Fall 11-17-2018

Multidimensional Feature Engineering for Post-Translational Modification Prediction Problems

Norman Mapes Jr.

MULTIDIMENSIONAL FEATURE ENGINEERING FOR POST-TRANSLATIONAL MODIFICATION PREDICTION PROBLEMS

by

Norman "John" Mapes Jr. B.Sc.

A Dissertation Presented in Partial Fulfillment
of the Requirements of the Degree
Doctor of Philosophy

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

February 2019

LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

OCTOBER 8, 2018
Date

We hereby recommend that the dissertation prepared under our supervision by

Norman "John" Mapes Jr. B.Sc.

entitled **Multidimensional Feature Engineering For**
Post Translational Modification Prediction Problems

be accepted in partial fulfillment of the requirements for the Degree of

Doctor of Philosophy in Engineering with a Concentration in Cyberspace

Supervisor of Dissertation Research

Head of Department

Engineering
Department

Recommendation concurred in:

Advisory Committee

Approved:

Director of Graduate Studies

Dean of the College

Approved:

Dean of the Graduate School

ABSTRACT

Protein sequence data has been produced at an astounding speed. This creates an opportunity to characterize these proteins for the treatment of illness. A crucial characterization of proteins is their post translational modifications (PTM). There are 20 amino acids coded by DNA after coding (translation) nearly every protein is modified at an amino acid level. We focus on three specific PTMs. First is the bonding formed between two cysteine amino acids, thus introducing a loop to the straight chain of a protein. Second, we predict which cysteines can generally be modified (oxidized). Finally, we predict which lysine amino acids are modified by the active form of Vitamin B6 (PLP/pyridoxal-5-phosphate.) Our work aims to predict the PTM's from protein sequencing data. When available, we integrate other data sources to improve prediction.

Data mining finds patterns in data and uses these patterns to give a confidence score to unknown PTMs. There are many steps to data mining; however, our focus is on the feature engineering step i.e. the transforming of raw data into an intelligible form for a prediction algorithm. Our primary innovation is as follows: First, we created the Local Similarity Matrix (LSM), a description of the evolutionarily relatedness of a cysteine and its neighboring amino acids. This feature is taken two at a time and template matched to other cysteine pairs. If they are similar, then we give a high probability of it sharing the same bonding state. LSM is a three step algorithm, 1) a matrix of amino acid probabilities is created for each cysteine and its neighbors from an alignment. 2) We multiply the

square of the BLOSUM62 matrix diagonal to each of the corresponding amino acids. 3) We z-score normalize the matrix by row.

Next, we innovated the Residue Adjacency Matrix (RAM) for sequential and 3-D space (integration of protein coordinate data). This matrix describes cysteine's neighbors but at much greater distances than most algorithms. It is particularly effective at finding conserved residues that are further away while still remaining a compact description. More data than necessary incurs the curse of dimensionality. RAM runs in O(n) time, making it very useful for large datasets.

Finally, we produced the Windowed Alignment Scoring algorithm (WAS). This is a vector of protein window alignment bit scores. The alignments are one to all. Then we apply dimensionality reduction for gains in speed and performance. WAS uses the BLAST algorithm to align sequences within a window surrounding potential PTMs, in this case PLP attached to Lysine. In the case of WAS, we tried many alignment algorithms and used the approximation that BLAST provides to reduce computational time from months to days. The performances of different alignment algorithms did not vary significantly.

The applications of this work are many. It has been shown that cysteine bonding configurations play a critical role in the folding of proteins. Solving the protein folding problem will help us to find the solution to Alzheimer's disease that is due to a misfolding of the amyloid-beta protein. Cysteine oxidation has been shown to play a role in oxidative stress, a situation when free radicals become too abundant in the body. Oxidative stress leads to chronic illness such as diabetes, cancer, heart disease and Parkinson's. Lysine in concert with PLP catalyzes the aminotransferase reaction.

Research suggests that anti-cancer drugs will potentially selectively inhibit this reaction. Others have targeted this reaction for the treatment of epilepsy and addictions.

APPROVAL FOR SCHOLARLY DISSEMINATION

The author grants to the Prescott Memorial Library of Louisiana Tech University the right to reproduce, by appropriate methods, upon request, any or all portions of this Dissertation. It is understood that “proper request” consists of the agreement, on the part of the requesting party, that said reproduction is for his personal use and that subsequent reproduction will not occur without written approval of the author of this Dissertation. Further, any portions of the Dissertation used in books, papers, and other works must be appropriately referenced to this Dissertation.

Finally, the author of this Dissertation reserves the right to publish freely, in the literature, at any time, any or all portions of this Dissertation.

Author ___NORMAN "JOHN" MAPES JR.____

Date _____OCTOBER 8, 2018______________________

DEDICATION

This dissertation is written for those who love to learn. In memory of Dr. Tom Higginbotham and his wonderful wife Kathleen who have always been encouraging so many to pursue their dreams in higher education.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Special thanks to my supportive wife Monique and my father, Norman. The best comes from our Lord Jesus. I never could have achieved this dream without Dr. Sumeet Dua, Dr. Jim Palmer and Dr. Pradeep Chowriappa believing in me. Special thanks to Chris Rodriguez. I also would like to thank Dr. Jean Gourd, Dr. Weizhong Dai, Dr. Tom Bishop, Dr. Galen Turner and Dr. Ben Choi and all the faculty at Louisiana Tech University for their wonderful courses that helped me in my studies. I especially appreciated all of the professors' ability to explain difficult concepts so well. I'd also like to thank all the staff and students that have helped me in ways that differed but were so helpful, especially Rodney "Roman" Bozeman, Hatwib Mugasa, Richard Appiah and Marsha Smith.

CHAPTER 1

INTRODUCTION

1.1 Overview of Dissertation and Organization

The common thread in all three of the major sections, Chapters 2, 3 and 4, is original contributions in feature engineering, innovative approaches to data mining, predicting post-translational modifications of amino acids in a protein, dataset curation or creation and overcoming computational challenges.

Chapter one is contains important background information, definitions and explanations of the techniques. Chapter 2 is the feature engineering process as applied to the cysteine bonding problem using the Local Similarity Matrix. Chapter 3 is concerned with the cysteine oxidation prediction problem where we present the Residue Adjacency Matrix. Chapter 4 is using Windowed Alignment Scoring based feature engineering on the lysine-PLP PTM prediction problem. Chapter 5 covers the novel contributions and directions for future research.

1.1.1 Local Similarity Matrix Based Feature Engineering

Domain knowledge of evolutionary patterns in proteins was used to formulate a new source of features. There are three major steps in this process. 1) The probability of each residue occurring in a window surrounding cysteine was recorded in a matrix. The probability tables were generated by creating a list of proteins that have evolved in different organisms yet are sufficiently conserved. 2) The two matrices are joined (augmented) to describe the bonding state of a cysteine pair. The number of features is 2*20*(2*k+1). There are two tables joined, twenty amino acids, 2 halves of the window, k residues to the left or right and one cysteine. We multiply the square of the BLOSUM62 matrix diagonal to each of the corresponding amino acids (both matrices have entries for specific amino acids). 3) Finally, we row z-score normalize the augmented matrix.

1.1.2 Residue Adjacency Matrix Based Feature Engineering

The domain knowledge that cysteine distances to neighboring amino acids was used to generate a source of features. In particular, the amino acids, cysteine and tryptophan, are conserved throughout evolution. The distances of these conserved amino acids and others were recorded in a matrix. Scalability was linear with the number of data points in a dataset for sequential distances. Then we integrated 3-D coordinate data from the Protein Data Bank and used MODELLER to create coordinate files if they did not exist. Using these coordinate files, we extracted the Euclidean distance of cysteine to each of the remaining amino acids. The number of features is compact. This is very useful for large scale predictions such as phosphorylation of serine, threonine and tyrosine on the Uniprot/Trembl Dataset.

1.1.3 Windowed Alignment Scoring Based Feature Engineering

This algorithm has four major steps, as applied to the lysine-PLP PTM. 1) We cluster the proteins extracted from the Swiss-Prot database to the desired level of homology. 2) We window each lysine. This is accomplished by extracting the residues neighboring the lysine into a file. This process is repeated for the entire protein until all the lysines have been windowed. We repeat this process for each protein in the dataset. 3) We generate a vector of bit score alignments by aligning each file to all the remaining files. If the alignments do not have sufficient similarity, a zero is returned. 4) We reduce the dimensionality of all the vectors.

1.1.4 Innovative Approaches to Data Mining

We took a "standing on the shoulder of giants" approach to data mining to construct methods by integrating prior works as much as possible. Howeve,r when they were not sufficient we created our own tools. For example, we drew on the Position Specific Scoring Matrix (PSSM) for many of our original ideas, but when we noticed that they have been used to do most of the heavy lifting of predicting PTM's in the last 30 years, we modified it with an entirely new approach. Notably we discarded the inverse amino acid frequency, incorporated conservation of amino acids. After all that is what PSSM was designed for (See section 2.4.1). We continued to innovate as much as possible and wrote several thousand lines of code for data mining frameworks and have made them available for others to reproduce our work.

1.1.5 Post Translational Modifications of Amino Acids in Protein

Our focus is on two amino acids, cysteine and lysine. Cysteine can bond to other cysteines creating a disulfide bridge that bonds the two cysteines together. Cysteine can undergo oxidation to form new molecules. Lysine is found in enzymes that catalyze the transamination reaction using PLP. These PTMs do not show up in sequencing data and need expensive and time consuming methods such as NMR, X-ray crystallography and tandem mass spectrometry. Unfortunately, many of the results are of highly similar proteins. We hope our work will help to redirect the flow of wet-lab techniques to more distant proteins and let our data mining approaches find PTMs for highly similar proteins, and even some distantly related ones.

1.1.6 Dataset Curation or Creation

When available, we used datasets that were referenced in reputable publications so that we could make fair comparisons of our techniques to previous works. When unavailable or an indepedent test dataset was needed, we created it from quality repositories such as the Swiss-Prot database. See Sections 2.2, 3.3.1 and 4.2.3.

1.1.7 Computational Challenges and Approximations

There were many computational challenges to our work. Oftentimes, experiments required days to months to complete and making changes was time consuming. To alleviate these obstacles, we would often draw on the expertise of others and incorporate multi-processing approaches. When these were not sufficient to overcome our time constraints, we would incorporate an approximation, especially if the performance did not degrade significantly (see 4.3.2 and 4.5 for a salient example). Our hope is that others may find these approximations on challenging problems and use them as an opportunity to improve their approach.

1.2 Statistical Methods

1.2.1 Paired T-Test

We must first establish a null hypothesis and alternative hypothesis H_0 and $H_{1,}$ respectively. The goal is to reject or fail to reject the null hypothesis. The values for mu are the means of the difference from each group taken a pair at a time:

$$H_0: \mu_d = 0$$ **Eq. 1-1**

$$H_1: \mu_d \neq 0$$ **Eq. 1-2**

The test statistic for our hypothesis is t_0 and p-values can be found in the appendix of [1]. n is the number of samples in each group. y is the measurement in group one or two at the j'th observation:

$$t_{0=}\frac{\bar{d}}{S_d/\sqrt{n}}$$ **Eq. 1-3**

$$\bar{d} = \frac{1}{n}\sum_{j=1}^{n} d_j$$ **Eq. 1-4**

$$d_j = \ y_{1j} - y_{2j}$$ **Eq. 1-5**

$$S_d = \left[\frac{\sum_{j=1}^{n}(d_j - \bar{d})^2}{n-1}\right]^{1/2}$$ **Eq. 1-6**

This statistic is crucial for determining if a method is significantly different than another method on a collection of datasets. It is also useful for showing that a computational approximation is not significantly different and therefore appropriate.

1.2.2 Two Sample Kolmogorov Smirnov Test

The two sample KS test is a non-parametric test used for determining if two empirical cumulative distribution functions (F_1 and F_2) are from the same distribution. The test statistic is D_n as follows:

$$D_n = \underset{x}{supremum} \, |F_1(x) - F_2(x)|$$ **Eq. 1-7**

The p-values and distribution can be obtained from the R software stats package. This test can be used to predict if the correlations of an engineered feature to a class label is the same as another engineered feature.

1.3 Data Mining Definitions

1.3.1 Data mining and Feature Engineering

Data mining is a misnomer, in the same way that gold mining would be called dirt mining. Data is the substrate from which patterns are extracted. The process involves selecting and integrating data into a set of parameters that each uniquely describe the data point. Next, feature engineering transforms the data into a meaningful representation for the classification, regression or clustering algorithm (1.3.2,1.3.3,1.3.4) (Classification, regression and clustering are pattern recognition techniques). Feature engineering transformations include cleaning, normalizing, organizing and applying algorithms to the data. In our work, we compute probabilities from sequence alignments, perform matrix operations on extracted probabilities, window sequences, calculate Euclidean distances from sequences and 3-D coordinates, row normalize, reduce dimensionality and perform conditional operations on selected features. Classification, regression and clustering all need to be validated. A simple method of holding out a portion of the data for training and another portion for testing may be utilized or a slightly more complicated and immensely useful n-fold cross validation (see 1.3.5). Following validation of classification, regression and clustering a metric of performance is assigned such as the ratio of the sum of squares for "between" and sum of squares of "total" for clustering, accuracy for classification and R^2 for regression. More complicated and very important methods are described in section 1.4.

1.3.2 Classification

Classification takes each known data point's features and class label to generate a model that describes each of the unknown data point's labels. For example, cysteines may oxidize or not, corresponding to a 1 or 0 class label. We can then take the features such as a Residue Adjacency Matrix to describe each data point. A model will be created to predict which cysteines oxidize that were not trained on in the model. Often, it is useful to generate a confidence score for each unknown cysteine either for human interpretation or for more advanced metrics of success.

1.3.3 Clustering

Clustering is an unsupervised learning approach to data. Given a set of features without class labels, the algorithm will construct class labels for each data point. For example, clustering can be used to generate a class label for a set of proteins that have lysine-PLP PTM(s). We can specify the percent homology of our clusters and return one protein from each cluster. This is effective at reducing homology bias, because if there are proteins too similar in the data set, then a fair comparison of the data mining framework to other data mining frameworks is not possible. Highly similar datasets will show an increase in performance while low homology datasets will have lower performance with all other factors being equal.

1.3.4 Regression

Regression takes a set of data points and operates on the independent variables to predict the dependent variable. We use regression to predict the probability of a pair of cysteines bonding using the Local Similarity Matrix as the independent variables. In the case of random forest regression, we attempt to find the most important variable that will describe the class label and create a binary tree at a split that allows the independent variable to predict the dependent variable best. We repeat this process until a termination criteria is met. Predicting new points is done by following the tree from root to leaf in a series of if-else statements, then averaging all the trees together. This is an involved algorithm and is detailed in 2.3.5.

1.3.5 Validation (Specifically N-Fold Cross Validation)

. Validation can be done by holding out a percentage of data for prediction while using the remaining data for "training". Cross validation is a more sophisticated technique that we employ, the most common form being 10-fold. For the case of ten-folds, we find patterns in 90 percent of the data, then we predict on the remaining 10 percent. We repeat this process until all of the data has a prediction value. Finally, we compare the predicted value to the known value and "grade" our results using a performance metric such as accuracy. Often, accuracy is not enough. We need to describe our performance in terms of how many false positives, false negatives, true positives, true negatives there are and equations using the same. We take our confidence score (predicted value) and vary the threshold for classifying a point as one of the four possibilities (FP,FN,TP,TN) to generate a better description of our results.

1.3.6 Dimensionality Reduction

Dimensionality reduction is the process of taking a large number of features for a dataset and reducing the number of features. This technique can be useful to either speed the remaining data mining (less data) or for improving performance (curse of dimensionality) of the data mining technique. The curse of dimensionality refers to the fact that as the points increase the dimensions then so does the space between them. This results in a sparse dataset. Sparse datasets in effect have less combinations of values describing a class label or dependent variable making comparisons of one data point to another less meaningful and reducing the predictive power of the data mining.

1.4 Metrics of Performance

1.4.1 Receiver Operating Characteristic Curve (ROC Curve)

The ROC curve shows the effect of varying the threshold for generating a confusion matrix on the confidence scores output from a classifier for a binary classification problem. A confusion matrix contains the total number of true positives, true negatives, false positives and false negatives (TP,TN,FP,FN). Plotted on the y-axis is sensitivity and the x-axis is 1 - specificity. This conveniently shows that if most of the data points are classified in the 0 class, then you have a highly specific result. The corollary is that if the threshold is selected so that most points fall into the 1 class then you have a highly sensitive results. For example, a cancer test should be highly specific so that healthy patients do not undergo chemotherapy and surgery. On the other hand, an Ebola test should be highly sensitive because the consequences of one un-quarantined individual greatly increases the chances of the disease spreading. This ROC curve follows the left and top corner for a perfect test and is a diagonal line for random predictions. This makes sense because there is a tradeoff between sensitivity and specificity. By visualizing this, you can determine what the situation warrants and provide the desired level of either sensitivity or specificity and let the other one vary.

1.4.2 Area Under the ROC Curve (AUC)

There is a single number that describes the ROC curve. By taking the integral of the ROC curve, you can describe the chart regardless of what threshold is chosen. It is often important to have this single number for convenience of comparing two models against one another. AUC generally varies between 0.5 for random predictions and 1.0 for a perfect prediction. It is possible to get a number between 0 and 0.5, and if it is close to zero then you can reverse the predictions of your model. Although this rarely happens, it is a good diagnostic of a model building error.

1.4.3 Matthew's Correlation Coefficient

Matthew's Correlation Coefficient (MCC) is a single number that describes the confusion matrix much as AUC and the ROC graph. It varies between -1 and 1, with 0 being a model with no predictive power, 1 being perfect and -1 indicating the model should be reversed. MCC is a commonly reported statistic for the quality of a model. It is defined in 3.3.11.

1.4.4 Q_2 / Q_p / Q_c

These are the accuracy ratios used in Chapter 2. Q_2 is defined as the overall accuracy or the ratio of correct to total. Q_p is defined as the ratio of proteins whose bonding states are predicted with 100% accuracy to the number of proteins tested. Q_c is the ratio of bonds correct to the total number of bonds in the dataset.

.

CHAPTER 2

CYSTEINE DISULFIDE CONNECTIVITY AND THE LOCAL SIMILARITY MATRIX

2.1 Overview

Accurately predicting three-dimensional protein structures from sequences would present us with targets for drugs via molecular dynamics that would treat cancer, viral infections and neurological diseases. These treatments would have a far reaching impact to our economy, quality of life and society. The goal of this research was to build a data mining framework to predict cysteine connectivity in proteins from the sequence and oxidation state of cysteines. Accurately predicting the cysteine bonding configuration improves the TM-Score, a quantitative measurement of protein structure prediction accuracy. We provided state of the art Qp and Qc on the PDBCYS and IVD-54 Datasets. Furthermore, we have produced a Local Similarity Matrix that compares favorably to the default PSSMs generated from PSI-Blast in a statistically significant way. Our Qp for SP39, PDBCYS and IVD-54 were 90.6, 80.6 and 68.5, respectively.

2.2 Introduction

Protein folding errors cause cancer, heart disease and Alzheimer’s Disease [2, 3, 4]. Predicting how and why a protein arranges itself in three-dimensional space over time through molecular dynamics [5, 6] is crucial to understanding the diseases caused by aberrant folding and in turn their potential treatments. One of the most important amino acids for protein folding is cysteine. Cysteine residues form strong disulfide bonds with each other, causing the protein to impose rigid constraints on the folding. A disulfide bond is a covalent bond between sulfur atoms of two cysteines.

It is computationally challenging to predict the connectivity of cysteines in a protein due to the high order graph search that is detailed below. Our hypothesis is if we generate a local similarity matrix, then we will achieve higher scores than using the default PSSM generated by PSI-Blast. We aim to create a more effective method for predicting the cysteine disulfide bond pattern on the SP39, IVD-54, and PDBCYS benchmarks than exists currently in the literature for fewer than 6 bonds. The benchmarks were based on the publically available datasets. Some important proteins in these datasets are P05067 amyloid beta A4 protein for Alzheimer's disease and HIV protein P12506.

Disulfide bond prediction has several steps. First, it must be determined if the cysteine will even bond. Cysteines that form disulfide bonds with other cysteines are called oxidized cysteines and those that do not are called reduced cysteines. Secondly, it must be determined which of the oxidized cysteines will form pairs. If both of these predictions are correct, the known disulfide bonds can be a powerful indicator of the protein’s shape as evidenced by increased template modeling scores (TM-score) in [7]. TM-score is a measure of similarity between two proteins, the actual protein and the predicted model protein. The score is used to assess the quality of a model and is independent of protein length unlike a traditional root mean squared deviation (RMSD) measure. These measures are a metric of success different than specificity, sensitivity and accuracy in that the global topology of the proteins are measured for correct folding. A high level overview of the problem is seen in Figure 2-1.

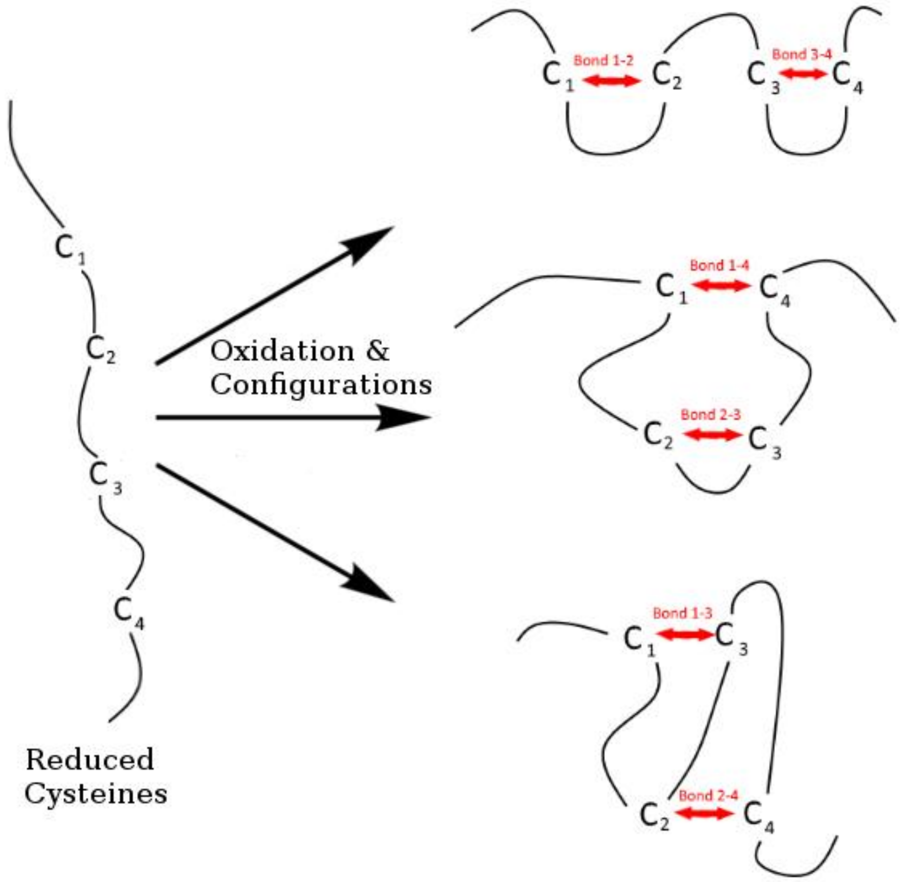


Figure 2-1: Protein Sequence with unbonded cysteines to potential bindings.

This study focuses on predicting cysteine bonding patterns once the oxidation state of the cysteines are known. Provided these parameters, we hope to develop a more accurate technique for connectivity prediction in order to improve the accuracy of existing programs like DiANNA [8] and Disulfind [9] that already handle the oxidized pair prediction.

Given the volume of protein sequences available in the post-genomics era, a data mining approach can be used to solve the three dimensional structure of proteins in order to create novel proteins, advance the treatment of disease by improved drug designs and lower the cost and time of performing X-ray crystallography and NMR. Furthermore, it is more efficient computationally than molecular dynamics simulations. Prior works have introduced the data mining approach to disulfide connectivity prediction in [10, 11, 12]. The methods used in this paper can be combined with other prediction methods to output more accurately a three-dimensional shape of a protein given that protein's amino acid sequence as seen by the constraints given to Quark to improve TM-Scores [7].

It is important to note that although we list many different sources and sizes of each feature, only 4 sources of features were found to be useful, the PSSM modified as Local Similarities LS, the distance of oxidized cysteines DOC, the angstrom distance provided by Modeller [13] and the cysteine separation profile, CSP [14], that was binary coded to provide 1 for divergence less than 4 and 0 for divergence greater than 4. The salient features of our data is seen in Table 2-1.

Table 2-1: Summary Description of Data

Dataset	**No. Proteins**	**Instances**	**Bonds**	**Imbalance**
SP39	446	7923	1371	4.8
PDBCYS-R	263	4688	804	4.8
IVD-54	54	386	146	1.6

2.2.1 Prior Works

P. Fariselli and R. Casadio [15] were among the first to determine disulfide connectivity computationally from protein sequences alone. In our survery of the literature, they appear to have established the 446 protein dataset, SP39, its four-fold cross validation and metrics of performance Qp and Qc. Furthermore, their work focused on predicting the edge connectivity with prior knowledge of the oxidation states of cysteines [16]. Determining the bonding state of cysteines was pioneered by Muskal *et al* in his work [17] using neural networks. This work was built upon in [18, 15, 19] by Fiser *et al.* Fariselli *et al*. and Fiser and Simon, respectively.

Next, A. Vullo and P. Fransconi introduced the recursive neural network, RNN, a connectionist model to work with the position-specific scoring matrix PSSM from PSI-Blast [20]. Vullo's model incorporated a patternwise search rather than pairwise. Pairwise being the predominant method in the literature is composed of two windows centered around designated cysteines and carries local information [31]. Patternwise carries global information and ranks alternative connectivity patterns but are limited by the availability of information because there are few bonding configurations available as the number of bonds increase [21]. CysView is a webserver that compares known annotated databases to the query sequence.

Distance of oxidized cysteines were incorporated into the PreCys pairwise SVM model in [21] by C.H. Tsai. DiANNA webserver was brought online that both predicted the oxidation state and the connectivity pattern using wmatch for Edmond-Gabow's Algorithm [22]. They utilized the now commonly used PSIPRED [23] software for their secondary structures. Likewise, the SCRATCH protein structure server was introduced with DIpro disulfide bridge prediction.

DISULFIND debuted in [9] with a SVR and bidirection recurrent neural network, BRNN, for the prediction of bonding state. Then the connectivity pattern is assigned to a score with a regression mode recurrent neural network rather than using Edmond-Gabow's Algorithm. B.J. Chen *et al.* [24] began using the normalized cysteine separation profile for SP39 and a two level framework that first assigns pairwise SVM, and then the second level uses patternwise SVM with the CSP.

J. Song *et al*. employed multiple sequence feature vectors to encode each cysteine pair in a pairwise manner using SVR and Edmond-Gabow's Algorithm [25]. Their work improved upon the SP39 Qp and Qc and laid the groundwork for the following efforts to improve cysteine connectivity prediction. C.H. Lu continued the work of J. Song by introducing a genetic algorithm and replacing Edmond-Gabow’s Algorithm with a connectivity matrix [26].

Disulfide connectivity prediction from protein sequences using Modeller was first used by H.H. Lin in his seminal conference paper. The metrics of performance Qc and Qp reached a record that still holds today for SP39. His work utilized the now common EG Algorithm and genetic algorithms for SVR tuning [27]. It is unclear if they limited their identity thresholds for sequence alignments that Modeller requires. Most notably, the search sequences themselves were included. D.J. Yu incorporated random forest into their regression algorithm. Using random forest regression is novel and also what we found to perform optimally [28].

2.3 Methodology

The key components of our methodology is feature extraction, normalization, regression, high order weighted graph matching and cross validation found in Figure 2. Together, these processes predict the final cysteine connectivity from the protein sequence and prior knowledge of the bonding state of the cysteines. Notable is the local similarities that we created. The cross validation n-folds are set to those found in prior works so that an objective comparison can be made. The flow chart of our process is illustrated in Figure 2-2.

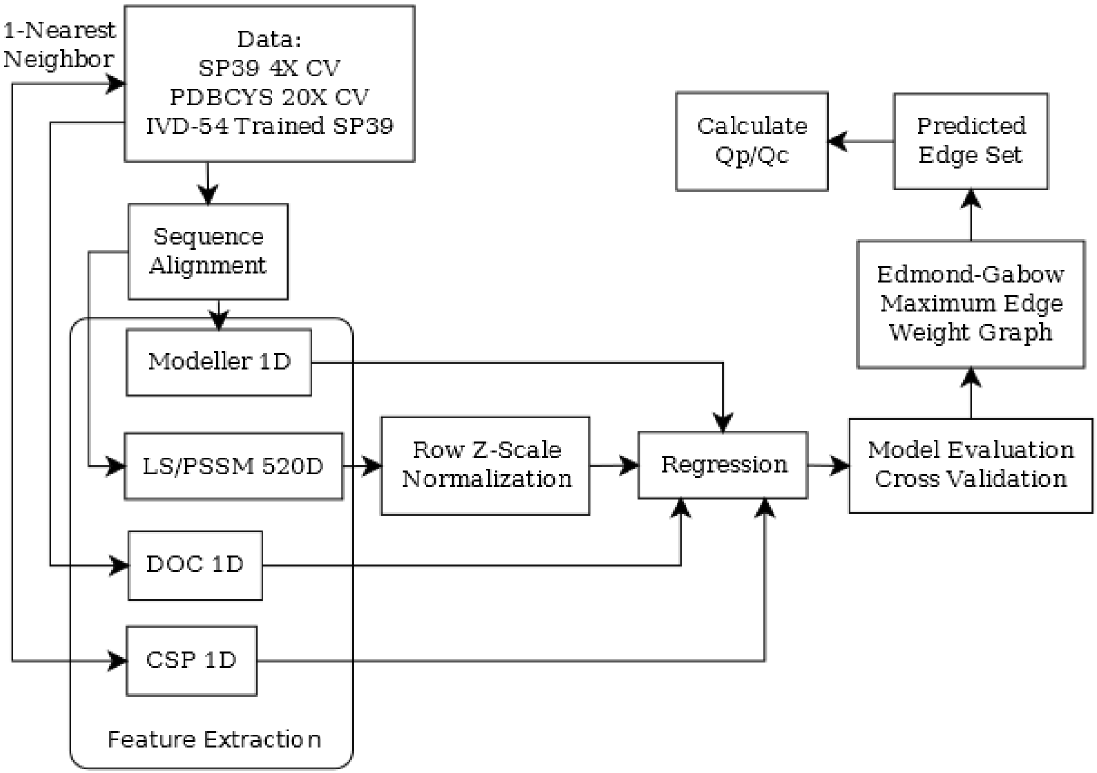

1-Nearest Neighbor
Data: SP39 4X CV PDBCYS 20X CV IVD-54 Trained SP39
Sequence Alignment
Modeller 1D
LS/PSSM 520D
DOC 1D
CSP 1D
Feature Extraction
Row Z-Scale Normalization
Regression
Model Evaluation Cross Validation
Edmond-Gabow Maximum Edge Weight Graph
Predicted Edge Set
Calculate Qp/Qc

Figure 2-2: Block diagram of process.

2.3.1 Feature Extraction

In the following sections, we describe the 523 dimensional vector's composition and its derivation from the data. We produce a 520D local similarity matrix, a 1D distance of oxidized cysteines, a 1D Modeller angstrom distance and a 1D cysteine separation profile.

2.3.1.1 *Position Specific Scoring Matrix PSSM vs. Local Similarities*

The local similarity matrix is obtained by calculating the probabilities of an amino acid occurring at a position relative to two possibly-bonding cysteines. BlastP sequence alignment was performed on the target sequence in order to find sequences that are similar to the target sequence with an E-value of less than 0.005. The returned sequences might have insertions (amino acids that occur in the returned sequences but not the target sequence) and omissions (amino acids that occur in the target sequence but not the returned sequence) as seen in Figure 2-3.

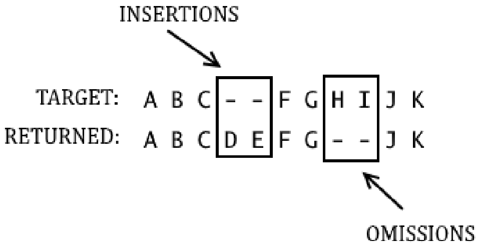
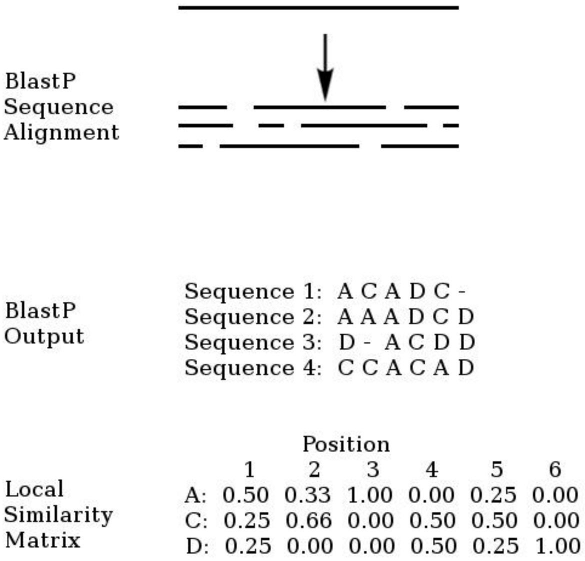

INSERTIONS
TARGET: A B C - - F G H I J K
RETURNED: A B C D E F G - - J K
OMISSIONS

Figure 2-3: Depiction of insertions and omissions.

Insertions are removed from the returned sequence, and if the alignment occurs at the termini of the original sequence, then the tails are padded to the left and right with dashes. The dashes are not counted when summing the occurrences of the amino acids for the probabilities in the Local Similarity Matrix. We then focus on the k amino acids neighboring a given cysteine. For our experiments, we chose a value of 6 for k. This equates to 13 positions: the 6 positions to the left of the cysteine, the 6 to the right of the cysteine, and the cysteine position itself. Including the position for the cysteine is important because BlastP may return a mutated amino acid instead of a cysteine. The frequency of each of the 20 amino acid occurrences at each of the 13 neighboring locations is calculated by summing the number of occurrences in all of the returned sequences. These frequencies are then converted to a probability by dividing the total number of sequences. If the returned sequence has an omission at a given position, it is not counted in the total number of sequences at that position. Figure 2-4 details the process of obtaining a table of probabilities.



```
Sequence 1:  A C A D C -
Sequence 2:  A A A D C D
Sequence 3:  D -  A C D D
Sequence 4:  C C A C A D
```

	Position					
	1	2	3	4	5	6
A:	0.50	0.33	1.00	0.00	0.25	0.00
C:	0.25	0.66	0.00	0.50	0.50	0.00
D:	0.25	0.00	0.00	0.50	0.25	1.00

Figure 2-4: Example PSSM or Local Similarity Matrix.

The final table of probabilities is a 2D matrix of 20 amino acids by 13 positions (260 elements). A table like this must be created to model the neighborhood of every oxidized cysteine. A row instance is created by combining the elements of two of these tables. The number of rows that are created for each input sequence is equal to the number of possible cysteine pairs (n choose 2). The number of possible pairs can be calculated by the following equation:

$$\binom{n}{2} = nC2 = \frac{n!}{(n-2)! \times 2!} \qquad \textbf{Eq. 2-1}$$

Where n is the number of cysteines in the sequence. A row of data has 520 columns (the result of concatenating the two 260 element tables together), which will be used for our features. The order of concatenations of the 260 dimensional row did not result in better scores. In other words, training on C1-C2 versus C2-C1 did not improve our Qp.

Also, the decision was made to use simple probabilities instead of applying a log transform to the probabilities as in Equation (2) where $b_k = 1/k$ and $k = 20$ which by definition is a PSSM. The PSSM stands in contrast to our Local Similarity (LS) Matrix as shown in Figure 2-4. Probabilities with ignored padding dashes, omissions and removed insertions defines the LS matrix:

$$\mathrm{PSSM(i,j)} = \log_2\left(\frac{\mathrm{P(i,j)}}{\mathrm{b_k}}\right) \qquad \textbf{Eq. 2-2}$$

We found the accuracy to be higher with the simple probabilities. When PSI BLAST is set to return a PSSM, it returns the log probabilities instead of the simple probabilities, so care must be taken when using those outputs from the program.

The difference in accuracy when using e-values greater than 0.005 was also negligible, but we did progressively increase the e-value to 100 for short sequences that did not return any similar sequences in order to obtain enough data for testing. Using the Swissprot database whose size was 197 megabytes uncompressed was as good as the

Trembl database whose size was 24 gigabytes. We attempted to use weighted averages using both the distance and inverse distance for the weights. Neither of these methods generalized well to increase the scores. Figures 2-5 and 2-6 show the calculation of PSSM and LSM from their variables RIF, PPM and Conservancy.

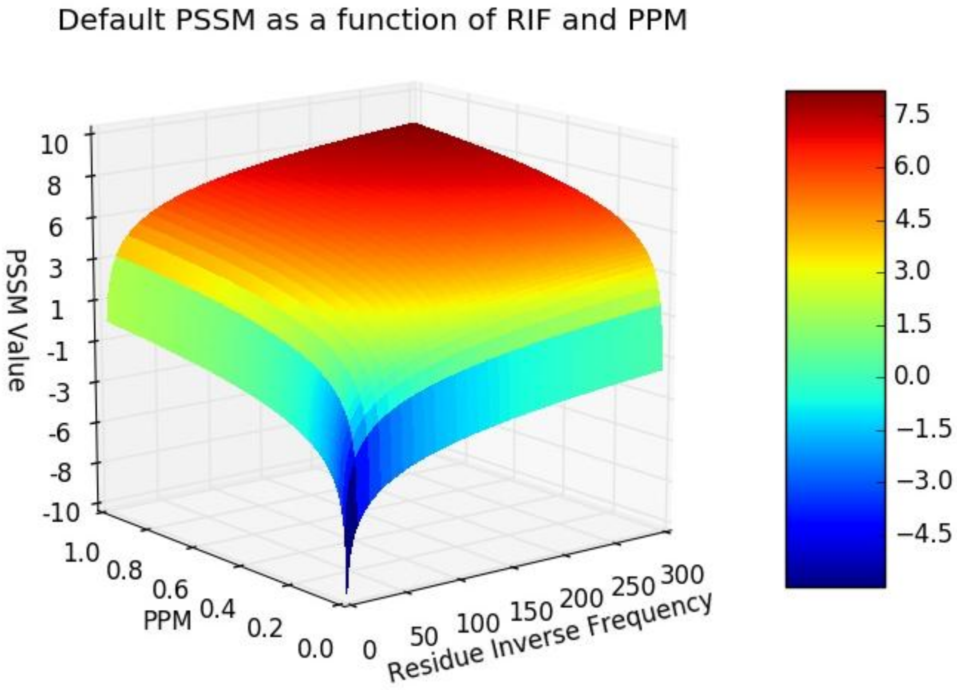


Figure 2-5: Analytic Solution for the PSSM.

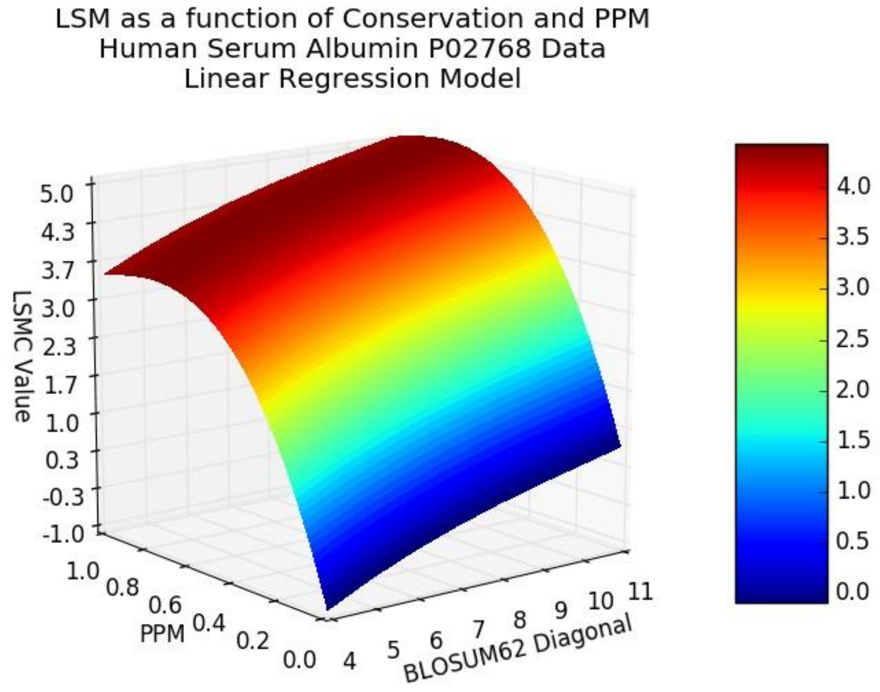


Figure 2-6: Linear Regression Approximation of Local Similarity Matrix with Conservation.

We incorporate a second variable into the local similarity matrix, conservation times the PPM value. Conservation is measured as the degree to which an amino acid is inversely substituted. This value is found on the diagonal of the BLOSUM62 matrix, a matrix used for BLAST alignment scoring. This variant is called the local similarity matrix with conservation or LSMC. This model performs better on the SP39 and PDBCYS-R datasets but was not confirmed by the IVD-54 dataset.

2.3.1.2 *Distance Oxidized Cysteines*

We can also use the one-dimensional distance of our two cysteines in the sequence (DOC) as yet another feature. We take the sequence index of our first cysteine as i and the sequence index of our second cysteine as j and then we find the distance between them via the absolute value of i-j.

2.3.1.3 *Cysteine Separation Profiles*

Cysteine separation profile used a divergence threshold of 4 for all data sets. The method was calculated per Zhao *et al.* in [14]. Cysteine separation profiles are defined for protein i, with n bonds, 2n cysteines C, and separation s:

$$\begin{aligned} CSP_i &= (s_1, s_2 \ldots\ s_{2n-1}) \\ &= (C_2 - C_1, C_3 - C_2 \ldots C_{2n} - C_{2n-1}) \end{aligned} \qquad \textbf{Eq. 2-3}$$

Then divergence, D, between two proteins i and j is calculated as:

$$D_{i,j} = \sum_k \left| s_k^i - s_k^j \right| \qquad \textbf{Eq. 2-4}$$

Finally, the one dimensional feature is a 1-NN (nearest neighbor) search to find the protein with a divergence less than 4 that is minimal. If it is found, then the feature is assigned for all bonds in protein i matching the bonds in protein j a 1 and 0, otherwise. In the event of a tie for divergence, then one protein is selected randomly. Figure 2-7 shows a histogram of CSP divergence as it relates to bonding and nonbonding proteins. Borrowing from the graph problem notation in Equations 2-10,2-11 and 2-12, we assign a 1 to the edges in protein i that intersect the edges in protein j and all other edges a 0:

$$\forall e \in E_i = \begin{cases} 1 & if \min\limits_{j, i \neq j} D_{i,j} < 4\, , E_i \cap E_j \\ 0 & otherwise \end{cases} \qquad \textbf{Eq. 2-5}$$

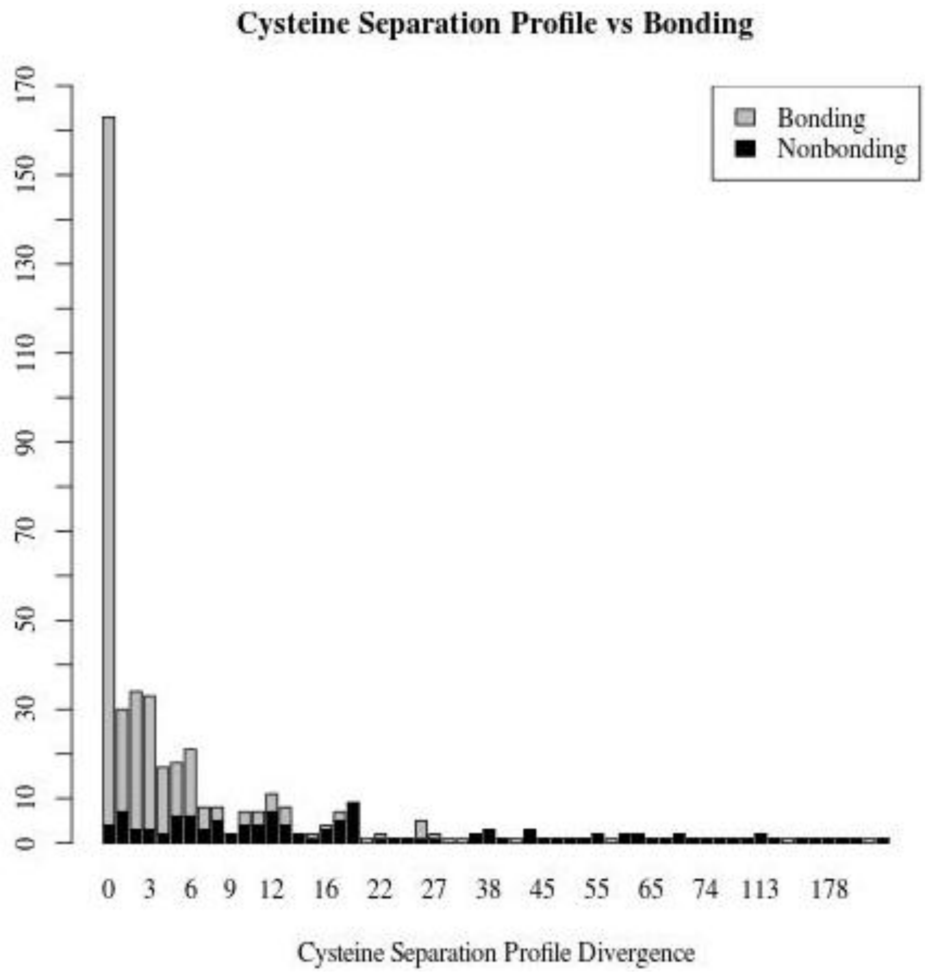


Figure 2-7: Histogram showing the number of proteins at each divergence separated by bonding and nonbonding for SP39. There is a high degree of homology because most of the proteins had low divergence and were bonding. This was not the case for PDBCYS and IVD-54. Qualitatively this shows the dataset has the potential to be solved at a higher Qp and Qc metric of success.

2.3.1.4 *Modeller*

Next, we use the Modeller software to extract the predicted 3D coordinates of the protein. As a result, we have a 1-D feature by taking the Euclidean distance between the two cysteines. Modeller produces the predicted 3D structure by using structures of proteins that are similar to the target protein and whose structures are known through X-ray crystallography and similar methods. The similarity is measured by identity output from a BlastP sequence alignment. The highest identities up to the thresholds were chosen for each protein. Once the tertiary structure is produced, we extract the 3D coordinates of the two possibly binding cysteines that are being modeled. We compute the Euclidean distance (12) between them and use this as a one dimensional feature:

$$\mathrm{Distance}(\mathrm{C1}, \mathrm{C2}) = \sqrt{(\mathrm{x}_1 - \mathrm{x}_2)^2 + (\mathrm{y}_1 - \mathrm{y}_2)^2 + (\mathrm{z}_1 - \mathrm{z}_2)^2} \qquad \textbf{Eq. 2-6}$$

Modeller requires an alignment of the target sequence, and this was selected as the ten alignments with the highest identity below the threshold. The thresholds are described in Table 2-7 and Figure 2-11, their values were 40%, 60%, 80% and 100%. 100% identity included all available data except the sequence itself. If the alignments did not satisfy the constraints that Modeller imposes we dropped the lowest identity alignment repeatedly. If all alignments were dropped, then the E-Value output by BlastP was increased and the process was repeated until Modeller's constraints were satisfied. The PDBAA database whose size was 25 megabytes compressed was used for alignments. Figure 2-8 illustrates the effect of the identity of the alignments and therefore the corresponding quality of the PDB's used to generate the feature.

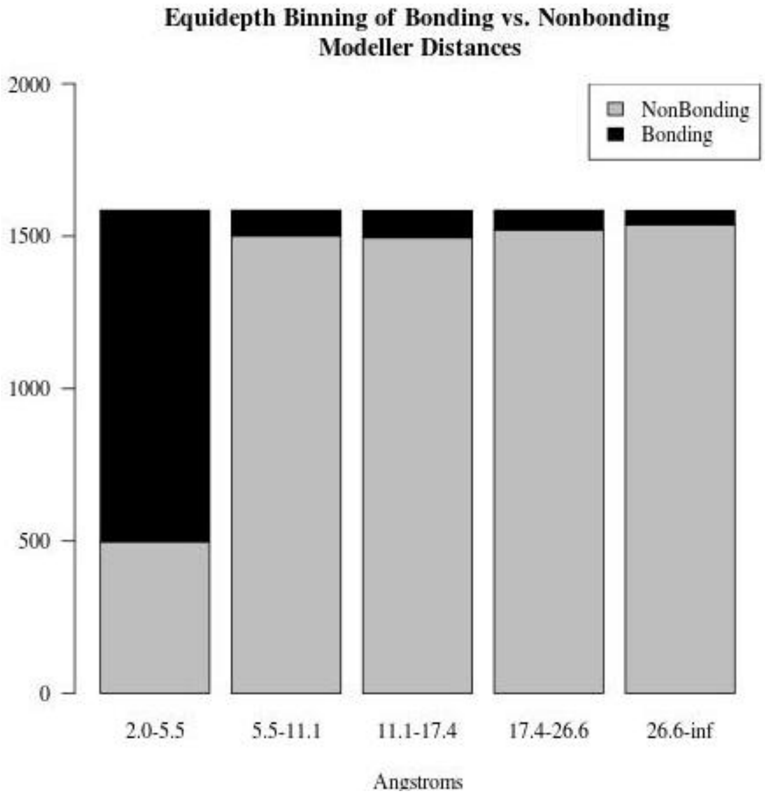


Figure 2-8: Equidepth binning of Modeller at 40 percent identity comparing bonding to nonbonding cysteines.

2.3.1.5 *PSS - Predicted Secondary Structure*

In order to improve accuracies, the data on the Predicted Secondary Structure (PSS) can also be a useful predictor. In the same way that data was produced from the frequency of amino acids occurring at positions relative to the cysteine, data can also be produced by the frequency of the secondary structure (alpha helix, beta sheet, or coil) which is for each amino acid position. This produces a probability table of the three possibilities for secondary structure by the 26 positions being viewed, as shown in Table 2-2. Compressed down to a single row of data, this gives us 78 more features to analyze for a k of 6. Although it has a Qp of nearly 60 by itself, it was not incorporated into the final model due to confounding results.

Table 2-2: Predicted Secondary Structure.

Cysteine	**1**	**1**	**1**	**1**	**2**	**2**	**2**
Position	**k-6**	**k-5**	**…**	**k+6**	**k-6**	**...**	**k+6**
Alpha Helix	0.98	0.68	…	0.28	0.2	…	0.45
Beta Sheet	0.01	0.02	…	0.4	0.7	…	0.15
Coil	0.01	0.4	…	0.32	0.1	…	0.4

2.3.2 Normalization

To improve the accuracy of our regression algorithms, the raw probabilities of each LS/PSSM can be converted into z-scores, with a high z-score indicating a strong probability of a particular amino acid occurring at its relative position:

$$\mathrm{Z-score(i)} = \frac{\mathrm{P(i)} - \overline{\mathrm{P_{row}}}}{\sigma_{\mathrm{row}}} \qquad \textbf{Eq. 2-7}$$

Equation (7) is the equation for z-score normalization, and note that the whole LS/PSSM data row for each of the proteins (520 elements) was used as the set for the z-score scaling instead of just the 20 amino acid probabilities for each position. Z-score normalization gave zero means and standard deviation of unity for each row instance of the local similarity 520 dimensional feature.

Various normalization attempts were made to the Local Similarity Matrix, most notably sigmoid and min-max normalization. These did not result in a higher Qp or Qc. Whem log transformations were also implemented, the scores did not improve with this method.

2.3.3 Regression

We formulate the regression problem as follows:

$$D_i = (y_i x_i)\ , y_i \in \{0,1\}, x_i \in \mathbb{R}^{523} \\ \text{i = 1,2,..., |instances|} \qquad \textbf{Eq. 2-8}$$

Here y_i is selected to be 1 for a bonded data row i and 0 for a nonbonding data row i. Each data row consists of a 520 dimensional LS/PSSM that specifies 260 = (20*13) floating point numbers for each cysteine. One dimension for each of two cysteines distance in the primary structure DOC, angstrom distance in the tertiary structure from Modeller, and the cysteine separation profile divergence nearest neighbor's bonding state. We included secondary structure helix, sheets and coils and taken alone were valuable in predicting the correct edge set but did not improve the results as a fifth source of features.

An ensemble of regressors was used to improve the scores. The ensemble consisted of a layered approach where all the regressors were trained and their score was used as a feature to a final regressor. For instance, we had used the 523D data for each of the 7 regressors. These 7 regressors would then output one feature each. These 7 features were then used by a final regressor to output the final score that was used by the Edmond Gabow Maximum Weight Matching Algorithm. We tried each of the seven regressors as the final regression. Unfortunately, the ensemble approach was found not to be effective. Furthermore, unsupervised K-means clustering was used on the datasets with a sparse coded class label prior to inputting to the regressors and was found to have no effect.

For cross validation, we considered each vertex set independently and breaks were chosen at the closest protein. Otherwise, there would be mixing of the edge sets across the validation sets.

2.3.4 Maximum Edge Weight Perfect Matching and Performance Metrics

The bonding of cysteines can be reduced to a maximum edge weight perfect matching graph problem where the cysteines are vertices in the graph, the potential bonds among them are the edges, and the likelihood of these bonds are the edge weights. To predict which oxidized cysteines will bond with each other, we first must determine a list of all possible bond patterns by running through all combinations of two element groupings in the oxidized cysteine list, $n\ choose\ 2$, $\binom{n}{2}$, possibilities exist where n is the number of vertices or oxidized cysteines. The bond pattern is a list of cysteine bonding pair tuples that represents how all cysteines in a sequence are bonded. Because every cysteine must bond with exactly one other cysteine, there are n/2 bonding pair tuples in a bond pattern (where n is the number of oxidized cysteines in the sequence). The number of possible bonding patterns is calculated via (n-1)!! where !! represents the factorial of the odd integers and n is still the number of oxidized cysteines. Figure 2-9 shows an example of all possible bond patterns (three) for a protein sequence with four cysteines. For only 10 cysteines, there are five bonding pairs and 9×7×5×3×1 or 945 possible bonding patterns:

$$|\text{Patterns}| = \prod_{0 < i \leq B} 2\text{i} - 1 \qquad \textbf{Eq. 2-9}$$

Randomly guessing the correct bond pattern for 4 cysteines is 33% likely. Randomly guessing the correct bond pattern for 10 cysteines has a probability of 1/945 or about 0.1%. In order to make accurate predictions, each bonding pair obtained in the previous section is run through a regression model to obtain a real number score. Recall that each instance represents the bond between two cysteines, so the score returned can be thought of the likelihood of that bond occurring. We sum the scores for each bond in a bond pattern in order to get the total score for that bond pattern. The bonds that are part of the bond pattern with the highest score are then chosen as the predicted bonds.

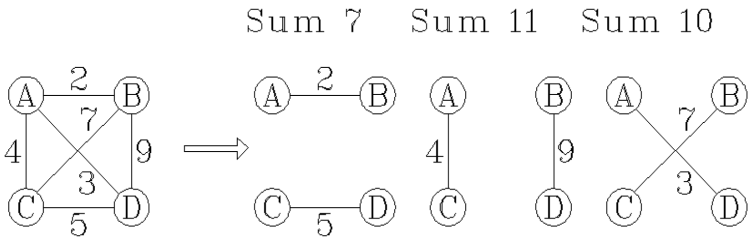


Figure 2-9: Maximum Edge Weight Perfect Matching Graph Combination Problem, Combination 2, A-C and B-D have the highest sum, so this pairing is maximum and would be chosen.

Gabow-Edmond's Maximum Edge Weight Perfect Matching algorithm [8, 29, 30] was used. The algorithms's worst case computational complexity is bounded at O(v3) where v is the number of vertices or oxidized cysteines for a particular protein. Figure 2-9 illustrates the problem that is solved by Gabow-Edmund's algorithm.

Formulating the dataset as a graph theory problem, there is a dataset D that contains the actual and predicted (*) undirected graphs:

$$G_i = (V_i, E_i) \text{ and } G_i^* = (V_i, E_i^*) \qquad \textbf{Eq. 2-10}$$

where V_i is the vertex set (cysteines) of G_i and E_i is the connectivity pattern of G_i. The predicted edge set E_i* contains the connectivity pattern output from regression and Gabow-Edmond Algorithm. Performance measures are formally computed as follows:

$$Q_c = \frac{\sum_D |E_i \cap {E_i}^*|}{\sum_D |E_i|} * 100\% \qquad \textbf{Eq. 2-11}$$

$$Q_p = \frac{\sum_D |E_i = E_i^*|}{|D|} * 100\% \qquad \textbf{Eq. 2-12}$$

Here $E_i = E_i^*$ is 1 if the sets are exactly identical and 0 if the sets are not exactly equal. E_i^* is computed from edge e, belonging to all possible edge sets ε whose cardinality is Equation (9). Edmonds-Gabow is used to calculate (13) by completing the maximum weight matching problem:

$$E_i^* = \max_{e \in \varepsilon} Score(e) \qquad \textbf{Eq. 2-13}$$

The regressor R is used in (14):

$$Score(e) = \sum_e R(e_i) \qquad \textbf{Eq. 2-14}$$

Together, Equations (13,14) specify the final predicted disulfide connectivity pattern. Equations (11,12) are the performance metrics that are used in place of specificity, sensitivity or accuracy because they prioritize the bonding state rather than the accuracy as a whole that would include the nonbonding state. The nonbonding state is class imbalanced and Qc would be high if nonbonding predictions were included.

The cardinality of any E is equal to B, the cardinality of any V is equal to 2B and the degree(v) = 1 for any v∈V, thus perfect matching, where B is the number of bonds in a protein and 2B is the number of oxidized cysteines.

2.3.5 Random Forest Regression

In our experiments, we used the bagging approach , where each tree is constructed using a bootstrap sample of the data and the output is an average of all regression trees output. This is in contrast to the boosting approach where successive trees depend upon earlier trees [31]. Breiman introduced an extra layer of randomness to the bagging. Each tree is constructed using a subset of the features whose cardinality is mtry or max_features in sklearn (our setting was 20). A standard regression tree uses the entire set of features to find the best split at each node [32].

The expression to maximize in a random forest regressor [33] at each split is:

$$\frac{S_L^2}{n_L} + \frac{S_R^2}{n_R} \ where \ S_L = \sum_{D_L} y_i \ and \ S_R = \sum_{D_R} y_i \qquad \textbf{Eq. 2-15}$$

Where nL and nR are the number of data points to the left of the split and the right of the split, respectively. DL and DR are the data points that lie to the left and right of the split as well. This is iteratively solved by the following dynamic programming algorithm in Figure 2-10:

Algorithm: Splitting continuous features for regression using if − else logic
Input: Real-valued N× P data matrix X.
Output: The next if-else rule on which the trees of a regression forest are built, also providing feature importance scores implicitly.

```
function BestSplit(X,y)
best ← 0;
for Column in X
    Sr ← sum(y); Sl ← 0; nR ← length(y); nL ← 0
    Sort Column and y by Column
    for Xi,yi in sorted data
        Sl ← Sl + yi; Sr ← Sr-yi
        nR ← nR-1; nL ← nL+1
        if Xi != Xi+1:
            split ← Sl2/nL+Sr2/nR
            if split > best
                best ← split
                cut ← (Xi+Xi+1)/2
Split X and y according to cut into XL, XR, yL, yR
Set Variable LeftTermination, RightTermination
if LeftTermination
    Leaf(yL)
else
    BestSplit(XL,yL)
if RightTermination
    Leaf(yR)
else
    BestSplit(XR,yR)
```

Figure 2-10: Algorithm for generating splits for the trees in a random forest regressor for continuous valued features.

The columns are the parameters and the rows are the data instances. The recursive nature of BestSplit is that it calls itself until all the nodes are pure or their termination criteria have been met. Once terminated, the leaf nodes are created. Creating the tree itself is a matter of tracking the nodes created and has been omitted to keep the algorithm clear and concise. To implement the bagging, one must randomly select a subset of the data (bootstrapped) and then enter the function BestSplit. To create multiple trees in a forest, it is necessary to call BestSplit on the bootstrapped data once for each estimator desired. The mtry parameter was not included in the algorithm, but basically it is a simple limit to the number of columns (features) explored for each BestSplit run. After training the trees and then the output, h, from the random forest, the regression model is the unweighted average of all the estimators T with individual trees t:

$$h(X) = \frac{1}{T}\sum_{i=1}^{|T|} t_i(X) \qquad \textbf{Eq. 2-16}$$

2.4 Results

2.4.1 Local Similarity vs. PSSM

Modeller's identity threshold was set at 40 percent to show the effect of Local Similarity vs. PSSM. Higher Modeller thresholds brought the Qp above 90 and the difference was still present but the difference was not as pronounced. Table 2-3 shows the results of our experiment

Table 2-3: Local Similarity Performance.

Dataset	Method	B=2		B=3		B=4		B=5		B=2-5	
		Qp	Qc	Qp	Qc	Qp	Qc	Qp	Qc	Qp	Qc
SP39	Default PSSM	83.3	83.3	70.5	78.3	76.8	84.3	51.1	64.9	74.4	79.0
SP39	Local Similarity	91.7	91.7	82.2	86.1	78.8	85.6	60.0	73.8	82.5	85.2
PDBCYS	Default PSSM	75.0	75.0	55.3	66.7	43.9	61.6	16.2	44.3	55.5	62.6
PDBCYS	Local Similarity	82.0	82.0	55.3	65.9	58.5	70.1	24.3	54.6	61.6	68.2
IVD-54	Default PSSM	65.5	65.5	66.7	73.3	0.0	35.7	33.3	46.7	55.6	60.3
IVD-54	Local Similarity	79.3	79.3	80	82.2	14.3	32.1	33.3	53.3	68.5	68.5

The differences of each dataset's treatment was then run through statistical analysis software and found to be statistically significant with a p-value of 0.0041 for a one-sided paired t-test. The mean difference for the default PSSM vs. Local Similarity was 7.47 points Qp. The 95 percent confidence interval was in the range of 5.47 to infinity. Figure 2-11 shows a comparison of LSM vs PSSM.

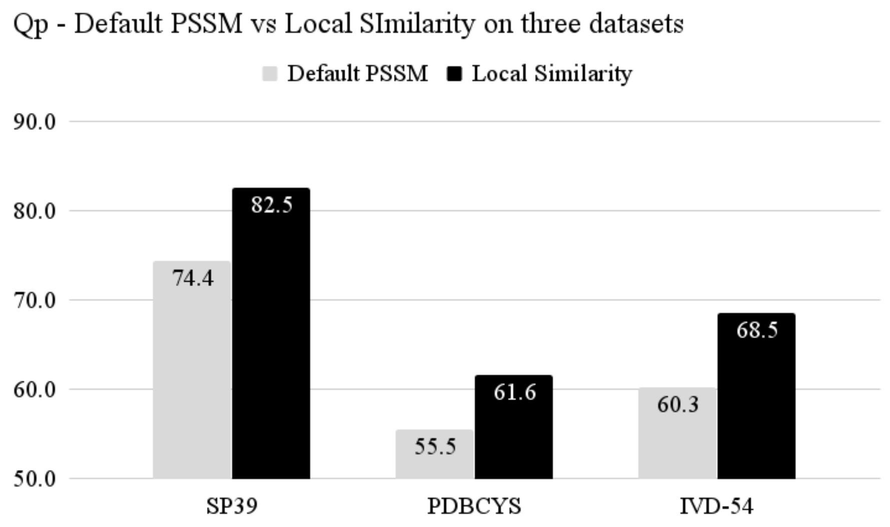


Figure 2-11: The differences between PSI-Blast's PSSM and Local Similarities at the 40% identity threshold for all three datasets.

2.4.2 Performance on three Datasets

We ran the amino acid sequences of the SP39, PDBCYS-R, and IVD-54 datasets through the process described in the methodology section to obtain a table of features that various regression algorithms could utilize. We used Support Vector Regression from both the R and Python libraries, Random Forest Regression in both R and Python, K Nearest Neighbor Regression, Neural Network, LassoCV, Ridge Regression and Bayesian Ridge from Sklearn. A 4-fold cross-validation was used with the SP39 dataset, 20-fold cross validation was used with the PDBCYS-R dataset, and the models that ran on the IVD-54 dataset were trained on the SP39 dataset. The Qp accuracy is the percentage of complete bond patterns that the model predicted correctly. The Qc accuracy is the percentage of cysteine bond pairs that the model predicted correctly out of all the positive (cysteine bonding) instances (Equations 11,12). For a sequence with only four oxidized cysteines (two bridges, B = 2), Qp is equal to Qc because if the model predicts the right bond pattern. The bond pairs must be correct, and if the model predicts the wrong bond pattern, there is no possible way for any of the predicted bond pairs to be correct, as no pair is shared between the three bond patterns of a sequence with four cysteines. Qc is greater than Qp for sequences with more than four oxidized cysteines because, if the pattern is predicted correctly, all bond pairs are predicted correctly, and if the pattern is incorrect, that pattern may still happen to contain bond pairs that are correct. The overall results are listed in Figure 2-12.

The Random Forest parameters were 500 trees and a maximum of 20 features per split were utilized. SVR utilized a cost of 5 and a gamma of 0.005 using the radial basis function kernel exp(-gamma*|u-v|2). LassoCV utilized default parameter settings. Neural networks had 5,000 hidden units. KNN regression was weighted by inverse distance and the number of neighbors varied on the datasets from 5 to 30 neighbors. Ridge Regression utilized an alpha of 1000 and Bayesian Ridge was set to default parameters. Feature selection was implemented but did not have a beneficial effect likely due to Random Forest Regression selecting the best features in the algorithm itself. Restricted Boltzmann Machines and Principal Component Analysis were used to generate additional features as well as stand alone inputs. This information was input to the regression models and was not found to improve Qp or Qc. The results of our experiment with different regressors is shown in Figure 2-12, Table 2-4, 2-5 and 2-6.

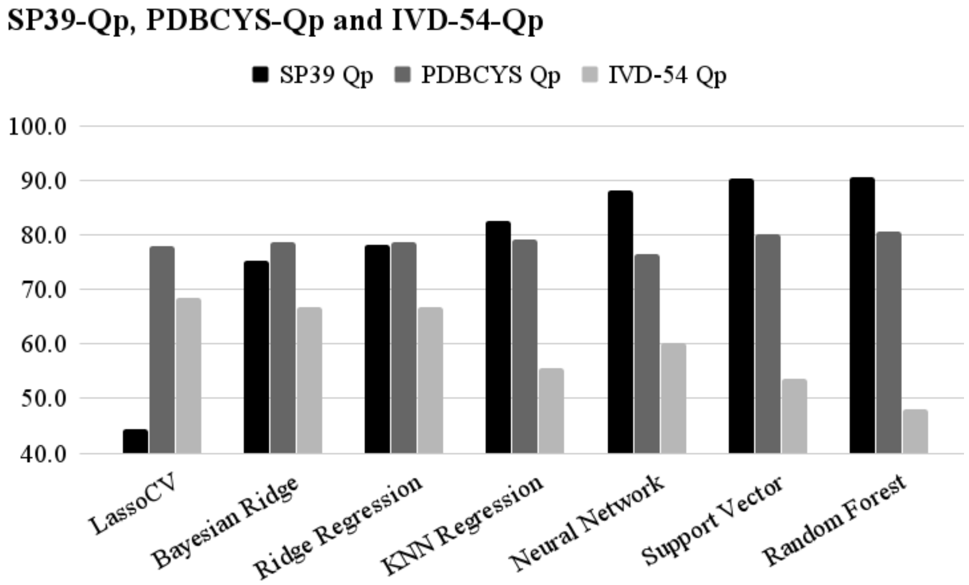


Figure 2-12: Seven regressors were chosen for the three datasets falling into two categories. The first category consisted of ordinary least squares regressors such as LassoCV, Bayesian Ridge and Ridge Regression. The second category was random forest regression, support vector regression, neural networks and K-nearest neighbor regression. The first two datasets SP39 and PDBCYS performed better with the second type of regressor while the IVD-54 did so with the first type of regressor.

Table 2-4: SP39 Results.

SP39 4-Fold CV	B=2		B=3		B=4		B=5		Overall	
Model	**Qp**	**Qc**	**Qp**	**Qc**	**Qp**	**Qc**	**Qp**	**Qc**	**Qp**	**Qc**
Random Forest	94.9	94.9	88.4	91.1	88.9	93.7	86.7	91.1	90.6	92.7
Support Vector	96.2	96.2	89.0	91.6	85.9	90.9	84.4	90.7	90.4	92.3
Neural Network	95.5	95.5	87.0	90.0	83.8	89.9	75.6	83.6	88.1	90.2
KNN Regression	96.8	96.8	87.0	90.4	81.8	88.1	71.1	85.8	87.7	90.4
Ridge Regression	85.3	85.3	80.8	85.8	77.8	86.6	46.7	71.1	78.3	83.5
Bayesian Ridge	85.3	85.3	76.7	82.0	72.7	83.6	42.2	64.9	75.3	80.4
LassoCV	66.7	66.7	37.7	45.4	27.3	38.9	26.7	43.6	44.4	48.1

SP39's CSP displayed the highest homology and as expected showed the greatest Qp and Qc. Random forest regression was found to be optimal with SVR closely following as seen in other publications [28].

Table 2-5: PDBCYS-R Results.

PDBCYS 20-Fold CV	**B=2**		**B=3**		**B=4**		**B=5**		**Overall**	
Model	**Qp**	**Qc**	**Qp**	**Qc**	**Qp**	**Qc**	**Qp**	**Qc**	**Qp**	**Qc**
Random Forest	90.0	90.0	80.0	85.5	68.3	78.0	70.3	80.5	80.6	84.0
Support Vector	90.0	90.0	83.5	87.1	61.0	75.0	67.6	79.5	80.2	83.6
KNN Regression	93.0	93.0	81.2	84.7	58.5	77.4	59.5	79.5	79.1	84.1
Bayesian Ridge	91.0	91.0	81.2	85.5	56.1	72.6	64.9	77.3	78.7	82.3
Ridge Regression	91.0	91.0	81.2	85.5	56.1	72.6	64.9	77.3	78.7	82.3
LassoCV	89.0	89.0	78.8	83.9	58.5	75.6	67.6	75.1	77.9	81.5
Neural Network	90.0	90.0	80.0	83.9	58.5	72.0	51.4	69.7	76.4	79.7

Again random forest regression with SVR closely following were the optimal regressors.

Table 2-6: IVD-54 Trained on SP39 Results.

IVD-54 Trained SP39	B=2		B=3		B=4		B=5		Overall	
Model	**Qp**	**Qc**	**Qp**	**Qc**	**Qp**	**Qc**	**Qp**	**Qc**	**Qp**	**Qc**
LassoCV	79.3	79.3	80.0	82.2	14.3	32.1	33.3	53.3	68.5	68.5
Ridge Regression	79.3	79.3	80.0	82.2	0.0	35.7	33.3	46.7	66.7	68.5
Bayesian Ridge	79.3	79.3	80.0	82.2	0.0	35.7	33.3	46.7	66.7	68.5
Neural Network	75.9	75.9	80.0	80.0	0.0	14.3	0.0	26.7	60.3	63.0
KNN Regression	72.4	72.4	53.3	60.0	0.0	28.6	33.3	40.0	55.6	56.8
Support Vector	69.0	69.0	53.3	62.2	0.0	42.9	33.3	46.7	53.7	59.6
Random Forest	55.2	55.2	53.3	60.0	14.3	21.4	33.3	46.7	48.1	49.3

Curiously, IVD-54 achieved the highest Qp and Qc using modified ordinary least squares regressors. Paradoxically, Modeller identities of less than 40 percent for both the SP39 training and IVD-54 testing were found to be optimal. IVD-54 was trained on SP39 rather than cross validation in keeping with the literature methods.

2.4.3 Prior Work Performance Comparison

The metrics of success Qp and Qc are compared with previous works across the differing datasets. This is not an exhaustive list but only the most competitive scores were included. The comparisons between our work and previous works are found in Figures 2-13, 2-14 and 2-15.

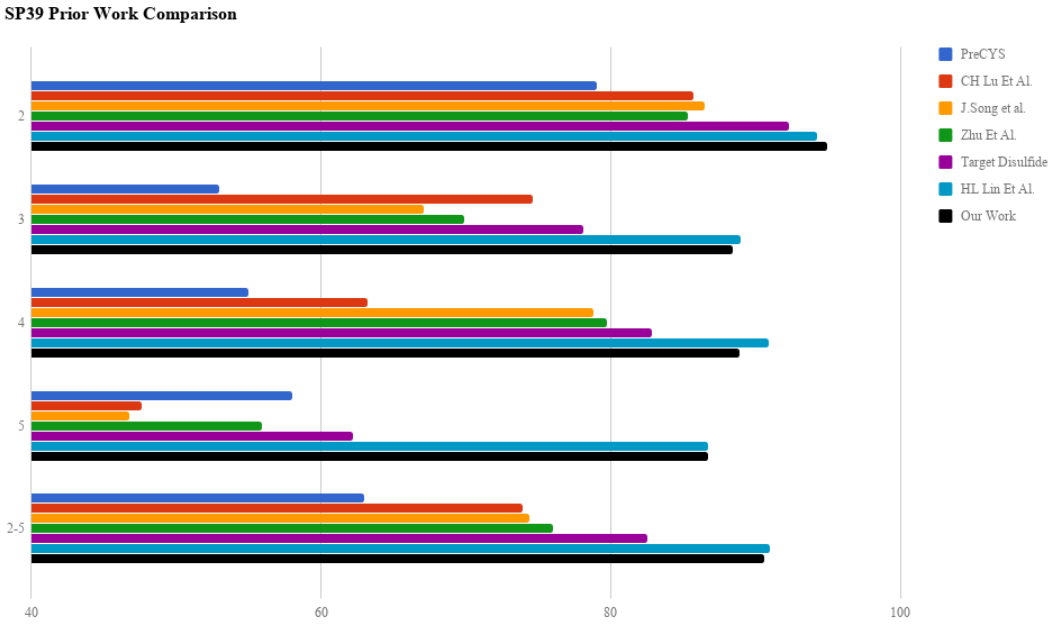


Figure 2-13: Prior works compared for SP39.

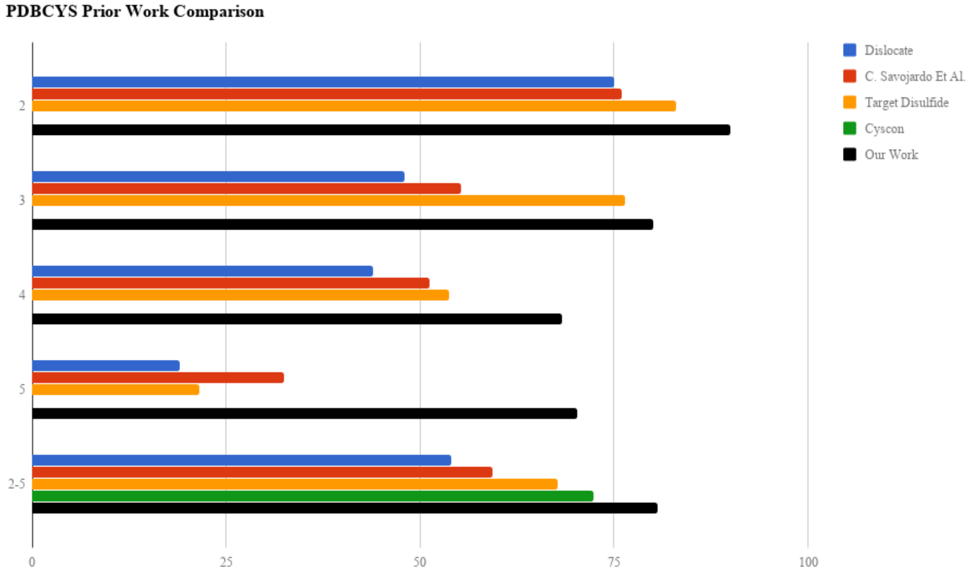


Figure 2-14: Prior works compared for PDBCYS-R.

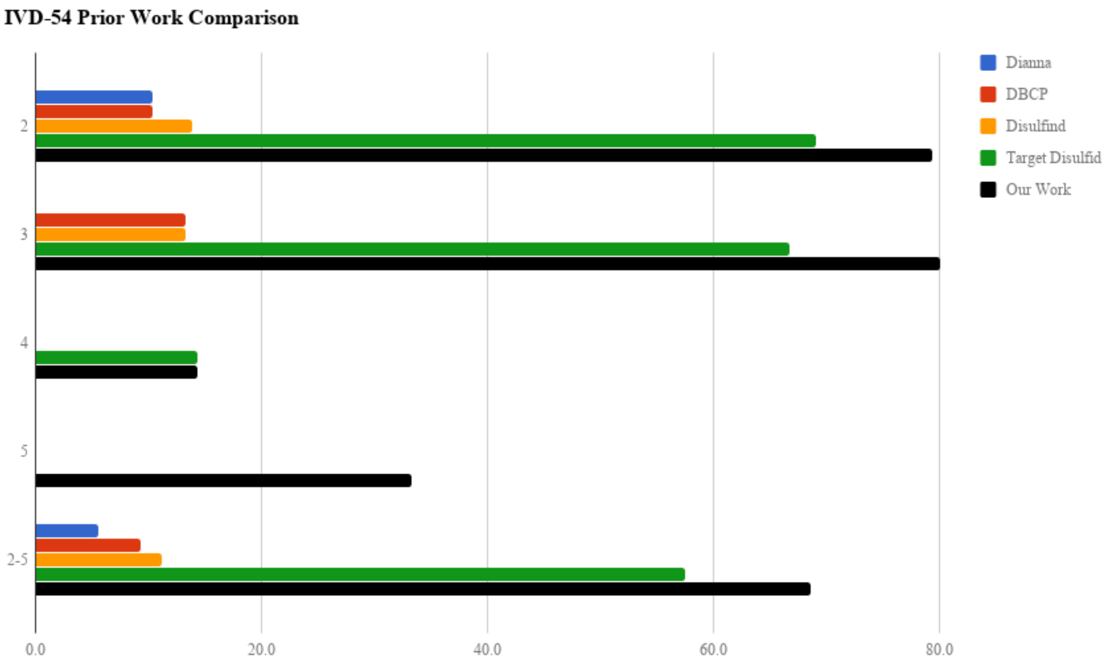


Figure 2-15: Prior works compared for IVD-54.

2.4.4 Modeller Percent Identity Threshold

It was noted that varying the identity produced marked differences in the Qp and Qc. It is not clear what was used in previous research regarding 100% identity and if identical sequences are included. Table 2-7 and Figure 2-16 show the effect of varying Modeller's alignment identities and therefore the quality of PDB's used in the experiment.

Table 2-7: Varying Modeller Identity Threshold for SP39 Dataset.

Dataset: SP39	**B=2**		**B=3**		**B=4**		**B=5**		**Overall**	
	Qp	**Qc**	**Qp**	**Qc**	**Qp**	**Qc**	**Qp**	**Qc**	**Qp**	**Qc**
40	91.7	91.7	82.2	86.1	78.8	85.6	60	73.8	82.5	85.2
60	91.7	91.7	81.5	85.3	81.8	85.6	64.4	75.1	83.4	85.2
80	93.5	93.5	84.9	88.1	83.8	87.9	73.3	81.8	86.5	88.3
100	94.9	94.9	88.4	91.1	88.9	93.7	86.7	91.1	90.6	92.7

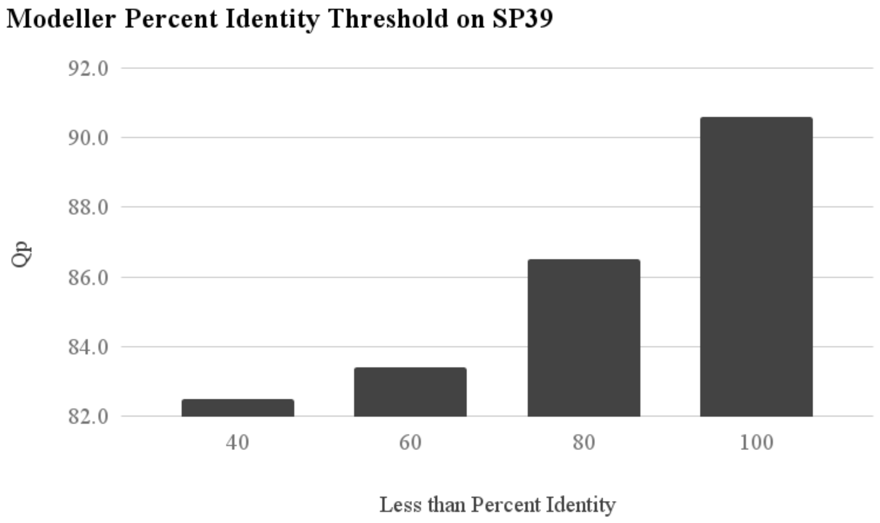


Figure 2-16: Modeller identity thresholds effect on Qp.

Less than 100 percent identity means there was no threshold and all matches were included except for those matching the protein identically. It is assumed that the proteins for which pdb's do not exist would use all available information except for their own pdb that is unknown.

2.5 Conclusion

This paper provides methods to improve accuracy of cysteine bond prediction against the existing benchmarks. As evidenced by the accuracies shown in the Results section, we managed to outscore the results of many prior works which contained the most accurate method to predict cysteine bonds to date. We believe this was due to a combination of z-score normalization and the local similarities instead of default PSSMs. In addition to simply improving on the accuracies from the prior works, we also give a more straightforward breakdown of cysteine bond prediction. Our goal was to provide a new benchmark of cysteine bond prediction accuracy for future researchers to build upon.

In summary, the sources of the features that can be used to predict cysteine bonding are correlated mutations (CM) of the sequence, distance of oxidized cysteines (DOC) from the amino acid sequence, position specific scoring matrix (PSSM) or Local Similarity Matrix (LS) from the PSI Blast software, predicted three-dimensional distance between two cysteine residues (PDTCR) using Modeller software; predicted secondary structure (PSS) through the PSI Pred software.

Prior work utilized SVM, neural nets and random forests to achieve accuracies as shown in the tables above. The random forest regressor and lassoCV regressors were determined to be optimal.

CHAPTER 3

CYSTEINE REDOX SUSCEPTIBILITY AND THE RESIDUE ADJACENCY MATRIX

3.1 Overview

Free radicals that form from reactive species of nitrogen and oxygen can react dangerously with cellular components and are involved with the pathogenesis of diabetes, cancer, Parkinson's, and heart disease. Cysteine amino acids, due to their reactive nature, are prone to oxidation by these free radicals. Determining which proteins are affiliated with oxidized cysteines is crucial to our understanding of these chronic diseases. Wet lab techniques, like differential alkylation, to determine which cysteines oxidize are often expensive and time-consuming. We utilize machine learning as a fast and inexpensive approach to identifying cysteines with oxidative capabilities.

We created the original features RAMmod and RAMseq for these machine learning algorithms. We also incorporated well known features such as PROPKA, SASA, PSS and PSSM. Our algorithm requires only the protein sequence to operate; however, we do use template matching by MODELLER to acquire 3D coordinates for additional feature extraction. There was a mean improvement of RAM over D by 20.45%. It was statistically significant with a p-value of 0.0078 and a mean improvement of 0.078

Matthew's Correlation Coefficient. The 95% confidence interval for the one-sided paired student's t-test was 0.049 to infinity. RAM provided a MCC increase of 0.173 over PSSM with a p-value of 0.040 and an average 70.08% improvement.

3.2 Introduction

Free radicals are known to adversely alter various biological structures (like lipid, proteins, and DNA) by introducing uneven charge distributions over these complex molecules. If these free radicals become too abundant, then a condition known as oxidative stress occurs. This condition can lead to various chronic illnesses. Oxidation susceptible cysteines in the mitochondria have been proven to play a critical role in defense against free radicals by absorbing these species [34]. Cysteines also assist the body's antioxidant defense responses by inducing the glutathione response pathways [35]. Due to cysteine's critical role in combating oxidative stress, there has been a growing interest in determining oxidation susceptible cysteines [36].

Cysteine is a unique amino acid that is a functional site in many proteins. It can be nitrosylated and glutathionylated, and can form sulfinic acid, sulfenic acid, sulfonic acid, disulfide bonds, selenocysteine, coordinate metals as well as other less common oxidations [37]. Our research and the prior works to which we compare our results are limited to the former six chemistries. Some additional distinguishing properties of cysteine are its chemical plasticity, nucleophilicity, high reactivity, relative rarity, involvement in structural stabilization, catalytic activity, its status as a most common metal coordinator, and its high degree of conservation [38, 39]. Cysteine plays an interesting role in redox regulation and signaling, but this role is not completely understood. Through our prediction and scoring of cysteines that are redox susceptible, our hope is that researchers can more easily understand the role of cysteine in free radical and disease states for the advancement of treatment options.

Our tools seek to assist researchers who wish to profile oxidized cysteines in order to better understand the complications that arise from oxidative stress and how to relieve the condition. We hypothesize that RAMseq (Residue Adjacency Matrix for sequences) and RAMmod (Residue Adjacency Matrix for MODELLER) features outperform known feature sources. We also incorporate PROPKA, SASA, PSS, and PSSM in addition to RAMseq and RAMmod. Our technique notably includes both features from a template matched 3D model (PROPKA, SASA, and RAMmod) and techniques that just require the amino acid sequence (PSS, PSSM, and RAMseq). A description of the data used is in Table 3-1.

Table 3-1: Summary description of data. The special case RSC758 6,6 is RAM chosen with n = 6 for both RAMseq and RAMmod.

Data	Oxidized Cys	Reduced Cys	RAMseq n	RAMmod n	Features
RSC758	758	758	12	18	901
BALOSCTdb	161	161	6	7	561
OSCTdb	161	376	5	6	521
RSC758 6,6	758	758	6	6	541

3.2.1 Prior Works

DISULFIND [9] and DIANNA [8] were among the first to incorporate machine learning techniques to predict the oxidation state of cysteines in proteins. They operated only using the amino acid sequence information as inputs. These tools first predicted which of the inputted cysteines would form disulfide bonds via an SVM. Their work focused solely on the ability to discriminate disulfide bonds from non-disulfide bonds and did not consider other oxidation states of the cysteines. After the SVM finished its predictions, the bonding state of the oxidized cysteines was determined and returned to the user with a confidence score.

After DISULFIND and DIANNA, COPA (Sanchez *et al*., 2008) was invented to classify cysteines into the four potential reactivity groups: those that form disulfide bonds, those that coordinate with metals, those that remain in the reduced state, and those that are susceptible to reversible oxidation. Their program required 3-D coordinates, so it could only work on proteins that have their structural information provided in the Protein Data Bank. ROCD [40], Reversibly Oxidized Cysteine Detector, was created to work in a similar fashion to COPA. The program also required 3-D coordinates to operate. Lee's study focused on redox regulatory networks in order to better understand oxidative stress. Doulias, in his 2010 paper [41], characterized nitrosocysteine using solvent accessible surface area, pKa, and predicted secondary structure in order to determine the post-translational role of nitric oxide in proteins.

Hydrogen bonding and its relation to pKa was investigated for redox sensitive cysteines to gain biochemical insights into signaling [42]. Thiol chemistry and specifically cysteine redox susceptibility was studied using quantum mechanics computational simulations for finding catalysis and regulation [43]. RSCP [44], Redox Sensitive Cysteine Prediction, was made to predict redox-sensitive cysteines. RSCP was slightly less accurate than COPA and ROCD, but the program was applicable to a wider range of proteins because it only required the amino acid sequence, eliminating the need for expensive wet-lab techniques like X-Ray Crystallography or NMR.

CPIPE was invented next to provide a comprehensive computational platform on which to study various properties of cysteine residues [45]. The program attempts to tie together several machine learning approaches to determine cysteine reactivity. It can work with either the sequence data alone or with both the sequence data and additional structural data. Our work improves accuracies of the tools that already exist by utilizing new features, RAMseq and RAMmod, to feed into our machine learning algorithms.

3.3 Methods

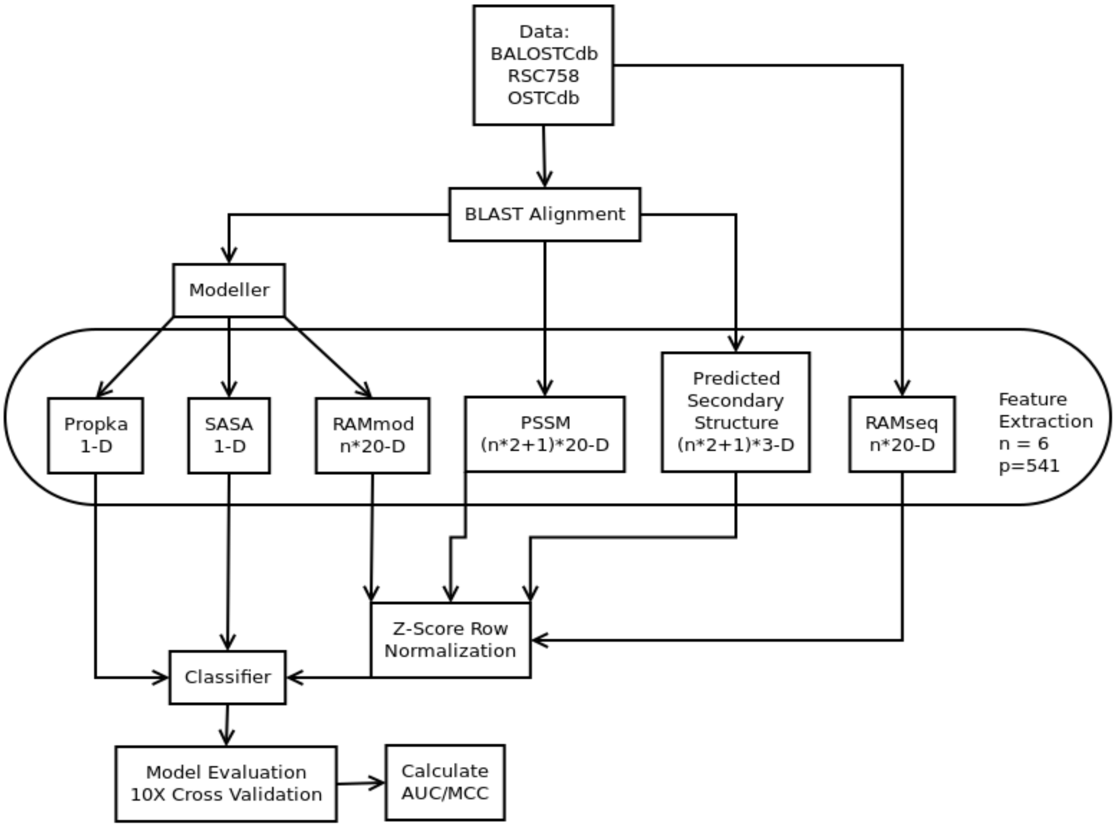


Figure 3-1: Description of process via flowchart diagram.

Fig. 3-1 highlights the major components of our data mining framework. In general terms, our tool takes amino acid sequence data as an input, compares it to databases of related sequences for additional data, extracts 6 features from the collected data (for a total of 541 dimensions to be used in our predictors). These features are then sent to a trained classifier. Finally, a list of the cysteines from the original sequence that are the most likely to oxidize are returned along with their confidence scores. Our tool is not only a useful aggregation of the most prevalent features to date but is also more accurate than previous tools because of our inclusion of our originally engineered features: RAMseq and RAMmod.

We decided to take only the amino acid sequence of the protein as input, and not the 3-D coordinates of the protein. Researchers have extracted sequence data from around 93 million proteins, whereas only 130 thousand proteins have known 3-D structures as of the date of this writing. Our work, therefore, remains general enough to be useful to a larger portion of the proteomics community. Our hope is that researchers who do not have access to expensive techniques like X-ray Crystallography can still get accurate estimates of cysteine oxidation from the sequence data alone.

Although we start with only the primary amino acid sequence, we do use predictive algorithms to estimate the secondary and tertiary structures for use in some of our features. These predictive algorithms (like MODELLER and PSSPred) estimate structural information from the original sequence in order to use them in additional features. If structural information of the protein exists in the Protein Data Bank, we can then use that information directly instead of relying on estimations from MODELLER.

3.3.1 Dataset Creation

In order to score and validate our methods, we decided to use two datasets: BALOSCTdb, and RSC758. Sanchez and his team [46] created the independent dataset

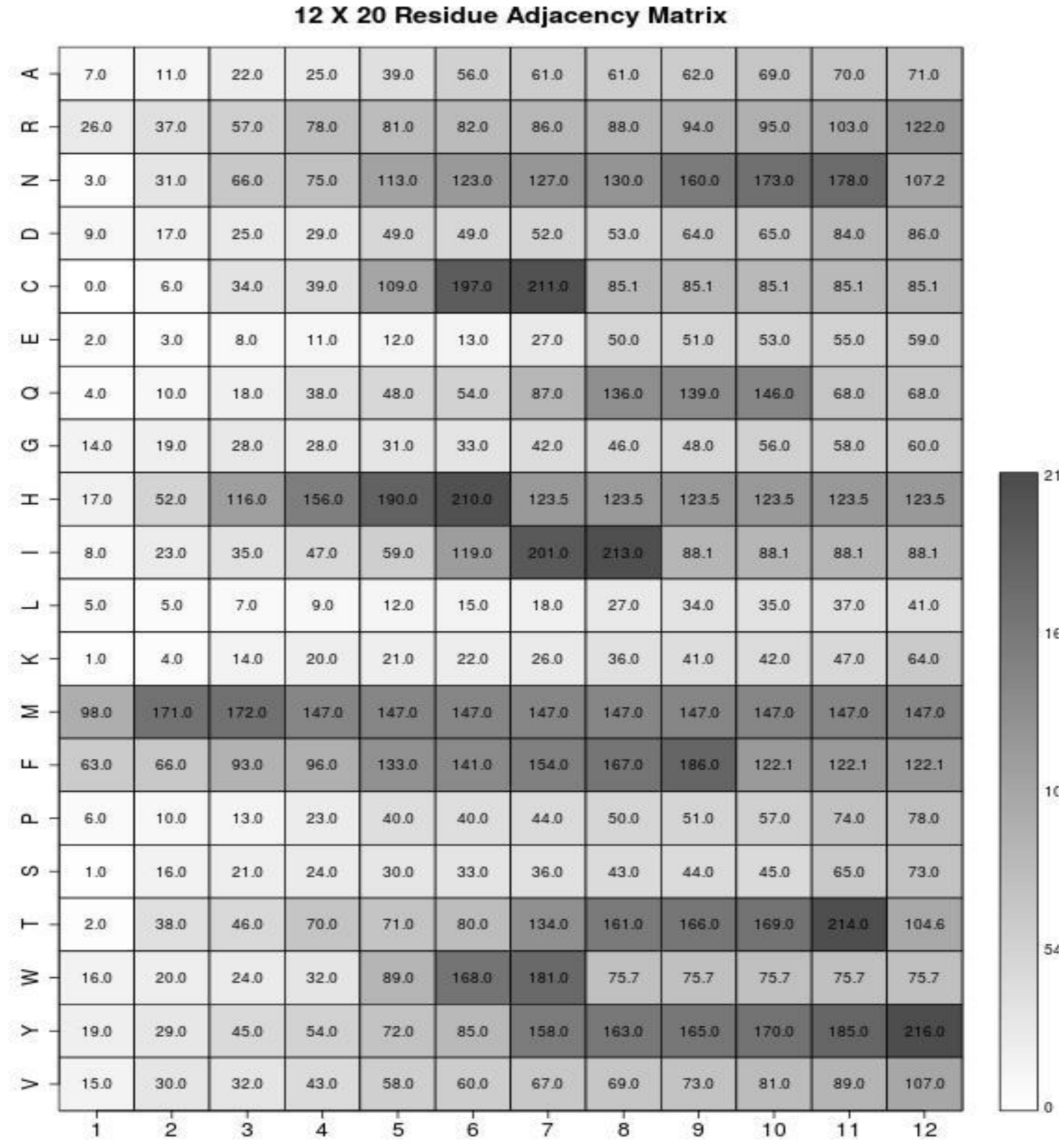
OSCTdb (Oxidation susceptible cysteine thiol database) in 2008 by using the blastall program of the BLAST software package [47] to reduce similar records that had identities greater than 35% and e-values less than unity. OSCTdb has 161 oxidation-susceptible cysteines, 301 oxidation-non-susceptible cysteines, and a total of 100 polypeptides. The BALOSCTdb (BALanced OSCTdb) dataset was created from OSCTdb by limiting the non-oxidation-susceptible cysteines to 161 and matching them to the 161 cysteines that undergo oxidation in order to balance the number of oxidization-susceptible cysteine thiols with the number of non-oxidation-susceptible cysteine thiols. RSC758, Redox-Sensitive Cysteine 758, [44] was created next and was intended to be similar to BALOSCTdb but with a greater number of entries. RSC758 has 758 entries for both oxidized and non-oxidized cysteines, and, like BALOSCTdb, it ensures a balance between the number of oxidation-susceptible cysteines and cysteines which are not susceptible to oxidation. We only use the sequence data from the datasets; however, the 3-D structures for some of these proteins have been identified and are available in the template databases that we use.

3.3.2 RAMseq

The RAMseq (Residue Adjacency Matrix from sequence data), an original feature used in this work, can be calculated on the raw sequence data without any other accompanied data (like 3-D coordinates or the secondary structure). The RAMseq is calculated by taking the absolute value of the distance from the target cysteine to each of the twenty amino acids found in human proteins. We attempt to find the n nearest amino acids of each type. In other words, the distance of the target cysteine to each amino acid in the sequence is recorded along with the type of the amino acid. We choose the n shortest distances for each type. This forms a matrix that is twenty rows long (one row for each amino acid) and n columns wide;. n is chosen for each dataset for optimal performance, although leaving it at a constant six largely does not affect the accuracy.

RAMseq is similar to a cysteine separation profile [14], but is used for all amino acid residues instead of solely cysteine. RAMseq is a type of homology match because similar RAM matrices result in similar reactivities of thecysteine. This attribute makes RAMseq effectively act as a template matching process. The data for cysteine and tryptophan distances consistently score as one of the most prominent features. Interestingly enough, these are the two most conserved amino acids residues, as indicated by the diagonals on the BLOSUM62 matrix (a substitution matrix used for sequence alignment of proteins) [48].

RAMseq compliments a PSSM (Position Specific Scoring Matrix) in several key ways. Firstly, RAMseq measures amino acid residue proximity to the target cysteine, whereas a PSSM only measures the frequency of each amino acid in a certain window. RAMseq's data can also extend to positions that are further away than a PSSM reaches without oversaturating models with redundant data. As n increases, the dimensionality of a PSSM increases by 20*n, whereas the size of RAMseq increases by simply n. RAMseq works directly on the inputted sequence data without relying on sequence alignments like a PSSM must, which results in features that are much more relevant to studying the protein in question as well as a much shorter processing time. An example of our RAM calculation is shown below in Figure 3-2.



	1	2	3	4	5	6	7	8	9	10	11	12
A	7.0	11.0	22.0	25.0	39.0	56.0	61.0	61.0	62.0	69.0	70.0	71.0
R	26.0	37.0	57.0	78.0	81.0	82.0	86.0	88.0	94.0	95.0	103.0	122.0
N	3.0	31.0	66.0	75.0	113.0	123.0	127.0	130.0	160.0	173.0	178.0	107.2
D	9.0	17.0	25.0	29.0	49.0	49.0	52.0	53.0	64.0	65.0	84.0	86.0
C	0.0	6.0	34.0	39.0	109.0	197.0	211.0	85.1	85.1	85.1	85.1	85.1
E	2.0	3.0	8.0	11.0	12.0	13.0	27.0	50.0	51.0	53.0	55.0	59.0
Q	4.0	10.0	18.0	38.0	48.0	54.0	87.0	136.0	139.0	146.0	68.0	68.0
G	14.0	19.0	28.0	28.0	31.0	33.0	42.0	46.0	48.0	56.0	58.0	60.0
H	17.0	52.0	116.0	156.0	190.0	210.0	123.5	123.5	123.5	123.5	123.5	123.5
I	8.0	23.0	35.0	47.0	59.0	119.0	201.0	213.0	88.1	88.1	88.1	88.1
L	5.0	5.0	7.0	9.0	12.0	15.0	18.0	27.0	34.0	35.0	37.0	41.0
K	1.0	4.0	14.0	20.0	21.0	22.0	26.0	36.0	41.0	42.0	47.0	64.0
M	98.0	171.0	172.0	147.0	147.0	147.0	147.0	147.0	147.0	147.0	147.0	147.0
F	63.0	66.0	93.0	96.0	133.0	141.0	154.0	167.0	186.0	122.1	122.1	122.1
P	6.0	10.0	13.0	23.0	40.0	40.0	44.0	50.0	51.0	57.0	74.0	78.0
S	1.0	16.0	21.0	24.0	30.0	33.0	36.0	43.0	44.0	45.0	65.0	73.0
T	2.0	38.0	46.0	70.0	71.0	80.0	134.0	161.0	166.0	169.0	214.0	104.6
W	16.0	20.0	24.0	32.0	89.0	168.0	181.0	75.7	75.7	75.7	75.7	75.7
Y	19.0	29.0	45.0	54.0	72.0	85.0	158.0	163.0	165.0	170.0	185.0	216.0
V	15.0	30.0	32.0	43.0	58.0	60.0	67.0	69.0	73.0	81.0	89.0	107.0

Figure 3-2: Typical Residue Adjacency Matrix computed from protein APEX_HUMAN1. Depicted is RAMseq based on Cysteine 99, that is involved in reversible disulfide bonding and glutathionylation. The sequence is ...ETKCSEN… where cysteine 99 is centered. Note the values do not strictly increase, because when there is not enough amino acids of the correct type, the mean of the previous amino acids is used. These matrices are used to template match each other, where similar matrices have similar redox sensitivity.

$$RAM_{i,j} = \left|C - AA_j^i\right| \qquad \textbf{Eq. 3-1}$$

In Equation 3-1 shown above, RAM is the value of the residue adjacency matrix, C is the index of the cysteine in question, and AA is the index of the amino acid. When there are not enough amino acids of the specified type to fill the matrix, an ARIMA Model, Auto Regressive Integrated Moving Average, can then be utilized. ARIMA is a time series statistical technique that can provide the missing data points. Using the forecast package from R and selecting (p, d, q) according to the PACF and ACF plots gives us either a trend or a mean prediction. Although means had a strong positive performance capability, trends were not found to improve the scores. If two cysteines have a similar mean distance to every amino acid of a certain type, then the two likely share similar reactivity. For instance, the mean of every tryptophan's distance to cysteine was chosen as an important feature by the random forest classification model for determining reactivity. As an example, given the n shortest distances of [6,12,NA,NA,NA,NA], the final RAMseq is taken as [6,12,9,9,9,9]. In the rare case that no amino acids of a certain type are present in the protein, then the mean distance of the n-nearest of all the other types of amino acids is copied along the row n times.

3.3.3 BLAST Alignments

BLAST (Basic Local Alignment Search Tool) is a widely used software tool that allows one to query a database for a list of similar sequences to a target sequence. Many of the features that we use (including the PSSM and the PSS tables) require sequence alignments. All of our 3-D features also implicitly rely on BLAST because MODELLER requires BLAST alignments to make its predictions on the tertiary structure of the target proteins. In fact, RAMseq was the only feature that we used which did not require a BLAST alignment to work. BLAST uses heuristic methods to search large databases for sequence matches quickly. Although it does not necessarily find the optimal alignments (like the Smith-Waterman algorithm can), the speed with which it can search huge genomes make it a practical choice for our purposes.

BLAST works by first making a k-letter word list from the target sequence (for instance, with k=3 and a sequence of PLDAG, BLAST would make a word list of PLD, LDA, and DAG). Next, possibly matching words are scored for each entry by use of a substitution matrix (usually BLOSUM62). Words that exceed a given threshold are designated as "high-scoring words" and are used for the remaining searches. The database is scanned for an exact match with one of the high-scoring words. On a hit, a window of the neighbors of the exact hit is expanded and scored (using the same substitution matrix from before) until the score decreases (i.e. an unlikely substitution is caught).

The score of this window is recorded, and if found significant, it is combined with other so-called high scoring pairs into a longer alignment. The expect score (the probability that an unrelated sequence would obtain a higher score by chance) is calculated for the alignment, and the alignments with e-values above the threshold are returned.

3.3.4 PSSM

PSSMs (Position Specific Scoring Matrices), also known as Position Weight Matrices, are a useful data structure that captures the amino acid frequency profile of a certain window in a protein sequence. They were first introduced by Gary Stormo and his colleagues in their 1982 paper [49] to explore patterns in E-coli nucleotide sequences. We use PSSMs as a feature in our machine learning algorithms in order to capture the amino acid compositions of sequences that are similar to our target sequence.

A PSSM is calculated from alignments at a position by dividing the observed substitutions of a certain amino acid by the expected number of substitutions. A ratio greater than one indicates that the amino acid substitution is favored. Ratios less than one indicate that the amino acid substitution is not favored [50]. For a window size of 2*k+1 (k positions to the left of the target cysteine, k to the right, and the target cysteine position itself) and the twenty major amino acids, we get a matrix that is twenty rows long and 2*k columns wide for a total of 20*(2*k+1) features. We chose 6 for k, so our final PSSMs had a row dimension of 260 entries. Blastp was used from the BLAST software suite with an e-value of 0.005, and the out_pssm setting enabled. The PSSM output from blastp is comparable to a manual calculation with a local similarity matrix but with an amino acid inverse frequency multiplication and log base 2 transform applied.

PSSMs reveal evolutionary patterns in a local (position specific) manner. Proteins are known to generally conserve their structure as they mutate, so cysteine reactivity being conserved through small mutations is a logical extension. Therefore, our use of PSSMs should effectively increase the amount of data that can be fed to our machine learning algorithms because the similar sequences that we gather probably have identically oxidized cysteines.

3.3.5 PSS - Predicted Secondary Structure

Segments of amino acids can arrange themselves into unique local 3-D structures. These structures generally fall into three classes: alpha helices, beta sheets, or coils. In the same way that we can computationally estimate 3-D structural information from our protein sequences, we can also predict which type of secondary structure that the amino acid at a certain position belongs. We used the PSIpred software to make these predictions. PSIpred builds two neural networks. The first network has 315 input neurons and 3 output neurons, and the second network has 60 input neurons with 3 output neurons. PSIpred requires an alignment outputted from a BLAST to operate. We use a window size of thirteen positions (the target cysteine plus the six positions to the left and the six positions to the right) for the PSS matrix. The final matrix is then thirteen by three (the confidence score for the three classes of secondary structures) which results in a thirty-nine dimensional feature source for the classifier.

3.3.6 MODELLER

We used the MODELLER software through a Python API to estimate the 3-D structure of a protein using a technique known as comparative modeling. Comparative modeling predicts the 3-D structure of a protein based on BLAST alignments to other proteins which have a known structure. The comparative modeling algorithm that MODELLER utilizes consists of four general steps: fold assignment, target-template alignment, model building, and model evaluation. MODELLER first obtains an alignment of a target sequence and a database of template structures. MODELLER then automatically calculates a model containing all non-hydrogen atoms and returns a PDB file containing the estimated 3-D coordinates of the target protein.

In our work, we used MODELLER with the default settings. We decided to take the ten closest protein structures as our template database for MODELLER. If the protein of interest has a 3-D structure available, it was used in the template. However, the MODELLER algorithm was still run on the sequence (in essence, estimating a structure that is already known). Sometimes, the alignments that we chose to feed into MODELLER had insufficient overlap, which caused the model building to fail. If the templates that we chose broke MODELLER in this way, then we simply dropped the offending template or templates and tried again.

3.3.7 RAMmod - Residue Adjacency Matrix from MODELLER Data

RAMmod is the second original feature that we used in our work. Like RAMseq, RAMmod works by building a proximity matrix of the n-nearest amino acids. Rather than using the simple positional differences like RAMseq, RAMmod uses the Euclidean distance of the target cysteine to the residues obtained from the protein's 3-D structure. For each amino acid, we take the n closest Euclidean distances to the cysteine to build the matrix. Like RAMseq, if there are not n amino acids in the whole sequence of a certain type, then the mean of the Euclidean distances for the available amino acids of the specified type is used to fill in the remainder of the row. If no amino acids of a certain type exist, then that row is filled with the mean distance of every other amino acid type. RAMmod is a 20*n dimensional matrix like RAMseq. Like RAMseq, the data for cysteine and tryptophan distances score as the most prominent features.

3.3.8 PROPKA - Protein pKa Data

We determined the pKa values of our target cysteine sulfur atoms by using the PROtein PKA software, PROPKA [51, 52, 53]. The equation for determining pKa values is shown in Equation 3-2:

$$pKa = pK_{Model} + \Delta pKa \qquad \textbf{Eq. 3-2}$$

pKModel is set at 9.00 while $\boldsymbol{\Delta}$pKa was determined from hydrogen bonds, desolvation, and charge interactions. PROPKA requires 3-D coordinates, which we provide from MODELLER. The pKa values typically vary from 0.00 to 14.00, but we assign a special value of 99.99 to indicate a disulfide bond. The pKa was determined to be an important feature for determining the reactivity of cysteine. It was the third best discriminator in COPA's decision tree. A pKa value greater than nine strongly indicates the reactivity of cysteine.

3.3.9 SASA Data

The solvent-accessible surface area (SASA) is the surface area (measured in square angstroms) of a molecule that is available to a given solvent. We used FreeSASA [54] with the Naccess [55] settings in order to determine the SASA of our target proteins. FreeSASA requires 3-D coordinates which, again, we gather through MODELLER. The SASA value of a protein is helpful for determining the reactivity of a target cysteine. Proteins with similar SASA scores are likely to have similar redox sensitivity. SASA was the second most important discriminator in COPA's decision trees. Values greater than 1.3 angstroms squared tend to indicate a reactive cysteine.

3.3.10 Normalizing the Data

Before we inputted our features into our machine learning algorithms, we experimented with applying both Z-score Normalization (Eq. 3) and Min-Max Normalization (Eq. 4) to our data. We normalized on the sets of each feature array at each row. Features with a dimensionality of one (like SASA) were not normalized. Z-score Normalization was found to be more effective than Min-Max Normalization. Normalizing the entire feature matrix or the entire row was ultimately found to be less effective.

$$Z_i = \frac{row_i - row_{mean}}{row_{sd}} \qquad \textbf{Eq. 3-3}$$

$$Z_i = \frac{row_i - row_{min}}{row_{max} - row_{min}} \qquad \textbf{Eq. 3-4}$$

3.3.11 Classification and Metrics of Performance

We experimented with classification using a random forest algorithm, an SVM, and KNN. Random forest was ultimately found to be the most effective. Random forests are resistant to overfitting due to bootstrapping and a limit on the number of features considered at each split. Pruning the trees (by setting the max_depth parameter) in the random forest can help to prevent overfitting. Random forests can also rank features by their importance. A collection of binary decision trees each evaluate the reactivity of our target cysteine. The average of the trees is then evaluated to a receiver operating characteristic curve, ROC. This curve plots the sensitivity against the false positive rate (1 - specificity). The area under this curve, AUC, is a single number that describes the ability of the classifier to separate the data into two classes (in our case, cysteines that undergo oxidation and those that do not). A confusion matrix is then made to determine the Matthew's Correlation Coefficient using Equation 3-5.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \qquad \textbf{Eq. 3-5}$$

3.4 Results

We provide the following figures to show the ability of RAMseq and RAMmod to determine more accurately the oxidation susceptibility of cysteines. In the figures below, we use RAM to refer to the combined feature matrix of RAMseq and RAMmod. The feature D is the absolute value of the distances of the n nearest cysteines to the target cysteine. D provided the highest discriminative ability of redox susceptible cysteines before this work.

3.4.1 RAM vs. D

A comparison between D and RAM is made using the metrics of success MCC and AUC for three datasets in Figure 3-3, the actual numbers we experimentally determined are in Table 3-2.

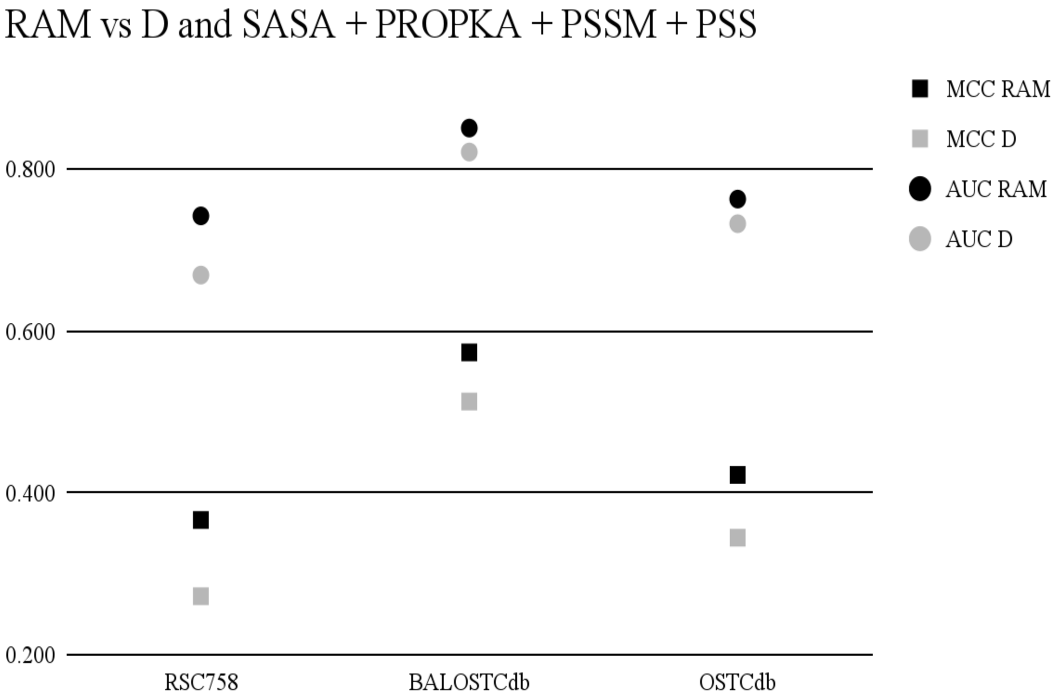


Figure 3-3: A comparison of the area under the receiver operating characteristic curve and Matthew's Correlation Coefficient. Both RAM and D had the SASA, pKa values, the PSSM and the PSS included as additional features. Therefore, the only difference between the two feature systems is RAM and D.

Table 3-2: Results obtained comparing RAM vs D. Including the SASA, pKa values, the PSSM and the PSS for both RAM and D.

RAMseq + RAMmod + SASA + PROPKA + PSSM + PSS					
Dataset	**ACC**	**SN**	**SP**	**MCC**	**AUC**
RSC758	0.679	0.569	0.789	0.367	0.743
BALOSCTdb	0.773	0.621	0.925	0.574	0.851
OSCTdb	0.703	0.547	0.859	0.422	0.763
D + SASA + PROPKA + PSSM + PSS					
Dataset	**ACC**	**SN**	**SP**	**MCC**	**AUC**
RSC758	0.627	0.45	0.805	0.272	0.669
BALOSCTdb	0.742	0.578	0.907	0.513	0.822
OSCTdb	0.683	0.646	0.721	0.345	0.733

3.4.2 RAM vs PSSM

PSSM is a frequently used method in proteomics and genetics. Because RAM has been shown below to outperform PSSM, there is a great deal of promise for using RAM in broader applications. When we compared RAM to PSSM we experimentally determined the difference as seen in Figure 3-4 and Table 3-3.

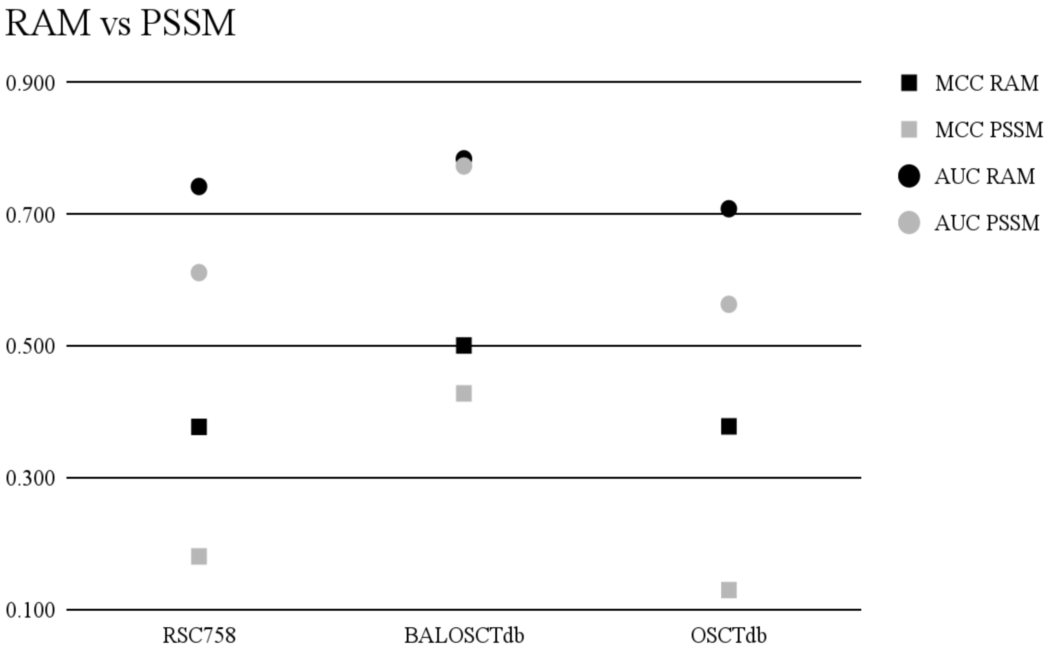


Figure 3-4: Matthew’s Correlation Coefficient and area under the receiver operating characteristic curve. No other features are included.

Table 3-3: Results obtained comparing RAM vs PSSM without any other features.

RAMseq + RAMmod					
Dataset	**ACC**	**SN**	**SP**	**MCC**	**AUC**
RSC758	0.676	0.496	0.856	0.378	0.743
BALOSCTdb	0.748	0.683	0.814	0.501	0.785
OSCTdb	0.674	0.472	0.875	0.378	0.709
PSSM					
Dataset	**ACC**	**SN**	**SP**	**MCC**	**AUC**
RSC758	0.586	0.745	0.426	0.181	0.612
BALOSCTdb	0.711	0.627	0.795	0.428	0.774
OSCTdb	0.567	0.752	0.383	0.130	0.564

3.4.3 Prior Works

In the following section, we make comparisons between RAM, RSCP and COPA. RSCP's primary contribution was the ability to use sequential features without the need of solved 3-D structural data. RSCP, therefore, is more broadly applicable than algorithms like COPA, which requires a PDB to predict cysteine redox susceptibility. However, COPA's accuracy was higher than RSCP's accuracy. RAM is a hybrid approach that accepts structural features but is able to use MODELLER predictions when only sequential data is given. When we compared self-reported results of RAM to two other prior publications we found the following differences in Figure 3-5 and Table 3-4.

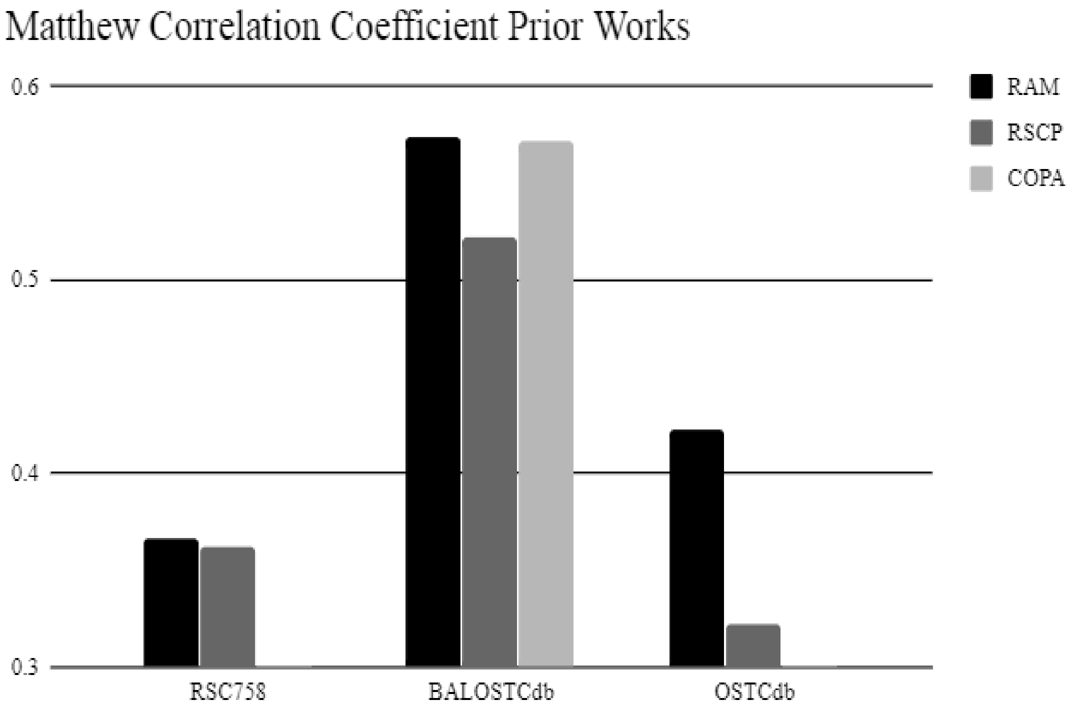


Figure 3-5: A comparison of RAM with all supplementary features against the other two methods (RSCP and COPA) on our 3 datasets (RSC758, BALOSTCdb and OSTCdb). RAM has the highest MCC of all methods on all datasets.

Table 3-4: Comparison of RAM to prior works COPA and RSCP. NA indicates data not provided by prior works.

RAM					
Dataset	**MCC**	**AUC**	**ACC**	**SN**	**SP**
RSC758	0.383	0.743	0.687	0.573	0.800
BALOSCTdb	0.574	0.851	0.773	0.621	0.925
OSCTdb	0.422	0.763	0.703	0.547	0.859
RSCP					
Dataset	**MCC**	**AUC**	**ACC**	**SN**	**SP**
RSC758	0.362	0.727	0.679	0.602	0.756
BALOSCTdb	0.522	0.821	0.761	0.770	0.752
OSCTdb	0.322	NA	0.629	0.789	0.561
COPA					
Dataset	**MCC**	**AUC**	**ACC**	**SN**	**SP**
RSC758	NA	NA	NA	NA	NA
BALOSCTdb	0.572	0.823	0.786	0.776	0.795
OSCTdb	NA	NA	NA	NA	NA

3.4.4 Using an n of 6 for RSC758

We defaulted to a value of n = 6 but found RAMseq n = 12 and RAMmod n = 18 gave us the highest accuracy for the RSC758 dataset. Optimizing the values of n for RAMseq and RAMmod increased the AUC by 4.5% and MCC by 16.8%. However, optimizing the value of n may lead to overfitting the data. The performance metrics are in Table 3-5.

Table 3-5: Results from adjusting the n parameter on both RAMseq and RAMmod.

RAMseq + RAMmod + SASA + PROPKA + PSSM + PSS					
Dataset and n	**ACC**	**SN**	**SP**	**MCC**	**AUC**
RSC758 6,6	0.646	0.460	0.831	0.314	0.711
RSC758 12,18	0.679	0.569	0.789	0.367	0.743

3.4.5 Choosing an optimal Matthew's Correlation

Below in Figure 3-6 is the effect of varying the classification threshold of confidence scores for the confusion matrix on Matthew's Correlation Coefficient.

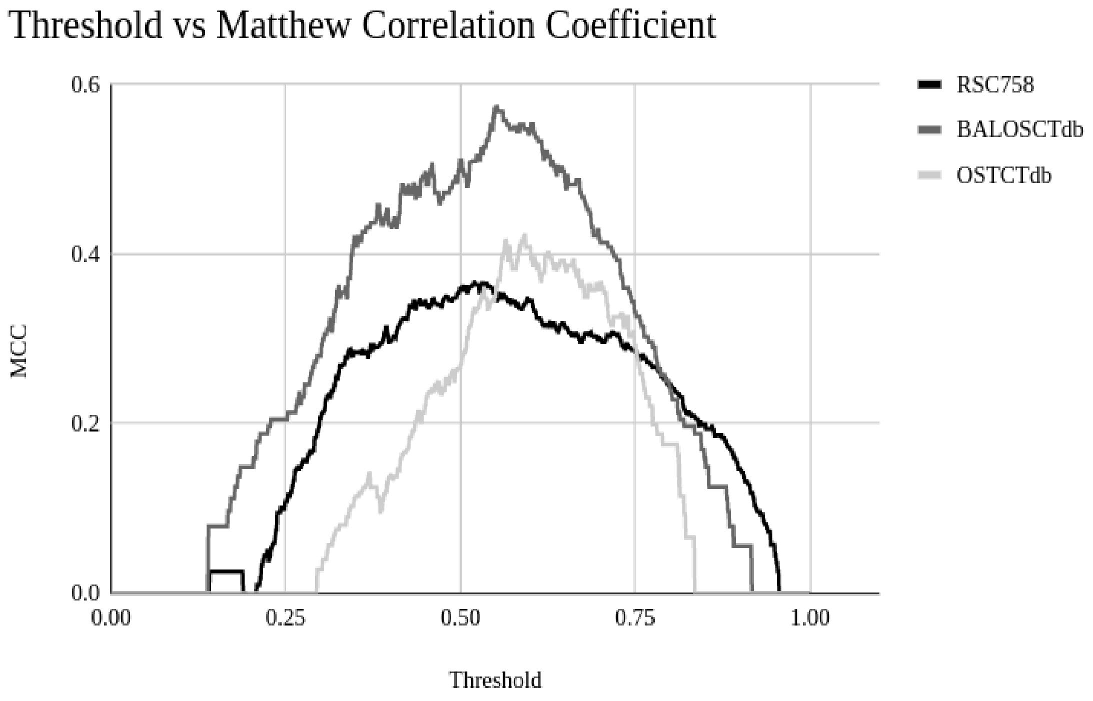


Figure 3-6: Matthew's Correlation Coefficient as a function of threshold. We chose the optimal threshold for MCC after varying classifier, classifier parameters and feature parameters for optimal AUC.

3.5 Discussion

Our results clearly show the benefit of applying our original features, RAMseq and RAMmod, to machine learning approaches for cysteine reactivity predictions. By every metric on which we scored, a feature system including RAM outperformed a system simply using D. D is a subset of RAM, so this performance increase is expected. While data mining solutions for predicting cysteine oxidation are certainly not new, we hope the methods presented here will serve as another step towards more accurate and generalizable techniques that are useful for a large range of researchers. Our work achieved state-of-the-art accuracies, yet only required the primary amino acid sequence of the target proteins. The simplicity of our program allows for accurate estimations of cysteine redox state without having to resort to expensive techniques like X-ray crystallography or NMR. Still, if one does have structural data of the target protein, our techniques can use that information to produce even more accurate predictions.

RAM is readily comparable to a PSSM. Because of the prevalence of PSSM's in thecurrent literature, a similar feature such as RAM could be useful in improving the accuracies of a great deal of proteomic and genetic machine learning techniques. PSSM does well conducting local searches but will frequently fail on distant conserved regions due to its small window size. RAM can handle these distant conserved regions quite well. RAM is more global in nature, while still acting as a local feature.

RAM can be applied to just about any problem a PSSM can be applied to. For instance, RAM can be modified to work with DNA. For DNA, the matrix is 4*n, and has the rows A, T, C and G. Future work where RAM data is used for DNA may yield results surpassing those of PSSM for genetic problems as has been shown in this work for cysteine reactivity.

CHAPTER 4

PREDICTING PYRIDOXAL-5-PHOSPHATE LYSINE POST-TRANSLATIONAL MODIFICATION ON THE PLP SWISSPROT DATABASE USING WINDOWED ALIGNMENT SCORING

4.1 Overview

Post translational modifications (PTM) are an extension of the repertoire of proteins' building blocks (the twenty amino acids). PTMs are involved in regulating protein activity, signalling the degradation of the protein, biomarkers of oxidative stress, etc. Alzheimers, for instance, is a result of hyperphosphorylation and glycosylation of tau protein. Both of these modifications are PTMs. Immune function and its dysfunction autoimmune disease, blood sugar regulation and its pathogenic state, diabetes, are also dependent upon proper post translational modification. This body of work predicts the pyridoxal-5-phosphate (PLP) lysine PTM for further research by those dependent on expensive techniques like tandem mass spectrometry. These techniques cannot keep pace with the discovery of newly sequenced proteins: thus a data mining approach is utilized that can relatively quickly discern which lysines are post-translationally modified by PLP in proteins. To accomplish these means Windowed alignment scoring, WAS, an original engineered feature source, was introduced. However, it is computationally expensive taking months to evaluate the experiments posed below. To reduce the computation time, an approximation was utilized. This reduced the time to two days and was found to be statistically equivalent in terms of performance by failing to reject the null hypothesis that the two methods are different with a p-value of 0.24. Its ability to determine the PLP Lysine PTM was 0.89 sensitivity and 0.98 specificity at 85% homology threshold.

4.2 Introduction

4.2.1 Biological Significance and Background Information

The role of pyridoxal-5-phosphate's post-translational modification of lysine residues is involved in the transamination reaction. All aminotransferases are catalyzed the same way. Specifically, there are three steps to complete the reaction. For the first transimination, see the figures below; then tautomerization and finally hydrolysis producing an alpha keto acid from the substrate amino acid. Selective inhibitors of this reaction has been implicated in the treatment of cancer [56]. Furthermore, GABA aminotransferase inhibitors cause the buildup of GABA in the synapse of neurons. Low GABA levels have been associated with Parkinson's [57], epilepsy [58], Huntington's [59] and Alzheimer's diseases [60]. Potential therapeutic inhibitors could target this reaction and have been created specifically for the treatment of epilepsy [61] and addictions [62].

More recently, it has been shown that the Pdxl subunits of the PLP synthase complex utilize the basic lysines 98 and 166 to catalyze the reaction of glyceraldehyde-3-phosphate and ribose-5-phosphate to PLP [63]. Generally, the transimination reaction can be seen in the following chemistry: PLP first joins to the amine functional group of lysine using a Schiff Base Link, then the amine of a free amino acid (or any amine ion) bonds to the PLP catalyzed by the lysine residue. M. Rodrigues *et al*. states, "A conspicuous gap in knowledge concerns the use of covalent lysine imines in the transfer of carbonylgroup-containing intermediates, despite their wide use in enzymatic catalysis." This suggests strongly that determining which lysines are active would shed light on this problem, hence our data mining and feature engineering solution. An overview of this reaction is illustrated in Figures 4-1, 4-2 and 4-3.

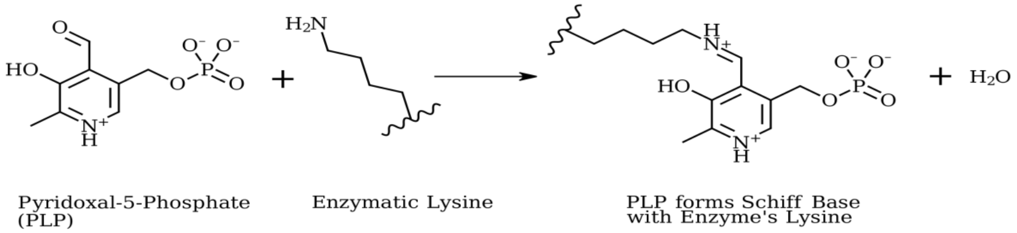

Pyridoxal-5-Phosphate (PLP)
Enzymatic Lysine
PLP forms Schiff Base with Enzyme's Lysine
H2O

Figure 4-1: First half transimination reaction of PLP with the protein's lysine. Step 1 of the transamination reaction.

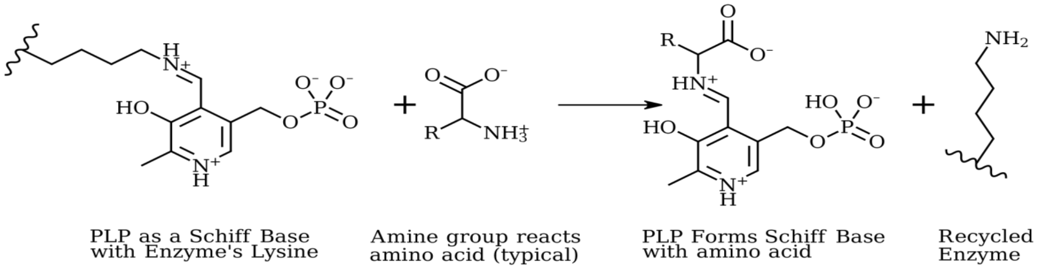

PLP as a Schiff Base with Enzyme's Lysine
Amine group reacts amino acid (typical)
PLP Forms Schiff Base with amino acid
Recycled Enzyme

Figure 4-2: Second half transimination reaction of PLP with the protein's lysine. Continuation of step 1 of the transamination reaction.

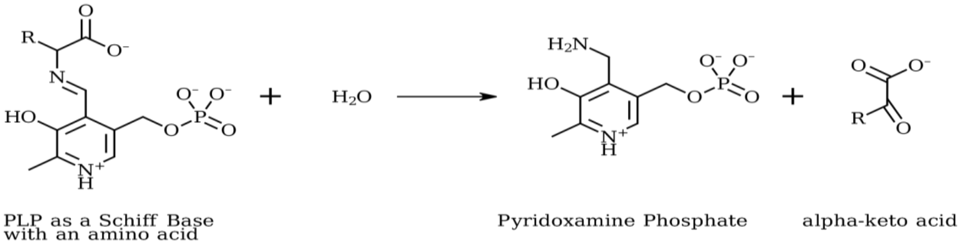


Figure 4-3: Hydrolysis and completion of transamination reaction.

4.2.2 Summary of Data and Computational Challenges

Table 4-1: Summary of PLP Swissprot Database.

Pyridoxal Phosphate Swissprot Annotation Dataset **PLP Swissprot Database** **Summary Description of Data**				
Percent Homology Threshold	**Number of Datapoints**	**Number of PLP Lysines (positive class)**	**Time BLAST (minutes)**	**Time NW (minutes)**
40	11,459	483	11	513 (measured)
50	21,595	924	44	1,822 (estimated)
60	35,218	1,549	136	4,846 (est.)
70	50,800	2,264	291	10,082 (est.)
80	65,916	2,957	481	16,975(est.)
85	**74,340**	**3,351**	**564**	**21,591 (est.)**
90	83,136	3,759	671	27,002 (est.)
95	93,236	4,198	859	33,962 (est.)

Table 4-2: Summary of PLP Swissprot Database with evidence of existence at two levels, transcript and protein level. This is a stricter level of existence that support experimentally derived results.

	Pyridoxal Phosphate Swissprot Existence at Protein or Transcript Level Annotation Dataset **Summary Description of Data**		**Pyridoxal Phosphate Swissprot Existence at Protein Level Only Annotation Dataset** **Summary Description of Data**	
Percent Homology Threshold	**Number of Datapoints**	**Number of PLP Lysines (positive class)**	**Number of Datapoints**	**Number of PLP Lysines (positive class)**
40	7,150	298	6784	279
50	10,519	426	9489	386
60	13,452	543	11841	483
70	16,236	649	13549	553
80	18,765	734	14901	606
85	**20,680**	**806**	**15980**	**647**
90	22,109	856	16851	678
95	24,268	936	18029	724

In Tables 4-1 and 4-2 we described the data used in our experiments. The 85% homology threshold is used as the representative due to it being the average homology percent of homo sapiens to mus musculus. In other words, discoveries of PLP lysines in humans would be inferred by this knowledge based system at that level. Average class imbalance is 21.6X. Running all of the experiments using the Needleman-Wunsch Algorithm would take a total of 81 days and requires more than 32 GB of RAM.

4.2.3 Dataset Creation

The Swiss-Prot Database was curated from swiss-prot extracting “N6-(pyridoxal phosphate)lysine.” search from the FT lines from the uniprot_sprot.dat found at (Uniprot 2018) on the 12th of June 2018. Furthermore, the Swiss-Prot Existence Database was created on June 30, 2018 in a similar fashion but restricted only to those proteins that have proof of existence at the protein or transcript level. The CD-HIT clustering algorithm [64] reduces the number of positives that are nearly identical. A representative of each cluster is chosen and the remainder of each cluster is discarded due to a homology threshold. Higher homology thresholds have more small clusters and lower homology thresholds have fewer large clusters. Datapoints in the dataset are those proteins that contain at least one PLP lysine PTM.

4.2.4 Relevance to Biological Workflows

Similar organisms have similar proteins and by homology predictions can be made for newly sequenced proteins, thus directing the biological experimental workflow to more distantly related organisms can be made. “On average, the protein-coding regions of the mouse and human genomes are 85 percent identical; some genes are 99 percent identical while others are only 60 percent identical” [65]. Eighty-five percent homology threshold was chosen due to the resemblance of human to mouse proteomes, but this threshold could also be useful for choosing experiments where a choice must be made between an organism and another so that highly confident results are not duplicated unnecessarily. One hundred percent homology means duplicates are allowed and therefore is not a good measure of accuracy.

4.2.5 Hypothesis

The null hypothesis is that Windowed Alignment Scoring (WAS) using a blast approximation is the same as an optimal global alignment using the Needleman-Wunsch algorithm [66] which is similar to the Smith-Waterman algorithm,[67, 68]. We will test this hypothesis using a Student's T-Test on a fraction of the PLP Swissprot database. The alternative hypothesis is that local alignment searches (BLAST)[47] on the highest scoring alignments of the data differ from a global alignment (Needleman-Wunsch) for WAS. Practically, a complete global alignment of the PLP Swissprot higher percent homology databases was not feasible, so the 40 percent homology was chosen to validate the hypothesis.

4.3 Methodology

4.3.1 The Original Engineered Feature Windowed Alignment Scoring (WAS)

A window of 100, i-50 to i+50 was chosen where i is the index of the lysine of interest. Sometimes a window could not be exactly 100 residues long. In this case, the longest sequence fragment possible was returned. Windows of sizes 50 and 200 were also experimented with but did not provide optimal results. The Needleman-Wunsch algorithm is used when the global alignment quality is of utmost importance.

Table 4-3: Example WAS features for 5 proteins A-E.

Example Windowed Alignment Scoring (WAS) Features					
	A	**B**	**C**	**D**	**E**
A	1	0.7	0.9	0.3	0.2
B	0.7	1	0.65	0.12	0.3
C	0.9	0.65	1	0.25	0.25
D	0.3	0.12	0.25	1	0.9
E	0.2	0.3	0.25	0.9	1

In Table 4-3 we show protein segment (centered on a lysine) A and C are similar because they share alignment scores with each other so they will show up in the classifier to have a high score. If A is modified, then C is likely to be modified or vice versa. However A-E and C-E are not likely to share a likelihood of being modified because of their higher Euclidean distance and dissimilarity of alignment scores. Random Forest and SVM operate similar to Euclidean distance voting of the k nearest neighbors.

Advantages include insertions and deletions common in proteins do not impact the score as dramatically as other methods like PSSM. The number of components retained was chosen optimally at 50; 25 and 100 components were chosen and did not perform as well. Truncated Singular Vector Decomposition was chosen over PCA due to its extreme efficiency on large sparse matrices. The feature generation is n^2 where n is the size of the dataset. The classifier chosen was Random Forest with 40 trees and 10 features considered at each split using the Sklearn package [69]. Validation was 10-fold cross. We illustrate the flowchart of our data mining process in Figure 4-4.

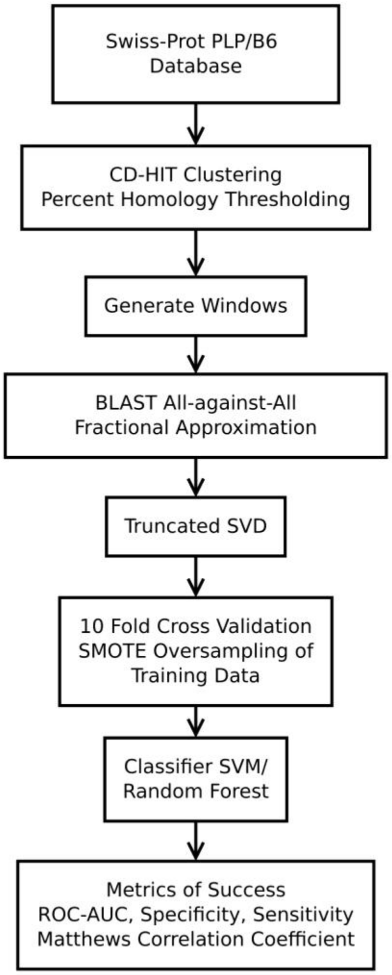

Swiss-Prot PLP/B6 Database
CD-HIT Clustering Percent Homology Thresholding
Generate Windows
BLAST All-against-All Fractional Approximation
Truncated SVD
10 Fold Cross Validation SMOTE Oversampling of Training Data
Classifier SVM/ Random Forest
Metrics of Success ROC-AUC, Specificity, Sensitivity Matthews Correlation Coefficient

Figure 4-4: Flowchart overview of experimental procedure.

4.3.2 Computational Challenges and Approximations

Because the dataset was 20:1, imbalanced synthetic minority oversampling technique (SMOTE) [70] was used only to generate additional training data and not testing data. To validate our hypothesis, we completed an entire nXn square and plot performance against percent of highest scoring alignments. Matthew's Correlation Coefficient, MCC, was the metric of choice because predicting all negative class resulted in an accuracy of 95% while the same prediction would give an MCC of 0.

When we use a sparse matrix for storing the blast results, all sequences not meeting an e-value of 0.001 are left as zero. Alternatively, we considered replacing it with the mean; further testing is needed to determine which missing values should be imputed. An e-value is like the p-value of a statistical test in its likelihood of a match occuring due to random chance. In other words, only the best matches that are 0.001 probable are entered into the sparse matrix. We considered replacing alignment scores with zero, the mean for each window overlapping by n, for both the query and subject and take the highest value other than itself for each protein as its true score in the complete global alignment scheme. By blasting there is no need to find more distantly related sequences due to the fact that only the most similar sequences are returned.

4.4 Results

Windowed Alignment Scoring based features provide a novel approach to classifying PTMs. The Lysine PLP PTM has been an interesting dataset and our hope is that the results we obtained can be generalized to other datasets. Results obtained from these methods indicate that BLAST alignments are on par with Needleman-Wunsch and Smith-Waterman Algorithms statistically. Tables 4-4, 4-5and 4-6 as well as Figures 4-5, 4-6 and 4-7 show the results of our experiments.

Table 4-4: Results from the experiment. There are ten replications. Randomness is introduced from SMOTE, shuffling cross validations and random forest. The 95% confidence intervals +/- were 0.01 or less using the following formula: 2.26 (t-distribution on 9 degrees of freedom) * s.d. / sqrt(10).

Performance Metrics using 10 fold Cross Validation for BLAST 0.5% Results Full Swissprot				
Percent Homology Threshold	**Mean MCC**	**Mean AUC**	**Mean Sensitivity**	**Mean Specificity**
40	0.44	0.92	0.57	0.96
50	0.59	0.96	0.72	0.97
60	0.68	0.98	0.81	0.98
70	0.75	0.98	0.85	0.98
80	0.78	0.99	0.88	0.98
85	**0.80**	**0.99**	**0.89**	**0.98**
90	0.81	0.99	0.90	0.99
95	0.83	0.99	0.91	0.99

Table 4-5: Same as the above table but results from the transcript and protein level experimental existence.

Performance Metrics using 10 fold Cross Validation for BLAST 0.5% Results Swissprot Protein and Transcript Evidence Only				
Percent Homology Threshold	**Mean MCC**	**Mean AUC**	**Mean Sensitivity**	**Mean Specificity**
40	0.36	0.88	0.5	0.95
50	0.49	0.93	0.61	0.96
60	0.54	0.95	0.67	0.97
70	0.59	0.96	0.72	0.97
80	0.62	0.97	0.75	0.97
85	**0.65**	**0.97**	**0.78**	**0.98**
90	0.64	0.97	0.78	0.97
95	0.68	0.97	0.82	0.98

Table 4-6: Same as before with results from the protein level experimental existence only.

Performance Metrics using 10 fold Cross Validation for BLAST 0.5% Results Swissprot Protein Evidence Only				
Percent Homology Threshold	**Mean MCC**	**Mean AUC**	**Mean Sensitivity**	**Mean Specificity**
40	0.34	0.86	0.46	0.95
50	0.43	0.91	0.57	0.96
60	0.51	0.94	0.65	0.96
70	0.57	0.95	0.70	0.97
80	0.59	0.96	0.73	0.97
85	**0.60**	**0.96**	**0.73**	**0.97**
90	0.63	0.97	0.76	0.97
95	0.64	0.97	0.78	0.97

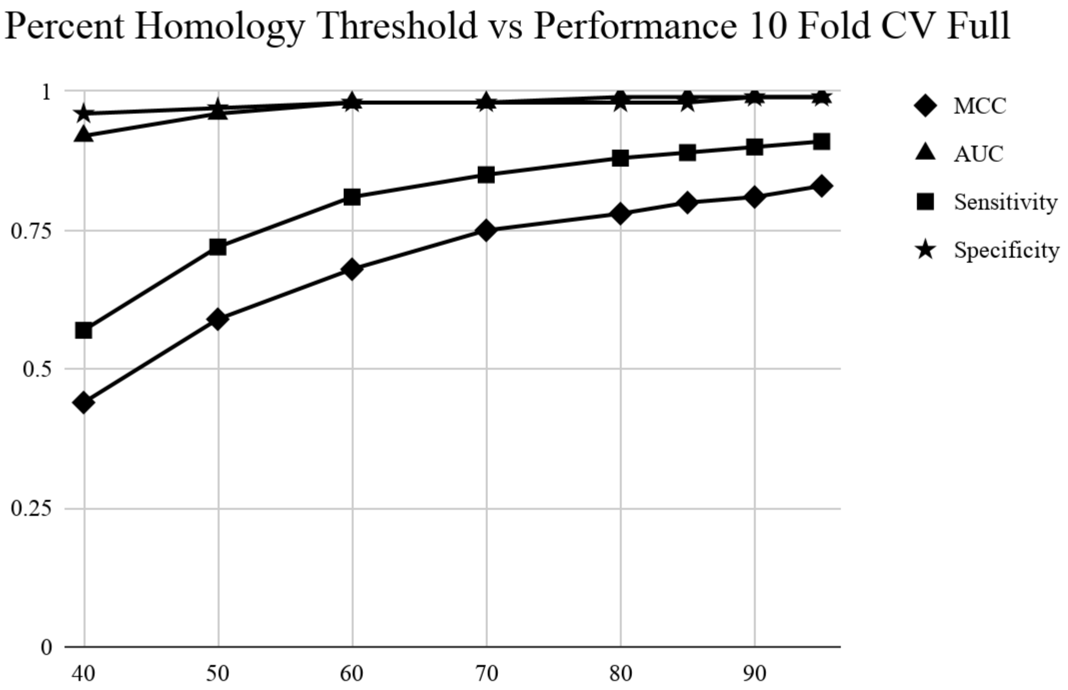


Figure 4-5: The relationship between homology level of dataset to performance. The y-axis is the 10-fold cross validation metric of success and the x-axis is the percent homology threshold from the CD-HIT clustering.

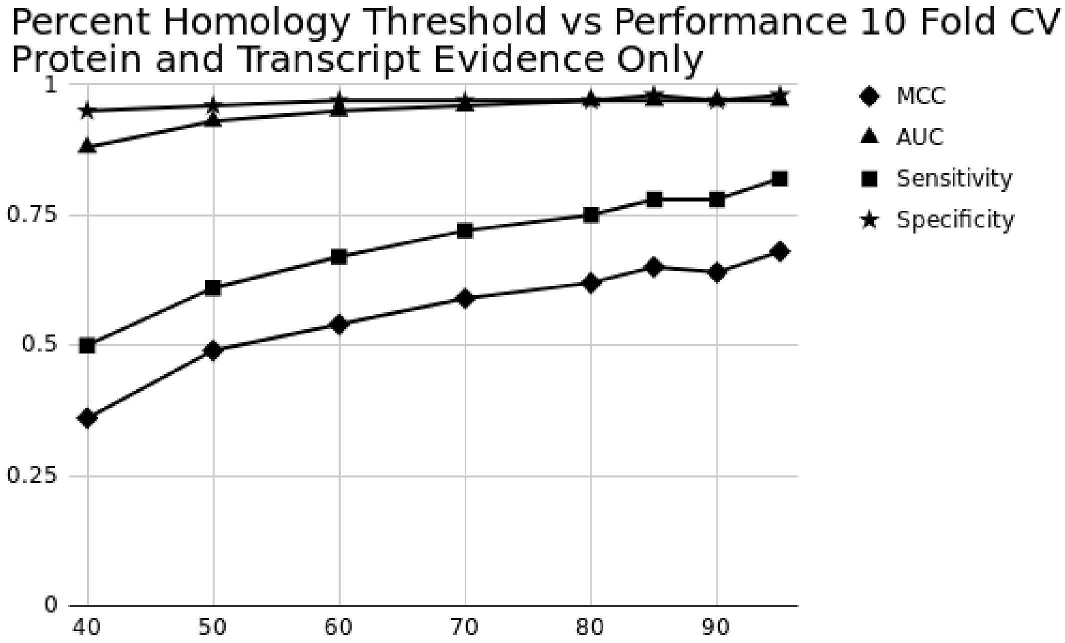


Figure 4-6: The relationship between homology level of dataset to performance for protein and transcript level existence. Same description as prior figure.

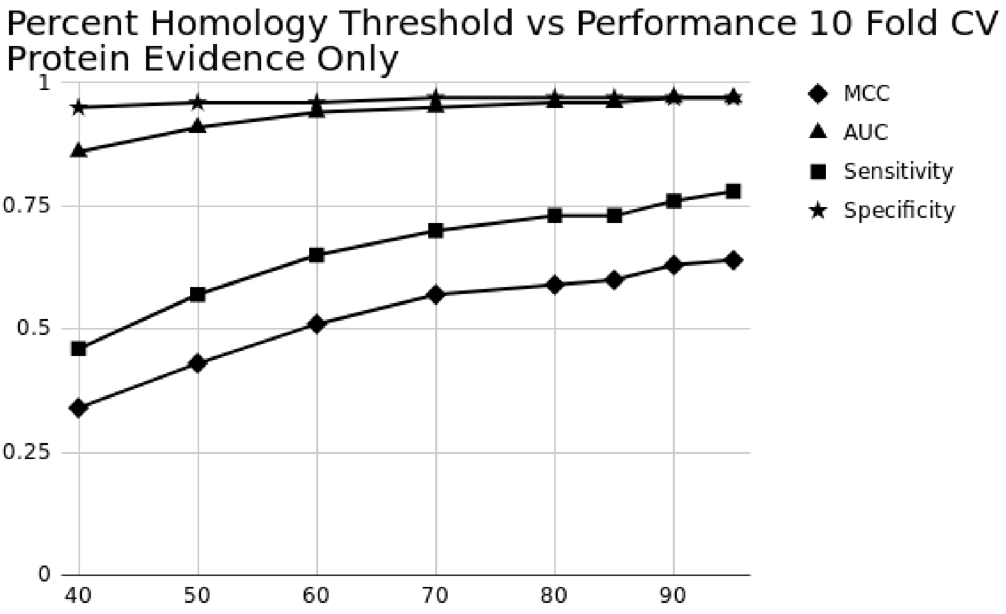


Figure 4-7: The relationship between homology level of dataset to performance for protein and transcript level existence. Same description as before.

Eighty-five percent homology threshold is a fair assessment of the performance because that is the level of homology between humans and mouse protein coding genomes. In other words, mouse Active Vitamin B6 (PLP) Lysine PTMs should also occur in humans with a sensitivity of 0.89 and a specificity of 0.98.

Table 4-7: The performance at the 40% homology level of the dataset comparing the two algorithms BLAST and Needleman-Wunsch, suggesting that BLAST, although not quite as effective as NW, is a good approximation saving computational time.

40% Homology Threshold Retaining Top Percent Similarities for Needleman-Wunsch				
NW Top % Similarities	**MCC**	**AUC**	**Sensitivity**	**Specificity**
0.1%	**0.2**	**0.74**	**0.33**	**0.93**
0.5%	**0.45**	**0.92**	**0.57**	**0.96**
1%	**0.47**	**0.93**	**0.55**	**0.97**
10%	0.44	0.92	0.53	0.97
20%	0.41	0.92	0.5	0.97
30%	0.42	0.91	0.5	0.97
40%	0.43	0.92	0.52	0.97
50%	0.4	0.9	0.48	0.97
60%	0.41	0.91	0.49	0.97
70%	0.37	0.91	0.45	0.97
80%	0.4	0.91	0.48	0.97
90%	0.41	0.92	0.49	0.97
100%	0.42	0.93	0.48	0.97
BLAST 0.1%	**0.12**	**0.67**	**0.25**	**0.92**
BLAST 0.5%	**0.44**	**0.92**	**0.57**	**0.96**
BLAST 1%	**0.45**	**0.93**	**0.56**	**0.97**

Interestingly, without Truncated SVD decomposition, the results are 0.10

MCC lower for NW even though 10X more trees were used. More trees did not improve performance. Also noteworthy is that the lower scoring alignments (Higher NW Top % similarities) reduced the performance of the model. This suggests that lower scoring alignments add unnecessary noise to the model and only top alignments should be used. Because 0.001 results in a large performance degradation, it seems there are too many alignments discarded. Unlike BLAST, Needleman-Wunsch must compute the full similarity matrix ascertain the top percent retained. Table 4-7 shows the difference between Needleman Wunsch and BLAST. Figure 4-8 shows how retaining only the top percent affects metrics of success.

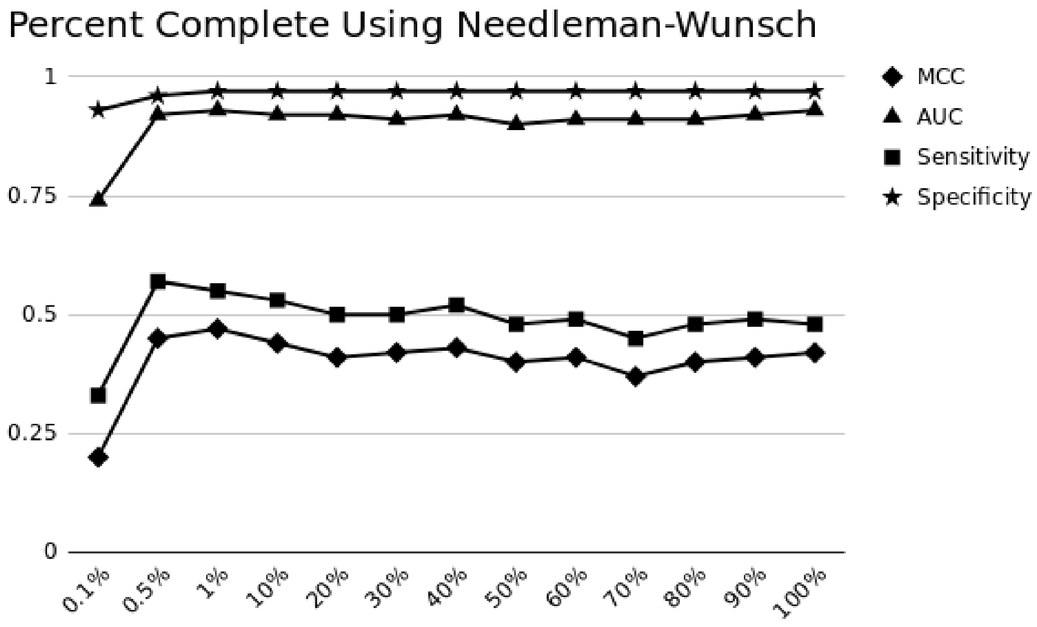


Figure 4-8: Comparing NW to BLAST performance metrics. The y-axis is the of success for a ten fold cross validation and the x-axis is the top scoring ities retained.

Although Needleman-Wunsch Global Alignments have slightly better scores than BLAST, the difference is not significant; thus, we fail to reject the null hypothesis. The speedup approximation by using BLAST seems appropriate.

A paired t-test was performed at the 0.1%, 0.5% and 1% level for Needleman-Wunsch and 0.1% 0.5% and 1% for BLAST using the e-values of 1e-20, 1e-3, and 1.

```
data: c(0.12, 0.44, 0.45) and c(0.2, 0.45, 0.47)
t = -1.6775, df = 2, p-value = 0.2354
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.13071460 0.05738127
sample estimates:
mean of the differences
-0.03666667
```

We found works in related areas of Lysine PTM research in Table 4-8.

Table 4-8: Prior Works and their salient contributions.

Prior Studies for Prediction of Lysine Post-translational Modifications			
Body of Work	**Features/Classifier**	**Numerosity**	**Significance**
iSuc-PseAAC [71] **Lysine Succinylation**	PseAAC SVM	26,649	Peptide Position-Specific Propensity
NetGlycate [72] **Lysine Glycation**	Lysine Position and Amino Acid Composition Neural Networks	215	Use of Balloting of Votes from Ensemble of Neural Networks
LysAcet [73] **Lysine Acetylation**	Protein Sequence Coupling Patterns SVM	11,474	Innovated Coupling Pattern Features
RUBI [74, 75] **Lysine Ubiquitination**	SVM Bidirectional Recurrent Neural Networks Multiple Sequence Alignment Frequencies	304,443	Most Datapoints
This Work **Lysine Pyridoxal-5-Phosphate**	BLAST, Random Forest, Truncated SVD	74,340	Introduced Windowed Alignment Scoring and PLP Swissprot Database

4.5 Conclusion

A paired student’s t-test was performed comparing the BLAST approximations to the NW Global Alignments at percent complete. There is not a statistical difference between the two methods p-value = 0.24. This indicates that the heuristics and approximations that we have chosen are suitable. Thus, we fail to reject the null hypothesis The standard methods of finding lysine-PLP PTMs is time consuming and is the least preferred method. A relatively quick and inexpensive data mining approach using the engineered feature source WAS can help to redirect biological workflows so that more distantly related and therefore less well characterized proteins can be done. Our approach will then fill in the gaps. It can be seen that at the 85% homology level using BLAST, we are able to obtain a MCC of 0.88 and AUC of 0.99 on the full Swiss-Prot database. These results indicate that our method can generalize well to unknown lysine-PLP PTMs.

.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

Our goal in this work is to develop novel solutions in feature engineering through data mining frameworks on the problem of predicting post-translational modifications of proteins. We have tested each approach on multiple datasets and therefore believe each has the potential to be applied to new problems such as genetic sequences. We have performed statistical tests in each of the major chapters, thus providing a solid basis for other researchers to adopt our work and even improve it. Some of the specific contributions and results are as follows:

5.1 Contribution to Feature Engineering with LSM and the Cysteine Disulfide Connectivity Problem

The Local Similarity Matrix based feature engineering is an innovative solution to a well known problem, cysteine connectivity. We have presented an entire framework that encompasses other's works and have shown statistical significance of our work compared to all other works (except on one dataset). PSSM has appeal beyond cysteine connectivity, and the verification of our results on three datasets indicates it may compete with a thirty year established use of PSSM.

5.2 Contribution to Feature Engineering with RAM and the Cysteine Redox Susceptibility Problem

The Residue Adjacency Matrix is an approach that was built upon the work of others. The nearest n cysteines to a possible PTM were calculated and found to be useful. We expanded upon this idea and found the n nearest of each residue to a potential PTM. This approach to feature engineering could be expanded to other techniques. By broadening a subset of any problem, it is possible to attain a result that has potential beyond its capability. The cysteine oxidation problem is crucial to the treatment of oxidative stress diseases such as cancer, diabetes and heart disease. By improving the quality of predictions for which cysteines undergo oxidation, researchers can use this as the basis of discovering treatments.

5.3 Contribution to Feature Engineering with WAS and the Lysine Pyridoxal-5-Phosphate Prediction Problem

One of the core challenges of making predictions is to work on reasonable timeframes. An experiment that takes a quarter of a year is not workable for others to build upon. We have found an approximation that takes two days and gives room for further experimentation. This approximation did not significantly differ in terms of performance from the other methods tried; thus, its use is justified. Researchers that are utilizing the lysine-PLP pathway in their drug design or otherwise treating illness may incorporate our work to further their progress.

5.4 Future Work

New datasets for new problems in proteomics and genetics are a promising avenue where we could focus our future efforts. There may even be applications in any sequence data; for instance, natural language processing and time series analysis both operate on sequences. We hope that these methods can be applied in many different ways and not just in the bioinformatics community.

APPENDIX A

BLAST AND DATA CHARACTERISTICS

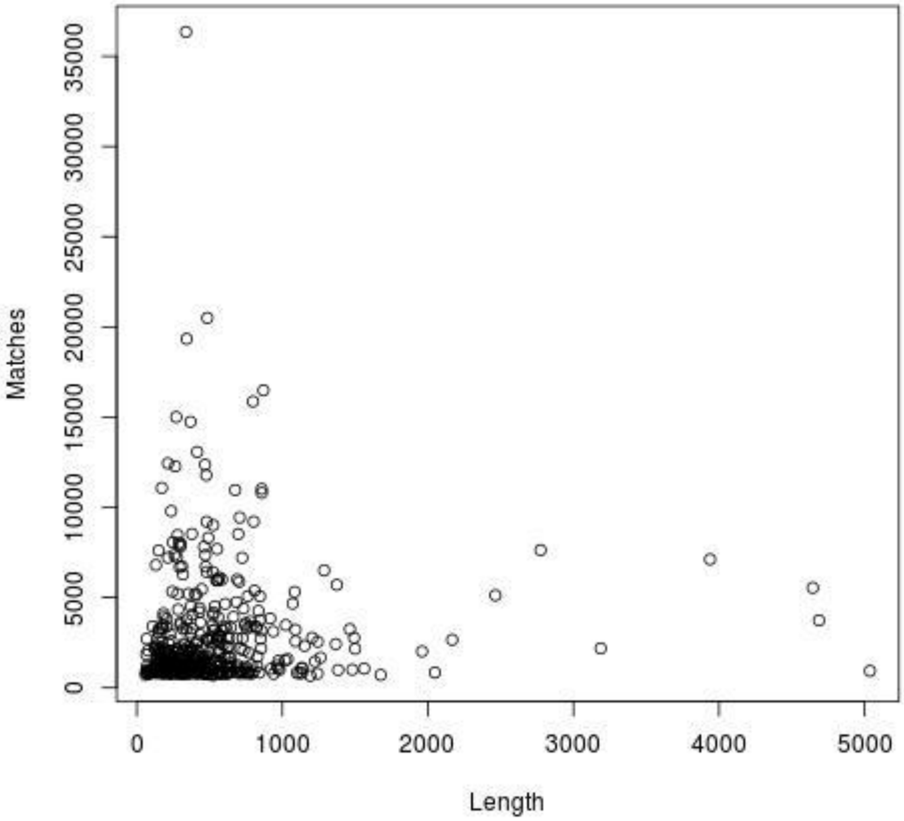


Figure A-1: Protein length does not affect the number of matches returned by a BLAST Search. Based on a linear regression mode,l the length of a protein predicts 0.3% of the variability of the number of BLAST matches returned. The p-value is not significant at the 0.1 significance level. Thus the length of a protein is not a good predictor of the number of matches returned by BLAST in which the adjusted R-squared is 0.003. The figure is for the RSC758 dataset.

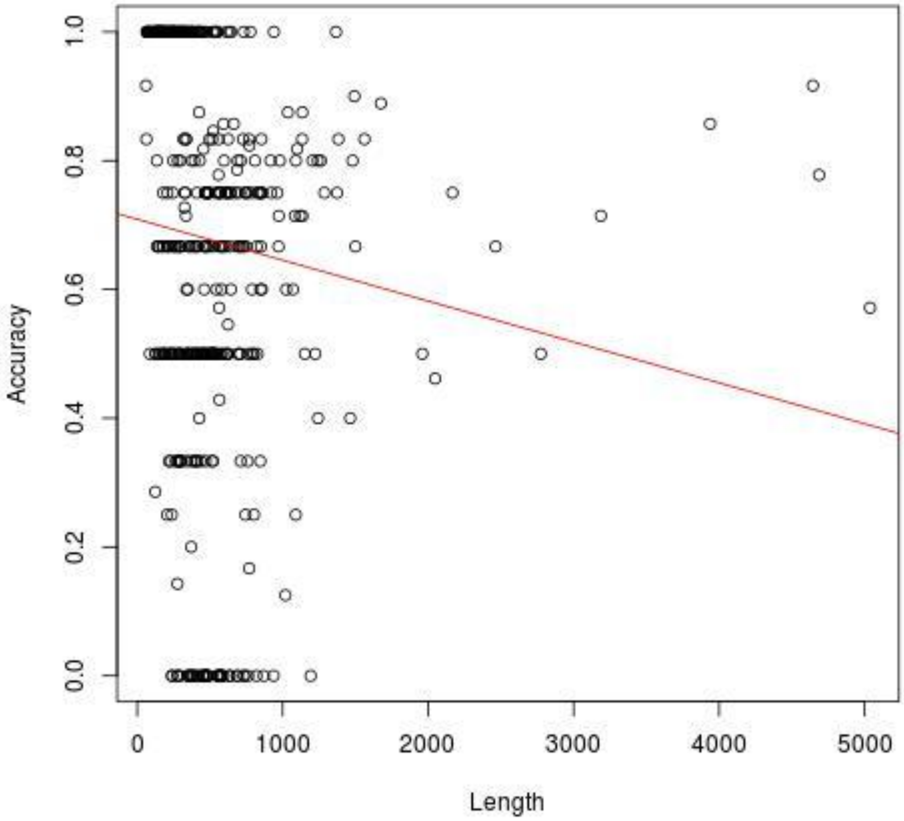


Figure A-2: Protein length predicts accuracy negatively for cysteine redox susceptibility on the RSC758 dataset with a p-value of 0.02, but this only explains 1% of the variability in the data where the adjusted R-squared is 0.01 and the p-value is 0.02. The mean number of amino acids in a protein is 525.8.

APPENDIX B

CYSTEINE SEPARATION PROFILES FOR THE LOCAL SIMILARITY MATRIX

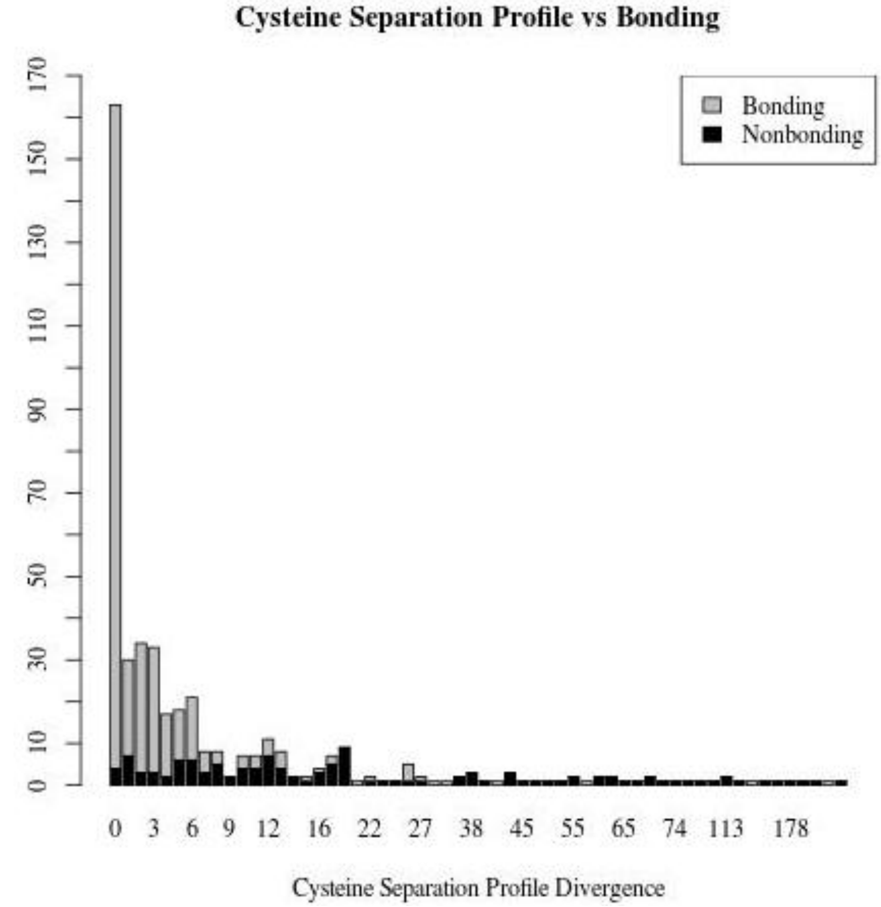


Figure B-3: Histogram showing the number of proteins at each divergence separated by bonding and nonbonding for SP39. There is a high degree of homology because most of the proteins had low divergence and were bonding. This was not the case for PDBCYS and IVD-54. Qualitatively, this shows the dataset has the potential to be solved at a higher Qp and Qc metric of success.

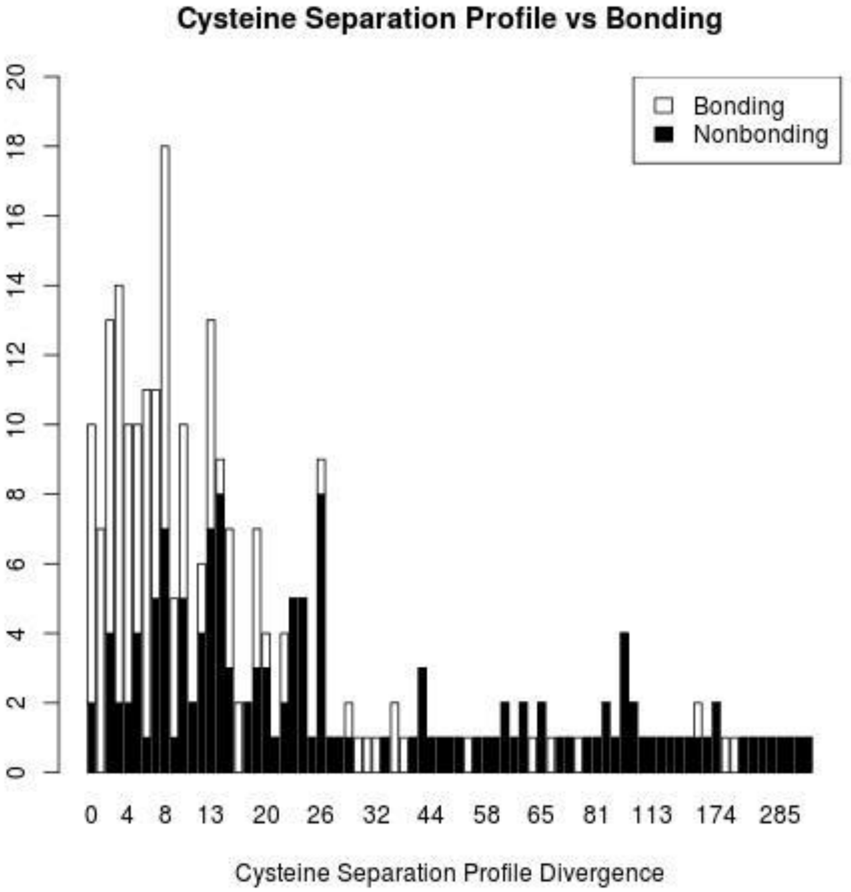


Figure B-4: Cysteine separation profile divergence and bonding for PDBCYS-R. Lower homology is noted by the low divergences which do not make up a majority of the data as they did for SP39. This qualitative fact indicates a more challenging dataset than SP39.

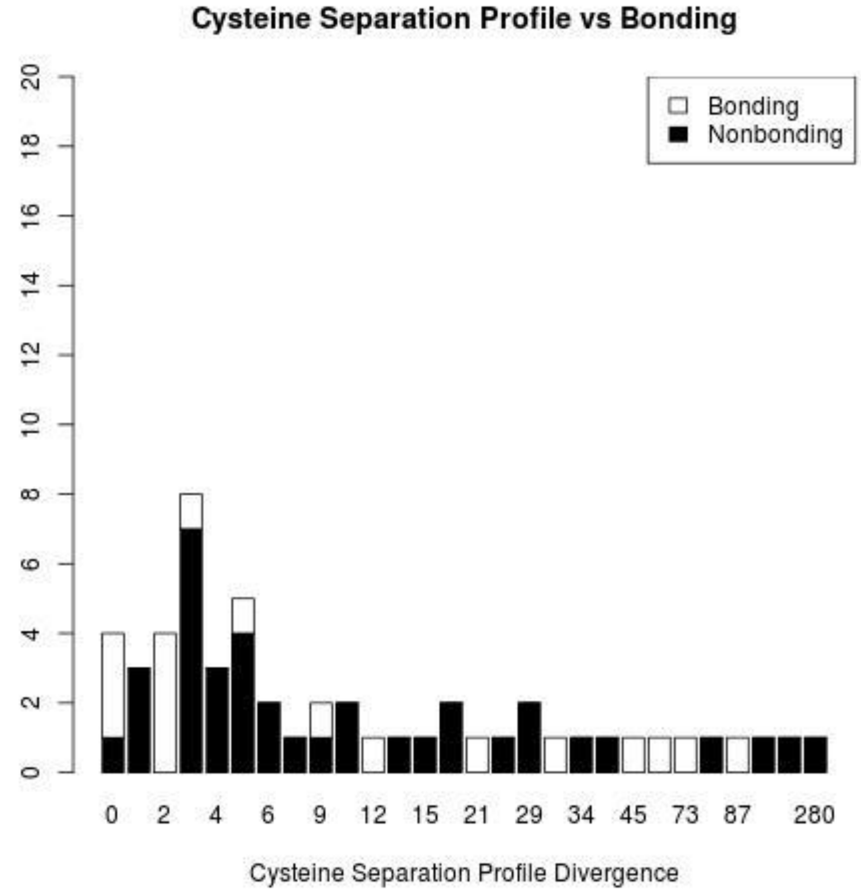


Figure B-5: Cysteine separation profile divergence and bonding for IVD-54. The least low divergences of the three datasets. This qualitatively indicates the most challenging dataset confirmed by Qp and Qc metrics of success.

APPENDIX C

TABLES COMPARING THE LOCAL SIMILARITY MATRIX TO PREVIOUS WORKS

Table C-1: Data describing the prior works results on SP39 dataset compared to Local Similarity Matrix.

B	**PreCYS** [21]	**CH Lu Et al.** [26]	**J.Song et al.** [25]	**Zhu Et al.** [76]	**Target Disulfide** [28]	**HL Lin Et al.** [77]	**Our Work**
2	79	85.7	86.5	85.3	92.3	94.2	94.9
3	53	74.6	67.1	69.9	78.1	89.0	88.4
4	55	63.2	78.8	79.7	82.8	90.9	88.9
5	58	47.6	46.8	55.9	62.2	86.7	86.7
2-5	63	73.9	74.4	76.0	82.5	91.0	90.6

Table C-2: Data describing the prior works results on PDBCYS dataset compared to Local Similarity Matrix.

B	**Dislocate** [78]	**C. Savojardo Et Al.** [79]	**Target Disulfide** [28]	**Cyscon** [7]	**Our Work**
2	75	76.0	83.0	N/A	90
3	48	55.3	76.4	N/A	80
4	44	51.2	53.7	N/A	68.3
5	19	32.4	21.6	N/A	70.3
2-5	54	59.3	67.7	72.3	80.6

Table C-3: Data describing the prior works results on IVD-54 dataset compared to Local Similarity Matrix.

B	**Dianna** [8]	**DBCP** [77]	**Disulfind** [9]	**Target Disulfide** [28]	**Our Work**
2	10.3	10.3	13.8	69.0	79.3
3	0.0	13.3	13.3	66.7	80.0
4	0.0	0.0	0.0	14.3	14.3
5	0.0	0.0	0.0	0.0	33.3
2-5	5.6	9.3	11.1	57.4	68.5

APPENDIX D

VISUALIZATION, FEATURE CORRELATIONS AND PRIOR WORK COMPARISONS FOR RESIDUE ADJACENCY MATRIX

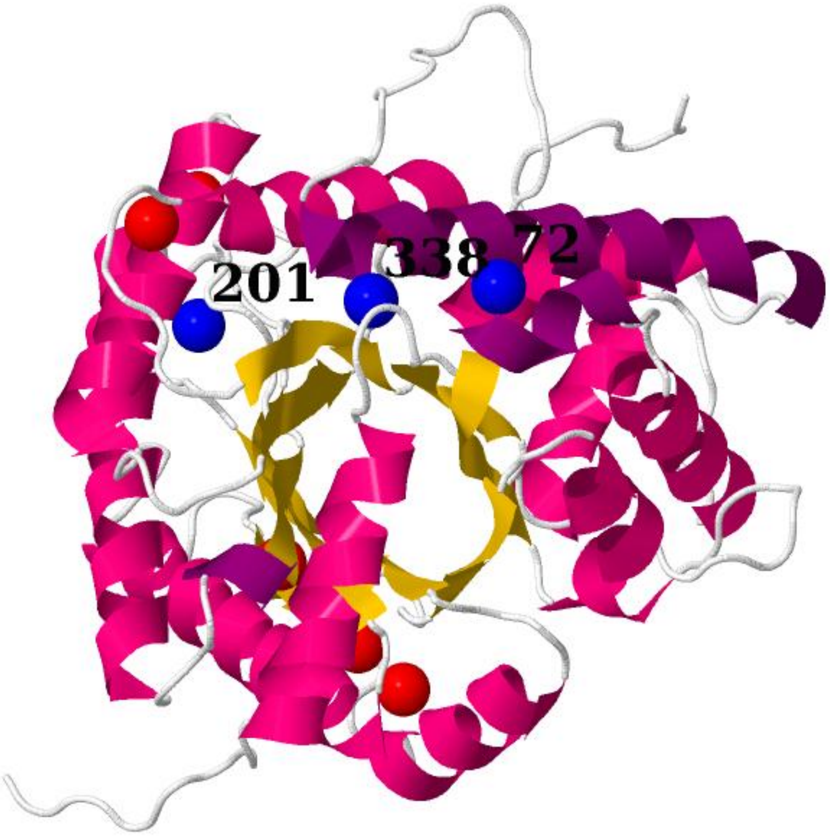


Figure D-6: In the image above, the sulfur atoms of reactive cysteines (residues 201, 338, and 72) in the protein 1ADO are emphasized with a blue sphere. The red spheres in the protein correspond to the sulfur atoms of the non-reactive cysteines.

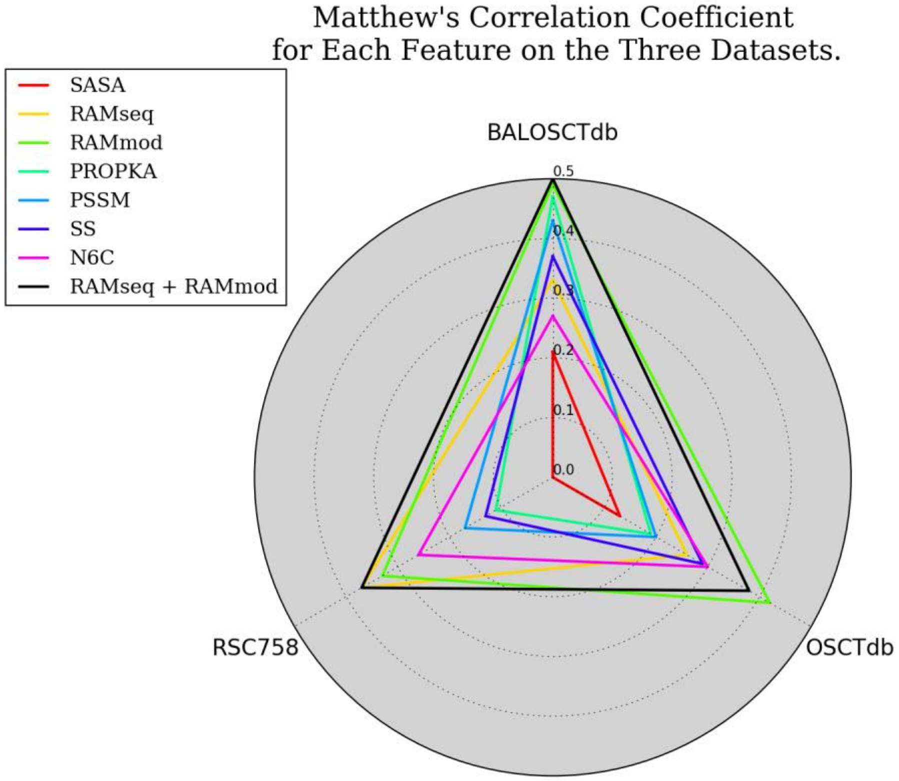


Figure D-7: Note that the radar chart shows that RAM sits on the outer edges of the chart compared to other features. This indicates that the features have a higher performance on every dataset compared to all features in prior works.

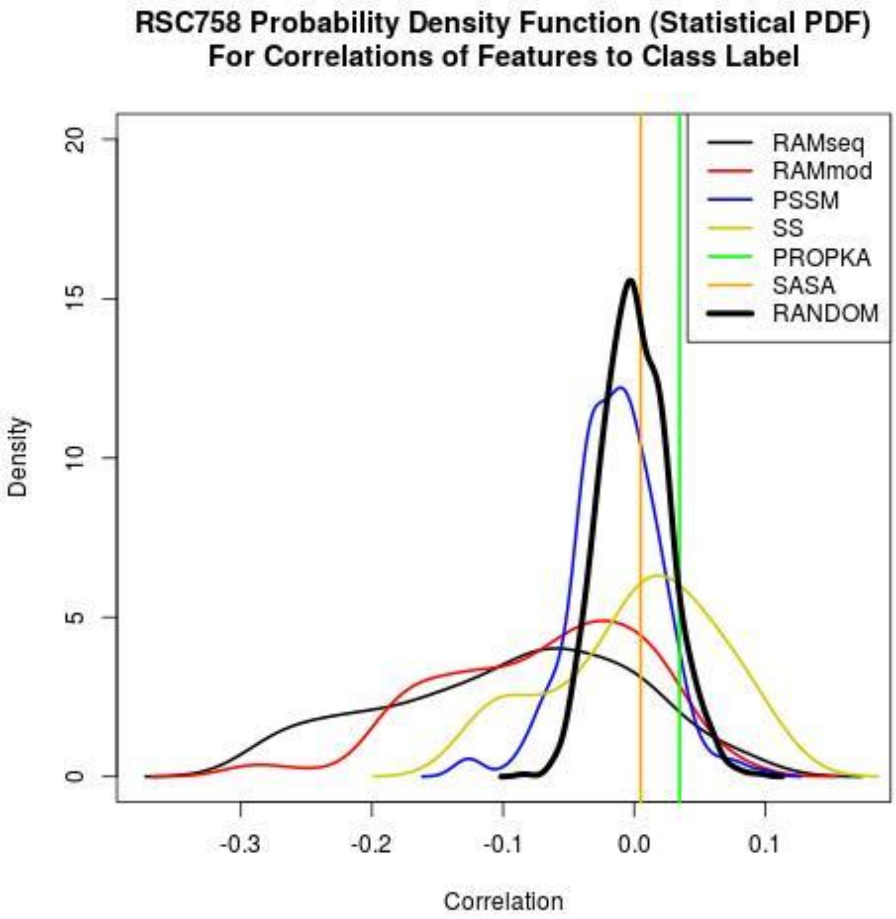


Figure D-8: Shown below is the probability density function approximated using the statistical software R. The density function in the stats package was used with default parameters. Note the vertical lines for SASA and PROPKA are one dimensional features therefore, the correlation pdf is a vertical line

The probability of a correlation existing in a range is found by taking the integral between the min and max of any two points. With this in mind, the plot indicates that RAMmod and RAMseq have a large probability of a correlation with the class label, albeit negative. This suggests that the goodness of the features can be observed using non-classification tools such as ordinary least squares (OLS). It is important to note that each curve has an area below it equal to one.

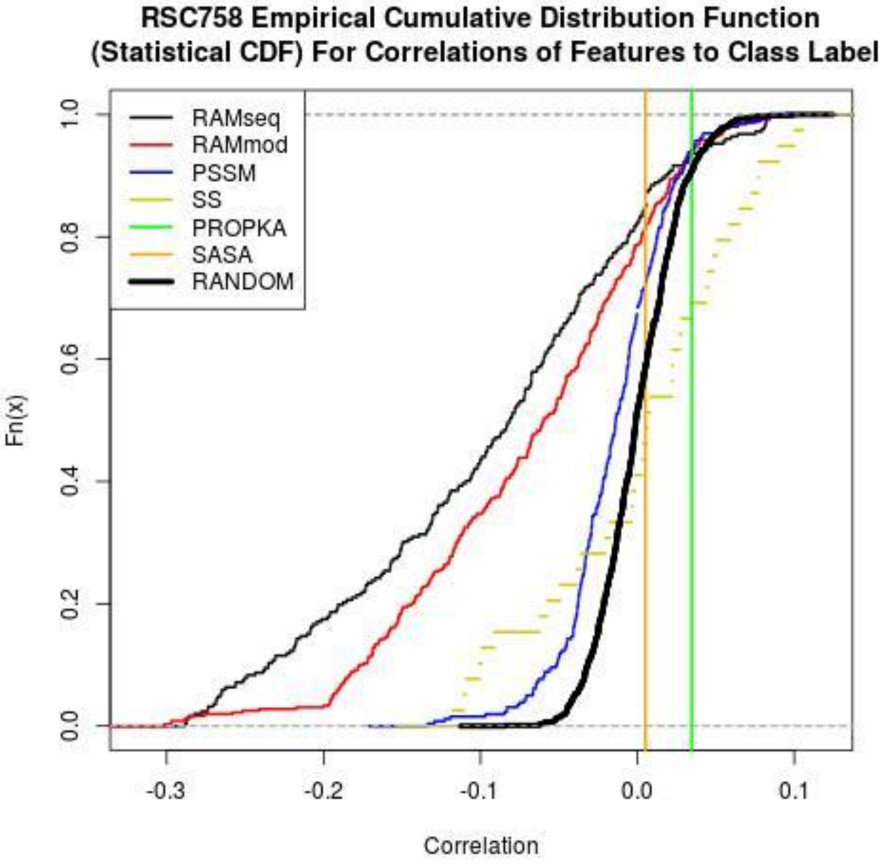


Figure D-9: By transforming the PDF to a CDF we can see the probability of a feature's correlation being equal or less than a particular value. We make this transform so that we can run a statistical test, the Two Sample Kolmogorov-Smirnov test or simply KS test. Our p-value is < 2.2e-16 comparing RANDOM to RAMseq and RANDOM to RAMmod.

BIBLIOGRAPHY

[1] D. C. Montgomery, Design and Analysis of Experiments, Eighth ed., John Wiley & Sons, 2013.

[2] M. D. Scott and J. Frydman, "Aberrant protein folding as the molecular basis of cancer," in Protein misfolding and disease, Springer, 2003, pp. 67-76.

[3] J. S. Pattison and J. Robbins, "Protein misfolding and cardiac disease: establishing cause and effect," Autophagy, vol. 4, no. 6, pp. 821-823, 2008.

[4] M. Stefani and C. M. Dobson, "Protein aggregation and aggregate toxicity: new insights into protein folding, misfolding diseases and biological evolution," Journal of molecular medicine, vol. 81, no. 11, pp. 678-699, 2003.

[5] M. J. Harvey and G. De Fabritiis, "High-throughput molecular dynamics: the powerful new tool for drug discovery," Drug discovery today, vol. 17, no. 19-20, pp. 1059-1062, 2012.

[6] A. L. Beberg, D. L. Ensign, G. Jayachandran, S. Khaliq and V. S. Pande, "Folding@ home: Lessons from eight years of volunteer distributed computing," 2009.

[7] J. Yang, B.-J. He, R. Jang, Y. Zhang and H.-B. Shen, "Accurate disulfide-bonding network predictions improve ab initio structure prediction of cysteine-rich proteins," Bioinformatics, vol. 31, no. 23, pp. 3773-3781, 2015.

[8] F. Ferrè and P. Clote, "DiANNA 1.1: an extension of the DiANNA web server for ternary cysteine classification," Nucleic acids research, vol. 34, no. suppl_2, pp. W182--W185, 2006.

[9] A. Ceroni, A. Passerini, A. Vullo and P. Frasconi, "DISULFIND: a disulfide bonding state and cysteine connectivity prediction server," Nucleic acids research, vol. 34, no. suppl_2, pp. W177--W181, 2006.

[10] D. Tessier, B. Bardiaux, C. Larré and Y. Popineau, "Data mining techniques to study the disulfide-bonding state in proteins: signal peptide is a strong descriptor," Bioinformatics, vol. 20, no. 16, pp. 2509-2512, 2004.

[11] J. Cheng, M. J. Sweredoski and P. Baldi, "Accurate prediction of protein disordered regions by mining protein structure data," Data mining and knowledge discovery, vol. 11, no. 3, pp. 213-222, 2005.

[12] V. J. Kartik, T. Lavanya and K. Guruprasad, "Analysis of disulphide bond connectivity patterns in protein tertiary structure," International journal of biological macromolecules, vol. 38, no. 3-5, pp. 174-179, 2006.

[13] N. Eswar, B. Webb, M. A. Marti-Renom, M. S. Madhusudhan, D. Eramian, M.-y. Shen, U. Pieper and A. Sali, "Comparative protein structure modeling using Modeller," Current protocols in bioinformatics, vol. 15, no. 1, pp. 5-6, 2006.

[14] E. Zhao, H.-L. Liu, C.-H. Tsai, H.-K. Tsai, C.-h. Chan and C.-Y. Kao, "Cysteine separations profiles on protein sequences infer disulfide connectivity," Bioinformatics, vol. 21, no. 8, pp. 1415-1420, 2004.

[15] P. Fariselli, P. Riccobelli and R. Casadio, "Role of evolutionary information in predicting the disulfide-bonding state of cysteine in proteins," Proteins: Structure, Function, and Bioinformatics, vol. 36, no. 3, pp. 340-346, 1999.

[16] P. Fariselli and R. Casadio, "Prediction of disulfide connectivity in proteins," Bioinformatics, vol. 17, no. 10, pp. 957-964, 2001.

[17] S. M. Muskal, S. R. Holbrook and S.-H. Kim, "Prediction of the disulfide-bonding state of cysteine in proteins," Protein engineering, design and selection, vol. 3, no. 8, pp. 667-672, 1990.

[18] A. Fiser, M. Cserzö, É. Tüdös and I. Simon, "Different sequence environments of cysteines and half cystines in proteins application to predict disulfide forming residues," Febs Letters, vol. 302, no. 2, pp. 117-120, 1992.

[19] A. Fiser and I. Simon, "Predicting the oxidation state of cysteines by multiple sequence alignment," Bioinformatics, vol. 16, no. 3, pp. 251-256, 2000.

[20] A. Vullo and P. Frasconi, "Disulfide connectivity prediction using recursive neural networks and evolutionary information," Bioinformatics, vol. 20, no. 5, pp. 653-659, 2004.

[21] C.-H. Tsai, B.-J. Chen, C.-h. Chan, H.-L. Liu and C.-Y. Kao, "Improving disulfide connectivity prediction with sequential distance between oxidized cysteines," Bioinformatics, vol. 21, no. 24, pp. 4416-4419, 2005.

[22] F. Ferrè and P. Clote, "Disulfide connectivity prediction using secondary structure information and diresidue frequencies," Bioinformatics, vol. 21, no. 10, pp. 2336-2346, 2005.

[23] L. J. McGuffin, K. Bryson and D. T. Jones, "The PSIPRED protein structure prediction server," Bioinformatics, vol. 16, no. 4, pp. 404-405, 2000.

[24] B.-J. Chen, C.-H. Tsai, C.-h. Chan and C.-Y. Kao, "Disulfide connectivity prediction with 70% accuracy using two-level models," PROTEINS: Structure, Function, and Bioinformatics, vol. 64, no. 1, pp. 246-252, 2006.

[25] J. Song, Z. Yuan, H. Tan, T. Huber and K. Burrage, "Predicting disulfide connectivity from protein sequence using multiple sequence feature vectors and secondary structure," Bioinformatics, vol. 23, no. 23, pp. 3147-3154, 2007.

[26] C.-H. Lu, Y.-C. Chen, C.-S. Yu and J.-K. Hwang, "Predicting disulfide connectivity patterns," Proteins: Structure, Function, and Bioinformatics, vol. 67, no. 2, pp. 262-270, 2007.

[27] H.-H. Lin, J.-C. Hsu and Y.-F. Chen, "Disulfide bonding pattern prediction server based on normalized pair distance by MODELLER," in Computer, Consumer and Control (IS3C), 2012 International Symposium on, 2012.

[28] D.-J. Yu, Y. Li, J. Hu, X. Yang, J.-Y. Yang and H.-B. Shen, "Disulfide connectivity prediction based on modelled protein 3D structural information and random forest regression," IEEE/ACM transactions on computational biology and bioinformatics, vol. 12, no. 3, pp. 611-621, 2015.

[29] J. Edmonds, "Paths, trees, and flowers," Canadian Journal of mathematics, vol. 17, no. 3, pp. 449-467, 1965.

[30] H. N. Gabow, "An efficient implementation of Edmonds' algorithm for maximum matching on graphs," Journal of the ACM (JACM), vol. 23, no. 2, pp. 221-234, 1976.

[31] A. Liaw, M. Wiener and Others, "Classification and regression by randomForest," R news, vol. 2, no. 3, pp. 18-22, 2002.

[32] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5-32, 2001.

[33] L. F. R. A. Torgo, "Inductive learning of tree-based regression models," 1999.

[34] M. P. Murphy, "Mitochondrial thiols in antioxidant protection and redox signaling: distinct roles for glutathionylation and other thiol modifications," Antioxidants & redox signaling, vol. 16, no. 6, pp. 476-495, 2012.

[35] C. Klomsiri, P. A. Karplus and L. B. Poole, "Cysteine-based redox switches in enzymes," Antioxidants & redox signaling, vol. 14, no. 6, pp. 1065-1077, 2011.

[36] S. M. Marino and V. N. Gladyshev, "Analysis and functional prediction of reactive cysteine residues," Journal of Biological Chemistry, vol. 287, no. 7, pp. 4419-4425, 2012.

[37] N. M. Giles, A. B. Watts, G. I. Giles, F. H. Fry, J. A. Littlechild and C. Jacob, "Metal and redox modulation of cysteine protein function," Chemistry & biology, vol. 10, no. 8, pp. 677-693, 2003.

[38] K. G. Reddie and K. S. Carroll, "Expanding the functional diversity of proteins through cysteine oxidation," Current opinion in chemical biology, vol. 12, no. 6, pp. 746-754, 2008.

[39] D. E. Fomenko, S. M. Marino and V. N. Gladyshev, "Functional diversity of cysteine residues in proteins and unique features of catalytic redox-active cysteines in thiol oxidoreductases," Molecules and Cells, vol. 26, no. 3, pp. 228-235, 2008.

[40] H.-m. Lee, K. J. Dietz and R. Hofestädt, "Prediction of thioredoxin and glutaredoxin target proteins by identifying reversibly oxidized cysteinyl residues.," Journal of Integrative Bioinformatics, vol. 7, no. 3, 2010.

[41] P.-T. Doulias, J. L. Greene, T. M. Greco, M. Tenopoulou, S. H. Seeholzer, R. L. Dunbrack and H. Ischiropoulos, "Structural profiling of endogenous S-nitrosocysteine residues reveals unique features that accommodate diverse mechanisms for protein S-nitrosylation," Proceedings of the National Academy of Sciences of the United States of America, vol. 107, no. 39, pp. 16958-16963, 2010.

[42] G. Roos, N. Foloppe and J. Messens, "Understanding the pKa of Redox Cysteines: The Key Role of Hydrogen Bonding," Antioxidants & Redox Signaling, vol. 18, no. 1, pp. 94-127, 2013.

[43] A. Zeida, C. M. Guardia, P. Lichtig, L. L. Perissinotti, L. A. Defelipe, A. Turjanski, R. Radi, M. Trujillo and D. A. Estrin, "Thiol redox biochemistry: insights from computer simulations," Biophysical Reviews, vol. 6, no. 1, pp. 27-46, 2014.

[44] M.-a. Sun, Q. Zhang, Y. Wang, W. Ge and D. Guo, "Prediction of redox-sensitive cysteines using sequential distance and other sequence-based features," BMC bioinformatics, vol. 17, no. 1, p. 316, 2016.

[45] I. Soylu and S. M. Marino, "Cp i pe: a comprehensive computational platform for sequence and structure-based analyses of Cysteine residues," Bioinformatics, vol. 33, no. 15, pp. 2395-2396, 2017.

[46] R. Sanchez, M. Riddle, J. Woo and J. Momand, "Prediction of reversibly oxidized protein cysteine thiols using protein structure properties," Protein Science, vol. 17, no. 3, pp. 473-481, 2008.

[47] S. F. Altschul, W. Gish, W. Miller, E. W. Myers and D. J. Lipman, "Basic local alignment search tool," Journal of molecular biology, vol. 215, no. 3, pp. 403-410, 1990.

[48] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks," Proceedings of the National Academy of Sciences of the United States of America, vol. 89, no. 22, pp. 10915-10919, 1992.

[49] G. D. Stormo, T. D. Schneider, L. Gold and A. Ehrenfeucht, "Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli," Nucleic Acids Research, vol. 10, no. 9, pp. 2997-3011, 1982.

[50] A. E. Márquez-Chamorro and J. S. Aguilar-Ruiz, "Soft computing methods for disulfide connectivity prediction," Evolutionary Bioinformatics, vol. 11, pp. EBO--S25349, 2015.

[51] C. R. Søndergaard, M. H. M. Olsson and J. H. Rostkowski Michałand Jensen, "Improved treatment of ligands and coupling effects in empirical calculation and rationalization of p K a values," Journal of chemical theory and computation, vol. 7, no. 7, pp. 2284-2295, 2011.

[52] H. Li, A. D. Robertson and J. H. Jensen, "Very fast empirical prediction and rationalization of protein pKa values," Proteins: Structure, Function, and Bioinformatics, vol. 61, no. 4, pp. 704-721, 2005.

[53] M. H. M. Olsson, C. R. Søndergaard, M. Rostkowski and J. H. Jensen, "PROPKA3: consistent treatment of internal and surface residues in empirical p K a predictions," Journal of chemical theory and computation, vol. 7, no. 2, pp. 525-537, 2011.

[54] S. Mitternacht, "FreeSASA: An open source C library for solvent accessible surface area calculations," F1000Research, vol. 5, 2016.

[55] S. Hubbard and J. Thornton, "NACCESS v. 2.1. 1-Atomic Solvent Accessible Area Calculations," Department of biochemistry and molecular biology, 1996.

[56] H. Lee, J. I. Juncosa and R. B. Silverman, "Ornithine aminotransferase versus GABA aminotransferase: implications for the design of new anticancer drugs," Medicinal research reviews, vol. 35, no. 2, pp. 286-305, 2015.

[57] T. L. Perry, S. Hansen and D. Lesk, "Plasma amino acid levels in children of patients with Huntington's chorea," Neurology, vol. 22, no. 1, p. 68, 1972.

[58] K. Gale, "GABA in epilepsy: the pharmacologic basis," Epilepsia, vol. 30, pp. S1—-S11, 1989.

[59] J.-Y. Wu, E. D. Bird, M. S. Chen and W. M. Huang, "Abnormalities of neurotransmitter enzymes in Huntington's chorea," Neurochemical research, vol. 4, no. 5, pp. 575-586, 1979.

[60] F. M. Sherif and S. S. Ahmed, "Basic aspects of GABA-transaminase in neuropsychiatric disorders," Clinical biochemistry, vol. 28, no. 2, pp. 145-154, 1995.

[61] B. LIPPERT, B. W. METCALF, M. J. JUNG and P. CASARA, "4-amino-hex-5-enoic acid, a selective catalytic inhibitor of 4-aminobutyric-acid aminotransferase in mammalian brain," European Journal of Biochemistry, vol. 74, no. 3, pp. 441-445, 1977.

[62] Y. Pan, M. R. Gerasimov, T. Kvist, P. Wellendorph, K. K. Madsen, E. Pera, H. Lee, A. Schousboe, M. Chebib, H. Bräuner-Osborne and Others, "CPP-115, a potent gamma-aminobutyric acid aminotransferase inactivator for the treatment of cocaine addiction," Journal of Medicinal Chemistry, vol. 55, no. 1, p. 357, 2012.

[63] M. J. Rodrigues, V. Windeisen, Y. Zhang, G. Guédez, S. Weber, M. Strohmeier, J. W. Hanes, A. Royant, G. Evans, I. Sinning and Others, "Lysine relay mechanism coordinates intermediate transfer in vitamin B6 biosynthesis," Nature chemical biology, vol. 13, no. 3, p. 290, 2017.

[64] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," Bioinformatics, vol. 22, no. 13, pp. 1658-1659, 2006.

[65] Genome.gov, "Why Mouse Matters," 2000. [Online]. Available: https://www.genome.gov/10001345/importance-of-mouse-genome/. [Accessed 1 7 2018].

[66] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," Journal of molecular biology, vol. 48, no. 3, pp. 443-453, 1970.

[67] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," Journal of Molecular Biology, vol. 147, no. 1, pp. 195-197, 1981.

[68] T. F. Smith and M. S. Waterman, "Comparison of biosequences," Advances in applied mathematics, vol. 2, no. 4, pp. 482-489, 1981.

[69] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg and Others, "Scikit-learn: Machine learning in Python," Journal of machine learning research, vol. 12, no. Oct, pp. 2825-2830, 2011.

[70] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of artificial intelligence research, vol. 16, pp. 321-357, 2002.

[71] Y. Xu, Y.-X. Ding, J. Ding, Y.-H. Lei, L.-Y. Wu and N.-Y. Deng, "iSuc-PseAAC: predicting lysine succinylation in proteins by incorporating peptide position-specific propensity," Scientific reports, vol. 5, p. 10184, 2015.

[72] M. B. Johansen, L. Kiemer and S. Brunak, "Analysis and prediction of mammalian protein glycation," Glycobiology, vol. 16, no. 9, pp. 844-853, 2006.

[73] S. Li, H. Li, M. Li, Y. Shyr, L. Xie and Y. Li, "Improved prediction of lysine acetylation by support vector machines," Protein and peptide letters, vol. 16, no. 8, pp. 977-983, 2009.

[74] I. Walsh, T. Di Domenico and S. C. E. Tosatto, "RUBI: rapid proteomic-scale prediction of lysine ubiquitination and factors influencing predictor performance," Amino acids, vol. 46, no. 4, pp. 853-862, 2014.

[75] W.-R. Qiu, X. Xiao, W.-Z. Lin and K.-C. Chou, "iUbiq-Lys: prediction of lysine ubiquitination sites in proteins by extracting sequence evolution information via a gray system model," Journal of Biomolecular Structure and Dynamics, vol. 33, no. 8, pp. 1731-1742, 2015.

[76] L. Zhu, J. Yang, J.-N. Song, K.-C. Chou and H.-B. Shen, "Improving the accuracy of predicting disulfide connectivity by feature selection," Journal of computational chemistry, vol. 31, no. 7, pp. 1478-1485, 2010.

[77] H.-H. Lin and L.-Y. Tseng, "DBCP: a web server for disulfide bonding connectivity pattern prediction without the prior knowledge of the bonding state of cysteines," Nucleic acids research, vol. 38, no. suppl_2, pp. W503--W507, 2010.

[78] C. Savojardo, P. Fariselli, M. Alhamdoosh, P. L. Martelli, A. Pierleoni and R. Casadio, "Improving the prediction of disulfide bonds in Eukaryotes with machine learning methods and protein subcellular localization," Bioinformatics, vol. 27, no. 16, pp. 2224-2230, 2011.

[79] C. Savojardo, P. Fariselli, P. L. Martelli and R. Casadio, "Prediction of disulfide connectivity in proteins with machine-learning methods and correlated mutations," BMC bioinformatics, vol. 14, no. 1, p. S10, 2013.