ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Mustafa Özber YÜCEKUL

USING MACHINE LEARNING METHODS TO PREDICT PROTEIN CONTACT MAP

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADANA, 2015

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

USING MACHINE LEARNING METHODS TO PREDICT PROTEIN CONTACT MAP

Mustafa Özber YÜCEKUL

MSc THESIS

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

We certified that the thesis titled above was reviewed and approved for the award of degree of the Master of Science by the board of jury on 19/01/2015.

…….....	…….	…….....
Assoc. Prof. Dr. Turgay İBRİKÇİ	Prof. Dr. Hamza EROL	Assoc. Prof. Dr. Sami ARICA
SUPERVISOR	MEMBER	MEMBER

This MSc Thesis is written at the Department of Institute of Natural And Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

Not: The usage of the presented specific declerations, tables, figures, and photographs either in this thesis or in any other reference without citiation is subject to "The law of Arts and Intellectual Products" number of 5846 of Turkish Republic

ABSTRACT

MSc THESIS

USING MACHINE LEARNING METHODS TO PREDICT PROTEIN CONTACT MAP

Mustafa Özber YÜCEKUL

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Supervisor : Assoc. Prof. Dr. Turgay İBRİKÇİ
Year: 2015, Pages: 61
Jury : Assoc. Prof. Dr. Turgay İBRİKÇİ
: Prof. Dr. Hamza EROL
: Assoc. Prof. Dr. Sami ARICA

Protein is the crucial biological macromolecules which is consist chains of amino acid residues joined together by peptide bonds in human body. Since protein's structure is determined by its sequence, predicting protein structure from its sequence has become one of the most fundamental issue in the bioinformatics. The residue contact map of protein contains knowledge which represents three dimensional conformation of protein.

In this thesis, machine learning approaches to protein contact map prediction problem based on neural networks, logistic regression, support vector machines and other a few known machine learning methods were presented.

The results are compared. These results demonstrate that the Logistic Regression, Support Vector Machine, Decorate, and Dagging algorithms give the better results as protein contact map prediction on our dataset.

Key Words: Protein Contact Map, Artificial Neural Network, Logistic Regression, Support Vector Machines, Machine Learning

ÖZ

YÜKSEK LİSANS TEZİ

USING MACHINE LEARNING METHODS TO PREDICT PROTEIN CONTACT MAP

Mustafa Özber YÜCEKUL

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK ANABİLİM DALI

Danışman : Doç. Dr. Turgay İBRİKÇİ
Yıl: 2015, Sayfa: 61
Jüri : Doç. Dr. Turgay İBRİKÇİ
: Prof. Dr. Hamza EROL
: Doç. Dr. Sami ARICA

Protein, peptit bağları ile birbirne bağlanmış amino asit rezidülerinden oluşan insan vücudunda çok önemli biyolojik makro moleküldür. Protein yapısı, onun dizileriyle belirlendiği için protein dizilerinden protein yapısı tahmini biyoinformatikte en önemli temel problem olmuştur. Protein rezidülerinin temas haritası proteinin üç boyutlu biçiminin bilgisini içermektedir.

Bu tezde protein temas haritası tahmini; sinir ağları, lojistik regresyon, destek vektör makineleri ve diğer bilinen birkaç makine öğrenme metotları temel alınarak sunulmuştur.

Bu tezde, sonuçlar karşılaştırılmıştır. Bu sonuçlar, Lojistik Regresyon, Destek Vektör Makineleri, Decorate ve Dagging algoritmalarının protein rezidü temas haritalarını tahmin etmede bizim veri setimiz üzerinde daha iyi sonuçlar verdiğini göstermiştir.

Anahtar Kelimeler: Protein Temas Haritası, Yapay Sinir Ağları, Lojistik Regresyon, Destek Vektör Makineleri, Makine Öğrenmesi

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Assoc. Prof. Dr. Turgay İBRİKÇİ for his guidances and help throughout the thesis work. Many thanks for all of their supports, patiences, co-operations, suggestions on initiating, improving and completing this study. I am grateful to my committe members Prof. Dr. Hamza EROL, Assoc. Prof. Dr. Sami ARICA for their endless support and encouragement.

I would present my special thanks to my wife Aylin and my son Mert Şeref. She gave me the strongest support through this thesis. I would also like to acknowledge valuable help from my friends and Çukurova University Department of Electrical and Electronics Engineering staff.

CONTENTS **PAGE**

ABSTRACT I
ÖZ II
ACKNOWLEDGEMENTS III
CONTENTS IV
LIST OF TABLES VI
LIST OF FIGURES VIII
NOTATIONS X
1. INTRODUCTION 1
1.1. Biological Background 2
1.1.1. Proteins 3
1.2. Problem Statement 4
2. ARTICLE REVIEWS 9
3. MATERIAL AND METHODS 15
3.1. Material 15
3.1.1. Dataset 15
3.1.2. CMView 16
3.1.3. Features 17
3.2. Methods 19
3.2.1. Artificial Neural Networks 19
3.2.1.1. Backpropagation Algorithm 21
3.2.1.2. Delta Bar Delta Algorithm 22
3.2.2. Logistic Regression 23
3.2.3. Support Vector Machines 26
3.2.3.1. Soft Margin Classifier 28
3.2.3.2. Kernel Trick 30
3.2.3.4. Other Used Machine Learning Methods 31
3.3. Performance Assessments 34
4. EXPERIMENTS, RESULTS AND DISCUSSIONS 38
4.1. Classification by WEKA 39

4.2. ANN Results 39
4.3. Logistic Regression Results 43
4.4. Support Vector Machines Results 46
4.5. Comparison with Other Methods 49
5. CONCLUSION 53
6. FUTURE WORKS 55
REFERENCES 57
BIOGRAPHY 62

LIST OF TABLES **PAGE**

Table 3.1. The protein database 15
Table 3.2. The transformation of features 18
Table 3.3. List of amino acids and codes that represent them 18
Table 3.4. Confusion matrix 35
Table 4.1. Residue based contact prediction result (a) for ANN 40
Table 4.2. Residue based contact prediction result (b) for ANN 40
Table 4.3. Residue based contact prediction confusion matrix result for ANN 40
Table 4.4. Protein 1A32 contact map prediction result (a) for ANN 41
Table 4.5. Protein 1A32 contact map prediction result (b) for ANN 41
Table 4.6. Protein 1A32 confusion matrix result for ANN 41
Table 4.7. Protein 1A62 contact map prediction result (a) for ANN 42
Table 4.8. Protein 1A62 contact map prediction result (b) for ANN 42
Table 4.9. Protein 1A62 confusion matrix result for ANN 42
Table 4.10. Residue based contact prediction result (a) for Logistic Regression 43
Table 4.11. Residue based contact prediction result (b) for Logistic Regression 43
Table 4.12. Residue based contact prediction confusion matrix result for Logistic Regression 43
Table 4.13. Protein 1A32 contact map prediction result (a) for Logistic Regression 44
Table 4.14. Protein 1A32 contact map prediction result (b) for Logistic Regression 44
Table 4.15. Protein 1A32 confusion matrix result for Logistic Regression 44
Table 4.16. Protein 1A62 contact map prediction result (a) for Logistic Regression 45
Table 4.17. Protein 1A62 contact map prediction result (b) for Logistic Regression 45
Table 4.18. Protein 1A62 confusion matrix result for Logistic Regression 45
Table 4.19. Residue based contact prediction result (a) for SVM 46
Table 4.20. Residue based contact prediction result (b) for SVM 46

Table 4.21. Residue based contact prediction confusion matrix result for SVM..... 46
Table 4.22. Protein 1A32 contact map prediction result (a) for SVM..... 47
Table 4.23. Protein 1A32 contact map prediction result (b) for SVM..... 47
Table 4.24. Protein 1A32 confusion matrix result for SVM..... 47
Table 4.25. Protein 1A62 contact map prediction result (a) for SVM..... 48
Table 4.26. Protein 1A62 contact map prediction result (b) for SVM..... 48
Table 4.27. Protein 1A62 confusion matrix result for SVM..... 48
Table 4.28. Residue based contact prediction result for all methods..... 49
Table 4.29. Protein 1A32 contact map result for all methods..... 50
Table 4.30. Protein 1A62 contact map result for all methods..... 51

LIST OF FIGURES **PAGE**

Figure 1.1. The formation of protein 3
Figure 1.2. The general structure of an amino acid 3
Figure 1.3. Protein structures.....4
Figure 1.4. An image of residue-residue contacts 5
Figure 1.5. Protein contact map of 1XERA.....6
Figure 3.1. Example of dataset format.....17
Figure 3.2. Model of an artificial neuron.....20
Figure 3.3. A hyperplane classifies two classes of training data which is two dimensional 27
Figure 4.1. Inputs of features.....38
Figure 4.2. Accuracy and recall values of residue based prediction result.....49
Figure 4.3. Accuracy and recall values of 1A32 contact map prediction result.....50
Figure 4.4. Accuracy and recall values of 1A62 contact map prediction result.....51

NOTATIONS

CASP	: Critical Assessment of protein Structure Prediction
3D	: Three Dimension
DNA	: Deoxyribonucleic Acid
mRNA	: Messenger Ribonucleic Acid
A	: Adenine nucleobase
G	: Guanine nucleobase
C	: Cytosine nucleobase
U	: Urasil nucleobase
T	: Thymine nucleobase
AA	: Amino Acid
C_α	: Alpha carbon
C_β	: Beta carbon
Å	: Angstrom=$1x10^{-10}$ m
NMR	: Nuclear Magnetic Resonance
ANN	: Artificial Neural Networks
SVM	: Support Vector Machines
MLP	: Multi Layer Perceptron
PDB	: Protein Data Bank
w	: Synaptic weights
y	: Output matrix
x	: Input matrix
i	: Input node
j	: Output node
α	: Learning rate
LR	: Logistic Regression
μ	: Momentum coefficient
Θk	: Distributed random vectors
PART	: Partial Decision Trees
N	: Total number of training vectors

$\Phi(.)$	: Activation function
K(.)	: Kernel function
C	: Tradeoff parameter between error and margin
ζ_i	: Slack variables
DECORATE	: Diverse Ensemble Creation by Oppositional, Relabeling of Artificial Training Examples
η	: Learning rate
γ_x	: The equation of the margin
β	: Model parameters
θ	: Parameter vector
κ	: Kappa statistic coefficient
MAE	: Mean Absolute Error
RMSE	: Root Mean Squared Error
RAE	: Relative Absolute Error
RRSE	: Root Relative Squared Error
Acc	: Accuracy

1. INTRODUCTION

Bioinformatics is an interdisciplinary scientific area which is the application of computer technology to the management of biological information. Machine learning methods are widely used in bioinformatics. The development of machine learning methods for protein structure prediction, one of the most fundamental problems in bioinformatics.

In a cell which is basic unit of life, proteins interact with each other and with other molecules to carry out assorted types of biological functions ranging from enzymatic catalysis, to gene regulation and control of growth and differentiation, to transmission of nerve impulses (Laskowski et al., 2003). Extensive biochemical experiments have shown that a protein's function is determined by its structure (Travers, 1989; Bjorkman and Parham, 1990). Thus, elucidating a protein's structure is key to understanding its function, which in turn is essential for any related biological, biotechnological, medical, or pharmaceutical applications. In spite of progress in electronics and other areas, experimental determination of a protein structure can still be expensive, labor intensive, time consuming, and not always possible. (Cheng et al., 2008).

The software for protein structure prediction software is becoming a vital instrument in understanding phenomena in bioinformatics and has important applications in medical sciences for instance drug design (Jacobson and Sali, 2004). Every two years, the performance of current methods is assessed in Critical Assessment of protein Structure Prediction (CASP). CASP goal is to help advance the methods of identifying protein structure from sequence information. The participants of CASP only know the amino acid sequences of proteins and are desired to predict the native structure using their algorithms.

The aim of this thesis is to find an efficient machine learning method which can provide information about the protein contact map which is two dimensional representation of the three dimensional conformation of the protein. Thus, 3D structure of previously unknown protein structure used machine learning method contributes the living organizm task of this protein or find the function. This thesis

considers a method to predict the native fold of a protein using only the sequence by means of machine learning algorithms. If residue residue contact is extracted out of the sequence, complete protein is constructed (Glasgow, 2006).

1.1. Biological Background

In this section, the fundamental background information is given for requiring to understand the biological aspects of the thesis. These basic concepts of biology shed light on the protein contact map prediction problem to explain for better understanding of the problem.

All the knowledge of how human being are construct is described in its genes within the DNA. Extracting the correct information out of all the sequence data is a essential and challenge for a bioinformatician (Branden, 1999).

The role of protein structure prediction is now more important than ever due to enormous amounts of protein sequence data are produced by modern large-scale DNA sequencing efforts such as the Human Genome Project (Robins, 1992; OLSON, 1993). The chain of amino acids is assembled in the living cell which is called protein synthesis. Protein synthesis begins in the nucleus with transcription. One part of the deoxyribonucleic acid (DNA) strand is copied and encoded into a molecule called messenger ribonucleic acid (mRNA). The mRNA is then transported out of the nucleus membrane and translated into a chain of amino acids by the ribosomes in the cytoplasm. The translation of mRNA nucleic acids into amino acids is specified by rules called the genetic code. There are four different nucleotides, so mRNA (and DNA) can be represented by strings of an alphabet of 4 letters. Usually A,G,C and T for DNA and A, G, C and U for mRNA. The genetic code is a mapping from every triplet of nucleotides giving 64 combinations into one of the 20 amino acids. The construction of the actual amino acid chain, also called the polypeptide chain, occurs when neighbouring amino acids react and create peptide bonds. (Paluzwevski, 2008)

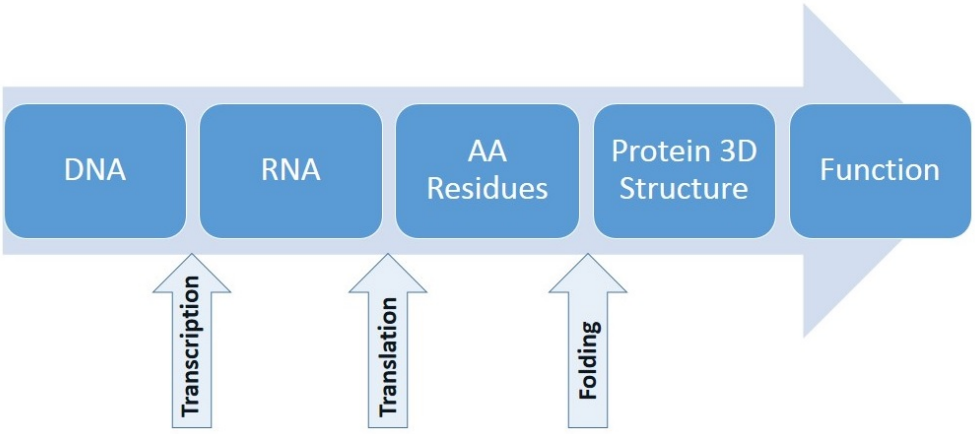


Figure 1.1. The formation of protein

1.1.1. Proteins

Protein is a macromolecule composed of 20 different amino acids which are linked by peptide bonds in the linear order (Sanger and Thompson, 1953). The linear polypeptide chain is called the primary structure of the protein. The primary structure can be represented as a sequence of 20 different letters, where each letter denotes an amino acid. Protein primary structure, simplest way to represent proteins, consists to sequence of amino acids which basic chemical units of protein.

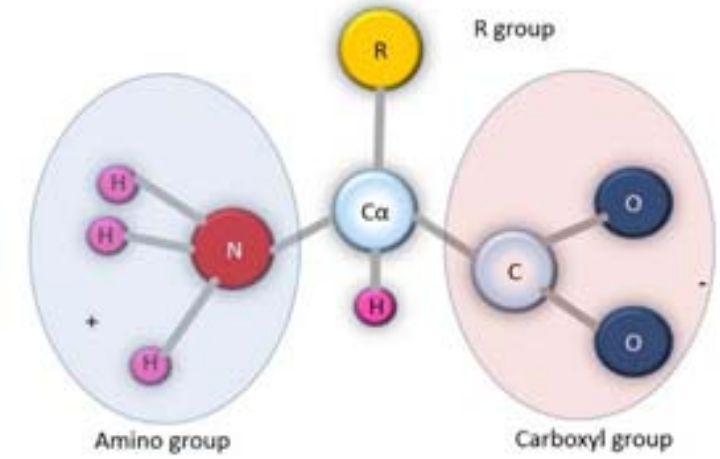


Figure 1.2. The general structure of an amino acid

In the native state, the amino acids (or residues) of a protein fold into local secondary structures including alpha helix, beta sheet, and non-regular coil (Pauling

et al., 1951). The secondary structure elements are further packed to form tertiary structure due to hydrophobic forces and side chain interactions between amino acids (Kendrew et al., 1962; Dill, 1990). The tertiary structures of several related proteins can bind together to form protein complex called quaternary structure. (Cheng, 2008) The variety of protein functionality is very large and proteins are typically grouped into three categories: structural proteins, enzymatic proteins and functional proteins.

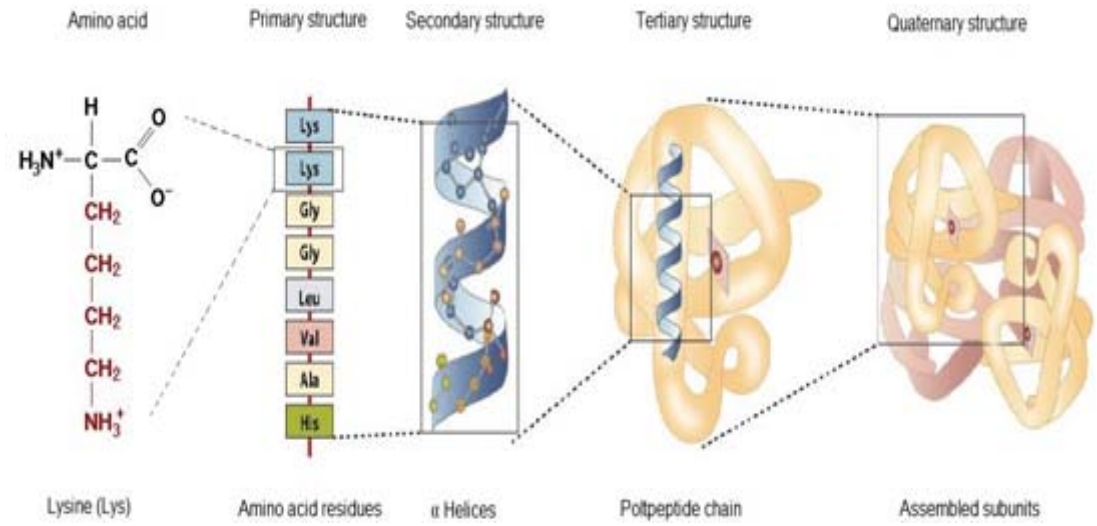


Figure 1.3. Protein structures (http://chemwiki.ucdavis.edu)

1.2. Problem Statement

A protein backbone's three-dimensional structure can be demonstrated by its residue contact map. The contact map for a protein of *n* residues is a *n x n* symmetrical, square, binary matrix of pairwise where C_{ij} represents the contact between residue *i* and residue *j*.

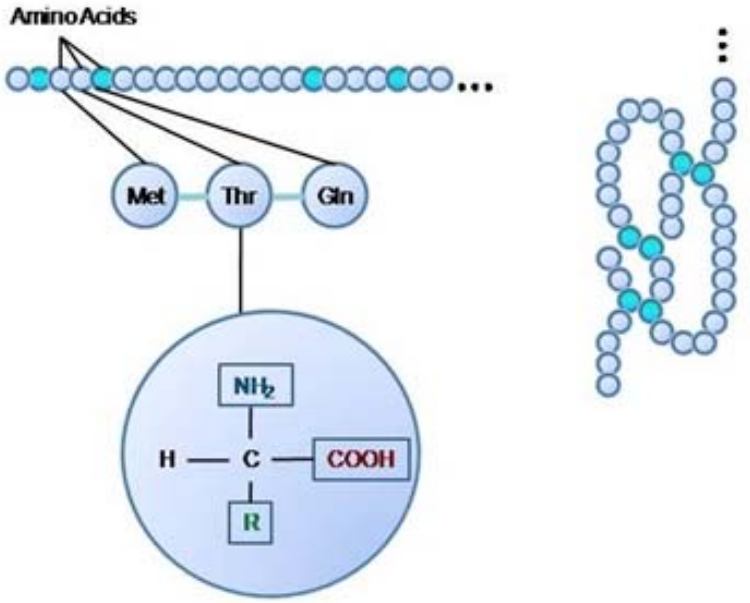


Figure 1.4. An image of residue-residue contacts (ocean blue dots), from sequence (left) to the 3-D structure (right).

In defining contacts for interresidue within protein is specified by two criteria: First criteria is to define reference points of residue: Alpha carbons (C_α) or beta carbons (C_β). The second criteria is a distance threshold between the determined reference points of interresidue. In this thesis, 8 angstrom is chosen in order to compare to previous results. Routinely, a contact is said to exist when a certain distance is below a threshold. The distance may be that between C_α atoms (Vendruscolo et al., 1997), C_β atoms, or it may be the minimum distance between any pair of atoms belonging to the side chain or to the backbone of the two residues (Fariselli et al., 1999a). A contact map which is symmetrical binary matrix of pairwise is a two dimensional representation of the three dimensional conformation of the protein. Therefore, the contact map is a minimalist representation of a protein native three dimensional structure (Krasnogor, 2004).

Depending on the separation of two residues along the sequence, the residue-residue contact is classified into three distinct classes:

- Long-range contacts: with sequence separation ≥ 24 residues
- Medium-range contact: with sequence seperation between 12 and 23 residues

- Short-range contacts: with sequence seperation between 6 and 11 residues (Yang et. al., 2011).

In this thesis, sequence separation is chosen 5 residue in order to balance to dataset. As previously mentioned, contact map representations are particularly interesting due to their invariance with respect to rotations and translations. Given an accurate contact map, several algorithms can be used to reconstruct the corresponding protein 3-D (Cheng et al., 2008; Wu et. al. 2008).

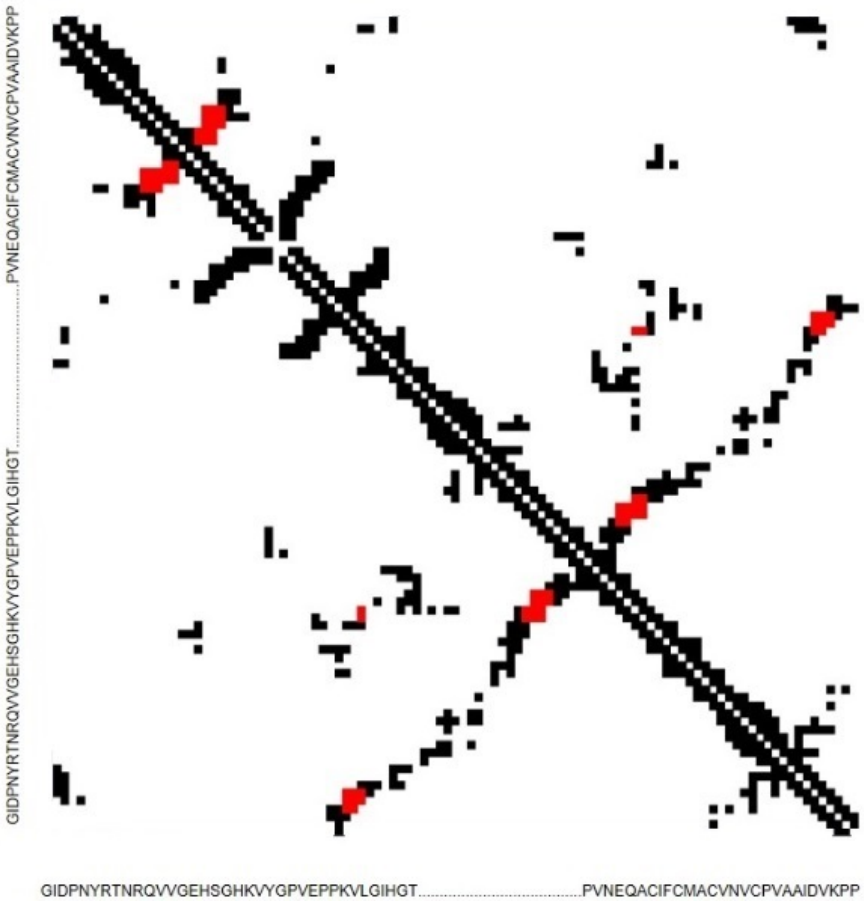


Figure 1.5. Protein contact map of 1XERA which generated from CMView. Red dots beta strands of secondary structure 1XERA

In general, there is a lot of variation in proteins and their structures. Precise structural contact prediction is at this moment a problem within the bigger problem of the prediction of the actual three-dimensional structure of a protein. Residue-

residue contacts is closely involved in guiding protein folding and maintaining the native fold of proteins (Gromiha et al., 2004). It has long been observed that with sufficient correct information about a protein's residue-residue contacts it is possible to elucidate the fold of the protein (Gobel et al., 1994; Olmea and Valencia, 1997).

2.ARTICLE REVIEWS

The rules that resolve the folding process of a protein are still unknown. Protein contact map prediction is useful for protein folding rate prediction, model selection and 3D structure prediction. Predicting unknown protein structures is important to the discovery of the functions of proteins, since the 3D protein structure determines the functional properties of the protein. Since, experimental analyses are quite expensive, complex and timeconsuming, machine learning research on bioinformatics have advanced rapidly in past decade. Therefore, many approaches have been proposed for protein residues contact map prediction for researcher in recent years.

Berg et al. (2001) gives a basic understanding of nuclear magnetic resonance (NMR) spectroscopy and X-ray crystallography. These are two of the most important techniques used to determine the 3D structure proteins. NMR spectroscopy is able to reveal the atomic structure of macromolecules in solution and depends on the fact that certainatomic nucleic are magnetic. Currently, NMR spectroscopy can only determine the 3D structures of relatively small proteins. X-ray crystallography currently provides the finest visualization of protein 3D structure (Drenth, 1994; Brunger, 1992). Each protein is crystallized in some type of salt to form highly ordered crystals. An X-ray is then directed at the crystallized protein, and the X-rays are detected by X-ray film. Then, the image of the protein is reconstructed from the X-ray film. Next, the electrodensity map is produced which gives the 3D structure of the protein. By mid 2000, the structures more than 10,000 proteins had been produced by either NMR spectroscopy or X-ray crystallography (Poulson, 2007).

Several machine learning algorithms, including neural networks (Fariselli et. al., 2001; Pollastri and Baldi, 2002; Pollastri and Baldi, 2003; Shackelford and Karplus, 2007), and support vector machines (Cheng and Baldi, 2007) have been applied to contact map prediction. Standard feedforward neural networks and support vector machines approaches use two windows around two target amino acids, *i* and *j*, to predict whether they are in contact or not. This can be viewed as a binary

classification problem. Each position in a window is usually a vector consisting of 20 numbers corresponding to the 20 profiles probabilities.

Fariselli and Casadio (2001) attempted to predict contact map by using neural network. The dataset is used is the same that is used for the I-site training and testing. The research (Fariselli et al., 1999b) is also a pioneer to the work in (Fariselli et al., 2001). The work done (Fariselli et al., 1999b) is used five different inputs for a neural network. The topology of the neural networks is a classical feed-forward with a standard back-propagation algorithm. Each of the five neural networks has one output neuron for the contact propensity. There is one hidden layer with two neurons, and the input layer has a different number neurons depending on the information encoded. All five of the networks contain an input layer that represents the normalized residue gap and another input for the normalized sequence separation. The first neural network, Netl, contains 212 input neurons. The first 210 neurons represent all the possible order couples of residues. The input neuron that encodes for the residues at positions *i* and *j* is set to 1, and the other 209 are set to 0. The rest of two input neurons are for the protein length and sequence separation, respectively. The next neural network, Net2, has 214 input neurons. The first 212 inputs are the same as Netl, and the two new inputs encode the hydrophobic neighborhood of the residue couples. The Net3 network has 216 input neurons. The two new inputs in Net3 encode for the conservation weights of the positions *i* and *j* in the alignment. The next network, Net4, adds much more information. This network uses a window of both parallel and antiparallel pairing of both *i* and *j.* The last network, Net5, has 1056 input neurons since two more inputs are added for the conservation weights as in Net3. The accuracy for the Netl on all proteins is 0.144. For Net2, accuracy is 0.147. Net3 accuracy is 0.150. For Net4 accuracy is 0.151. Net5 accuracy is 0.160.

The work done by Fariselli et al. (2001) focuses on whether more input features would help in determining the most accurate contact map; the test set these researchers used had five different neural networks. Each successive neural network used a more complex set of input features. This work adds to the work done in (Fariselli et. al., 1999a) by adding more encodings to the inputs for the neural

networks. The accuracy of the Neural Networks and the random predictor are the same as in the paper (Fariselli et. al., 1999a).

Zhao and Karypis (2003) also presented an approach for protein contact map prediction based on machine learning algorithm that employed SVM in their study. They identified five sets of features that capture different aspects of the amino acids and locations *i* and *j*: sequence conversation, sequence separation, correlated mutations analysis, predictied secondary structures and sequence profiles. They trained 15 SVM models by using set of features. They combined the five sets of features in various ways to evaluate their effectiveness. The dataset they used in training and testing their predictors, contained 177 proteins with known 3D structure from PDB. Their predictor achieved an avarage accuracy of 0.2238. Their experimental results indicated that different set of features achieved the best performance for various protein fold families.

The other paper is prepared by Cheng and Baldi (2007). It was about contact prediction using Support Vector Machines (SVM) and large future set. They developed contact map predictor called SVMcon integrated large number of useful information including profiles, secondary structure, relative solvent accessibilty, contact potentials, residue types, segment window information and protein level information. In the comparision with CMAPpro, they used the same training and testing dataset that used to CMAPpro. For long-range contacts, the accuracy of the new CMAPpro predictor is close to 30%, a significant increase over existing approaches. The method yielded a 4% improvement over the CMAPpro.

In 2009, Tegge et. al. attempted to predict contact map of by fast and reliable predictor called NNcon 2D-recursive neural network models. NNcon was ranked among the most accurate residue contact predictors in the Eighth Critical Assessment of Techniques for Protein Structure Prediction (CASP8), (Tegge et. al., 2009).

Yang and Chen (2011) tried to predict which combined the prediction result from several predictors by using a logistic regression techniques. This techniques called LRcon has two dataset. The first dataset was collected the targets in the CASP9 experiment. The second dataset was harvested from Protein Data Bank (PDB) using the selected protein chains from the PDB_select 25%. This dataset, they

were denoted by D856, has total of 856 chains. LRcon achieved 41.5% accuracy on the D856 dataset for the top L/10 longe-range contact predictions. The length *L* refers to the sum of the lengths of helix/strand elements in the protein sequence. (Di Lena, 2012)

Howe and Mohamad et al. (2011) also presented an approach for contact map prediction based on SVM model in their study. They were used 424 and 48 proteins for both training and testing respectively. Their research was to construct different prediction models that consist of different combination of features. There were total of five different kinds of features applied in their research which were local window feature, pairwise information feature, central segment window features segment avarage information feature and protein information feature. Their research combined different features into several combinations and was used to construct several prediction models. In their research, with the construction of the multiple prediction models with different combination of features, effectiveness of the features that affect the performance of the prediction are identified, and further improve the knowledge regard to effective information to be used in protein residue contact prediction.

An intelligent prediction technique is proposed (Di Lena et. al., 2012) to determine whether any two residues in a protein sequence are spatially close to each other in tertiary strucure. They developed machine learning method for residue contact map prediction using three processes. First, they applied 2D recursive neural networks to predict coarse contacts and orientations between secondary structure elements. Second, they applied an energy based method to align secondary strucutre elements and predict contact probabilities between residues in contacting alpha helices or beta strands. Third, they applied a deep neural network architecture to organize and progressively refine the prediction of contacts, integrating information over both space and time. They trained the architecture on a large set of proteins and test it on a large set. These architectures had been shown to predict contact map with an accuracy close to 30%.

As seen from above mentioned papers, contact map prediction is a popular hot topic for researchers in bioinformatics. Different methods are used for contact

map prediction from protein sequence information and residue features. Their accuracies are still quite low. Different ideas are given by the previous studies. Previous studies show that machine learning algortihms have disadvantages and advantages on contact map prediction is used by current feature sets. Consequently, different viewpoints which provide previous studies and their results get used to gain to the study.

3. MATERIAL AND METHODS

3.1. Material

3.1.1. Dataset

In the problem of residue contact map prediction, proteins which include residues are selected from Protein Data Bank (PDB). The Protein Data Bank (PDB) is a repository for 3-D structural data of large biological molecules such as proteins. These data, typically obtained by X-ray crystallography or NMR spectroscopy and submitted by biologists and biochemists, from around the world, also are freely on the internet websites of its member organisation such as PDBe, RCSB. The RCSB website is publicly accessible at http://www.rcsb.org/ .

Protein selection in the process is recommended that proteins, which are used before study, used in the previous study results (Howe and Mohammad, 2011). However, this thesis datasets, which is not taken completely before studies (Pollastri and Baldi 2003; Cheng and Baldi 2007; Di Lena et. al., 2012) datasets, is tought that used datasets are original. In this reason all data are collected, processed, and created before. It is showed below the Table 3.1. that the proteins length ranging from 33 to 203.

Table 3.1. The protein database

Protein	Length	Protein	Length	Protein	Length	Protein	Length
1A32A	88	1DZOA	123	1A62A	130	1MSCA	129
1EN2A	89	1E9MA	106	1AIHA	170	1QJPA	171
1F94A	63	1E87A	118	1AMXA	180	1RSYA	152
1FK5A	93	1ELRA	131	1BMGA	98	1VHHA	162
1I27A	73	1F4PA	147	1CEWA	108	1WHIA	122
1JB6A	33	1H9OA	112	1CHDA	203	1XERA	103
1VFY	73	1HDKA	141	1CXQA	162	3NULA	130
1VIEA	62	1IDOA	189	1D2SA	170	9WGAA	171
2OVOA	56	1J75A	67	1DBUA	158		
3PYPA	125	1JSFA	130	1MAIA	131		

In this thesis, residues base assessment is used instead of protein base assessment in different from previous studies. Proteins are catagorized are not test dataset and train dataset. All of the proteins which are belong to residues dataset are created. These residues datasets which are independent proteins divide into two which are 75% train dataset and 25% test dataset. The main factors which led to this approach, among the residues number of the contacts datasets are few in the datasets which more or less 3% take place.

3.1.2. CMView

Whether in this thesis, protein chain located interresidues contact or not is used CMView software (Vehlow et. al., 2011). CMView is an interactive contact map visualization and analysis tool for protein structures. CMView also allows users to display the contact map and interact with it as well as to show features of the contact map in the corresponding 3-dimensional structure by using the PyMol molecular viewer.

Real contacts (CMView, which is given PDB file, generates the results), which is used desired output in the study, is required for multiple selection and definitions that corresponding to use contact map definition in the literature.

In defining pairs of residues contacts in a protein chain, there are two decisions to be taken. The first decision is to chose the reference points. There are two sensible choices: alpha carbons or beta carbons. In this thesis, as an atom choice backbone carbon atoms are selected. (Donmez, 2008)

The second decision involves the threshold between the chosen reference points. In this thesis, threshold between residue pairs was chosen as 8 Å. Another subject among the residue contacts ouster of local contacts. For this reason among the residues which is chosen sequence separation is equal or bigger than five.

3.1.3. Features

There are three features in the dataset for this thesis: amino acid sequence which is information concerning primary structure, protein secondary structure information, and relative solvent accessibility. Amino acid sequence which is information about the primary structure was taken from PDB by FASTA format. Secondary structure and solvent accessibility information are acquired from the SCRATCH software that is a server for predicting protein tertiary structure. The SCRATCH server suite includes predictors for secondary structure, relative solvent accessibility, disordered regions, domains, disulfide bridges, single mutations stability, residue contacts versus average and tertiary structure. (Cheng et. al., 2005). The SCRATCH is publicly accessible as a web server at http://scratch.proteomics.ics.uci.edu/.

In this thesis, SSPro module which located in SCRATCH software was used for secondary structure while ACCPro module was used for solvent accessibility. These SSPro and ACCPro modules take the amino acid sequence as input, and output a categorical prediction for each residue. For secondary structure, the program predicts and separates each residue of the protein into three class: helix (H), sheet (E) or neither (C). For relative solvent accessibility, there are only two categories; namely buried (b) and exposed (e). Thus ACCPro module which include SCRATCH predicts each residue of protein into two class.

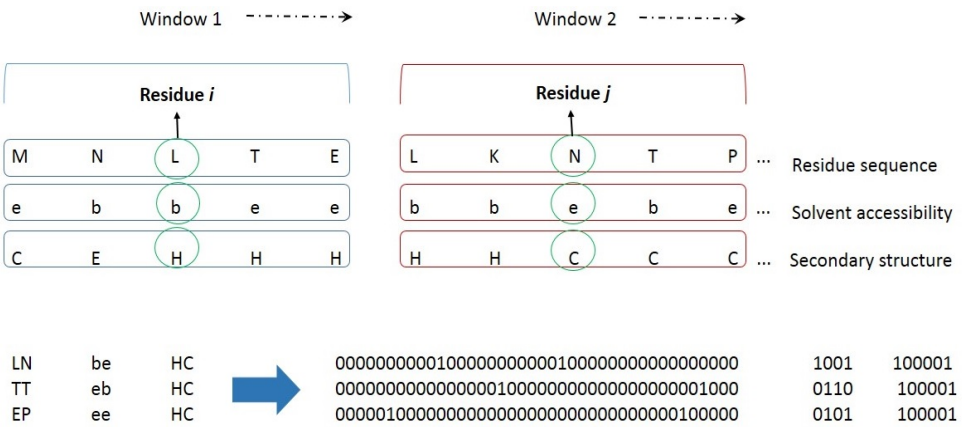


Figure 3.1. Example of dataset format

Table 3.2. The transformation of features

Feature Type	Class	Abbreviation	Binary Encoding
Secondary Structure	Helix	H	100
	Sheet	E	010
	Neither	C	001
Relative Solvent Accessibility	Buried	b	10
	Exposed	e	01

Table 3.3. List of amino acids and codes that represent them

Amino Acid	Three Letter Code	One Letter Code	Binary Encoding
Alanine	ALA	A	10000000000000000000
Arginine	ARG	R	01000000000000000000
Asparagine	ASN	N	00100000000000000000
Aspartic Acid	ASP	D	00010000000000000000
Cysteine	CYS	C	00001000000000000000
Glutamic Acid	GLU	E	00000100000000000000
Glutamine	GLN	Q	00000010000000000000
Glycine	GLY	G	00000001000000000000
Histidine	HIS	H	00000000100000000000
Isoleucine	ILE	I	00000000010000000000
Leucine	LEU	L	00000000001000000000
Lysine	LYS	K	00000000000100000000
Methionine	MET	M	00000000000010000000
Phenylalanine	PHE	F	00000000000001000000
Proline	PRO	P	00000000000000100000
Serine	SER	S	00000000000000010000
Threonine	THR	T	00000000000000001000
Tryptophan	TRP	W	00000000000000000100
Tyrosine	TYR	Y	00000000000000000010
Valine	VAL	V	00000000000000000001

3.2. Methods

In this thesis, several machine learning methods like neural network, logistic regression and support vector machines for prediction of protein contact map are proposed. For this reason, other machine learning methods are briefly mentioned.

3.2.1. Artificial Neural Networks

The brain is approximately composed of about 10 billion neuron (or nerve cell), each connected to about 10000 other neurons in the cortex. Artificial Neural Networks (ANN) are a computer algorithm that mimic the brain inspired by biological neural systems.

ANN are consisted artificial neurons or processing elements, interconnected with weights to solve specific problems. An artificial neuron is a computational model inspired in the natural neurons. Each natural neuron receives electrochemical signals from other natural neurons at the dendrites. If the sum of these electrical signals received is sufficiently strong (exceeds a certain threshold) to activate the neuron, it transmits an electrochemical signal along the axon, and passes this signal to the other neurons whose dendrites are attached at any of the axon terminals.

An artificial neural network consists of neurons which are connected by synaptic connections. Learning occurs in an artificial neural network by adjustment of these connections. Each neuron model has weighted inputs, activation function and one output. The Figure 3.2 shows the model for a neuron.

First, the synapses of the neuron are characterized by weights which are the strengths of the connections between a neuron and an input. In the next step, an adder sums up all the weighted input signals. This combination is a linear operation.

Finally, the sum of the input signals is passed through the activation function which is typically nonlinear (Haykin, 1994). In Figure 3.2, the arriving signals, called inputs, multiplied by the connection weights (adjusted) are first summed (combined) given in Equation (3.1.) and then passed through an activation function to produce the output neuron given in Equation (3.2.).

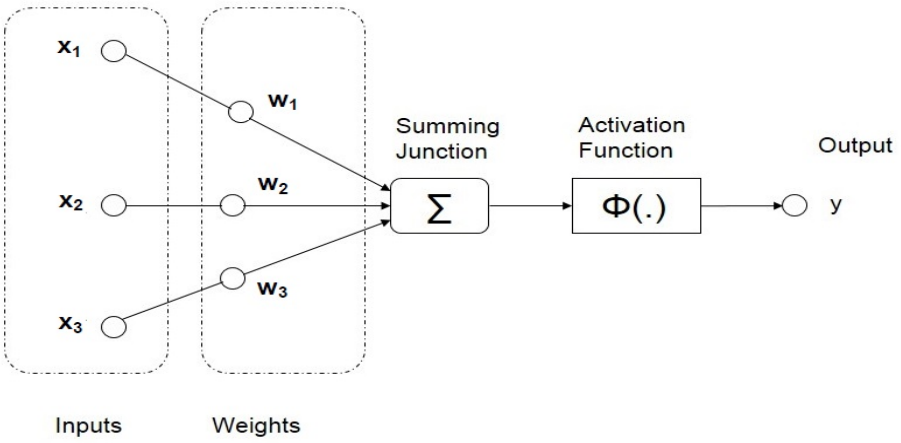


Figure 3.2. Model of an artificial neuron

$$Summation\ junction = \sum x_i w_i \tag{3.1.}$$

$$y = \varphi(\sum x_i w_i) \tag{3.2.}$$

Learning algorithm are defined into three main paradigms: supervised learning, unsupervised learning and reinforcement learning. The goal in supervised learning is to predict the target values from the input variables. The supervised neural network reads the input and output values in the training dataset and makes changes to the values of the weighted connections to reduce the difference between the predicted output and target output values which called error. This is repeated in many iterations (this is called epoches) until the network reaches specified level of accuracy or stopping criteria.

In this thesis, the main objective is to prediction whether contact between residue pairs in the protein chain. Because of this reason, supervised learning task have been used and is emphasized especially.

3.2.1.1. Backpropagation Algorithm

Backpropagation can be used with any feed-forward network that uses a threshold function that is differentiable. It is this derivative function that we will use during training processing. To train the neural network a method should be determined to calculate the error. As the neural network is trained, the net is presented with samples from the training set. The result obtained from the neural network is then compared with the desired result that is part of the training set. The degree to which the output from the neural network matches this desired output is the error. To train the neural network, this error should be tried to minimize. The error the neuron connection weights should be modified to minimize the error. The choosen error function will calculate the error of the network. This error function should be differentiable. Because the network uses a differential threshold function the activations of the output neurons can be thought of as differentiable functions of the input, weights. If the error function is also differentiable error, such as the "sum of square" error function the error function itself is a differentiable function of these weights. This allows us to evaluate the derivative of the error using the weights, then using these derivatives weights and bias will minimize the error. There are several ways that weights that minimize the error. One of them is the gradient descent algorithm. The most popular is by using the gradient descent method. The algorithm evaluates the derivative of the error for backpropagation, because it propagates the errors backward through the network.

Backpropagation algorithm can be summarized as follow:

1. Represent a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and a scaling factor, how much lower or higher the output should be adjusted to match the desired output. This is the local error.
4. Adjust the weights of each neuron to lower the local error.

5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat from step 3 on the neurons at the previous level, using each one's "blame" as its error.

According to backpropagation algorithm that is presented by Rumelhart (Rumelhart, 1986) change of current connection between i^{th} and j^{th} neurons in k^{th} iteration is defined as $\Delta w_{ji}(k)$.

$$\Delta w_{ji}(k) = -\alpha \frac{\partial E}{\partial w_{ji}(k)} + \mu \Delta w_{ji}(k-1) \tag{3.3.}$$

Here, α is learning rate, μ momentum coefficient and $\Delta w_{ji}(k-1)$ is weight change of previous.

There would be possible to obtain fast and stable solutions by optimal values of learning and momentum coefficients. Hovewer, there are some restrictions over some applications. One of the most important problems of Multilayer Perceptron (MLP) is long training times to reach the optimal weight values. Delta-bar-delta, training algorithm that is developed for the learning process, accelerates training time.

Performance decreases when the size of network increases especially in pattern recognition problems. SCG algorithm is preferred especially for networks with samples of large sized. Selection of proper ANN structure, training method and choose of backpropagation algorithm type is necessary to perform the best result for the problem (Bektaş, 2013).

3.2.1.2. Delta Bar Delta Algorithm

Delta-bar-delta learning algorithm is used to arrange all the weight of network (Jacobs, 1988) and change of current connection between i^{th} and j^{th} neurons in k^{th} iteration is defined as $\Delta w_{ji}(k)$.

$$\Delta w_{ji}(k) = -\alpha_{ji}(k)\frac{\partial E}{\partial w_{ji}(k)} \tag{3.4.}$$

$$\Delta\alpha ji(k) = \begin{cases} A, & D_{ji}(k-1)\dfrac{\partial E}{\partial w_{ji}(k)} > 0 \\ -\alpha_{ji}(k), & D_{ji}(k-1)\dfrac{\partial E}{\partial w_{ji}(k)} \\ 0, & others \end{cases} \tag{3.5.}$$

$$D_{ji}(k-1) = (1-\theta)\frac{\partial E}{\partial w_{ji}(k-1)} + \theta\frac{\partial E}{\partial w_{ji}(k-2)} \tag{3.6.}$$

According to equations, the algorithm increases learning coefficients linearly and decreases geometrically.

3.2.2. Logistic Regression

One of the most basis and simplest classification model is linear model, which assigns a binary class $c \in \{1,2\}$ to the classified sample according to a score value given by the linear combination of the set of d features $x \in R^d$ and bias term such that

$$c = \begin{cases} 1, & if \quad \beta_0 + \beta^T x < 0 \\ 2, & if \quad otherwise \end{cases} \tag{3.7.}$$

where $\theta = (\beta_0, \beta^T)^T$ are the model parameters.

In logistic regression, parameters are estimated such that the posterior probability of the model given training data is maximized. As the first step towards understanding the basics of the logistic classifier, this is the logistic function.

$$P(t) = \frac{1}{1 + e^{(-t)}} \tag{3.8.}$$

The idea of the logistic regression model is to use the logistic function to map the linear combination of the set of d features $x \in R^d$ into an estimate of the probability that x belongs to the c^{th} class. Thus, logistic regression is a form of probabilistic classification.

In a binary classification problem, the probability of class one given the d-dimensional feature vector $x \in R^d$ is modeled as

$$P(C = 1 \mid X = x) = \frac{1}{1 + e^{(-(\beta_0 + \beta^T x))}} = \frac{e^{(\beta_0 + \beta^T x)}}{1 + e^{(\beta_0 + \beta^T x)}} \tag{3.9.}$$

Where $\beta_0 \in R$ and $\beta \in R^d$ are the model parameters, collectively denoted by the parameter vector $\theta = (\beta_0, \beta^T)^T$.

The generalization of the logistic regression model into multinomial case for classes $c = 1, \ldots, K$ is achieved by modeling the probability of each class separately as

$$p(c \mid x) = \frac{e^{(\beta_{c0} + \beta_c^T x)}}{1 + \sum_{k=1}^{K-1} e^{(\beta_{k0} + \beta_k^T x)}} \qquad c=1,\ldots,K\text{-}1 \tag{3.10.}$$

$$p(K \mid x) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{(\beta_{k0} + \beta_k^T x)}} \tag{3.11.}$$

where $p(c \mid x)$ is a short hand notation for $p(C=c \mid X=x)$.

The Equation (3.11) gives the tradational way of defining the multinomial logistic regression model. Alternative and slightly more convenient definition is given by the symmetric model that models the class probabilities as

$$p(c \mid x) = \frac{e^{(\beta_{c0} + \beta_c^T x)}}{\sum_{k=1}^{K} e^{(\beta_{k0} + \beta_k^T x)}} \qquad c=1,..., K \qquad (3.12.)$$

The parameter vector is now of the form $\theta = (\beta_{10}, \beta^T_{1,} ..., \beta_{K0,}, \beta^T_{K})^T$ regardless of whether the tradational or symmetric model is used, the predicted class $\dot{c} = 1, ...,K$ for sample x is given by the maximum of the class probabilities, i.e.,

$$\dot{c} = argmax\ p(c \mid x)$$

The above maximization problem is equivalently written as

$$\dot{c} = argmax\ log(p(c \mid x)) = argmax\ g_c(x)$$

In logistic regression, the parameter estimation is usually done with maximum likelihood. Maximum likelihood method maximizes the probability of getting the observed results given the fitted regression coefficients. A consequence of this is that the goodness of fit and overall significance statistics used in logistic regression is different from linear regression.

$$L(\theta; D) = \prod_{i=1}^{N} p(c_i \mid x_i) \qquad D = \{x_i, c_i\}_{i=1}^{N}, \qquad (3.13.)$$

Equivalently, one can maximize the log-likelihood function

$$l(\theta) = \log L(\theta; D) = \sum_{i=1}^{N} \log p(c_i \mid x_i) \tag{3.14.}$$

The log-likelihood function in Equation (3.14) behaves nicely from optimization point of view because it is concave and has analytical first and second derivatives (Manninen, 2014).

3.2.3. Support Vector Machines (SVM)

One of the most popular machine learning techniques is the SVM which depends on the statistical learning theory in recent years. The SVM is developed by Vapnik (Vapnik, 1995) to solve classification problem, but later it has been extended to solve both classification and regression problems. The SVM classifier is widely used in bioinformatics due to its high accuracy.

The key idea of the SVM is mapping the input vectors to a high dimensional space non-linearly in order to reduce the effect of dimensionality. A linear decision area is constructed in this space and this area provides high generalization ability without errors. The advantage of the method is to be able to overcome two problems. Firstly, it can deal with the complex functions at high dimensional space and has high generalization ability by using quadratic optimization formulation. It also performs on numerical optimization in a high dimensional space by using the dual kernel representation of linear functions (Cherkassky and Mulier, 1998).

Data point is represented by an (n)-dimensional vector where n denotes the number of the criteria. Set of this kind of data points are tried to be separated into classes geometrically with an (n-1) dimensional hyperplane. These separative hyperplanes are also called as linear classifiers. SVM aims to find to optimal hyperlane. There are many hyperplanes; however, the best one is the one that represent the largest margin between the two classes. This hyperplane is called as the maximum-margin hyperplane and the linear classifier defined by this hyperplane is called as a maximum margin classifier.

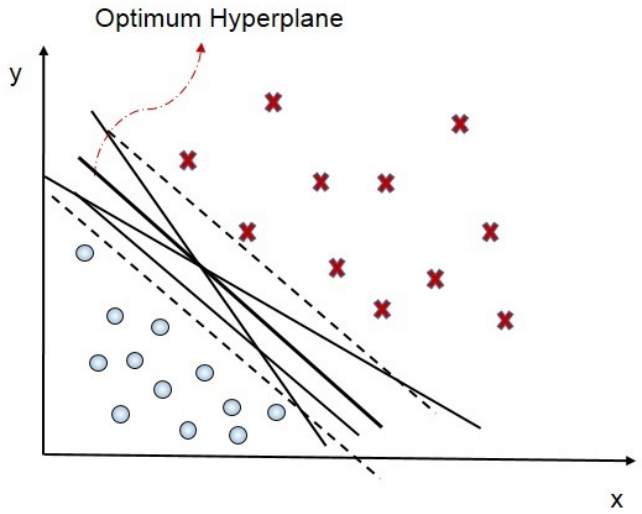


Figure 3.3. A hyperplane classifies two classes of training data which is two dimensional.

The objective is to correctly classify data in calculating SVM. The decision boundary must classify all points correctly and should prevent data points from falling into the margin. Let n be the dimension of the feature vector and $\{x_1,...,x_n\}$ be the representation of a sequence x and let $y_x \in \{1,-1\}$ be the class label of x. The feature vectors of the training set are used to build classifier. Weight vector w is of the same dimension as the feature vector and it is represented as $w=\{w_1,...,w_n\}$. The label of the sequence is predicted as 1 if $w_x+b \geq 1$ and it is predicted as -1 if $w_x+b \leq 1$ where, b is a threshold.

Therefore,

a) if $y_x=1;\ wx+b \geq 1$

b)if $y_x=-1;\ wx+b \leq 1$ (3.15.)

The equation of the margin γ_x is given by

$$\gamma_x = y_x\ (wx+b) \quad (3.16.)$$

The value of γ determines the accuracy of SVMs in predicting the label of the sequence. If γ is positive, it implies the correct prediction, and a negative value of γ implies the incorrect prediction. The values of the weight vector w and threshold b are updated every time, if it has an incorrect prediction.

SVM consists simple process: the weight vector w and threshold b are initialized to 0. k keeps count of the number of incorrect predictions which is also initialized to 0. R is the radius of the hypersphere and initialized to the maximum distance between the origin of the hypershere and a training vector. N is the total number of training vectors. η shows the learning rate. The decision hyperplane is given by the equation

$$h(x) = sgn\ (w_k x + b_k) \tag{3.17.}$$

The final weight vector w is in the form

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{3.18.}$$

where α_i is the number of incorrect predictions made on the example. By using Equation (3.17) and Equation (3.18), the equation for the hyperplane becomes

$$h_x = \operatorname{sgn} \sum_{i=1}^{N} (\alpha_i y_i \langle x_i, x \rangle + b) \tag{3.19.}$$

3.2.3.1 Soft Margin Classifier

In real world scenarios the decision boundaries might appear curved and there could be no straight line dividing the data points into two different classes in the data space. It is desirable to have a smooth decision boundary which ignores a few data points. By doing this, it is straight instead of being curved around some outliers.

Thus, we allow error ζ_i in the classification. ζ_i are just "slack variables" in optimization theory.

$$\left.\begin{array}{ll} wx_i + b \geq 1 - \zeta_i & y_i = 1 \\ wx_i + b \geq \zeta_i & y_i = -1 \\ \zeta_i \geq 0 & \end{array}\right\} \quad (3.20.)$$

it is needed to minimize

$$\min\left[\frac{\|w\|^2}{2} + C\sum_{i=1}^{r}\xi_i\right] \quad (3.21.)$$

where C is the tradeoff parameter between error and margin.

Some of training datasets outliers optimal hyperplane so this problem is solved defining an artificial variable $(i\xi)$. Balance among maximizing the boundary and minimizing the misclassification is controlled with a C parameter which has positive values $(0 < C < \infty)$. Mathematical illustration of optimization problem for nonlinear datasets is in Formul (3.23).

According to the equation above restrictions are described as:

$$y_i(w.\varphi(x_i) + b) - 1 \geq 1 - \xi_i \quad (3.22.)$$

$$\xi_i \geq 0 \text{ and } i = 1,....., N \quad (3.23.)$$

Samples which are not linearly separability is shown in a high dimensional space which is described as feature space. Thus samples are discriminated and hyper plane can be determined between classes.

3.2.3.2. Kernel Trick

A separating hyper plane is used in the classification of linear data. However, in real world problems, the data sets are mostly non-linear. In such cases, kernels are used to non-linearly map the input data to a high-dimensional space. The new mapping is then linearly separable. The space x_i are in called the input space and the space of $\varphi(x_j)$ after transformation is called feature space. Linear operation in the feature space is equivalent to non-linear operation in input space.

Mathematically notation of SVM is $K(x_i,x_j)=\varphi(x).\varphi(x_j)$ which is identified as kernel functions. With the help of kernel function it is probable to make nonlinear transformations and kernel function allows samples to be linearly separable in high dimension. As a result of this, the decision rule of kernel function with two classes is in Equation (3.24) (Osuna at al., 1997).

$$f(x) = sign\left(\sum_i \alpha_i y_i \varphi(x).\varphi(x_i) + b \right) \tag{3.24.}$$

Kernel function and optimal parameters that belong to this function are defined for classification with SVM. Whenever mathematical notation is seem to be easy the increase the degree of the polynomial may cause the algorithm to be more complicated. This issue not only significantly increases the duration of process but also decreases the classification accuracy. In this thesis, polynomial kernel is used. The types of kernel functions usually used with SVM are:

1. Polynomial kernel with degree d.

$$K(x,y) = ((x.y)+1)^d \tag{3.25.}$$

2. Normalized polynomial kernel with degree d.

$$K(x,y) = \frac{((x.y)+1)^d}{\sqrt{((x.x)+1)^d ((y.y)+1)^d}} \quad (3.26.)$$

3. Radial Basis Function kernel with kernel dimension γ.

$$K(x,y) = e^{-\gamma\|(x-x_i)\|^2} \quad (3.27.)$$

4. Pearson vii (puk) kernel with pearson expanse parameters σ,ω.

$$\frac{1}{\left[1+\left(\frac{2.\sqrt{\|x-y\|^2\sqrt{2^{(1/\omega)}-1}}}{\sigma}\right)^2\right]^{\omega}} \quad (3.28.)$$

3.2.4. Other Machine Learning Methods

In this thesis random tree, random forest, part, dagging, decorate, random committee, k-nearest neighbor methods are studied in addition to neural networks, logistic regression and, support vector machines algorithms. Comprised thesis main methods which are three methods compared with only the other algorithms results, for seven methods which has not got detail information are not placed.

The random forest machine learner, is a meta-learner; meaning consisting of many individual learners (many decision trees). The random forest uses multiple random trees classifications to votes on an overall classification for the given set of inputs. In general in each individual machine learner vote is given equal weight. This algorithm was modified to perform both un-weighted and weighted voting. The forest chooses the individual classification that contains the most votes.

Random trees have been introduced by Leo Breiman (2001). The algorithm can deal with both classification and regression problems. Random trees is a

collection (ensemble) of tree predictors that is called *forest* further in this section (the term has been also introduced by L. Breiman). The classification works as follows: the random trees classifier takes the input feature vector, classifies it with every tree in the forest, and outputs the class label that received the majority of "votes". In case of a regression, the classifier response is the average of the responses over all the trees in the forest.

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. A random forest is a classifier consisting of a collection of tree structured classifiers *{h(x,Θk), k=1, ...}* where the *{Θk}* are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input *x*. Random forests are an effective tool for prediction (Breiman, 2001).

K-nearest-neighbor classification is one of the most fundamental and simple classification methods. K-nearest neighbors is a instance-based algorithm, also known as a lazy learning algorithm (Aha et. al., 1991). The algortihm is based on the idea the similar instance tend to have similar solution. For each instance in the training set, it finds the *k* nearest instance and assign the original instance to the class which occurs the most within the *k* nearest objects. (Gou et. al., 2011).

Dagging creates a number of disjoint, stratified folds out of the data and feeds each chunk of data to a copy of the supplied base learner. Predictions are made via majority vote for classification problems and via averaging procedure for regression problems. (Anyfantis et. al., 2007)

This meta classifier creates a number of disjoint, stratified folds out of the data and feeds each chunk of data to a copy of the supplied base classifier. Predictions are made via majority vote, since all the generated base classifiers are put into the vote meta classifier. Useful for base classifiers that are quadratic or worse in time behavior, regarding number of instances in the training data. (Darzentas, 2008)

In another classification meta-learner (DECORATE, Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples) is presented that uses a learner (one that provides high accuracy on the training data) to build a

diverse committee. This is accomplished by adding different randomly constructed examples to the training set when building new committee members. These artificially constructed examples are given category labels that disagree with the current decision of the committee, thereby directly increasing diversity when a new classifier is trained on the augmented data and added to the committee. (Anyfantis et. al., 2007)

Rule learners are prominent representatives of supervised machine learning approaches. Basically, this type of learner tries to induce a set of rules for a collection of training instances. These rules are then applied on the test instances for classification purposes. Two well-known members of the family of rule-learners are C4.5 and RIPPER. Both approaches perform two steps to induce their rule sets. First, an initial rule set is determined, and second, these rules are adjusted or discarded according to a global optimization strategy. C4.5, for instance, generates an unpruned decision tree and transforms this tree into a set of rules. For each path from the root node to a leaf a rule is generated. Then, each rule is simplified separately followed by a rule-ranking strategy. Finally, the algorithm deletes rules from the rule set as long as the rule set's error rate on the training instances decreases. RIPPER implements a divideand-conquer strategy to rule induction. One rule is only generated at a time and the instances from a training set covered by this rule are removed. It iteratively derives new rules for the remaining instances of the training set. Frank and Witten (2005) describe a rule induction approach without the need for applying a global optimization strategy to generate appropriate rules. PART (Partial Decision Trees) adopts the divide-and-conquer strategy of RIPPER and combines it with the decision tree approach of C4.5. More precisely, PART generates a set of rules according to the divide-and-conquer strategy, removes all instances from the training collection that are covered by this rule and proceeds recursively until no instance remains. To generate a single rule, PART builds a partial decision tree for the current set of instances and chooses the leaf with the largest coverage as the new rule. Afterwards, the partial decision tree is discarded which avoids early generalization. The PART algorithm generates rules by repeatedly producing partial decision trees, and is a

hybrid of C4.5 and RIPPER. The algorithm works by forming pruned partial decision trees and converting these rules. (Berger et. al., 2006)

3.3. Performance Assessments

In bioinformatic machine learning applications, recall and specificity measures are widely used besides the common accuracy measure. Formal definitions for these performance criteria are given in Equations (3.27), (3.28), (3.29), (3.30) and (3.31) respectively and they are derived from the confusion matrix given in Table 3.4.

- True Positive (TP) : indicate the number of correct prediction that an instance is positive.
- True Negative (TN) : indicate the number of correct prediction that an instance is negative.
- False Positive (FP) : indicate that number of incorrect prediction that an instance is negative
- False Negatice (FN) : indicate that number of incorrect prediction that an instance is positive.

$$Accuracy = \frac{(TP + TN)}{(TP + FN + TN + FP)} \tag{3.27.}$$

$$\mathrm{Re}\,call(Sensitivity) = \frac{(TP)}{(TP + FN)} \tag{3.28.}$$

$$\mathrm{Pr}\,ecision = \frac{(TP)}{(TP + FP)} \tag{3.29.}$$

$$F - Measure = \frac{2.\mathrm{Re}\,call.\mathrm{Pr}\,ecision}{\mathrm{Re}\,call + \mathrm{Pr}\,ecision} \tag{3.30.}$$

$$Specificity = \frac{(TN)}{(TN + FP)} \tag{3.31.}$$

Table 3.4. Confusion matrix TP: true positive, FN: false positive, FP: false positive, TN: true negative

		Predicted	
		P	**N**
Actual	**P**	TP	FN
	N	FP	TN

The accuracy for unbalanced dataset cannot be used for the assessing usefulness of classification. Fort this reason there is an useful statistical characteristic that it is called Receiver Operating Characteristic (ROC).

Kappa Statistic (Cohen's Kappa coefficient) is a statistical measure of inter-rater or inter-annotator agreement for qualitative categorical items (Cohen, 1968). Kappa Statistic is a measure of the agreement between the predicted and the actual classifications in a dataset. *K=1* indicates excellent agreement, *K=0* indicates chance agreement.

$$\kappa = \frac{\Pr(\alpha) - \Pr(e)}{1 - \Pr(e)} \tag{3.32.}$$

where *Pr(a)* is the relative observed agreement among raters, and *Pr(e)* is the hypothetical probability of chance agreement.

Mean Absolute Error (MAE) can define as statistical measure of how far an estimate from actual values i.e. the average of the absolute magnitude of the

individual errors. It is usually similar in magnitude but slightly smaller than the root mean squared error.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\, e_i \,| \tag{3.33.}$$

Where e_i is average of absolute errors.

The Root Mean Square Error (RMSE) (also called the root mean square deviation) calculates the differences between values predicted by a model (an estimator) and the values actually observed from the thing being modeled (estimated). RMSE is used to measure the accuracy. It is ideal if it is small. Root Mean Squared Error (RMSE), Relative Absolute Error (RAE), and Root Relative Squared Error (RRSE) are given in Equations (3.34), (3.35), and (3.36) respectively.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(e_i)^2} \tag{3.34.}$$

$$RAE = \frac{\sum_{i=1}^{n}|\, p_i - a_i \,|}{\sum_{i=1}^{n}|\, a - a_i \,|} \tag{3.35.}$$

Where p_i is the predicted target value, and a_i is the actual target value.

$$RRSE = \frac{\sum_{i=1}^{n}(p_i - a_i)^2}{\sum_{i=1}^{n}(a - a_i)^2} \tag{3.36.}$$

4. EXPERIMENTS, RESULTS AND DISCUSSIONS

In the problem of prediction of residue contact maps, the first step for applying the machine learning techniques is to prepare the dataset. Belong to 38 proteins which consist of 297389 residue pairs are diminished 22238 residue pairs. This reason among the all residue pairs contact number is around 3%. In order to balanced dataset, it which contact quantity close rate noncontact is selected, and remainders of noncontacts are eleminated. Noncontacts which protein selected in the selection operation of two proteins (1MSCA and 1H9OA) having the highest performance results of the previously study (Cheng and Baldi, 2007) is selected. In the dataset 22238 instances are used. This residue pairs are 13116 noncontacts, 9122 residue pairs are contacts.

Three features in the dataset which are amino acid sequences, second protein secondary structures, and relative solvent accessibility are created with 50 inputs. This 50 inputs separate in two part which belong to 25 residues *i* and 25 residues *j*. Belong to 25 input residues *i* which are corresponded 20 input amino acid type, 3 input secondary structure, and 2 input relative solvent accessibility binary value is shown transformation Table 3.3.

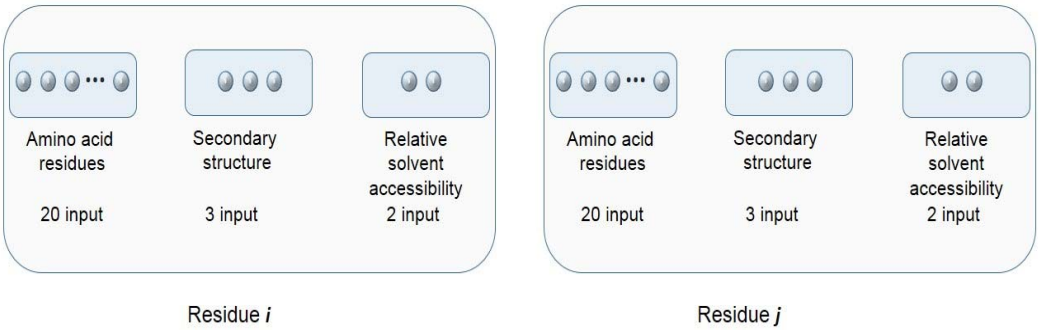


Figure 4.1. Inputs of Features

The goal of the thesis is to test the success of the prediction of the proposed method together with chosen optimum parameters and to compare the results with a few known methods. Furthermore, which machine learning algorithms suitable for the using of this thesis dataset is determined.

4.1. Classification by WEKA

All experiments were performed in WEKA machine learning tool. Weka is a collection of state of the art machine learning algorithms, written in Java, and available under the GNU licence and developed at the University of Waikato in New Zealand (Chaudhari et. al, 2012). Many classification schemes like decision trees, rule learners, naive Bayes, decision tables, locally weighted regression, SVM, instance based learners, logistic regression, voted perceptrons and multilayer perceptrons are supported by Weka. ANN, LR, SVM, random forest, random tree, k-nearest neighbors, dagging, decorate, part, and random committee are applied to the problem this thesis. The performance of an all classifier can be determined by the confusion matrix which is part of the text based result.

4.2. MLP (ANN) Results

Firstly, standard feedforward neural networks approaches use two windows around two target amino acids, *i* and *j*, to predict if they are in contact or not. This can be viewed as a binary classification problem. In WEKA, every row of data is considered as an instance and the features in the data are known as attributes. Results show different parameters such as correctly and incorrectly classified instances, kappa statistic, mean absolute and root mean squared error, relative absolute error, root relative squared error, confusion matrix.

Table 4.1 shows the classification accuracies and errors of MLP algorithms on the residues test data that contains 2 classes and 5559 instances. It also shows the number of correctly and incorrectly classified instances. From Table 4.15., the accuracy of classification using the MLP algorithm is 75.283%.

Table 4.1. Residue based contact prediction result (a) for MLP

Correctly Classified Instances	4185	75.283 %
Incorrectly Classified Instances	1374	24.717 %
Kappa statistic	0.471	
Mean absolute error	0.361	
Root mean squared error	0.432	
Relative absolute error %	74.713	
Root relative squared error %	87.920	
Total Number of Instances	5559	

Table 4.2 lists different evaluation criteria results for MLP methods on residue test data.

Table 4.2. Residue based contact prediction result (b) for MLP

	TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
	0.857	0.401	0.759	0.857	0.805	0.741	**n**
	0.599	0.143	0.741	0.599	0.663	0.74	**c**
W. Avg.	0.753	0.296	0.751	0.753	0.747	0.741	

Table 4.3 shows the confusion matrix for the classification results obtained from MLP classifier. Confusion matrix is a visualization tool typically used in supervised learning, which makes it easy to analyze the classification results. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. From the Table 4.3 it can be seen that all the 1349 instances belonging to contact class have been correctly classified.

Table 4.3. Residue based contact prediction confusion matrix result for MLP

n	**c**	**<-- Classified as**
2836	472	**n**
902	1349	**c**

Table 4.4 shows the classification accuracies and errors of MLP algorithms on the 1A32 test data that contains 2 classes and 3486 instances. It also shows the number of correctly and incorrectly classified instances. From Table 4.4, the accuracy of classification using the MLP algorithm is 68.646%.

Table 4.4. Protein 1A32 contact map prediction result (a) for MLP

Correctly Classified Instances	2393	68.646 %
Incorrectly Classified Instances	1093	31.354 %
Kappa statistic	0.022	
Mean absolute error	0.363	
Root mean squared error	0.513	
Relative absolute error %	88.229	
Root relative squared error %	124.348	
Total Number of Instances	3486	

Table 4.5 lists different evaluation criteria results for MLP methods on the 1A32 test data.

Table 4.5. Protein 1A32 contact map prediction result (b) for MLP result for MLP

	TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
	0.686	0.303	0.996	0.686	0.813	0.734	**n**
	0.697	0.314	0.021	0.697	0.040	0.734	**c**
W. Avg.	0.686	0.303	0.987	0.686	0.805	0.734	

Table 4.6 shows the confusion matrix for the classification results obtained from MLP Classifier on the 1A32 test data. From the Table 4.6 it can be seen that all the 23 instances belonging to contact class have been correctly classified.

Table 4.6. Protein 1A32 confusion matrix result for MLP

n	**c**	**<-- Classified as**
2370	1083	**n**
10	23	**c**

The total number of classes in the protein 1A32 test dataset is 2 and the number of instances are 3486. The classification accuracy is 68.646%, with 2293 samples belonging to 10 contact class getting misclassified.

Table 4.7 shows the classification accuracies and errors of MLP algorithms on the 1A62 test data that contains 2 classes and 7875 instances. It also shows the number of correctly and incorrectly classified instances. From Table 4.7, the accuracy of classification using the MLP algorithm is 55.149%

Table 4.7. Protein 1A62 contact map prediction result (a) for MLP

Correctly Classified Instances	4343	55.149 %
Incorrectly Classified Instances	3532	44.851 %
Kappa statistic	0.033	
Mean absolute error	0.482	
Root mean squared error	0.616	
Relative absolute error %	116.199	
Root relative squared error %	148.027	
Total Number of Instances	7875	

Table 4.8 lists different evaluation criteria results for MLP methods on the 1A62 testdata.

Table 4.8. Protein 1A62 contact map prediction result (b) for MLP

	TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
	0.546	0.263	0.987	0.546	0.703	0.668	**n**
	0.737	0.454	0.044	0.737	0.083	0.668	**c**
W. Avg.	0.551	0.268	0.961	0.551	0.686	0.668	

The total number of classes in the protein 1A62 test dataset is 2 and the number of instances are 7875. The classification accuracy is 55.149%, with 2293 samples belonging to 57 contact class getting misclassified.

Table 4.9. Protein 1A62 confusion matrix result for MLP

n	**c**	**<-- Classified as**
4183	3475	**n**
57	160	**c**

4.3. Logistic Regression Results

Table 4.10 shows the classification accuracies and errors of LR algorithms on the residues test data that contains 2 classes and 5559 instances. It also shows the number of correctly and incorrectly classified instances. From Table 4.10, the accuracy of classification using the LR algorithm is 73.826%.

Table 4.10. Residue based contact prediction result (a) for Logistic Regression

Correctly Classified Instances	4104	73.826 %
Incorrectly Classified Instances	1455	26.174 %
Kappa statistic	0.449	
Mean absolute error	0.360	
Root mean squared error	0.425	
Relative absolute error %	74.513	
Root relative squared error %	86.616	
Total Number of Instances	5559	

Table 4.11 lists different evaluation criteria results for LR methods.

Table 4.11. Residue based contact prediction result (b) for Logistic Regression

	TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
	0.810	0.367	0.764	0.810	0.786	0.787	**n**
	0.633	0.190	0.694	0.633	0.662	0.787	**c**
W. Avg.	0.738	0.295	0.736	0.738	0.736	0.787	

Table 4.12 shows the confusion matrix for the classification results obtained from LR classifier. From the Table 4.12 it can be seen that all the 1426 instances belonging to contact class have been correctly classified.

Table 4.12. Residue based contact prediction confusion matrix result for Logistic Regression

n	**c**	**<-- classified as**
2678	630	**n**
825	1426	**c**

Table 4.13 shows the classification accuracies and errors of LR algorithms on the 1A32 test data that contains 2 classes and 3486 instances. It also shows the number of correctly and incorrectly classified instances. From Table 4.13, the accuracy of classification using the LR algorithm is 91.279%.

Table 4.13. Protein 1A32 contact map prediction result (a) for Logistic Regression

Correctly Classified Instances	3182	91.279 %
Incorrectly Classified Instances	304	8.721 %
Kappa statistic	0.090	
Mean absolute error	0.240	
Root mean squared error	0.295	
Relative absolute error %	58.297	
Root relative squared error %	71.419	
Total Number of Instances	3486	

Table 4.14 lists different evaluation criteria results for LR methods on 1A32 test data.

Table 4.14. Protein 1A32 contact map prediction result (b) for Logistic Regression

	TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
	0.916	0.455	0.995	0.916	0.954	0.727	**n**
	0.545	0.084	0.059	0.545	0.106	0.727	**c**
W. Avg.	0.913	0.451	0.986	0.913	0.946	0.727	

Table 4.15 shows the confusion matrix for the classification results obtained from LR classifier. From the Table 4.15. it can be seen that all the 18 instances belonging to contact class have been correctly classified.

Table 4.15. Protein 1A32 confusion matrix result for Logistic Regression

n	**c**	**<-- classified as**
3164	289	**n**
15	18	**c**

Table 4.16 shows the classification accuracies and errors of LR algorithms on the 1A62 test data that contains 2 classes and 7875 instances. It also shows the number of correctly and incorrectly classified instances. From Table 4.16, the accuracy of classification using the LR algorithm is 68.203%.

Table 4.16. Protein 1A62 contact map prediction result (a) for Logistic Regression

Correctly Classified Instances	5371	68.203 %
Incorrectly Classified Instances	2504	31.797 %
Kappa statistic	0.044	
Mean absolute error	0.397	
Root mean squared error	0.459	
Relative absolute error %	95.677	
Root relative squared error %	110.294	
Total Number of Instances	7875	

Table 4.17 lists different evaluation criteria results for LR methods on 1A62 test data.

Table 4.17. Protein 1A62 contact map prediction result (b) for Logistic Regression

	TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
	0.685	0.415	0.983	0.685	0.807	0.684	**n**
	0.585	0.315	0.050	0.585	0.092	0.684	**c**
W. Avg.	0.682	0.412	0.957	0.682	0.788	0.684	

Table 4.18 shows the confusion matrix for the classification results obtained from LR Classifier. From the Table 4.18 it can be seen that all the 127 instances belonging to contact classs have been correctly classified.

Table 4.18. Protein 1A62 confusion matrix result for Logistic Regression

n	**c**	**<-- classified as**
5244	2414	**n**
90	127	**c**

4.4. Support Vector Results

Table 4.19 shows the classification accuracies and errors of SVM algorithms on the residues test data that contains 2 classes and 5559 instances. It also shows the number of correctly and incorrectly classified instances. From Table 4.19, the accuracy of classification using the SVM algorithm is 73.718%.

Table 4.19. Residue based contact prediction result (a) for SVM

Correctly Classified Instances	4098	73.718 %
Incorrectly Classified Instances	1461	26.282 %
Kappa statistic	0.446	
Mean absolute error	0.263	
Root mean squared error	0.513	
Relative absolute error %	54.384	
Root relative squared error %	104.426	
Total Number of Instances	5559	

Table 4.20 lists different evaluation criteria results for MLP methods on residue testdata.

Table 4.20. Residue based contact prediction result (b) for SVM

	TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
	0.814	0.375	0.761	0.814	0.787	0.719	**n**
	0.625	0.186	0.695	0.625	0.658	0.719	**c**
W. Avg.	0.737	0.299	0.734	0.737	0.735	0.719	

Table 4.21 shows the confusion matrix for the classification results obtained from SVM classifier. From the Table 4.21 it can be seen that all the 1406 instances belonging to contact class have been correctly classified.

Table 4.21. Residue based contact prediction confusion matrix result for SVM

n	**c**	**<-- classified as**
2692	616	**n**
845	1406	**c**

Table 4.22 shows the classification accuracies and errors of SVM algorithms on the 1A32 test data that contains 2 classes and 3486 instances. It also shows the number of correctly and incorrectly classified instances. From Table 4.22, the accuracy of classification using the SVM algorithm is 89.816%.

Table 4.22. Protein 1A32 contact map prediction result (a) for SVM

Correctly Classified Instances	3131	89.816 %
Incorrectly Classified Instances	355	10.184 %
Kappa statistic	0.076	
Mean absolute error	0.102	
Root mean squared error	0.319	
Relative absolute error %	24.722	
Root relative squared error %	77.401	
Total Number of Instances	3486	

Table 4.23 lists different evaluation criteria results for SVM methods on 1A32 test data.

Table 4.23. Protein 1A32 contact map prediction result (b) for SVM

	TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
	0.902	0.455	0.995	0.902	0.946	0.723	**n**
	0.545	0.098	0.050	0.545	0.092	0.723	**c**
W. Avg.	0.898	0.451	0.986	0.898	0.938	0.723	

Table 4.24 shows the confusion matrix for the classification results obtained from SVM classifier. From the Table 4.24 it can be seen that all the 18 instances belonging to contact class have been correctly classified.

Table 4.24. Protein 1A32 confusion matrix result for SVM

n	**c**	**<-- classified as**
3113	340	**n**
15	18	**c**

Table 4.25 shows the classification accuracies and errors of SVM algorithms on the 1A62 test data that contains 2 classes and 7875 instances. It also shows the number of correctly and incorrectly classified instances. From Table 4.25, the accuracy of classification using the SVM algorithm is 70.083%.

Table 4.25. Protein 1A62 contact map prediction result (a) for SVM

Correctly Classified Instances	5519	70.083 %
Incorrectly Classified Instances	2356	29.918 %
Kappa statistic	0.047	
Mean absolute error	0.299	
Root mean squared error	0.547	
Relative absolute error %	72.060	
Root relative squared error %	131.415	
Total Number of Instances	7875	

Table 4.26 lists different evaluation criteria results for SVM methods on 1A62 test data.

Table 4.26. Protein 1A62 contact map prediction result (b) for SVM

	TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
	0.704	0.429	0.983	0.704	0.821	0.638	**n**
	0.571	0.296	0.052	0.571	0.095	0.638	**c**
W. Avg.	0.701	0.425	0.957	0.701	0.801	0.638	

Table 4.27 shows the confusion matrix for the classification results obtained from SVM classifier. From the Table 4.27 it can be seen that all the 124 instances belonging to contact class have been correctly classified.

Table 4.27. Protein 1A62 confusion matrix result for SVM

n	**c**	**<-- classified as**
5395	2263	**n**
93	124	**c**

4.5. Comparison with Other Methods

The accuracies and recall of all ten classifiers have been represented in Table 4.28. As seen on table 4.28, methods of decorate and random committee have the highest rate of success in 10 methods of residue based prediction. However it will not be enough to take success ratio as evaluation criteria. Contact ratio must be considered together with the success ratio. As can be seen, decorate and random

forest methods are more successful according to other methods in residue based prediction.

Table 4.28. Residue based contact prediction result for all methods

Method	Acc	Recall	True contact/All contact
MLP	0,753	0,599	1349/2251
Logistic Regression	0,738	0,633	1426/2251
SVM	0,737	0,625	1406/2251
Random Forest	**0,795**	**0,697**	**1570**/2251
Random Tree	0,794	0,663	1493/2251
PART	0,797	0,689	1550/2251
Dagging	0,739	0,614	1382/2251
Decorate	**0,802**	**0,693**	**1559**/2251
Random Committe	0,802	0,681	1534/2251
K-Nearest Neighbors	0,790	0,643	1448/2251

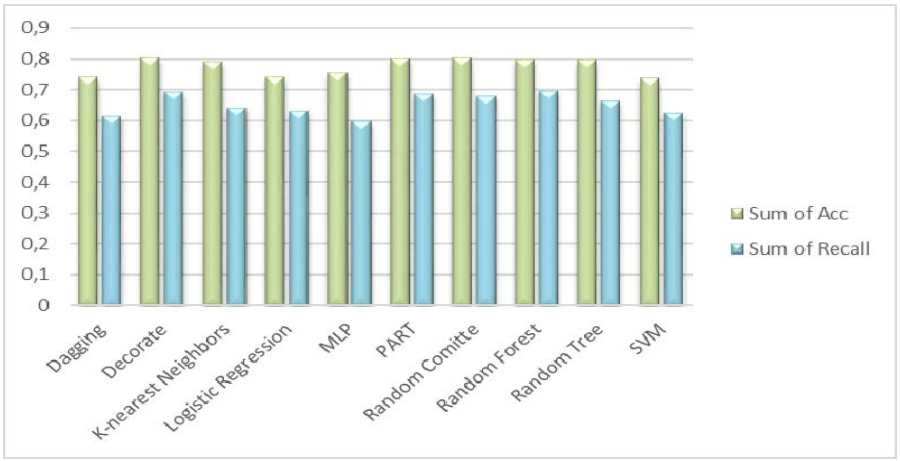


Figure 4.2. Accuracy and recall values of residue based prediction result

As is evident from Table 4.29, LR and dagging have higher accuracy as contact and noncontact classifications for 1A32. On the other hand, accuracy rate is only not efficient as evaluation criteria. For this reason, recall value should also be considered. According to this evaluations, decorate and random committee methods stands out one step with dagging and LR methods according to the other methods.

Table 4.29. Protein 1A32 contact map result for all methods

Method	Acc	Recall	True contact/All contact
MLP	0,686	0,697	23/33
Logistic Regression	**0,913**	0,545	18/33
SVM	0,898	0,545	18/33
Random Forest	0,672	0,788	26/33
Random Tree	0,657	0,788	26/33
PART	0,737	0,788	26/33
Dagging	**0,913**	0,545	18/33
Decorate	**0,766**	**0,758**	25/33
Random Committee	**0,766**	**0,758**	25/33
K-Nearest Neighbors	0,744	0,788	26/33

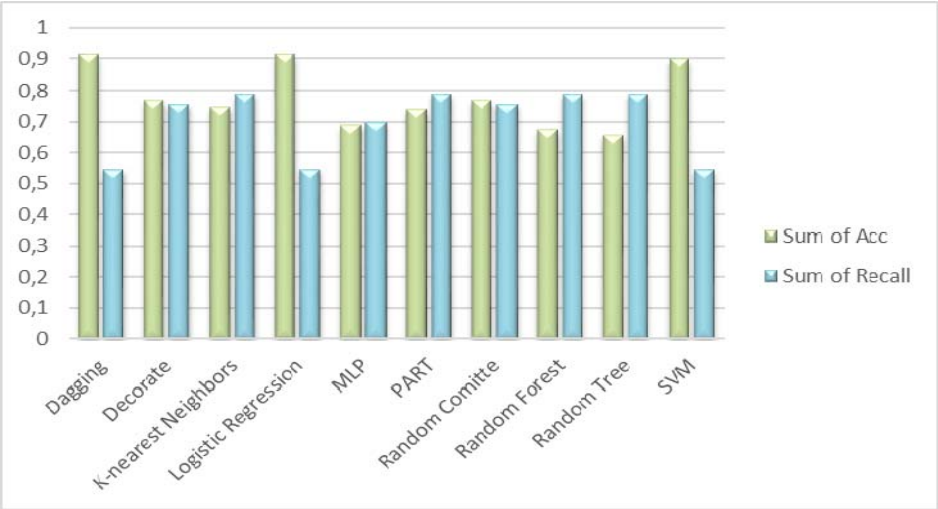


Figure 4.3. Accuracy and recall values of 1A32 contact map prediction result

As is evident from Table 4.30, SVM and dagging have higher accuracy as contact and noncontact classifications for 1A62. However, accuracy rate is only not enough as evaluation criteria. Therefore, recall value should also be considered. In the light of this assessment, besides the methods dagging and SVM, LR are said to be more successful methods.

Table 4.30. Protein 1A62 contact map result for all methods

Method	Acc	Recall	True Contact/All contact
MLP	0,551	0,737	160/217
Logistic Regression	**0,682**	0,585	**127**/217
SVM	**0,701**	0,571	124/127
Random Forest	0,514	0,783	170/217
Random Tree	0,553	0,760	165/217
PART	0,550	0,742	161/217
Dagging	**0,717**	0,562	122/217
Decorate	0,584	0,724	157/217
Random Committe	0,522	0,760	165/217
Knearestneighbours	0,623	0,760	165/217

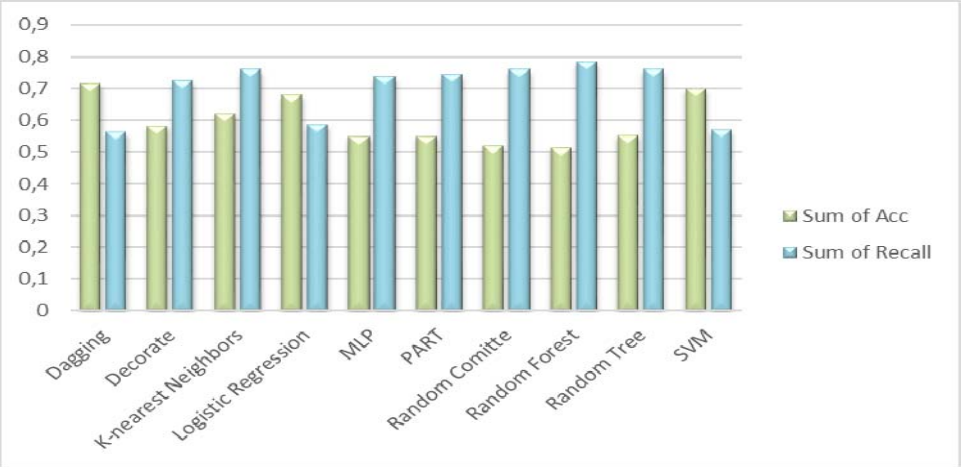


Figure 4.4. Accuracy and recall values of 1A62 contact map prediction result

Another conclusion gathered from experiment results is that success ratio of methods varies directly with test data. For example while decorate method stands out in residue based prediction, dagging and SVM methods stands out in protein based prediction.

5. CONCLUSIONS

Reveal the 3D structure of the proteins structure of the protein determines its function is a very important place for biology and bioinformatics. In this thesis, protein 3D structure is tried to predict whether among protein residue contact determining by using machine learning methods or not.

Previous studies dataset contains a lot of proteins, predicting protein which is unknown its structure before, contact map create a primary object. On the other hand, unlike previous study this thesis, prepared dataset prediction of protein contact map which is more suitable for the machine learning algorithms is aimed to reveal. Hence, the dataset used in this thesis is smaller than previous used the dataset studies, intstead of prediction of protein contact map prediction of residues contact success is examined.

The names of methods used in experiments are neural networks, logistic regression, support vector machines, random forest, random tree, dagging, decorate, part, random committee, and k-nearest neighbors. The results of methods are compared. At the point, when the resultant data is extracted throughout experiments performed.

The thesis indicates that residue base prediction gives the best result with decorate method. But, 1A32 protein structure prediction gives the best result with logistic regression, and 1A62 protein structure prediction gives the best result dagging method. In this study salient issues, in a contrast general success rate is low, contact prediction which is the best successful result of the random forest method has come forward. As general assessment, it is appropriated to observe that decorate, logistic regression, dagging and support vector machines methods on the dataset, and these methods are suitable for protein residue contact map prediction too.

6. FUTURE WORKS

The protein contact map prediction is a difficult problem in protein structure prediction. However, Protein 3D structure which has an important information about protein contact map by conducted study has advanced rapidly for the last past decade. Machine learning approach become effective technique for protein contact map prediction.

For the future work of this thesis Restricted Boltzmann Machines (RBMs) can apply on proteins residue dataset that is binary form. This format is useful for RBMs. Restricted Boltzmann Machine is a stochastic neural network. It consists of one layer of visible units (neurons) and one layer of hidden units. Units in each layer have no connections between them and are connected to all other units in other layer. Connections between neurons are bidirectional and symmetric. This means that information flows in both directions during the training and during the usage of the network and that weights are the same in both directions (Salakhutdinov, 2009a, 2009b). Nowadays, popularities rising deep learning which is called machine learning is shown efficient method of RBMs within study field recently. It is probably expected that this method has a succesful result a about protein contact map area.

Besides that used of this thesis dataset is expanded. Among the protein residues number of contacts are too lower than residue number of nonconctats (~3%). When the number of protein uses in dataset is higher, it is expected that like exposed of this thesis among the similar protein residues predict success is seen as the same as different properties of proteins. Moreover, different features can be added datasets, for instances correlated mutations, sequence profiles. Sequence separation can be changed, for instance long-range contacts.

In the foreseeable future, machine learning methods are played key role forecast of protein structure. Furthermore, in many cases machine learning methods are relatively fast compared to other methods. In bioinformatics, machine learning method will continue about protein structure prediction success and to cause time spent as little for studies conducted in this area to create oppurtunites.

REFERENCES

AHA, D.V., KIBLER, D., ALBERT, K.M., 1991. Instance Based Learning Algorithms. Machine Learning, 6(1): 37-66.

ANYFANTIS, D.S., KARAGIANNOPOULOS, M.G., KOTSIANTIS, S.B., PINTELAS, P.E., 2007. Local Dagging of Decision Stumps for Regression and Classification Problems. Proceeding of the 15th Mediterranean Conference on Control and Automation. Athens, Greece.

BEKTAŞ, J., 2013. Use of data Mining Techniques to Determine Presence of Coronary Artery Disease and Deriving a Risk Score by Employing Risk Factors. Dissertation Proposal, Çukurova University.

BERG, J., TYMOCZKO, J., STRYER L., 2001. Biochemistry. Section 4.5: Three-Dimensional Protein Structure Can Be Determined by NMR Spectroscopy and X-Ray Crystallography.

BERGER, H., MERKL, D., DITTENBACH, M., 2006. Exploiting Partial Decision Trees for Feature Subset Selection in eMail Categorization. Proceedings of the 2006 ACM Symposium on Applied Computing. New York, USA.

BJORKMAN, P.J., and PARHAM, P., 1990. Structure, Function and Diversity of Class I Major Histocompatibility Complex Molecules. Annual Review of Biochemistry, 59: 253–288.

BRANDEN C., TOOZE J., 1999. Introduction to Protein Structure, Garland, New York, 302p.

BREIMAN, L., 2001. Random Forest. Machine Learning, 45(1):5-32.

BRUNGER, A., 1992. X-PLOR: A System for Crystallography and NMR. Yale University Press, New Haven, CT, 418p.

CHAUDHARI, B., PARIKH, M., 2012. A Comparative Study of Clustering Algorithms Using Weka Tools. IJAIEM, 1(2): 154-158.

CHENG, J. and BALDI, P., 2007. Improved Residue Contact Prediction Using Support Vector Machines and a Large Feature Set, BMC Bioinformatics, 8: 113.

CHENG, J., RANDALL, A.Z., SWEREDOSKI, M. J., and BALDI, P., 2005. SCRATCH: A Protein Structure and Structural Feature Prediction Server. Nucleic Acids Research, 33: 72–76.

CHENG, J., TEGGE A. N., and BALDI P., 2008. Machine Learning Methods for Protein Structure Prediction. IEEE Reviews in Biomedical Engineering 1: 41-49.

CHERKASSKY, V., MULIER, F., 1998. Learning from Data: Concepts, Theory, and Methods. New York: Wiley, 538p.

CORTES, C., VAPNIK, V., 1995. Support Vector Networks, Machine Learning, 20(3): 273-97.

DARZENTAS, J., 2008. Artificial Intelligence: Theories, Models and Applications: 5th Hellenic Conference on AI, SETN 2008, Syros, Greece.

DI LENA, P., NAGATA, K. and BALDI, P., 2012. Deep Architectures for Protein Contact Map Prediction. Bioinformatics, 28: 2449-2457.

DILL, K., 1990. Dominant Forces in Protein Folding. American Chemical Society, 29(31):7133-7155.

DONMEZ, N., 2008. Protein Contact Map Prediction using Neural Networks and Associative Markov Networks. Msc Thesis, University of Toronto.

DRENTH, J., 1994. Principles of Protein X-ray Crystallography. Springer-Verlag New York Inc., New York, 345p.

FARISELLI, P., and R. CASADIO, 1999b. A Neural Network Based Predictor of Residue Contacts in Proteins. Vulo Protein Engineering Design and Selection 12(1): 15-21.

FARISELLI, P., OLMEA, O., VALENCIA, A., and CASADIO, R., 2001. Prediction of Contact Maps with Neural Networks and Correlated Mutations. Protein Engineering, 13: 835–843.

FARISELLI, P., RICCOBELLI, P., and CASADIO, R., 1999a. Role of Evolutionary Information in Predicting the Disulfide-bonding State of Cysteine in Proteins. Proteins, 36: 340–346.

GLASGOW, J., KUO, T., and DAVIES, J., 2006. Protein Structure from Contact Maps: A case-based Reasoning Approach. Information Systems Frontiers, 8: 29-36.

GOBEL, U., SANDER, C., SCHNEIDER, R., and VALENCIA, A., 1994. Correlated Mutations and Residue Contacts in Proteins. Proteins, 18: 309-317.

GOU, J., XIOUNG, T., KUANG, Y., 2011. A Novel Weighted Voting for K-Nearest Neighbor Rule. Journal of Computers, 6(5): 833-840.

GROMIHA, M.M., and SELVARAJ, S., 2004. Inter-residue Interactions in Protein Folding and Stability. Progress in Biophysics and Molecular Biology, 86: 235-277.

HAYKIN, S., 1994. Neural Networks: A Comprehensive Foundation, Prentice Hall New York.

HOWE, C.W., and MOHAMAD, M.S., 2011. Protein Residue Contact Prediction using Support Vector Machine, World Academy of Science, Engineering and Technology, 5: 1394-1399.

JACOBS, R.A., 1988. Increased Rates of Convergence Through Learning Rate Adaptation. Neural Networks, 1: 295-307.

JACOBSON, M., SALI, A., 2004. Comparative Protein Structure Modeling and Its Applications to Drug Design, Annual Reports in Medical Chemistry, 39: 259-276.

KENDREW, J.C., 1962. Myoglobin and the Structure of Proteins. Nobel Lecture, December 11: 676-698.

KRASNOGOR, N., 2004. Self Generating Metaheuristics in Bioinformatics: The Proteins Structure Comparision Case, Genetic Programming and Evolvable Machines, 5: 181-201.

LASKOWSKI, R.A., WATSON, J.D., and THORNTON, J.M., 2003. From Protein Structure to Biochemical Function? Journal of Structural and Functional Genomics, 4: 167–177.

MANNINEN, T., 2014. Predictive Modeling Using Sparse Logistic Regression with Applications. Phd Thesis, Tampere University of Technology.

OLMEA, O., and VALENCIA, A., 1997. Improving Contact Predictions by the Combination of Correlated Mutations and Other Sources of Sequence Information. Folding and Design 2: 25-32.

OLSON, M.V., 1993. The Human Genome Project. Proceedings of the National Academy of Sciences, 90(10): 4338-4344.

OSUNA, E., FREUND, R., and GIROSI, F., 1997. Support Vector Machines: Training and Applications, MIT AI laboratory.

PALUZEWSKI, M., 2008. Algorithms for Protein Structure Prediction. University Of Copenhagen, Phd Thesis.

PAULING, L., COREY, R., 1951. The Structure of Proteins: Two Hydrogen-Bonded Helical Configurations of the Polypeptide Chain. Proceedings of the National Academy Sciences, 37(4):205-211.

POLLASTRI, G., and BALDI P., 2002. Prediction of Contact Maps by GIOHMMs and Recurrent Neural Networks Using Lateral Propagation from All Four Cardinal Corners. Bioinformatics, 18: 62–70.

POLLASTRI, G., and BALDI, P., 2003. The Principled Design of Large-Scale Recursive Neural Network Architectures–DAG-RNNs and the Protein Structure Prediction Problem. Machine Learning Research, 4:575-602.

POULSON, V., 2007. Learning Protein Structure Contact Maps Using Neural Networks, Msc Thesis, Utah State University.

ROBINS, R.J., 1992. Challenges in the Human Genome Projects. IEEE Engineering in Biology and Medicine, 14: 746-759.

RUMELHART, D.E., HINTON, G.E., and WILLIAMS, R.J., 1986. Learning Representations by Back-Propagating Errors, Nature, 323: 533-536.

SALAKHUTDINOV, R. and HINTON, G. E., 2009a. Deep Boltzmann machines. In Proceedings of the International Conference on Artificial Intelligence and Statistics, 5: 448–455.

SALAKHUTDINOV, R., 2009b. Learning Deep Generative Models. Phd thesis, University of Toronto.

SANGER, F., Thompson E., O., 1953. The Amino-acid Sequence in the Glycyl Chain of Insulin. The Journal of Biochemistry, 53(3):366-374.

SHACKELFORD G., and KARPLUS, K., 2007. Contact Prediction Using Mutual Information and Neural Nets. Proteins, 69: 159–164.

TEGGE, A.N., 2009. NNcon: Improved Protein Contact Map Prediction Using 2D-Recursive Neural Networks. Nucleic Acids Research, 37:515-518.

TRAVERS, A., 1989. DNA Conformation and Protein Binding. Annual Review of Biochemistry, 58: 427–452.

VEHLOW, C., STEHR, H., WINKELMANN, M., DUARTE, J.M., PETZOLD, L., DINSE, J., LAPPE, M., 2011. CMView: Interactive Contact Map Visualization and Analysis. Bioinformatics: 27(11): 1573-1574.

VENDRUSCOLO, M., KUSSELL, E., and DOMANY, E., 1997. Recovery of Protein Structure from Contact Maps. Folding Design, 2: 295–306.

WITTEN, I.H., FRANK, E., 2005. Data Mining Practical Machine Learning Tools and Techniques. San Francisco, 525p.

WU, S. and ZHANG, Y., 2008. A Comprehensive Assessment of Sequence-based and Template-based Methods for Protein Contact Prediction, Bioinformatics, 24: 924-931.

YANG, J., Chen, X., 2011. A Consensus Approach to Predicting Protein Contact Map via Logistic Regression. Bioinformatics Research and Applications - 7th International Symposium, Changsha, China.

ZHAO, Y., KARYPIS, G., 2003. Prediction of Contact Maps Using Support Vector Machines. In Proceedings of the IEEE Symposium on Bioinformatics and BioEngineering: 26-36

BIOGRAPHY

Mustafa Özber YÜCEKUL was born on May 4, 1983 in Adana, TURKEY. He graduated from Hacı Ahmet Atıl High School in 2001. He received her B.Sc. degree from Electrical and Electronics Engineering Department of Mersin University in 2007.

After he graduated from University, he started as a master student at the Electrical and Electronics Engineering Department of Çukurova University. He worked as a engineer at Türk Telekom A.Ş. between August 2010 and January 2012. He is currently working as a engineer in Governorship of Adana at Republic of Turkey Interior Ministry since January 2012.