


8-2016

Protein Residue-Residue Contact Prediction Using Stacked Denoising Autoencoders

Joseph Bailey Luttrell IV

Follow this and additional works at: http://aquila.usm.edu/honors_theses

 Part of the [Bioinformatics Commons](#), and the [Computer Sciences Commons](#)

Recommended Citation

Luttrell, Joseph Bailey IV, "Protein Residue-Residue Contact Prediction Using Stacked Denoising Autoencoders" (2016). *Honors Theses*. Paper 428.

This Honors College Thesis is brought to you for free and open access by the Honors College at The Aquila Digital Community. It has been accepted for inclusion in Honors Theses by an authorized administrator of The Aquila Digital Community. For more information, please contact Joshua.Cromwell@usm.edu.

The University of Southern Mississippi

Protein Residue-Residue Contact Prediction Using Stacked Denoising Autoencoders

by

Joseph Bailey Luttrell IV

A Thesis
Submitted to the Honors College of
The University of Southern Mississippi
in Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science
in the School of Computing

August 2016

Approved by

Zheng Wang, Ph.D., Thesis Adviser
Assistant Professor of Computing

Andrew H. Sung, Ph.D., Director
School of Computing

Ellen Weinauer, Ph.D., Dean
Honors College

Abstract

Protein residue-residue contact prediction is one of many areas of bioinformatics research that aims to assist researchers in the discovery of structural features of proteins. Predicting the existence of such structural features can provide a starting point for studying the tertiary structures of proteins. This has the potential to be useful in applications such as drug design where tertiary structure predictions may play an important role in approximating the interactions between drugs and their targets without expending the monetary resources necessary for preliminary experimentation. Here, four different methods involving deep learning, support vector machines (SVMs), and direct coupling analysis were trained on a dataset of proteins from the 9th Critical Assessment of Techniques for Protein Structure Prediction (CASP 9). The models that were the most successful after training on the CASP 9 data were selected to perform the contact predictions in each method. After performing a blind test on CASP 11 targets, we have determined that further optimizations to the training process may be necessary to improve performance.

Key Words: protein residue-residue contact prediction, deep learning, ensemble learning, direct coupling analysis, stacked denoising autoencoders

Dedication

Dr. Zheng Wang:

Thank you for your tireless dedication to the planning and execution of this project over these past two years.

Acknowledgements

I would like to thank Danita Luttrell, Joseph Luttrell III, Charles Luttrell, and everyone else in my family for giving me support and encouragement during my undergraduate career. Next, I would like to recognize Dr. Zheng Wang and the entire School of Computing faculty for the wonderful instruction and guidance that I received in my classes. Finally, I am grateful to all of the support and advisement I received from the Honors College faculty during the process of completing this thesis.

Table of Contents

List of Tables and Figures.....	vii
Chapter 1: Introduction and Review of the Literature	1
Chapter 2: Methodology	6
Chapter 3: Evaluation Criteria and Results.....	12
Chapter 4: Discussion and Future Work.....	15
Literature Cited	17

List of Tables and Figures

Figure 1.1 - An Example of the Two Window Configuration.....	7
Table 3.1 - Blind Test Results at L/5	13
Table 3.2 - Blind Test Results at L	13
Table 3.3 - Blind Test Results at 5L	13
Table 3.4 - Blind Test Results at 10L	14
Table 3.5 - Blind Test Results at 50L	14
Table 3.6 - Blind Test Results at 150L	14

Chapter 1: Introduction and Review of the Literature

As the advancement of high-throughput sequencing continues to inundate biologists with large amounts of uncharacterized amino acid sequences, residue-residue contact prediction is becoming a more viable way to gain valuable information about proteins before they even reach biological laboratories. For example, the ability to make predictions about which residues within a protein are in contact can give researchers an idea of how the native structure of that protein is formed before they expend considerable amounts of time, money, and other resources to experimentally determine it ¹. This has a diverse set of applications and has long been sought after as an aid in drug development and the design of drugs and other novel molecules ². Unfortunately, the incredibly complex nature of proteins creates many challenges that protein structure prediction methods must face if they hope to deliver their predictions in any reasonable amount of time. The ensuing race to accelerate these methods has spawned entirely new areas of computational research that aim to make supplementary predictions for use in the structure prediction process.

An important part of accomplishing these prediction speedups involves identifying any information that is available to help guide the structure prediction process without slowing it down. As it turns out, the sequence itself can be an invaluable source for finding such information. This is because Anfinsen's dogma asserts that, in many cases, the amino acid sequence for a protein encodes enough information to determine the native structure of that protein ^{3,4}. However, from a computational standpoint, predicting the structure of a protein from its amino acid sequence can be shown to be an NP-hard problem ⁵. Therefore, it may be beneficial to place restraints on the prediction of protein

structures by providing more information beforehand. One possible set of restraints that would satisfy this condition can be provided by incorporating information from residue-residue contact prediction methods.

These restraints make use of findings that suggest that the overall stability of native protein structures is influenced by intra-molecular interactions among amino acid residues ⁶. In this way, making predictions about which residues in a given protein may be in contact can provide information that is useful for eliminating structure predictions that would most likely not occur in nature. If a large enough subset of possible protein conformations is eliminated in this way before a protein structure prediction method is used, the underlying algorithms that generate the candidate structures will now be operating on a much smaller search space of conformation possibilities. Of course, residue-residue contact prediction carries its own unique set of challenges. As a result, many different methods for making these predictions have arisen over the years. In general, these methods share enough similarity to be categorized into a few broad groups. However, these groups are unique enough to merit further explanation.

In most residue-residue contact prediction scenarios, the desired end result is simply a prediction that labels a pair of residues as either in contact or not in contact. This means that the challenges of making these predictions can be readily framed as a classification problem ⁷. As a result, machine learning methods are among the most common solutions that are being explored and make use of a wide variety of techniques such as support vector machines (SVMs), hidden Markov models, and neural networks ⁸⁻¹⁰. Cheng et al. utilized SVMs and focused on improving their training data by focusing on feature selection and its effects on performance ⁹. Zhang et al. also utilized SVMs but

did so in a hybrid setup with another method that assumes certain pairs of residues have a higher chance of interacting in an inter-helical contact ⁸. In this way, it is still possible to obtain a contact prediction if a situation that causes the knowledge-based SVM method to fail at producing a result occurs. Eickholt et al. took a different approach and developed a method based on deep networks with boosting applied ¹⁰. The classification of contacts in their method was performed by a combination of restricted Boltzmann machines (RBMs) trained to form deep networks (DNs). The boosting process involved using the classifiers to modify the weights of the training set based on their own performance.

In general, these methods approach residue-residue contact prediction as a statistical problem based on the principles of machine learning. While this can deliver solid prediction performance, other methods aim to take more direct advantage of the abundance of protein information that is already available from experimental results. For example, template based methods make their contact predictions by utilizing threading or homology to identify similarities between a query protein and previously known structures (templates) before using the residue interaction data present in those templates to make informed predictions that may be more biologically relevant ¹¹⁻¹⁴. An early example of a hybrid method that combined a template-based approach with hidden Markov models can be seen in HMMSTR-CM as described by Shao and Bystroff ¹⁴. HMMSTR-CM's initial output is a contact map calculated from the target sequence by the hidden Markov model. Then, the target contact map is aligned against a set of pre-calculated contact maps derived from a collection of template structures before being modified according to a set of predefined rules composed of assumptions made with knowledge of folding pathways. Wu et al. combined SVMs and contact predictions from

multiple, locally installed threading methods into a method called SVM-LOMETS¹¹. The inclusion of SVMs was done to train the parameters for Euclidean distance cutoff between residue pairs and the alignment qualities on the contact map.

Another category of contact prediction methods employs the concept of correlated mutation analysis (CMA) to search for mutually related discrepancies in the amino acid sequence across multiple sequence alignments (MSAs) and then predict contacts based on this¹⁵⁻¹⁸. In other words, these methods look for situations in which a residue in one column of an alignment frequently changes in a way that corresponds to changes seen in one other column of the alignment. These residue positions are seen as having a higher chance of sharing some sort of biological pairing and are predicted to be potential contacts. CCMpred was developed by Seemayer et al. and utilizes a statistical technique known as pseudo-likelihood maximization in order to better distinguish direct couplings between pairs of columns in the multiple sequence alignment from pairs that are simply correlated¹⁶. Their goal to deliver fast and precise contact predictions was accomplished by using the parallelization abilities offered by modern GPUs to compute the gradient of the pseudo-likelihood. Jacob et al. were able to show that expanding upon the original premise of correlated mutation analysis to utilize both amino acid MSAs and codon MSAs results in a meaningful increase of contact prediction performance¹⁷. They assert that direct contacts are more likely if the correlation of the MSA is high at the amino acid level but low at the codon level. To make use of this in filtering predictions, a scoring function was proposed that can be used with existing CMA methods. An earlier approach to CMA that also incorporated filtering predictions was described by Kundrotas and Alexov with their webserver named RECON (REsidue CONtacts)¹⁸. They managed to

filter the results from their correlated mutation-based predictions with a set of rules in the form of biophysical constraints (hydrophobic pairing, ionic pairing, disulfide bridges, etc).

Here, we present our findings from benchmarking four distinct residue-residue contact prediction methods. The techniques used in these methods were based on deep learning ensembles, SVMs, direct coupling analysis (DCA) based on multiple sequence alignments, and an ensemble that incorporated both SVM and DCA. Our two baseline methods, referred to as the "SVM baseline" and DCA_cpp, used only SVM and DCA, respectively. Next, our deep learning approach, referred to as SDAplusDCA_Deep, combined stacked denoising autoencoders (SdAs), DCA, and SVM into a deep learning ensemble. Based on our survey of the literature, SdAplusDCA_Deep is novel in the field of residue-residue contact prediction. Our second ensemble method, referred to simply as SVMplusDCA, combined DCA_cpp with the SVM baseline. These four methods were then benchmarked in a blind test using data from CASP 11 (The 11th Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction) and the results were presented according to a set of evaluation metrics ¹⁹.

Chapter 2: Methodology

The first step to performing predictions with our four methods was to generate the features that were used to train our models. To do this, a set of proteins was picked from the CASP 9 targets. Any protein that was over 1,000 residues in length was excluded from the set. In total, the data from 87 of these targets made it into the final training data. Each of the 64,172 lines in the training data is a single example that represents the interaction between a pair of residues that are separated by at least six residues in the sequence. This interaction information is encoded into each example with 1,612 features. To generate these features, the amino acid sequence from each of the proteins in this training set was used as the input for a software pipeline that we developed. The first step was using PSIPRED version 3.5 and ACCpro version 5.1 (a member of the SCRATCH 1.0 package) to predict the secondary structure and solvent accessibility of each target protein^{20,21}. PSIPRED's predictions are encoded in each example as three probability values (a decimal number between zero and one) representing the chance of each residue being within either a beta sheet, alpha helix, or a coil. These predictions are made using two feed-forward neural networks that incorporate data obtained from PSI-BLAST. ACCpro's predictions use bidirectional recurrent neural networks and are encoded in each example with a binary prediction value that represents whether or not the residue in question is solvent accessible²².

With these predictions complete, the next part of the feature generation process is based entirely on aspects of the target protein's sequence itself and is accomplished through the use of a pair of sliding windows. Each example in the training data represents a pair of residues centered within two sliding windows that each contain 11 residues.

These windows are placed with their centers at a minimum separation of six residues apart in the sequence at all times. (Fig. 1) illustrates one example of a pairing between the first residue of a protein centered in the window labeled "A" and the 62nd residue centered in the window labeled "B". The first five residues on the left of the sequence (in the segment labeled "D") are "empty" residues that are generated when one of the two windows extends past the boundaries of the amino acid sequence and simply have all of their features set to zero.

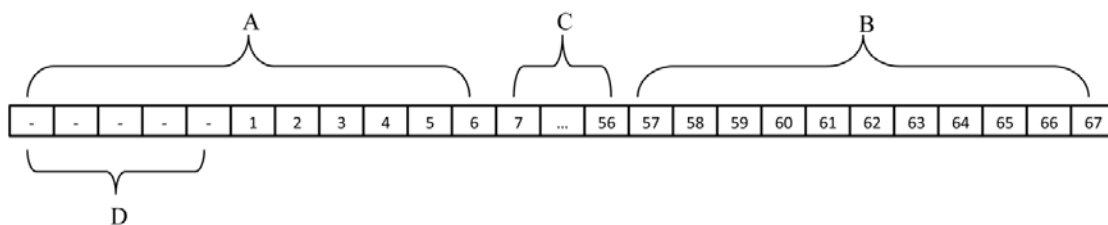


Figure 1.1 - An Example of the Two Window Configuration. Segment A represents the first window and is centered at residue 1. Segment D is part of the first window, but is composed of "empty" residues since the window extends beyond the residue range of the protein at this position. Segment B represents the second window and is centered at residue 62. Segment C represents the 50 residues between the first and second windows.

After the windows have been placed, the 1,612 features that make up each example are selected according to a standard procedure. The amino acid type of each residue is encoded by 20 features in an array made up of 20 bits. For each type of residue, only one unique bit is set to '1' (true) and the remaining 19 bits are set to '0' (false). The next additions are three features that encode PSIPRED's secondary structure prediction, one feature that encodes ACCpro's predictions, and two features that encode whether or not it is within the boundaries of the first or the second window. Therefore, each residue is represented by a set of 26 features. The largest set of features in each example is

composed of a selection of exactly 50 residues between the centers of the sliding windows as seen in the segment of (Fig. 1) labeled "C". Regardless of the current amount of residues that separate the two window centers, 50 residues are always selected. In other words, the features for each residue are repeated if there are not enough residues in this range or skipped if there are too many. This was implemented by first defining a scale value that is determined by dividing the total number of residues between the window centers by 50. Next, a set of 50 numbers described by the sequence $(a_n)_{n=1}^{n=50} = s * n$ where s is the scale value was generated. This sequence is iterated over and the residue at the sequence position of the integer value at each point in the sequence is added to the example. In addition to the two window centers, this means that this internal section of example is 52 residues in length and comprises 1,352 features (52 residues multiplied by 26 features). The remaining 260 features are generated from the external edge of the two windows.

The target value for each training example is a binary value that denotes whether or not the pair of residues at the center of the first and second windows is in contact. The information used to determine this value was obtained from the three dimensional coordinates provided for each target protein in the CASP 9 data set. Here, we define a pair of residues to be in contact (a positive example) if their alpha carbons ($C\alpha$) are less than or equal to 8 Å apart as measured by the formula for Euclidean distance in three dimensions. If the Euclidean distance does not fit this requirement, the residue pair is marked as a negative example and is not considered to be in contact. Once all of the examples had been generated for the CASP 9 training data set, we noticed that there was an abundance of negative target values (examples in which the residue pairs were not in

contact). In order to potentially help alleviate bias during training, we balanced the training set by keeping every positive example and selecting a random sample of negative examples equal to the number of positive examples. After performing this balancing operation, the final training file consisted of 102,480 examples. Each method except for DCA_cpp used this file as the basis for training.

DCA_cpp is a C++ implementation of the direct coupling analysis methods introduced by Morcos et al ²⁹. First, HHblits is used to generate an MSA (multiple sequence alignment) against the HMM (Hidden Markov Model) database "uniprot20_2015_06"³⁰. With this MSA, a sequence profile and contact map is obtained using the methods presented by Morcos et al ²⁹. The end result is a contact map with predictions scored by confidence value for every possible contact. Since no training was needed to use this method, the input was only the amino acid sequence for each target protein that was used in generating the training data for the other three methods. The resulting predictions were simply sorted with the highest confidence values listed first.

The SVM models used in all of the methods were created with an implementation of SVM called SVM_light ²⁸. First, the SVM baseline was trained by splitting the training set (without any additional predictions incorporated) into five equal pieces and performing a five-fold cross-validation. In each of these five folds, four of the pieces of data were used for training and only one piece was used for testing. One of the top performing models was chosen to execute the contact predictions.

SVMplusDCA incorporated these predictions into its training data by adding a set of 122 new features to each example. Half of these features represent the probability of

the left window center being in contact with any residue in the example to its right and the other half represents the probability of the right window center being in contact with any residue in the example to its left. This brings the total number of features in each example to 1,734. After the resulting predictions for each residue pair were added back into the training data, an SVM model was trained on this new dataset. The same process and parameters used to train the SVM baseline was used to train this new model. A five-fold cross validation was performed and one of the top performing models was chosen to produce the final contact predictions in the blind test. During the blind test, DCA_cpp predictions are performed on each target and added to the examples in the same way as in training.

SdAplusDCA_Deep utilized stacked denoising autoencoders based on Theano to form a consensus of multiple deep networks as we have implemented before^{23,24}. An autoencoder (also known as an autoassociator) is a type of mathematical model that learns a representation (an encoding) of its input in a way that enables it to reconstruct the input²⁵. When an autoencoder is able to reconstruct the input from a corrupted version of that same input, it is known as a denoising autoencoder and can be stacked in series to create a Stacked Denoising Autoencoder (SdA)^{26,27}. Here, ten models were trained using four-fold cross-validations and varying different parameters such as the corruption level and the number of hidden units. The dataset used in these cross-validations already contained DCA_cpp predictions in each example and was first split into 11 pieces. Ten of these pieces each make up eight percent of the dataset. Next, each of these pieces was split so that three percent was used for pre-training, three percent was used for training, one percent was used for validation, and the remaining one percent was used for testing.

One of the models with the highest accuracy on the test set was chosen from each four-fold cross validation to participate in the ensemble. All of the target proteins' potential contacts were simply classified by all ten of these models separately. Each of these models produced a positive and negative confidence value for each potential contact. This gives 20 features that are incorporated into each example of the remaining 20 percent of the training data that was not used to train the SdA models. Finally, an SVM model is trained using this data and produces the final contact predictions. During classification in the blind test, these twenty features were added along with DCA_cpp predictions using the same procedure.

Chapter 3: Evaluation Criteria and Results

After the training phase was completed, 86 targets were selected from the full set of CASP 11 targets. Once the features were generated for the 86 targets, all four methods were used to produce predictions for each target. These predictions represent potential contacts between pairs of residues and are sorted with the predictions having the highest confidence values listed first. For the purpose of this evaluation, predictions that do not expect a pair of residues to be in contact were discarded (negative predictions). Then, all of the predictions that met these requirements were scored using two performance metrics known as accuracy (the number of true predicted contacts divided by the number of false predicted contacts plus the number of true predicted contacts) and coverage (the number of true predicted contacts divided by the number of actual true contacts). Furthermore each accuracy and coverage score was categorized into one of three different "sequence separation" tiers.

These tiers are organized so that tier 6 contains all contacts separated by six or more (but less than 12) residues, tier 12 contains all contacts separated by 12 or more (but less than 24) residues, and tier 24 contains all contacts separated by 24 or more residues. This organization scheme applied not only to the predicted contacts, but also to the true contacts (the contacts that were actually in contact in the structure). The resulting scores for each method are listed in Table 3.1, 3.2, 3.3, 3.4, 3.5 and 3.6. Each table is differentiated by the number of top contacts that was used in the evaluation of accuracy and coverage. L represents the number of residues in the sequence of each target. For example, L/5 indicates that a number of contacts equal to only 1/5th of the length of each target's sequence were selected to be evaluated. Each row contains the name of the

method abbreviated as svm (the SVM baseline), svm+ (SVMplusDCA), sda (SdAplusDCA_Deep), and dca (DCA_cpp).

	acc6	acc12	acc24	cov6	cov12	cov24	avgAcc	avgCov
svm	5.238%	3.611%	3.567%	2.195%	0.568%	0.113%	4.139%	0.959%
svm+	4.991%	1.292%	0.499%	0.130%	0.032%	0.031%	2.261%	0.065%
sda	8.137%	6.170%	1.642%	0.291%	1.062%	0.180%	5.317%	0.511%
dca	7.048%	18.288%	0.898%	0.235%	0.434%	0.115%	8.745%	0.261%

Table 3.1: Blind test results at L/5. The averaged results of the blind test of the four methods evaluated with the top L/5 predicted contacts from 86 CASP 11 targets (where L is the length of the target). Accuracy (Acc), average accuracy (avgAcc), coverage (cov) and average coverage (avgCov) are listed at 3 different sequence separation levels (6, 12, 24) as described in the main text. The columns for avgAcc and avgCov are averages of the three sequence separation columns for each metric.

	acc6	acc12	acc24	cov6	cov12	cov24	avgAcc	avgCov
svm	7.892%	7.346%	3.263%	12.829%	6.453%	0.708%	6.167%	6.663%
svm+	4.991%	1.292%	0.499%	0.130%	0.032%	0.031%	2.261%	0.065%
sda	12.140%	6.073%	2.169%	2.047%	4.299%	1.133%	6.794%	2.493%
dca	24.659%	29.390%	1.751%	2.290%	2.056%	1.293%	18.600%	1.880%

Table 3.2: Blind test results at L. The results of the blind test of the four methods evaluated with the top L predicted contacts from 86 CASP 11 targets (where L is the length of the target) listed in the same format as table 3.1.

	acc6	acc12	acc24	cov6	cov12	cov24	avgAcc	avgCov
svm	8.755%	8.016%	5.926%	56.314%	37.443%	6.516%	7.566%	33.424%
svm+	4.991%	1.292%	0.499%	0.130%	0.032%	0.031%	2.261%	0.065%
sda	8.190%	5.817%	2.260%	8.830%	15.399%	5.054%	5.422%	9.761%
dca	9.313%	15.475%	3.254%	20.868%	16.059%	10.400%	9.347%	15.776%

Table 3.3: Blind test results at 5L. The results of the blind test of the four methods evaluated with the top 5L predicted contacts from 86 CASP 11 targets (where L is the length of the target) listed in the same format as table 3.1.

	acc6	acc12	acc24	cov6	cov12	cov24	avgAcc	avgCov
svm	7.315%	6.555%	5.083%	81.862%	60.880%	13.292%	6.318%	52.011%
svm+	4.991%	1.292%	0.499%	0.130%	0.032%	0.031%	2.261%	0.065%
sda	8.166%	5.469%	2.188%	13.979%	20.865%	7.728%	5.274%	14.191%
dca	7.252%	8.113%	3.214%	37.237%	28.227%	18.788%	6.193%	28.084%

Table 3.4: Blind test results at 10L. The results of the blind test of the four methods evaluated with the top 10L predicted contacts from 86 CASP 11 targets (where L is the length of the target) listed in the same format as table 3.1.

	acc6	acc12	acc24	cov6	cov12	cov24	avgAcc	avgCov
svm	5.407%	4.291%	3.274%	97.420%	93.095%	41.018%	4.324%	77.178%
svm+	4.991%	1.292%	0.499%	0.130%	0.032%	0.031%	2.261%	0.065%
sda	8.090%	5.333%	2.178%	16.813%	24.066%	9.620%	5.200%	16.833%
dca	5.422%	4.157%	1.938%	82.014%	75.419%	60.576%	3.839%	72.670%

Table 3.5: Blind test results at 50L. The results of the blind test of the four methods evaluated with the top 50L predicted contacts from 86 CASP 11 targets (where L is the length of the target) listed in the same format as table 3.1.

	acc6	acc12	acc24	cov6	cov12	cov24	avgAcc	avgCov
svm	5.407%	4.291%	3.274%	97.420%	93.095%	41.018%	4.324%	77.178%
svm+	4.991%	1.292%	0.499%	0.130%	0.032%	0.031%	2.261%	0.065%
sda	8.090%	5.333%	2.178%	16.813%	24.066%	9.620%	5.200%	16.833%
dca	5.145%	3.736%	1.610%	94.793%	92.655%	85.717%	3.497%	91.055%

Table 3.6: Blind test results at 150L. The results of the blind test of the four methods evaluated with the top 150L predicted contacts from 86 CASP 11 targets (where L is the length of the target) listed in the same format as table 3.1.

Chapter 4: Discussion and Future Work

Starting with table 3.1, only the top $L/5$ contacts were selected for evaluation. This proved to be too narrow of a margin for most of the targets that were scored. At this point, DCA_cpp was the best performing method in terms of accuracy. By increasing the number of contacts evaluated to be equal to the length of each target (L), there was a noticeable increase in both accuracy and coverage for all of the methods except for SVMplusDCA. The relatively unchanging results for SVMplusDCA were most likely a result of the low number of predictions that it provided. DCA_cpp showed its highest accuracy at L top contacts and began to decline as the value of L was multiplied. Interestingly, the scores for the SVM baseline and SdAplusDCA_Deep continued to fluctuate until roughly $50L$ top contacts were evaluated. This would seem to indicate that the sorting of the predictions or the assignment of confidence values in these two methods was not adequate. However, two important errors in the execution of the blind test were detected at the end of this study that most likely affected the appearance of these results.

First, the sorting procedure of the contact predictions of all four methods mistakenly imposed a limit of 1.0 on the confidence value of each prediction. If a prediction was ranked with a higher confidence, it was rounded back down to 1.0. This meant that many predictions that would have been sorted to the top remained further down in the list. Multiplying the value of L in an attempt to pick up more of these contacts didn't have much of an effect on accuracy because of the increasing number of false predictions that were being picked up at the same time. The next error prevented the features for DCA_cpp predictions and SdA predictions from being added into the

examples of the blind test targets. Essentially, this means that SVMplusDCA and SdAplusDCA_Deep were being used to classify examples of contacts that they were not trained to predict. It is very likely that the evaluation results will improve when these features are properly incorporated. Future work for this project will be to correct these errors and conduct the blind test again. Also, a more focused literature review can be conducted to search for other ways to improve the training procedure.

Literature Cited

1. Wang Z, Eickholt J, Cheng J. MULTICOM: a multi-level combination approach to protein structure prediction and its assessments in CASP8. *Bioinformatics*. 2010;26(7):882–888.
2. Blundell TL, Sibanda BL, Sternberg MJE, Thornton JM. Knowledge-based prediction of protein structures and the design of novel molecules. *Nature*. 1987;326(6111):347–352.
3. Anfinsen CB. Principles that govern the folding of protein chains. *Science (New York, N.Y.)*. 1973;181(4096):223–230.
4. Anfinsen CB. Some observations on the basic principles of design in protein molecules. *Comparative Biochemistry and Physiology*. 1962;4:229–240.
5. Unger R, Moult J. Finding the lowest free energy conformation of a protein is an NP-hard problem: proof and implications. *Bulletin of mathematical biology*. 1993;55(6):1183–1198.
6. Gromiha MM, Selvaraj S. Inter-residue interactions in protein folding and stability. *Progress in Biophysics and Molecular Biology*. 2004;86(2):235–277.
7. Tegge AN, Wang Z, Eickholt J, Cheng J. NNcon: improved protein contact map prediction using 2D-recursive neural networks. *Nucleic acids research*. 2009;37(suppl 2):W515–W518.
8. Zhang H, Huang Q, Bei Z, Wei Y, Floudas CA. COMSAT: Residue contact prediction of transmembrane proteins based on support vector machines and mixed integer linear programming. *Proteins: Structure, Function, and Bioinformatics*. 2016;84(3):332–348.
9. Cheng J, Baldi P. Improved residue contact prediction using support vector machines and a large feature set. *BMC Bioinformatics*. 2007;8:113.
10. Eickholt J, Cheng J. Predicting protein residue–residue contacts using deep networks and boosting. *Bioinformatics*. 2012;28(23):3066–3072.
11. Wu S, Zhang Y. A comprehensive assessment of sequence-based and template-based methods for protein contact prediction. *Bioinformatics*. 2008;24(7):924–931.
12. Skolnick J, Kihara D, Zhang Y. Development and large scale benchmark testing of the PROSPECTOR_3 threading algorithm. *Proteins: Structure, Function, and Bioinformatics*. 2004;56(3):502–518.

13. Misura KMS, Chivian D, Rohl CA, Kim DE, Baker D. Physically realistic homology models built with rosetta can be more accurate than their templates. *Proceedings of the National Academy of Sciences*. 2006;103(14):5361–5366.
14. Shao Y, Bystroff C. Predicting interresidue contacts using templates and pathways. *Proteins: Structure, Function, and Bioinformatics*. 2003;53(S6):497–502.
15. Olmea O, Valencia A. Improving contact predictions by the combination of correlated mutations and other sources of sequence information. *Folding and Design*. 1997;2, Supplement 1:S25–S32.
16. Seemayer S, Gruber M, Söding J. CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics*. 2014;30(21):3128–3130.
17. Jacob E, Unger R, Horovitz A. Codon-level information improves predictions of inter-residue contacts in proteins by correlated mutation analysis. *eLife*. [accessed 2016 May 25];4. <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4602084/>
18. Kundrotas PJ, Alexov EG. Predicting residue contacts using pragmatic correlated mutations method: reducing the false positives. *BMC Bioinformatics*. 2006;7:503.
19. Moult J, Fidelis K, Kryshtafovych A, Tramontano A. Critical assessment of methods of protein structure prediction (CASP)—round IX. *Proteins: Structure, Function, and Bioinformatics*. 2011;79(S10):1–5.
20. McGuffin LJ, Bryson K, Jones DT. The PSIPRED protein structure prediction server. *Bioinformatics*. 2000;16(4):404–405.
21. Cheng J, Randall AZ, Sweredoski MJ, Baldi P. SCRATCH: a protein structure and structural feature prediction server. *Nucleic Acids Research*. 2005;33(suppl 2):W72–W76.
22. Pollastri G, Baldi P, Fariselli P, Casadio R. Prediction of coordination number and relative solvent accessibility in proteins. *Proteins: Structure, Function, and Bioinformatics*. 2002;47(2):142–153.
23. Liu T, Wang Y, Eickholt J, Wang Z. Benchmarking Deep Networks for Predicting Residue-Specific Quality of Individual Protein Models in CASP11. *Scientific Reports*. 2016;6:19301.
24. Wang Y, Liu T, Xu D, Shi H, Zhang C, Mo Y-Y, Wang Z. Predicting DNA Methylation State of CpG Dinucleotide Using Genome Topological Features and Deep Networks. *Scientific Reports*. 2016;6:19598.
25. Bengio Y. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*. 2009;2(1):1–127.

26. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *J. Mach. Learn. Res.* 2010;11:3371–3408.
27. Vincent P, Larochelle H, Bengio Y, Manzagol P-A. Extracting and Composing Robust Features with Denoising Autoencoders. In: *Proceedings of the 25th International Conference on Machine Learning*. New York, NY, USA: ACM; 2008. p. 1096–1103. (ICML '08). <http://doi.acm.org/10.1145/1390156.1390294>
28. Joachims T. Making large scale SVM learning practical. Universität Dortmund; 1999. <https://eldorado.tu-dortmund.de/handle/2003/2596>
29. Morcos F, Pagnani A, Lunt B, Bertolino A, Marks DS, Sander C, Zecchina R, Onuchic JN, Hwa T, Weigt M. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*. 2011;108(49):E1293–E1301.
30. Remmert M, Biegert A, Hauser A, Söding J. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature Methods*. 2012;9(2):173–175.