status based upon the closed frequent itemsets that have already been added to the results prior to adding a node to the tree. This check is done prior to adding a node into the tree as to prevent the work of having to process that node's children since they also will not be closed itemsets. This prevents unnecessary tree traversals as depicted as the green dashed hexagon in figure 4.3 where node CD was determined to be **not** closed because of closed frequent itemset ACD with identical transaction list.

We have modified this algorithm to focus its subset finding on only those subsets that show a relationship between response and explanatory variables. The first level nodes in the tree must contain at least one response variable and one explanatory variable as shown in figure 4.3. Assuming that the set of response variables is smaller and more sparely populated but more consistent with regards to variability we use this set of data to seed the first level tree nodes. Specifically we create all closed frequent itemsets that involve one explanatory variable and as many of the response variables as possible as long as the minimum support threshold is met. The results are collected as the seed nodes; they contain the response variable(s) and their children as the list of the explanatory variables that can form a potential closed itemset with that response variable(s). The ability to target one closed itemset finding based upon itemset composition using the seed nodes is unique to our algorithm, and enables the user to focus the analysis upon only subsets that contain two or more response variables or specific response and explanatory variable combinations.

The process to form the seed nodes involves running a modified version of our closed frequent itemset mining algorithm. First all explanatory variables are sorted by frequency and only kept if they meet the minimum support threshold. They are treated as the first level nodes

in the tree. Their children are all the response variables sorted by frequency and kept only if they in combination with the explanatory variable assigned to the first level node meets the minimum support threshold. Next each first level node is run through our closed frequent itemset mining algorithm to determine all closed sets formed from that one explanatory variable and all of its response variable children. The results of running the closed frequency itemset mining algorithm on all first level nodes are collected in a hash table. The hash table's key is the response variable(s) and the hash value is a list of all explanatory variables that form closed sets with the hash key. Each hash key will be used to form the seed nodes that make up the first level nodes of the tree our closed frequent itemset mining algorithm. The children of each of those first level nodes are composed of the hash value, the list of explanatory variables that formed a closed set with the response variable(s) hash key.

This assumption that the set of response variables is smaller, more sparely populated and more consistent with regards to variability holds true for our real world example of ToxCast data and in most typical modeling schemes. Generally researchers collect fewer final response (endpoint) variables and collect them with more care because of their critical impact with regards to the results of their research. Conversely with explanatory variables, researchers attempt to collect many and may add in ones of questionable consistency in their efforts to help ease the explanation of what is seen with their response variable(s). Many researchers assume that useless explanatory variables will be determined and removed through the modeling process. Often they have to guess what possible explanatory variables might influence their response variables apriori based upon scant data provided by pilot studies and previous research.

The final improvements to the algorithm enable it to quickly search through the closed frequent itemsets when checking if a frequent itemset is defined as **not** closed because its items are a subset of the items in a closed frequent itemset and also have an identical transaction list. This part of the algorithm, dependent upon the number of closed frequent itemsets already defined, can be time consuming. Specifically these efficiencies pertain to limiting the existing

closed itemsets hash table to only those with potential overlap with the frequent itemsets and creating an efficient hash key for this check.

When finding the seed nodes this efficiency involves addressing each explanatory variable, meeting the minimum support threshold, separately finding all combinations of response variables that formed closed sets with the explanatory variable. Specifically with seed finding, the hash table that holds the resulting closed frequent itemsets can be recreated anew to collect closed itemsets for each explanatory variable; thus, limiting the number of applicable closed sets one must process through when determining if a frequent itemset is closed. This is because each resulting seed node must contain a single explanatory variable to be relevant; therefore, no two seeds spawned from different explanatory variables will ever be subsets of each other.

When determining closed sets using each of the seed nodes, this efficiency involves creating an efficient hash key and the observation that only certain seed nodes will have overlapping response variables. Seed nodes will overlap in response variable depending on whether or not their response variables are subsets of each other. Therefore, once the seed nodes are defined our algorithm builds a mapping to identify which seed nodes are subsets of each other. This minimizes the number of closed sets one must consider when determining which frequent itemsets are closed. Moreover, this part of the algorithm builds a hash key that consists of the response variables and the number of items that the closed set contains. This enables a more efficient search when determining whether or not a frequent itemset is closed because this check should only consider certain seed nodes that the itemset is a subset of and only closed sets that contain at least one more item than the frequent itemset.

One can visualize the outcome of our algorithm's closed itemset mining by the cartoon in figure 4.4. The left-hand side of figure 4.4 depicts a binary dataset, where ones are white and zeros are black. The yellow boxes represent subsets of the data that have been identified through data mining of the binary matrix for closed frequent itemsets (pink boxes). Our closed frequent itemset mining algorithm identifies the pink boxes where response and explanatory

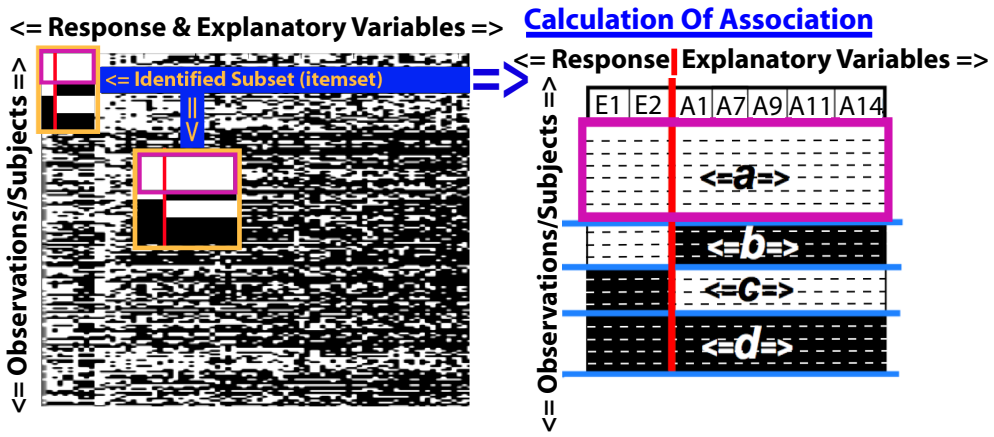


Figure 4.4: Closed Frequent Itemset Mining to Identify Subsets within Binary Data. The left-hand side depicts the discovery of closed frequent itemsets (pink boxes) to identify subsets of data (yellow boxes). The right-hand side details how identified subsets of data are composed of four states of association, ***a***-***d***, between the response and explanatory variables.

variables are both active (ones). Once these closed itemsets have been identified, the algorithm determines the observations that make up the other three of the four possible states of the response and explanatory variables that are defined by the closed itemset. The right-hand side of figure 4.4 depicts the four states, ***a***-***d***, of the subset identified by closed frequent itemset mining for response and explanatory variable activity (pink box). The red line indicates the division between response and explanatory variables. The number of observations that determines a closed frequent itemset, ***a***, is at least the minimum support threshold. The number of observations that compose the other three states, ***b***-***d***, is determined by the data.

The use of closed frequent itemset mining to determine subsets of data means that our algorithm will produce overlapping closed frequent itemsets. Figure 4.5 depicts two response and five explanatory variables (the seven rows) over a number of observations (columns) that are included in one of the three closed frequent itemsets (colored boxes labeled by ***1***-***3***). The colors within the boxes that denote the closed frequent itemsets indicates the activity overlap between the three itemsets. The light green shade indicates observations, response and explanatory variables are active (ones) for all three closed frequent itemsets, light teal indicates activity for two of the three closed itemsets, and light blue indicates activity for only one of

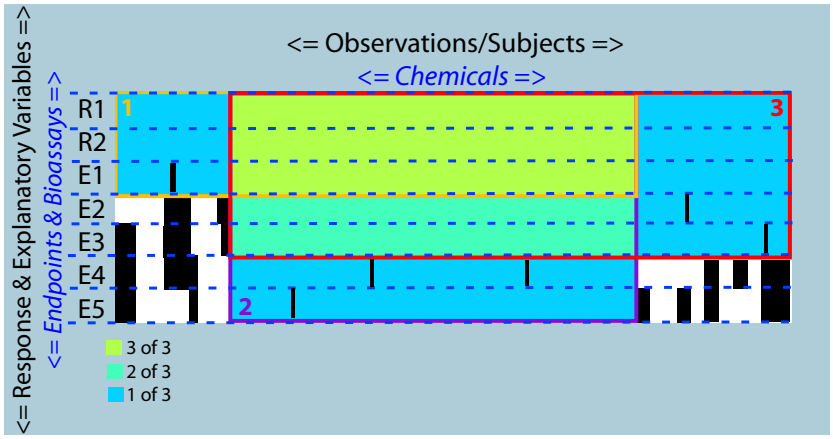


Figure 4.5: Overlapping Closed Itemsets. Rows indicated the response, ***R1***,***R2***, and explanatory variables, ***E1***-***E5***, columns the observations for which these three closed itemsets as indicated by the three colored boxes include, ***1***-***3***. The color within the itemsets indicates the activity (ones) overlap between them where light green shows active (ones) observations for response/explanatory variables common to all 3 itemsets, light teal indicates active observations common to 2 of the sets, and light blue indicates active observations common to 1 of the sets. The black cells within the itemsets indicates inactive observations that are included because it is an approximate itemset. The black and white cells located outside of the itemsets indicates explanatory variable activity for observations outside of the 3 closed itemsets.

the three closed itemsets. The black shade indicates inactivity and white indicates activity but not membership to any of the three closed frequent itemsets. The black cells within the three closed frequent itemsets indicate that these are approximate frequent itemsets whose formation will be discussed in the next section.

All of the efficiencies mentioned above exist in the pseudocode described in detail in Algorithms 5, 6, and 7. Algorithm 5 creates the initial seed nodes and Algorithm 6 uses those seed nodes to create a depth-first tree that is used to determine all closed frequent itemsets that meet minimum support threshold and response/explanatory variable criteria. Algorithm 7 is used by both algorithms 5 and 6 as the recursive function that process each node in the depth-first tree that finds all closed frequent itemsets. Notice that the resulting hash tables passed into and out of Algorithm 7 are pass-by-reference so that results can be added and accessed during the recursion. Additionally, the input *SeedMap* is only input by Algorithm 6 because this algorithm uses the efficiency to only search closed itemsets where a possible overlap can occur when determining a frequent itemsets closed status. Finally the exact form of the output

of Algorithm 7 is dependent upon the algorithm in which this function is called.

Algorithm 5 FindSeeds (Finds Response Variable Seeds)

Input: $SeedNodeHash = \{(\{R_{x1}, \cdots\}, [E_{x1}, \cdots, E_{xL}]), \cdots, (\{R_{y1}, \cdots\}, [E_{y1}, \cdots, E_{xO}])\}$, Sorted List Explanatory Variables $E' = \{E_1, E_2, \cdots, E_M\}$, Minimum Support Threshold S

Output: $SeedNodeHash$ = $\{(\{R_{x1}, \cdots\}, [E_{x1}, \cdots, E_{xL}]), \cdots, (\{R_{y1}, \cdots\}, [E_{y1}, \cdots, E_{xO}])\}$, where key = $\{R_{z1}, \cdots\}$ and value = $[E_{z1}, \cdots, E_{zP}]$) pair of Hash Table.

$SeedNodeHash \leftarrow \emptyset$

for *every* $R_i \in R'$, **do**

  $ExplanatoryVHash \leftarrow \emptyset$;

  $E'' \leftarrow returnsChildList(R_i, E')$ where $E_j \in E'$ and $E_j \in E'' \mid support(R_i, E_j) \geq S$;

  $processNode(R_i, E'', S, ExplanatoryVHash, \emptyset)$;

  $SeedNodeHash \leftarrow addsResults(ExplanatoryVHash)$;

end

return $SeedNodeHash$.

Algorithm 6 FindClosedFreqItemsets (Finds Closed Frequent Itemsets)

Input: $SeedNodeHash$ = $\{(\{R_{x1}, \cdots\}, [E_{x1}, \cdots, E_{xL}]), \cdots, (\{R_{y1}, \cdots\}, [E_{y1}, \cdots, E_{xO}])\}$, Minimum Support Threshold S, Hash Maps Overlap between Seeds $SeedMap$

Output: $closedFreqSetHash$ = $\{(\{R_{x1}, \cdots\}, \{E_{x1}, \cdots, E_{xL}\}), \cdots, (\{R_{y1}, \cdots\}, \{E_{y1}, \cdots, E_{xO}\})\}$,where key = $\{R_{z1}, \cdots\}$ and value = $\{E_{z1}, \cdots, E_{zP}\}$) pair of Hash Table.

$closedFreqSetHash \leftarrow \emptyset$

for every $Key_i \in SeedNodeHash$ **do**

  $E'' \leftarrow returnsChildList(Key_i, Value_i)$ where $E_j \in Value_i$ and $E_j \in E'' \mid$ $support(Key_i, E_j) \geq S$;

  $processNode(Key_i, E'', S, closedFreqSetHash, SeedMap)$;

end

return $closedFreqSetHash$.

Algorithm 7 processNode( )

Input: ParentNode P, ChildList $E' = \{E_1, E_2, \cdots, E_M\}$, Minimum Support Threshold S, Pass-By-Reference $ResultsHash$, Hash Maps Overlap between Seeds $SeedMap$

Output: Pass-By-Reference $ResultsHash$ = $\{(\{KEY\}, \{VALUE\}), \cdots, (\{KEY\}, \{VALUE\})\}$

$done \leftarrow 0$;
$stop \leftarrow length(E')$;
$i \leftarrow 1$;
while $done$ = 0 and $i \leq stop$ **do**
  if i = 1 and $IsNOTClosed(P, E_i)$ **then**
    if $stop$ = 1 **then**
      $nChildren \leftarrow \emptyset$;
      $cNodeId \leftarrow getNodeId(P)$;
      $done \leftarrow 1$
    else
      $i = i + 1$;
      $nChildren = -1$;
    end
  else
    $cNodeId \leftarrow addNode(P, E_i)$
    $E'' \leftarrow returnsChildList(cNodeId, \{E_{i+1}, \cdots\})$ where $E_{i+1} \in E' \mid$
    $support(R_i, E_{i+1}) \geq S$;
    $addChildren(cNodeId, E'')$
    $nChildren \leftarrow length(E'')$
    $pNodeId \leftarrow getParentId(cNodeId)$
    if $nodeIsNOTClosed(pNodeId)$ **then**
      $deleteNode(pNodeId)$
    end
    if $nChildren \geq 1$ **then**
      $processNode(cNodeId, E'', S, ResultsHash)$;
    end
    $i = i + 1$;
  end
  if $nChildren$ = $\emptyset$ **then**
    while $cNodeId \neq \emptyset$ **do**
      $nxtNodeId \leftarrow getParentId(cNodeId)$
      $addNodeToResults(cNodeId, ResultsHash)$
      $deleteNode(cNodeId)$
      if $nodeProcessedAllChildren(nxtNodeId)$ **then**
        $cNodeId \leftarrow nxtNodeId$
      end
    end
  end
end

4.1.2 Approximate Frequent Itemsets

To help deal with the challenge of identifying association between response and explanatory variables amongst noisy, inconsistent data our methodology allows for unbiased integration of fuzziness into the results. An approximate (fuzzy) frequent itemset can be defined as one which includes items that might not be fully supported (active/binary value of one) for all transactions included in a particular itemset. For our algorithm, this means that some of the explanatory variables that define an approximate (fuzzy) frequent itemset might not fully be supported (maybe inactive/binary value zero) for some of the observations that are included in that approximate itemset. Specifically, the user defines row and column bounds for fuzziness (minimum proportion of ones for any column or row) for an approximate frequent itemset and the algorithm provides all approximate itemsets within those bounds based upon existing closed frequent itemsets using methods similar to ones described by Cheng et al. (2008). Response variables were excluded from this fuzziness (inactivity) for an approximate itemset because we required a stronger association amongst response variables if they are multivariate (more than one response variable in an itemset). Additionally, with our real world example, there is less inconsistency amongst the response variables as compared to the explanatory variables.

With our real world example, approximateness is restricted to explanatory variables because the imprecision of the results can be attributed to the variability within the explanatory variables. With the ToxCast data given the variability of the technology, cell/tissue and organism type, the bioassay results may contain between bioassay imprecision and experience a weakened signal for certain chemicals. Thus, it is possible that the activity (chemically active or inactive) of some of the bioassays might have been incorrectly classified. Allowing for approximate frequent itemsets provides the sets of bioassays and endpoint(s) that share chemical activity for a subset of chemicals in the presence of the result variability of the bioassays. Approximate frequent itemsets could help distinguish which pathways were activated by allowing for a larger number of bioassays to be included within the result set, because more

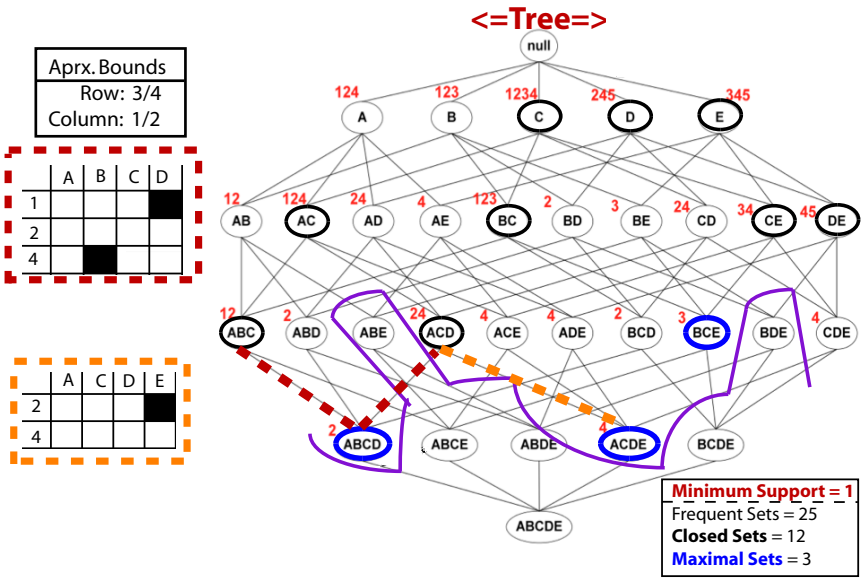


Figure 4.6: Mining for Approximate Frequent Itemsets. Given row constraint of 0.75 and column constraint of 0.50 proportion of ones, the two matrices depict two approximate frequent itemsets based upon the closed frequent itemsets in the tree on the right. The red highlighted matrix is an approximate itemset formed by taking union of closed itemsets *ABC*, *ACD*, and *ABCD*.The orange highlighted matrix is an approximate itemset formed by taking union of closed itemsets *ACD* and *ACDE*.

bioassays provide more information about the itemset. Since many bioassays are involved in multiple pathways, increasing the number of bioassays included within a result set may clarify the activated pathway with this increase in information.

Our algorithm creates approximate frequent itemsets from the resulting closed frequent itemsets based upon user defined row and column constraints on the minimum proportions of ones required for the rows and columns that compose an itemset. The approximate frequent itemsets are created by taking the union of select resulting closed itemsets; such that, the approximate frequent itemsets are added to and the closed itemsets that they originated from are removed from the final results set. Our method focuses solely upon creating approximate frequent itemsets using the maximal frequent itemsets (leaf nodes) in a top-down fashion that collapses these sets with the closed frequent itemsets located levels above as depicted in figure 4.6. This collapse consists of taking the union of a maximal frequent itemset with a set located one level above and accepting the resulting set if it meets the row and/or column constraints.

The collapse considers all combinations of sets that have connections within the the tree and the collapse continues up the levels of the tree until row or column constraints are broken or the root node is reached. Note that the inclusion of an approximate frequent itemset within the results concludes with the removal of its founding closed itemsets from the results.

Looking at figure 4.6 given the row constraint of 0.75 and column constraint of 0.50 with regards to the proportion of ones (white boxes depicted in matrix), this figure shows two approximate itemsets. The first approximate itemset is formed by the union of closed itemsets *ABC* and *ACD* with maximal itemset *ABCD* and depicted by the matrix highlighted with red dashed line. The second approximate itemset is depicted by union of closed itemset *ACD* with maximal itemset *ACDE* and depicted by matrix highlighted with orange dashed line. The closed itemsets textitABC, *ACD*, *ABCD*, and *ACDE* are removed from the resulting closed itemsets since they are now included as part of the two approximate itemsets.

The methods we employ are similar to the AC-Close algorithm described by Cheng et al. (2008) with regards to using row and column constraints to collapse closed itemsets in a top-down fashion breadth-first manor that takes the union of closed sets to form approximate frequent itemsets. One primary difference between AC-Close and our algorithm is that we only consider approximate frequent itemsets whose creation is initiated from a maximal frequent itemsets instead of also including those that originate from closed frequent itemsets that are not maximal. Additionally our algorithm considers all possible combinations of closed sets whom meet the row and column constraints when taking the union of closed sets as they move up to levels closer to the root node. Whereas, the AC-Close algorithm will solely consider the union of all closed sets that meet row constraints as a single unit that will either pass the column constraint and be considered an approximate frequent itemset or fails to meet the constraint and be pruned. In this way our methodology is similar to Liu et al. (2006) AFI algorithm because they both produce overlapping approximate itemsets, as illustrated in figure 4.5. The primary difference between AFI, and our algorithm is that it is a bottom-up as opposed to top-down algorithm. The top-down methodology allows our algorithm to be scalable

for much larger datasets than AFI. The final difference is our algorithm has the core pattern factor, α, set to one instead of allowing α to range between one and zero.

Our algorithm considers approximate frequent itemset formation originating solely from maximal itemsets (leaf nodes) because these nodes will most likely result in the formation of approximate frequent itemsets given stringent column and row constraints. Moreover, the process is likely to collapse itemsets that contain a good deal of overlap together into a single approximate itemset, which simplifies the results. Because of the uncertainty on what provides the optimal union of closed sets when forming approximate itemsets, our algorithm allows all combinations to occur and relies upon the statistic of association to determine which of the resulting itemsets are most significant.

4.1.3 Statistic of Association

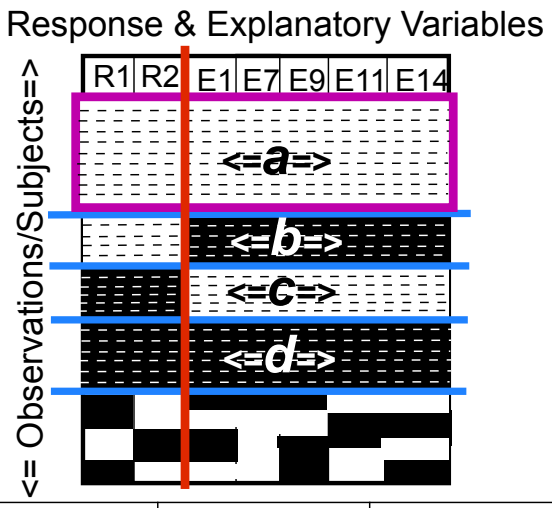
Strength of association between the two binary variable sets, the response variables and the explanatory variables, is assessed using the phi coefficient. The phi coefficient is commonly used to determine the strength of association between two binary variables (Chedzoy, 2006). Specifically the phi coefficient is an expression of the amount of consistency (both variables share same value) and inconsistency (variables differ in value) that exist between the two binary variables (Chedzoy, 2006). To calculate the phi coefficient (see Equation 4.1), the number of observations that are associated with each of the four categories that exist between the two binary sets of variables is determined as depicted in the contingency table in figure 4.7. A chi-square statistic with one degree of freedom can then be calculated based upon the phi coefficient (see Equation 4.1) as shown by Equation 4.2 using the counts in the contingency table in figure 4.7. Based upon this chi-square statistic, a p-value can be calculated to determine the significance of the strength of association between the two sets of binary variables. In cases where cell sizes of the contingency table are small, a Fisher's exact calculation is more appropriate to prevent one from failing to identify a significant association. This should not be an issue for truly consistent subsets because the minimum support threshold should be large

enough that cell ***a*** should overshadow small cell sizes in cells ***b*-*d***. When performing multiple tests of strength of association for different binary variables based upon the same set of observation it is necessary to adjust the p-value to account for the increase in type-I error that occurs due to multiple testing. The p-values used to determine significant results have been adjusted for multiple testing using the False Discovery Rate (FDR) adjustment correction, details can be found in Benjamini and Hochberg (1995) paper.

$$\Phi = \frac{(a * d) - (b * c)}{\sqrt{(a + b) * (b + d) * (a + c) * (c + d)}} \tag{4.1}$$

$$\chi^2 = (a + b + c + d) * \Phi^2 \tag{4.2}$$

Our use of the phi coefficient differs slightly from its traditional use because we are looking at association between to **sets** of binary variables as opposed to two binary variables. Therefore, these sets of binary variables represent distinct patterns where all variables in a set can be active(ones) or inactive(zeros). Because we are looking at sets of variables this means certain observations may remain undefined because they contain a mixture of activity (ones and zeros) for given a set of variables. In this sense, we subset the observations under consideration down to data that can be defined as active or inactive given a variable set definition and we test for strength of association between the two variable set combinations. Using the phi coefficient we are attempting to judge the consistency between two variable sets, here response and explanatory variables, over a subset of data in which the phi coefficient can be clearly defined. Because the four categories of the contingency table are so closely related to each other, we only have to mine for one of the four cells of the contingency table to define the others when calculating strength of association between two variable sets. In our case, we use closed frequent itemset mining to define all combinations of response and explanatory

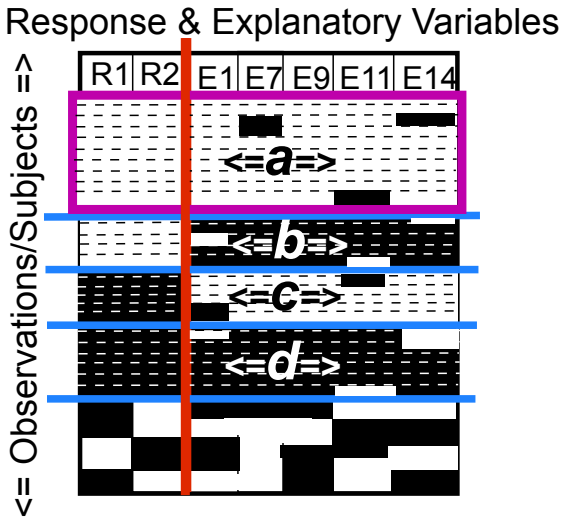


	111...(Explanatory)	000...(Explanatory)
1 -(Response(s))	1 111… [Consistent] ***a***	1 000… [Inconsistent] ***b***
0 -(Response(s))	0 111… [Inconsistent] ***c***	0 000… [Consistent] ***d***

Figure 4.7: Statistic of Association. Binary matrix shows mined closed frequent itemset *R1,R2,E1,E7,E9,E11,E14* as indicated by pink box labeled *a*. The other three categories of data denoted by *b, c*, and *d* define the remaining 3 combinations of response and explanatory variables in the contingency table below the matrix. This contingency table depicts the data partitioning used to determine strength of association between the two binary sets, response variables and explanatory variables. Observations that cannot be classified by the 4 categories are ignored.

variable sets that are all active (ones). The only criteria placed upon the combinations discovered is that they contain at least one response and explanatory variable and that they meet the minimum support threshold for active observations. The cartoon in figure 4.4 depicts the closed frequent itemsets where response and explanatory variables sets are all active (***a***, pink box) and its associated subset that contains all four categories (***a-d***, yellow box).

The data matrix on top of figure 4.7 depicts an example closed frequent itemset *R1, R2, E1, E7, E9, E11, E14* as indicated by the pink box labeled *a* and its similarly labeled cell in the contingency table below the matrix. The remaining three cells of the contingency table, *b-d*, are indicated by the blue lines and letter labels in the data matrix. Once the mined closed itemsets have been defined, in our case where both response and explanatory variables are all active, the remaining three categories can be calculated by a single pass through the



	110...(Explanatory)	010..(Explanatory)
1 -(Response(s))	1 110… [Consistent] **a**	1 010… [Inconsistent] **b**
0 -(Response(s))	0 110… [Inconsistent] **c**	0 010… [Consistent] **d**

Figure 4.8: Statistic of Association for Approximate Itemsets. Primary difference between calculating phi coefficient for approximate itemsets as compared to closed itemsets is that approximate itemsets allow some proportion of zeros in the observations that define the four categories of the contingency table for the explanatory variables (only).

data. This determines the strength of association for each of the variable sets as defined by the mined closed itemsets and adjusts this value to determine its significance with regards to the entire dataset. Therefore given a threshold of support, in our case the number of observations that must be all active, we can rank and determine which subsets of data are most statistically significant, regarding strength of association, given the entire distribution of the data. This provides a powerful way to determine which subsets of data most likely show a true association between explanatory and response variables when considering all possible enumerations. Given noisy and inconsistent data, this methodology can provide insight into determining novel associations between explanatory and response variables that could not be efficiently discovered without prior knowledge of a likely relationship between explanatory and response variables.

The phi coefficient could also be calculated for approximate sets in the results, where the

explanatory variable set could now contain a mixture of ones and zeros as long as they did not exceed the row and column constraints placed upon approximate sets as defined by the user. The count for the contingency cell that represented when the response and explanatory variables were considered active is indicated by the pink box and letter *a* in figure 4.8 is given by the number of observations (support) that an approximate frequent itemset has. The more challenging issue is to define the remaining three categories while allowing the same user defined approximation for the explanatory variables. The algorithm first uses the results from the approximate sets to define the sets of response and explanatory variables under consideration. Then in a single pass through the algorithm defines three sets of observations that meet the row and column constraints placed upon approximate sets. Next for each of the three sets, the algorithm combines the observations in similar fashion to finding an approximate set based upon closed frequent itemsets to determine the maximum number of observations that can be combine without breaking column and row constraints. Whichever combination gives the maximum count becomes the observations used for the set associated to the cell of the contingency table when calculating the phi coefficient. The chi-square statistic and FDR adjustment of the p-value are then calculated using these counts. The approximate frequent itemsets represent the union of closed frequent itemsets; thus, results including approximate frequent itemsets also include closed frequent itemsets that could not be made approximate and the FDR adjustment of the p-values account for all these results.

4.2 Results with Real World Example

The purpose of our method is to identify relationships that show association between sets of response and explanatory variables when the data under consideration is noisy and inconsistent. The data relationships are so complicated and contradictory that the only reliable information that can be deduced is in a binary sense that either a response occurred or it did not occur. Additionally determining relationships amongst the data proved to be so difficult

that using traditional methods of analysis failed to produce substantial results. A real world example of such data is the EPA's ToxCast and ToxRefDB programs, ToxRefDB contains a multitude of animal study endpoints and ToxCast contains bioassay responses to the same set of potentially toxic chemicals (Dix et al., 2007; EPA, 2010a; Judson et al., 2009; Martin et al., 2009a; Knudsen et al., 2009; Martin et al., 2009b; EPA, 2010b). As discussed previously preliminary investigations by both Reif et al. (2010) and DiMaggio et al. (2010) indicated that the data collected from the first round of ToxCast is quite sparse, with highly variable amounts of inconsistency when associating toxicity of the chemicals based upon the associations between the bioassay and the animal study endpoints. The goal was to find significant associations between the 289 ToxRefDB animal study endpoints, as the response variables, and the 659 ToxCast bioassays, as the explanatory variables, over the set of 283 potentially toxic chemicals that were common to both studies, see table 3.1.

4.2.1 Closed and Approximate Frequent Itemsets

There were 22,881 closed frequent itemsets formed with a frequency support threshold of approximately one-sixteenth (40) of the chemicals common to both studies. These itemsets include up to six ToxRefDB animal endpoints and as many as ten ToxCast HTS bioassays with chemical support threshold of at least 40 of the chemicals. The initial seed nodes of these 22,881 closed itemsets consisted of 58 unique sets of ToxRefDB animal endpoints that had an association to at least one ToxCast HTS bioassay. The approximate itemset algorithm is run on the maximal frequent itemsets (leaf nodes) of these closed sets with row and column thresholds set to 0.70 or greater proportion of ones in the 18,098 resulting approximate sets. All 58 seed nodes had at least one approximate frequent itemset and up to twelve HTS bioassays were included in these approximate itemsets. The approximate itemsets involved combining as few as two and as many as six different closed sets; thus, resulting in a total of 25,380 closed or approximate itemsets, where 18,098 were approximate and 7,282 were closed frequent itemsets that were not involved in the creation of the approximate itemsets.

The phi coefficient and its corresponding p-value were calculated for all 25,380 resulting sets, these p-values were then adjusted for multiple testing using the FDR correction provided by the R p.adjust() function in the stats package. Table 4.1 displays all 25,380 resulting itemsets grouped by the 58 seed nodes and classified by their type (closed or approximate) and their significance. Table 4.1 shows that 6.2 percent of the itemsets could be attributed to sets that include more than one animal endpoint (multivariate response) and of those 25.1 percent had adjusted p-values that were significant at α of 0.05. This is a higher rate of significance as compared to all 25,380 itemsets, only 19.2 percent of these had significant adjusted p-values.

The multivariate (2+ animal endpoints) significance has not been explored in depth by other research groups like Reif et al. (2010) and DiMaggio et al. (2010); therefore, the next sections are going to focus upon the analysis of two of the thirty-eight seed nodes that represent this multivariate response with regards to the association between sets of response and explanatory variables. The two selected seed nodes are indicated by *** in table 4.1, we chose one of the smallest and one of the largest two endpoint seed nodes, where size refers to the number of itemsets associated to a seed node. These were selected to demonstrate one way to effectively analyze and verify the significant itemsets of a seed node regardless of the number of itemsets associated to the seed node.

ToxRefDB Animal Endpoints	#Apx Sets	#Cls Sets	#Total	#Signif
(6) Chrn: Mouse: Liver Tumors, Any Liver Lesion, Preneoplastic Liver Lesion, Tumorigen, Prolifertve Liver Lesions, Neoplastic Liver Lesion	6	24	30	1
(5) Chrn: Mouse: Preneoplastic Liver Lesion, Tumorigen, Any Liver Lesion, Rat: Any Liver Lesion, Prolifertve Liver Lesions	0	1	1	0
(5) Chrn: Mouse: Liver Tumors, Preneoplastic Liver Lesion, Tumorigen, Any Liver Lesion, Neoplastic Liver Lesion	6	30	36	2
(4) Chrn: Mouse: Preneoplastic Liver Lesion, Any Liver Lesion, Prolifertve Liver Lesions , Rat: Any Liver Lesion	6	21	27	5
(4) Chrn: Mouse: Preneoplastic Liver Lesion, Tumorigen, Any Liver Lesion, Prolifertve Liver Lesions	6	36	42	3
(3) Chrn: Mouse: Any Liver Lesion Rat: Any Liver Lesion, MultiGenReprdctv: Rat: Liver	0	6	6	2
(3) Chrn: Mouse: Tumorigen, Any Liver Lesion Rat: Any Liver Lesion	0	3	3	0
(3) Chrn: Mouse: Any Liver Lesion Rat: Any Liver Lesion, Prolifertve Liver Lesions	0	1	1	1
(3) Chrn: Mouse: Any Liver Lesion Rat: Any Liver Lesion, Liver Hypertrophy	0	1	1	1
(3) Chrn: Mouse: Any Liver Lesion, Liver Hypertropy Rat: Any Liver Lesion	0	1	1	0
(3) Chrn: Mouse: Preneoplastic Liver Lesion, Tumorigen, Any Liver Lesion	5	32	37	3
(3) Chrn: Mouse: Preneoplastic Liver Lesion, Any Liver Lesion, Prolifertve Liver Lesions	143	309	452	122
(3) Chrn: Rat: Prolifertve Liver Lesions, Any Liver Lesion, Preneoplastic Liver Lesion	6	14	20	4
***(2) Chrn: Mouse: Any Liver Lesion Rat: Any Liver Lesion	44	102	146	69
(2) Chrn: Mouse: Any Liver Lesion Develpmt: Rat: Skeletal Axial	5	7	12	0
(2) Chrn: Mouse: Any Liver Lesion Develpmt: Rabbit: Pregnancy Rel Matrnl Preg Loss	0	5	5	0
(2) Chrn: Mouse: Any Liver Lesion MultiGenReprdctv: Rat: Liver	2	40	42	15
(2) Chrn: Mouse: Any Liver Lesion Rat: Tumorigen	6	11	17	2
(2) Chrn: Mouse: Preneoplastic Liver Lesion, Any Liver Lesion	95	255	350	111
(2) Chrn: Mouse: Tumorigen, Any Liver Lesion	15	73	88	8
(2) Chrn: Mouse: Any Liver Lesion Rat: Any Kidney Lesion	0	3	3	0
(2) Chrn: Mouse: Any Liver Lesion Rat: Prolifertve Liver Lesions	0	2	2	0
(2) Chrn: Mouse: Any Liver Lesion, Liver Hypertrophy	2	18	20	2
***(2) Chrn: Rat: Any Liver Lesion Develpmt: Rat: Skeletal Axial	6	8	14	3
(2) Chrn: Rat: Any Liver Lesion MultiGenReprdctv: Rat: Liver	2	23	25	10
(2) Chrn: Rat: Tumorigen, Any Liver Lesion	17	34	51	15
(2) Chrn: Mouse: Tumorigen Rat: Any Liver Lesion	0	8	8	1
(2) Chrn: Rat: Any Liver Lesion Develpmt: Rat: Genrl Fetal Weight Redctn	0	1	1	0
(2) Chrn: Rat: Any Liver Lesion, Any Kidney Lesion	2	8	10	1
(2) Chrn: Rat: Prolifertve Liver Lesions, Any Liver Lesion	5	22	27	3
(2) Chrn: Rat: Any Liver Lesion, Liver Hypertrophy	16	31	47	7
(2) Develpmt: Rat: Skeletal Axial, Rabbit: Pregnancy Rel Matrnl Preg Loss	0	4	4	0
(2) Develpmt: Rat: Skeletal Axial MultiGenReprdctv: Rat: Liver	2	7	9	3
(2) Chrn: Rat: Tumorigen Develpmt: Rat: Skeletal Axial	0	2	2	0
(2) Develpmt: Rat: Genrl Fetal Weight Redctn, Skeletal Axial	3	15	18	0
(2) Chrn: Rat: Tumorigen MultiGenReprdctv: Rat: Liver	0	1	1	0
(2) MultiGenReprdctv: Rat: Kidney, Liver	2	7	9	0
(2) Chrn: Mouse: Any Kidney Lesion, Kidney Pathology	0	1	1	0
(1) Chrn: Mouse: Any Liver Lesion	4,638	1,477	6,115	1,297
(1) Chrn: Rat: Any Liver Lesion,	12,024	3,019	15,043	2,907
(1) Develpmt: Rat: SkeletalAxial	102	341	443	2
(1) Develpmt: Rabbit: Pregnancy Rel Matrnl Preg Loss	39	115	154	5
(1) MultiGenReprdctv: Rat: Liver	648	497	1,145	273
(1) Chrn: Rat: Tumorigen	145	248	393	8
(1) Chrn: Mouse: Tumorigen	45	204	249	6
(1) Develpmt: Rat: Genrl Fetal Weight Redctn	20	52	72	0
(1) Chrn: Rat: Any Kidney Lesion	18	67	85	0
(1) MultiGenReprdctv: Rat: Kidney	4	16	20	0
(1) Develpmt: Rabbit: Pregnancy Rel Embryo Fetal Loss	0	18	18	0
(1) MultiGenReprdctv: Rat: ViabilityPND4	5	21	26	0
(1) Chrn: Rat: Prolifertve Liver Lesions	8	14	22	3
(1) Develpmt: Rat: Pregnancy Rel Embryo Fetal Loss	0	7	7	0
(1) Develpmt: Rabbit: Skeletal Axial	0	6	6	0
(1) Chrn: Rat: Any Thyroid Lesion	0	5	5	1
(1) Develpmt: Rat: Pregnancy Rel Matrnl Preg Loss	0	1	1	0
(1) Develpmt: Rat: Skeletal Appendicular	0	2	2	0
(1) Develpmt: Rabbit: Genrl Fetal Weight Redctn	0	2	2	0
(1) Chrn: Rat: Any Testes Lesion	0	3	3	0
Total:	***18,098***	***7,282***	***25,380***	***4,896***
Total 2+ Endpoints:	***402***	***1,167***	***1,569***	***394***

Table 4.1: Closed & Approximate Itemsets by seed node. Column(1) indicates seed node endpoints, col(2) number of approximate itemsets, col(3) number of closed itemsets, col(4) total number of itemsets, and col(5) number of total itemsets with significant adjusted p-value for α at 0.05 level.

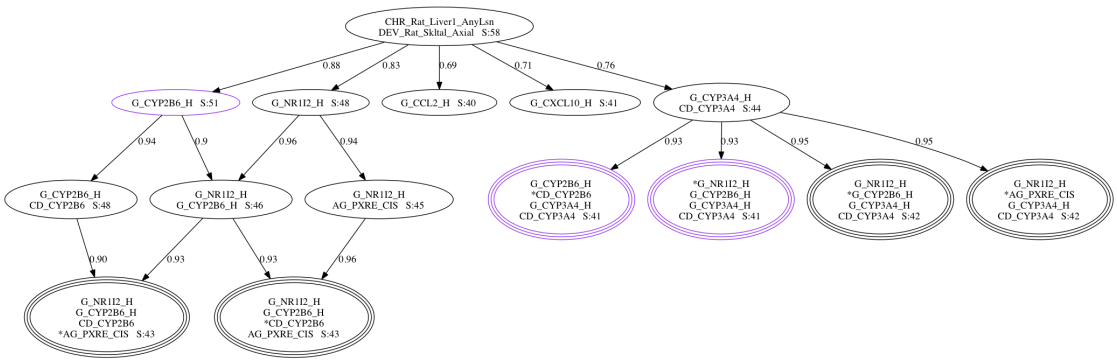


Figure 4.9: Rat Skeletal Development and Liver Lesions Tree. The tree root is two endpoints, each node (oval) off this root represents a results set. Each set is labeled with HTS bioassays that make up its composition and its chemical support as S:xx, where xx defines number of chemicals that support the set. Connections between sets represent percent overlap of chemicals shared between connecting sets. Six approximate sets are those that are sets encircled three times with an * to indicate the approximate HTS bioassays. The 3 nodes encircled in purple are significant for α of 0.05.

4.2.2 2 Endpoints: Rat Skeletal Development and Liver Lesions

As described in the methods section, itemsets were created such that animal endpoints and HTS bioassays were all active for the same subset of chemicals. The number of active chemicals for a grouping of bioassays and endpoints that compose a set is referred to as the set's chemical support. The resulting itemsets where then grouped together based upon commonality of animal endpoint(s) that represented the response variables that made up the initial seed nodes. One of the smallest two endpoint seed nodes is one that includes the endpoints: developmental rat skeletal axial, defined as the variations or abnormalities of the vertebral column, ribs or sternum in a rat fetus, and chronic rat liver lesions, defined as any chronic rat liver lesions. This endpoint seed node originally included 16 closed itemsets that contained from 1 to 4 different HTS bioassays and a chemical support that ranged from 40 to 51 chemicals. These 16 closed sets were then used to create approximate sets which resulted in 14 itemsets, where 6 of those are approximate. A representation of this two endpoint seed node with the six approximate itemsets and 8 closed itemsets is depicted in Figure 4.9.

The following five human genes are ones associated with the activated bioassays of the two endpoint tree: CCL2 chemokine (C-C motif) ligand 2, CXCL10 chemokine (C-X-C mo-

tif) ligand 10, NR1I2 nuclear receptor subfamily 1 group I member 2, CYP3A4 cytochrome P450 family 3 subfamily A polypeptide 4, and CYP2B6 cytochrome P450 family 2 subfamily B polypeptide 6. Although the mechanism behind the association between liver disease and metabolic bone disease is not fully understood, Ferencz et al. (2005) demonstrated that experimentally induced liver cirrhosis in rats arrests skeletal growth and influences other aspects involved in bone metabolism (Ferencz et al., 2005). This two endpoint seed seems to indicate that these five genes might be involved in the mechanisms that are associated with liver disease, abnormal skeletal development and the association between the two. The literature indicates that expression of these five genes can be linked to diseases of the liver and other biological responses associated with the two rat endpoints. For example, gene expression profiling of alcoholic liver disease identified the genes CCL2 and CXCL10 are involved in the immune response and gene CYP3A4 is involved with alcohol and xenobiotic metabolism(Seth et al., 2003). Furthermore, expression profiling indicates that CXCL10 gene is highly activated due to its association with inflammation in hepatic tissue of those who suffer from non-alcoholic steatohepatitis, NASH (Baker et al., 2010). NASH is a form of non-alcoholic fatty liver disease that is often associated with obesity and insulin resistance (Baker et al., 2010). Additionally, Maglich et al. (2002) demonstrate how nuclear pregnane X receptor (PXR - NR1I2) can regulate the expression of both CYP3A isozymes, like CYP3A4, and CYP2B genes, like CYP2B6, in the detoxification response to a large range of chemicals as observed in human hepatocytes (Maglich et al., 2002).

Figure 4.9 depicts the three sets out of the fourteen sets that are significant for α at 0.05 level. These three sets have bioassays that are associated with the genes NR1I2 nuclear receptor subfamily 1 group I member 2, CYP3A4 cytochrome P450 family 3 subfamily A polypeptide 4, and CYP2B6 cytochrome P450 family 2 subfamily B polypeptide 6. Maglich et al. (2002) demonstrate these three genes are involved in detoxification response in hepatocytes; specifically, how nuclear pregnane X receptor (PXR - NR1I2) can regulate the expression of both CYP3A isozymes and CYP2B genes in the detoxification response in human hepatocytes

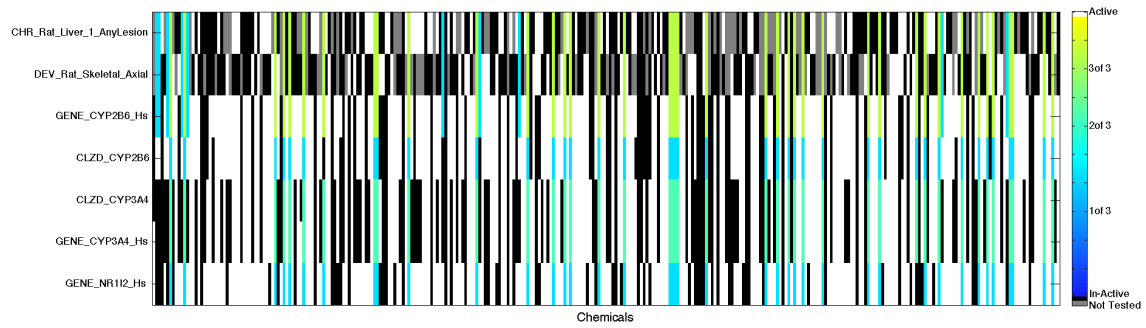


Figure 4.10: Rat Skeletal Development and Liver Lesions heat map shows intricacy of the relationship that exists between the rat skeletal development and liver lesion endpoints and the five bioassays that constitute the three significant sets belonging to those endpoints regarding chemical activity. The 320 columns represent the 309 chemicals of ToxCast and the 7 rows represent the endpoints and bioassays. The blue to light green color on the graphic indicates active chemicals that are members of one or more of the significant sets. White indicates chemical activity but failure to be grouped as a member of one the significant sets and black indicates chemical inactivity. Gray indicates chemicals that were not tested on the endpoints.

(Maglich et al., 2002). The complexity of the chemical activity as associated with the endpoints and bioassays of these three significant sets across the 309 ToxCast chemicals can be seen in Figure 4.10. As depicted in Figure 4.10 when considering the entirety of the ToxCast chemicals it is difficult to see the relationship between these two endpoints and the five bioassays of the significant set with regards to chemical activity. Figure 4.11 focuses on only those 50 chemicals that are active and members of at least one of the three significant sets, the figure indicates which chemicals activate both endpoints and bioassays of the significant sets. Notice that the coloring schemes in both figures 4.10 and 4.11 have the same meaning as in figure 4.5. As the color bar on right show, shade of blue and green indicate overlap between the sets, gray indicates chemical that weren't tested for those endpoints, white indicates chemical activity that is not part of a set, and black indicates chemical in-activity.

Although not significant at the 0.05 level, ten of the remaining eleven sets had an adjusted p-value of less than 0.2350 and were associated with the same three genes, NR1l2, CYP3A4 and CYP2B6, in addition to CXCL10 chemokine ligand 10. CXCL10 is is known to be associated with immune response in alcoholic liver disease (Seth et al., 2003) and inflammation of

Chemical	CAS_RN	Is Apx	Sig Sets	Mjr Sets
2,4-DB	94-82-6			
2,4-Dichlorophenoxyacetic acid (2,4-D)	94-75-7			
3-Iodo-2-propynylbutylcarbamate	55406-53-6		x	x
Acetamiprid	135410-20-7			
Acetochlor	34256-82-1		x	x
Acibenzolar-S-Methyl	135158-54-2			
Butylate	2008-41-5		x	
Carbaryl	63-25-2		x	x
Carfentrazone-ethyl	128639-02-1		x	
Chlorpropham	101-21-3		x	
Clodinafop-propargyl	105512-06-9	x	x	
Cyproconazole	94361-06-5		x	x
Cyprodinil	121552-61-2		x	x
Dichloran	99-30-9		x	x
Difenoconazole	119446-68-3			
Emamectin benzoate	155569-91-8			
Fenamidone	161326-34-7			
Fenbuconazole	114369-43-6		x	x
Fentin	76-87-9		x	
Fluazinam	79622-59-6		x	x
Flufenacet	142459-58-3		x	x
Flusilazole	85509-19-9		x	x
Hexaconazole	79983-71-4		x	x
Isoxaflutole	141112-29-0		x	x
Lactofen	77501-63-4		x	x
Lindane	58-89-9		x	x
Linuron	330-55-2		x	
Metalaxyl	57837-19-1		x	x
Myclobutanil	88671-89-0		x	x
Nitrapyrin	1929-82-4		x	x
Oxadiazon	19666-30-9		x	x
Oxasulfuron	144651-06-9		x	
Paclobutrazol	76738-62-0		x	x
Permethrin	52645-53-1		x	
Propanil	709-98-8		x	x
Propiconazole	60207-90-1		x	x
Pyrithiobac-sodium	123343-16-8	x	x	
Quintozene	82-68-8		x	x
S-Bioallethrin	28434-00-6		x	x
Sethoxydim	74051-80-2		x	x
Simazine	122-34-9			
Tebufenpyrad	119168-77-3		x	x
Tetraconazole	112281-77-3		x	x
Thiacloprid	111988-49-9		x	
Thiazopyr	117718-60-2		x	x
Thiram	137-26-8			
Tralkoxydim	87820-88-0		x	
Triadimefon	43121-43-3		x	
Triflumizole	68694-11-1		x	x
Triticonazole	131983-72-7		x	

Table 4.2: Chemicals Common to Significant Sets for rat developmental skeletal axial and chronic liver lesions endpoints seed node. 50 chemicals that are common to at least one of the three significant sets for endpoints. First two columns indicate chemical name and CAS registry numbers with the third column indicating 'approximate' chemicals (inactivity for a few bioassays in the significant sets). Fourth column indicates the 41 chemicals common to all 3 significant sets and fifth column indicates 28 chemicals common to 13 sets with adjusted p-values less than 0.2350.

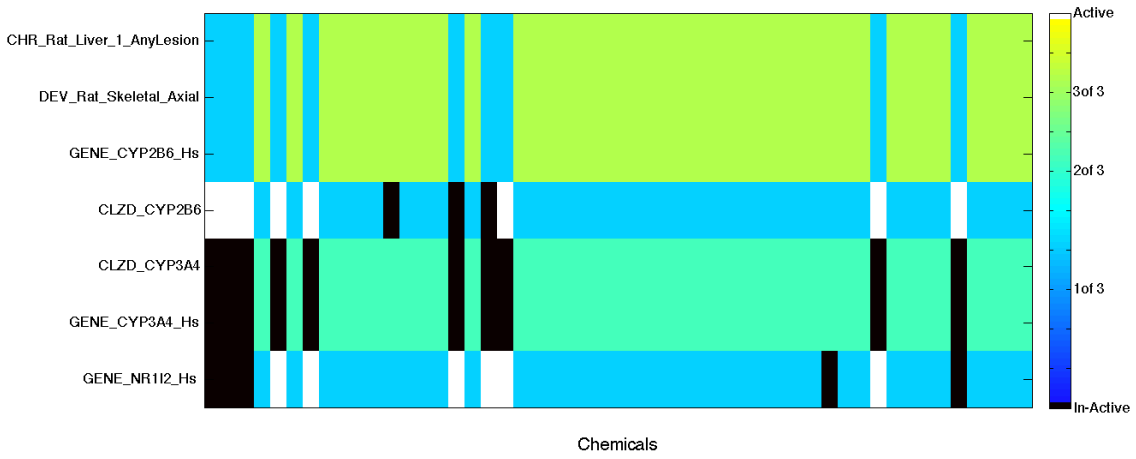


Figure 4.11: Rat Skeletal Development and Liver Lesions heap map focuses on 50 chemicals that are members of at least one of the three significant sets for the rat skeletal development and liver lesion endpoint seed depicted in figure 4.9. The 7 rows represent the endpoints and bioassay and the columns represent the 50 chemicals associated to the significant sets. The blue to light green color indicates active chemicals that are members of at least one of the three significant sets. White indicates chemical activity but failure to be grouped as a member of one of the significant sets and black indicates chemical inactivity.

hepatic tissue with non-alcoholic steatohepatitis (Baker et al., 2010). Table 4.2 highlights the chemicals that are active for both rat skeletal development and liver lesions endpoints along with the bioassays of the sets that are associated to the four genes mentioned above. Table 4.2 contains the 50 chemicals that occur in at least one of the three significant sets, the fourth column indicates the 41 chemicals that common to all three significant sets. The fifth column indicates the 28 chemicals that were common to the three significant sets along with the ten sets with adjusted p-values of less than 0.2350.

From the literature we demonstrate how the five gene targets of the activated bioassays of the significant sets that compose this two endpoint seed node can be involved in diseases of the liver, abnormal skeletal development, and the association between the two. For this two endpoint seed node the 50 chemicals listed in Table 4.2 may precipitate diseases of the liver, abnormal skeletal development, and the association amongst those two endpoints. The fact that Ferencz et al. (2005) experimentally demonstrated that induced liver cirrhosis in rats leads to arrested skeletal development adds to the credence of this unexpected data mining

discovery found within the overlap of ToxCast and ToxRefDB data.

4.2.3 2 Endpoints: Rat and Mouse Liver Lesions

The larger two endpoint seed node included chronic mouse and rat endpoints for any liver lesion. This seed node contained 146 sets, where 44 were approximate sets. These 146 sets were composed of 1 to 7 different HTS bioassays and chemical support that ranged from 40 to 72 chemicals. The adjusted p-value classified 69 of the 146 sets significant for α at 0.05 level. When considering the analysis of these 69 significant sets, we further filtered them to consider only those that had a consistency of at least seventy percent or higher. Consistency was calculated using definition provided for in Figure 4.7, where its value is calculated as sum of cells ***a*** and ***d*** over the sum total of all cells ***a***-***d***. The analysis describe below considers only 33 of the 69 significant sets; where, the sets had both statistical significance and high consistency ($\geq$70%).

The 33 significant sets of the 146 that compose this seed node encompass the activation of 15 different HTS bioassays that are associated to 8 different genes. The following eight genes are ones associated with the activated bioassays in these 33 sets: CCL2 chemokine (C-C motif) ligand 2, NFE2L2 nuclear factor (erythroid-derived 2)-like 2, PLAUR plasminogen activator urokinase receptor, HLA-DRA major histocompatibility complex class II DR alpha, PPARg peroxisome proliferator-activated receptor gamma, NR1I2 nuclear receptor subfamily 1 group I member 2, CYP3A4 cytochrome P450 family 3 subfamily A polypeptide 4, and CYP2B6 cytochrome P450 family 2 subfamily B polypeptide 6. Three of the eight genes are associated the cytochrome P450 drug-metabolizing enzymes receptors: pregnane X receptor (PXR) and constitutive androstane receptor (CAR). Specifically, the thirty-three significant sets of this seed node included two assays for the gene NR1I2, which is directly associated to PXR, and two assays for the gene CYP3A4, which is PXR-inducible. Furthermore, these sets comprised of two assays are associated with gene CYP2B6, that can be induced by CAR. Thus, these three genes can be associated to the activity of 6 out of the 15 HTS bioassays of

the significant sets of this seed node that are directly linked to a specific gene response.

There exists some uncertainty as to whether gene CYP2B6 was directly activated by CAR or if cross-talk amongst these cytochrome P450 receptors or another mechanism accounts for its activation in the absence of response from bioassays associated with CAR, NR1I3. Kohle and Bock (2009) demonstrate that there exists substantial cross-talk between the cytochrome P450 receptors PXR, CAR, and AHR (Kohle and Bock, 2009). Plant and Aouabdi (2009) suggest that besides CYP3A genes, PXR can activate the expression of other genes like CYP2B6 (Plant and Aouabdi, 2009). Younossi et al. (2009) support this claim by suggesting that new substrates of CYP2B6 might share specificity with CYP3A4; thus, regulation of CYP3A4's transcriptional activation might be similar with regulation of CYP2B6 (Younossi et al., 2009). Omiecinski et al. (2011) demonstrate that PRX and CAR cytochrome P450 receptors share overlap amongst chemical ligands and within the genes they target (Omiecinski et al., 2011). Specifically, receptors PRX and CAR are both able to transcriptionally activate CYP2B6 and CYP3A4 which makes these receptors seem to perform as a dynamic, parallel set of gene regulators with regards to xenobiotic metabolism (Omiecinski et al., 2011). Moreover, Omiecinski et al. (2011) support that there exists significant cross-talk amongst PXR and CAR receptors with regards to regulatory pathways involved with xenobiotic detoxication, adverse drug reactions, bile acid toxicity and pathophysiological conditions such as lipid metabolism and cholestatic liver disease(Omiecinski et al., 2011). Finally as mentioned earlier, Maglich et al. (2002) demonstrate how nuclear PXR can regulate the expression of both CYP3A isozymes, like CYP3A4, and CYP2B genes, like CYP2B6, in the detoxification response as observed in human hepatocytes (Maglich et al., 2002). This seems to suggest that both directly and indirectly the drug-metabolizing enzyme receptor PXR is activated by chemicals of the thirty-three sets that are associated with the genes NR1I2, CYP3A4 and CYP2B6.

The complexity of the chemical activity as associated with these rat and mouse liver lesion endpoints and the bioassays of these thirty-three significant sets across the 320 ToxCast chemicals can be seen in Figure 4.12. Figure 4.12 show the entirety of chemical activity of the

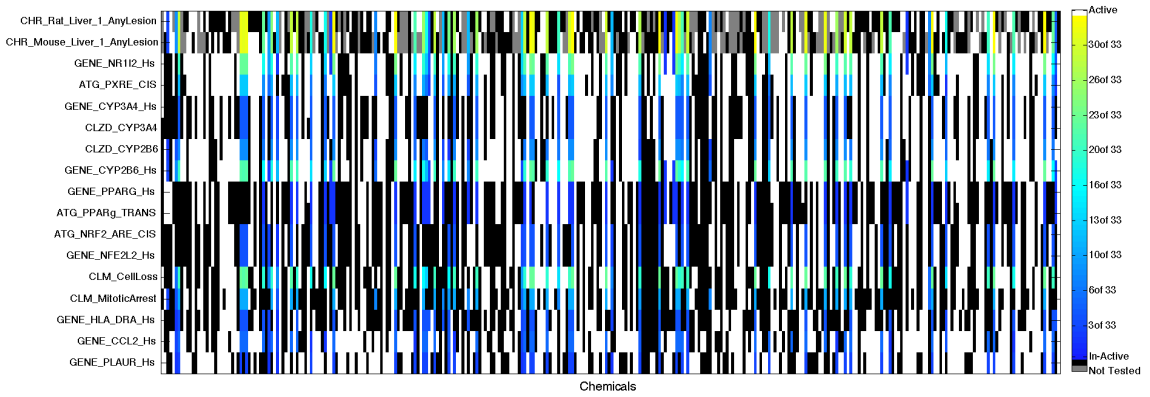


Figure 4.12: Rat and Mouse Liver Lesions heat map shows intricacy of the relationship that exists between the rat and mouse liver lesion endpoints and the fifteen bioassays that constitute the thirty-three significant sets belonging to seed node with regards to chemical activity. The 320 columns represent the 309 chemicals of ToxCast and the 17 rows represent the endpoints and bioassays. The blue to yellow color on the graphic indicates active chemicals that are members of at least one or more of the significant sets. White indicates chemical activity but failure to be grouped as a member of one of the significant sets and black indicates chemical inactivity. Gray indicates chemicals that were not tested on the endpoints.

ToxCast chemicals, from which it is difficult to see the relationship between these two endpoints and the fifteen bioassays of the significant sets with regards to chemical activity. Figure 4.13 focuses on those 62 chemicals that are active and members of at least one of the thirty-three significant sets, and indicates which chemicals activate both endpoints and bioassays of the significant sets. The six bioassays associated with the three genes (NR1I2, CYP3A4 CYP2B6) that can be attributed to PXR activation are the common to the many of the thirty-three sets as indicated by the light blue and green coloring of the rows associated to the six bioassays in Figure 4.13. This implies that the chemicals common to many of the significant sets can be attributed to PXR activation as hepatic response xenobiotic detoxication, bile acid toxicity, lipid metabolism and cholestatic liver disase.

Besides three genes related to PXR activation; there are two bioassays related to gene target for PPARγ. Omiecinski et al. (2011) indicate that thiazolidineiones, a class of xenobiotics, are potent to PPARγ agonists (Omiecinski et al., 2011). Moran-Salvador et al. (2011) demonstrate that PPARγ expression in hepatocytes can act as steatogenic inducer gene and that

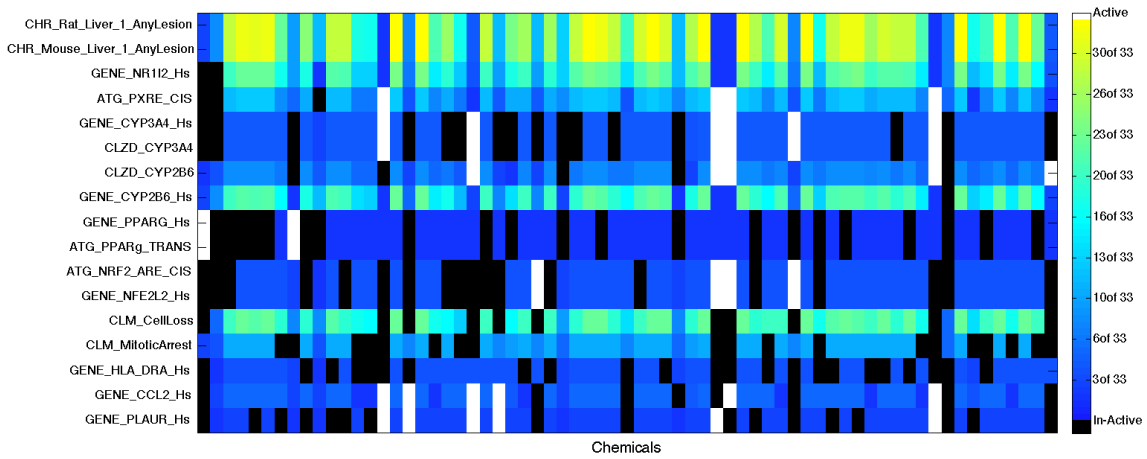


Figure 4.13: Rat and Mouse Liver Lesions heap map focuses on 62 unique chemicals that are members of at least one of the thirty-three significant sets of the rat and mouse liver lesion endpoint seed node. The 17 rows represent endpoints and bioassays and the columns represent the 62 chemicals associated to the significant sets (note chemical set includes 5 duplicate chemicals). The blue to yellow color on the graphic indicates active chemicals that are members of at least one of the significant sets. White indicates chemical activity but failure to be grouped as a member of one of the significant sets and black indicates chemical inactivity.

administration of thiazolidinediones in situations where PPARγ is already highly expressed in the liver can lead to a steatogenic response (Moran-Salvador et al., 2011). Rogue et al. (2010) indicated that thiazolidinediones have been shown to regulate cytochrome P450 activities such as the induction of CYP3A4 and CYP2B6 (Rogue et al., 2010). This implies that chemicals associated to the sets that include bioassays that target PPARγ and CYP3A4 and CYP2B6 are likely that share common chemical properties to those of the thiazolidineiones. Other genes that are included in the composition of some of the thirty-three significant sets have been associated with PXR activation are genes: CCL2 chemokine (C-C motif) ligand 2 and PLAUR plasminogen activator urokinase receptor. As stated earlier the chemokines are associated with immune response and inflammation and can be associated with alcoholic liver disease(Seth et al., 2003). Younossi et al. (2009) has associated these two genes in particular to those that are present with liver steatosis and/or fibrosis as associated with chronic hepatitis C (Younossi et al., 2009).

Chemical	CAS_RN	Is Apx	All Sets
*Bensulide	741-58-2	x	
Clofentezine	74115-24-5		x
d-cis,trans-Allethrin	584-79-2		x
Fenarimol	60168-88-9		x
Fenbuconazole	114369-43-6		
Fluazinam	79622-59-6		
Fludioxonil	131341-86-1		x
Fluthiacet-methyl	117337-19-6		
Indoxacarb	173584-44-6		x
Lactofen	77501-63-4		x
Malathion	121-75-5	x	
MGK	113-48-4		x
Paclobutrazol	76738-62-0		
Permethrin	52645-53-1	x	
Prallethrin	23031-36-9		
Resmethrin	10453-86-8		x
Thiazopyr	117718-60-2		x
Triflumizole	68694-11-1		x

Table 4.3: Chemicals Common to 2 Endpoint seed node of Rat and Mouse Any Liver Lesions. Depicts 18 chemicals that are common to at least 90 percent of the 33 significant sets of the seed node for the chronic rat and mouse any liver lesions endpoints. First two columns indicate chemical name and CAS registry numbers, the third column indicates 'approximate' chemicals(inactivity for a few bioassays in the significant sets), and final column indicates 10 chemicals common to all the significant sets. '*' indicates all 3 duplicates are represented in results for Bensulide.

Table 4.3 highlights the chemicals that are active for both rat and mouse liver lesions endpoints along with the bioassays of the sets that are associated to the eight genes mentioned above. The 18 chemicals in Table 4.3 are those that are common to at least 90 percent ($\geq$ 30) of the sets that constitution the thirty-three significant sets mentioned above. The fourth column of the table depicts the 10 chemicals that common to all thirty-three significant sets. From the literature we demonstrate how the three gene targets are involved in PXR activation, how CYP gene targets and PPARγ may both be activated by thiazolidineiones-like chemicals in association to steatogenic liver response, and how two genes associated with inflammation and immune response are also associated to PXR activation in connection to ailments of the liver (Maglich et al., 2002; Omiecinski et al., 2011; Moran-Salvador et al., 2011; Rogue et al., 2010; Younossi et al., 2009; Seth et al., 2003). This endpoint seed node implies that the 18 chemicals listed in Table 4.3 may result in diseases of the liver as associated with the rat and mouse liver lesions endpoints.

4.3 Comparison to Biclustering

An indirect comparison can be made between our results (Table 4.1) and the results provided by DiMaggio et al. (2010) with their biclustering and logistic regression framework in analyzing the ToxCast data. Biclustering is used as means of data reduction in a much larger modeling scheme, but the methods did produce results that gave the most optimal association between 18 of the endpoints and a set of bioassays from the ToxCast data. Besides not providing a measure of goodness of fit to assess the quality of their optimal models, they were only able to address 18 of the animal endpoints and did not provide any multivariate (more than one endpoint) models. Our results grouped by the 58 seed nodes in table 4.1 encompass 29 unique animal endpoints, of which 13 were the same as those addressed by DiMaggio et al. (2010) logistic regression models. In addition our methods provide multivariate associations as shown by the 38 seed nodes in table 4.1 that are associated with two to six animal endpoints.

Score	Prop. Ones	# Endpoints	#Rows	#Columns
12.49	0.91	0	8	205
8.06	0.91	10	133	4
7.12	1.00	7	121	2
6.85	0.87	0	31	101
6.75	0.99	0	4	116
6.31	1.00	0	2	75
5.91	1.00	6	80	2
4.99	1.00	0	3	78
4.96	1.00	9	178	1
4.86	1.00	23	125	2
4.55	1.00	51	139	1
4.52	1.00	0	2	22
4.44	1.00	51	130	1
4.33	1.00	0	4	18
4.32	1.00	16	89	2

Table 4.4: 15 Top Scoring Biclusters found with BicBin Algorithm.

Furthermore, of the 25,380 subsets depicted in table 4.1, 4,896 show statistically significant association between the endpoints and bioassays of Toxcast, where 1,569 of the subsets are multivariate of which 394 have statistically significant association. This demonstrates that our methods not only revealed the multivariate associations from amongst the data, but also provided a measure of statistical relevance for those results.

A direct comparison can be made between our results (Table 4.1) and the results provided by van Uitert et al. (2008) in analyzing the ToxCast data. Our algorithm most resembles biclustering of binary data as provided by van Uitert et al. (2008) methodology because it allows for some level of approximation (zeros) to be considered within the resulting biclusters. Using the ToxCast dataset we compared the results our algorithm produced with those produced by van Uitert et al. (2008) method of biclustering binary data (BicBin). One issue with using BicBin is that it only provides the highest scoring bicluster. Thus, to get more than one bicluster, the previously discovered bicluster data must be set to zero prior to searching for the next top scoring bicluster within the data. This is repeated until the BicBin algorithm no longer provides a bicluster given the input parameters or the dataset contains only zeros. This produces fewer results than our algorithm because BicBin provides non-overlapping biclusters whose dimensions (number rows and columns) are not restricted by the algorithm. Whereas,

	Itemset Mining		BicBin Biclustering	
	All	Sig@0.05	All	Sig@0.05
All Subets	25,380	4,896	528 / 308	203
Aprox Subsets	18,098	3,542	10	2

Table 4.5: Comparison of Closed/Approximate Itemset Mining to BicBin Biclustering for All & Approximate Subsets. Significant subsets are determined at α 0.05 level for p-values adjusted for multiple testing using FDR.

our algorithm produces the equivalent to overlapping biclusters (subsets) whose dimensions are restricted by the minimum support threshold requirement.

The BicBin algorithm gives little control over how large the proportion of zeros, the approximation, of the produced results. To make the BicBin results comparable to our results we used a range of BicBin parameters to discover the top scoring biclusters for a given parameter set. Once we discover biclusters for each parameter set, we greedily selected the highest scoring bicluster that did not exceed the row and column thresholds for approximation (proportion of zeros). This bicluster's data is zeroed out and the selection process was repeated until no more biclusters could be identified. The fifteen top scoring BicBin biclusters are depicted in table 4.4. Notice that only eight of the fifteen have any response variables (endpoints) included in their bicluster. This is problematic for showing association between response and explanatory variables. Furthermore, there is not much balance between size of rows and columns. This is also problematic because ideally one would want to find a number of response and explanatory variables (rows) that show some association that holds true over a decent number of observations (columns). Only five out of the fifteen BicBin biclusters have at least four rows and at least for columns and only one of those five has any response variable (endpoint).

The BicBin algorithm identifies 528 biclusters of which only 308 contained at least one endpoint. The overall results are displayed in table 4.5, one can see that only 10 of the 528 biclusters were approximate. For the 308 results that contained at least one endpoint, strength of association was assessed between the endpoints and bioassays of the bicluster using the phi

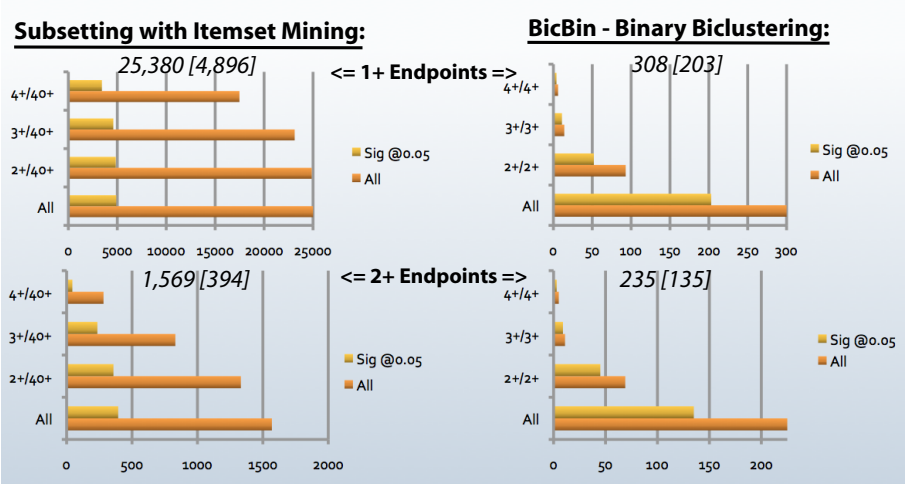


Figure 4.14: Comparison of Closed/Approximate Itemset Mining to BicBin Biclustering for all subsets classified by number of endpoints included within the subset. Top two charts for subsets with 1+ endpoints and bottom two charts for subsets with 2+ endpoints. Reports all subsets (orange bars) and the significant subsets (yellow bars) determined at α 0.05 level for p-values adjusted for multiple testing using FDR.

coefficient. The significance of the biclusters is based upon the chi-square statistic associated to the phi coefficient and the p-values are adjusted for multiple testing using FDR as described in section 4.1.3. Table 4.5 shows how our methods provided more statistically significant results and allow for a higher degree of approximation to be incorporated into the results.

The four charts in figure 4.14 break the comparison down by number of endpoints and by the dimensions of the subsets/biclusters. The number above each chart shows the total number of results and the number of statistically significant results in brackets. The orange bars represent the total results and the yellow bars show only the statistically significant results. The bottom two bars labeled *All* display all the results as is represented by the numbers above each chart. The next three sets of bars break the results down by their dimensions, as in they only display the results that meet or exceed the dimensions (column/row) labeling the left side of the chart. The column dimension only include the number of bioassays because the top two charts display results associated with at least one endpoint and the bottom two charts only depict the results of the multivariate associations (2 or more endpoints). The minimum support threshold requirement provides that all of our results are associated with at least 40 rows

(chemicals). Whereas, the BicBin bicluster algorithm provides no restrictions on the bicluster dimensions; thus, the dimension classifications were selected to match for the minimum number of rows and columns. The results in figure 4.14 show that our algorithm produces more significant results and that our results are more balanced in dimensions (top rows) for both univariate (1 endpoint) and multivariate (2+ endpoints) associations.

The results of the comparison to BicBin as shown in tables 4.4 and 4.5 and figure 4.14 demonstrate how our methodology provides more appropriate results than those provided by biclustering algorithms, like BicBin. Specifically, our methods only provide appropriate results with regards to always including both response and explanatory variables (endpoints and bioassays) and our methodology is better able to incorporate approximation into the results. The restrictions on dimensions provide for larger and more balanced (equality in number of rows and columns) subsets of data, which provide for a larger quantity of significant results that are more easily verified. Specifically, the more bioassays provided in the column dimension the more information is provided about biological pathways perturbed by the chemicals in the row dimension. In addition with our methodology's use of the seed nodes, one can easily target their analysis to focus on only specific associations.

4.4 Timing

The primary bottleneck of this method is mining for closed frequent itemsets and using those resulting itemsets to determine the subset of data that defines the association between the response and explanatory variables. The time and space required for this computation is dependent the number closed frequent itemsets discovered. The correlation between runtime and number of closed frequent itemsets discovered is 0.92 as depicted by figure 4.15. The number of closed frequent itemsets found is dependent upon the density of ones within the binary dataset, the association of explanatory and response variables as related to the density of ones, and the minimum support threshold selected for the closed itemset mining. Table

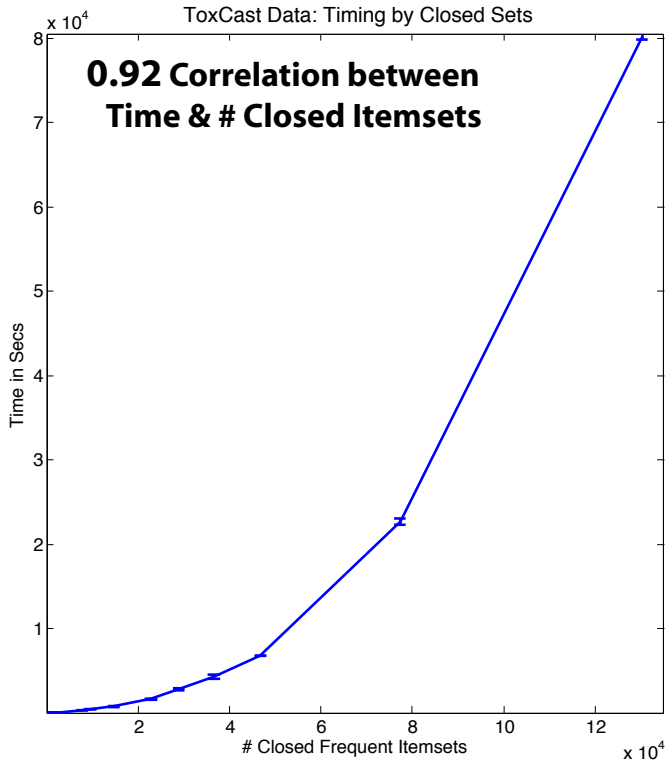


Figure 4.15: Timing plotted for Closed Itemset creation.

4.6 contains the average runtime given different minimum support thresholds on the ToxCast dataset. The ToxCast dataset was used because it provided an association between the response and explanatory variables that had the complexity that one might see with a real world example. All jobs were run on UNC's research computing cluster KillDevil each running on a single Intel EM64T 2.0-2.93 GHz CPU with access to at least 8 GB of memory. In table 4.6 *indicates the first row where average run times exceed one minute and **indicates the first row where average run times exceed one hour. Figure 4.15 and table 4.6 demonstrate how the computational complexity increase as density of ones and size of the data increase as simulated by the lowering of the support threshold to increase the number of discovered closed frequent itemsets. The primary approach to dealing with such increasing complexity is to limit the scope of analysis through the use of the seed nodes to target the analysis to focus only on specific associations. Other means of controlling the computational complexity involve using larger minimum support thresholds, reducing the number of input variables via

Min. Support Threshold proportion (n)	# Closed Itemsets	Average in Secs	StdDev in Secs
0.300 (75)	134	6.0	0.1
0.280 (70)	210	8.0	0.4
0.260 (65)	387	12.6	0.4
0.252 (63)	482	15.3	0.8
0.240 (60)	698	21.1	1.5
0.228 (57)	1,070	30.8	0.2
0.220 (55)	1,401	41.2	1.8
*0.208 (52)	2,220	64.5	0.4
0.200 (50)	3,106	92.9	1.6
0.180 (45)	7,816	297.0	7.5
0.176 (44)	9,564	417.3	23.9
0.168 (42)	14,724	761.4	38.4
0.160 (40)	22,881	1,639.5	77.4
0.156 (39)	28,879	2,837.5	131.1
**0.152 (38)	36,616	4,280.4	248.5
0.148 (37)	46,716	6,757.4	44.4
0.140 (35)	77,332	22,641.1	373.3
0.132 (33)	130,197	80,178.4	356.3

Table 4.6: Timing of Closed Frequent Itemset Mining using different support thresholds on ToxCast Data. *Indicates first row where average runtime exceeds one minute and **indicates where average runtime exceeds one hour.

data reduction techniques and to mine for the negative cases when dealing with extremely dense data. It should be noted that mining for negative cases changes the type of results one finds, but given extremely dense datasets the negative space will provide more informative results.

4.5 Conclusions

Identification of relationships that exist amongst response and explanatory variables when the data under consideration has a nontrivial amount of inconsistency and noise within it can prove to be difficult using classical methods of analysis. Instead of considering the entire data record in dealing with association amongst noisy data, we developed methodology that focuses on subsets of the data that would be considered the most consistent (noise free). We provided a means to successfully identify subsets of data that show association between re-

sponse and explanatory variables and to quantify the strength of this relationship using the phi coefficient. We account for the increase in type I error that is seen with multiple testing using the FDR correction on the p-values associated with the phi coefficient. Additionally, we demonstrated how some level of user defined approximation (fuzziness) can be introduced into the results to help account for the inconsistency within the data. We illustrated how the most statistically significant subsets of data can be analyzed using the ToxCast data. Since biclustering methodology also identified subsets of data; we directly compare our methodology to van Uitert's method of biclustering binary data, BicBin, using the ToxCast data. We established how our methodology is more adept than BicBin at focusing exclusively on subsets of data that have an association between response and explanatory variables. Additionally, we provide techniques that can be used to enable our methodology to provide results for datasets of larger size and higher density than the ToxCast data.

Chapter 5

Mining for Association with Improved Statistic

5.1 Motivation

As established in Chapter 4, one can use closed and approximate frequent itemset mining to identify subsets of data that have association between response and explanatory variables. Namely the p-value associated with the phi coefficient can be used to determine which discovered subsets of data are statistically significant (important) while accounting for the increase in type I error that is associated with multiple testing by using the False Discovery Rate (FDR) correction. The only issue that arises occurs when the phi coefficient fails to adequately quantify the association between response and explanatory variables. For example, in figure 5.1 there are six subsets of data that all have the same consistency, but vastly different associated p-values, where $consistency = (a + d)/(a + b + c + d)$. Comparing the top row (black text) to the two below it (blue text), notice that the associated p-value increases in significance (becomes smaller in value) as there is less balance in *inconsistent* cells b and c, where balance means equality in value. Comparing the top row (black text) to the fourth and fifth rows (purple text), notice how the associated p-value increases in significance as there is more balance in *consistent* cells a and d. If one deems the metric consistency more important regarding the association between response and explanatory variables than the phi coefficient, figure 5.1 demonstrates how the methods described in Chapter 4 will not adequately discover

Consistent		InConsistent					
11 - **a**	00 - **d**	10 - **b**	01 - **c**	Phi-coef	Phi ChiSq	P-value	Consistency
40	**20**	**12**	**13**	**0.3775**	**12.1129**	**0.0005**	**0.7059**
40	20	**8**	**17**	0.3944	13.2215	**0.0003**	0.7059
40	20	**5**	**20**	0.4260	15.4259	**0.0001**	0.7059
50	**10**	12	13	0.2447	5.0892	**0.0241**	0.7059
55	**5**	12	13	0.1008	0.8634	**0.3528**	0.7059
55	5	2	23	0.2453	5.1152	0.0237	0.7059

	111...(Explanatory)	000...(Explanatory)
1 -(Response(s))	1 111… [Consistent] **a**	1 000… [Inconsistent] **b**
0 -(Response(s))	0 111… [Inconsistent] **c**	0 000… [Consistent] **d**

Figure 5.1: Depicts issue with using statistics based upon p-value. Shows how same consistency can give vastly different phi coefficient p-values.

all important subsets of data with regards to consistency using the phi coefficient.

In the preceding chapter we propose a bootstrap methodology that has been adapted to free ourselves from having to use metrics associated with p-values to determine statistically significant subsets of data discovered through the process of closed frequent itemset mining. The primary benefit of this bootstrap methodology is that it can be used with *any* statistic or property of the dataset without dependence upon a p-value. Moreover the method provides ways to incorporate multiple metrics into the methodology, such that the final significance can be associated with multiple properties of the data. The advantage of this is that if one can provide a statistic that incorporates the integration of three or more datasets, one can effectively extend the method to consider associations between three or more datasets as depicted in figure 5.2. One naive way to associate three or more datasets would be to create metrics for all pairwise associations between all pairs of the datasets and use these multiple metrics with the bootstrap methodology.

Another benefit is that the bootstrap methodology allows one to appropriately account for type I error associated with multiple testing. Similar to the selection of an α value in hypothesis testing, one selects δ, the probability that the significance threshold selected will exceed the number of false positives selected, as the threshold criterion. This gives one more control over the probability of observing false positives within the results deemed significant than

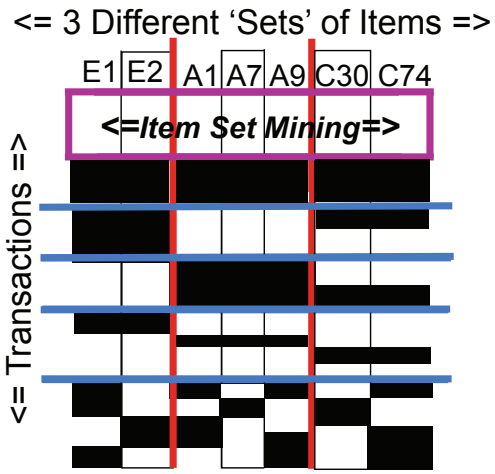


Figure 5.2: Association for Multiple Data Sources.

what is observed with standard corrective measures like the FDR correction. Additionally, the analysis can be focused solely on multivariate associations or other particular associations of interest with the use of the seed nodes in the closed frequent itemset mining algorithm. Where multivariate is defined as including more than one response variable. This enables one to focus the analysis on the associations they are most interested in. As demonstrated in Chapter 4 with the identification of approximate frequent itemsets, the bootstrap methodology can also be used in such a way as to incorporate fuzziness into the results to provide a method that is robust with noisy, inconsistent data. The bootstrap methodology can be adapted to work with approximate itemsets as long as the incorporation of approximation does not impair the rationale behind one's metric of interest.

The primary weakness of this method is intuitive for all methods that use bootstrapping; the resampling and the measurement of the metric of interest on the bootstrap samples coupled with the summarization of the bootstrap results are computationally expensive in both time and space. The most computationally expensive process is the closed and approximate itemset mining. As discussed in Chapter 4, both are time consuming dependent upon the size of the original binary dataset, the density of ones in the dataset, and the support threshold used. Running the FDR correction on the phi coefficients p-values is more efficient than using the bootstrap method. Computing closed or approximate itemsets on the bootstrap samples can

be much more computationally expensive than was observed with the original dataset. This is because generating the bootstrap samples using random sampling with replacement of original dataset can result in producing a much more dense (more ones) bootstrap dataset.

Thus for datasets that produce on average more than 25,000 rules, the method can be computationally infeasible. For example, it took approximately 27 minutes to mine and output results for 22,881 closed itemsets for the ToxCast data and running 1000 such bootstrap samples would take approximately 19 days of runtime if each bootstrap was run sequentially. Furthermore, depending upon the density and size of the original dataset some of the bootstrap samples can produce 4 or more times the number of closed itemsets than what was observed in the original dataset. An increase from 25,000 to 125,000 closed itemsets would result in taking a day as opposed to an hour just to mine all itemsets, let alone to attempt to summarize the results on a bootstrap sample of that size. Therefore, for the bootstrap methodology to be computationally feasibly applied, one must find ways to limit the number of itemsets they want to quantify to be no greater than 15,000 to 20,000. The seed nodes used in our closed frequent itemset mining algorithm provide a natural way to limit ones results as to enable the bootstrap methodology to be used.

The final weakness is that using too large of a support threshold can result in the bootstrap method producing results that were too stringent with regards to the number of significant results provided. We demonstrate how greater minimum support thresholds tend to produce results that are more stringent than what is seen with the FDR correction. The method seems to work best for support thresholds that are around 0.10 or less. This is somewhat dependent upon the density of ones in the original dataset; thus, we provide methodology for determining the criteria that should be used with regards to δ and the support threshold as compared to the results that would have been provided by using FDR correction.

5.2 Methods

5.2.1 Bootstrap Method

Using this methodology frees one from having to use metrics that are associated with a p-value to account for the increase in type I error that is associated with multiple testing. We use the bootstrap methodology to determine which subsets of data that associate response and explanatory variables are most significant regarding the properties that are determined to be most important regardless of whether or not they can be associated to a p-value. To fully enumerate all subsets of data with regards to the association between explanatory and response variables, we use closed frequent itemset mining as we did with the phi coefficient described in Chapter 4. The primary difference is that now we can calculate *any* statistic the user determines to be most important as opposed to solely relying upon the phi coefficient as we did in Chapter 4. The bootstrap methodology that we use is based upon the methods described in Chapter 11 of the text *Quality Measures in Data Mining*, see Lallich et al. (2007) for further details. The methodology used to mine for closed and approximate itemsets to produce subsets of data was described in detail in Chapter 4. We adapted the bootstrap methodology as presented by Lallich et al. (2007) to determine which of our subsets of data are significant while controlling for increase in type I error due to multiple testing as described in detail below.

1. **Empirical Assessment on Original Data:** All rules R are measured using metric M on set of transactions T creating set $M(r)$, $r \in R$. The user must define V_0 as number of false discoveries not to be exceeded given δ. δ defines the probability that the number of false discoveries exceeds V_0. One can think of δ in the same way as one thinks of α with regards to hypothesis testing. Where α is the probability of committing a Type I error, δ is the probability of that the number of false discoveries will exceed V_0. At this first step the user needs to define V_0, the number of false discoveries, δ, the probability of exceeding V_0, and create rule set R using metric M on transaction set T. We created rule set R, such that our rules are the subsets produced using our

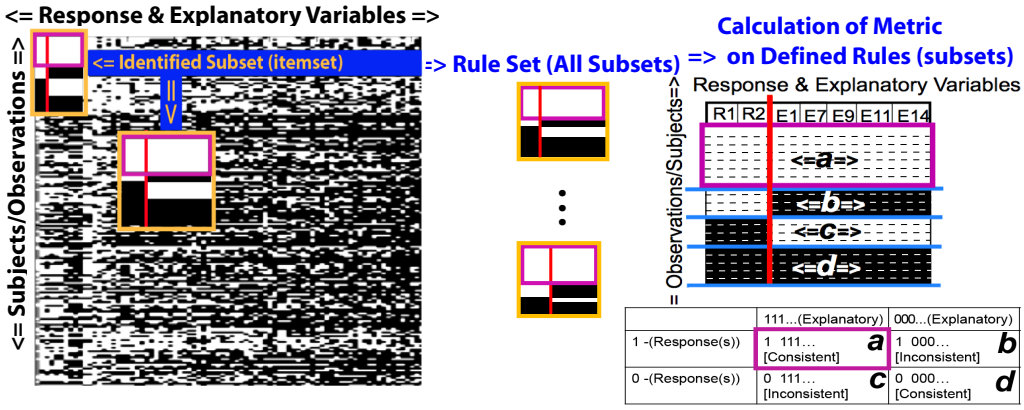


Figure 5.3: Creation of Rules (Subsets) from Transaction Set (Dataset) and calculation of the metric for each rule. Far left depicts original transaction set (dataset) T from which the Rule set R is defined by itemset mining for subsets of the data. Metric, M, is calculated for each of these rules (subsets) based upon contingency table presented on far right.

closed frequent itemset mining algorithm with a minimum support threshold criterion that requires a certain frequency (number) of transactions (observations) on the rules (subsets) that are members of the rule set R. Additionally, our algorithm also requires all resulting rules involve at least one response and at least one explanatory variable. The metric on which each of the rules (subsets) is evaluated is calculated based upon the contingency table associated with each rule (subset), see figure 5.3. Figure 5.3 depicts a graphic that demonstrates rule finding (defining subsets) in the original transaction set (dataset).

2. **Bootstrap to Determine Significant Results while Accounting for Multiple Testing:**

   Repeat l ***times:***

   - Sample with replacement and equal probability n transactions (observations) from T; thus, creating T' where cardinality of T' is the cardinality of T as depicted in section **A** of figure 5.4.
   - Compute $M'(r)$ from T' using M on T' as depicted in section **B** of figure 5.4.
   - Calculate the difference, $M'(r) - M(r)$, and then compute $\varepsilon(V_0, i)$ by using these sorted differences specifically for each $i = 1, 2, \ldots, l$ bootstraps.

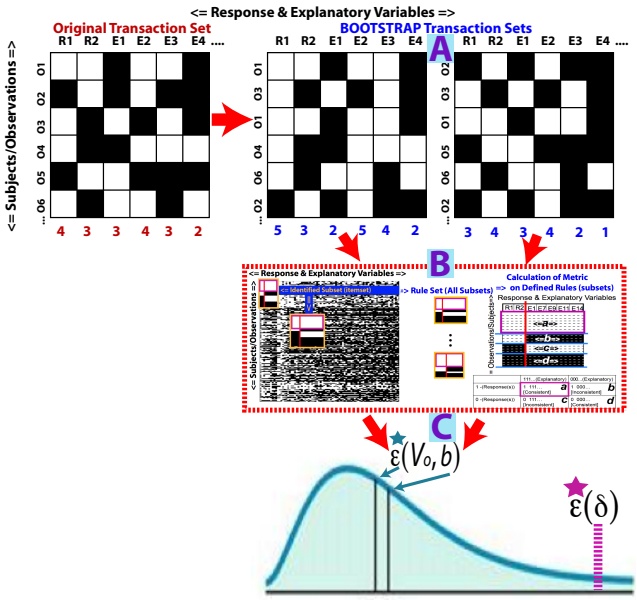


Figure 5.4: Bootstrap Sample Creation using Original Transaction Set (Dataset) and Summarization to find $\varepsilon(\delta)$ Threshold. Top section **A** depicts the resampling of the observations to create the bootstrap sample transaction sets. Middle section **B** shows similar to depicted in figure 5.3 rule set creation and metric calculation for each bootstrap sample. The bottom **C** shows the bootstrap sample summarization with the distribution of $\varepsilon(V_0)$ to find $\varepsilon(\delta)$ threshold.

(a) Rank the differences, $M'(r) - M(r)$, from largest to smallest difference.

(b) Find $\varepsilon(V_0, i)$ given i by finding the $(V_0 + 1)^{th}$ largest element in the ranked difference list that defines this $\varepsilon(V_0, i)$ for any given i.

3. **Summarize the Bootstraps:** This takes all l bootstraps and determines the $\varepsilon(\delta)$ threshold for which rules from rule set R will be judged.

   - Bootstraps $i = 1, 2, \ldots, l$ provides l values of $\varepsilon(V_0, i)$, sort this list of l values of $\varepsilon(V_0, i)$ in descending order.
   - Compute $\varepsilon(\delta)$ using the $\varepsilon(V_0, i)$ list by determining the $(1 - \delta)^{th}$ quantile of this list as the $l * (\delta + 1)$ item in the list such that only $l * \delta$ values in the list are larger than this selected $\varepsilon(\delta)$ as depicted in section **C** of figure 5.4.

4. **Determine Significant Results from Rule Set R using $\varepsilon(\delta)$ Threshold:** Rule set R^* is the significant rules from rule set R based upon threshold $\varepsilon(\delta)$, such that rule r from

transaction set T given metric M where $M(r) > \varepsilon(\delta)$ holds true.

In practice, we discovered that the random sampling that is used to generate the bootstrap samples could produce bootstrap samples that had little overlap in rules as compared to the original rule set. This issue of overlap increased as the minimum support threshold used in rule generation was increased. To control for this issue, we evaluate all bootstrap samples prior to using them in this method to determine how much a bootstrap sample rule set overlaps with the original rule set. A threshold that represents the number of standard deviations below the mean rule set overlap is used to exclude bootstrap samples that are outliers in this regard. For the examples shown in this paper, if a bootstrap sample is more than three standard deviations below the mean rule set overlap it is excluded from use within the method. Under the assumption of normality of the samples, three standard deviations from the mean account for 99.7% of the sample; thus, greater than three standard deviations away from the mean is the standard definition of an outlier.

This bootstrap methodology carefully controls for type I error, probability of false positive, with regards of the selection of δ. Lallich et al. (2007) indicate that the risk of type II errors, probability of false negative, generated by the methodology can be optimized using Hadamard differentiable transformations of the metric to make the measures homogenous through standardization, see van der Vaart and Wellner (1996) for the details. The method as outlined above assumes that the metric of interest is most significant (important) for the largest values of the metric. If this is not the case for your metric of interest, one can simply reverse the scale to effectively use the method as described above. For example, a p-value ranges in value from zero to one, where the most significant values have a value closest to zero. To use this as the metric with the bootstrap methodology as stated above, one can simply subtract the calculated p-values from one to reverse the metric such that the most significant values are the largest.

The bootstrap method as stated above uses a single measure, statistic, that is calculated in determining the significant rules. A nice property about the bootstrap methodology is that if

one has multiple metrics of interest, $M_1, M_2, \ldots, M_n$, all metrics can be used to define a new composite metric ${}^*M(r)$. This is where the composite metric ${}^*M(r) = minimum\{M_1, M_2, \ldots, M_n\}$ allows for the consideration of more than one metric of interest, specifically it ranks a rule's importance by its worst performing metric. To use this multiple metric methodology one should standardize multiple metrics to the same scale, such that each metric considered is given equal weight. However, if one wants to account for multiple properties of interest in a more complex manner, one should combine all properties of interest into a single metric as opposed to using the composite metric ${}^*M(r)$, the minimum of multiple metrics of interest. This ability to include multiple metrics into a composite metric provides one with a means to associate three or more datasets. One could use all pairwise associations between datasets as the $M_1, M_2, \ldots, M_n$ metrics that compose the composite metric ${}^*M(r)$.

We considered three metrics when discussing the bootstrap method. The first metric we considered is the p-value associated with the phi coefficient. This p-value metric is used to verify that our implementation of the bootstrap is correct. We also use the p-value to quantify how δ effects the number of significant rules found as compared to other corrective measures, such as the False Discovery Rate (FDR) and Bonferroni correction. The two metrics that are selected as important with regards to the resulting rules are consistency and a measure of difference from random chance. Consistency is defined as depicted in figure 5.1, as the proportion of consistent results over all observations for which the association between response and explanatory variables are defined, specifically $consistency = (a + d) \ / \ (a + b + c + d)$. Difference from random chance is defined as the difference between the expected and observed probabilities of finding both explanatory and response variables 'on' (one), cell *a* from the contingency table in figure 5.1 represents the value of the observed probability. The expected probability is calculated using the probabilities of the appearance of a one across all observations in the input datasets for the explanatory and response variables. Using these probabilities, one can calculate the expected probability of observing the pattern of ones for a given rule (closed frequent itemset) under the premise that the occurrence of the pattern is

no different than what would be expected given the probabilities observed across the input datasets. The greater the difference between the observed and expected probabilities, indicates the greater association between the explanatory and response variables of the rule. The difference from random chance is simply: $difference = (Prb_{obs} - Prb_{exp})$, when $Prb_{obs} \geq Prb_{exp}$ and $difference = 0$ when $Prb_{obs} < Prb_{exp}$.

Both metrics, consistency and difference from random chance, can range in value between zero and one, and the closer to one in value the more significant the rule should be. The composite metric we use considers both metrics by taking the minimum value between the two metrics for each given rule as described above with metric ${}^{*}M(r)$. To equally weight the metrics, both are rescaled such that each fully span the range zero to one. This is achieved by multiplying each value of the metric by a factor as to ensure that it fully spans the range of one unit, followed by subtracting a second factor from each value to adjust the measure to be between one and zero. The multiplication factor is computed as $multFactor = (1 \ / \ (Max_{metric} - Min_{metric}))$ and the subtraction factor is computed as $subFactor = (Max_{metric} * multFactor) - 1$. The metric must first be calculated on all rules of a rule set to determine the maximum, Max_{metric}, and minimum, Min_{metric}, values. Once these have been determined the multiplication and subtraction factors can be calculated and applied to all rules in a rule set to effectively rescale the metric to fully span the range one to zero.

5.2.2 Method Verification

To verify the bootstrap method, we first created a realistic simulated dataset to quickly evaluate the association between response and explanatory variables. This simulated dataset had to consist of explanatory and response variables that included both observations that supported a strong association and those that showed relatively no association. We selected ten closed frequent itemsets that were mined from the ToxCast data when the minimum support threshold was set to 40 observations (chemicals). All ten selected closed frequent itemsets contained the same two response variables, where four of the itemsets showed a strong association with

ToxCast Closed Frequent Sets used to Create Simulated Data										
Response Variables	Explanatory Variables	R1E1 (a)	R0E0 (d)	R1E0 (b)	R0E1 (c)	Consistent (a+b)/(n)	Phi Coeff	PhiCoeff ChiSqr	P-Value	FDR Adjusted P-Value
e659 e660	a1 a10 a14 a15	41	13	4	15	0.7397	0.4319	13.6156	0.00022	0.02919
e659 e660	a1 a7 a8 a10	45	14	4	17	0.7375	0.4316	14.9053	0.00011	0.02919
e659 e660	a0 a3 a10 a34	40	16	5	15	0.7368	0.4451	15.0575	0.00010	0.02919
e659 e660	a1 a2 a7 a8 a10	44	14	4	17	0.7342	0.4287	14.5205	0.00014	0.02919
e659 e660	a11 a12	41	25	42	36	0.4583	-0.0953	1.3075	0.25284	0.33294
e659 e660	a21 a22	42	35	41	26	0.5347	0.0790	0.8982	0.34325	0.42011
e659 e660	a18 a27	40	36	40	23	0.5468	0.1094	1.6631	0.19718	0.27671
e659 e660	a5 a13	42	22	26	28	0.5424	0.0580	0.3968	0.52874	0.59287
e659 e660	a19	42	32	41	29	0.5139	0.0303	0.1318	0.71654	0.76001
e659 e660	a26	40	34	43	27	0.5139	0.0389	0.2183	0.64032	0.69338

Figure 5.5: ToxCast Closed Itemsets that create Simulated Data. First four closed sets had consistency > 0.73 and FDR adjusted p-values < 0.03. Last six closed sets had consistency < 0.52 and FDR adjusted p-values > 0.33. Ten closed sets composed a dataset of 2 response variables, 20 explanatory variables and 141 observations to create the simulated data.

FDR adjusted p-values of less than 0.03 and consistency greater than 0.73. The remaining six itemsets showed lack of association with FDR adjusted p-value greater than 0.33 and consistency that is less than 0.51. All ten closed frequent itemsets have the properties as described in figure 5.5 and created a simulated dataset of 141 observations that spanned 2 response variables and 20 explanatory variables. This was the simulated dataset that was used hence forth.

The simulated dataset produces a different number of rules (closed frequent itemsets) dependent upon the minimum support threshold selected, here thresholds of 0.125, 0.10 and 0.05 and no threshold (all rules). Figure 5.6 shows, for different thresholds, the number of rules produced and the number of rules determined significant given FDR correction with α of 0.05. Additionally, the table depicts the p-value thresholds and number of rules for the Bonferroni Correction at α of 0.05, 0.10 and 0.125 given the four minimum support thresholds. The primary goal of the corrections methods, including our bootstrap method, FDR and Bonferroni Correction, is to account for the increase in type I error due to multiple testing by providing a more stringent threshold for determining the number of significant rules.

Recall that hypothesis testing provides a formal method for determining whether or not the null hypothesis was rejected based upon the evaluation of a statistic using sample data. The α level, significance level, provides the probability of committing a type I error. A type

Threshold/Support	*All Rules*	0.050	0.100	0.125
# Rules/Results	2,926	2,808	1,900	1,257
Rules for FDR @0.05	191	299	461	463
Est #False Pos FDR @0.05	9.49	14.94	22.98	23.14
Bonferroni Corr @0.05	1.71E-05	1.78E-05	2.63E-05	3.98E-05
Rules for Bonferroni @0.05	0	0	0	0
Bonferroni Corr @0.10	3.42E-05	3.56E-05	5.26E-05	7.96E-05
Rules for Bonferroni @0.10	0	0	1	3
Bonferroni Corr @0.15	5.13E-05	5.34E-05	7.89E-05	1.19E-04
Rules for Bonferroni @0.15	1	1	3	8

Figure 5.6: Significance thresholds based upon FDR and Bonferroni correction on the simulated dataset.

I error describes the error made when one rejects the null hypothesis when it is actually true. Thus, this error is also referred to as the probability of accepting a false positive. The formal methodology of hypothesis testing is designed such that the probability supported by the α value is only guaranteed for a single test. Testing a sample multiple times for the same hypothesis increases the probability that one will observe a significant test, when in actuality the test is not truly significant at the selected α level. This increase in probability reflects that one is observing an increase in the α probability that is not reflected by the selected α value since that value is based upon only testing once. A simple but often overly stringent way to account for this increase in type I error is the Bonferroni Correction, where one divides the selected α level by the number of tests performed. This typically provides a too stringent correction for this type I error; thus, other corrective methods are typically used instead. Figure 5.6 demonstrates how Bonferroni correction is much more stringent than FDR adjustment with regards to significant rules.

The False Discovery Rate, FDR, corrective method is an accepted way to account for the increase of type I error without being overly stringent as with the Bonferroni Correction. FDR provides greater power and less stringency as compared to the other corrections like the familywise error rate (FWER), especially when the number of tests is large (Storey, 2002). Storey (2002) indicates that the FDR assumes that the true proportion of null hypotheses is one; therefore, an even less stringent means of correcting for type I error is through the use

of q-values. Q-values estimate the proportion of true null hypotheses instead of assuming it is one, as is done with FDR correction(Storey, 2002). For our validation and exploration of δ regarding the bootstrap methodology we will compare our results to those produced by using FDR adjustment on the p-value produced by the phi coefficient. The author feels the FDR correction provides an accepted level of stringency given the large number of tests/rules that will result from the method. Moreover, in practice the author has had issue with using q-values when the estimated proportion of truly null hypotheses is small; thus, the FDR correction prevents any failures in adjustment regardless of the true proportion of null hypotheses.

Based upon figure 5.6 we selected the minimum support threshold of 0.05 (7 observations) because it provided that an association had to occur in at least five percent of all observations and resulted in over fifty percent increase in significant rules given total rule decrease (2,926 to 2,808) of four percent. In addition to added strength of association as seen through increase in the minimum number of observations required for a significant rule, this support threshold only saw an increase in false positives at α of 0.05 by five. The top left-hand table in figure 5.7 depicts the results of running the bootstrap method with 1000 bootstraps on the simulated dataset with minimum support threshold set to 0.05 (7 observations). The top table on the far right represents the number of expected significant rules given the 0.05 minimum support threshold using FDR to adjust the p-values for multiple testing. The first column gives an α level, given this α value the second column reports the expected number of FDR corrected significant rules and the third column gives the expected number of false positives.

Both methods are using the same metric, the p-value associated with the phi coefficient. This means if both methods were to report 34 significant rules, they would be the same 34 significant rules because both rule sets were evaluated on the same metric. Therefore, we can determine which δ levels will produce the same number of rules as is seen with the FDR correction. This provides one with more control over the number of false positives observed, given that δ represents the probability of exceeding the selected number of false positives selected (V_0). The dark gray cells in the table in figure 5.7 represent the first δs where the

5% Support Threshold Bootstrap Sampling results using Phi Coefficient P-Value

#False Positives δ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.05	5	17	31	38	47	60	60	84	100	100	100	123	140	143	156
0.10	31	40	40	60	93	100	143	156	181	214	258	267	331	390	453
0.15	36	40	60	100	143	161	192	258	297	390	400	479	563	645	710
0.20	40	60	100	143	176	258	331	453	522	602	655	710	837	921	1033
0.25	60	96	131	161	258	400	517	655	710	822	921	1002	1127	1232	1296
0.30	60	100	156	267	400	563	655	776	921	1055	1157	1264	1452	1532	1663
0.35	100	156	258	400	522	701	829	974	1127	1307	1497	1548	1755	1844	2034
0.40	100	192	390	517	659	871	1127	1251	1429	1552	1764	1843	2041	2178	2288
0.45	156	258	510	655	871	1127	1307	1530	1744	1825	2034	2178	2312	2427	2481
0.50	156	384	655	921	1127	1309	1532	1744	1942	2152	2286	2412	2488	2545	2588
0.55	258	517	835	1127	1349	1656	1825	2034	2196	2371	2477	2541	2585	2616	2644
0.60	297	684	1055	1307	1663	1936	2121	2315	2421	2497	2546	2599	2628	2646	2673

5% Threshold - FDR

α	Expected Results	# False Positives
0.0281	34	0.96
0.0380	54	2.05
0.0410	68	2.79
0.0425	99	4.21
0.0427	129	5.51
0.0445	136	6.06
0.0469	150	7.04
0.0476	180	8.58
0.0482	187	9.01
0.0489	214	10.45
0.0489	229	11.20
0.0491	288	14.15
0.0500	***299***	***14.94***

16% Support Threshold Bootstrap Sampling results using Phi Coefficient P-Value

#False Positives δ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.15	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2
0.20	0	0	0	0	0	0	1	1	1	2	3	4	4	8	11
0.25	0	0	0	0	1	2	4	8	10	15	16	19	27	29	30
0.30	0	0	1	3	4	10	16	19	28	30	32	35	35	36	38
0.35	0	1	4	8	16	21	29	32	35	36	38	44	46	47	50
0.40	1	4	14	20	29	32	36	38	45	46	51	53	57	60	64
0.45	4	14	27	33	36	40	46	50	53	57	63	66	69	72	76
0.50	14	29	35	40	45	51	54	63	67	70	73	77	90	102	112
0.55	27	35	40	51	57	63	70	73	77	90	102	116	137	159	177
0.60	35	46	55	63	70	74	84	101	121	145	159	185	217	231	247
0.65	45	57	66	74	82	106	127	167	196	223	245	258	266	288	308
0.70	56	68	76	105	135	178	217	242	251	260	285	312	339	377	407

16% Threshold - FDR

α	Expected Results	# False Positives
0.0155	66	1.02
0.0223	88	1.97
0.0223	143	3.20
0.0234	174	4.07
0.0244	205	5.00
0.0260	228	5.92
0.0286	243	6.95
0.0315	254	8.01
0.0344	262	9.02
0.0369	271	10.01
0.0393	280	11.01
0.0414	290	12.02
0.0438	299	13.10
0.0457	307	14.04
0.0474	**316**	**14.98**

Figure 5.7: Verification of Bootstrap Method. Tables on right shows FDR adjusted p-values α, corresponding number of significant rules, and estimated number of false positives for those values. Tables on left shows the number of significant rules on bootstrap method at 0.05 and 0.16 minimum support thresholds for 1000 bootstraps using the phi coefficient's p-value as the metric. Light gray shaded cells indicate first δ where the number of rules exceeds the number of false positives, dark gray shaded cells indicate first δ where the number of rules exceeds those resulting from FDR correction (right-hand tables).

number of rules provided by the bootstrap method exceeds those given by the FDR correction. Similarly the light gray cells indicate the first δs where the number of rules provided exceed the number of false positives, V_0. Selecting a δ as small as 0.05 will provide significant rules that exceed the false positives and selecting a δ of 0.10 will give more results than was observed with the FDR correction for 10 or more false positives. The top table in figure 5.7 demonstrates that using the bootstrap method on simulated data with minimum support threshold of 0.05 can provide the same results as the FDR correction with the allowance of only 13 false positives when a δ of 0.10 was selected (instead of 15 with the standard FDR correction). This demonstrates that the bootstrap method as implemented can effectively replicate the FDR adjusted results, in addition to providing a measure of sensitivity through the selection of δ. The δ allows one to bound the likelihood of exceeding the selected number of false positives with regards to the significant results.

It is important to note that with all the analysis presented from this point forth the phi coefficient's the p-value, consistency and the composite metric may not be directly comparable in the sense that they likely have different underlying distributions in the original dataset. We use dark gray shaded cells to indicate the first δs that exceed the number of results provided by the standard FDR adjustment of the phi coefficient's p-value, but *this does not mean we expect similar numbers of significant results* to be given **unless** the bootstrap metric is the the phi coefficient's p-value. We provide this information only to help demonstrate how different minimum support thresholds, scaling, and adjustments to the bootstrap methods compare.

We tested the bootstrap method at higher minimum support thresholds and this test indicated one drawback of the method. As the minimum support threshold increases the stringency of the bootstrap method increases such that the threshold provided by the method would be even more stringent than the Bonferroni Correction at reasonable δ levels. This happens because the greater the minimum support threshold, the less overlap there is between the original rule set and the bootstrap rule sets, where overlap is defined as finding same rule in both rule sets. This causes a problem because it artificially inflates the threshold set by $\varepsilon(\delta)$; thereby,

identifying fewer rules from the original rule set to be significant. This in turn can make the bootstrap method too stringent and even more stringent than the Bonferroni Correction.

Recall that each bootstrap sample will always maintain the same items (response and explanatory variables), but the frequency (number of observations) that each of these items occur will change with each bootstrap sample. The super set of all possible rules is represented by all possible combinations of the items; therefore, the lower the minimum support threshold the greater the likelihood of representing a larger proportion of all possible rules with the original rule set. In turn, representing a larger proportion of all possible rules increases the overlap between the original and bootstrap samples. The overlap between rule sets is important because it sets the threshold $\varepsilon(\delta)$ from which a rule from the original set's significance is determined. When a bootstrap rule does not have a corresponding rule in the original rule set, its difference, $M'(r) - M(r)$, becomes its value of the metric $M'(r)$ because $M(r)$ is zero. The more frequently this occurs, the more this occurrence artificially inflates the threshold $\varepsilon(\delta)$ to be at such a high level that few rules from the original set will be determined to be significant. This problem is exacerbated when the metric of interest is highly dependent upon the observed frequency of a rule. Specifically, if an increase in frequency of the rule results in similar increase in value of the metric, differences between rules, $M'(r) - M(r)$, will be more effected by the lack of overlap between the rule sets. The phi coefficient's p-value is influenced by frequency; whereas, metric like consistency is not. Therefore, this problem will be more pronounced with the p-value as compared to a metric like consistency.

The number of rules observed at different minimum support thresholds is dependent upon the density of ones and the size of the original dataset. If one can mine the data as we did with our simulated data down to a minimum support of one, one can approximate the coverage of a rule set generated by the minimum support threshold. This is done by computing the proportion of the rules generated by a selected minimum support threshold divided by the rules generated when the minimum support threshold is set to one. Using a minimum support threshold of 0.05 generated 96% of all rules (minimum support of one), see figure 5.6. In

comparison, using minimum support threshold of 0.16 (23 observations) will generate 22% of all rules. To demonstrate how poorly the bootstrap method works with too high of a support threshold using a metric that is dependent upon frequency, see the bottom tables in figure 5.7. The left-hand table represents the number of significant rules using the p-value as the metric for 1000 bootstrap samples using minimum support threshold of 0.16 given δ and V_0. The right-hand table reports the results using the same metric and minimum support threshold for the FDR correction. The bottom left-hand table in figure 5.7 demonstrates that to get the same results as what is seen with the FDR correction one must use a δ of 0.70 for V_0 of 11 or more (dark gray cells). Additionally, the number of false positives is only exceeded by the results for δs starting at 0.45 level (light gray cells). This demonstrates how the using a support threshold that is too high can produce results that are more stringent than what is seen with the FDR correction.

Since the stringency of the bootstrap method is dependent upon the overlap between original and bootstrap rule sets and the proportion of rules falling at the minimum support threshold, it is likely that choosing a support threshold of 0.05 or less will provide the desired thresholds. Selecting minimum support thresholds as large as 0.10 may also provide the desired results dependent upon the underlying properties of the dataset. In cases where one cannot compute down to the lowest minimum support threshold to determine the number of *all* rules, one can use the phi coefficient's p-value and the FDR correction as we did in figure 5.7 to determine the likely performance with regards to stringency that the bootstrap method will provide. Note that overly stringent results fail to give the full complement of significant results. However, they do not give more false positives than would be expected given a δ and V_0. In this way being overly stringent is only misleading in the sense that one fails to account for all significant results that could be provided by an alternative method, like FDR correction, if one exists. It should be noted that when dealing with highly inconsistent data one should take care not to set the minimum support threshold too low, for this threshold prevents the discovered subsets of data from being too small in size. Recall that the larger the size of the

subset with regards to shared activity (ones) for both explanatory and response variables, the more likely the discovered association will not be due to random chance.

Figure 5.8 shows results of using the metric consistency in place of the phi coefficient's p-value and explores the effect of rescaling of a metric. The bootstrap method was performed on the simulated data using a minimum support threshold of 0.05 for 1000 bootstrap samples. The consistency metric and the formula for rescaling it to span the entire range of zero to one is described in the methods section. The top table shows the number of significant rules given a δ and V_0 using the metric rescaled consistency and the bottom table depicts results of unscaled consistency. The light gray cells denote the first δ where significant rules exceed the number of false positives (V_0) and the dark gray cells denote the first δ where significant rules exceed the number of rules given by the FDR adjusted p-value (see figure 5.7). Considering the positioning of the gray cells, rescaling consistency seems to provide very similar results. This is because 96% of all rules are represented by the using minimum support threshold of 0.05. In running our bootstrap methodology on other minimum support thresholds with scaled and unscaled consistency, we saw the biggest gains in improvement in stringency when the minimum support threshold was too high. The rescaling guarantees that all rules span the complete range (one to zero), which helps reduce the effects seen by the lack of overlap between original and bootstrap rule sets.

As discussed previously, one advantage of the bootstrap methodology is that it removes the need to use a metric that relies upon having a computable p-value to determine rules of interest. Moreover, using the bootstrap methodology enables one to consider more than a single metric. This composite metric's value is based upon the minimum value of all metrics used. Recall that being able to consider more than one metric offers the advantage of being able integrate three or more datasets using this methodology. As suggested in the methods section one can rescale all metrics involved to give them equally weighted influence on the composite metric used within the bootstrap method. Using the simulated dataset we computed a composite metric ${}^*M(r)$ which was computed as the minimum of the rescaled versions of

5% Support Threshold Bootstrap Sampling results using Rescaled Consistency Metric															
#False Positives δ	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0.05	0	0	0	0	0	0	0	0	0	0	3	11	11	15	15
0.10	0	0	0	0	0	0	3	7	11	15	15	20	25	26	30
0.15	0	0	0	0	3	7	15	15	20	25	26	36	45	53	66
0.20	0	0	0	3	11	15	20	25	26	39	47	62	78	93	100
0.25	0	0	0	11	15	25	26	36	45	62	78	95	116	149	182
0.30	0	0	3	15	20	26	36	62	78	93	107	145	191	252	298
0.35	0	0	11	20	26	39	62	79	100	137	155	244	298	401	512
0.40	0	3	15	26	36	62	79	106	155	207	273	394	512	652	806
0.45	0	11	20	36	62	79	116	155	209	298	416	607	781	887	1023
0.50	0	15	26	53	79	107	182	252	352	419	635	845	1018	1156	1336
0.55	7	25	45	78	107	155	275	408	550	775	887	1091	1290	1487	1571
0.60	15	30	62	100	155	275	419	659	844	1060	1272	1499	1571	1781	1924
0.65	20	47	93	155	273	512	781	962	1197	1466	1571	1776	1900	2029	2147
0.70	26	79	137	325	532	887	1176	1388	1564	1776	1922	2009	2107	2213	2321

5% Support Threshold Bootstrap Sampling results using Unscaled Consistency Metric															
#False Positives δ	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0.05	0	0	0	0	0	0	0	0	0	0	3	7	11	11	15
0.10	0	0	0	0	0	0	3	7	11	11	15	20	20	26	30
0.15	0	0	0	0	3	7	11	15	20	20	26	30	39	53	66
0.20	0	0	0	3	11	15	20	26	26	39	47	66	79	79	107
0.25	0	0	0	11	15	20	26	30	47	66	71	79	107	155	155
0.30	0	0	3	15	20	30	39	53	66	79	107	155	199	252	298
0.35	0	0	11	20	26	39	63	79	107	121	155	252	352	419	659
0.40	0	3	15	26	39	66	79	107	155	252	298	419	659	806	1024
0.45	0	11	20	36	53	79	121	155	252	333	419	740	887	1212	1572
0.50	0	15	30	53	79	121	209	298	419	659	806	1091	1572	2029	2550
0.55	7	26	39	79	121	209	298	419	740	1024	1291	2029	2391	2737	2808
0.60	15	30	66	107	155	333	659	887	1212	1977	2391	2737	2808	2808	2808
0.65	20	53	107	209	333	740	1212	1572	2190	2721	2806	2808	2808	2808	2808
0.70	30	100	155	419	1024	1572	2190	2721	2808	2808	2808	2808	2808	2808	2808

Figure 5.8: Verification of the Rescaling Metrics. Top (scaled) and bottom (unscaled) tables show number of significant rules given δ and number of false positives for the metric consistency using 0.05 minimum support threshold with 1000 bootstrap samples on the simulated dataset. Light gray shaded cells indicate first δ where the number of rules exceed the number of false positives, dark gray shaded cells indicate first δ where the number of rules exceeds those resulting from FDR correction (right-hand table in figure 5.7).

consistency and difference from random (as detailed in the methods section). The results of running 1000 bootstrap samples using this composite metric can be found in the table in figure 5.9. The dark gray shaded cells indicate the first δ where the number of rules exceed those given by the FDR corrected p-value associated with the phi coefficient. The light gray cells indicate the first δ where the number of rules exceeds the of false positives.

Figure 5.9 indicates that one can successfully use a composite metric when calculating the significant rules with the bootstrap methodology. The gray shaded cells occur for the same or smaller δs for the composite metric and indicate that it produces a slightly higher number of results. This suggests that using difference from random chance to create a composite metric increases the uniqueness of the simulated rule set. Here uniqueness is defined by how many of the simulated rule set fall within the tail of the distribution, where the distribution is the one formed by computing the metric on all rules (simulated rule set and bootstrap rule sets). On the simulated dataset, the phi coefficient's p-value produces more significant results than either rescaled consistency or the composite metric (see table 5.7). This is because the phi coefficient's p-value classifies more of the simulated dataset's rules to be in the tail of the distribution of all rules. This property of the simulated dataset is only important if the phi coefficient is more meaningful than rescaled consistency or the composite metric with regards to one's analysis.

Given the potential problems one can encounter with attempting to select a low enough minimum support threshold while being able to run the bootstrap method in a reasonable amount of time, we suggest two solutions to enable one to run this bootstrap algorithm on larger more dense datasets. Because in the first step of the algorithm we list out all the seed nodes that involve at least one response and one explanatory variable, one can use this list to pick the seed nodes that one wants to explore. This selection could be as simple as selecting all seed nodes that contain two or more response variables or by picking a specific response variable to explore. In this way, one limits their results to a manageable number of combinations to allow the algorithm to run on more dense data at appropriate minimum support

5% Support Threshold Bootstrap Sampling results using Rescaled Consistency Metric

#False Positives δ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.05	0	0	0	0	0	0	0	0	0	0	3	11	11	15	15
0.10	0	0	0	0	0	0	3	7	11	15	15	20	25	26	30
0.15	0	0	0	0	3	7	15	15	20	25	26	36	45	53	66
0.20	0	0	0	3	11	15	20	25	26	39	47	62	78	93	100
0.25	0	0	0	11	15	25	26	36	45	62	78	95	116	149	182
0.30	0	0	3	15	20	26	36	62	78	93	107	145	191	252	298
0.35	0	0	11	20	26	39	62	79	100	137	155	244	298	401	512
0.40	0	3	15	26	36	62	79	106	155	207	273	394	512	652	806
0.45	0	11	20	36	62	79	116	155	209	298	416	607	781	887	1023
0.50	0	15	26	53	79	107	182	252	352	419	635	845	1018	1156	1336
0.55	7	25	45	78	107	155	275	408	550	775	887	1091	1290	1487	1571
0.60	15	30	62	100	155	275	419	659	844	1060	1272	1499	1571	1781	1924
0.65	20	47	93	155	273	512	781	962	1197	1466	1571	1776	1900	2029	2147
0.70	26	79	137	325	532	887	1176	1388	1564	1776	1922	2009	2107	2213	2321

5% Support Threshold Bootstrap Sampling results using Composite Metric

#False Positives δ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.05	0	0	1	1	1	2	4	8	12	17	17	20	21	26	30
0.10	0	1	1	2	6	12	17	20	26	26	30	36	39	47	62
0.15	1	1	2	8	17	20	26	30	36	45	53	62	70	78	78
0.20	1	2	5	17	20	30	30	45	53	65	77	92	99	120	142
0.25	2	4	12	20	30	36	45	62	77	92	105	120	153	179	202
0.30	2	8	17	26	36	47	62	78	99	115	142	179	250	322	371
0.35	4	17	26	36	47	63	78	105	135	179	204	281	390	510	570
0.40	8	17	30	45	65	78	106	146	179	249	347	419	595	778	958
0.45	12	26	39	53	78	105	146	202	273	347	510	701	843	995	1174
0.50	17	30	47	77	99	146	204	322	415	537	778	959	1090	1332	1486
0.55	20	39	65	92	136	203	329	419	700	884	1058	1284	1465	1569	1839
0.60	26	53	78	142	183	322	517	773	958	1196	1486	1570	1785	1961	2049
0.65	39	70	106	183	329	517	884	1087	1351	1570	1790	1946	2049	2149	2247
0.70	52	99	154	368	722	1016	1291	1569	1790	1990	2099	2214	2250	2346	2406

5% Threshold - FDR

α	Expected Results	# False Positives
0.0281	34	0.96
0.0380	54	2.05
0.0410	68	2.79
0.0425	99	4.21
0.0427	129	5.51
0.0445	136	6.06
0.0469	150	7.04
0.0476	180	8.58
0.0482	187	9.01
0.0489	214	10.45
0.0489	229	11.20
0.0491	288	14.15
0.0500	***299***	***14.94***

Figure 5.9: Top (scaled consistency) and bottom (scaled composite) tables show the number of significant rules given δ and number of false positives for the metric using 0.05 minimum support threshold with 1000 bootstrap samples on the simulated dataset. Table on the far right shows FDR adjusted p-values α, the corresponding number of significant rules, and estimated number of false positives for those values. Light gray shaded cells indicate first δ where the number of rules exceed the number of false positives, dark gray shaded cells indicate first δ where the number of rules exceed those resulting from FDR correction (right-hand table).

threshold levels. An alternative is to compare the bootstrap rules to rules produced by running the algorithm at a lower support threshold. This helps ameliorate the issue of overlap between bootstrap samples and the original rule set because it increases the overlap between the bootstrap sample rules and the data set they are compared with when determining the threshold set by $\varepsilon(\delta)$. This can be done because the algorithm's rules are closed frequent itemsets; therefore, rules created at lower support thresholds are super sets of rules created at higher ones. Using a rule set that has a lower minimum support threshold than the bootstrap samples and the original rule set, effectively prevents setting the threshold set by $\varepsilon(\delta)$ so high that it will cause the bootstrap method to be too stringent. Additionally, using metrics that are not dependent upon frequency, like consistency, and rescaling those metrics will also alleviate issues related to the bootstrap method's increased stringency at higher minimum support thresholds.

5.3 Results

5.3.1 ToxCast and Thresholding Issues

Using a minimum support threshold of 0.16 (40 observations), the bootstrap method proved to be too stringent in comparison to FDR adjustment on the ToxCast data. Figure 5.10 demonstrates that a minimum support threshold of 0.16 is so stringent that one needs to set δ at 0.45 or greater for at least 350 false positives to get just a few significant results. Compared to the standard FDR adjustment where one would expect over 5,500 results for 350 false positives as depicted in the right-hand table in figure 5.10. Selecting lower support threshold for the ToxCast data becomes computationally infeasible at levels much lower than 0.16 due to the variability in run time and number of rules produced by the bootstrap. Specifically, table 4.6 demonstrates how increasing to more than 25,000 rules increases runtime to be over an hour and table 5.2 establishes how the initial rule set will contain about the median number of rules as compared to the rule sets of the bootstrap samples. Thereby, illustrating that decreasing the minimum support threshold below 0.16 for ToxCast example is computationally infeasible.

16% Support Threshold Bootstrap Sampling results using Phi Coefficient P-Value

#False Positives δ	5	10	25	50	75	100	125	150	200	250	300	350	400	500	750	1000	1250
0.05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2
0.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	5
0.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	5	12
0.35	0	0	0	0	0	0	0	0	0	0	0	0	0	1	5	11	31
0.40	0	0	0	0	0	0	0	0	0	0	0	1	2	2	9	32	56
0.45	0	0	0	0	0	0	0	0	0	0	1	2	2	5	26	55	97
0.50	0	0	0	0	0	0	0	0	0	2	2	5	5	11	45	87	160
0.55	0	0	0	0	0	0	0	0	2	2	5	6	12	31	78	160	265
0.60	0	0	0	0	0	0	0	2	3	5	8	17	28	47	113	232	414
0.65	0	0	0	0	0	0	2	2	5	10	22	38	48	72	198	376	613
0.70	0	0	0	0	1	2	4	5	12	31	47	60	83	141	359	666	1074

16% Support Threshold Bootstrap Sampling results using Rescaled Consistency

#False Positives δ	5	10	25	50	75	100	125	150	200	250	300	350	400	500	750	1000	1250
0.05	1	3	9	17	27	41	55	62	81	100	128	145	168	213	329	430	550
0.10	1	3	12	27	44	62	77	91	129	160	194	224	260	321	479	646	794
0.15	2	6	13	38	59	77	100	121	160	197	250	289	329	422	611	837	1036
0.20	3	6	17	50	72	100	125	151	202	266	318	354	429	537	805	1073	1326
0.25	3	9	21	59	87	121	147	194	260	318	389	467	537	647	977	1283	1574
0.30	3	9	27	66	108	145	168	224	305	389	467	537	615	766	1136	1457	1860
0.35	5	11	30	72	125	168	224	266	350	467	548	615	727	901	1302	1671	2289
0.40	6	13	41	91	146	197	266	318	422	523	609	709	805	1005	1497	1954	2380
0.45	6	13	50	108	168	240	289	350	470	578	687	803	908	1099	1647	2289	2739
0.50	9	17	59	129	196	281	333	395	537	647	794	902	1037	1302	1954	2664	3197
0.55	9	18	66	147	229	305	389	470	636	805	963	1099	1284	1603	2289	3040	3697
0.60	9	22	81	168	266	350	467	537	730	908	1073	1283	1420	1846	2705	3532	4340
0.65	13	30	91	194	305	422	523	615	837	1037	1255	1416	1641	1954	3105	3955	4920
0.70	15	39	110	229	350	497	609	730	977	1254	1455	1649	1954	2358	3505	4512	5569

16% Threshold - FDR

α	Expected Results	# False Positives
0.0292	402	11.7
0.0302	866	26.2
0.0349	1,436	50.2
0.0388	1,914	74.3
0.0419	2,386	100.0
0.0447	2,798	125.1
0.0470	3,203	150.5
0.0516	3,875	200.1
0.0556	4,496	249.8
0.0593	5,073	300.9
0.0635	5,514	350.0
0.0673	5,946	400.2
0.0745	6,723	500.8
0.0917	8,180	750.3
0.1075	9,305	999.9
0.1225	10,211	1,250.4

Figure 5.10: Threshold Problem with ToxCast Data. Right table shows FDR adjusted p-values as α, expected rules produced at minimum support threshold 0.16 given α, and corresponding number of expected false positives. Left tables show number of significant rules using bootstrap method at 0.16 support threshold for 1000 bootstraps using the phi coefficient's p-value(top) and scaled consistency(bottom) as the metrics. Light gray shaded cells indicate first δ where number of rules exceed number of false positives, none of the results exceeded the number of rules resulting from FDR correction (right-hand table).

This makes the ToxCast data a good example of how one can use the alternative solutions enable the use of the bootstrap methodology on a real world example. The bottom table in figure 5.10 demonstrates how using scaled consistency for δ of 0.30 or more and for 150 or greater false positives will provide significant rules that more than exceed the number of false positives. This illustrates that even in cases where the number of significant rules fails to exceed the number of false positives when the phi coefficient is used, other metrics that are not dependent upon frequency may produce an adequate number of results.

One solution to the problem of not being able to use a low enough minimum support threshold is to limit the seed nodes to ones that the user is most interested in. For ToxCast data we selected a support threshold of 0.11(28 observations), for all seed nodes that had three or more response variables. This generated 16,523 rules where the standard FDR adjustment resulted in 6 percent of the rules being significant (1,048) given an α of 0.05. Note that the

FDR adjustment on this dataset was odd in the sense that only 2 of the 1,048 rules were clearly significant (p-value 0.0030) and the remaining 1,046 had a borderline significance as demonstrated by adjusted p-values of at least 0.0488 (see right-hand table in figure 5.11). Table 5.1 show that the rules deemed significant show an association between sets of response and explanatory variable of almost equal size with average consistency of 0.66 and on average a subset of data that includes 88 observations. The bootstrap method was performed for 1000 bootstraps samples with support threshold of 0.11(28 observations) and was limited to rules that contained at least 3 or more response variables.

	# Resp	**# Explan**	**Consist**	**All *'One'* Obs**	**Tot Obs**
Max.	7	10	0 .77	65	243
Min.	3	1	0 .49	28	50
Med.	3	4	0 .66	30	83
Avg.	3.2	4.2	0 .66	31.6	88.2
Std.	0.49	1.39	0 .03	4.32	23.36

Table 5.1: Statistics on 1,048 significant rules given FDR Adjustment at α 0.05. Table contains statistics describing the number of response variables (col 1), the number of explanatory variables (col 2), the consistency (col 3), size of cell *a* from figure 5.1 (col 4), and sum of cells *a-d* from figure 5.1 (col 5).

The tables on the left in figure 5.11 displays the number of significant rules produced using the bootstrap method given a selected δ and V_0 for the metrics phi coefficient's p-value (top) and scaled consistency (bottom). The table on the right in figure 5.11 shows the number of significant rules expected using the standard FDR adjustment, where the dark gray shaded cells indicate the first δ values that will return at least as many significant results as the standard FDR adjustment. Due to only two of the rules being truly significant with regards to the FDR correction, only V_0 of 30 or less have rules that exceed the number expected by the FDR correction. The light gray shaded cells indicate the first δs where the number of rules exceeds the number of false positives. Only the scaled consistency metric provides significant rules that exceed the number of false positives expected. Seemingly, if one selects a δ of at least 0.40, the results will more than exceed the number of false positives. The performance of the bootstrap method coupled with the oddness of phi coefficient, as noted above, indicate that

11% Support Threshold Bootstrap Sampling results using Phi Coefficient P-Value 3+Response Variables

#False Positives δ	5	10	15	20	25	30	35	40	45	50	55	60	65	70	85
0.05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0.20	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2
0.25	0	0	0	0	0	0	0	1	1	2	2	2	2	2	2
0.30	0	0	0	0	1	1	2	2	2	2	2	2	2	2	2
0.35	0	0	0	1	2	2	2	2	2	2	2	2	2	2	2
0.40	0	0	1	2	2	2	2	2	2	2	2	2	2	2	2
0.45	0	1	2	2	2	2	2	2	2	2	2	2	2	2	2
0.50	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2
0.55	1	2	2	2	2	2	2	2	2	2	2	2	3	3	3
0.60	2	2	2	2	2	2	2	2	2	3	3	3	3	4	5
0.65	2	2	2	2	2	2	3	3	3	3	4	4	5	7	8
0.70	2	2	2	2	2	3	3	3	4	7	7	8	8	9	10

11% Support Threshold Bootstrap Sampling results using Rescaled Consistency 3+Response Variables

#False Positives δ	5	10	15	20	25	30	35	40	45	50	55	60	65	70	85
0.05	1	6	6	8	8	12	13	20	21	24	28	29	30	34	35
0.10	4	6	8	8	12	20	24	29	29	34	35	35	36	39	44
0.15	5	7	8	15	21	28	29	34	35	36	38	40	44	52	60
0.20	6	8	12	21	28	30	35	35	38	41	44	52	58	60	69
0.25	6	8	20	24	30	35	36	40	44	52	58	60	64	69	79
0.30	6	12	21	29	34	36	40	44	56	60	64	70	73	79	94
0.35	7	12	24	30	35	39	44	56	60	66	70	79	79	92	113
0.40	8	18	29	35	39	44	56	60	69	76	79	92	98	105	129
0.45	8	20	29	36	41	52	60	70	79	92	94	103	113	120	141
0.50	8	24	34	39	52	60	70	79	92	103	113	120	129	136	169
0.55	12	29	36	44	58	69	79	92	103	119	129	136	150	155	190
0.60	17	34	40	56	66	79	94	106	124	132	144	155	172	181	230
0.65	20	35	52	64	79	92	106	125	141	155	176	190	213	230	301
0.70	24	39	62	79	95	116	130	153	170	188	211	230	255	288	349

11% Threshold - FDR

α	Expected Results	# False Positives
0.0030	2	0.01
0.0488	673	32.85
0.0494	682	33.68
0.0495	700	34.67
0.0497	870	43.23
0.0500	880	43.99
0.0500	**1048**	**52.40**
0.0501	1297	64.95
0.0501	1310	65.63
0.0502	1314	66.00
0.0503	1704	85.66

Figure 5.11: Rules Reduction Solution to Threshold Problem. Tables reflect results that contained 3 or more response variables to allow use of lower support threshold. Right-hand table shows FDR adjusted p-values as α, expected rules produced at support threshold 0.11 given α, and corresponding number of false positives. Left-hand tables show the number of significant rules using bootstrap method at 0.11 support threshold for 1000 bootstraps using the phi coefficient's p-value(top) and scaled consistency (bottom) as the metrics. Light gray shaded cells indicate first δ where the number of rules exceed the number of false positives, dark gray cells indicate first δ where the number of rules exceed those resulting from FDR correction (right-hand table).

at most only 2 rules are truly significant regarding the phi coefficient. This example seems to present a case where the researcher might be truly interested in a metric other than the phi coefficient to quantify significant results (rules).

Another solution to the issue of not being able to select a lower minimum support threshold, is to use a lower minimum support threshold when calculating the difference, $M'(r) - M(r)$, between rules of the bootstrap samples and the original rule set. Specifically, one would use this lower minimum support threshold *only* on the original dataset to produce a larger rule set to only to be used when calculating this difference between the bootstrap rule sets and the original rule set. This rule set at the lower minimum support threshold, would be a super set of the one calculated at a higher minimum support threshold. Using this larger rule set should help with the issue of not having adequate overlap between the original and bootstrap rule

sets. Recall that this lack of overlap causes the threshold set by $\varepsilon(\delta)$ to be so high that it will make the bootstrap method too stringent.

For ToxCast data, the minimum support threshold set to 0.16 (40 observations) to define the original and bootstrap rule sets. For only the difference calculation, $M'(r) - M(r)$, we use a rule set that was generated by the minimum support threshold of 0.12 (29 observations). The tables on the left in figure 5.12 display the number of significant rules produced using the bootstrap method given a selected δ and number of false positives for the metrics phi coefficient's p-value (top) and scaled consistency (bottom). The table on the right in figure 5.12 shows the number of significant rules expected using the standard FDR adjustment, where the dark gray shaded cells indicate the first δs that exceed the number of results given by the standard FDR adjustment. The light gray shaded cells indicate the first δs whose results exceed the number of false positives. This alternative method is able to effectively improve the performance with regards to the phi coefficient, especially at V_0 of 150 or greater. Notice how now with this adjustment a δ of 0.50 for V_0 will produce results similar to what was observed with the FDR correction. Additionally one can now select δs of 0.15 or greater to more than exceed the number of false positives with regards to the scaled consistency metric.

To demonstrate using a different statistic on real data, we used consistency as the metric on ToxCast data. For this comparison we did not use the composite statistic because we wanted to look specifically at the metric that provided motivation for exploring the use of this bootstrap methodology. The metric used is scaled consistency whose calculation is described in detail in the methods section. The bootstrap method was performed for 1000 bootstrap samples and the tables in the figure 5.13 indicate the number of rules that would be consider significant based upon scaled consistency. The first two tables in 5.13 are the results of using scaled consistency as the metric of interest for the entire ToxCast dataset. The top table is the original using minimum support of 0.16, the next is one using minimum support 0.16 but a lower support threshold (0.12) for the calculation of difference for the bootstrap samples. These two tables are directly comparable since they encompass the same 22,881 rule set. The

16% Support Threshold Bootstrap Sampling results using Phi Coefficient P-Value with %12 Threshold for Rules Comparison

#False Positives δ	5	10	25	50	75	100	125	150	200	250	300	350	400	500	750	1000	1250
0.05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	6	26
0.10	0	0	0	0	0	0	0	0	0	0	0	1	2	5	41	105	226
0.15	0	0	0	0	0	0	0	0	2	5	8	21	35	73	246	645	1388
0.20	0	0	0	0	0	0	1	2	9	26	43	66	99	220	890	2457	4380
0.25	0	0	0	0	2	3	6	16	62	126	225	362	527	1080	3064	6145	10372
0.30	0	0	0	2	6	26	53	80	187	336	632	1035	1444	2797	8408	12865	16341
0.35	0	0	0	6	41	103	189	317	776	1393	2059	3387	4788	7764	14737	17164	18472
0.40	0	0	2	31	92	232	431	794	1709	3073	4937	6433	8721	12410	17020	18637	19358
0.45	0	0	8	83	307	721	1348	2078	4033	6566	8947	11459	13452	16026	18230	19280	19862
0.50	0	2	41	230	643	1347	2401	3833	7915	11192	13747	15460	16384	17487	19019	19738	20197
0.55	0	5	83	607	1787	3421	5746	9681	13509	15349	16377	17125	17586	18406	19473	20061	20533
0.60	2	17	271	1494	3946	7744	11441	13348	15496	16661	17288	17892	18308	18871	19805	20381	20771
0.65	10	80	753	4091	9525	12753	14357	15538	16724	17482	18011	18436	18753	19288	20089	20656	21013
0.70	43	230	2279	8907	13154	14641	15812	16507	17466	18116	18587	18959	19237	19707	20429	20861	21230

16% Support Threshold Bootstrap Sampling results using Rescaled Consistency with %12 Threshold for Rules Comparison

#False Positives δ	5	10	25	50	75	100	125	150	200	250	300	350	400	500	750	1000	1250
0.05	3	6	17	50	66	95	128	147	202	281	345	422	473	601	943	1326	1671
0.10	6	13	41	87	147	194	266	318	430	567	690	837	943	1183	1693	2493	3196
0.15	9	17	62	142	222	305	384	497	686	886	1054	1260	1497	1954	3238	4647	5722
0.20	12	27	91	226	345	473	609	738	1073	1416	1726	1954	2427	3106	5088	7069	9125
0.25	17	44	145	318	537	738	933	1136	1591	1954	2522	3040	3538	4551	7123	10198	13360
0.30	27	72	229	491	738	1036	1397	1671	2349	3106	3883	4657	5587	7088	11902	15419	19779
0.35	45	110	331	761	1283	1647	2289	2895	3996	5091	6227	7317	8236	10780	18485	22313	22576
0.40	73	168	491	1175	1948	2687	3476	4122	5722	7544	9643	11468	13654	17695	22449	22586	22654
0.45	114	241	738	1641	2687	3702	5091	6323	8966	11504	14114	16507	19095	22124	22574	22646	22680
0.50	168	345	1005	2294	3923	5719	7701	9301	12791	16988	20035	21919	22315	22493	22627	22675	22697
0.55	266	567	1603	3538	6124	8547	11557	14596	20013	22113	22352	22440	22493	22565	22661	22692	22711
0.60	395	901	2493	5776	9487	14731	18894	21110	22283	22408	22471	22522	22554	22606	22680	22704	22725
0.65	647	1351	4310	10667	17019	20548	22071	22267	22432	22500	22540	22577	22606	22651	22692	22719	22742
0.70	1005	2289	7852	16740	21631	22157	22326	22409	22500	22552	22581	22608	22644	22673	22708	22736	22757

16% Threshold - FDR

α	Expected Results	# False Positives
0.0292	402	11.7
0.0302	866	26.2
0.0349	1,436	50.2
0.0388	1,914	74.3
0.0419	2,386	100.0
0.0447	2,798	125.1
0.0470	3,203	150.5
0.0516	3,875	200.1
0.0556	4,496	249.8
0.0593	5,073	300.9
0.0635	5,514	350.0
0.0673	5,946	400.2
0.0745	6,723	500.8
0.0917	8,180	750.3
0.1075	9,305	999.9
0.1225	10,211	1,250.4

Figure 5.12: Lower Comparison Threshold Solution to Threshold Problem. Right table shows FDR adjusted p-values as α, expected rules produced at support threshold 0.16 given α, and corresponding number of false positives. Left tables show number of significant rules using bootstrap method at 0.16 support threshold for 1000 bootstraps with support threshold 0.12 for rules comparison and phi coefficient's p-value(top) and scaled consistency(bottom) as the metrics. Light gray shaded cells indicate first δ where number of rules exceed number of false positives, dark gray cells indicate first δ where number of rules exceed those resulting from FDR correction (right-hand table).

bottom table is the one where the results have been restricted to 3 or more response variables and minimum support threshold of 0.11 is used. It is not directly comparable to the two tables above it because its 16,523 rule set is a superset of all rules from the 22,881 rule set that contain 3 or more response variables. The dark gray cells indicates the first δs where the number of rules exceed those given by the phi coefficient's FDR adjusted p-value shown in figures 5.11 and 5.10. The light gray cells indicate the first δs where the number of rules exceed the number of false positives.

The top table in figure 5.13 demonstrates how even when the minimum support threshold is too high to give results similar to what is seen with FDR correction (see figure 5.10); using a metric that is not dependent on frequency can provide results that exceed the number of false positives at reasonable levels of δ and V_0. The top table shows that for V_0 of 150 or greater and for δ of 0.30 or greater one can get a reasonable number of results. The next table in figure 5.13 depicts how using a lower minimum support threshold when calculating the difference between the bootstrap and original rule sets can alleviate this issue of a high of a minimum support threshold giving too stringent of results. Top two tables are directly comparable, demonstrating how selecting a δ as small as 0.10 will now provide results that more than exceed the number of false positives for V_0 of 150 or greater. The bottom table establishes that by restricting the results to allow one to use of a smaller minimum support threshold can also provides a viable solution to the threshold problem. In this case selecting a δ of 0.35 or greater will provide results that exceed the number of false positives for all V_0s. In light of the issues that occur with the minimum support threshold, figure 5.13 suggests that scaled consistency might be a better indicator of association than phi coefficient for dealing with noisy inconsistent data. Because the rule sets are not directly comparable; it is difficult to determine which of the two methods used to deal with the threshold issue provided a better result. The rule restriction method encourages the analyst to focus their analysis; thus, perhaps making it a more ideal strategy to use.

16% Support Threshold Bootstrap Sampling results using Rescaled Consistency

#False Positives / δ	5	10	25	50	75	100	125	150	200	250	300	350	400	500	750	1000	1250
0.05	1	3	9	17	27	41	55	62	81	100	128	145	168	213	329	430	550
0.10	1	3	12	27	44	62	77	91	129	160	194	224	260	321	479	646	794
0.15	2	6	13	38	59	77	100	121	160	197	250	289	329	422	611	837	1036
0.20	3	6	17	50	72	100	125	151	202	266	318	354	429	537	805	1073	1326
0.25	3	9	21	59	87	121	147	194	260	318	389	467	537	647	977	1283	1574
0.30	3	9	27	66	108	145	168	224	305	389	467	537	615	766	1136	1457	1860
0.35	5	11	30	72	125	168	224	266	350	467	548	615	727	901	1302	1671	2289
0.40	6	13	41	91	146	197	266	318	422	523	609	709	805	1005	1497	1954	2380
0.45	6	13	50	108	168	240	289	350	470	578	687	803	908	1099	1647	2289	2739
0.50	9	17	59	129	196	281	333	395	537	647	794	902	1037	1302	1954	2664	3197
0.55	9	18	66	147	229	305	389	470	636	805	963	1099	1284	1603	2289	3040	3697
0.60	9	22	81	168	266	350	467	537	730	908	1073	1283	1420	1846	2705	3532	4340
0.65	13	30	91	194	305	422	523	615	837	1037	1255	1416	1641	1954	3105	3955	4920
0.70	15	39	110	229	350	497	609	730	977	1254	1455	1649	1954	2358	3505	4512	5569

16% Support Threshold Bootstrap Sampling results using Rescaled Consistency with %12 Threshold for Rules Comparison

#False Positives / δ	5	10	25	50	75	100	125	150	200	250	300	350	400	500	750	1000	1250
0.05	3	6	17	50	66	95	128	147	202	281	345	422	473	601	943	1326	1671
0.10	6	13	41	87	147	194	266	318	430	567	690	837	943	1183	1693	2493	3196
0.15	9	17	62	142	222	305	384	497	686	886	1054	1260	1497	1954	3238	4647	5722
0.20	12	27	91	226	345	473	609	738	1073	1416	1726	1954	2427	3106	5088	7069	9125
0.25	17	44	145	318	537	738	933	1136	1591	1954	2522	3040	3538	4551	7123	10198	13360
0.30	27	72	229	491	738	1036	1397	1671	2349	3106	3883	4657	5587	7088	11902	15419	19779
0.35	45	110	331	761	1283	1647	2289	2895	3996	5091	6227	7317	8236	10780	18485	22313	22576
0.40	73	168	491	1175	1948	2687	3476	4122	5722	7544	9643	11468	13654	17695	22449	22586	22654
0.45	114	241	738	1641	2687	3702	5091	6323	8966	11504	14114	16507	19095	22124	22574	22646	22680
0.50	168	345	1005	2294	3923	5719	7701	9301	12791	16988	20035	21919	22315	22493	22627	22675	22697
0.55	266	567	1603	3538	6124	8547	11557	14596	20013	22113	22352	22440	22493	22565	22661	22692	22711
0.60	395	901	2493	5776	9487	14731	18894	21110	22283	22408	22471	22522	22554	22606	22680	22704	22725
0.65	647	1351	4310	10667	17019	20548	22071	22267	22432	22500	22540	22577	22606	22651	22692	22719	22742
0.70	1005	2289	7852	16740	21631	22157	22326	22409	22500	22552	22581	22608	22644	22673	22708	22736	22757

11% Support Threshold Bootstrap Sampling results using Rescaled Consistency 3+Response Variables

#False Positives / δ	5	10	15	20	25	30	35	40	45	50	55	60	65	70	85
0.05	1	6	6	8	8	12	13	20	21	24	28	29	30	34	35
0.10	4	6	8	8	12	20	24	29	29	34	35	35	36	39	44
0.15	5	7	8	15	21	28	29	34	35	36	38	40	44	52	60
0.20	6	8	12	21	28	30	35	35	38	41	44	52	58	60	69
0.25	6	8	20	24	30	35	36	40	44	52	58	60	64	69	79
0.30	6	12	21	29	34	36	40	44	56	60	64	70	73	79	94
0.35	7	12	24	30	35	39	44	56	60	66	70	79	79	92	113
0.40	8	18	29	35	39	44	56	60	69	76	79	92	98	105	129
0.45	8	20	29	36	41	52	60	70	79	92	94	103	113	120	141
0.50	8	24	34	39	52	60	70	79	92	103	113	120	129	136	169
0.55	12	29	36	44	58	69	79	92	103	119	129	136	150	155	190
0.60	17	34	40	56	66	79	94	106	124	132	144	155	172	181	230
0.65	20	35	52	64	79	92	106	125	141	155	176	190	213	230	301
0.70	24	39	62	79	95	116	130	153	170	188	211	230	255	288	349

Figure 5.13: Bootstrap Results from *Scaled* Consistency Metric using ToxCast Data. The tables present the number of statistically significant rules for a given δ and false positive. The light gray cells indicate the first δs where the number of rules exceed the number of false positives and dark gray cells indicates the first δs where the number of rules exceed those given by the FDR adjusted p-value shown in figures 5.11 and 5.10.

5.3.2 Approximate Itemsets

The primary difficulty in making the algorithm work for approximate itemsets resides within the comparison of rules between the bootstrap and the original rule sets. Since an approximate itemset is a superset of the original closed frequent itemsets, calculating it can represent a reduction in itemsets from the original set in the sense that certain itemsets with fewer items will disappear because they were incorporated into the approximate set. This results in a decrease of the overlap between the original rule set and the bootstrap rule sets; thus, causing one to choose an even smaller support threshold to attempt to get similar stringency as seen with FDR correction. The initial attempts on incorporating approximate closed frequent itemsets proved to provide results that were too stringent due to the problems with rule overlap mentioned above.

To fix these issues with rule overlap we will implement the following as part of our future work. We will allow a relaxing in the rules comparison such that an approximate rule (rule from approximate itemset) matches any rule that it shares all the same item labels with whether or not those items match on approximation. Note that an item is approximate if some of the observations associated to that item are zero. The priority is given such that if an approximate rule matches another rule exactly (including all approximate items matching), then the difference is calculated using that rule match. If an approximate rule matches a number of other rules by their item labels (not in approximateness), then the median rule (as judged by the metric of interest) is used for the calculation of the difference. Additionally, for the purposes of calculation of the difference for the bootstrap samples, itemsets that were removed because they were incorporated into an approximate itemset are added back to the original dataset (not the bootstraps) for calculation of this difference only. These two adjustments should resolve the stringency issues observed when attempting to get the bootstrap algorithm work with approximate itemsets.

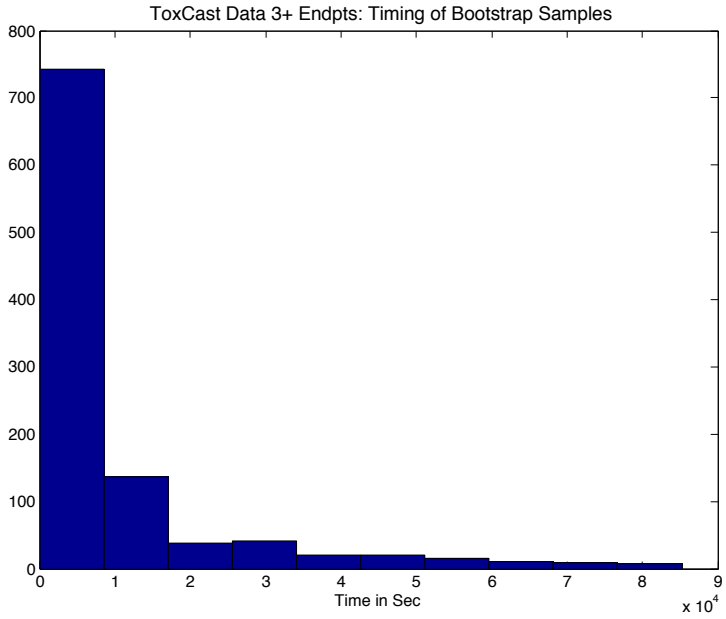


Figure 5.14: Histogram of timing statistics on 1,000 bootstrap samples create for analysis of rules with 3+ response variables.

5.4 Timing

	#Rules	Prop. Rule Overlap	Tot Runtime (min)	Tot Runtime (sec)
Max.	71,060	0.9909	1,420.7	85,242.6
95 Ptile	46,612	0.8546	802.8	48,169.1
75 Ptile	26,406	0.6336	163.7	9,823.5
Median	14,992	0.4617	37.6	2,258.8
25 Ptile	8,134	0.3109	9.6	577.5
5 Ptile	3,054	0.2016	1.7	103.2
Min.	974	0.1440	0.4	24.6

Table 5.2: Rule and timing statistics on 1,000 bootstrap samples create for analysis of rules with 3+ response variables. Table provides the number of rules (col 1), the proportion of overlap between bootstrap sample and original rule set (col 2), total runtime in minutes (col 3), total runtime in seconds (col 4).

The most time consuming part of the method is the closed frequent itemset mining of each of the bootstrap samples. As stated in the introduction, this problem is even further exasperated because a bootstrap sample may have many times the number of rules (closed itemsets) as the original. The worse case scenario, regarding the maximum number of rules observed with a bootstrap sample, is determined by the density and size of the binary dataset

with regards to ones and the minimum support threshold selected. For ToxCast dataset used in the analysis which focused on rules that had three or more response variables, the worst case was an increase from 16,523 to 71,060 rules. The statistics quantifying runtime and number of rules (subsets) generated by the bootstrap samples can be found in the table 5.2. Figure 5.14 is a histogram of runtime calculated in seconds for each of these 1,000 bootstrap samples. The total time the program spent computing the 1,000 bootstraps is 154,480 hours. Additionally, summarizing the bootstrap samples for the phi coefficient's p-value, consistency and scaled consistency took an additional 5 hours and 26 minutes. The programs were all run on UNC's research computing cluster KillDevil, each running on a single Intel EM64T 2.0-2.93 GHz CPU with access to at least 8 GB of memory. In light of these timing issues, we suggest the user limit themselves to 10,000 - 12,000 rule maximum and use a support threshold no greater than 0.15. The ideal runtime scenario was depicted with the simulated data, with less than 5,000 rules and a minimum support threshold of 0.05.

5.5 Conclusions

We have demonstrated that the proposed bootstrap methodology can be adapted to allow a researcher to select any metric of interest with regards to determining which subsets of data show a strong association between response and explanatory variable while accounting for the increase in type I error associated with multiple testing. We established how scaled consistency could be used in place of the phi coefficient in determining which subsets of data are most significant for both real and simulated data. We determined how one can successfully incorporate multiple metrics of interest into the methodology, which enables one to to extend the analysis to integrate more than two datasets in a meaningful way. Furthermore we showed how one could easily restrict their analysis of interest to include only multivariate relationships (those involving two or more response variables). Using the phi coefficient's FDR corrected p-values, we demonstrated how the δ parameter provide a greater deal of control over the

number of false positives in the result. Additionally, using the phi coefficient we demonstrated how the selection of the minimum support threshold influences the stringency of the results returned. We established two methods for preventing the selected minimum support threshold from creating results that are too stringent. Furthermore, we illustrated that certain metrics that are not influenced by the frequency of the observations can provide adequate results even at higher minimum support thresholds. Higher minimum support thresholds may provide the ideal results when the data under analysis is highly inconsistent because larger subsets can demonstrate truer associations. Additionally, we indicated methods for enabling the bootstrap methodology to be adapted for use with approximate itemsets. The weaknesses of this method are intuitive for all methods that use bootstrapping, the bootstrap methodology is computationally expensive and can be infeasible for certain sized datasets given the minimum support threshold that would be required. We provided methodology for restricting one's analysis to those association of greatest interest, which enables this methodology to be used with larger datasets of greater complexity.

Chapter 6

Concluding Remarks

6.1 Conclusions

Classical methods of analysis can fail to identify associations within the data when the data under consideration has nontrivial amount of inconsistency because these methods consider the entire data record. The approaches I develop focus the analysis on subsets of data with internal consistency obscured by the standard methods of analysis. My initial research within the field of metabolomics provides a means to represent NMR spectra as hundreds of aligned peaks as opposed to thousands of unaligned points. Representing spectra in this focused manner improves the relevance of the results generated as detailed in Chapter 2. Additional improvements to enhance NMR spectra analysis can be achieved though the identification of associations that exist over subsets of PCANS aligned data. As demonstrated in Chapters 4 and 5 when analyzing inconsistent data, identifying associations that exist between response and explanatory variables over subsets of data improves the relevance of results. Specifically, our methodology is more adept at focusing exclusively on subsets of data that have an association between response and explanatory variables in comparison to similar methods of subset analysis, like biclustering. As highlighted in Chapter 5, improvements are made to allow the user to employ *any* metric of interest to quantify strength of association between between response and explanatory variables while accounting for the increase in type I error

associated with multiple testing. The methodology discussed in Chapter 5 establishes how multiple metrics of interest can be combined into a measure of strength of association; thus, enabling the integration of three or more datasets in a meaningful way. This methodology has limitations regarding usage on large, complex datasets because of its reliance on bootstrapping. Techniques to handle such limitations are demonstrated in Chapter 5 to illustrate how these methods provide relevant results on real data.

6.2 Future Directions

As mentioned in Chapter 2, there are a number of improvements to be made to the PCANS algorithm. The primary improvement was an update to the peak picking method that now employs an automated statistically based algorithm based on work done by Abdo et al.(Abdo et al., 2006). Another involved adding more functionality to the PCANS webtool as to allow for the visualization of the consensus spectrum based upon the original spectra. Two additional improvements can be made to the alignment algorithm. One would allow for different sized alignment windows to be used based upon the position in the spectrum and the other updates the alignment framework to better handle the alignment of hundreds of spectra.

In Chapter 2, it was discussed how metabolomics is being used to determine multiple responder phenotypes wherein the treated group may contain several subgroups characterized by distinctly different spectra. Standard methods of analysis that consider the entire data record may fail to identify such subgroups. Using the closed/approximate frequent itemset mining methodology of Chapter 4, would provide the means to do such an analysis on PCANS aligned spectra. These methods focus upon subsets of data, which will enable them to identify multiple responder phenotypes within a specific group because they do not require the association to be true for the entire data record.

This type of analysis would use the results of PCANS alignment, a data matrix and the consensus spectrum. One can convert this data matrix into a binary dataset by converting all

peaks to ones. The chemical shift values (peak positions) would be treated as the observations (transactions) and the spectra would be treated as the response and explanatory variables (items). The group classification could define the two classes of variables. For example, control spectra could be the response variables and all treatment spectra could be the explanatory variables. In this sense, strength of association would find subsets (peaks & spectra) where the two groups were most similar. If instead one wanted subsets where the two groups were most different, they could either use the bootstrap method or subtract one from the initial p-values prior to FDR correction.

Another analysis of interest might be to treat all spectra as explanatory variables and create the response variables using the consensus spectrum. The consensus spectrum would be used to create a profile spectrum that contains the majority of peaks that are associated to a specific group, like control or treatment. In this way subgroups strongly associated with the two response variables, the control and treatment profiles, identify peaks that are common to both groups. Whereas, subgroups strongly associated to one response variable, either the control or the treatment profile, identify peaks that are common to only a specific group.

There are three avenues of research that can be further explored based upon the bootstrap methodology in Chapter 5. As outlined in Chapter 5, I would like to account for the problems I encountered when attempting to use the bootstrap methodology with approximate itemsets. In the first attempts, the methodology proved to give results that were too stringent due to the issue with overlap between bootstrap and original rule sets. This issue was exacerbated by the approximate itemsets since they were difficult to compare if one did not relax the rules of comparison. This relaxation of rules comparison would allow itemsets that shared the same item labels to match even if they did not share the same level of fuzziness (approximation). The first avenue of future research would be implementation of the relaxation of rules comparison and evaluate its success to incorporate approximate itemsets into the bootstrap methodology. It may be necessary to restrict the level of approximation allowed and require the metric of interest to be one that is not associated with frequency, like scaled consistency.

Additionally, I would like to demonstrate the integration of three or more datasets through the use of the composite metric. If real data cannot be identified, I can create simulated data to demonstrate how three or more datasets can successfully be integrated with the composite metric. The metrics that compose the composite metric are the pairwise associations between the datasets. Initially, these pairwise associations will be scaled as to give them equal weight. Unequal weighting of the pairwise associations will also be explored.

Finally we will explore using the bootstrap methodology of Chapter 5 to associate statistical importance with Reif et al. (2010) ToxPi measure. This bootstrap methodology can be modified to provide the ToxPi metric with a measure of statistical significance while controlling the probability of observing a false positive results. Specifically, Lallich et al. (2007) defined a rule set, R, that is defined by a metric, M, that is computed based upon transactions, T, defined as *ones* in a binary dataset. Lallich et al. (2007) defined each rule r in the rule set based upon the items that composed a specific transaction. The methodology they use perturbs the transaction set and determines thresholds in which rules are still significant in light of this perturbation. This is no different than the introduction of noise into a mathematical solution to test the robustness of its composition.

In this sense we can define the chemicals that ToxPi is defined upon as our rule set, R. Our transaction set can be defined as the 90 measures that compose the calculation of ToxPi. Since ToxPi is not binary but is measured on a unit circle, each of the 90 measures that compose the transaction set will be given a weight proportionate to their contribution to the ToxPi measure for a given chemical. This weight will be equivalent to the number of times the measure is represented in the initial transaction set. For example, say that attagene assay *ATG_A_TRANS*, that is one of the five assays that compose the *AR* section of ToxPi, contributes to 0.0680 to that of the unit circle of ToxPi for chemical HPTE. If 1000 units compose the transaction set, then 68 of those assigned to assay *ATG_AR_TRANS* will be indicated as one for chemical HTPE. The bootstrap sampling of this transaction set will perturb the units that are assigned to the 90 measures the compose ToxPi for each chemical (rule). The metric would be the

proportion of the ToxPi unit circle a chemical defines. In this way each chemical is a ”rule” with its ToxPi value as the metric on which it is assessed. This would enable the bootstrap methodology to determine statistically important chemicals (rules) as those that fall within the tail of the distribution of the ToxPi metric. If certain (pie) sections of ToxPi are more important with regards to chemical toxicity, their weights can be adjusted to account for this known importance. Furthermore, chemicals with known toxicity can be included within the metric and their positioning can be used to help determine thresholds of toxicity for chemicals of unknown toxicity.

Bibliography

Abdo, Z., Schuette, U., Bent, B. J., Williams, C. J., Forney, L. J., and Joyce, P. (2006). Statistical methods for characterizing diversity of microbial communities by analysis of terminal restriction fragment length polymorphisms of 16s rrna genes. *Environmental Microbiology*, 8(5):929 – 938. 46, 143

Agency, U. E. P. (2011). Summary of the toxic substances control act. 49

America, A. H. P. and Cordewener, J. H. G. (2008). Comparative lc-ms: A landscape of peaks and valleys. *Proteomics*, 8(4):731–749. 12

Anthony, M., Sweatman, B., Beddell, C., Lindon, J., and Nicholson, J. (1994). Pattern recognition classification of the site of nephrotoxicity based on metabolic data derived from proton nuclear magnetic resonance spectra of urine. *Mol Pharmacol*, 46(1):199–211. 3, 12

Baker, S. S., Baker, R. D., Liu, W., Nowak, N. J., and Zhu, L. (2010). Role of alcohol metabolism in non-alcoholic steatohepatitis. *PLoS ONE*, 5(3):e9570. 92, 95

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300. 65, 84

Bradford, B. U., O'Connell, T. M., Han, J., Kosyk, O., Shymonyak, S., Ross, P. K., Winnike, J., Kono, H., and Rusyn, I. (2008). Metabolomic profiling of a modified alcohol liquid diet model for liver injury in the mouse uncovers new markers of disease. *Toxicology and Applied Pharmacology*, 232(2):236–243. 14, 39

Chedzoy, O. B. (2006). Phi-coefficient. In Kotz, S., Balakrishnan, N., Read, C. B., and Vidakovic, B., editors, *Encyclopedia of Statistical Sciences, 2nd ed.* John Wiley and Sons. 83

Cheng, H., Yu, P. S., and Han, J. (2008). Approximate frequent itemset mining in presence of random noise. In Maimon, O. and Rokach, L., editors, *Soft Computing for Knowledge Discovery and Data Mining*, pages 363–389. US Springer. 59, 80, 82

Clemencet, M.-C., Muzio, G., Trombetta, A., Peters, J. M., Gonzalez, F. J., Canuto, R. A., and Latruffe, N. (2005). Differences in cell proliferation in rodent and human hepatic derived cell lines exposed to ciprofibrate. *Cancer Letters*, 222(2):217–226. 63

Cloarec, O., Dumas, M.-E., Craig, A., Barton, R. H., Trygg, J., Hudson, J., Blancher, C., Gauguier, D., Lindon, J. C., Holmes, E., and Nicholson, J. (2005). Statistical total correlation spectroscopy: An exploratory approach for latent biomarker identification from metabolic 1h nmr data sets. *Analytical Chemistry*, 77(5):1282–1289. 15, 42

DiMaggio, P. A., Subramani, A., Judson, R. S., and Floudas, C. A. (2010). A novel framework for predicting in vivo toxicities from in vitro data using optimal methods for dense and sparse matrix reordering and logistic regression. *Toxicological Sciences*, 118(1):251–265. 4, 5, 7, 47, 51, 57, 61, 88, 89, 101

Dix, D. J., Houck, K. A., Martin, M. T., Richard, A. M., Setzer, R. W., and Kavlock, R. J. (2007). The toxcast program for prioritizing toxicity testing of environmental chemicals. *Toxicological Sciences*, 95(1):5–12. 49, 50, 52, 88

EPA (2010a). U.s. environmental protection agencys national center for computational toxicology toxcast project. 49, 50, 88

EPA (2010b). U.s. environmental protection agencys national center for computational toxicology toxrefdb (toxicology reference database). 49, 54, 88

Ferencz, V., Horvath, C., Kari, B., Gaal, J., Meszaros, S., Wolf, Z., Hegedus, D., Horvath, A., Folhoffer, A., and Szalay, F. (2005). Bone disorders in experimentally induced liver disease in growing rats. *World Journal of Gastroenterology*, 11(45):7169–7173. 92, 95

Gartland, K., Beddell, C., Lindon, J., and Nicholson, J. (1991). Application of pattern recognition methods to the analysis and classification of toxicological data derived from proton nuclear magnetic resonance spectroscopy of urine. *Mol Pharmacol*, 39(5):629–642. 3, 12

Gibbons, D. L. and Spencer, J. (2011). Mouse and human intestinal immunity: same ballpark, different players; different rules, same score. *Mucosal Immunology*, 4(2):148–157. 63

Han, J. and Kamber, M. (2006). *Data Mining: Concepts and Techniques, 2nd ed.* Morgan Kaufmann Publishers. 58, 70

Houck, K. A., Dix, D. J., Judson, R. S., Kavlock, R. J., Yang, J., and Berg, E. L. (2009). Profiling bioactivity of the toxcast chemical library using biomap primary human cell systems. *Journal of Biomolecular Screening*, 14(9):1054–1066. 53, 62

Judson, R., Richard, A., Dix, D. J., Houck, K., Martin, M., Kavlock, R., Dellarco, V., Henry, T., Holderman, T., Sayre, P., Tan, S., Carpenter, T., and Smith, E. (2009). The toxicity data landscape for environmental chemicals. *Environmental Health Perspectives*, 117(5):685–695. 49, 50, 52, 88

Kavlock, R. J., Austin, C. P., and Tice, R. (2009). Toxicity testing in the 21st century: Implications for human health risk assessment. *Risk Analysis*, 29(4):485–487. 50

Kim, S., Wang, Z., and Duran, C. M. (2006). A bayesian approach for the alignment of high-resolution nmr spectra. In *Proceedings of the INFORMS Artificial Intelligence and Data Mining Workshop*, pages 1–6, Pittsburgh , PA, USA. 3, 12

Knight, A. W., Little, S., Houck, K., Dix, D., Judson, R., Richard, A., McCarroll, N., Akerman, G., Yang, C., Birrell, L., and Walmsley, R. M. (2009). Evaluation of high-throughput genotoxicity assays used in profiling the us epa toxcast^{tm} chemicals. *Regulatory Toxicology and Pharmacology*, 55(2):188 – 199. 53

Knudsen, T. B., Martin, M. T., Kavlock, R. J., Judson, R. S., Dix, D. J., and Singh, A. V. (2009). Profiling the activity of environmental chemicals in prenatal developmental toxicity studies using the u.s. epa's toxrefdb. *Reproductive Toxicology*, 28(2):209–219. 37th Annual Conference of the European Teratology Society. 49, 54, 88

Kohle, C. and Bock, K. W. (2009). Coordinate regulation of human drug-metabolizing enzymes, and conjugate transporters by the ah receptor, pregnane x receptor and constitutive androstane receptors. *Biochemical Pharmacology*, 77:689–699. 97

Lallich, S., Teytaud, O., and Prudhomme, E. (2007). Association rule interestingness: Measure and statistical validation. In Guillet, F. J. and Hamilton, H. J., editors, *Quality Measures in Data Mining*, volume 43 of *Studies in Computational Intelligence*, pages 251 – 275. Springer-Verlag Berlin Heidelberg. 113, 116, 145

Liu, J., Paulsen, S., Xu, X., Wang, W., Nobel, A., and Prins, J. (2006). Mining approximate frequent itemsets in the presence of noise: Algorithm and analysis. In *Proceedings of the 6th SIAM Conference on Data Mining (SDM)*. 60, 82

Maglich, J. M., Stoltz, C. M., Goodwin, B., Hawkins-Brown, D., Moore, J. T., and Kliewer, S. A. (2002). Nuclear pregnane x receptor and constitutive androstane receptor regulate overlapping but distinct sets of genes involved in xenobiotic detoxification. *Molecular Pharmacology*, 62(3):638–646. 92, 93, 97, 101

Martin, M. T., Dix, D. J., Judson, R. S., Kavlock, R. J., Reif, D. M., Richard, A. M., Rotroff, D. M., Romanov, S., Medvedev, A., Poltoratskaya, N., Gambarian, M., Moeser, M., Makarov, S. S., and Houck, K. A. (2010). Impact of environmental chemicals on key transcription regulators and correlation to toxicity end points within epas toxcast program. *Chemical Research in Toxicology*, 23(3):578–590. 52, 62

Martin, M. T., Judson, R. S., Reif, D. M., Kavlock, R. J., and Dix, D. J. (2009a). Profiling chemicals based on chronic toxicity results from the u.s. epa toxref database. *Environmental Health Perspectives*, 117(3):392–399. 49, 54, 88

Martin, M. T., Mendez, E., Corum, D. G., Judson, R. S., Kavlock, R. J., Rotroff, D. M., and Dix, D. J. (2009b). Profiling the reproductive toxicity of chemicals from multigeneration studies in the toxicity reference database. *Toxicological Sciences*, 110(1):181–190. 49, 54, 88

Mestas, J. and Hughes, C. C. W. (2004). Of mice and not men: Differences between mouse and human immunologye. *The Journal of Immunology*, 172(5):2731–2738. 63

Moran-Salvador, E., Lopez-Parra, M., Garcia-Alonso, V., Titos, E., Martinez-Clemente, M., Gonzalez-Periz, A., Lopez-Vicario, C., Barak, Y., Arroyo, V., and Claria, J. (2011). Role for ppar? in obesity-induced hepatic steatosis as determined by hepatocyte- and macrophage-specific conditional knockouts. *The Journal of the Federation of American Societies for Experimental Biology*, 25(8):2538–2550. 98, 99, 101

Nicholson, J. K., Lindon, J. C., and Holmes, E. (1999). Metabonomics: Understanding the metabolic responses of living systems to pathophysiological stimuli via multivariate statistical analysis of biological nmr spectroscopic data. *Xenobiotica*. 11

Omiecinski, C. J., Heuvel, J. P. V., Perdew, G. H., , and Peters, J. M. (2011). Xenobiotic metabolism, disposition, and regulation by receptors: From biochemical phenomenon to predictors of major toxicities. *Toxicological Sciences*, 120(S1):s49–s75. 97, 98, 101

Plant, N. and Aouabdi, S. (2009). Nuclear receptors: the controlling force in drug metabolism of the liver? *Xenobiotica*, 39(8):597–605. 97

Reif, D., Martin, M., Tan, S., Houck, K., Judson, R., Richard, A., Knudsen, T., Dix, D., and Kavlock, R. (2010). Endocrine profiling and prioritization of environmental chemicals using toxcast data. *Environmental Health Perspectives*. 5, 51, 55, 56, 88, 89, 145

Robertson, D. G. (2005). Metabonomics in toxicology: A review. *Toxicological Sciences*. 11

Rogue, A., Spire, C., Brun, M., Claude, N., and Guillouzo, A. (2010). Gene expression changes induced by ppar gamma agonists in animal and human liver. *PPAR Research*, 2010:1–16. 99, 101

Sasic, S., Muszynski, A., and Ozaki, Y. (2000). A new possibility of the generalized two-dimensional correlation spectroscopy. 1. sample-sample correlation spectroscopy. *Journal of Physical Chemistry A*, 104(27):6380–6387. 16, 46

Savorani, F., Tomasi, G., and Engelsen, S. (2010). icoshift: A versatile tool for the rapid alignment of 1d nmr spectra. *Journal of Magnetic Resonance*, 202(2):190 – 202. 3, 12

Seth, D., Leo, M. A., McGuinness, P. H., Lieber, C. S., Brennan, Y., Williams, R., Wang, X. M., McCaughan, G. W., Gorrell, M. D., and Haber, P. S. (2003). Gene expression profiling of alcoholic liver disease in the baboon (papio hamadryas) and human liver. *The American Journal of Pathology*, 163(6):2303–2317. 92, 93, 99, 101

Storey, J. D. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society. Series B (Methodological)*, 64(3):479–498. 120, 121

Torgrip, R. J. O., Åberg, K. M., Karlberg, B., and Jacobsson, S. P. (2003). Peak alignment using reduced set mapping. *Journal of Chemometrics*, 17(11):573–582. 3, 12

van der Vaart, A. W. and Wellner, J. A. (1996). *Weak convergence and Empirical Processes*. Springer-Verlag New York. 116

van Uitert, M., Meuleman, W., and Wessels, L. (2008). Biclustering sparse binary genomic data. *Journal of Computational Biology*, 15(10):1329 – 1345. 4, 5, 6, 7, 48, 57, 102

Veselkov, K. A., Lindon, J. C., Ebbels, T. M. D., Crockford, D., Volynkin, V. V., Holmes, E., Davies, D. B., and Nicholson, J. K. (2009). Recursive segment-wise peak alignment of biological 1h nmr spectra for improved metabolic biomarker recovery. *Analytical Chemistry*, 81(1):56–66. 3, 12

Wang, J., Zeng, Z., and Zhou, L. (2006). Clan: An algorithm for mining closed cliques from large dense graph databases. *Data Engineering, International Conference on*, 0:73. 58, 71

Webb-Robertson, B.-J. M., McCue, L. A., Beagley, N., McDermott, J. E., Wunschel, D. S., Varnum, S. M., Hu, J. Z., Isern, N. G., Buchko, G. W., Mcateer, K., Pounds, J. G., Skerrett, S. J., Liggitt, D., and Frevert, C. W. (2009). A bayesian integration model of high-throughput proteomics and metabolomics data for improved early detection of microbial infections. In *Proc.14th Pacific Symposium on Biocomputing*, pages 451 – 463. 5, 47, 55, 61

Wiklund, S., Johansson, E., Sjostrom, L., Mellerowicz, E. J., Edlund, U., Shockcor, J. P., Gottfries, J., Moritz, J., and Trygg, J. (2008). Visualization of gc/tof-ms-based metabolomics data for identification of biochemically interesting compounds using opls class models. *Analytical Chemistry*, 80(1):115 – 122. 15, 42

Wong, J. W. H., Durante, C., and Cartwright, H. M. (2005). Application of fast fourier transform cross-correlation for the alignment of large chromatographic and spectral datasets. *Analytical Chemistry*, 77(17):5655–5661. 3, 12

Wu, W., Daszykowski, M., Walczak, B., Sweatman, B. C., Connor, S. C., Haselden, J. N., Crowther, D. J., Gill, R. W., and Lutz, M. W. (2006). Peak alignment of urine nmr spectra using fuzzy warping. *Journal of Chemical Information and Modeling*, 46(2):863–875. 3, 12

Younossi, Z., Afendy, A., Stepanova, M., Hossain, N., Younossi, I., Ankrah, K., Gramlich, T., and Baranova, A. (2009). Gene expression profile associated with superimposed non-alcoholic fatty liver disease and hepatic fibrosis in patients with chronic hepatitis c. *Liver International*, 29(9):1403–1412. 97, 99, 101

Zhang, X., Huang, S., Zou, F., and Wang, W. (2010a). Team: efficient two-locus epistasis tests in human genome-wide association study. *Bioinformatics*, 26(12):i217 – i227. 5, 7, 55

Zhang, X., Pan, F., Xie, Y., Zou, F., and Wang, W. (2010b). Coe: a general approach for efficient genome-wide two-locus epistasis test in disease association study. *Journal of Computational Biology*, 17(3):401–415. 5, 7, 55

Zhang, X., Zou, F., and Wang, W. (2008). Fastanova: an efficient algorithm for genome-wide association study. *ACM Transactions on Knowledge Discovery from Data*, 3(4):821–829. 55

Zhang, X., Zou, F., and Wang, W. (2009). Fastchi: an efficient algorithm for analyzing gene-gene interactions. *Pacific Symposium On Biocomputing*, pages 528–539. 55