8-8-2009

Improved Algorithms for Discovery of New Genes in Bacterial Genomes

Nan Wang

IMPROVED ALGORITHMS FOR DISCOVERY OF NEW GENES IN BACTERIAL GENOMES

By

Nan Wang

A Ph.D Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computer Science
in the Department of Computer Science and Engineering

Mississippi State, Mississippi

August 2009

IMPROVED ALGORITHMS FOR DISCOVERY OF NEW GENES IN BACTERIAL GENOMES

By

Nan Wang

Approved:

Susan Bridges Professor of Computer Science and Engineering (Major Professor)	Eric Hansen Associate Professor of Computer Science and Engineering (Committee Member)
Edward Luke Associate Professor of Computer Science and Engineering (Committee Member)	Changhe Yuan Assistant Professor of Computer Science and Engineering (Committee Member)
Shane Burgess Professor of Basic Sciences College of Veterinary Medicine (Committee Member)	Edward B. Allen Associate Professor of Computer Science and Engineering and Graduate Coordinator

Sarah Rajala
Dean of the Bagley College of
Engineering

Name: Nan Wang

Date of Degree: August 8, 2009

Institution: Mississippi State University

Major Field: Computer Science

Major Professor: Dr. Susan Bridges

Title of Study: IMPROVED ALGORITHMS FOR DISCOVERY OF NEW GENES IN BACTERIAL GENOMES

Pages in Study: 118

Candidate for Degree of Doctor of Philosophy

In this dissertation, we describe a new approach for gene finding that can utilize proteomics information in addition to DNA and RNA to identify new genes in prokaryote genomes. Proteomics processing pipelines require identification of small pieces of proteins called peptides. Peptide identification is a very error-prone process and we have developed a new algorithm for validating peptide identifications using a distance-based outlier detection method. We demonstrate that our method identifies more peptides than other popular methods using standard mixtures of known proteins. In addition, our algorithm provides a much more accurate estimate of the false discovery rate than other methods. Once peptides have been identified and validated, we use a second algorithm, proteogenomic mapping (PGM) to map these peptides to the genome to find the genetic signals that allow us to identify potential novel protein coding genes called expressed Protein Sequence Tags (ePSTs). We then collect and combine evidence for ePSTs we generated, and evaluate the likelihood that each ePST represents a true new protein coding

gene using supervised machine learning techniques. We use machine learning approaches to evaluate the likelihood that the ePSTs represent new genes.

Finally, we have developed new approaches to Bayesian learning that allow us to model the knowledge domain from sparse biological datasets. We have developed two new bootstrap approaches that utilize resampling to build networks with the most robust features that reoccur in many networks. These bootstrap methods yield improved prediction accuracy. We have also developed an unsupervised Bayesian network structure learning method that can be used when training data is not available or when labels may not be reliable.

Keywords: peptide validation, proteogenomic mapping, outlier detection, novel gene generation, novel gene evaluation, Bayesian network

DEDICATION

To Ella and Ryan.

ACKNOWLEDGMENTS

I would like to express profound gratitude to my advisor, Dr. Susan Bridges, for her valuable support, encouragement, supervision, useful suggestions and patience throughout this research work. Her moral support and continuous guidance enabled me to complete my work successfully; without her emotional encouragement it would not have been possible to complete this work. I also express my appreciation to other members of my committee, Dr. Shane Burgess, Dr. Changhe Yuan, Dr. Eric Hansen, and Dr. Edward Luke for their useful suggestions. Special thanks to Dr. Shane Burgess for providing me the experimental dataset and valuable suggestions and to Dr. Changhe Yuan for advising me on the algorithm design.

I am grateful for the cooperation of biologists at the College of Veterinary Medicine at Mississippi State University. First, I really appreciate the kindness of Dr. Mark Lawrence, Dr. Fiona McCarthy and Dr. Bindu Nanduri for providing me with important data for analysis. Second, the students and research assistants working for Dr. Shane Burgess and Dr. Mark Lawrence have helped me on many aspects of my work.

I am as ever, especially indebted to my mother, Mrs Qunfang Kou for her love and support throughout my life. I am grateful to my beloved husband, Junxiao for his emotional support and my wonderful kids Ella and Ryan for bringing joy to my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

One of the major accomplishments in biology over the past 20 years is the development of technologies for determining the genomic sequence of living organisms. The genome of an organism can be viewed as a sequence of four nucleotides (abbreviated A, T, C, and G) comprising its DNA and containing all of the biological information needed to build and maintain life. The size of genomes varies widely from 5386 characters for the virus Phi-X to 3.3×10^9 characters for human [43]. Each character is called a base-pair (bp) due to the double stranded nature of DNA. The explosion in genome sequencing has driven the development of computational techniques to identify functional elements such as genes in the genomes [20]. Most computational gene finders use the sequence of known genes and features of the nucleic acid sequence to build models of gene structure that can then be used identify genes in the genome [7, 20, 56]. Figure 1.1 shows a simplified version of the Central Dogma of Molecular Biology and illustrates how genes (DNA) are transcribed to messenger RNA (mRNA) and then translated to protein. In this dissertation, we will describe a new approach for gene finding that can utilize proteomics information in addition to DNA and RNA to identify new genes in the genome. The first step in this process is identification of small pieces of proteins called peptides. Peptide identification is a very error-prone process and we have developed a new algorithm for validating peptide

identifications using a distance-based outlier detection method. Once peptides have been identified and validated, we use a second algorithm to map these peptides to the genome to find the genetic signals that allow us to identify potential new genes called expressed Protein Sequence Tags (ePSTs). We then collect and combine evidence using supervised machine learning techniques to evaluate the likelihood that the ePST represents a new protein coding gene based on the training dataset provided by biological experts. We have applied Bayesian networks to learn models of these potential novel protein coding genes and to determine the likelihood that the potential new genes are actually protein coding genes. We have developed a new data-driven unsupervised Bayesian learning algorithm and used the models learned by this algorithm to evaluate potential new genes.

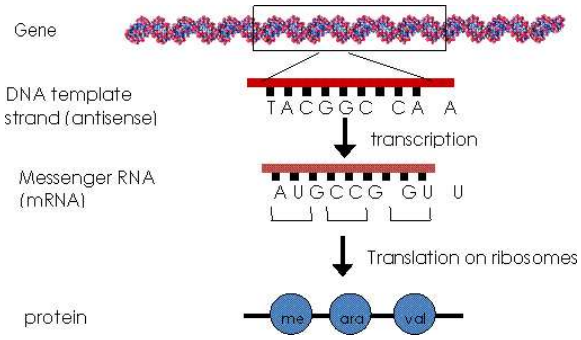


Figure 1.1

Central Dogma of Molecular Biology

1.1 Brief Biology Background

All hereditary information about an organism is contained in its genome [43]. The genome is organized in chromosomes that are composed of long strands of deoxyribonu-

cleic acid (DNA) [43]. Discrete units of the chromosomes contain the code for proteins and these units are called genes [43] as shown in Figure 1.1. The DNA is transcribed to mRNA and then translated to protein. While both DNA and RNA are composed from a 4-letter alphabet, proteins are composed of long chains of amino acids and can be viewed as strings from a 20-letter alphabet [43]. Proteins are the major functional molecules in cells. The genetic code is used by organisms to translate from the 4-letter DNA/RNA alphabet to the 20-letter protein alphabet as shown in Figure 1.1. Each amino acid in a protein is specified by at least one 3-letter DNA/RNA codon. In addition, there are special codons called stop codons that specify the end of a gene.

Organisms such as bacteria that do not have membrane bound organelles (prokaryotes) have a much simpler gene structure than organisms such as plants and animals with membrane bound organelles (eukaryotes). In this dissertation, we focus on gene-finding in prokaryotes.

1.2 Motivation

The explosion in genomic sequencing has led to the availability of a large number of genomes over the last decades and these genome sequences are now publicly available [40]. However, the genome sequences by themselves are of little use. True value is derived from the genome sequence only after the genes have been identified (structural annotation) and the function of the protein product has been determined (functional annotation). After a genome has been sequenced, gene prediction programs are used to predict genes. For example, most computational programs for structural annotation in prokaryotes (bacteria)

are based on features in the nucleic acid sequence. GeneMark [56] and Glimmer [20] are popular gene finding tools, and both are based on Hidden Markov models (HMMs) at the nucleic acid sequence level. Although these tools are widely used, they are known to have a number of shortcomings including false negative identifications (failing to identify genes that exist), false identifications, and incorrect identification of gene boundaries [47]. Because of the high false identification rate obtained for prediction of short genes, these algorithms usually use a somewhat arbitrary length cutoff and are therefore particularly ineffective at identifying novel short genes.

Mass spectrometry [57] is a popular technique for detection of proteins in biological samples. It provides direct molecular evidence of the existence of the protein in the living cell. Proteins are typically identified using mass spectrometry by computationally matching experimental mass spectra against theoretical spectra derived from a protein database. However, several groups have recently reported the use of mass spectral data to identify genes on the genome [47]. This process was named proteogenomic mapping by Jaffe *et al.* [40]. We have designed algorithms that use experimental protein data from mass spectrometry to find genes on genomes.

Mass spectrometry cannot be used to identify proteins directly. Instead, the proteins are cleaved into small pieces called peptides by enzymes such as trypsin that cut the protein in specific places. The peptides are assigned to mass spectrometry and identified by matching the resulting spectra against a database of theoretical spectra. Once the peptides have been identified, they are mapped to the parent protein. Peptide identifications based on mass spectrometry are extremely noisy and represent a mixture of true and false

identifications. Most popular proteomics search algorithms such as SEQUEST [16] and Mascot [46] provide a set of scores for each peptide assignment. There are usually many more false identifications (noise) than true identifications

and thus the true identifications can be viewed as outliers in the search-score-space as shown in Figure 1.2. The x and y axes in this graph represent two scores commonly used for peptide validation with the SEQUEST search algorithm. We have developed a distance-based outlier detection algorithm [5] to distinguish correct peptide identifications from noise. Based on the distances to the K nearest known false identifications in search-score-space, we build a probabilistic model that is used to calculate the probability that a peptide with a certain distance score is a true identification. This method can be used to validate peptides identified by searching against either a protein database or a translated nucleotide database. We demonstrate that our new algorithm identifies many more peptides than the most popular methods with standard datasets of known proteins. In addition, our new algorithm identifies as many peptides as a new algorithm recently published in [45] but with a much more accurate false discovery rate.

In proteogenomic mapping, peptides are identified by matching mass spectra against a database generated by translating the genomic sequence. Once peptides searched against this database have been validated, they can be mapped back to genomic sequence to identify potential protein coding genes. Many of these will be a part of proteins from known genes, but some others will map to places in the genome where no genes have been identified and thus they represent potential novel protein coding genes. We map peptides to the genome and then extend the nucleotide sequence corresponding to the peptide in both

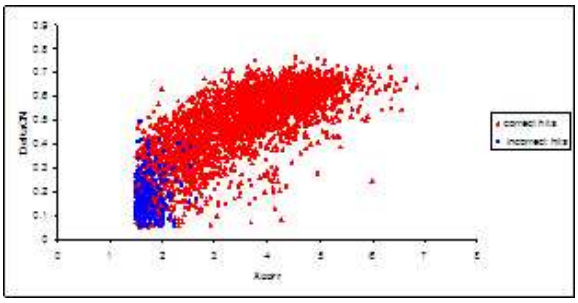


Figure 1.2

Distribution of known correct and incorrect peptide identifications for a control sample in search-score-space

directions identifying a possible start codon (beginning of a gene) and a stop codon (end of a gene). This extended sequence is called an ePST. But even with accurate peptide identifications, we know that some of the ePSTs may not correspond to genes. Therefore, we collect orthogonal evidence for the validity of each ePST as a gene. We represent this evidence in a feature vector and use machine learning techniques to determine the likelihood that an ePST is a true protein coding gene.

1.3 Statement of Hypothesis

Our hypothesis is that a computational proteogenomic mapping pipeline for structural annotation of bacterial genomes can be used effectively to confirm the existence of predicted genes, to identify novel genes, and to correct boundaries of predicted genes. A design for the pipeline is shown in Figure 1.3. Two key components of the pipeline are novel algorithms for 1) validating peptides and 2) integrating evidence supporting or refuting novel genes discovered by the pipeline. A semi-supervised machine learning technique is used for accurate assignment of peptides to spectra (peptide validation) using probabilistic

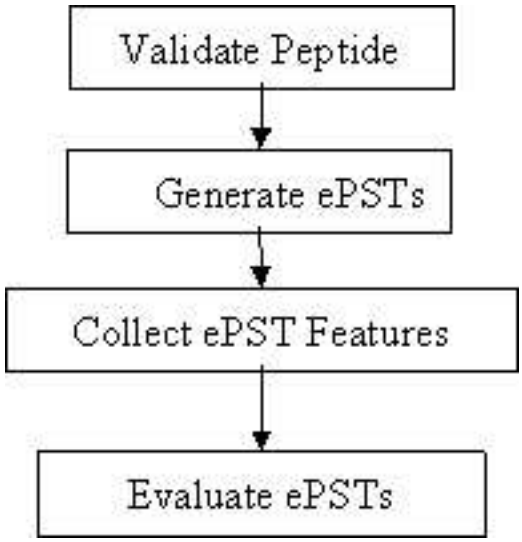


Figure 1.3

Proteogenomic Mapping Pipeline

approaches that model the distribution of noise and true signal. Integration and evaluation of the evidence supporting or refuting novel genes is accomplished using machine learning techniques.

1.4 Contributions

The dissertation describes a set of algorithms that have been developed and experiments that have been conducted making the following contributions:

1. We demonstrate the efficacy of a concatenated database when using a target-decoy strategy to determine the false discovery rate during peptide validation.

2. A new algorithm for validation of peptide identifications based on outlier detection combined with Bayesian reasoning has been developed.

3. A new algorithm for discovering potential novel protein coding genes (ePSTs) has been developed. This algorithm combines results from searching MS/MS spectra against both a genome database translated in 6 reading frames and a protein database to identify those peptides that represent potential novel genes or that can be used to correct gene boundaries.

4. A new algorithm has been developed for collecting relevant features of potential new genes and using the features with machine learning algorithms to evaluate the likelihood that the ePSTs represent novel genes or corrections to boundaries of known genes.

1.5 Organization

The remainder of this dissertation is organized as follows. Chapter 2 provides a review of the background literature in computational gene finding, peptide validation and gene evaluation. Chapter 3 presents the algorithm for peptide validation based on a distance-based outlier detection method. Chapter 4 presents our proteogenomic mapping algorithm for discovering potential new genes, and for collecting features describing these potential new genes, and for evaluating the likelihood that the genes are "real." Chapter 5 describes Bayesian learning approaches for evaluating potential novel genes and a new unsupervised Bayesian network model. Finally, we summarize the algorithms developed and their significance and discuss future extensions.

CHAPTER 2

LITERATURE REVIEW

The explosion in genomic sequencing has produced many publicly available, complete genomic sequences [81]. Recently developed sequencing technologies are producing complete genomic sequences at an unprecedented rate [27]. At the time of this dissertation, 863 complete microbial genomes have been deposited in the NCBI GenBank database. These new genomes contain thousands of new genes, which are put into public databases and become the basis for further research. Therefore accurate microbial gene identification is becoming more important than ever. Computational gene-prediction algorithms are the standard method for identification of genes in newly sequenced genomes with manual curation used only as a last step. Therefore it is essential that these algorithms be as accurate as possible.

Currently, there are three categories of computational methods for identification of new genes in prokaryotic genomes. Homology search methods, such as BLASTX [2, 30], FASTA [3] and ORPHEUS [2, 26] discover new genes in a genome based on sequence similarity to known genes in other species . Computational gene finding programs that do not rely on sequence similarity include the GeneMark series [9, 56], Glimmer [20], ZCURVE [34], GS-Finder [70] and MED [89]. These algorithms use a variety of methods to build models of genes including Hidden Markov models, Z-curve representation, and

other statistical techniques. One system, EasyGene, combines model building and homology search [53]. In recent years, tandem mass spectrometry [16, 29, 48, 83] has been used increasingly for high-throughput analysis of protein samples. Using the advances in proteomics, a number of researchers [38, 40, 46, 47, 57, 60, 84] have demonstrated using a combination of genomic and proteomic data can be used to improve structural annotation of genomes. This process has been termed proteogenomic mapping by Jaffe [40]. This dissertation describes new algorithms for proteogenomic mapping in prokaryotic genomes.

This chapter first provides an overview of traditional methods for computational gene finding based on sequence similarity and on models of genes in genomic sequence. Because the method that we have developed for gene finding is dependent upon accurate identification of peptides in complex protein mixtures analyzed by MS/MS, we have developed a new method for peptide validation. We therefore review previous work in peptide validation and then describe previous work in proteogenomic mapping. Because our work also includes a component that evaluates gene models, we also review prior work in gene evaluation methods.

2.1 Computational Gene Finding and Evaluation in Prokaryotic Genomes

This section reviews two major approaches for computational gene finding in prokaryotic genomes and some methods used to evaluate predicted genes.

2.1.1 Homology Search Gene Finding Methods

Early computational gene prediction methods were based on sequence similarity search using program such as BLASTX [2, 30], FASTA [3] and ORPHEUS [2, 26]. Sequence

similarity between a translated nucleotide sequence and a known biological protein can provide strong evidence for the presence of a homologous coding region, even between distantly related genes [2]. For example, the computer program BLASTX [30] translates the nucleotide query sequence in all six possible reading frames and then searches a protein database for the sequences similar to the translated sequences. The sensitivity of BLASTX recognition is characterized to the presence of substitution, insertion and deletion errors in the query sequence and to sequence divergence [2]. BLASTX can be used with large scale sequencing projects, even when the sequence may contain errors such as frame shifts. The BLAST family of algorithms is the most widely used bioinformatics program and is undergoing constant improvement [2, 30]. FASTA is another sequence similarity algorithm [3] that includes a heuristic to generate a gapped alignment and that has been used for homology based gene finding [2, 3, 30]. However, many genes in newly sequenced prokaryotic genomes do not show significant similarity with known genes and therefore cannot be identified using homology search [69].

2.1.2 Model Building Gene Finding Methods

Major methods used for gene finding in prokaryotes are based on signal processing methods: Hidden Markov Models and Z-curve. The most widely used tools are GeneMark and Glimmer. The GeneMark series [8, 9, 56] and Glimmer [19, 20] both employ inhomogeneous (frame dependent) Markov models trained with existing gene data. The learned model provides an estimation of the likelihood that a DNA segment belongs to a protein coding sequence. GeneMark [9, 56] and Glimmer [19, 20] build Markov chains

for both coding and non-coding regions and combines these models with Bayes' decision making. This yields an Interpolated Markov Model [19] that combines Markov models from 1^{st} through 8^{th} order, weighting each model according to its predictive power.

ZCURVE [33, 34] is based on the Z curve representation of the DNA sequences and relies on global statistical features of protein-coding genes by taking the frequencies of bases at the three codon positions into account. In ZCURVE, a total of 33 parameters are used to characterize the coding sequences.

Another group of tools uses the same basic approaches for gene finding as GeneMark and Glimmer, but follows the basic gene finding step with an evaluation phase that evaluates evidence that the potential ORF is a true gene. Tools that use this approach are reviewed below.

ORPHEUS [26] combines diverse evidence to recognize genes in completely sequenced bacterial genomes. It is based on the assumption that coding regions derived from similarity searches are more reliable than statistical data. The analysis starts with a database similarity search to identify reliable gene fragments (seed ORF). The reliable gene fragments are then used to derive statistical characteristics of protein-coding regions and ribosome-binding sites, and used to calculate coding potential parameters. At the next step, the sample of ORFs with possible start codon is used to derive the RBS recognition matrix. The ORF with start codon having strong RBS is selected as potential novel gene.

Easy Gene [53] estimates the statistical significance of a predicted gene. The first step is to apply a gene finder based on a hidden Markov model. The HMM is estimated by extracting a training set of genes from the genome using extensions of similarities in a

comprehensive protein database. Putative genes are then scored with the HMM, and based on the score and length of the ORF, the statistical significance is calculated. The measure of statistical significance for an ORF is the expected number of ORFs in one megabase of random sequence at the same significance level or better, where the random sequence has the same statistics as the genome in the sense of a third order Markov chain.

MED [89] is a non-supervised gene prediction algorithm for bacterial and archaeal genomes. It is based on a comprehensive statistical model of protein coding Open Reading Frames (ORFs) and Translation Initiation Sites (TISs). MED first applies an ORF model based on a linguistic "Entropy Density Profile" (EDP) of coding DNA sequence to identify potential coding ORFs. This sequence is used as input for a TIS refinement component that checks for several relevant features related to translation initiation. The flow chart of process of MED is shown as Figure 2.1. This approach is similar to ours in a broad sense in that it first looks for potential ORFs and then looks for additional evidence to support the ORF as a protein coding gene.

FrameD [76] was initially designed to predict coding regions in GC rich bacterial genomes that may contain frame shifts. FrameD is based on a graph model where gene overlap is specifically modeled leading to a good specificity of its predictions. This model includes RBS finding, probabilistic coding models and possible protein similarities.

GS-Finder [70] finds bacterial gene start sites with a self-training method without *pri-ori* knowledge of rRNA in the genomes concerned. GS-Finder includes a two step process. The first step is finding potential novel ORFs using existing gene finding programs, and the second step is evaluating potential novel ORFs. Features evaluated include mononu-

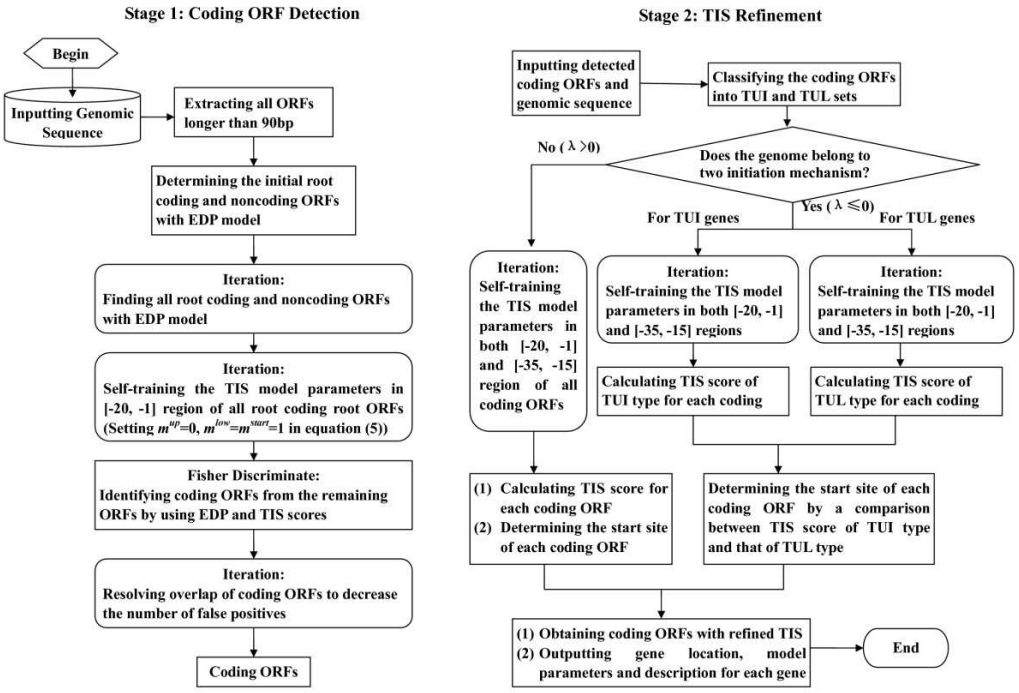


Figure 2.1

Flow chart of gene prediction process with MED system

cleotide distribution patterns near the start codon, the start codon itself, the coding potential, and the distance from the left-most start codon to the start codon. The self-training method is also used to relocate the translation start sites of putative ORFs of genomes.

All of these methods are based on analysis of genome sequence and on the assumption that all genes in the organism will have the same characteristics as previously known genes.

2.2 Outlier Detection

Our peptide validation method is an outlier detection based method. An outlier is defined by Hawkins as "an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism" [35]. In other words, an outlier can also be viewed as an exception of a dataset. Outlier/exception detection is one of many general categories of knowledge discovery. Some applications of outlier detection include detections of credit card fraud [25], network intrusion detection [24], monitoring of criminal activities in electronic commerce [21]. In the current study, Peptide identifications generated from SEQUEST search results are mixture of correct and incorrect identifications and correct identifications can be viewed as outliers of incorrect identifications (noise). Thus, an outlier detection approach can be used to discriminate correct peptide identifications from incorrect peptide identifications.

Outlier detection approaches include supervised-based methods, where each example is labeled as exceptional or not [25], and unsupervised-learning based methods, where labels are not needed [5, 21, 24]. The existing supervised outlier detection approaches are statistical-based models where the data is assumed to follow a certain parametric distribu-

tion [25]. These statistical model-based approaches do not work well in high-dimensional spaces, and it is also hard to find the right distributions to describe the dataset. To overcome these limitations, researchers have proposed non-parametric approaches including distance-based approaches [5, 21, 24], clustering-based approaches [1], and density-based approaches [54].

In the real world, labeled datasets are not always available, and thus unsupervised-learning based outlier detection methods are widely used. Distance-based outlier detection approaches were first presented by Knorr *et al.* [5, 21, 24], who define a point to be a *distance outlier* if at least a user-defined fraction of the points in the data set are further away than some user-defined minimum distance from that point. The distance-based outlier detection can be done for large datasets and for multi-dimensional datasets. When the dataset is huge, the calculation of distance among the points becomes expensive. Also when the data variables are scaled in different ranges by normalization, the distance can avoid the bias from the data variables. Related to distance-based methods are methods that cluster data and find outliers as part of the process of clustering. Points that do not cluster well are labeled as outliers [86]. In density-based approaches to outlier detection, a local outlier factor (LOF) is computed for each point [58]. The LOF of a point is based on the ratios of the local density of the area around the point and the local densities of its neighbors.

Angiulli *et al.* [5] proposed an algorithm based on K-nearest-neighbor distance for outlier detection and prediction. There are two steps of the algorithm. A distance-based outlier detection method finds the top outliers and provides a subset of the outliers called

the outlier detection solving set. This solving set is used to predict the outlierness of new objects. The solving set includes a sufficient number of points that can be used for detection of the top outliers by considering only a subset of all the data. The properties provide subquadratic time requirements for detection and prediction of a new point. In the dissertation, a K-nearest-neighbor distance based outlier detection method with Fabrizios strategy is used for peptide validation.

2.3 Peptide Validation

The goal of proteomics research is to identify the set of proteins expressed in a cell or tissue. In recent years, tandem mass spectrometry [16] has been used increasingly for high-throughput analysis of protein samples. As shown in Figure 2.2, proteins in a sample are digested into peptides and the peptides are then ionized and fragmented to produce signature MS/MS spectra that are used for identification. Peptide identifications are made by searching MS/MS spectra against theoretical spectra generated from a protein sequence database and finding the best matching spectra. The identified peptides are then mapped back to the protein sequences and protein identifications are made based on peptide identifications. Thus, accurate identification of peptides is essential for accurate identification of proteins. In addition, in our proteogenomic mapping pipeline, we identify potential novel genes based on peptide identifications and thus accurate peptide identification is also an essential component of this process.

A variety of algorithms for automated identification of peptides based on matching their masses and fragmentation patterns have been developed including SEQUEST [64],

Mascot [46], and X!Tandem [16]. These algorithms compare an observed MS/MS fragmentation pattern from an unknown peptide (observed spectrum) with those fragmentation patterns predicted (theoretical spectra) for all peptides of equivalent mass within a given protein database and return the peptide sequence with a theoretical spectrum that best matches the observed spectrum. Each returned peptide sequence is assigned a set of scores that reflects various aspects of the fit between the observed spectrum and the theoretical spectrum. Figure 2.2 illustrates the process of protein identification.

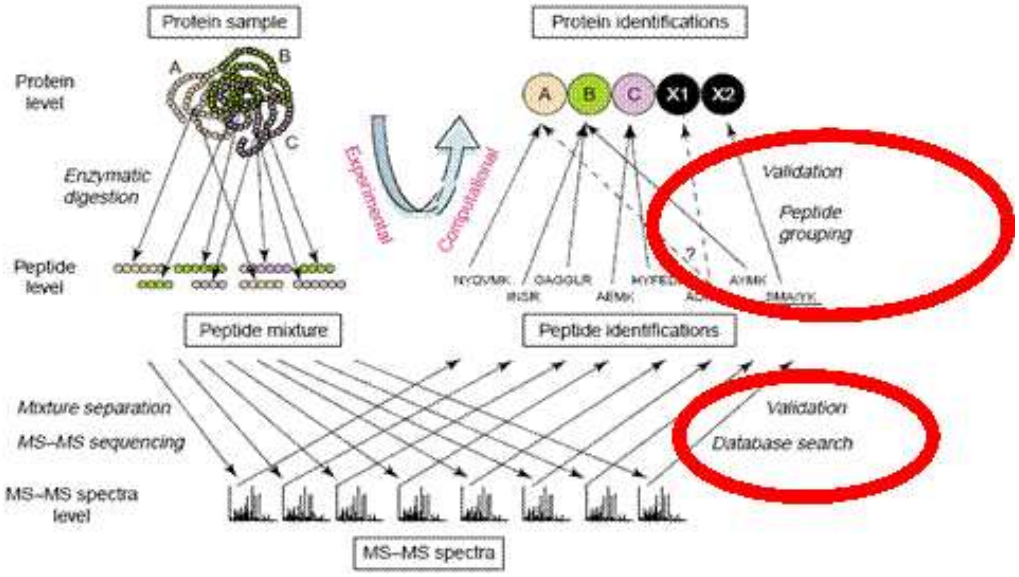


Figure 2.2

Protein identification

All of these algorithms may lead to false positive peptide identifications due to noisy spectra, imperfect matches, or a coincidental similarity in MS/MS fragmentation patterns. The current challenge for high-throughput proteomics is to use database search

results generated by searching large volumes of MS/MS spectra to derive true identifications from the database search results. In small datasets, manual validation by experts can be used to achieve this goal. However, this time-consuming and labor-intensive approach is not practical for high-throughput peptide analysis. The most commonly used methods for distinguishing correct peptide identifications from incorrect peptide identifications are threshold methods [4, 7, 23, 46, 47, 55, 64, 65, 66, 67, 68, 73, 75, 82, 84], target-decoy strategies for false positive rate measurement [22, 38, 46, 73], or statistical models [13, 22, 42, 45, 49, 72].

2.3.1 Threshold Methods

Threshold methods treat peptides that are identified with database search scores above a user defined threshold as correct identifications and those with scores below the threshold as incorrect. Figure 2.3 shows the basic idea of threshold methods for peptide validation. The different database search algorithms use different scoring systems for the quality of peptide assignments to mass spectra. In this example, the MS/MS spectra for a known mixture of 18 proteins [50] has been queried against a database containing the sequences for these proteins using the SEQUEST search algorithm. The values for two quality scores generated by SEQUEST(ΔCn and Xcorr) are shown for all peptide identifications. Xcorr is the cross correlation between the theoretical and observed spectrum and is used to produce the final ranking of the candidate peptides. ΔCn is a measure of the difference of the Xcorr for a peptide assignment compared to the Xcorr of the next best hit. In Figure 2.3 the red points represent scores of assignments known to be correct and those in blue represent

assignments known to be incorrect. For a specific database search algorithm, empirical methods are used to determine cutoffs for a set of scores. Note that when analyzing a proteomics sample, the scientist will not have prior knowledge of which assignments are correct and which are incorrect.

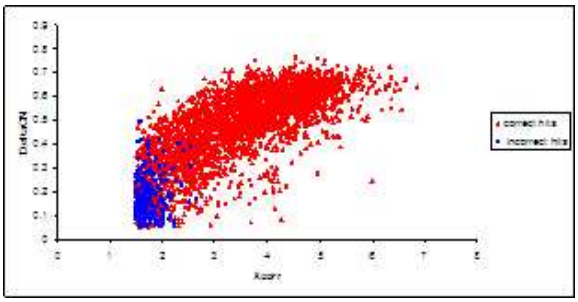


Figure 2.3

Threshold method for peptide validation.

For example, for the widely used SEQUEST search algorithm for peptide identification, Yates *et al.* have published several widely used sets of thresholds for Xcorr and ΔCn [4, 7, 23, 46, 47, 55, 64, 65, 66, 67, 68, 73, 75, 82]. Two of the Yates cutoffs are shown in Figure 2.4. Other thresholds have been determined for other algorithms such as Mascot and X!Tandem [46]. The threshold method has a number of shortcomings due to its dependence on the database search algorithm, database size, sample complexity and peptide charge states. Trade-offs between sensitivity and specificity are not supported by this method and the user cannot choose an error rate (false discovery rate) indicating the level of confidence in the search results [4, 7, 23, 46, 47, 55, 64, 65, 66, 67, 68, 73, 75, 82].

2.3.2 Target-Decoy Strategy

Recently, a target-decoy strategy [22] has been widely adopted as a method for estimating the false discovery rate for peptide identification [13, 67]. The target database contains all possible protein sequences for a given organism. The decoy database contains an equivalent number of nonsense protein sequences that should not be present in the sample. The decoy database can be generated by randomly scrambling or reversing the sequences within the target database or by using a Markov chain derived from the target database [13, 22, 67]. The basic assumption of most methods that use the target-decoy strategy is that the number of peptide assignments made against the decoy database should reflect that of coincidental peptide assignments drawn from the sequences of real proteins [13, 22, 67, 73]. For example, Qian *et al.* [73], conduct separate searches of MS/MS spectra against the target database and decoy database, and then, after applying a threshold x, calculate a false discovery rate as

$$\frac{\#decoyhit > x}{\#targethits > x} \tag{2.1}$$

Figure 2.4 illustrates how this target decoy method can be used to compute the false discovery rate when thresholds are used for peptide validation. Other groups such as Huttlin, Elias and Gygi [22] advocate searching the spectra against a concatenated target and decoy database. In this case, the false discovery rate is computed as

$$\frac{2 \times \#decoyhit > x}{\#decoyhits > x + \#targethits > x} \tag{2.2}$$

Although the controversy of whether to use concatenated or separate searches continues to be debated in the literature [13, 22, 72], it is widely accepted that the target-decoy strategy combined with the threshold method provides a reasonable estimate of the false discovery rate. In addition, the target-decoy strategy is easy to implement and requires no manual analysis by the researcher. The target decoy strategy is also employed by several tools for peptide validation based on machine learning or statistical modeling as described in the next section.

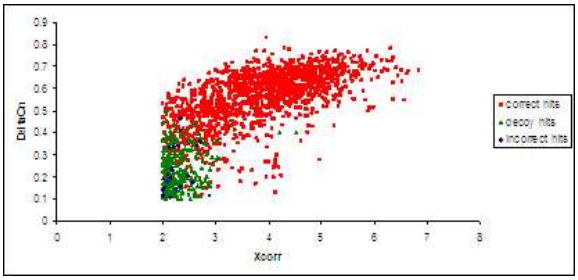


Figure 2.4

Peptides using Yates' low cutoff (ΔCn>0.1 and Xcorr>2.0) and false discovery rate of 254/1483=17.13% computed using target decoy strategy of Qian.

2.3.3 Machine Learning and Statistical Modeling Approaches

Unlike threshold methods, statistical modeling and machine learning methods develop a model of the distribution of incorrect and correct peptide assignments and determine a decision boundary based on this model.

The most widely known statistical method is Peptide Prophet [49]. Peptide Prophet first computes a single linear discriminant score for each peptide based on several different

SEQUEST scores such as Xcorr, and ΔCn as shown in Equation 2.3 where $x_1, x_2, ..., x_S$ are scores from the search algorithm, c_0 is a constant weight, and $c_1, c_2, ..., c_S$ are weights for each search score from the search algorithm.

$$F(x_1, x_2, ..., x_S) = C_0 + \sum_{i=1}^{S} C_i x_i \tag{2.3}$$

The form and parameters of the discriminant function are learned based on a training dataset. For a specific dataset, the discriminant scores are calculated for all peptides and then a histogram is generated for these scores. Peptide Prophet assumes that the discriminate scores for noise follow a Gamma distribution and the discriminant scores for correct identifications follow a Gaussian distribution. An Expectation Maximization (EM) algorithm is used to learn the parameters of the two distributions and Bayesian statistics are used to compute the probability that a match with a given discriminant score is correct. Choi and Nesvizhskii [14] have recently described an extension to Peptide Prophet that uses a decoy database to estimate the parameters of the noise distribution.

The statistical model, Peptide Prophet, uses a linear function to combine some search scores to a single discriminant score and consider all scores simultaneously. The probabilities computed by Peptide Prophet can be used to estimate the likelihood of the presence of peptide. Although Peptide Prophet has been used successfully to develop statistical models for peptide validation, it still has some limitations. Peptide Prophet needs a training dataset to build an accurate discriminant function (learning the coefficients of the discriminant function). Peptide Prophet also assumes that correct hits and incorrect hits follow certain standard distributions. This assumption has not been theoretically proved. And the

parameters of these standard form distributions are learned by an EM algorithm, which is sensitive to a starting point. The improved Peptide Prophet with the target-decoy strategy avoids EM learning, but it still assumes standard forms for the distributions of correct and incorrect hits.

Kunec's product method [52] assigns a product of $Xcorr$ and ΔCn for each peptide, and discriminates incorrect assignments from correct assignments based on the product score for the target and decoy search results, and calculates the FDR as:

$$FDR_{product} = \frac{\#decoyhits > T}{\#totalhits > T}, \tag{2.4}$$

where T is the threshold of product of $Xcorr$ and ΔCn.

Lukas Käll *et al.* describe their tool Percolator [45] that uses a semi-supervised machine learning method based on support-vector machines to discriminate between correct and incorrect peptide assignments. Percolator uses a three phase process. In phase 1, Percolator runs separate searches of MS/MS spectra against target and decoy databases using an algorithm such as SEQUEST. For each spectrum, the top-scoring peptide match (PSM) against each database is stored. For each target and decoy hit, a vector of 20 features is computed including scores generated by SEQUEST plus some additional features. The set of decoy hits are divided into two sets, one half of the hits are used in phase 2 and the remainder in phase 3. Phase 2 is an iterative process where each iteration consists of three steps: 1) selecting a subset of high-confidence target PSMs to serve as a positive training set; 2) training a SVM (Support Vector Machine) to discriminate between the positive and the decoy PSMs; and 3) re-ranking the entire set of PSMs using the trained classifier. After

a fixed number of iterations, a stable SVM is built. In Phase 3, the trained SVM is applied to the entire set of target PSMs and second set of decoy PSMs. The resulting ranked list gives an estimate of the q-value for each target PSM.

Percolator does not assume that correct and incorrect hits are linearly separable; instead, it trains a Support Vector Machine based on decoy hits as negative examples. Although Percolator avoids the assumption of distributions for correct and incorrect hits, the SVM training has limitations. The SVM is trained based on the high-confidence target hits as a positive set and decoy hits as negative hits. Therefore it does not have detailed information about "borderline" positive examples and thus may misclassify some correct hits with relative low-confidence scores.

Zhang *et al.* [87] describe a method that uses a linear discriminant function (LDF) with the concatenated target-decoy strategy to filter SEQUEST database search results. Their linear function is of the form $dCn = k(b - Xcorr)$. An estimated false discovery rate is calculated for each (k, b) pair using Equation 2.2 where k and b are varied by a fixed increment within a range determined from the data. The number of target peptide hits is recorded for each (k, b) pair that yields an FDR close to the desired FDR. The (k, b) pair that yields the largest number of hits is used for the final LDF for peptide validation.

Zhang's method also assumes that correct and incorrect hits are able to be separated linearly but does not assume that they follow certain distributions. The method only takes into account two search scores Xcorr and ΔCn.

Artificial neural networks have also been used for peptide validation. Baczek *et al.* [6] use a set of 13 features and a small training set of known proteins to build a neural network

that classifies a peptide identification as valid or invalid. Jane *et al.* [74] use a target decoy strategy when building their neural network. Identifications from the decoy are used as negative training examples and those from the target are used as positive training examples. After training, all target identification are classified using the neural network.

The advantages and disadvantages of using neural networks for peptide validation with and without a target-decoy strategy are similar to those encountered when using Support Vector Machines. The neural network model trained using the high-confidence hits for positive examples and decoy hits as negative examples may exhibit a bias to when classifying correct hits with low-confidence scores.

2.4 Proteogenomic Mapping for the Structural Annotation of Prokaryotic Genomes

Accurate genome annotation is a critical step in genomics. Most current methods for genome annotation in prokaryotes are based exclusively on features of the genomic sequence. More recently, mass spectrometry data searched against translated genome sequence has been used as a complementary method to provide direct evidence of expression for genome annotation. Here we present a review of research on the use of mass spectral data for structural annotation of genomes.

Jaffe *et al.* [40] at Harvard Medical School first proposed the use of proteomics data to annotate genomes in 2004. Jaffe *et al.* predicted the set of ORFs in the genome of *Mycoplasma pneumoniae* based principally on expressed protein-based evidence. They queried the mass spectra against a database consisting of the genome translated in six reading frames, selected high confidence peptide identifications, mapped these peptides

onto the genome, and extended the peptide hits into Open Reading Frames (ORFs) bound by traditional genetic signals such as start codons, stop codons, etc. The resulting annotation was called a "proteogenomic map" of potential novel protein-coding genes. The ORFs generated by mapping a peptide to its genome were used to confirm predicted ORFs, to detect new ORFs, and to correct boundary errors such as various N-terminal extensions.

Subsequently, a number of other researchers have reported the use of proteogenomic mapping to annotate other prokaryotic genomes [18, 63, 85]. All of these researchers use essentially the same process described by Jaffe *et al.* [40]. Little research has been reported on evaluation of the potential ORFs identified by proteogenomic mapping. There has also been some research extending proteogenomic mapping to eukaryotes where the gene structure is more complicated [15, 77, 81].

CHAPTER 3

PEPTIDE VALIDATION

The most widely used tool for identifying the sets of proteins that are present in a complex biological mixture is mass spectrometry followed by database search algorithms such as SEQUEST, Mascot, or X!Tandem. As introduced in Chapter 1, proteins are first digested into smaller pieces called peptides and identification is actually done at the peptide level. The database search algorithms may lead to false positive peptide identifications due to noisy spectra, imperfect matches, or a coincidental similarity in MS/MS fragmentation patterns.

We have developed a machine learning model based on distance-based outlier detection to estimate the accuracy of peptide assignments to tandem mass (MS/MS) spectra. In the model, the distribution of the quality measures from database search algorithms of incorrect peptide assignments to spectra is estimated by searching the spectra against a decoy (nonsense) database. A distance score for each peptide assignment is computed as the sum of the distances of quality measures of assignments from both the target (real) database and the decoy database to the K-nearest assignments from the decoy database. We then employ Bayes' rule to compute the probability of a peptide assignment being correct. The computed probabilities have allowed us to distinguish correctly and incorrectly assigned peptides with a predictable false identification error rate without requiring the use

of training datasets or expert participation. Using standard protein mix dataset provided by the Institute of System Biology, we are able to identify as many peptides as state-of-the-art computational methods, but with a much more accurate estimate of the false positive rate.

3.1 Background

A major goal of proteomics research is to identify proteins expressed in a cell or tissue. In recent years, tandem mass spectrometry has been used increasingly for high-throughput analysis of protein samples. Proteins in a sample are digested into peptides. Peptides are then ionized and fragmented to produce signature MS/MS spectra that are used for identification. Peptide identifications are made by searching MS/MS spectra against a protein sequence database and finding the best matching database peptide. A variety of algorithms for automated identification of peptides based on matching their masses and fragmentation patterns have been developed, including SEQUEST [23], Mascot [71], and X!Tandem [29]. These algorithms compare an observed MS/MS fragmentation pattern from an unknown peptide with those fragmentation patterns predicted for all peptides of equivalent mass within a given protein database and return the peptide sequence with a predicted fragmentation pattern that best matches the observed spectrum. Each returned peptide sequence is assigned a set of scores that reflect various aspects of the fit between the observed spectrum and the theoretical spectrum. All of these algorithms may lead to false positive peptide identifications due to noisy spectra, imperfect matches, or a coincidental similarity in MS/MS fragmentation patterns.

The current challenge for high-throughput proteomics is to use database search results generated by searching large volumes of MS/MS spectra to derive true identifications from the database search results. In small datasets, manual validation by experts can be used to achieve this goal. However, this time-consuming and labor-intensive approach is not practical for high-throughput peptide analysis. The most commonly used methods for distinguishing correct peptide identifications from incorrect peptide identifications are threshold methods [84] (add another one), target-decoy strategies for false positive rate measurement [22, 38, 46, 73] or statistical models such as Peptide Prophet [49] [88]. Threshold methods treat peptides that are identified with database search scores above a user defined threshold as correct identifications and those with scores below the threshold as incorrect. The threshold method has a number of shortcomings due to its dependence on database search algorithm, database size, sample complexity and peptide charge states. Trade-offs between sensitivity and specificity are not supported by this method and the user cannot choose an error rate (false positive rate) indicating the level of confidence in the search results. More recently, a target-decoy strategy has been used to estimate false positive rate by several research groups [22, 38, 73]. For example, Qian *et al.* [73], conduct separate searches of MS/MS spectra against the target database and decoy database, and then, after applying a threshold T, calculate a false positive rate as

$$FDR = \frac{\#decoyhits > T}{\#targethits > T}. \tag{3.1}$$

Elias *et al.* [22] argue that the search should be performed on a concatenated target-decoy database with the FDR calculated as:

$$FDR_{Elias} = \frac{2 \times \#decoyhits > T}{\#decoyhits > T + \#targethits > T} \tag{3.2}$$

where T is a threshold such as $Xcorr$. The controversy of concatenated versus separate searches continues to be debated in the literature (cite someone). We demonstrate (see Section 3.3) that decoy hits provide a better estimate of the quality scores of target incorrect hits with concatenated search.

Unlike threshold methods, statistical methods such as Peptide Prophet [49], develop a model of the distribution of incorrect and correct peptide assignments. Peptide Prophet first computes a single linear discriminant score for each peptide based on several different SEQUEST scores such as $Xcorr$, and ΔCn. The form and parameters of the discriminant function are learned based on a training dataset. For a specific dataset, the discriminant scores are calculated for all peptides and then a histogram is generated for these scores. Peptide Prophet assumes that the discriminate scores for noise follow a Gamma distribution and the discriminant scores for correct identifications follow a Gaussian distribution. An Expectation Maximization (EM) algorithm is used to learn the the parameters of the two distributions and Bayesian statistics are used to compute the probability that a match with a given discriminant score is correct. Choi and Nesvizhskii [13] have recently described an extension to Peptide Prophet that uses a decoy database to estimate the parameters of the distribution of noise. Zhang *et al.* [88] adopt the discriminant score from PeptideProphet, and demonstrate that the distribution of discriminant score of decoy hits reflects that of target incorrect hits. They use a non-parametric Bayesian model for pep-

tide validation that assigns each peptide a probability as true identification based on the discriminant score of each hit shown as:

$$P(+|F) = \frac{P(F|+)P(+)}{P(F|+)P(+) + P(F|-)P(-)}, \tag{3.3}$$

where F is a discriminant score calculated from a linear discriminant function. FDR of statistical model is defined as:

$$FDR_{Statisticalmodel} = 1 - P(+|F). \tag{3.4}$$

Kunec's product method [52] assigns a product of $Xcorr$ and ΔCn for each peptide, and discriminates incorrect assignments from correct assignments based on the product score for the target and decoy search results, and calculates the FDR as:

$$FDR_{product} = \frac{\#decoyhits > T}{\#totalhits > T}, \tag{3.5}$$

where T is the threshold of product of $Xcorr$ and ΔCn.

Lukas Käll *et al.* [45] describe their tool Percolator that uses a semi-supervised machine learning method based on support-vector machines to discriminate between correct and incorrect peptide assignments. Percolator runs separate searches against a target and a decoy database. The top-scoring target hits serve as positive samples, and decoy hits as negative samples. A SVM is trained iteratively on the training dataset. The FDR is calculated as

$$FDR_{percolator} = \pi_0 \frac{\#decoyhits > T / \#decoyhits}{\#targethits > T / \#targethits}, \tag{3.6}$$

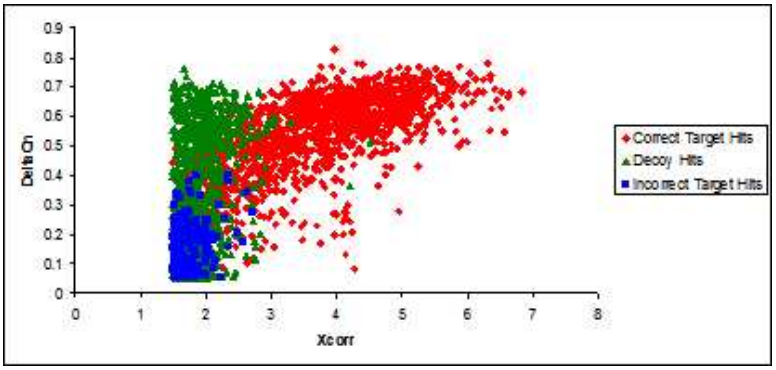
where π_0 is the estimated proportion of target hits that are incorrect, that is $P(-)$, and T is the Percolator score threshold. Percolator avoids the assumption of distributions for correct hits and incorrect hits.

In this chapter we present PepOut, a robust and efficient algorithm for the validation of peptide identifications made by MS/MS and database search. PepOut combines the target-decoy strategy, distance-based outlier detection, and Bayesian statistics to distinguish correct peptide identifications from incorrect peptide identifications. Fig. 3.1 shows a typical distribution of scores of peptide hits against a decoy database and a target database. The assumption of the target decoy strategy is that the distribution of scores of hits against the decoy database (green points in Fig. 3.1) can be used to estimate the distribution of scores of incorrect identifications. Hits against the target database (red points in Fig. 3.1) are a mixture of correct and incorrect identifications. The assumption of our approach is that identifications against the target database with scores that are distant from scores of identifications against the decoy database (outliers with respect to the decoy database) are more likely to be correct identifications. Thus distance-based K-nearest neighbor outlier detection in score space can be used to separate correct and incorrect identifications. We compute a distance score for each peptide assignment which is the sum of the distances of database search scores of assignments from both the target database and the decoy database to the K nearest assignments from the decoy database. A peptide identification from the target search results with larger distance score is more likely to be a true iden-

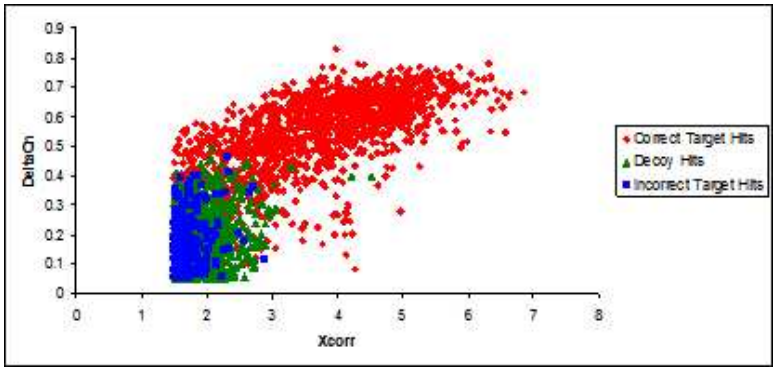
tification since it is further away from decoy hits. The only assumption of our outlier method is that none of the decoy hits are correct identifications and this assumption is supported by previous research by a number of different groups [22, 38, 46, 73]. Using this assumption, the score distribution of decoy search results enables us to estimate the score distribution of false identifications searched against the target database and to estimate the the prior probability of false identifications in the target search results. Bayes' Rule is employed to calculate the probability score for each peptide with a given distance score. Other statistical approaches such as Peptide Prophet [49] require a training dataset to build the model and assume that the distributions of correct and incorrect peptide identification follow standard distributions. PepOut uses a semi-supervised approach and so does not require a training set. In addition, PepOut implements a nonparametric density estimation technique to model the distributions of correct and incorrect assignments and thus makes no assumptions about the form of these distributions. PepOut accepts standard SEQUEST output and can easily be incorporated into any proteomics computational pipeline. Finally, our method provides a probability score for each peptide and allows the user to specify a false discovery rate for the entire dataset.

3.2 Experimental Section

Development of computational methods for proteomic data analysis is facilitated by the availability of high quality benchmark datasets. We used 'ISB standard protein mix' database that contains spectra generated from ten replicate analysis of a defined mixture of 18 proteins by the LCQ DECA XP instrument [50]. The spectra in this database were



(a) Separate search



(b) Concatenated search

Figure 3.1

Peptide assignments identified by searching spectra of ISB Mixture 1 against target and decoy databases for charge 2+. (a). Search conducted separately against the target and decoy databases. (b). Search conducted against database produced by concatenation of target and decoy database.

quality checked and searched against a *Haemophilus influenzae* database appended with 18 standard proteins and known contaminants using the SEQUEST algorithm. Confidence in peptide identifications estimated using Peptide Prophet have been previously reported [49]. Using the raw data provided for Mixtures 1, 2, and 3, we repeated SEQUEST searches exactly as described and processed the search results with PepOut.

3.3 Methods

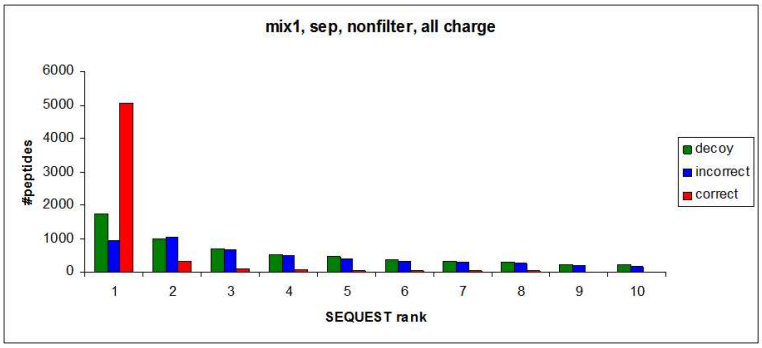
3.3.1 Motivation for using a decoy database

The target-decoy strategy is an empirical approach for estimating the false discovery rate (FDR) within a given dataset [22, 46, 67]. The target database contains all possible protein and peptide sequences for a given organism. The decoy database contains an equivalent number of nonsense protein and peptide sequences that should not be present in the sample. The decoy database can be generated by randomly scrambling or reversing the sequences within the target database or by using a Markov chain derived from the target database [22]. The major assumptions of the target-decoy strategy are that 1) all decoy hits are incorrect, and 2) the characteristics of decoy hits reflect those of target incorrect hits. There is a debate on how to use target-decoy strategy correctly: 1) should mass spectra be searched against a target and decoy database separately (separate search) or against a concatenated target-decoy database (concatenated search)? 2) does the number of decoy hits reflect the number of target incorrect hits (number characteristics) or does

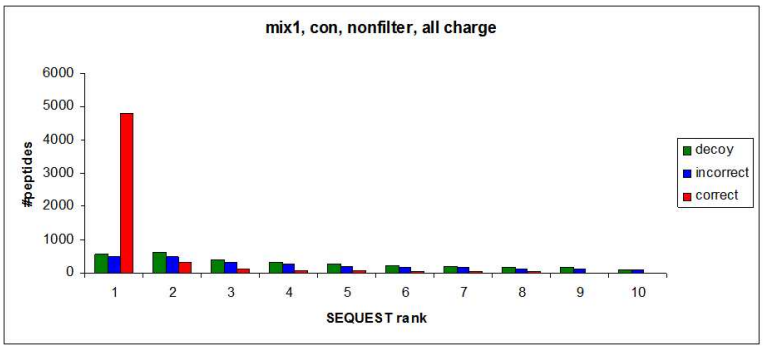
the database score distribution of decoy hits reflect score distribution of target incorrect hits (distribution characteristics)?

Elias *et al.* [22] have clarified the target-decoy strategy methodology based on observed decoy hit frequencies and demonstrated that the number of decoy hits are equally likely to that of target incorrect hits on a concatenated search results. Käll *et al.* [45] proposed a support vector machine classifier, Percolator, which uses distribution characteristics on separate search results. Qian *et al.* [73] estimated a false discovery rate using the number characteristics on separate search results. Keller *et al.* [49] proposed a statistic method, PeptideProphet, which combines database search scores linearly to a single discriminant score, and assume the discriminant score distribution of incorrect hits follow standard Gamma distribution. Choi *et al.* [13] improved PeptideProphet in the way that assumes that discriminant score distribution of decoy hits reflect that of target incorrect hits, also learn parameters of Gamma from decoy hits. Zhang *et al.* [88] adopt discriminant score from PeptideProphet, and believe that discriminant score distribution of decoy hits reflect that of target incorrect hits, then use a non-parametric Bayesian model for peptide validation. Choia's and Zhang's methods used distribution characteristics on concatenate search results. Kunec *et al.* [52] used a product of scores to discriminate correct hits from incorrect hits on a separate search results and estimates FDR by number characteristics.

Here we will demonstrate that 1) the score distribution of decoy hits reflect the characteristics of the score distribution of target incorrect hits but the number of decoy hits is not accurate estimate of the number of target incorrect hits, and 2) concatenated search is preferable to separate search when using the SEQUEST search algorithm. Table 3.1



(a) Separate search



(b) Concatenated search

Figure 3.2

Comparison of the number of decoy and target incorrect hits for separate search (a) and concatenated search (b).

demonstrates that for each mixture, the number of decoy hits is much larger than the number of target incorrect hits when the target and decoy database are searched separately. Table 3.1 also shows that when the search is performed on a concatenated target-decoy database, the number of decoy hits is closer to the number of target incorrect hits but it is still an over estimate. Fig. 3.2 shows the number of decoy, target incorrect and target correct hits grouped by rank. These results illustrate that for mix1, both separate and concatenated searches result in more decoy hits than target incorrect hits and the problem is much worse for separate searches. For database search algorithms such as SEQUEST, MS/MS spectra are compared to theoretical spectra generated from sequences of the target database. Sequences in the target database compete for the top-ranked score (Rsp score in SEQUEST) in a single search. If the search is performed separately, no target correct sequences compete for the top score with the sequences in the decoy database so that the rank scores of decoy hits are higher than the rank scores of target incorrect hits. In addition, the number of peptides identified from the decoy database with certain criteria (score threshold) is larger than the number of target incorrect peptides identified from the target database. If MS/MS spectra are searched against a concatenated target-decoy database, all target and decoy sequences compete for the top-ranked score and target incorrect and decoy hits will have the same score distribution. Although the score distributions of decoy and target incorrect hits are similar when using a concatenated search, the total number of decoy hits is still larger than the total number of target incorrect hits and therefore the number of decoy hits does not provide an accurate estimate of the number of target incorrect hits as shown in Fig. 3.2(b) and Table 3.1.

Table 3.1

Number of decoy hits vs. number of target incorrect hits

Searched against decoy and target database separately								
	Charge +1		Charge +2		Charge +3		Total	
	decoy	incorrect	decoy	incorrect	decoy	incorrect	decoy	incorrect
M1	4562	3784	3985	3007	1532	1273	10079	8064
M2	7627	6319	9614	7700	3334	2821	20575	16840
M3	3274	2733	4431	3701	2403	1882	10108	8316
Searched against a concatenated target-decoy database								
	Charge +1		Charge +2		Charge +3		Total	
	decoy	incorrect	decoy	incorrect	decoy	incorrect	decoy	incorrect
M1	2390	2055	2403	2015	1187	993	5980	5063
M2	3777	3413	6361	5269	2628	2244	12766	10926
M3	1743	1519	2869	2711	1842	1485	6454	5715

Fig. 3.1 shows that, when $Xcorr$ and ΔCn are considered simultaneously, hits against the decoy database (green dots) should provide an estimate of the distribution of incorrect hits(blue dots). The results in Fig. 3.1 demonstrate that the search against a concatenated target-decoy database results in similar distributions of decoy hits and incorrect hits, while separate searches results in the different distributions. Therefore we adopt a concatenated target-decoy search strategy.

Search algorithms such as SEQUEST assign a set of scores to a peptide identification based on the match quality. For SEQUEST, the scores generally used to validate peptide identifications are: $Xcorr$, ΔCn, Sp and Rsp. These scores will be discussed in detail later in section score transformation. The database search score distribution of decoy hits should reflect that of coincidental target hits. Fig. 3.3 shows box-and-whisker plots of these distribution of $Xcorr$ (a) and ΔCn (b) and Rsp (c) for ISB mix1. It is easy to see

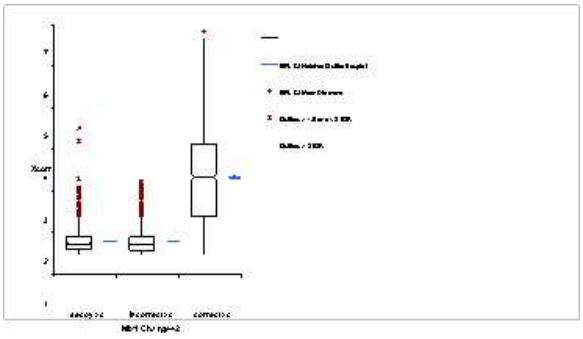
that the score distributions of decoy and target incorrect hits are very similar and different from the score distribution of target correct hits. Therefore, although the number of decoy hits is not an accurate reflection of the number of target incorrect hits, the score distribution of decoy hits provides an accurate model of target incorrect hits.

3.3.2 Motivation for using outlier detection for peptide validation

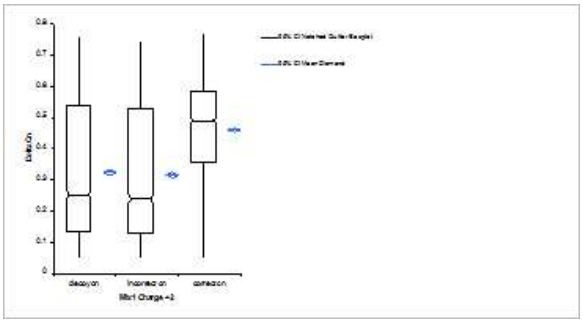
As Fig. 3.1 illustrates, correct peptide assignments can be viewed as outliers from incorrect peptide assignments. Outliers are observations which are far away from the rest of the data and may be indicative of data points that belong to a different population than the rest of the data. Outlier detection has been successfully used in a number of application areas for identifying data points from different populations including credit card fraud, calling card fraud, network intrusion detection and insurance fraud [5]. Because the peptide assignments made against the decoy database provide a mechanism for modeling the distribution of incorrect target hits, we can view correct target hits as outliers with respect to incorrect target hits as modeled by decoy hits. The distance-based outlier detection method we have used, K-nearest-neighbor, distinguishes an object as an outlier on the basis of the distance to the K-nearest points in the normal population. In our case, the normal distribution consists of incorrect peptide assignments.

3.3.3 Distance-based outlier detection for peptide validation

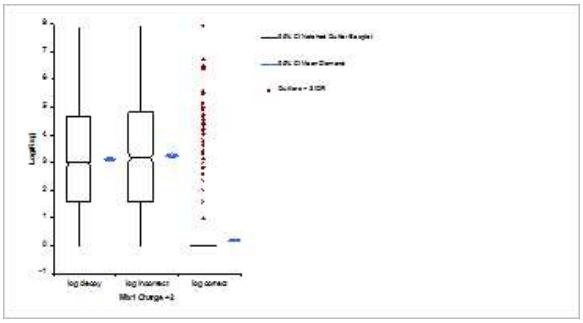
In our method, incorrect peptide assignments are viewed as noise and correct peptide assignments are viewed as outliers from the noise. The score distribution of noisy peptide assignments is estimated by the score distribution of decoy hits. We use Euclidean distance



(a) Xcorr



(b) ΔCn



(c) Rsp

Figure 3.3

Boxplot of Xcorr (a), ΔCn (b) and Rsp (c) for ISB mix

to measure the distance between peptide hits in score space. The Euclidean distance $d(p, q)$ between two points $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$, in Euclidean n-space, is defined as

$$d(p, q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{3.7}$$

In our method, data point p is a peptide assignment from the target hits, data point q is a peptide assignment from the decoy hits, and $p_1, p_2, \ldots, p_n$ and $q_1, q_2, \ldots, q_n$ are database search scores assigned by the database search algorithm, such as the $Xcorr$, ΔCn, and Rsp scores of SEQUEST, Ion Score and Homology Factor of MASCOT, and In Dot and ΔDot of X!TANDEM. A peptide is detected as an outlier from the noise based on its distance to its K-nearest-neighbors in the noise (decoy hits in our case). The Distance score $D(p)$ represents the sum of the distances of a target hit p to its K nearest neighbors in the decoy hits and is used to rank the peptide assignments as outliers [5]. The weight of a peptide assignment is defined as:

$$D(p) = \sum_{k=1}^{K} d(p, q_k) = \sum_{k=1}^{K} \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{3.8}$$

where q_k is the kth nearest neighbor of p. In our work, we have used a value of 5 for K. This value was chosen based on empirical comparisons of the performance with different K values. In our experiments, we are using the SEQUEST search algorithm. Some peptide assignments in the target hits have a distance score of $D = 0$ because these hits have exactly the same score values as K decoy hits and therefore the distance to their K nearest decoy neighbors is 0. These 0-weight peptide assignments are discarded from further

consideration because they clearly are not distinguishable from noise. Each distance D is converted to a log distance for further analysis:

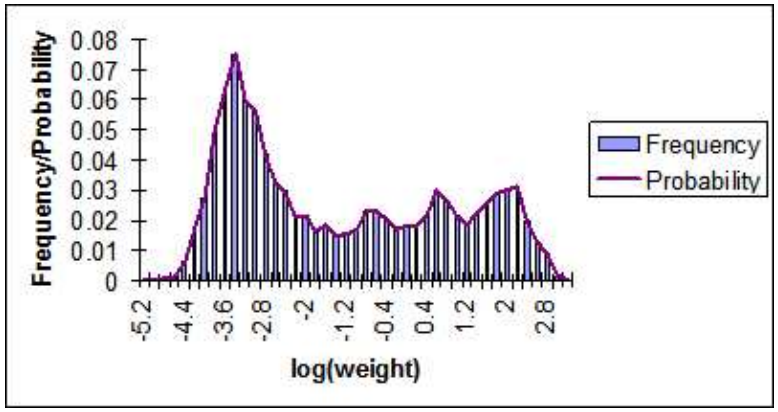
$$s = log(D) \tag{3.9}$$

The probability distributions of the log distance score s of target hits, $P(s)$ and of decoy hits, $P(s|-)$, are estimated using a standard nonparametric density estimation technique based on construction of a histogram with specified bin sizes. Figure 3.4 illustrates from assignment frequencies using a bin size of 0.2 for a bacterial proteomics dataset (see Section 3.3) for charge state +2. The advantage of using the histogram approach is that we need not assume any parametric families for these distributions.

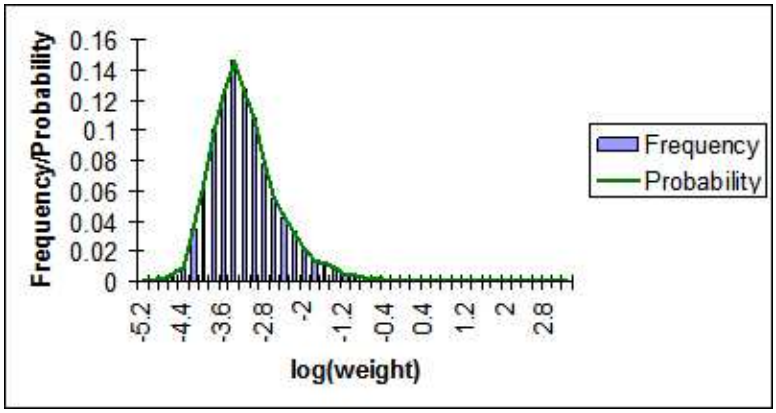
Figure 3.5 shows the distribution of distance score s for known correct, and incorrect target hits, and of the decoy hits for ISB Standard Mixture 1. This result demonstrates that the distribution of log distance score for decoy hits provides an accurate estimate of the distribution of log distance score of incorrect target hits.

Based on the distribution of log distance scores of target hits and decoy hits, Bayes' rule is applied to calculate the probability that a peptide assignment is correct given a specific log distance score value s. We denote correct and incorrect peptide assignments as + and -, respectively. The probability that a peptide with a log weight score s is correct can be computed as:

$$P(+|s) = \frac{P(s|+)P(+)}{P(s|+)P(+) + P(s|-)P(-)} = \frac{P(s) - P(s|-)P(-)}{P(s)} = 1 - \frac{P(s|-)P(-)}{P(s)}. \tag{3.10}$$



(a) Histogram of distance scores of matches against the target database



(b) Histogram of distance scores of matches against the decoy database

Figure 3.4

Histogram and derived distribution for target hits P(s) and decoy hits $P(s|-)$ for charge state +2 for the *M. haemolytica* Dataset.

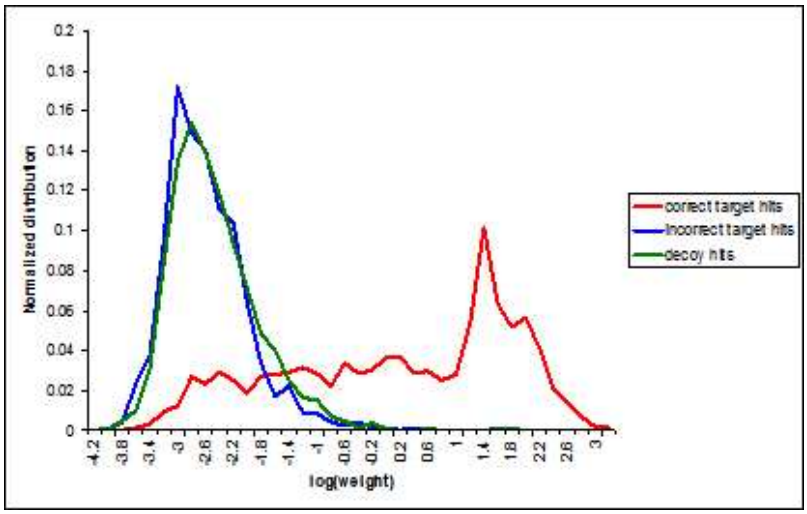


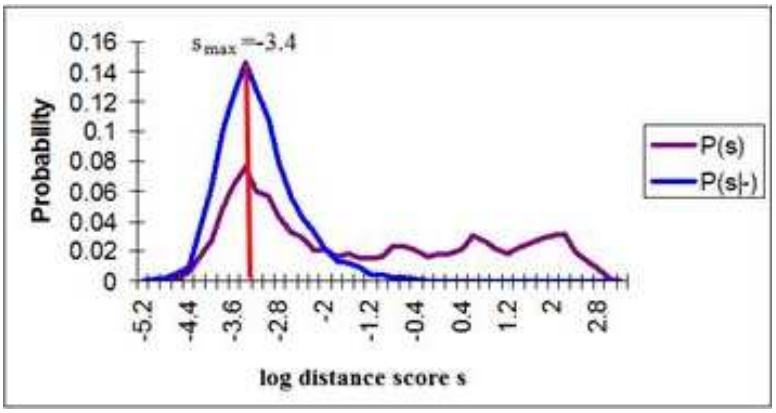
Figure 3.5

Distributions of log distance score s for correct target hits, incorrect target hits and decoy hits using ISB standard Mixture 1.

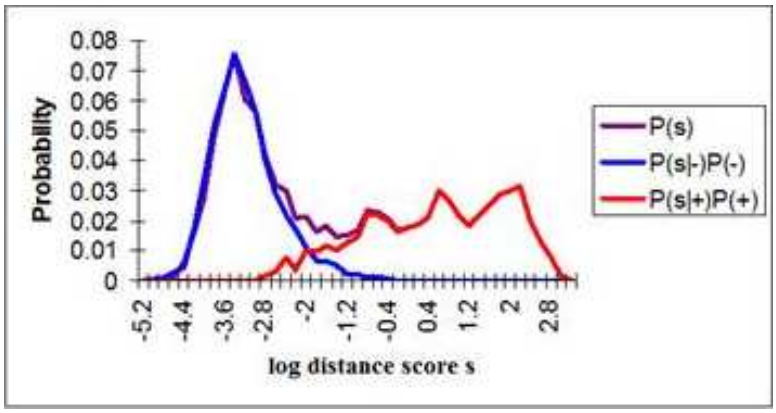
The probability of an incorrect peptide assignment $P(s|-)$ is estimated by the probability of decoy peptide assignments as shown by the blue line in Figure 3.5.

We know that the distribution of target hits $P(s)$ is a mixture of incorrect peptide assignments $P(s|-)$ and correct peptide assignments. The basic assumption of the target-decoy strategy is that the decoy hits can be used for modeling the distribution of incorrect hits against the target database as illustrated in Fig. 3.4(a). The distribution of log distance score s of decoy hits can also be used to model the distribution of log distance score s of incorrect hits against the target database.

We use the following iterative procedure to estimate $P(-)$, the prior probabilities of incorrect peptide assignments and correct peptide assignments. First, the ratio of the frequency of target hits over the frequency of decoy hits is computed for all weights less than the maximum frequency weight (s_{max}) in the decoy hits shown in Fig. 3.4(a).



(a) Actual P(s) and P($s|-$) estimated from decoy hits



(b) Derived P($s|+$) and P($s|-$)

Figure 3.6

The process of estimating P(-) and P($s|+$).

$$r = \frac{P(s = s_{max})}{P(s = s_{max}|-)} \tag{3.11}$$

We assume that $P(s|+)$ does not contribute to $P(s)$ for values less than the maximum. Based on the initial estimate of $P(-)$, an iterative procedure is used to choose a value for $P(-)$ by minimizing the absolute error between $P(s) \times r$ and $P(s|-)$ for all weights less than s_{max}.

$$P(-) = \min_{r} \sum_{s \leq s_{max}} |P(s|-) \times r - P(s)| \tag{3.12}$$

The probability of a peptide assignment with log distance score s as a correct identification can now be calculated from Eq. 3.12. This probability can be used as additional information for protein identification [13, 14].

3.3.4 SEQUEST database score preprocessing

We apply our distance-based outlier detection method to SEQUEST database search scores $Xcorr$, ΔCn, RSp. 1)Cross Correlation ($Xcorr$) is a measure based on the number of peaks of common mass between observed and expected spectra, and thus tends to be larger for long peptide assignments. Short correct peptide assignments with relatively small $Xcorr$ scores are hard to distinguish from incorrect long peptide assignments with relatively large $Xcorr$ scores. To overcome this dependence between $Xcorr$ and the length of peptide assignment, we transform $Xcorr$ to $Xcorr'$ by equation:

$$Xcorr' = \begin{cases} \frac{ln(Xcorr)}{ln(N_L)}, & if L < L_C \\ \frac{ln(Xcorr)}{ln(N_C)}, & if L \geq L_C \end{cases} \tag{3.13}$$

where, L is the length of peptide assignment, L_C is 15 for charge +2 and 25 for charge +3. N_L is 2L for charge +2 and 4L for charge +3. N_C is 2*15 for charge +2 and 4*25 for charge +3. This equation is adapted from Peptide Prophet [49]. Transformations of $Xcorr$ score to $Xcorr'$ reduce the dependence between $Xcorr$ and the length of peptide and can significantly improve the discrimination power. In our distance-based outlier detection method, the distance in score space for each peptide assignment reflects how far the peptide assignment is from the nearest K decoy peptide assignments. Distance measurements taken on large value attributes will generally outweigh distance measurements taken on those with small values. It is well known that normalization will improve the accuracy and efficiency of mining algorithms involving distance measurements [24]. There are three major classes of normalization methods: min-max normalization, z-score normalization and decimal scaling normalization. Since we are trying to detect the outliers from the mixture dataset, z-score normalization is not an appropriate selection because it reduces the effect of outliers that dominate the minimum and maximum. Decimal scaling normalization moves the decimal point of values of an attribute. The number of decimal points moved depends on the maximum absolute value of the attribute.

Min-max normalization performs a linear transformation on the original data and preserves the relationships among the original data values and is the method we are using.

The min-max normalization method is applied on $Xcorr'$ to make $Xcorr'$ have the same range as ΔCn as:

$$n\_Xcorr' = \frac{Xcorr' - Min\_Xcorr'}{Max\_Xcorr' - Min\_Xcorr'}, \tag{3.14}$$

where, Min_Xcorr' and Max_Xcorr' are the minimum and maximum value of $Xcorr'$ in dataset.

2) ΔCn is the change in $Xcorr$ values between the first and second best hits. Because ΔCn is already in the range of $[0, 1)$, there is no need to normalize ΔCn value.

3) Sp is a preliminary score which is the score that SEQUEST uses to do an initial scoring of the all peptide candidates. It is more efficient to compute than $Xcorr$, but is considered less robust. Rank Sp is derived by sorting Sp in descending order and a rank is assigned to each peptide sequence (e.g. the topmost entry would have a $Rsp = 1$). Rank Sp is transformed by taking log on it to reduce the data spread as shown in equation:

$$Rsp' = log(Rsp). \tag{3.15}$$

As illustrated in Fig. 3.1, most correct peptide identifications have an Rsp value of one. The log Rsp value is normalized Rsp' in range of 0 to 0.1:

$$n\_Rsp' = \frac{Rsp'}{Max\_Rsp' - Min\_Rsp'} \times 0.1, \tag{3.16}$$

where, Min_Rsp' and Max_Rsp' are the minimum and maximum value of Rsp' in dataset. The maximum value of Rsp' is adopted from previous works of PeptideProphet [49].

Here, we only consider the SEQUEST scores $Xcorr$. ΔCn and Rsp in our distance calculation. However, our approach can easily be adapted to include other SEQUEST scores or to use scores from other search algorithms such as Mascot and X!Tandem.

3.3.5 Results of PepOut

PepOut discriminates correct peptide assignment from incorrect ones by a distance-based outlier detection method. As discussed previously, PepOut assigns a distance score from a peptide to its K-nearest-neighbor decoy hits, and the farther the distance, the higher the likelihood the assignment is correct. PepOut does not assume that incorrect and correct hits are linearly separable as shown in Fig. 3.2(b). Correct peptide identifications are detected as outliers of incorrect hits as modeled by decoy hits.

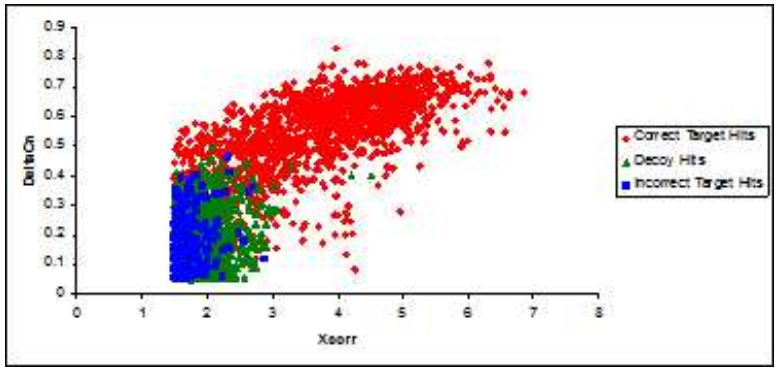
3.4 Results and comparison

Several major classes of peptide validation methods have been reported in the literature including threshold methods coupled with the target-decoy strategy [22], statistical models [49], semi-supervised statistical methods such as PeptideProphet coupled with target-decoy strategy [14], nonparametric discriminant score method [88], Kunec's product method, and a support vector machine classification model called Percolator [45]. We will compare the performance of PepOut to these methods using several criteria.

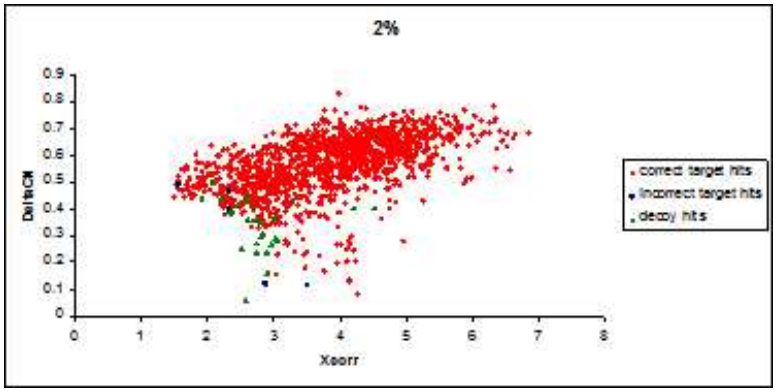
First, the ISB standard 18 protein MS/MS spectra [51] were searched by SEQUEST algorithm against a concatenated target-decoy database, where, the decoy database was generated using a 0th order Markov Chain of the target database resulting in a decoy database of the same size as the target database. The SEQUEST results are filtered by the following three criteria:

a. $Rsp = 1$

b. $\Delta Cn \geq 0.1$



(a) All target and decoy hits of charge 2+



(b) FDR = 2%

Figure 3.7

Results of outlier detection program for ISB standard Mix 1 for charge +2. (a) all target hits with correct in red and incorrect in blue. (b) outlier results with FDR cutoffs of 2%. Red points are correct hits, blue points are incorrect hits, and green points are decoy hits used to estimate the distribution of incorrect hits.

c. $\begin{cases} Xcorr \geq 1.0 & Charge + 1 \\ Xcorr \geq 1.5 & Charge + 2 \\ Xcorr \geq 2.0 & Charge + 3 \end{cases}$

We will compare PepOut to other peptide validation methods using several different measurements. Error rate gives the fraction of target hits that are correct for a specified false discovery rate (FDR):

$$error = 1 - precision = 1 - \frac{\#target\_correct\_hits > T}{\#total\_target\_hits > T} \tag{3.17}$$

$Recall$ gives as fraction of correct hits that are found for a specified FDR:

$$recall = \frac{\#target\_correct\_hits > T}{\#total\_target\_hits}, \tag{3.18}$$

where, T is threshold such as FDR, e.g. $T \leq 2\%$.

We also use five different statistical measures of the closeness of two distributions (P) and (Q):

1) Non-commutative Kullback-Leibler divergence:

$$D_{KL}(P, Q) = \sum_i P(i) log \frac{P(i)}{Q(i)} \tag{3.19}$$

2) Jeffreys divergence:

$$D_{JD}(P, Q) = \frac{1}{2}(\sum_i P(i) log \frac{P(i)}{Q(i)} + \sum_i P(i) log \frac{Q(i)}{P(i)}) = \frac{1}{2}(D_{KL}(P, Q) + D_{KL}(Q, P)). \tag{3.20}$$

3) Jensen-Shannon divergence:

$$D_{JS}(P,Q) = \frac{1}{2}(\sum_i P(i) log \frac{P(i)}{M(i)} + \sum_i P(i) log \frac{Q(i)}{M(i)}) = \frac{1}{2}(D_{KL}(P,M) + D_{KL}(Q,M)), \tag{3.21}$$

where $M = \frac{1}{2}(P+Q)$.

4) λ divergence:

$$D_\lambda(P,Q) = \lambda D_{KL}(P, \lambda P + (1-\lambda)Q) + (1-\lambda) D_{KL}(Q, \lambda P + (1-\lambda)Q), \tag{3.22}$$

where $\lambda \in [0,1]$, if $\lambda = 0.5$, then λ divergence becomes the Jensen-Shannon divergence. In the numerical results, we are testing the case $\lambda = 0.3$.

5) Generalized Kullback-Leibler divergence:

$$D_{GKL}(P,Q) = \sum_i P(i) log \frac{P(i)}{Q(i)} - \sum_i P(i) + \sum_i Q(i) \tag{3.23}$$

3.4.1 PepOut vs. Threshold methods with target-decoy strategy

Elias *et al.* [22] contend to estimate the false discovery rate correctly, the MS/MS spectra should be searched against a concatenated target-decoy database and the search results filtered with $Rsp = 1$. Their false discovery rate is calculated as Eq. 3.2 for each charge state respectively. Elias' threshold method only uses $Xcorr$. We extended this method by transforming $Xcorr$ to $Xcorr'$ as discussed in Section 3.3. This transformation attempts to eliminate the dependence between $Xcorr$ and the length of peptide sequence for charge state +2 and +3 respectively. Table 3.2 compares results of Elias' method to PepOut in terms of $recall$ and $errorrate$. PepOut consistently outperforms the threshold method us-

ing either $Xcorr$ or $Xcorr'$ in terms of $recall$. The $Xcorr'$ threshold method gives better results than $Xcorr$. In addition, the threshold methods overestimate the $errorrate$.

The threshold method with the target-decoy strategy only considers $Xcorr$ and ignores other useful scores such as ΔCn. This method estimates the FDR by assuming that the number of decoy hits reflects the number of target incorrect hits. We have already demonstrated that this assumption is not valid in Section 3.3.1.

3.4.2 PepOut vs. Statistical models

PeptideProphet [49] and its descendants (semi-supervised statistical model [14] and nonparametric statistics model [88] all assign each peptide a discriminant score which is a linear combination of database search scores:

$$F(Xcorr', \Delta Cn, ln(Rsp), d_M) = C_0 + C_1 Xcorr' + C_2 \Delta Cn + C_3 ln(Rsp),$$

where C_0,..,C_3 are coefficients learned from training datasets for each charge state respectively. We compare the similarity of the distribution of the discriminant scores calculated in three different ways and of our distance score to the true distribution of incorrect hits. The original PeptideProphet assumes that the discriminant score of incorrect hits follow a standard gamma distribution and learns the parameters of the gamma distribution using an EM (expectation maximization) algorithm. The newer semi-supervised PeptideProphet attempts to overcome the problem of local maximum encountered by EM by learning parameters of the gamma distribution from the discriminant scores of decoy hits. However this semi-supervised version of PeptideProphet still assumes that the gamma distribution. A more recent nonparametric statistical modification uses a discrete distribution of dis-

Table 3.2

Comparison of PepOut and Elias' threshold methods for Mix1

#total decoy hits		5980		
#total target hits		10682		
#total target correct hits		4806		
		PepOut	XCorr threshold	Xcorr' threshold
Expected ER 0%	#totaltargethits	1171	1350	1077
	#Correct	1171	1347	1077
	Error rate	0	0.3%	0
	Recall	24.5%	28.8%	22.4%
Expected ER 1%	#totaltargethits	3644	3358	3432
	#Correct	3615	3343	3418
	Error rate	0.7%	0.4%	0.4%
	Recall	75.2%	69%	71%
Expected ER 2%	#totaltargethits	4219	3743	3664
	#Correct	4141	3712	3643
	Error rate	1.8%	0.8%	0.6%
	Recall	86.2%	77.2%	76%
Expected ER 5%	#totaltargethits	4713	4220	4298
	#Correct	4511	4126	4220
	Error rate	4.3%	2.2%	1.8%
	Recall	94%	86%	88%
Expected ER 10%	#totaltargethits	5120	4651	4760
	#Correct	4726	4465	4561
	Error rate	7.6%	4.0%	4.2%
	Recall	98.3%	93%	95%

criminant scores and does not assume the form of the distribution for discriminant scores of incorrect hits.

In Fig. 3.8, PepOut is compared to these statistics models in terms of 1) the rationale behind the score used and 2) the closeness of the distribution to the true distribution.

PepOut does not assume that target incorrect and correct hits are linearly separable, but uses a distance score to indicate how far a target peptide is from its K-nearest-neighbor decoy hits. The distance score is calculated directly from the dataset and does not require a training dataset while the discriminant approach does.

To compare the similarity of the distributions to the true distribution, we calculate the distance between two distributions using Eqs. 3.14-3.17, in which the smaller distance values indicate more similar distributions. Table 3.3 and Fig. 3.8 clearly demonstrates that the distribution of distance scores of decoy hits used by PepOut are better models of the distribution of target incorrect hits than distributions based on a discriminant score.

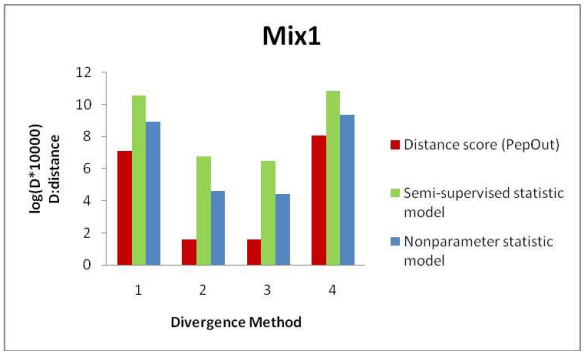
3.4.3 PepOut vs. Products method

The product method performs a separate search and assigns each peptide a score that is the product of $Xcorr$ and ΔCn. It then calculates the FDR as Eq. 3.5, where, T is a product score threshold. As illustrated in Fig. 3.9, the product method distinguishes incorrect and correct using a curve as shown in Fig. 3.9(c). PepOut and products method are compared in terms of error and recall in Table 3.4. These results show that PepOut identifies substantially more correct peptides and gives an expected error rate closer to the true error rate than the product method.

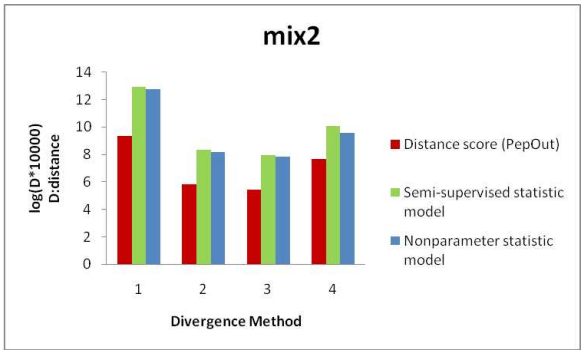
Table 3.3

Distribution closeness comparison for Mix1, Mix2 and Mix3

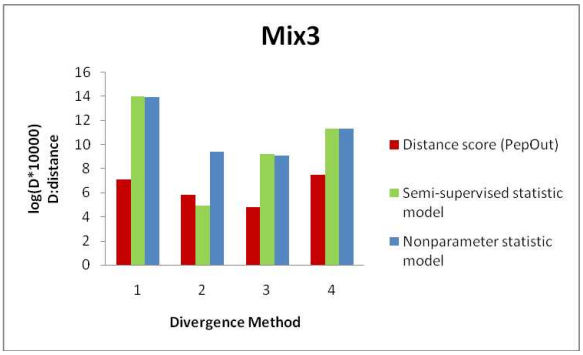
		D_{JD}	D_{JS}	D_{λ}	D_{GKL}
Mix1	Discriminant score distribution target incorrect vs. Gamma learned from decoy hits (**Semi-supervised statistics**)	0.7851	0.0322	0.0250	0.1076
	Discriminant score distribution target incorrect vs. Decoy (**Nonparametric statistics**)	0.6957	0.0292	0.0230	0.0760
	Distance score distribution target incorrect vs. decoy (**PepOut**)	0.0650	0.0056	0.0044	0.0202
Mix2	Discriminant score distribution target incorrect vs. Gamma learned from decoy hits (**Semi-supervised statistics**)	0.1486	0.0108	0.0089	0.1823
	Discriminant score distribution target incorrect vs. Decoy (**Nonparametric statistics**)	0.0490	0.0024	0.0021	0.0649
	Distance score distribution target incorrect vs. decoy (**PepOut**)	0.0138	0.0003	0.0003	0.0268
Mix3	Discriminant score distribution target incorrect vs. Gamma learned from decoy hits (**Semi-supervised statistics**)	1.6601	0.0030	0.0582	0.2605
	Discriminant score distribution target incorrect vs. Decoy (**Nonparametric statistics**)	1.6018	0.0667	0.0544	0.2607
	Distance score distribution target incorrect vs. decoy (**PepOut**)	0.0138	0.0056	0.0028	0.0176



(a) Mix1



(b) Mix2



(c) Mix3

Figure 3.8

Distribution closeness comparisons.

Table 3.4

Comparison of PepOut and Products method

		PepOut	Products
#total decoy hits		5980	10079
#total target hits		10682	13869
#total target correct hits		4806	5811
Expected error rate 0%	#TotalTargetHits	1171	2309
	#Correct	1171	2244
	Error rate	0	2.8%
	Recall	24.5%	38%
Expected error rate 1%	#TotalTargetHits	3644	5043
	#Correct	3615	3702
	Error rate	0.7%	26%
	Recall	75.2%	64%
Expected error rate 2%	#TotalTargetHits	4219	5923
	#Correct	4141	4087
	Error rate	1.8%	31%
	Recall	86.2%	70%
Expected error rate 5%	#TotalTargetHits	4713	6888
	#Correct	4511	4483
	Error rate	4.3%	35%
	Recall	94%	77%
Expected error rate 10%	#TotalTargetHits	5120	7753
	#Correct	4726	4803
	Error rate	7.6%	38%
	Recall	98.3%	83%

3.4.4 PepOut vs. Percolator

Percolator runs separate searches of MS/MS spectra against target and decoy databases. The subset of top-scoring target hits serves as a positive set, and decoy hits as a negative set. A vector of 20 features is computed for each hit. Percolator trains a support vector machine (SVM) iteratively on subsets of high-confidence target and decoy hits and assign a score for each hit. Percolator calculates the FDR using Eq. 3.6, where π_0 is the estimated proportion of target hits that are incorrect, that is $P(-)$, and T is Percolator score threshold. They report results based on π_0 = 0.9.

PepOut is compared to Percolator in terms of error rate and recall for different false discovery rates in Table 3.5. PepOut consistently identifies as many or more peptides than Percolator. Percolator always underestimates the FDR. Hulse *et al*. [37] has shown that SVMs perform poorly with unbalanced, noisy datasets. In addition, Percolator uses a separate target-decoy search and give ΔCn substantial weight. We have shown that the distribution of ΔCn scores in the decoy with separate searches does not accurately reflect the distribution of ΔCn scores for incorrect hits. Therefore the decoy hits probably do not provide a good training set for the negative examples. In addition, Percolator trains the SVM based on high-confidence target hits as the positive set if examples and may misclassify some correct hits with relative low-confidence scores. PepOut is completely data-driven and does not require a training set.

Table 3.5

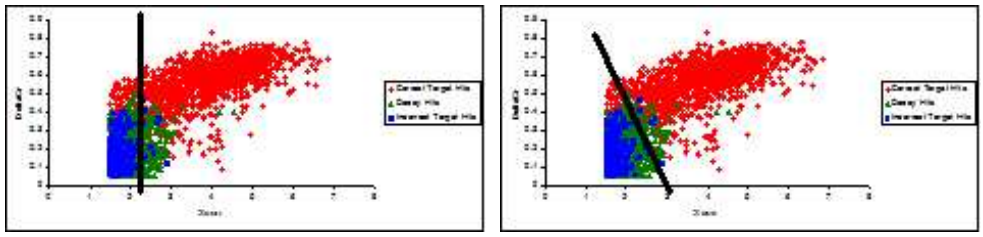
Comparison of PepOut and Percolator

		PepOut	Percolator
#total decoy hits		5980	11297
#total target hits		10682	15252
#total target correct hits		4806	5661
Expected error rate 0%	#TotalTargetHits	1171	2713
	#Correct	1171	2652
	Error rate	0	2.2%
	Recall	24.5%	46.8%
Expected error rate 1%	#TotalTargetHits	3644	4132
	#Correct	3615	3983
	Error rate	0.7%	3.6%
	Recall	75.2%	70.4%
Expected error rate 2%	#TotalTargetHits	4219	4333
	#Correct	4141	4168
	Error rate	1.8%	3.8%
	Recall	86.2%	73.5%
Expected error rate 5%	#TotalTargetHits	4713	4578
	#Correct	4511	4354
	Error rate	4.3%	5%
	Recall	94%	77%
Expected error rate 10%	#TotalTargetHits	5120	4991
	#Correct	4726	4568
	Error rate	7.6%	7.2%
	Recall	98.3%	83%

3.4.5 Summary of comparisons

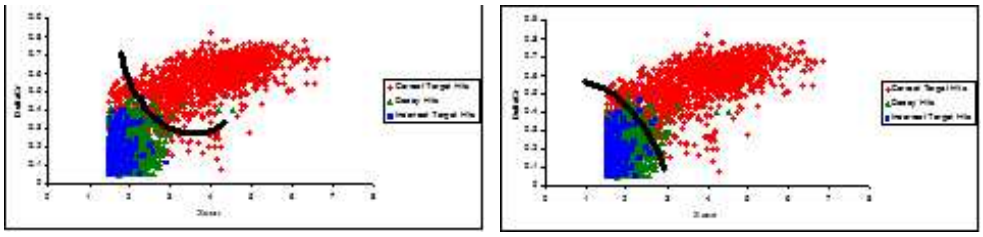
We have compare our PepOut method to several types of peptide validation tools in this chapter: threshold methods, statistical methods, the product method, and a machine learning method. Fig. 3.9 graphically illustrates the differences in the approaches used by these methods. The threshold method (Fig. 3.9(a)) bases discrimination on a single score. The statistical models (Fig. 3.9(b)) based on PeptideProphet assume the classes can be linearly discriminated. The products model (Fig. 3.9(c)) discriminates using a hyperbolic upper curve. PepOut (Fig. 3.9(d)) does not assume a particular shape of the discriminant curve, but adapts to the data. From Fig. 3.9 and the data we have presented, it is clear that PepOut identifies more peptides for a given FDR than the other methods.

Figure 3.10 demonstrates that PepOut (red line) has the closest estimation of true FDR among other methods. Figure 3.11 illustrates that Percolator has a highest recall among the other methods given a value of zero FDR, but it is known that Percolator has a highest true false discovery rate at zero of expected FDR level. Given a greater 1% of FDR value, PepOut identified the most percent of total correct peptides among these four methods. Figure 3.12 shows the number of peptide identified by four methods. According to the comparison of Fig. 3.12, Percolator identified the most peptide given a FDR zero. Recall percolator, Percolator searches mass spectra against target and decoy databases separately, and it means there are more peptide identifications than against a concatenated target and decoy database. When a greater 2% of FDR is specified, PepOut identifies the most peptides among these four methods.



(a) Xcorr threshold method

(b) Linear discriminant method



(c) Xcorr*ΔCn products method

(d) PepOut distance-based outlier method

Figure 3.9

Brief description of peptide validation methods.

According the comparison results of this section, Peptide estimates true FDR rate in a close manner, identifies almost 100% correct peptides given a 10% expected FDR.

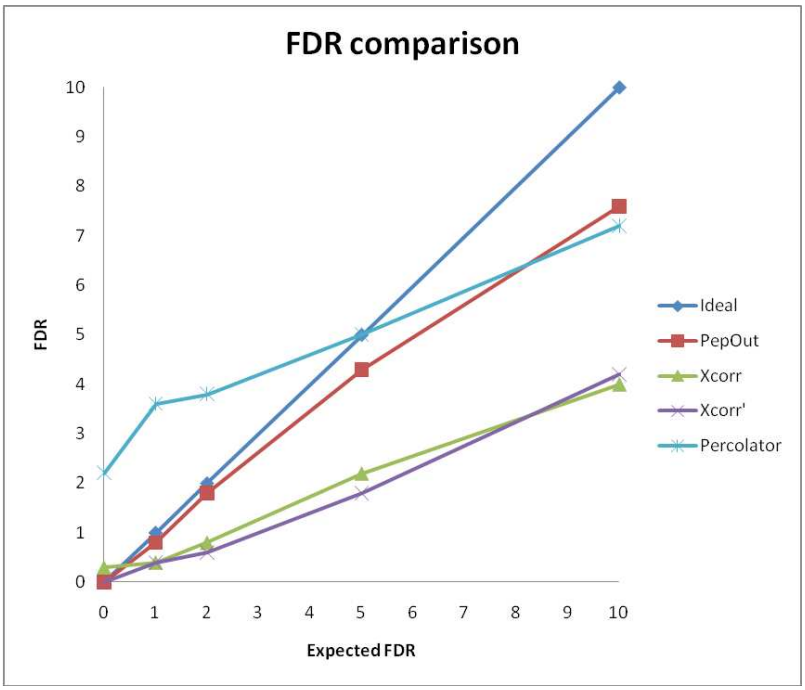


Figure 3.10

Expected FDR vs. True FDR for four methods.

3.5 Conclusions

For the high throughput analysis of MS/MS database search results, the distance-based outlier detection method described in the chapter can be used as efficient and cheap model for peptide validation since 1) the method requires minimum user interaction, the web based tool is available at http://agbase.msstate.edu/epst. The only input for validating peptides is SEQUEST search results made from a concatenated target-decoy database,

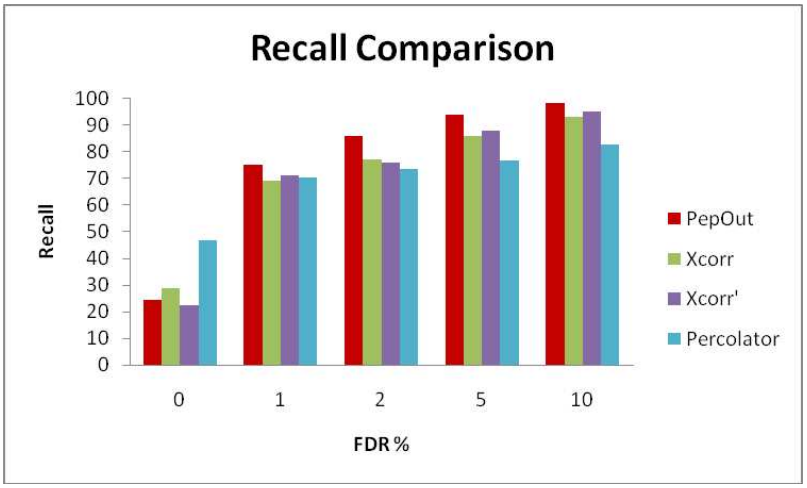


Figure 3.11

Recall comparison given an expected FDR for four methods.

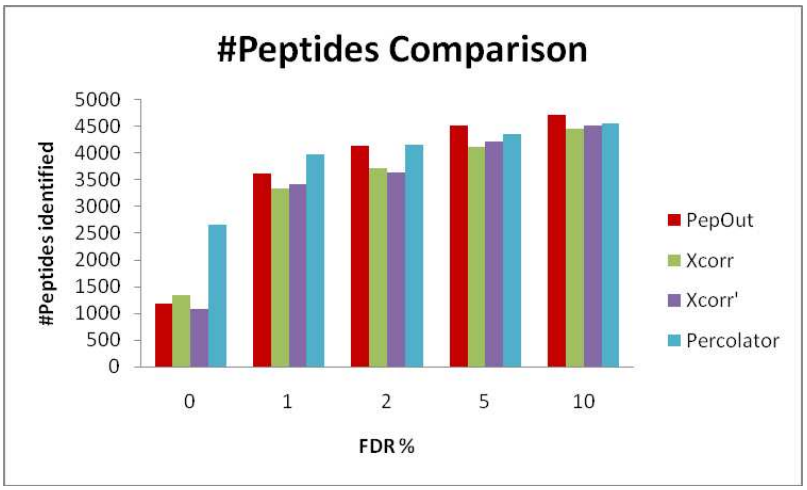


Figure 3.12

The number of peptide identified by four methods given a FDR.

with a given estimate false positive threshold; 2) the method is easy to be extended to other search scores rather than $Xcorr$ and ΔCn, also easy to be adapted to other database search algorithm such as MASCOT, X!TANDEM, and there is no additional knowledge about the search algorithm needed; 3). The outlier detection method does not need a training dataset to building the model, and use distance-based score to avoid the parameters of discriminant function; 4). The outlier detection method determines the trade-off between sensitivity and specificity by using a report FDR, also the report FDR can be used for comparison to other peptide validation method which also reports a FDR; 5) the method provides each peptide assignment a probability as true identification, which can be used as a very important evidence supporting protein-level validation.

CHAPTER 4

PROTEOGENOMIC MAPPING FOR GENE MODEL DETECTION

Structural annotation of genomes (identification of functional elements on the genome) is one of the major goals of genomics research. Most structural annotation of genomes is accomplished by computational pipelines, and we have reviewed some of these computational methods in Chapter 2. It is well-known that these computational methods have a number of shortcomings including false negative identifications (failing to identify genes that exist), false identifications, and incorrect identification of gene boundaries [40]. Proteomics data can be used to confirm the identification of genes identified by computational methods and to correct mistakes. A practical solution for generating accurate gene models for a particular genome is a combinatorial approach that includes computational predictions and experimental methods. When proteomics data is used for structural annotation, this approach is called proteogenomic mapping (PGM).

We will describe the method we have developed to use peptides identified from mass spectrometry for structural annotation of genomes. In this Chapter, we will give a brief introduction to PGM (ProteoGenomic Mapping) in section 4.1, describe the workflow we have developed for discovering potential protein coding genes in section 4.2, discuss methods for evaluating the validity of potential novel genes in section 4.3, and our experiments and results of using machine learning techniques for potential novel gene in section 4.4.

4.1 Introduction to Proteogenomic Mapping

The utility of a genome sequence in biological research depends entirely on the comprehensive description of all of its functional elements. Analysis of genome sequences is still predominantly gene centric (i.e. identifying gene models /open reading frames). In this chapter we describe a proteomics based method for identifying open reading frames that are missed by computational algorithms. Mass spectrometry based identification of peptides and proteins from biological samples provide evidence for the expression of the genome sequence at the protein level. This proteogenomic mapping method uses proteomics to both confirm computationally predicted genes and to identify novel gene models. In the chapter, we describe our proteogenomic mapping pipeline as a set of computational tools that automates the proteogenomic annotation work flow shown in Fig. 4.1.

Rapid advances in genome sequencing technologies and the resulting explosion in the availability of bacterial genome sequences highlight the need for identifying and annotating the biological function of all nucleotides in the sequence. The functional elements in bacterial genomes could be protein coding regions (genes), non-coding RNAs, as well as regulatory elements that are involved the expression of proteins and RNAs [31]. Here we focus on annotating protein coding genes and for the purposes of this dissertation, genome annotation refers to identification, demarcation and delineation of protein coding genes. Genome annotation for predicting open reading frames goes hand in hand with sequencing efforts, but most commonly relies solely on computational algorithms and does not include experimental data which is often collected for model organisms as EST/cDNA sequencing data [79]. Despite improvement in the accuracy of gene prediction programs

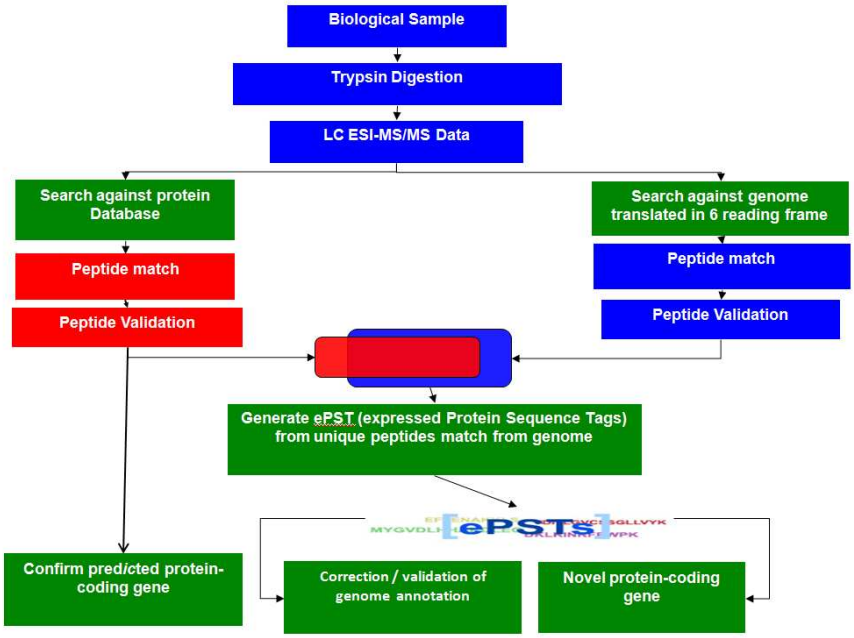


Figure 4.1

Flowchart for proteogenomic mapping used for discovery of potential novel protein-coding genes.

over the last few years, prediction of short genes still remains challenging [32]. PGM combines mass spectrometry-based proteomic workflows with computationally predicted genes to confirm expression of predicted proteins, correct gene prediction start and stop codons, identify protein post translational modifications, as well as identify novel genes missed by initial annotation [32, 59, 60, 61].

4.2 Discovery of Potential Novel Genes

Unique peptides are segments of expressed protein sequences, and can be used to discover potential protein coding genes in the genome. To discover these unique peptides, the proteogenomic mapping workflow requires a sequenced genome, the existing protein models for the genome, and a proteomics dataset which is specific to the prokaryotic genome under study.

From Fig. 4.1, biological samples are trypsin digested to peptides. These samples are run in an LC ESI-MS/MS mass spectrometer which generates mass spectra for the samples. Mass spectra then are searched against the protein database and the genome database translated in six reading frames. Those peptides that match the genome but not the protein database potentially represent novel genes or annotation errors. The peptide identifications are validated as shown in Fig. 4.1 by PepOut, the distance-based outlier detection method discussed in Chapter 3.

Figure 4.2 illustrates the process used to identify potential novel genes or to correct predicted genes. The genetic code uses three-letter nucleotide codons in DNA to specify a single amino acid in protein. Because DNA is double stranded, there are six possible

ways to translate the genome into protein sequence six possible reading frames. Some codons for amino acids are also used as start codons, indicating the beginning of translation. The genetic code also contains special codons called stop codons that signal the end of translation of nucleotides into proteins. Peptides with high confidence scores from PepOut are used to discover expressed protein sequence tags (ePSTs). As Fig. 4.2 illustrates, the genome sequence is translated in all six reading frames and the validated peptides are mapped to the translated sequence and are assumed to represent a segment of an expressed gene. The next task is to find the beginning and end of the gene. The peptide match is extended downstream to find an inframe stop codon. To find the start codon, we first locate an inframe upstream stop codon representing the end of an upstream gene. The start position of a potential protein coding gene should be located between the inframe stop codon of an upstream gene and the beginning of the peptide. In our method, we use the first start codon between the upstream inframe stop codon and the beginning of the peptide as the start codon of the potential protein coding gene. If there is no start codon between the inframe upstream stop codon and the peptide, the beginning of the peptide is used as the start position of the potential protein coding gene.

The ePSTs generated using the process described above are considered potential novel protein coding genes or extensions to predicted genes and are evaluated further as described in the next sections.

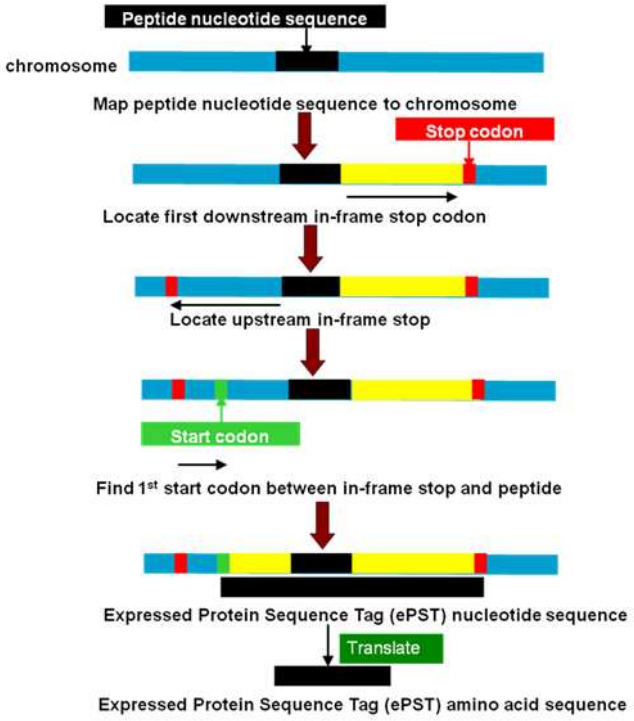


Figure 4.2

The process used to generate ePSTs from peptide sequences generated from tandem mass spectra.

4.3 Evaluation of the validity of potential novel genes

Due to noise from a variety of sources including the collection of biological samples, errors associated with mass spectrometry, sequencing errors etc, many of these potential new genes discovered by proteogenomic mapping may be false positive identifications. The most common method used for validating potential novel protein-coding genes predicted from expressed protein evidence is real-time PCR. However, real-time PCR is an expensive labor-intensive process and cannot be used for a large number of potential candidate genes. Therefore, there is a need to collect evidence of the coding potential of ePSTs and to develop machine learning methods for automatic evaluation. Biologists provided us with a list of potentially useful features for evaluating ePSTs. In this section, we will describe feature information recommended by biologists for ePST evaluation and describe how we derive this information. Table 4.1 lists the features we extract for each ePST that we will use to train our machine learning models. Note that some of these entries correspond to more than one feature. For example, for each homology search, when there is a match we also find the length of the match and the percent identity. Values of 0 were used for both the length of the match and the percent identity if there is no match.

Below we describe how some of the features in Table 4.1 are generated:

i. ePST_probability

   Each ePST may be generated by one or more peptides. All peptide matches for an ePST provide evidence that the ePST is a true identification. The ePST probability is calculated from the peptide probabilities provided by PepOut as follows:

Table 4.1

Features used for evaluating the identification of potential novel genes

Column Name	Description
ePST_length	Number of amino acids in ePST sequence
ePST_Probability	Probability that the ePST is a true identification based on the probability of the peptides used to generate the ePST.
Number_peptides	Number of peptide matched to the ePST
Coverage	Percentage of amino acids in the ePST covered by peptides Multiple peptides matching a single ePST may overlap. Amino acids matching several peptides are counted only one time.
StartCodon	The start codon for the ePST. "-" means no start codon
NewStartCodon	The start codon suggested by RBSFinder
RBS	Pattern of the ribosome binding site of the ePST generated by RBSFinder
CDD	Yes if the ePST has a conserved domain identified by CDD
Homology protein match	Yes if ePST has a homologous match in the protein database for a related organism, no otherwise. Additional information: hit length, percent identity
Homology DNA match	Yes if ePST has a homologous match in the genome database for a related organism, no otherwise. Additional information: hit length, percent identity.

$$P_{ePST} = 1 - (1 - p_1)(1 - p_2) \ldots (1 - p_n)$$

where P is the ePST probability, n is the number of peptides matching the ePST and p_i is a peptide probability provided by PepOut.

ii. ePST_Coverage

The coverage of the ePST is the percentage of the amino acids in the ePST covered by peptides, and is calculated as follows:

$$\frac{amino\_acids\_covered}{ePST\_length}$$

where $amino\_acids\_covered$ is the total number of amino acids in the ePST found in the peptides and $ePST\_length$ is the number of amino acids in the ePST. Multiple peptides matching a single ePST may overlap in different ways. Amino acids that are covered by more than one peptide are only counted once when $amino\_acids\_covered$ is computed.

iii. StartCodon

A codon is a group of three bases - A, T, C, or G - that specifies a single amino acid. A start codon is an amino acid codon that also serves as the start of translation. The codons for methionine are most often used as start codons. Bacteriologists have found that the most commonly used start codons in prokaryotes ranked by frequency of use are ATG, TTG, and GTG. Our bacteriology collaborators consider the presence of one

of these canonical start codons as positive evidence that a potential new gene is a true identification.

iv. RBS

A ribosomal binding site (RBS) is a sequence on mRNA that is bound by the ribosomes when initiating protein translation [80]. An ePST with a RBS is considered to have a higher probability as a true identification. We use the tool RBSFinder from NCBI to identify ribosome binding sites for ePSTs. RBSFinder uses previously identified genes to compute the probability of start codons. Therefore the input file for RBSFinder is a concatenation of descriptions of the genes previously predicted for the organism by a computational gene finder and descriptions of the potential new genes identified by proteogenomic mapping.

v. NewStartCodon

When RBSFinder is executed, it may suggest a new start codon suggested by RBSFinder based on the sequence shift. The new start codon could be used to solve the error in the chromosome sequence. This new start codon also will be used as a feature for ePST evaluation.

vi. CDD

Computational biologists define conserved domains based as sequence patterns that occur in many different proteins and are assumed to serve a specific function [28]. The CDD tool at NCBI imports domains from many databases including SMART, Pfam, COGs, PRK, and KOG and represents these domains using a position-specific score

matrix. To identify conserved domains in a protein sequence, the Conserved Domain Search service uses the reverse position-specific BLAST algorithm (rpsBLAST.exe from NCBI). We run CDD with all five conserved domain databases and if an ePST has a conserved domain in any one of these databases, will treat this ePST as having a conserved domain. Our biologist collaborators consider an ePST with a conserved domain to have an increased probability of being a true identification.

vii. Homologous matches to related species

The most commonly used method for structural annotation of a newly sequenced bacterial genome is to find the homologous matches in the sequences of related species. We have borrowed this idea to help verify the ePSTs we generate by our proteogenomic mapping pipeline. If an ePST has homologous match in a related species, our biology collaborators consider it to have a higher probability as a true identification. We search the ePSTs against protein databases of related species using blastp (BLAST from NCBI) to find matches to proteins previously identified in other species. It may be the case that our potential new gene is found in several species but has not been identified in any of them and therefore, we also look for homologous sequences in the genome of related species. In our experiments, we have used two related species for homology matches at both the protein and DNA level. Table 4.2 shows example features collected for the bacteria *Manhaemia haemolytica* as described in section 4.5.

The features described above to build evaluation classifiers to assess the validity of the potential novel genes. The training set used to train the classifiers is a subset of ePSTs

Table 4.2

Collected features for one of ePSTs from *Mannheimia haemolytica* dataset.

Unique ePST id	ePST2154
ePST_length_aa	812
ePST_Probability	0.99
NumOfPeptideMatch	7
Coverage	0.16
StartCodon	ATG
NewStartCodon	ATG
RBS	GGTAG
CDD	Y
HomologyProteinMatch1	Y
ProteinHitLength1	807
ProteinPercentIdentity1	0.72
HomologyDNAMatch1	Y
DNAHitLength1	2412
DNAPercentIdentity1	0.72
HomologyProtienMatch2	Y
ProteinHitLength2	803
ProteinPercentIdentity2	0.74
HomologyDNAMatch2	Y
DNAHitLength2	2409
DNAPercentIdentity2	0.74

that have been evaluated by biologists using exactly the same set of features used as input to our machine learning algorithms. Machine learning algorithms were trained to classify the ePSTs in the same way they are classified by human experts.

4.4 Experiments and Results

4.4.1 Data preparation

Our experiments were done on *Mannheimia haemolytica* which is the bacteria most commonly associated with BRD. *Mannheimia haemolytica* has been the most commonly isolated species. *M. haemolytica biotype* A serotype 1, a nonmotile, gram-negative, aerobic bacterium, is the most important etiologic agent of BRD [66]. While *M. haemolytica* normally exists at low levels as a commensal in the nasopharynx of healthy calves, it is readily isolated from stressed cattle and cattle suffering from BRD. Dr. Sarah Highlander at the Baylor College of Medicine directed genome sequencing of *M. haemolytica*. Funding was provided to complete draft coverage, but not a finished genome. The 2.4 Mbp genome of *M. haemolytica* strain PHL213, a serotype A:1 isolate from the lung of a pneumonic calf, was sequenced to 9X coverage (156 contigs $>$ 2000 bp). A list of 2434 predicted gene names and unique COG (clusters of orthologous group) numbers are available at the Baylor College of Medicine *M. haemolytica* genome web page. Currently, Dr. Highlander has organized a multi-institutional effort to annotate the *M. haemolytica* genome to standardize its gene ontology and make the data more useful to the BRD re-

search community. The genome sequence of another species in the genus *Mannheimia*, the rumen commensal M. succiniciproducens, was recently released.

SEQUEST [42] search algorithm is used for identifying peptide-mass matches. *M. haemolytica* mass spectra is search against a protein database and a genome database translated in six reading frames.

In the results, 3812 peptides were identified by PepOut with $p > 0.5$ and 3496 ePSTs were generated from 3812 peptides. Software was developed to select a training subset with all possible combinations of feature values that occur in the dataset where 10 examples are selected for each possible combination. This resulted in a training set consisting of 117 of the 3496 ePSTs generated. These 117 ePSTs in training data samples were labeled as T (true gene) or F (false gene) by two experts in bacterial genomics (Dr. Bindu nanduri and Dr. Mark Lawrence of the college of Veterinary Medicine). In the 117 training data samples, there are 47 of 117 positives and 70 of 117 negatives.

Feature selection is often used prior to training a classifier because features may be redundant, uninformative, or dependent. Use of feature selection reduces the search space to the most relevant features. We used several different feature selection algorithms provided by Weka to select a subset of the features. Since all of these algorithms have strengths and weaknesses, we selected the features most commonly found near the top of all ranked lists. It was obvious from our results that the biologists use the presence or absence of a homology match in a related species as a strong indicator of a true gene, but the length of the match or the percent identity is not a very strong indicator. Based on the feature

selection results with a number of different feature selection algorithms, 11 of 22 possible features were used to build the classification models shown in Table 4.3.

Table 4.3

Features selected for model learning and classification

Features selected	Data Type
ePST_length	Integer
ePST_probability	Real
number_peptides	Integer
Coverage	Real
hasCDD	Boolean
hasRBS	Boolean
hasStartCodon	Boolean
P1	Boolean
P2	Boolean
D1	Boolean
D2	Boolean

Preprocessing of the evaluation dataset was required because our features include a combination of real numbers and categorical data. Numerical features were discretized based on the experts' suggestions. The feature ePST length was discretized into three bins: 0-50 aa, 51-100 aa, and $>$ 100 aa, where aa is the number of amino acids. The feature ePST_probability was discretized into two bins: 0-70% and 71%-100%. The feature peptide_coverage is discretized into two bins: $<$ 30% or $\geq$ 30%. The feature ePST matches is divided into the categories: S single match or M multiple matches.

4.4.2 Model learning

We have implemented several different types of classifiers using the Weka machine learning toolkit [39]. The machine learning algorithms include tree based classifiers, rule based classifiers, function based classifiers and lazy classifiers. The accuracy of models is estimated using ten-fold cross validation. As Table 4.4 shows, all of machine learning algorithms resulted in high classification accuracy. By using a neural network model which resulted in a highest accuracy and ROC area, 242 out of 3496 ePSTs were classified as true genes and 3254 ePSTs were classified as false genes.

Table 4.4

Comparison of classification

Classifier	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Tree Classifier ID3	0.855	0.14	0.855	0.856	0.856	0.906
Tree Classifier J48	0.803	0.216	0.803	0.803	0.803	0.886
MultilayerPerception	0.838	0.158	0.843	0.838	0.839	0.933
SMO	0.838	0.172	0.838	0.838	0.838	0.833
NaïveBayes	0.863	0.12	0.874	0.863	0.864	0.95
Rule Based Classifier NNge	0.889	0.11	0.891	0.889	0.889	0.89
IBk	0.812	0.147	0.846	0.812	0.814	0.923

The high accuracy of these models indicates that the biologists are using a relatively simple and consistent set of rules to do the classification. Each of these methods has its strengths and weaknesses. Decision tree methods provide a clear visual picture of how the features are used to classify the ePSTs as shown in the example in Fig. 4.3. We tested

two types of function based classifiers - back propagation neural networks, and support vector machines. Although the neural network classifier had the highest accuracy of the classifiers tested, the resulting model is difficult to understand. The support vector machine resulted in lower accuracy than other classifiers. Rule based classifiers such as NNge (Non-Nested Generalized Exemplars) find a set of rules that can be used for classification. NNge is nearest neighbor-like algorithm using non-nest generalized exemplars which can be viewed as if-then rules. However, the resulting rules can be quite complex and difficult to understand as shown in Fig. 4.4. Instance based learning algorithms such as some nearest neighbor find the training instance closest to the given test instance, and predicts the same class as this training instance. These algorithms may result in high accuracy but do not provide any information about the knowledge domain.

The biggest limitation of the supervised classifiers we have described is the requirement of a set of labeled training data. In our domain, biological validation of potential genes is very expensive and time consuming and can only be done with a few examples using the most widely available techniques. Use of biological experts to label the data as we have done, relies on prior knowledge and biases of the experts. We have found that the experts often disagree in their evaluation. We conducted preliminary experiments with expert labeled data. After our experts examined the results in detail on a genome browser, they decided that some of their criteria had not been valid and they re-labeled the data based on their experience. In addition, although the ePSTs provide experimental evidence for new genes, the experts only classified a few of the potential new genes as true genes. This may reflect their biases rather than reality. New sequencing technologies

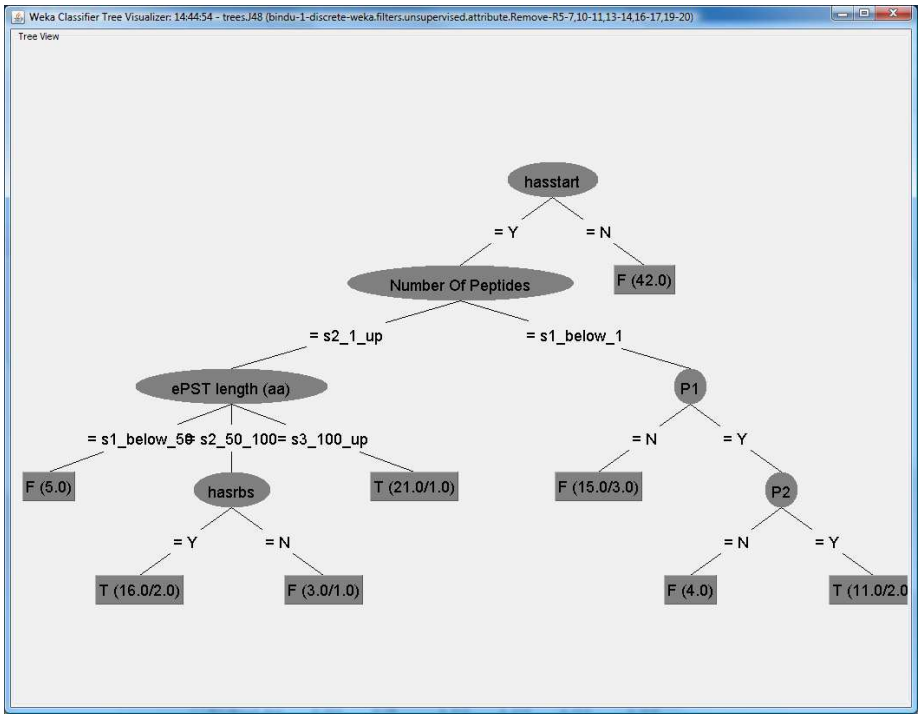


Figure 4.3

Decision tree structure (J48) for classifying ePSTs as true or false genes.

class F

IF : ePST length (aa) in {s3_100_up} ^ ePST probability in {s2_0_up,s1_below_0} ^

Number Of Peptides in {s2_1_up,s1_below_1} ^ peptide coverage in {s1_below_0} ^

if ePST has CDD in {Y} ^ P1 in {N,Y} ^ D1 in {N,Y} ^ P2 in {Y} ^ D2 in {Y} ^ hasstart in {N} ^ hasrbs in {Y} (3)

class T

IF : ePST length (aa) in {s3_100_up} ^ ePST probability in {s2_0_up} ^

Number Of Peptides in {s2_1_up} ^ peptide coverage in {s2_0_up,s1_below_0} ^

if ePST has CDD in {Y} ^ P1 in {N} ^ D1 in {N} ^ P2 in {N} ^ D2 in {N} ^ hasstart in {Y} ^ hasrbs in {Y} (3)

Figure 4.4

NNge model

have recently become available that will make it easier to validate ePSTs on a larger scale by determining expression at the mRNA level. It will be interesting to examine models based on this data compared to models based on expert evaluation when this data becomes available.

In next chapter, we will discuss an unsupervised machine learning technique, Bayesian Network classification, to learn a nature of the data and classify the data based on the data self.

CHAPTER 5

BAYESIAN NETWORKS FOR EVALUATION OF POTENTIAL NEW PROTEIN CODING GENES

Our Proteogenomic Mapping Toolbox identifies potential new protein coding genes, but because of noise from a variety of sources from the biological samples, the mass spectrometry [16], and the peptide identification processes [64], many of these potential new genes may be false positive identifications. The most common method used for validating potential novel protein-coding genes predicted from expressed protein evidence is real-time PCR [10]. However, real-time PCR is an expensive labor-intensive process and cannot be used for a large number potential candidate genes. Therefore, there is a need to develop methods for automatic evaluation of potential protein-coding genes.

In this chapter, a Bayesian network classifier will be used for automatic evaluation of potential genes. We will describe a new method we developed for the reconstruction of a Bayesian network structure using a bootstrapping strategy and a weighted bootstrapping reconstruction strategy. We also demonstrate a new method for unsupervised learning of Bayesian networks.

5.1 Background on Bayesian Networks

In this chapter a Bayesian Network classifier for evaluating potential new genes is proposed. Supervised machine learning techniques such as neural networks [62], decision trees [39], and support vector machines [4] are able to learn a model from a labeled training dataset and predict the quality of potential novel protein-coding genes using various evidential features as inputs. However, the models learned by these machine learning techniques typically do not reveal the conditional (in)dependence relations among the evidential features. Gaining insight into the relationships among features is important for biological domains. In biological research, the collected training data set is often incomplete and with very few data points and therefore methods that are robust to noisy data and low sample-size are required.

In this chapter, we describe methods for learning Bayesian networks for modeling the conditional (in)dependence relations among features of protein-coding genes and calculating confidence scores for potential novel genes based on their evidential features. To overcome the lack of data size, bootstrap methods are applied to assess the confidence measure on the arcs of the learned network structures and to identify a set of robust arcs in order to construct a final model for future predictions. The Bayesian network model learned from the current method was tested using a training experimental dataset. The results show that the method significantly improved the accuracy of the learned model in predicting potential novel genes.

Structural annotation of genomes is one of the main goals of genomic research. A proteogenomic mapping pipeline (PGM) [60, 61] has been discussed in previous chapters

for structural annotation of genomes based on proteomics data generated from mass spectrometry. This pipeline can be used to discover novel genes and provide experimental confirmation of the identification of genes predicted by computational methods. Mass spectra from proteomics experiments are matched against both a protein database and a genome database translated in all six reading frames. Those peptides that match the genome but not the protein database potentially represent evidence of expressed novel protein-coding genes. These short experimentally derived peptides are used to generate potential novel protein-coding genes by aligning the peptides to the genomic DNA and extending the translation in the 3' and 5' direction until an in-frame stop is encountered.

Although the availability of the proteogenomic mapping pipeline allows confirmation of genes identified by computational gene finders, identification of novel genes that were missed by the gene finding software, correction of the boundaries of genes that are predicted computationally, and correction of predicted splice sites in eukaryotic genomes, the peptide identifications (especially those found by querying mass spectra against the genome translated in all six reading frames) are known to generate a large number of false positive identifications and therefore many of the predicted novel genes are incorrect. Therefore, there is a need for methods to evaluate the potentially novel genes identified by proteogenomic mapping based on two types of orthogonal evidence: peptide level features and gene level features. These features have been discussed in Chapter 4 in detail. Peptide level features evaluate the strength of evidence for the peptide identification and include peptide probability provided by PepOut, the number of peptide matches to the potentially novel gene, and coverage of the potential novel gene by peptides. Gene level

features evaluate the likelihood that the predicted novel gene has characteristics expected of a gene and include the length of a potential novel gene, the probability of the potential novel gene calculated from peptide probabilities, the presence of a start codon, the presence of a ribosomal binding site pattern found by RBSFinder (for prokaryotes), the presence of conserved protein domains, and homology of the potential new gene with proteins or nucleotide sequences of related organisms. A detailed description of these features description has been given in Chapter 4.

Machine learning algorithms, such as decision trees, naïve Bayes [36], and Bayesian networks [36, 41, 44], can be used to evaluate these potential novel genes. Decision trees do not consider the correlations among the features. Nave Bayes assumes that all features are independent. Neural networks learn a model that best fits training data set but do not reveal dependence relationships among the features. In comparison, Bayesian networks learn the uncertain relationships among the features and provide a better understanding of the feature domain of genes.

Bayesian networks provide intuitive and compact representations of uncertain relations among the random variables in a domain and can be used to discover the conditional dependence or independence relationships among random variables in a knowledge domain. Bayesian networks have been applied in a broad range of computational biology problems. In biological research, it is usually time-consuming and expensive to collect training samples. A typical training dataset has relatively few data points in comparison to the number of random variables. For such a dataset, there may be many models that fit the data equally well but have very different structures. Using a single learned model to

predict new relations may generate many false positives. In this chapter, we address this problem by applying bootstrap methods to find robust features that appear most frequently in models learned from resampled datasets and by assembling these features into a final model to accurately predict the confidence scores for potential novel genes.

5.2 Experimental Datasets

The bovine respiratory disease pathogen *Mannheimia haemolytica* strain PHL213 serotype $A:1$ isolated from the lung of a pneumonic calf with a genome sequenced at 9X coverage was used in this study. Dr. Mark Lawrence's research group cultivated *M. haemolytica*, isolated the proteins, trypsin digested the proteins and analyzed the tryptic peptides by MuDPIT as described in [66]. Tandem mass spectra generated by 2D LC ESI MS/MS were searched using SEQUEST (Bioworks 3.2 cluster; ThermoElectron, except the mass spectra were searched against the genome sequence translated in all six potential frames in addition to searching against a subset of the non-redundant protein database (NRPD) consisting of all *M. haemolytica* proteins. We utilized a 0^{th} order random decoy database and our distance based outlier detection method to assign probabilities to peptide identifications. Peptides identified at $p \leq 0.5$ from translated DNA sequence were compared with peptides identified from the NRPD ($p \leq 0.5$), and peptides unique to the translated nucleotide dataset were utilized for generating potential novel genes using our proteogenomic mapping pipeline.

5.3 Methods and Results

This chapter focuses on the use of Bayesian networks for evaluating potentially novel genes identified using the proteogenomic mapping pipeline. These potentially novel genes are called expressed Protein Sequence Tags (ePSTs) [60, 61]. Due to noise and in accuracy from a number of sources, some of the potential novel genes discovered by the proteogenomic mapping pipeline probably do not represent true novel genes. It is thus important to develop methods for evaluating the quality of potential novel genes and computing their confidence scores for being true identifications.

Construction of a Bayesian network requires identification of relevant features (variables), discretization of continuous variables. We have discussed data preprocessing and feature selection in Chapter 4.

5.3.1 Features and Data Preprocessing

There are many evidential features which can be used for evaluating potentially novel genes. Some features play an essential role in supporting or refuting a gene as truly novel while others do not. There are also strong correlations among these features. A detailed description of these features was presented in Chapter 4. Some features are categorical values and some are continuous values. In order to learn general conditional probability distributions (CPDs), the continuous variables need to be discretized. The continuous variables are discretized using experts suggestion discussed in Chapter 4.

5.3.2 Model Construction

5.3.2.1 Training dataset

To build a robust Bayesian network classifier, we collected training samples representing all possible feature combinations. Two Ph.D. level bacteriologists who conducted the experimental work with *Mannheimia haemolytica* evaluated the training examples using the same evidence supplied to our machine learning algorithms and rated the peptides on a scale from 1-5 where 1 is lowest and 5 is highest. Based on these expert rankings, our goal was to build a Bayesian network that can provide an evaluation score for each ePST that is an estimate of the likelihood of a true identification.

Given the training dataset, we use Bayesian networks to analyze the correlations among these features and compute confidence scores for potential novel genes. Bayesian networks can be utilized to discover the conditional dependence or independence relationships of random variables in a knowledge domain.

5.3.2.2 Network Learning Using Standard Methods

We tested three Bayesian network learning algorithms in building our models: Naïve Bayes, Greedy thick thinning [78], and PC learning algorithms [12]. The naïve Bayes classifier assumes that all variables are independent, and there are no relationships among all features. The class nodes should have arcs to all feature nodes, and the parameter (conditional probability table) is learned from the data set. The greedy thick thinning algorithm first creates a draft model by computing pairwise closeness measures. After

that, the algorithm adds arcs when the pairs of nodes are not conditional independent given conditioning variables. Finally, each arc is reevaluated and will be removed if the two end nodes are independent. The PC algorithm is a method based on statistical testing. It first creates an undirected graph based on the results of pairwise independence testing. Then, it thins the model by sequentially removing edges with zero-order conditional independence relations, with first-order conditional independence relations, and so on.

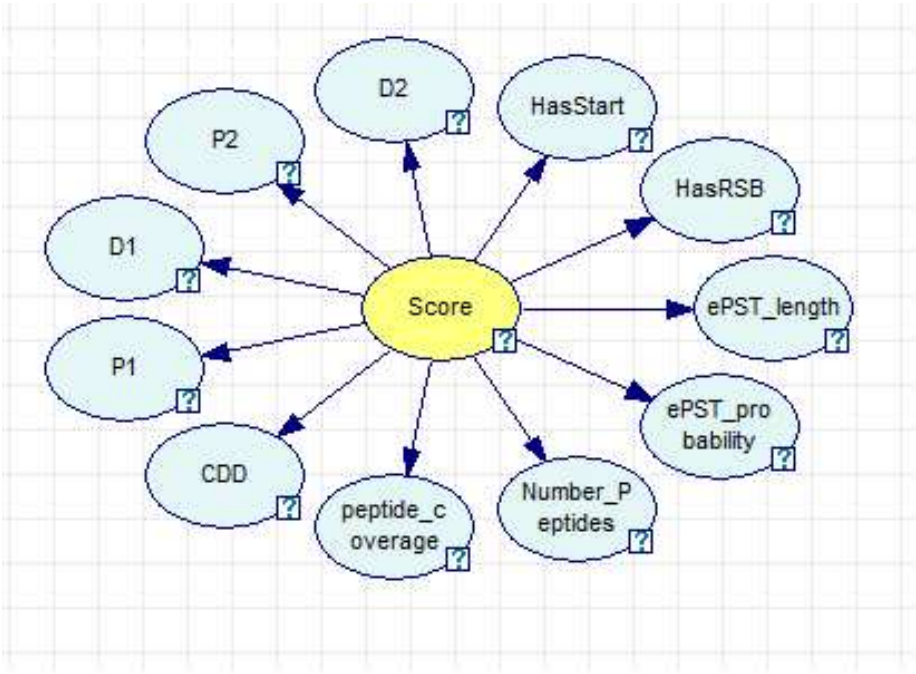


Figure 5.1

Network structure learned using naïve Bayes algorithm.

Figures 5.1- 5.3 show the networks learned by the naïve Bayes, greedy thick thinning and PC algorithms from a training dataset with 117 samples. The naïve Bayes leaning algorithm assume that all features (e.g. ePST_length, ePST_probability, StartCodon etc)

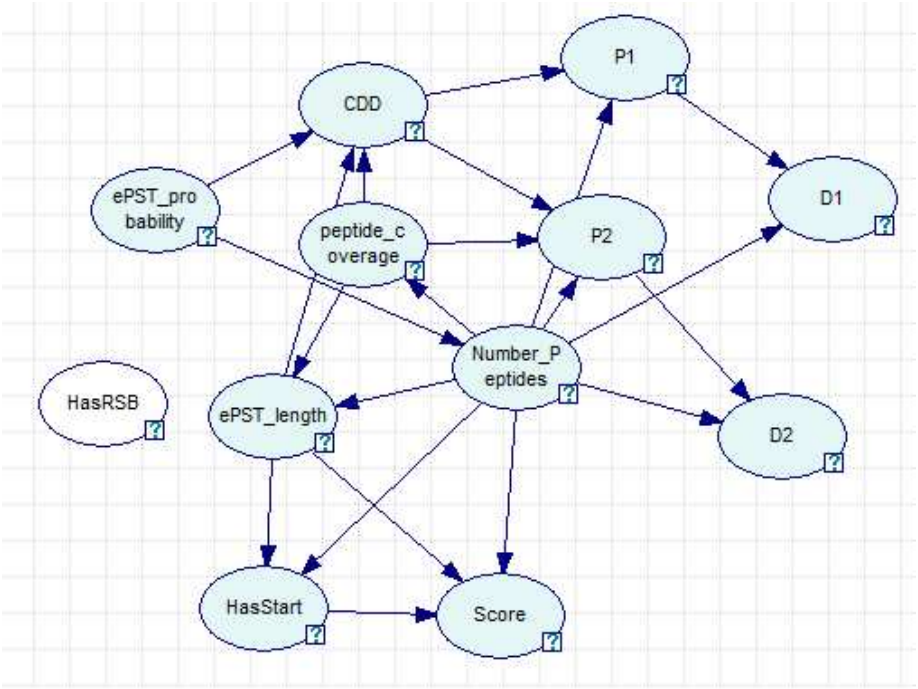


Figure 5.2

Network learned using greedy network structure learning algorithm.

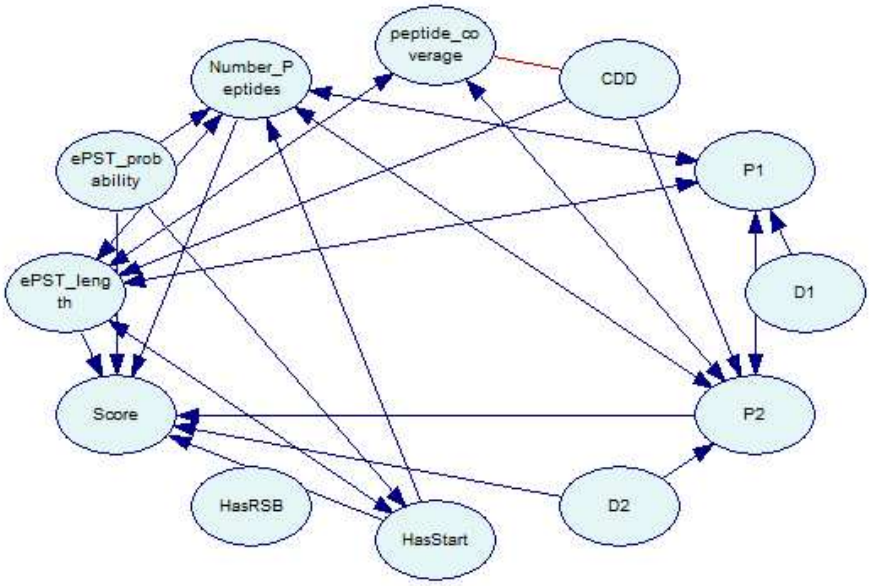


Figure 5.3

Network learned using PC network structure learning algorithm.

are independent, and the class node (node “Score”) is affected by all these features. The greedy algorithm [78] learned a simpler network structure than the PC algorithm [12] since the greedy algorithm only considers local closeness. The PC algorithm discovers the conditional relationship for all pair of variables and yields a much more complex network structure than the greedy algorithm. The PC algorithm is also much more computationally intensive than the other two algorithms requiring more than one day to complete with 100 data samples with 12 features.

The network structures obtained show that the length of potential novel gene is highly related to the presence of a start codon. These results indicate that longer open reading frames are more likely to have a traditional start codon. The networks also show that a match to a sequence in one related organism is also highly related to a match in the other

related organism. These relationships are meaningful in biology and have been verified by biologists. Such relationships can be used in future analysis.

We compared the classification accuracy of these learned Bayesian network models, the results of which are shown in Table 3. 10-fold cross validation was used to obtain the accuracy results. The results show that the model learned by the PC algorithm has higher accuracy than the models learned by other algorithms.

The performance of a model learned by a supervised learning method is affected by the quality and quantity of training dataset. Given that there are only few data points in the training dataset, a natural concern is whether the learned network models are reliable. Unfortunately, we have found that when we apply the learning methods above to different subsets of the training data, we obtain widely varying network structures. This motivated us to develop new methods for generating more robust models as described in the next sections.

5.3.2.3 Network Model Evaluation and Reconstruction

Evaluating the models is difficult given so few data points. One way to address the problem is to assess individual network features, e.g., edges. Cross-validation [44] and bootstrapping [17] are both methods for evaluating the accuracy of a classifier or predictor based on resampling [11]. The resulting estimates are often used for choosing among various models, such as different network architectures. Bootstrapping performs better than cross-validation in many cases. In the simplest form of bootstrapping, instead of repeatedly analyzing subsets of the data, we repeatedly analyze subsamples of the data.

Each subsample is a random sample with replacement from the full sample. There are many more sophisticated bootstrap methods that can be used not only for estimating the accuracy of a classifier but also for estimating confidence bounds for network outputs [17].

In our case, we have applied the bootstrap strategy to reconstruct a network structure based on the evaluation score of the model. The main idea is to resample from the original dataset with replacement to generate many pseudo datasets. Together, these pseudo datasets allows a high-scoring model to be created using learning methods. These models then serve as a set of network structures that are used to estimate the confidence of network features. In this paper, we focus our attention on first-order features, the arcs. The network features that have high confidence can be assembled to build a more reliable model. Figure 5.4 shows an intuitive graphical illustration of the process of using the bootstrap method to construct a robust model from a relatively small dataset. The number of times each edge appears in a learned network is entered into a matrix. If the number of times an edge appears is greater than a threshold value, the edge is selected for inclusion in the final model.

Once the network structure is learned by our bootstrap method, the CPTs (conditional probability tables) are learned from the original dataset. We applied the bootstrap strategy to the two Bayesian network learning algorithms, PC and greedy, described previously. Table 5.1 shows the classification accuracy of the new models based on 10-fold cross validation. It is clear that bootstrap method significantly improved the classification accuracy of the learned models for both the PC and the greedy learning algorithms. Note that naïve

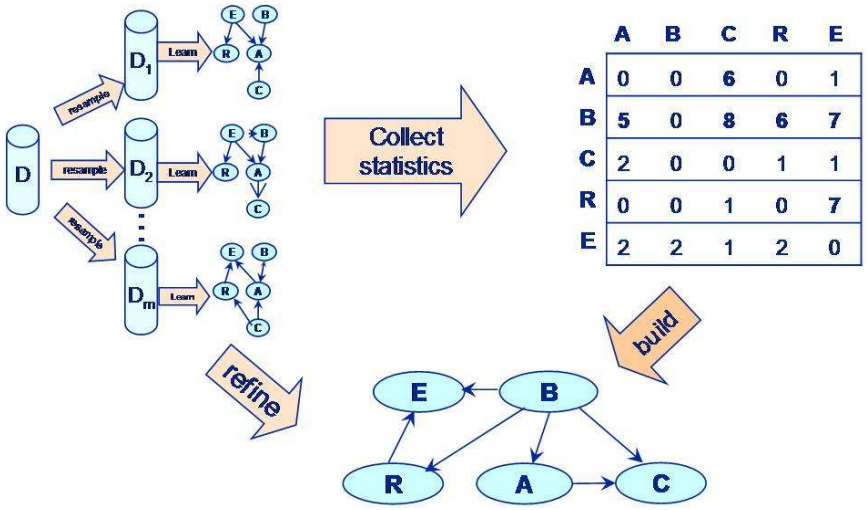


Figure 5.4

Workflow of bootstrap strategy for rebuilding a robust network model from a small dataset.

Bayes only supports a single network structure and thus cannot benefit from this reconstruction.

5.3.2.4 Weighted Model Reconstruction

In the process for model reconstruction described above, we simply set a threshold count for the frequency of occurrences of arcs. The accuracy of the model containing these arcs was not considered. In order to take into account of the quality measure of each network model learned from the resampling sub dataset, we weighted each model by the accuracy of the network. The confidence of the arc ($A \rightarrow B$) is calculated as follow:

$$Confidence(A \rightarrow B) = \sum_{i=0}^{N} (M_i(A \rightarrow B \in G_i)),$$

where N is the number of iterations, M_i is accuracy of the model i and G_i is the network structure of model i.

We reconstruct the network model using arcs with a confidence value above a threshold. The remainder of the process is same as illustrated in Figure 5.2.

We applied the weighted bootstrap strategy to the two Bayesian network learning algorithms: PC and greedy. Table 5.1 shows the classification accuracy of the new models reconstructed by the weighted bootstrap strategy. The method takes into account the quality of each network model and yields a higher accuracy network model.

Table 5.1

Comparison of learning methods

	TP Rate for Class Y	TP for Class N	Precision
Greedy Algorithm	78%	80%	79%
PC Algorithm	82%	81%	81.5%
Greedy with Bootstrap	81%	78%	79.5%
PC with Bootstrap	84%	82%	83%
Greedy with ranked Bootstrap	82%	78%	80%
PC with ranked Bootstrap	84%	82%	83%
Unsupervised Greedy	76%	80%	78%
Unsupervised PC	78%	81%	79.5%

5.3.2.5 Learning Network Models with Unlabeled Data

It is well known that collecting biological training samples is both time-consuming and expensive, especially when the training samples must be annotated by experts. In order to address this problem, we propose to first learn a network structure using an unlabeled dataset as illustrated in Figure 5.5. Note that there is no class label in this network. The network structure provides the prior relationships among features. It is known that all these

features contribute to the classification of potential novel genes. Therefore, we construct a new network by adding a label node to the network and adding arcs from the label node to all feature nodes. Figure 5.6 shows the newly constructed network structure for our example. After the structure of the network is determined, an EM algorithm is applied to learn the parameters of the new network from the test dataset. Note that only unlabeled data is used, even after adding the label node. The final model can then be used to evaluate the confidence score of the potential novel genes.

Although the accuracy of the model learned by this unsupervised method is not as good as for the supervised network structure learning methods, it is useful when no training dataset is available. In addition, the training set used for evaluating the accuracy of this model in predicting the likelihood that a potential gene is actually a gene is based on human expertise and is therefore biased by the biologists' knowledge of what "typical" genes look like. This unsupervised method has the potential to provide an unbiased evaluation of the potential genes and provides information about the intrinsic structure of the data.

5.4 Conclusions

In this chapter, we describe three new algorithms for constructing Bayesian networks from sparse biological data. The Bayesian network is used to model the correlations among various evidential features and to compute confidence scores for potential novel genes in order to classify them as true or false identifications. In order to alleviate the scarce data problem, a bootstrap method and weighted bootstrap method were developed to assess the confidence measure of features in the learned structure and the most robust

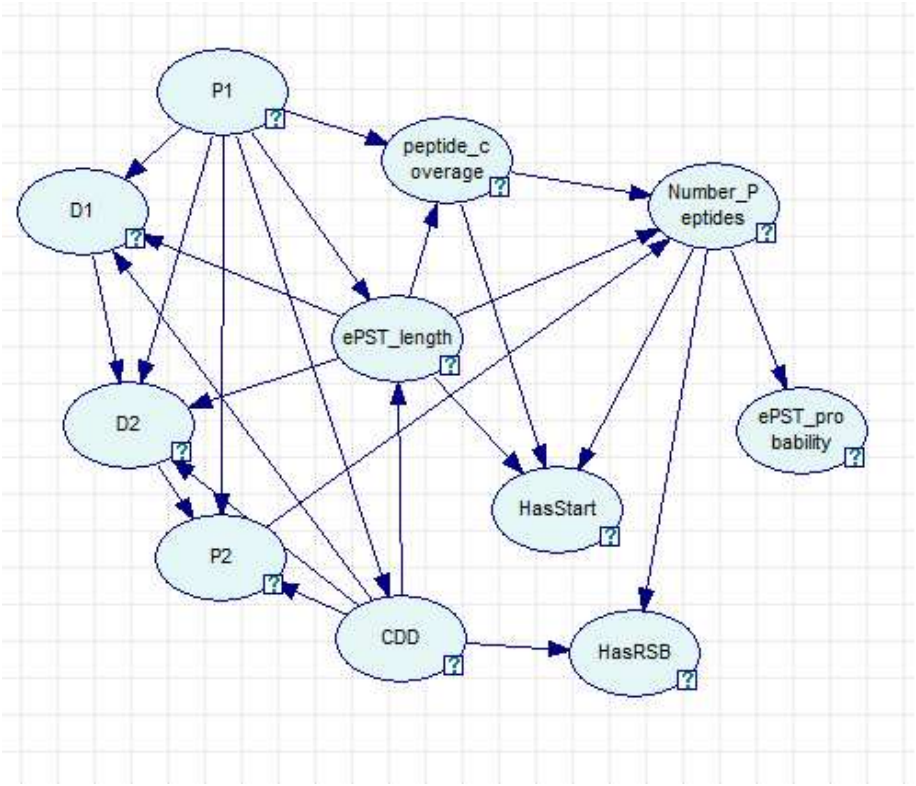


Figure 5.5

Network structure learned from unlabeled dataset.

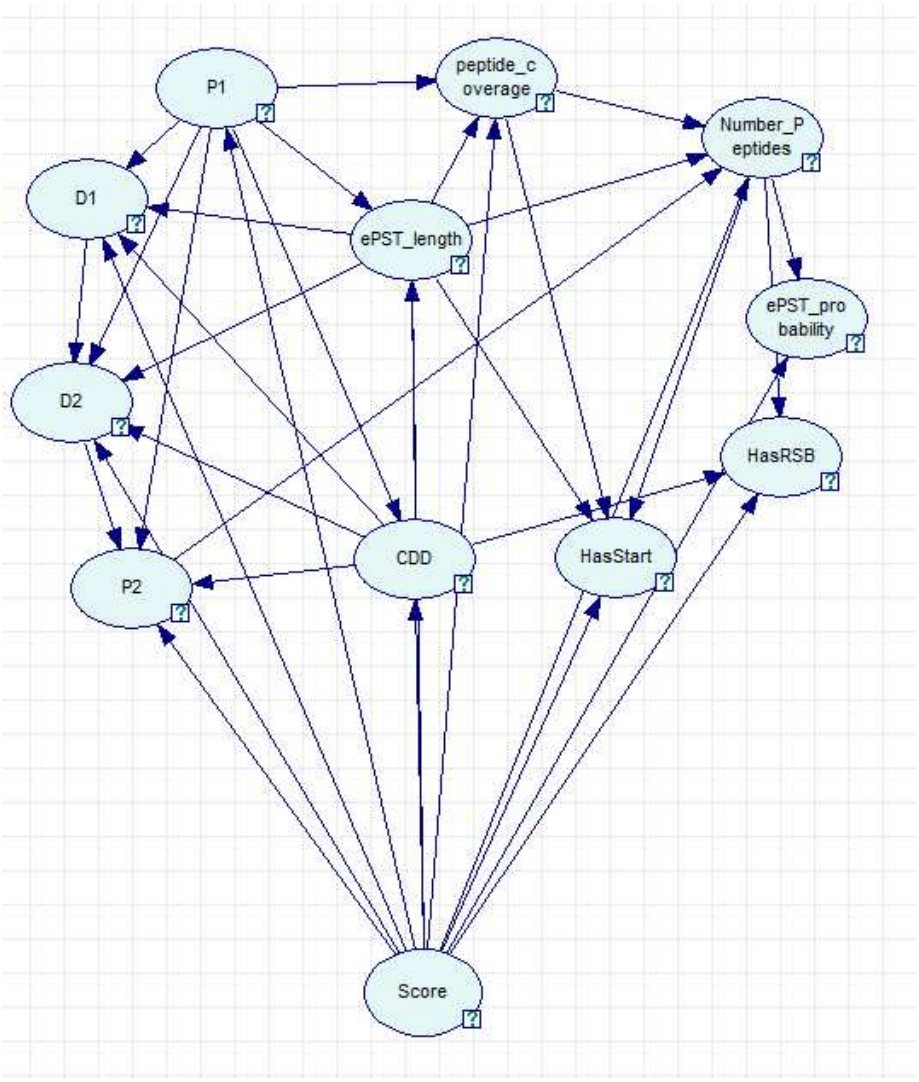


Figure 5.6

New network constructed by adding label node and arcs from label node to all feature nodes.

features are used to build more reliable models. We tested these methods on a training experimental dataset with 117 data points. The results show that the bootstrap methods yielded Bayesian networks with significantly improved accuracy. In addition, because labeled data is often not available, we have also developed a new unsupervised network structure learning method that learns an initial network structure from the unlabeled dataset and then constructs a new network by adding a label node to the network and adding arcs from the label node to all feature nodes. An EM algorithm is used to learn the parameters of the new model from the unlabeled data. This unsupervised Bayesian network structure learning method can be used when the training dataset is not available. It can also overcome biases of labels provided by human experts.

The work in this chapter focuses on assessing the robustness of first-order network features, the arcs. As future work, we plan to evaluate higher-order features such as V-structures and Markov blankets. Furthermore, there could be potential conflicts among the learned substructures. In the future we plan to develop approaches for resolving these conflicts in constructing a final robust and representative model.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Structural annotation of genomes is a major goal of genome research and the traditional tools for structural annotation of a genome are based on genome sequence. In Chapter 2, we addressed the limitations of traditional computational tools for structural annotation of microbial genomes. Homology search gene finding tools are based on sequence similarities between unknown genome and its related genomes and can only find genes which have been annotated in related genomes. Model building tools such as GeneMarker, Glimmer etc. build a Markov chain model based on existing genes and these model building tools have limitations for finding short genes or genes that differ substantially from known genes. The major contribution of this dissertation is to find novel protein coding genes based on protein expression data. Gene expression data at the protein level provides evidence of the existence of protein coding genes. In the dissertation, we demonstrated that the proteogenomic mapping method can be used as a supplement for computational gene finding tools.

6.1 Contributions

This dissertation makes several contributions to the field of data mining and genomic research. We developed a semi-supervised distance-based outlier detection algorithm for

peptide validation, a proteogenomic mapping algorithm for discovery of novel protein coding genes, and an unsupervised Bayesian network model to obtain insight into protein coding gene models.

6.1.1 Semi-supervised outlier detection for peptide validation

A unique algorithm, PepOut, was developed to estimate the accuracy of peptide assignments to tandem mass spectra (MS/MS) using a distance based method for outlier detection. Unlike other supervised machine learning techniques which use a labeled training data for model learning, or an unsupervised machine learning techniques which learn a model from data with no guidance, PepOut does not need training data for building a classifier to discriminate correct peptide assignments from incorrect peptide assignments and takes advantage of the target-decoy strategy which uses the decoy hits to model the incorrect hits to drive the semi-supervised learning process.

To utilize the target-decoy strategy correctly, we performed comprehensive analysis on SEQUEST output and concluded that the target-decoy strategy is misused by some researchers. As a matter of fact, the assumption of target-decoy strategy that the number of decoy hits reflects the number of target incorrect hits, is not precise. The distributions of database search scores of decoy hits, however, provides an accurate model of the distribution of target incorrect scores. A major controversy within the proteomics community has been how to perform the target-decoy search. Should the mass spectra be searched against a concatenated target-decoy database or should the two databases be searched separately? We addressed this question by analyzing SEQUEST search results of the ISB standard

protein mixture. We found that searching a concatenated target-decoy database results in a score distribution of decoy hits can be used to estimate that of target incorrect hits. When the two databases are searched separately, the distribution of decoy match scores does not provide an accurate estimate of the score distribution of target incorrect hits.

To better discriminate correct hits from incorrect hits, we calculate a distance score on the score space for each target and decoy hit to its 5 nearest decoy neighbors. This distance score does not assume that correct and incorrect hits are linearly separable. We estimate the priori probability of $P(-)$ based on the distance score distribution of decoy hits and do not subjectively assume this priori probability.

We demonstrate that PepOut identifies as many or more peptides for a given expected False Discovery Rate and that it provides a much more accurate estimate of the true FDR than other popular methods.

6.1.2 Proteogenomic mapping for discovery of novel protein coding genes

A novel proteogenomic mapping algorithm (PGM pipeline) was developed to generate potential protein coding genes by aligning the peptides to the genomic DNA and extending in both the 5 and 3 directions. The contribution of proteogenomic mapping algorithms is to take advantage of proteomics data for genomic research. Our pipeline identifies potential new protein coding genes and corrections to the boundaries of previously identified genes. Analysis by biology collaborators revealed that many genes previously identified as pseudo-genes by computational gene finders are actually expressed genes with sequencing errors.

We demonstrate that machine learning algorithms can be used to evaluate the evidence in support of potential new genes as actual genes. We used a training set labeled by biologists to build different kinds of models to predict confidence in the potential new genes. Decision trees and neural networks were shown to be the most accurate predictors.

6.1.3 Bayesian network with bootstrap strategy for evaluation of potential novel protein coding genes

We also use Bayesian network models for evaluating potential protein coding genes. The contribution of this study is the development of three new algorithms for constructing Bayesian networks from sparse biological data. A bootstrap method and a weighted bootstrap method were developed to assess the confidence measure of features in the learned structure and to select the most robust features to build more reliable models. We also developed an unsupervised method for learning Bayesian network structure that can be used to learn the intrinsic structure of the data and that can be applied when labeled training data is not available.

6.2 Future work

Although this dissertation contributed to the field of data mining and genomic research, there are many additional issues that are worth investigating. We plan to extend this research along several directions.

6.2.1 PepOut extensions

- The version of PepOut developed in the current study is for validating peptide assignments generated from the SEQUEST search algorithm. The method is easily extended to other database search algorithms and can also be used as an ensemble method for combining the scores of several different search algorithms.

- The current version of PepOut is not scalable for a very large datasets because its time complexity is near $O(nm)$ where n is the number of target hits and m is number of decoy hits but this algorithm. However, the algorithm can be easily parallelized and we plan to develop a parallel version of PepOut.

- The current version of PepOut is only for outlier detection. We plan to extend the algorithm for outlier prediction. An outlier prediction model will be built based on any mass spectra searched against a randomly generated protein database because it is known that search score distributions of decoy hits for one genome are similar to that of another genome. An outlier prediction model does not compute the distance scores for all target and decoy hits to their K-nearest-neighbor decoy hits. The outlier prediction model will provide a subset of decoy hits which will be used for distance score calculation and also provide a score distribution of decoy hits.

6.2.2 PGM extension

- The PGM (ProteoGenomic mapping) pipeline we developed for structural annotation of prokaryotic genomes. In the future this algorithm can be extended for eukaryotic genomes where intron-exon structure must be taken into account.

- PGM can also be parallelized to improve the performance.

6.2.3 Bayesian network model extension

- The current version of our new supervised Bayesian models focuses on assessing the robustness of first-order network features, the arcs. For future work, we plan to evaluate higher-order features such as V-structures and Markov blankets.

- When higher-order features are used, there could be potential conflicts among the learned substructures. We plan to develop approaches for resolving these conflicts in constructing a final robust and representative model.

- The network reconstruction method can be extended for gene regulation network construction and other research area.

REFERENCES

[1] "ROCK: A Robust Clustering Algorithm for Categorical Attributes," 1999.

[2] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J Mol Biol*, vol. 215, no. 3, Oct 5 1990, pp. 403–10.

[3] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Res*, vol. 25, no. 17, Sep 1 1997, pp. 3389–402.

[4] D. C. Anderson, W. Li, D. G. Payan, and W. S. Noble, "A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: support vector machine classification of peptide MS/MS spectra and SEQUEST scores," *J Proteome Res*, vol. 2, no. 2, Mar-Apr 2003, pp. 137–46.

[5] F. Angiulli, S. Basta, and C. Pizzuti, "Distance-based detection and prediction of outliers," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 2, 2006, pp. 145–160.

[6] T. Baczek, A. Bucinski, A. R. Ivanov, and R. Kaliszan, "Artificial Neural Network Analysis for Evaluation of Peptide MS/MS Spectra in Proteomics," *Anal. Chem.*, vol. 76, no. 6, 2004, pp. 1726–1732.

[7] J. Besemer and M. Borodovsky, "Heuristic approach to deriving models for gene finding," *Nucl. Acids Res.*, vol. 27, no. 19, October 1, 1999, pp. 3911–3920.

[8] J. Besemer, A. Lomsadze, and M. Borodovsky, "GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions," *Nucleic Acids Res*, vol. 29, 2001, pp. 2607 – 2618.

[9] M. Borodovsky and J. McIninch, "GeneMark: parallel gene recognition for both DNA strands," *Computers and Chemistry*, vol. Vol. 17, no. No. 19, 1993, pp. pp. 123–133.

[10] M. Campbell, "Real-time PCR, molecular beacon method," 2003.

[11] S. L. Cawley and L. Pachter, "HMM sampling and applications to gene finding and alternative splicing," *Bioinformatics*, vol. 19 Suppl 2, no. 2, Oct 2003, pp. ii36–41.

[12] J. Cheng, D. Bell, and W. Liu, "An algorithm for Bayesian belief network construction from data," *Proceedings of AI and STAT'97 (pp.83-90)*, 1997.

[13] H. Choi and A. I. Nesvizhskii, "False discovery rates and related statistical concepts in mass spectrometry-based proteomics," *J Proteome Res*, vol. 7, no. 1, Jan 2008, pp. 47–50.

[14] H. Choi and A. I. Nesvizhskii, "Semisupervised model-based validation of peptide identifications in mass spectrometry-based proteomics," *J Proteome Res*, vol. 7, no. 1, Jan 2008, pp. 254–65.

[15] J. Colinge, I. Cusin, S. Reffas, E. Mahe, A. Niknejad, P. A. Rey, H. Mattou, M. Moniatte, and L. Bougueleret, "Experiments in Searching Small Proteins in Unannotated Large Eukaryotic Genomes," 2005.

[16] R. Craig and R. C. Beavis, "TANDEM: matching proteins with tandem mass spectra," *Bioinformatics*, vol. 20, no. 9, Jun 12 2004, pp. 1466–7.

[17] A. C. Davison and D. V. Hinkley, *Bootstrap Methods and Their Application*, Cambridge University Press, 2006.

[18] G. A. de Souza, H. Malen, T. Softeland, G. Saelensminde, S. Prasad, I. Jonassen, and H. G. Wiker, "High accuracy mass spectrometry analysis as a tool to verify and improve gene annotation using Mycobacterium tuberculosis as an example," *BMC Genomics*, vol. 9, no. 316, 2008, p. 316.

[19] A. L. Delcher, K. A. Bratke, E. C. Powers, and S. L. Salzberg, "Identifying bacterial genes and endosymbiont DNA with Glimmer," *Bioinformatics*, vol. 23, no. 6, March 15, 2007, pp. 673–679.

[20] A. L. Delcher, D. Harmon, S. Kasif, O. White, and S. L. Salzberg, "Improved microbial gene identification with GLIMMER," *Nucleic Acids Res*, vol. 27, no. 23, Dec 1 1999, pp. 4636–41.

[21] M. K. Edwin and T. N. Raymond, "Algorithms for Mining Distance-Based Outliers in Large Datasets," 1998.

[22] J. E. Elias and S. P. Gygi, "Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry," *Nat Methods*, vol. 4, no. 3, Mar 2007, pp. 207–14.

[23] J. K. Eng, A. L. McCormack, and J. R. Yates, "An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database," *Journal of the American Society for Mass Spectrometry*, vol. 5, no. 11, 1994, pp. 976–989.

[24] A. Fabrizio and P. Clara, "Outlier Mining in Large High-Dimensional Data Sets," *IEEE Trans. on Knowl. and Data Eng.*, vol. 17, no. 2, 2005, pp. 203–215.

[25] T. Fawcett and P. Foster, "Adaptive Fraud Detection," *Data Min. Knowl. Discov.*, vol. 1, no. 3, 1997, pp. 291–316.

[26] D. Frishman, A. Mironov, H. W. Mewes, and M. Gelfand, "Combining diverse evidence for gene recognition in completely sequenced bacterial genomes," *Nucleic Acids Res*, vol. 26, no. 12, Jun 15 1998, pp. 2941–7.

[27] M. J. Fullwood, C.-L. Wei, E. T. Liu, and Y. Ruan, "Next-generation DNA sequencing of paired-end tags (PET) for transcriptome and genome analyses," *Genome Research*, vol. 19, no. 4, April 2009, pp. 521–532.

[28] L. Y. Geer, M. Domrachev, D. J. Lipman, and S. H. Bryant, "CDART: protein homology by domain architecture," *Genome Res*, vol. 12, no. 10, Oct 2002, pp. 1619–23.

[29] L. Y. Geer, S. P. Markey, J. A. Kowalak, L. Wagner, M. Xu, D. M. Maynard, X. Yang, W. Shi, and S. H. Bryant, "Open Mass Spectrometry Search Algorithm," 2004.

[30] W. Gish and D. J. States, "Identification of protein coding regions by database similarity search," *Nat Genet*, vol. 3, no. 3, Mar 1993, pp. 266–72.

[31] S. Gottesman, "Micros for microbes: non-coding regulatory RNAs in bacteria," *Trends Genet*, vol. 21, no. 7, Jul 2005, pp. 399–404.

[32] R. Guigo, P. Flicek, J. F. Abril, A. Reymond, J. Lagarde, F. Denoeud, S. Antonarakis, M. Ashburner, V. B. Bajic, E. Birney, R. Castelo, E. Eyras, C. Ucla, T. R. Gingeras, J. Harrow, T. Hubbard, S. E. Lewis, and M. G. Reese, "EGASP: the human ENCODE Genome Annotation Assessment Project," *Genome Biol*, vol. 7 Suppl 1, no. 1, 2006, pp. S2 1–31.

[33] F. B. Guo, O. H. Y, and C. T. Zhang, "ZCURVE: a new system for recognizing protein-coding genes in bacterial and archaeal genomes," *Nucleic Acids Res*, vol. 31, 2003, pp. 1780 – 1789.

[34] F. B. Guo and C. T. Zhang, "ZCURVE_V: a new self-training system for recognizing protein-coding genes in viral and phage genomes," *BMC Bioinformatics*, vol. 7, no. 9, 2006, p. 9.

[35] D. Hawkins, *Identification of Outliers*, Chapman and Hall, 1980.

[36] D. Heckerman, *A Tutorial on Learning With Bayesian Networks*, 1995.

[37] J. Hulse, T. Khoshgoftaar, and A. Napolitano, "Skewed class distributions and mislabeled examples," *IEEE International Conference on Data Ming*, 2007.

[38] E. L. Huttlin, A. D. Hegeman, A. C. Harms, and M. R. Sussman, "Prediction of error associated with false-positive rate determination for peptide identification in large-scale proteomics experiments using a combined reverse and forward peptide sequence database strategy," *J Proteome Res*, vol. 6, no. 1, Jan 2007, pp. 392–8.

[39] E. F. L. T. M. H. G. H. S. J. C. Ian H. Witten, "Weka: Practical Machine Learning Tools and Techniques with Java Implementations," 1999.

[40] J. D. Jaffe, H. C. Berg, and G. M. Church, "Proteogenomic mapping as a complementary method to perform genome annotation," *Proteomics*, vol. 4, no. 1, Jan 2004, pp. 59–77.

[41] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein, "A Bayesian Networks Approach for Predicting Protein-Protein Interactions from Genomic Data," October 17, 2003.

[42] X. Jiang, X. Jiang, G. Han, M. Ye, and H. Zou, "Optimization of filtering criterion for SEQUEST database searching to improve proteome coverage in shotgun proteomics," *BMC Bioinformatics*, vol. 8, no. 323, 2007, p. 323.

[43] A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. W. Bruce Alberts, *Molecular Biology of the Cell*, 2007.

[44] M. P. Jose, B. Johan, rkegren, and T. Jesper, "Learning dynamic Bayesian network models via cross-validation," 2005.

[45] L. Kall, J. D. Canterbury, J. Weston, W. S. Noble, and M. J. MacCoss, "Semi-supervised learning for peptide identification from shotgun proteomics datasets," *Nat Methods*, vol. 4, no. 11, Nov 2007, pp. 923–5.

[46] L. Kall, J. D. Storey, M. J. MacCoss, and W. S. Noble, "Assigning significance to peptides identified by tandem mass spectrometry using decoy databases," *J Proteome Res*, vol. 7, no. 1, Jan 2008, pp. 29–34.

[47] D. E. Kalume, S. Peri, R. Reddy, J. Zhong, M. Okulate, N. Kumar, and A. Pandey, "Genome annotation of Anopheles gambiae using mass spectrometry-derived data," *BMC Genomics*, vol. 6, no. 128, 2005, p. 128.

[48] E. Kapp and F. Schutz, "Overview of tandem mass spectrometry (MS/MS) database search algorithms," *Curr Protoc Protein Sci*, vol. Chapter 25, no. 2, Aug 2007, p. Unit25 2.

[49] A. Keller, A. I. Nesvizhskii, E. Kolker, and R. Aebersold, "Empirical Statistical Model To Estimate the Accuracy of Peptide Identifications Made by MS/MS and Database Search," 2002.

[50] J. Klimek, J. S. Eddes, L. Hohmann, J. Jackson, A. Peterson, S. Letarte, P. R. Gafken, J. E. Katz, P. Mallick, H. Lee, A. Schmidt, R. Ossola, J. K. Eng, R. Aebersold, and D. B. Martin, "The Standard Protein Mix Database: A Diverse Data Set To Assist in the Production of Improved Peptide and Protein Identification Software Tools," 2008.

[51] E. Kolker, J. M. Hogan, R. Higdon, N. Kolker, E. Landorf, A. F. Yakunin, F. R. Collart, and G. van Belle, "Development of BIATECH-54 standard mixtures for assessment of protein identification and relative expression," *Proteomics*, vol. 7, no. 20, Oct 2007, pp. 3693–8.

[52] D. Kunec, B. Nanduri, and S. Burgess, "Experimental annotation of Channel Catfish Virus by probabilistic proteogenomic mapping," *Proteomics*, 2009.

[53] T. Larsen and A. Krogh, "EasyGene - a prokaryotic gene finder that ranks ORFs by statistical significance," *BMC Bioinformatics*, vol. 4, no. 1, 2003, p. 21.

[54] D. Lian, X. Lida, G. Feng, L. Jun, and Y. Baopin, "A local-density based spatial clustering algorithm with noise," *Inf. Syst.*, vol. 32, no. 7, 2007, pp. 978–986.

[55] A. J. Link, J. Eng, D. M. Schieltz, E. Carmack, G. J. Mize, D. R. Morris, B. M. Garvik, and r. Y. J. R., "Direct analysis of protein complexes using mass spectrometry," *Nat Biotechnol*, vol. 17, no. 7, Jul 1999, pp. 676–82.

[56] A. V. Lukashin and M. Borodovsky, "GeneMark.hmm: new solutions for gene finding," *Nucl. Acids Res.*, vol. 26, no. 4, February 15, 1998, pp. 1107–1115.

[57] M. Mann and A. Pandey, "Use of mass spectrometry-derived data to annotate nucleotide and protein sequence databases," *Trends Biochem Sci*, vol. 26, no. 1, Jan 2001, pp. 54–61.

[58] M. B. Markus, K. Hans-Peter, T. N. Raymond, J, and S. rg, "LOF: identifying density-based local outliers," 2000.

[59] F. M. McCarthy, S. M. Bridges, N. Wang, G. B. Magee, W. P. Williams, D. S. Luthe, and S. C. Burgess, "AgBase: a unified resource for functional analysis in agriculture," *Nucleic Acids Res*, vol. 35, no. Database issue, Jan 2007, pp. D599–603.

[60] F. M. McCarthy, A. M. Cooksey, N. Wang, S. M. Bridges, G. T. Pharr, and S. C. Burgess, "Modeling a whole organ using proteomics: the avian bursa of Fabricius," *Proteomics*, vol. 6, no. 9, may 2006, pp. 2759–71.

[61] F. M. McCarthy, N. Wang, G. B. Magee, B. Nanduri, M. L. Lawrence, E. B. Camon, D. G. Barrell, D. P. Hill, M. E. Dolan, W. P. Williams, D. S. Luthe, S. M. Bridges, and S. C. Burgess, "AgBase: a functional genomics resource for agriculture," *BMC Genomics*, vol. 7, no. 229, 2006, p. 229.

[62] P. Mereghetti, M. Ganadu, E. Papaleo, P. Fantucci, and L. De Gioia, "Validation of protein models by a neural network approach," 2008.

[63] G. E. Merrihew, C. Davis, B. Ewing, G. Williams, L. Kall, B. E. Frewen, W. S. Noble, P. Green, J. H. Thomas, and M. J. MacCoss, "Use of shotgun proteomics for the identification, confirmation, and correction of C. elegans gene annotations," *Genome Res*, vol. 18, no. 10, Oct 2008, pp. 1660–9.

[64] R. E. Moore, M. K. Young, and T. D. Lee, "Protein identification using a quadrupole ion trap mass spectrometer and SEQUEST database matching," *Curr Protoc Protein Sci*, vol. Chapter 16, no. 16, may 2001, p. Unit 16 10.

[65] R. E. Moore, M. K. Young, and T. D. Lee, "Qscore: an algorithm for evaluating SEQUEST database search results," *Journal of the American Society for Mass Spectrometry*, vol. 13, no. 4, 2002, pp. 378–386.

[66] B. Nanduri, M. L. Lawrence, S. Vanguri, and S. C. Burgess, "Proteomic analysis using an unfinished bacterial genome: the effects of subminimum inhibitory concentrations of antibiotics on Mannheimia haemolytica virulence factor expression," *Proteomics*, vol. 5, no. 18, Dec 2005, pp. 4852–63.

[67] A. I. Nesvizhskii, O. Vitek, and R. Aebersold, "Analysis and validation of proteomic data generated by tandem mass spectrometry," *Nat Methods*, vol. 4, no. 10, Oct 2007, pp. 787–97.

[68] M. L. Nielsen, M. M. Savitski, and R. A. Zubarev, "Improving Protein Identification Using Complementary Fragmentation Techniques in Fourier Transform Mass Spectrometry," *Mol Cell Proteomics*, vol. 4, no. 6, June 1, 2005, pp. 835–845.

[69] T. Nishi, T. Ikemura, and S. Kanaya, "GeneLook: a novel ab initio gene identification system suitable for automated annotation of prokaryotic sequences," *Gene*, vol. 346, Feb 14 2005, pp. 115–25.

[70] H. Y. Ou, F. B. Guo, and C. T. Zhang, "GS-Finder: a program to find bacterial gene start sites with a self-training method," *Int J Biochem Cell Biol*, vol. 36, no. 3, Mar 2004, pp. 535–44.

[71] D. N. Perkins, D. J. Pappin, D. M. Creasy, and J. S. Cottrell, "Probability-based protein identification by searching sequence databases using mass spectrometry data," *Electrophoresis*, vol. 20, no. 18, Dec 1999, pp. 3551–67.

[72] T. S. Price, M. B. Lucitt, W. Wu, D. J. Austin, A. Pizarro, A. K. Yocum, I. A. Blair, G. A. FitzGerald, and T. Grosser, "EBP, a Program for Protein Identification Using Multiple Tandem Mass Spectrometry Datasets," *Mol Cell Proteomics*, vol. 6, no. 3, March 1, 2007, pp. 527–536.

[73] W. J. Qian, T. Liu, M. E. Monroe, E. F. Strittmatter, J. M. Jacobs, L. J. Kangas, K. Petritis, n. C. D. G., and R. D. Smith, "Probability-based evaluation of peptide and protein identifications from tandem mass spectrometry and SEQUEST analysis: the human proteome," *J Proteome Res*, vol. 4, no. 1, Jan-Feb 2005, pp. 53–62.

[74] J. Razumovskaya, V. Olman, D. Xu, E. C. Uberbacher, N. C. VerBerkmoes, R. L. Hettich, and Y. Xu, "A computational method for assessing peptide- identification reliability in tandem mass spectrometry analysis with SEQUEST," *Proteomics*, vol. 4, no. 4, Apr 2004, pp. 961–9.

[75] R. G. Sadygov and J. R. Yates, "A Hypergeometric Probability Model for Protein Identification and Validation Using Tandem Mass Spectral Data and Protein Sequence Databases," *Anal. Chem.*, vol. 75, no. 15, 2003, pp. 3792–3798.

[76] T. Schiex, J. Gouzy, A. Moisan, and Y. de Oliveira, "FrameD: a flexible program for quality check and gene prediction in prokaryotic genomes and noisy matured eukaryotic sequences," *Nucl. Acids Res.*, vol. 31, no. 13, July 1, 2003, pp. 3738–3741.

[77] J. R. Sevinsky, B. J. Cargile, M. K. Bunger, F. Meng, N. A. Yates, R. C. Hendrickson, and J. J. L. Stephenson, "Whole Genome Searching with Shotgun Proteomic Data: Applications for Genome Annotation," *J. Proteome Res.*, vol. 7, no. 1, 2008, pp. 80–88.

[78] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, Search*, MIT Press, 2000.

[79] L. Stein, "Genome annotation: from sequence to biology," *Nat Rev Genet*, vol. 2, no. 7, Jul 2001, pp. 493–503.

[80] B. E. Suzek, M. D. Ermolaeva, M. Schreiber, and S. L. Salzberg, "A probabilistic method for identifying start codons in bacterial genomes," *Bioinformatics*, vol. 17, no. 12, Dec 2001, pp. 1123–30.

[81] S. Tanner, Z. Shen, J. Ng, L. Florea, R. Guigo, S. P. Briggs, and V. Bafna, "Improving gene annotation using peptide mass spectrometry," January 2, 2007.

[82] D. B. Weatherly, I. A. James A., T. A. Minning, C. Cavola, R. L. Tarleton, and R. Orlando, "A Heuristic Method for Assigning a False-discovery Rate for Protein Identifications from Mascot Database Search Results," *Mol Cell Proteomics*, vol. 4, no. 6, June 1, 2005, pp. 762–772.

[83] J. Webster and D. Oxley, "Peptide mass fingerprinting: protein identification using MALDI-TOF mass spectrometry," *Methods Mol Biol*, vol. 310, 2005, pp. 227–40.

[84] J. R. Yates, J. K. Eng, A. L. McCormack, and D. Schieltz, "Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database," *Anal Chem*, vol. 67, no. 8, Apr 15 1995, pp. 1426–36.

[85] H. O. M. I. H. T. Yoko Ishino, "Mass spectrometry-based prokaryote gene annotation," *PROTEOMICS*, vol. 7, no. 22, 2007, pp. 4053–4065.

[86] W. Zeng and G. Ping, "Normalized distance, similarity measure, inclusion measure and entropy of interval-valued fuzzy sets and their relationship," *Inf. Sci.*, vol. 178, no. 5, 2008, pp. 1334–1342.

[87] J. Zhang, J. Li, H. Xie, Y. Zhu, and F. He, "A new strategy to filter out false positive identifications of peptides in SEQUEST database search results," *Proteomics*, vol. 7, no. 22, Nov 2007, pp. 4036–44.

[88] J. Zhang, J. Ma, L. Dou, S. Wu, X. Qian, H. Xie, Y. Zhu, and F. He, "Bayesian Nonparametric Model for the Validation of Peptide Identification in Shotgun Proteomics," *Mol Cell Proteomics*, vol. 8, no. 3, March 2009, pp. 547–557.

[89] H. Zhu, G.-Q. Hu, Y.-F. Yang, J. Wang, and Z.-S. She, "MED: a new non-supervised gene prediction algorithm for bacterial and archaeal genomes," *BMC Bioinformatics*, vol. 8, no. 1, 2007, p. 97.