

**Sequence Based Computational Methods for
Protein Attribute Prediction and Phylogeny Reconstruction**

by

Mohammad Saifur Rahman

DOCTOR OF PHILOSOPHY



Department of Computer Science and Engineering

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

DHAKA-1000 BANGLADESH

July 2018

PhD Thesis

SEQUENCE BASED COMPUTATIONAL METHODS FOR
PROTEIN ATTRIBUTE PREDICTION AND PHYLOGENY RECONSTRUCTION

by
Mohammad Saifur Rahman

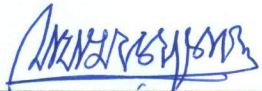
A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy
in
Computer Science and Engineering


Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)
Dhaka 1000

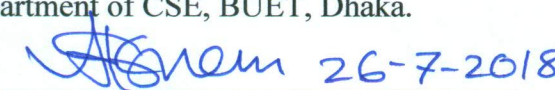
July 2018

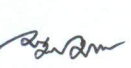
The thesis titled “SEQUENCE BASED COMPUTATIONAL METHODS FOR PROTEIN ATTRIBUTE PREDICTION AND PHYLOGENY RECONSTRUCTION”, submitted by Mohammad Saifur Rahman, Roll No. 1014054001P, Session October 2014, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Engineering and approved as to its style and contents. Examination held on July 26, 2018.

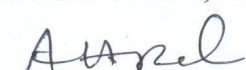
Board of Examiners


1. 


Dr. M. Kaykobad
Professor
Department of CSE, BUET, Dhaka.
Chairman
(Supervisor)
2. 

Dr. Md. Mostofa Akbar
Professor & Head
Department of CSE, BUET, Dhaka.
Member
(Ex-officio)
3. 

Dr. Md. Abul Kashem Mia
Professor
Department of CSE, BUET, Dhaka.
Member
4. 

Dr. Md. Mostofa Akbar
Professor
Department of CSE, BUET, Dhaka.
Member
5. 

Dr. Atif Hasan Rahman
Assistant Professor
Department of CSE, BUET, Dhaka.
Member
6. 

Dr. Swakkhar Shatabda
Associate Professor
Department of CSE
United International University, Dhaka.
Member
7. 

Dr. Wing-Kin Sung
Professor
School of Computing
National University of Singapore
Computing 1, 13 Computing Drive, Singapore, 117417.
Member
(External)

Candidate's Declaration

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Saifur 26-7-2018

Mohammad Saifur Rahman
Candidate

Acknowledgments

In the name of Allah, the most benevolent and the most merciful.

All praises and thanks are due to Him.

I thank Allah for all His mercy on me during this journey in pursuit of PhD degree. He has bestowed his mercy on me and my family so that I could move forward with my studies and research and my family could bear with me with patience during this time. As the pressure and demand of PhD at times felt unbearable, Allah made it easy right away whenever I was about to get frustrated. While one day the lack of satisfactory results would make me think of ditching a research project, some more experiments, and some prayers, would soon enough lead to something that is worth publishing. It was truly a blessing of Allah.

I must express my deep gratitude to Professor M. Kaykobad, my supervisor. Kaykobad sir has always been an inspirational figure during my graduate and post graduate studies. I do not think I have the caliber to appreciate enough his words of wisdom. Nonetheless, he was kind enough to take me as a PhD student and continue to inspire and guide me through this journey. As I near completion of my thesis, he has already started to advise me to ensure I become a good mentor for future post graduate students. “As soon as you graduate, take one PhD student” has been his recurrent advice lately, so that any unfinished work and future research opportunities that came out of the current work could actively be pursued right away.

I am grateful to the members of my Doctoral Committee, Professor Md. Abul Kashem Mia, Professor Md. Mostofa Akbar, Dr. Atif Hasan Rahman and Dr. Swakkhar Shatabda. In every checkpoint meeting, they have provided me with valuable input. They have also served as internal members of my examination board and provided encouragement and useful feedback

during my oral examination. I thank Professor Wing-Kin Sung for agreeing to be an external examiner and attending the examination through video conferencing. His valuable comments during the oral exam have definitely improved the quality of my thesis.

Several collaborators have had contributions to the research work presented in this thesis. I must acknowledge their contribution at this time. Mr. Khaledur Rahman and Professor M. Sohail Rahman contributed to the sub-Golgi protein type prediction work. Specially, Khaled brought this interesting problem to my attention and conducted some early analysis. Professor Sohail Rahman also encouraged me to work on the DNA-binding protein prediction task. The dataset for the task was provided by Dr. Swakkhar Shatabda in an easily consumable form. Dr. Shatabda also directed me to major works in this field so that I could quickly bring myself up to speed. I also thank Professor Leyi Wei and Professor Quan Zou for providing me with the dataset that was used to retrain our classifier before independent testing. The protective antigen prediction work was inspired through discussions with Dr. Muhammad Sougat Islam and Mr. Arif Khan. I thank them for those early discussions. I thank Dr. Christophe N. Magnan for providing me with the dataset that was used to train and test the protective antigen predictor. I specially thank Mr. Sanjay Saha for his contribution towards implementing web interfaces for all the protein attribute predictors resulting out of this thesis. I hope this will allow researchers all around the world to take advantage of these improved prediction models in their relevant research projects.

The work to estimate gene trees using minimal and relative absent words was partially supported by an INSPIRE Strategic Partnership Award, administered by the British Council, Bangladesh for the project titled “Advances in Algorithms for Next Generation Biological Sequences”. This gave me the opportunity to collaborate with researchers at King’s College London, namely Dr. Ali Alatabbi, Mr. Tanver Athar and Professor Maxime Crochemore. The work to boost QFM using DCM was proposed by Dr. Md. Shamsuzzoha Bayzid, who recently returned to BUET after doing some outstanding research in the University of Texas at Austin in the field of phylogeny. Dr. Bayzid has become an amazing source of knowledge for all the students in CSE Department, BUET who have opted to research in this field. I thank Dr. Bayzid for diligently guiding the students, including myself, through our efforts to reconstruct the *Tree of Life*.

All the experiments in this thesis were conducted using computing facilities provided by the Department of CSE, BUET. I thank the Department for providing me with state-of-the-art facilities. I thank all the supporting staff for solving any logistic problem in a timely fashion. I thank all the teachers of the Department for inquiring me from time to time about my research progress and inspiring me to remain motivated.

It is now time to express my deepest gratitude to my family. I start with my elder brother, Professor Dr. M. Sohel Rahman. Dr. Sohel has not just been a caring and supporting elder brother, he has also provided me with academic guidance actively. Having had years of experience in research in algorithms and bioinformatics, he became a shadow supervisor for me. In addition to my supervisor, he too has reviewed every line of this thesis and provided me with meaningful and actionable feedback. I also thank Dr. Sara Nowreen, my sister in law, and my niece Nazaha for creating a wonderful environment for me to work on my thesis, when I stayed at their home on different occasions.

My late mother had inculcated in me the values of education, application and being sincere in every aspect of life. This has certainly helped me in driving through the hurdles of PhD studies. I acknowledge her enormous contribution in every milestone I have achieved in my academic as well as professional life. I acknowledge my father who has always kept track of my studies. I thank my father in law and mother in law for doing the same. I thank my wife Tasneem for her patience, sacrifices and understanding. She has allowed me time off from my social responsibilities so that I could focus on my research, particularly for the last two years or so. I thank my 2 wonderful sons Abrar and Ahmad for accepting that their father too had to study! As working on my computer for long hours would start aching my back, taking a break to play with my kids or asking them to sit on my back has offered relief. My daughter Aafiah was born in end of May, 2018, with only one month left to submit my thesis to the board of examiners. It was such a joyful experience. Her presence has rejuvenated me and gave me strength to quickly complete my thesis with high quality. I thank Tasneem once again, and her parents, her brother Saad, sister Tazri and sister in law Faiza, her wonderful nephew Mahdi and nieces Maryam and Mustagfira, for looking after Aafiah and shielding me from the hard work that was needed to take care of her in early days. Finally, I thank all the members of my extended family for their support and interest in being a part of my journey towards achieving the PhD.

Abstract

The number of known protein sequences has grown exponentially in recent years, owing to rapid development of sequencing technologies. However, biologists are unable to catch up in finding different attributes of newly discovered protein sequences, as performing lab experiments is tedious and expensive. Computational methods to predict different attributes of proteins are thus being frequently sought. One of the principal tasks of this thesis is to pursue sequence based computational methods for several protein attribute prediction problems. These include Golgi Apparatus (GA) resident protein type prediction, DNA-binding protein (DNA-BP) prediction and protective antigen prediction. Through solving these problems using a sequence based methodology, our research empirically asserts the natural belief that a protein's functional and structural information are intrinsically encoded within its primary sequence.

Given a GA protein, an important research question is whether it is a *cis*-Golgi protein or a *trans*-Golgi protein. This is because correct classification of GA proteins can lead to drug development against various congenital, neurodegenerative and inherited diseases. We propose a sequence based prediction model for sub-Golgi protein types. A DNA-BP binds to a DNA to regulate and affect various cellular processes. As such, DNA-BPs can potentially be used for drug development in treating genetic diseases and cancers. We develop a DNA-BP predictor, that extracts meaningful information directly from the protein sequences, without any dependence on functional domain or structural information. Recursive Feature Elimination (RFE) is then applied to optimize the number of features used in the prediction process. Another important protein attribute prediction problem that we tackle is whether a given pathogenic protein has the ability of invoking adaptive immune response to subsequent exposure to the specific pathogen or related organisms. Such proteins are called protective antigens and are of immense importance in vaccine preparation and drug design. We propose a protective antigen predictor that, again, solely exploits sequence based features to provide a pathogen independent prediction model.

Our predictor can be used to quickly sift through any pathogen proteome and predict a list of potential protective antigens.

Through the exercise of building these three predictors, we formulate a general framework for feature extraction and selection that can be applied to any protein attribute prediction problem. One of the distinct characteristics of this framework is to exploit only the proteins' primary sequence based features, leaving out any structural, evolutionary or functional features, thereby making the whole framework lightweight. The framework involves counting small substrings, with or without gaps, in a protein sequence, to represent the protein in a discrete model, followed by a novel approach of feature selection.

Another focus of this thesis is phylogeny, which is the study of the evolutionary relationships among different species, genes or proteins (taxa). When gene copies are sampled from various species, the gene tree relating these copies might disagree with the species phylogeny. This discord can arise from horizontal gene transfer, incomplete lineage sorting (ILS), and gene duplication and extinction. Summary methods of species tree estimation work by first estimating the individual gene trees from respective gene sequence alignments, and then summarizing these gene trees to reconstruct the species phylogeny. To speed up the step of gene tree estimation, we propose a set of distance measures between two biological sequences utilizing the concepts of minimal and relative absent words. The computation of these distance measures is done in an alignment-free manner. We demonstrate the use of these techniques experimentally and show how the pairwise distance matrix thus produced can be used to reconstruct the gene phylogeny.

When the gene tree discordance is modeled by ILS, coalescent-based methods need to be applied to accurately estimate the species tree. One such method is Quartet FM (QFM), which is highly accurate but does not scale to large numbers of taxa. We propose boosting the scalability and performance of QFM through the application of disk covering methods (DCMs). Extensive experimentation on large simulated datasets demonstrates superiority of our method over ASTRAL, a widely used and highly accurate coalescent-based species tree estimation method that is statistically consistent under the multi-species coalescent model.

Overall, this thesis offers a generic framework for tackling protein attribute prediction problems using information solely from the protein sequence and attempts to scale existing phylogeny estimation methods to larger datasets.

Contents

Acknowledgments	vii
Abstract	xi
List of Tables	xix
List of Figures	xxiii
1 Introduction	1
1.1 Research Focus 1: Protein Attribute Prediction	2
1.2 Research Focus 2: Phylogeny Reconstruction	5
1.3 Our Contribution	7
1.4 Organization of the Thesis	10
1.5 Conclusion	11
2 Preliminaries	13
2.1 Protein Attribute Prediction	14
2.1.1 Protein Attribute Prediction Pipeline	16
2.1.2 Machine Learning	18
2.1.3 Classification and Regression	19

2.1.4	Support Vector Machine	19
2.1.5	Random Forests	20
2.1.6	Feature Selection	21
2.1.7	Chou’s General PseAAC	21
2.1.8	Testing a Predictor	24
2.1.9	Predictor Performance Metrics	25
2.2	Phylogeny Reconstruction	28
2.2.1	Rooted and Unrooted Trees	29
2.2.2	Binary and Non-binary Trees	30
2.2.3	Clade and Bipartition	30
2.2.4	Branch Length	31
2.2.5	Gene Tree-Species Tree Discordance	31
2.2.6	Gene Tree Parsimony	35
2.2.7	Statistical Consistency	35
2.2.8	Species Tree Estimation Methods	36
2.2.9	Evaluation of Species Tree Estimation Methods	36
2.3	Conclusion	38

I Protein Attribute Prediction 39

3 sub-Golgi Protein Type Prediction 41

3.1	Introduction	42
3.2	Material and Methods	46
3.2.1	Benchmark Dataset	46

3.2.2	Protein Sample Representation	47
3.2.3	Prediction Algorithm	50
3.2.4	Predictor Evaluation	55
3.2.5	Predictor Availability	56
3.3	Results	56
3.3.1	Impact of Feature Extraction Techniques	57
3.3.2	Impact of Data Imbalance in isGPT Learning Model	58
3.3.3	Comparison between isGPT and Existing Techniques	59
3.4	Discussion	60
3.4.1	Consistency Check of Earlier Results	61
3.4.2	Choice of Class Discriminating Threshold in isGPT	62
3.4.3	Large Feature Space Independent of PSSM	64
3.5	Conclusion	64
4	DNA-binding Protein Prediction	67
4.1	Introduction	68
4.2	Material and Methods	74
4.2.1	Benchmark Dataset	74
4.2.2	Protein Sample Representation	75
4.2.3	Prediction Algorithm	77
4.2.4	Predictor Evaluation	79
4.2.5	Predictor Availability	80
4.3	Results	80
4.3.1	Impact of Number of Features	81

4.3.2	Impact of Feature Extraction Techniques	83
4.3.3	Discriminant Visualization	85
4.3.4	Comparison between DPP-PseAAC and Existing Techniques	87
4.4	Discussion	90
4.4.1	Differentiation between DPP-PseAAC and Existing Predictors . . .	90
4.4.2	Some Errors in Results of Earlier Predictors	91
4.4.3	Unavailability of BLASTCLUST in the Latest Version of Stand- alone BLAST	92
4.4.4	Jackknife Cross-validation vs. Independent Testing	92
4.5	Conclusion	93
5	Protective Antigen Prediction	97
5.1	Introduction	98
5.2	Material and Methods	104
5.2.1	Benchmark Dataset	104
5.2.2	Protein Sample Representation	106
5.2.3	Prediction Algorithm	107
5.2.4	Predictor Evaluation	110
5.2.5	Predictor Availability	111
5.3	Results	111
5.3.1	Impact of Number of Features	112
5.3.2	Impact of Feature Extraction Techniques	115
5.3.3	Feature Importance Visualization	116
5.3.4	10-fold Cross-validation Results	117
5.3.5	Leave One Protein Set Out Cross-validation Results	119

5.3.6	Jackknife Cross-validation Results	120
5.3.7	Independent Test Results	121
5.4	Discussion	124
5.5	Conclusion	126
 II Phylogeny Reconstruction		129
 6 Gene Tree Estimation Using Absent Words		131
6.1	Introduction	132
6.2	Methods	135
6.2.1	Minimal Absent Word (MAW)	135
6.2.2	Distance Measures	136
6.2.3	Relative Absent Word (RAW)	138
6.2.4	Gene Tree Estimation Algorithms	139
6.3	Experiments	139
6.4	Results	141
6.4.1	Estimated β -globin Gene Trees	144
6.5	Conclusion	147
 7 Species Tree Estimation using DCM Boosted QFM		149
7.1	Introduction	150
7.2	Methods	151
7.3	Experiments	152
7.3.1	Mammalian Simulated Datasets	153
7.3.2	Species Tree Estimation Tools	154

7.4	Results	155
7.4.1	Running Time	155
7.4.2	Topological Accuracy	156
7.5	Conclusion	161
8	Conclusion	163
8.1	Protein Attribute Prediction	163
8.2	Phylogeny Reconstruction	167
8.3	Future Research Directions	168
	Appendices	173
A	Supplementary Materials for Gene Tree Estimation Using Absent Words	173
A.1	Distance Matrices	174
A.2	Estimated β -globin Gene Trees	182
B	Supplementary Materials for Species Tree Estimation Using DCM Boosted QFM	199
B.1	Species Trees on the 37 Taxa Mammalian Dataset	200
	Publications	245

List of Tables

2.1	List of 20 standard amino acids along with their one letter codes.	15
3.1	Area under ROC and PR curves for different number of top-ranked features selected.	54
3.2	Comparison of classification and regression models of isGPT.	58
3.3	Optimal parameters for classification and regression models of isGPT. . . .	59
3.4	Comparison of isGPT regression model with previous methods.	60
3.5	Comparison of of different steps in model building in <i>isGPT</i> vs. prior art.	61
4.1	Comparison of DPP-PseAAC with previous methods using jackknife cross-validation on the PDB1075 dataset.	88
4.2	Comparison of DPP-PseAAC with previous methods using independent test.	89
4.3	Structure based predictors at a glance.	90
4.4	Sequence based predictors at a glance.	95
5.1	Size and composition of the six protein sets used as the training set.	104
5.2	Comparison of Antigenic with VaxiJen and ANTIGENpro based on 10-fold cross-validation.	118
5.3	Comparison of accuracy between Antigenic and ANTIGENpro based on leave one protein set out cross-validation.	119

5.4	Jackknife cross-validation performance of Antigenic* and Antigenic.	121
5.5	Comparison of Antigenic with VaxiJen and ANTIGENpro based on independent testing.	121
5.6	Area under ROC and PR curves for different predictors on the Bartonella dataset.	123
5.7	Enrichment among top ranked proteins of Bartonella dataset, ranked by different predictors.	124
6.1	Functions used and compared in this chapter as distance measures.	135
6.2	The distance matrix based on the Length Weighted Index on RAW sets (on RC setting).	142
6.3	The sorted list of each species from a particular species (left most column of each row) according to the computed distance based on the Length Weighted Index on RAW sets (on RC setting).	143
6.4	The distance matrix based on the Jaccard distance on MAW sets (on RC setting).	143
6.5	The sorted list of each species from a particular species (left most column of each row) according to the computed distance based on the Jaccard distance on MAW sets (on RC setting).	144
7.1	Average running time (minutes) of boosted and native QFM for various model conditions.	156
7.2	Average FN rate of different species trees for various model conditions.	157
A.1	The distance matrix based on the Length Weighted Index on Symmetric Difference of MAW sets (on RC setting).	174
A.2	The distance matrix based on the Length Weighted Index on Intersection of MAW sets (on RC setting).	174

A.3	The distance matrix based on the Length Weighted Index on RAW sets (on RC setting).	175
A.4	The distance matrix based on the GCC Index on Symmetric Difference of MAW sets (on RC setting).	175
A.5	The distance matrix based on the GCC Index on Intersection of MAW sets (on RC setting).	176
A.6	The distance matrix based on the GCC Index on RAW sets (on RC setting).	176
A.7	The distance matrix based on the Jaccard Distance of MAW sets (on RC setting).	177
A.8	The distance matrix based on the Total Variation Distance of MAW sets (on RC setting).	177
A.9	The distance matrix based on the Length Weighted Index on Symmetric Difference of MAW sets (on NoRC setting).	178
A.10	The distance matrix based on the Length Weighted Index on intersection of MAW sets (on NoRC setting).	178
A.11	The distance matrix based on the Length Weighted Index on RAW sets (on NoRC setting).	179
A.12	The distance matrix based on the GCC Index on Symmetric Index of MAW sets (on NoRC setting).	179
A.13	The distance matrix based on the GCC Index on intersection of MAW sets (on NoRC setting).	180
A.14	The distance matrix based on the GCC Index on RAW sets (on NoRC setting).	180
A.15	The distance matrix based on the Jaccard Distance of MAW sets (on NoRC setting).	181

A.16 The distance matrix based on the Total Variation Distance of MAW sets
(on NoRC setting). 181

List of Figures

2.1	Four categories of protein structural class.	16
2.2	A phylogenetic tree relating four species: human, chimpanzee, gorilla and orangutan.	28
2.3	Rooted and unrooted trees.	29
2.4	Phylogenetic tree on a set of mammalian species, with fly as the outgroup.	29
2.5	Binary and non-binary phylogenetic trees of 6 taxa $\{A, B, C, D, E, F\}$	30
2.6	Gene tree-species tree discordance or incongruence	32
2.7	Example of gene tree-species tree discordance due to incomplete lineage sorting.	34
3.1	The Golgi apparatus and its synthesis process.	43
3.2	Amino Acid Composition (AAC), on average, for the different sub-Golgi protein classes in the training dataset.	47
3.3	isGPT model construction.	51
3.4	Categorized feature importance based on <i>mean decrease in accuracy</i>	53
3.5	ROC-Curves and PR-Curves.	54
3.6	Accuracy and MCC of different feature extraction techniques.	57
3.7	Response of different performance metrics against variation of class discriminating threshold.	63

4.1	DNA-binding proteins bound to respective target DNAs.	69
4.2	Steps in feature selection.	77
4.3	Categorized feature importance based on random forests model based ranking	77
4.4	ROC-Curves of prediction models	81
4.5	Area under ROC curve (auROC), accuracy, sensitivity, specificity and MCC of models with varying number of features.	82
4.6	Performance of different feature extraction techniques.	84
4.7	The discriminative weights of top 25 features.	86
5.1	Steps in feature selection.	107
5.2	Categorized feature importance based on random forests model based rank- ing.	108
5.3	ROC and PR curves of prediction models.	112
5.4	Area under ROC-curve (auROC), Area under PR-curve (auPR), Accuracy (Acc), Sensitivity (Sn), and Specificity (Sp) of models with varying number of features.	114
5.5	Performance of different feature extraction techniques.	115
5.6	The importance score of top 25 features.	117
5.7	ROC and PR curves on the independent test for different prediction tools.	123
6.1	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the RAW sets (on RC setting).	145
6.2	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on Symmetric Difference of the MAW sets (on RC setting).	145

6.3	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Jaccard distance on the MAW sets (on RC setting).	146
6.4	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix of [183].	146
7.1	The model species tree for the 37-taxon mammalian dataset of [252].	153
7.2	FN rates of MRP, ASTRAL and boosted versions of QFM on the simulated mammalian datasets with varying sequence length.	158
7.3	FN rates of MRP, ASTRAL and boosted versions of QFM on the simulated mammalian datasets with varying amounts of ILS.	159
7.4	FN rates of MRP, ASTRAL and boosted versions of QFM on the simulated mammalian datasets for different number of gene trees.	160
7.5	Species tree generated by DCM boosted QFM on the simulated dataset with the 37 taxa.	161
A.1	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Length Weighted Index on the Symmetric Difference of the MAW sets (on RC setting).	182
A.2	The β -globin gene tree for the 11 species computed using Neighboring Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the Symmetric Difference of the MAW sets (on RC setting).	182
A.3	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Length Weighted Index on the Intersections of the MAW sets (on RC setting).	183

A.4	The β -globin gene tree for the 11 species computed using Neighboring Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the Intersection of the MAW sets (on RC setting). . . .	183
A.5	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Length Weighted Index on the RAW sets (on RC setting).	184
A.6	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the RAW sets (on RC setting).	184
A.7	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the GC content on the Symmetric Difference of the MAW sets (on RC setting).	185
A.8	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the GC content on the Symmetric Difference of the MAW sets (on RC setting). . .	185
A.9	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the GC content on the Intersection of the MAW sets (on RC setting).	186
A.10	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the GC content on the Intersection of the MAW sets (on RC setting).	186
A.11	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the GC content on the RAW sets (on RC setting).	187
A.12	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the GC content on the RAW sets (on RC setting).	187

A.13	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Jaccard Distance of the MAW sets (on RC setting).	188
A.14	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Jaccard Distance of the MAW sets (on RC setting).	188
A.15	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Total Variation Distance of the MAW sets (on RC setting).	189
A.16	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Total Variation Distance of the MAW sets (on RC setting).	189
A.17	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Length Weighted Index on the Symmetric Difference of the MAW sets (on NoRC setting).	190
A.18	The β -globin gene tree for the 11 species computed using Neighboring Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the Symmetric Difference of the MAW sets (on NoRC setting).	190
A.19	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Length Weighted Index on the Intersections of the MAW sets (on NoRC setting).	191
A.20	The β -globin gene tree for the 11 species computed using Neighboring Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the Intersection of the MAW sets (on NoRC setting).	191

A.21	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Length Weighted Index on the RAW sets (on NoRC setting).	192
A.22	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the RAW sets (on NoRC setting).	192
A.23	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the GC content on the Symmetric Difference of the MAW sets (on NoRC setting).	193
A.24	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the GC content on the Symmetric Difference of the MAW sets (on NoRC setting).	193
A.25	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the GC content on the Intersection of the MAW sets (on NoRC setting).	194
A.26	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the GC content on the Intersection of the MAW sets (on NoRC setting).	194
A.27	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the GC content on the RAW sets (on NoRC setting).	195
A.28	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the GC content on the RAW sets (on NoRC setting).	195
A.29	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Jaccard Distance of the MAW sets (on NoRC setting).	196

A.30	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Jaccard Distance of the MAW sets (on NoRC setting).	196
A.31	The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Total Variation Distance of the MAW sets (on NoRC setting).	197
A.32	The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Total Variation Distance of the MAW sets (on NoRC setting).	197
B.1	The model species tree for the 37-taxon mammalian dataset.	200
B.2	Species tree generated by DCM boosted QFM for the model condition of 0.2X level of ILS, 200 genes of 500 bp each.	200
B.3	Species tree generated by DCM boosted QFM for the model condition of 0.5X level of ILS, 200 genes of 500 bp each.	201
B.4	Species tree generated by DCM boosted QFM for the model condition of 1X level of ILS, 200 genes of 500 bp each.	201
B.5	Species tree generated by DCM boosted QFM for the model condition of 2X level of ILS, 200 genes of 500 bp each.	202
B.6	Species tree generated by DCM boosted QFM for the model condition of 1X level of ILS, 50 genes of 500 bp each.	202
B.7	Species tree generated by DCM boosted QFM for the model condition of 1X level of ILS, 100 genes of 500 bp each.	203
B.8	Species tree generated by DCM boosted QFM for the model condition of 1X level of ILS, 400 genes of 500 bp each.	203
B.9	Species tree generated by DCM boosted QFM for the model condition of 1X level of ILS, 800 genes of 500 bp each.	204

B.10 Species tree generated by DCM boosted QFM for the model condition of 1X level of ILS, 200 genes of 250 bp each.	204
B.11 Species tree generated by DCM boosted QFM for the model condition of 1X level of ILS, 200 genes of 1000 bp each.	205
B.12 Species tree generated by DCM boosted QFM for the model condition of 1X level of ILS, 200 true genes.	205

Chapter 1

Introduction

A protein is a macromolecule consisting of one or more long chains of amino acid residues. Due to the rapid development of sequencing technologies, the number of sequence-known proteins has grown exponentially in recent years. However, as the biochemical experiments to learn the attributes of proteins are expensive and time consuming, a large gap exists between the number of sequence-known proteins and that of attribute-known proteins. To catch up, researchers have started to rely on computational methods to predict different attributes of proteins. These attributes include, but are not limited to, protein structural class, folding rate, cleavage site, antigenicity, subcellular location and so on [61]. This has given rise to the prominent new field of research of protein attribute prediction. One of the principal tasks of this thesis is to pursue sequence based computational methods for several protein attribute prediction tasks. These include sub-Golgi protein type prediction, DNA-binding protein (DNA-BP) prediction and protective antigen prediction. Sequence based phylogeny reconstruction is another focus of this thesis. An alignment-free method for gene tree estimation and a coalescent-based method for species tree estimation is proposed in this regard.

1.1 Research Focus 1: Protein Attribute Prediction

In this thesis, we have focused on three protein attribute prediction problems. Solving these prediction problems has the potential of new drugs or vaccine discovery that can alleviate, or perhaps even eradicate, several genetic and pathogen borne diseases. These include sub-Golgi protein type prediction, DNA-BP prediction and protective antigen prediction. The Golgi Apparatus (GA) is a key organelle within the eukaryotic cell that modifies and sorts proteins for transport throughout the cell. It comprises two types of proteins, namely, *cis*-Golgi proteins and *trans*-Golgi proteins. Any dysfunction of GA proteins can result in congenital glycosylation disorders, diabetes, cancer and cystic fibrosis. The exact classification of GA proteins may contribute to drug development against these diseases. A DNA-binding protein (DNA-BP), on the other hand, binds to a DNA to regulate and effect various cellular processes. As such, these proteins can potentially be used for drug development in treating genetic diseases and cancers. The prediction task here is to detect whether a protein of interest would bind to a DNA or not. Finally, a protective antigen is a pathogenic protein that has the ability of invoking adaptive immune response to subsequent exposure to the specific pathogen or related organisms. Such proteins are of immense importance in vaccine preparation and drug design. The related prediction problem therefore is to answer whether a protein from a pathogen can invoke protective immune response.

When a new protein sequence is discovered, one approach to predicting its attributes would be to align its amino acid sequence, also known as the *primary sequence*, against a database of protein sequences with known attributes. Sequence homology is expected to infer functional homology. Thus, sequence searching programs such as BLAST [14], FASTA [220], PSORT [204] etc. can be applied to identify similar sequences and infer attributes of a new sequence accordingly. However, such an approach fails when the target protein lacks any sequence similarity with the database of attribute-known proteins. Therefore, other approaches such as empirical statistical methods and machine learning (ML) look promising and more useful in this endeavor. In this thesis, we applied machine learning based approaches.

Any effort to establish a new protein attribute predictor generally takes a 5-step route [61]. In the rest of this section, we describe these 5 steps along with a brief review on the relevant literature as we proceed with the discussion. Firstly, a stringent benchmark dataset should be prepared to train and test the predictor. To avoid homology bias, the dataset should contain proteins with pairwise sequence similarity of no more than a certain threshold (e.g., 25%) [61]. Secondly, a set of relevant features should be extracted from the protein’s primary sequence and/or structure. The features should be informative enough for predicting the relevant attribute. Many different features have been used in literature to represent proteins. Some of these are based on the structures of the proteins, while some features are extracted directly from their the primary sequences.

The sub-Golgi protein type predictor proposed in [269] uses structural features in addition to sequence based features. Several structural features have been utilized in literature of DNA-BP prediction as well. These include electrostatic patches and surface clefts [256], dipole moment [261], statistical potential energy [106], predicted secondary structure (PredSS) [154, 191], predicted relative solvent accessibility (PredRSA) [191], secondary structure composition and occurrence, torsional angles bigram and auto-covariance, structural probabilities bigram and auto-covariance, accessible surface area composition etc. [65]. Several predicted structural features have also been used in protective antigen prediction, including predicted α and β residues, exposed residues fraction, number of domains, number of transmembrane helices (TMHs) etc. [193]. Plenty of software packages and web services exist that can extract structural features of a protein, provided that the structural composition of the protein is known. These include PSIPRED [194], SPINE-X [94], SPIDER2 [289], SSpro [49], DOMpro [50], ACCPro [49], TMHMM [151] etc.

However, structure-based predictors are applicable only when the structural information of a candidate protein is known. The predictors that solely rely on structural information of proteins are thus limited in their use. Sequence based methods, on the other hand, extract various discriminating features from the amino acid sequence. Examples of such features are amino acid composition (AAC), Dipeptides (Dip), n -Gapped-Dipeptides (nGDip), n -grams, Pseudo amino acid composition (PseAAC) [59], amino acid physico-

chemical properties and other indices from the AAIndex database [146], absolute charge per residue, molecular weight, GRAVY Index [156], Aliphatic index [134] etc. The sub-Golgi protein type predictors proposed in [10,79,80,140,141,288] use only sequence based features. Examples of prominent sequence based predictors of DNA-BPs can be found in [65,82,93,135,154,155,168,174,176,191,203,206,219,245,251,275,277,285,286,295,297]. VaxiJen [7], the most widely used protective antigen predictor, also uses sequence based features alone.

One class of sequence based features that has recently become very popular is the Position Specific Scoring Matrix (PSSM) based features. The PSSM can be computed from PSI-BLAST [15] by searching the non-redundant protein database using at least three iterations. PSSM based feature extraction has been used in several predictors of the protein attributes that we focus on [10,65,155,174,275,277,288]. One drawback of these predictors, however, is that the construction of PSSM is time consuming. Also, if the target protein does not have enough known homologous sequences, the generated PSSM cannot describe the protein adequately. Any prediction model depending on PSSM information will produce wrong predictions in such a case [164]. Our proposed predictors avoid structural and PSSM based features. We have only utilized features extracted directly from the amino acid sequence of the protein.

Coming back to the discussion of the 5-step route of establishing a predictor, the third step is to develop a suitable prediction algorithm. Among the ML algorithms, widely used in the field of protein attribute prediction are Artificial Neural Network (ANN) [111], Support Vector Machine (SVM) [32], Random forests [34] etc. In the protein attribute prediction problems of our interest, ANN has been applied in [11,256], random forests algorithm has been applied in [154,168,191,277,288]. Majority of the reviewed predictors used SVM [65,79,82,140,141,155,174,176,177,269,275]. All the protein attribute predictors proposed in this thesis have used random forests algorithm for ranking the features. SVM has been applied for sub-Golgi protein type as well as DNA-BP prediction. For protective antigen prediction, random forests algorithm was applied to train the predictor.

To conclude the brief discussion on the 5-step route, the fourth step in the pipeline is predictor evaluation, while the final step involves making the predictor publicly and widely available. We have evaluated our predictors thoroughly using several well-established testing methods. We have also made our predictors available through web interfaces for wide adoption.

Notably, there are several other techniques especially from the statistical domain that have been used in establishing such predictors in the literature. These tools are excluded from the discussion as our focus is on ML based algorithms and also because the latter have been shown to have outperformed the former in general.

1.2 Research Focus 2: Phylogeny Reconstruction

Another focus of our research is phylogeny, which is the study of the evolutionary relationships among different species, genes or proteins (taxa). The ultimate goal of this research field is to infer the *Tree of Life*, the phylogeny of all living organisms on earth. Research in phylogeny reconstruction has practical impact in other fields of biology, such as epidemiology, conservation biology, pharmaceutical research, protein structure prediction and so on [276].

The evolutionary history among a set of species, via the process of speciation, is represented by a species tree. On the contrary, a gene tree represents the evolution of a particular “gene” within a group of species. When species are split by speciation, the gene copies within species are also split into separate lineages of descent. However, when gene copies are sampled from various species, the gene tree relating these copies might disagree with the species phylogeny. This discord can arise from horizontal gene transfer, incomplete lineage sorting (ILS), and gene duplication and extinction [192]. This needs to be taken into consideration when attempt is made to recover the species tree from multiple gene sequences.

Species tree estimation from multiple genes is often performed using concatenation. In this approach, alignments are first estimated for each gene. Then these alignments for all the genes are concatenated into a supermatrix, which is then used to estimate the species tree. This process can accurately estimate the species tree only if gene trees have identical topologies. Unfortunately, this approach can confidently reach incorrect conclusions if gene trees differ from the species tree (and hence from each other) [74, 124, 158, 159, 181]. An alternate to this approach is to first estimate the individual gene trees from respective gene sequence alignments, and then to summarize these gene trees with a goal to reconstruct the species phylogeny. Such methods, known as summary methods, are now becoming more popular [124, 180, 181, 199, 200, 209, 236, 292].

In sequence based methods of gene tree estimation, a set of homologous sequences from different species are provided as the input. After obtaining an alignment of these sequences, different methods are applied to extract the phylogenetic information. One such class of methods, known as the *distance-based methods*, computes a distance matrix from the alignment that gives the pairwise distances among the sequences under consideration. This matrix is then used to estimate the gene tree using standard clustering methods or specially tailored methods [78, 91, 109, 128, 243, 248]. Another approach uses heuristics for either Maximum-Likelihood (ML) [96] or Maximum-Parsimony (MP) [101] both of which are NP hard optimization problems. The most popular tools of gene tree estimation are based on heuristics for Maximum-Likelihood [113, 226, 227, 254, 255]. Yet another approach, namely Bayesian Markov Chain Monte Carlo (MCMC), produces not just a single gene tree but a probability distribution of the trees or aspects of the evolutionary history [33, 124, 158]. All these methods rely on sequence alignment, which is a time consuming task. Additionally, if any error is introduced in the alignment process, the downstream processes get impacted, resulting in poor estimation of the gene tree. To mitigate this problem, we propose novel distance measures of biological sequences that are light-weight and alignment free.

When the gene tree discordance is modeled by ILS or deep coalescence, coalescent-based methods need to be applied to estimate the species tree. These methods provide

statistical guarantees of returning the true tree with a high probability, as the number of genes in the study increases. One such method is Quartet FM (QFM) [236], which is highly accurate but does not scale to large numbers of taxa. We propose boosting the scalability and performance of QFM through the application of disk-covering methods (DCMs) [132, 133, 205, 239].

1.3 Our Contribution

In Part I of this thesis, which focuses on protein attribute prediction problems, we have made the following contributions:

- We have created a sub-Golgi protein type predictor that can distinguish between *cis*-Golgi and *trans*-Golgi proteins. In our proposed classifier, we have extracted features solely from the protein sequence. We have then employed random forests algorithm for feature ranking and Support Vector Machine (SVM) to learn the classification model. As the benchmark dataset is significantly imbalanced, we have applied Synthetic Minority Over-sampling Technique (SMOTE) [43] to the dataset to make it balanced. Our method, *identification of sub-Golgi Protein Types (isGPT)*, achieves accuracy values of 95.4%, 95.9% and 95.3% for 10-fold cross-validation test, jackknife test and independent test respectively. According to different performance metrics, isGPT outperforms all the state-of-the-art techniques.
- We have developed a predictor that can determine whether a protein can bind to a DNA or not. Our predictor extracts meaningful information directly from the protein sequences, without any dependence on functional domain or structural information. After feature extraction, we have employed random forests algorithm to rank the features. Afterwards, we have used Recursive Feature Elimination (RFE) method to extract an optimal set of features and trained a prediction model using SVM with linear kernel. Our proposed method, named as *DNA-binding Protein Prediction model using Chou's general PseAAC (DPP-PseAAC)*, demonstrates supe-

rior performance compared to the state-of-the-art predictors on standard benchmark dataset. DPP-PseAAC achieves accuracy values of 93.21%, 95.91% and 77.42% for 10-fold cross-validation test, jackknife test and independent test respectively.

- We have implemented a protective antigen predictor that has a pathogen independent model which extracts class-discriminant information from the protein sequence alone. Thus, unlike state-of-the-art predictors, it can be used to quickly sift through any pathogen proteome and predict a list of potential protective antigens. Named *Antigenic*, our protective antigen predictor achieves accuracy, sensitivity and specificity values of 78.04%, 78.99% and 77.08% in 10-fold cross-validation testing respectively on the benchmark dataset. In jackknife cross-validation, the corresponding scores are 80.03%, 80.90% and 79.16% respectively.
- We have developed a bundle of web interfaces for the above three protein attribute predictors. isGPT is available at <http://isgpt.research.buet.ac.bd/>, DPP-PseAAC at <http://dpp-pseaac.research.buet.ac.bd/> and Antigenic at <http://antigenic.research.buet.ac.bd/>. These user friendly, publicly accessible interfaces are expected to encourage researchers to apply these prediction models in relevant research projects and thus be of interest to researchers and practitioners alike in the relevant fields.
- Finally, through the design and development of these three predictors, we have established a general framework for feature extraction and selection that can be applied to any protein attribute prediction problem. It involves counting small substrings, with or without gaps, in the protein sequence, to represent the protein in a discrete model, followed by a novel approach of feature selection. The framework is a learning from our efforts in solving the 3 protein attribute prediction problems as discussed above. A distinct and note-worthy property of this framework is essentially the focus on only the primary sequence. This is directly in contrast to the ongoing recent efforts that popularly utilize features from structural and functional domains as well as to the exploitation of time consuming and database dependent

evolutionary information like PSSM. While it seems appealing to use the structural and functional information of protein as features, our results promise the potential of only focusing on the primary sequence, which is light-weight, less time consuming, and can implicitly infer the structural information.

In Part II of this thesis, which focuses on the problem of phylogeny reconstruction, we have made the following contributions:

- We have proposed a set of distance measures between two biological sequences utilizing the concepts of minimal and relative absent words. The computation of these distance measures is done in an alignment-free manner. We demonstrate the use of these techniques on a set of 11 nucleotide sequences. We also provide recommendations to use the best distance measure based on our analysis. We have also implemented a related web-based tool with limited capacity here: <http://77.68.43.135/AWorDS/>.
- We have designed a gene tree estimation method based on the above distance measures. For a collection of gene sequences, we demonstrate how the pairwise distance matrix produced by any of these distance measures can be used to reconstruct the gene phylogeny. All the widely used gene tree estimation methods rely on sequence alignment, which is a time consuming task. Also, any error in the alignment significantly affects the downstream processes, resulting in poor estimation of the gene tree. As such, we make an effort towards a fast and alignment free solution to gene tree reconstruction.
- Finally, we have presented a species tree estimation method applying DCM to boost a celebrated method called QFM. Quartet FM (QFM) is a highly accurate species tree estimation method for a very small number of taxa. We apply disk-covering methods (DCMs) to boost the scalability and performance of QFM. Experiments with a simulated dataset of 37 taxa shows that DCM boosted QFM outperforms AS-TRAL [199,200,292], a highly accurate and popular species tree estimation method that is statistically consistent under the multi-species coalescent model.

1.4 Organization of the Thesis

The rest of the chapters are organized as follows.

In Chapter 2 we introduce preliminary concepts and terminologies that are used in describing the contribution of this research in subsequent chapters. We introduce the problem of protein attribute prediction and discuss the different steps in the process. We explain different methodology and metrics to objectively assess the performance of any protein attribute predictor. Since phylogeny is another focus of this thesis, we therefore discuss basic concepts of gene tree, species tree and reasons of discordance between the two.

We then enter the Part I of this thesis, comprising Chapters 3, 4 and 5, focusing on several protein attribute prediction problems. In Chapter 3, we introduce a sequence based computational model for classification of GA proteins. Given a GA protein sequence as input, the classifier can determine whether it is a *cis*-Golgi protein or a *trans*-Golgi protein.

In Chapter 4, we build a predictor for DNA-binding proteins (DNA-BPs). Our prediction model for DNA-BPs extracts meaningful information directly from the protein sequences, without any dependence on functional domain or structural information.

In Chapter 5, we propose a new protective antigen predictor. A reliable protective antigen predictor plays a vital role in vaccine discovery in the *Reverse Vaccinology* [224, 233] pipeline. Like in previous chapters, here too, we have worked to build a predictor that extracts meaningful information from the protein sequence alone.

We then move to the Part II of this thesis, where we shift our focus onto phylogeny reconstruction. This latter part of the thesis consists of Chapters 6 and 7. In Chapter 6 we explore the idea of using minimal and relative absent words to compute the distance between two biological sequences. A minimal absent word (MAW) is a word that is absent in a sequence but all its proper factors occur in that sequence. On the other hand, a relative absent word (RAW) is a word that occurs in a target sequence but is absent in

a reference sequence. A RAW is minimal if none of its proper factors are RAW for the same pair of target and reference sequences. For a pair of biological sequences, we propose several distance measures using MAW and RAW sets. We provide recommendations to use the best distance measure based on our analysis. For a collection of gene sequences, we demonstrate how the pairwise distance matrix thus produced can be used to reconstruct the gene phylogeny in an alignment-free manner.

In Chapter 7, we focus on species tree estimation, particularly when the gene tree discordance is modeled by incomplete lineage sorting (ILS) or deep coalescence. When the genes evolve down different tree topologies due to ILS, coalescent-based methods need to be applied to estimate the species tree. One such method is Quartet FM (QFM), which is highly accurate but does not scale to large numbers of taxa. We apply disk-covering methods (DCMs) to boost the scalability and performance of QFM.

Finally, we offer concluding remarks and direction for further research in Chapter 8.

1.5 Conclusion

In this introductory chapter we have introduced some background and motivation of the tasks we take up in this thesis. We have also clearly outlined the contribution of this research work. In the next chapter, *Preliminaries*, we will introduce technical background of various methods and concepts that are needed to comprehend and appreciate the research conducted in this thesis.

Chapter 2

Preliminaries

In this chapter, we introduce some preliminary concepts and terminologies that will be used in describing the contribution of this research in subsequent chapters. We begin this chapter by a discussion on proteins, what they are made of, what their structures and functions are. While the sequence known proteins abound, scientists are finding it difficult to predict their functions and attributes solely through biochemical experiments. Hence the notion of protein attribute prediction has emerged. We discuss the different steps in this process. Since this thesis solves several protein attribute prediction problems by machine learning, we therefore give a brief idea of what machine learning is. The different learning algorithms we have used are also discussed briefly. We then explain different methodology and metrics to objectively assess the performance of any protein attribute predictor.

We subsequently introduce phylogeny, which is another focus of this thesis. We discuss different aspects of a phylogenetic tree, followed by the concepts of gene tree and species tree, gene tree discordance, and species tree reconstruction from the genomic data. Any method for phylogeny reconstruction must be objectively assessed. Therefore we also discuss the measures of accuracy to evaluate species tree reconstruction methods.

2.1 Protein Attribute Prediction

A protein is a macromolecule consisting of one or more long chains of amino acid residues. The individual amino acid residues are bonded together by peptide bonds and adjacent amino acid residues. A protein is encoded by a gene. The sequence of amino acid residues in a protein is thus governed by the nucleotide sequence of the corresponding gene. Protein is an important building block of enzymes, hormones, other body chemicals, bones, muscles, cartilages, skin, blood, hair, nails etc. It is an important component in every cell of the body. As such, it serves a wide variety of functions - tissue building and repair, catalyzing metabolic reactions, DNA replication and recombination, responding to stimuli, transporting molecules from one location to another and so on.

Proteins comprise long chains of amino acid residues. Amino acids on the other hand consist of organic compounds containing amine ($-NH_2$) and carboxyl ($-COOH$) functional groups, along with a side chain (R group) specific to each amino acid. The key elements of an amino acid are carbon (C), hydrogen (H), oxygen (O), and nitrogen (N). While naturally occurring amino acids are around 500 in number, only 20 of these amino acids are encoded directly by triplet codons in the genetic code and are known as “standard” amino acids. Each of these amino acids is given a one letter code. This way a protein can be represented as a long sequence of letters, drawn from a 20 letter alphabet. The standard amino acids, together with their single letter encoding, are listed in Table 2.1. Apart from these, there are 2 other amino acids which are also found in proteins synthesized in some organisms. One of these is Selenocysteine which is found in many organisms, but is not coded directly by DNA. The other one, Pyrrolysine, is found only in some archea and one bacterium. These 2 amino acids are referred to as “non-standard” amino acids.

The sequence of amino acids in a protein forms its primary structure. This primary structure of a protein determines its native conformation. The position of specific amino acid residues in the polypeptide chain dictates which portions of the protein fold closely together, to form its three dimensional structure. However, formation of a secondary structure is the first step in the folding process. There are 2 types of secondary structures:

Table 2.1: List of 20 standard amino acids along with their one letter codes.

Amino acid	Code	Amino acid	Code
Alanine	A	Arginine	R
Asparagine	N	Aspartic acid	D
Cysteine	C	Glutamic Acid	E
Glutamine	Q	Glycine	G
Histidine	H	Isoleucine	I
Leucine	L	Lysine	K
Methionine	M	Phenylalanine	F
Proline	P	Serine	S
Threonine	T	Tryptophan	W
Tyrosine	Y	Valine	V

α -helices and β -sheets. These structures contain a hydrophilic portion and a hydrophobic portion. After formation of the secondary structures, folding occurs so that the hydrophilic sides are facing the aqueous environment surrounding the protein and the hydrophobic sides are facing the hydrophobic core of the protein. This gives way to tertiary structure formation.

Protein tertiary structure can be divided into four main classes based on the secondary structural content of the domain. These four category of structural classes are depicted in Figure 2.1. The all- α class has a domain core built exclusively from α -helices. The all- β class has a core composed of anti-parallel β -sheets, usually two sheets packed against each other. α/β domains are made from a combination of β - α - β motifs that predominantly form a parallel β -sheet surrounded by amphipathic α -helices. $\alpha+\beta$ domains are a mixture of all- α and all- β motifs.

With the basic understanding of proteins, we can now discuss the concepts of protein attribute prediction. There are significant number of attributes associated with a protein that researchers are interested in. For example, what is its folding rate? Which structural class does it belong to? Which subcellular location site does it reside? Can it simultaneously exist in or move between two and more subcellular locations? Is it an enzyme? Is

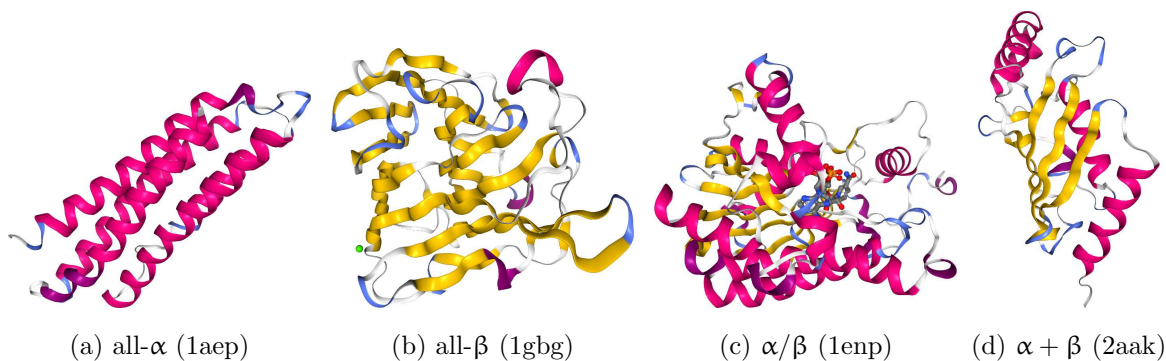


Figure 2.1: Four categories of protein structural class. The images were generated using NGL Viewer [237, 238]. The PDB codes used to generate the 3d structural views are noted in parenthesis beside each class name. The molecules are color coded by protein secondary structure. Alpha helices are colored magenta, beta sheets are colored yellow, turns are colored pale blue and all other residues are colored white.

it a DNA-binding protein? Is it an antigen? Is it a membrane protein or non-membrane protein? Which part of the protein serves as its signal sequence? Where are its cleavage sites? This list can go on. Answering all these interesting questions would require biochemical experiments which are tedious and expensive. Therefore, the research community has resorted to computational methods for predicting various attributes related to proteins, as highlighted in the above questions. This is known as protein attribute prediction.

2.1.1 Protein Attribute Prediction Pipeline

In the 2011 review paper, Chou [61] outlined a 5 step procedure for establishing a useful tool for any protein attribute prediction problem. These steps can be summarized as follows:

1. **Benchmark dataset preparation.** As the first step in the prediction pipeline, a stringent benchmark dataset should be prepared or collected to train and test the predictor. To avoid homology bias, the datasets should contain proteins with pair-

wise sequence similarity no more than a certain cutoff or threshold. Different cutoff values have been observed in literature, such as 25% [177], 30% [193], 40% [288]. Chou [61] recommended ensuring a 25% cutoff to create a stringent benchmark dataset.

2. **Protein sample representation.** The protein samples should be represented through a feature vector that is expressive enough so that the downstream processes can extract and utilize intrinsic information relevant to the attribute to be predicted. Chou's General PseAAC [59, 61], which has been widely adopted in this regard, is briefly described later in this chapter.
3. **Application of a prediction algorithm.** A powerful algorithm should be developed or an existing algorithm should be customized for the prediction process. Examples of prediction algorithms include sequence similarity based methods such as BLAST [14], FASTA [220], PSORT [204], empirical statistical methods and machine learning (ML) based algorithms. We have applied machine learning based approaches in our work and therefore briefly describe this concept later in this chapter. Among the ML algorithms, widely used in the field of protein attribute prediction are Artificial Neural Networks (ANN) [111], Support Vector Machine (SVM) [32], Random forests [34] etc. We briefly describe SVM and random forests algorithms later in this chapter, as we have extensively used these tools in our research. We also describe the concept of feature selection, which is generally applied to compress the protein sample representation before applying any prediction algorithm.
4. **Predictor evaluation.** The developed predictor should be objectively assessed. Several well-established testing methods exist that can assess the quality of the predictor while it is being trained as well as after the training has been completed. We briefly review these techniques later in the chapter.
5. **Make the predictor publicly available.** Generally, this is best done in form of a web interface that is user friendly and publicly accessible.

2.1.2 Machine Learning

Machine learning (ML) is the process of gaining knowledge from data, which can then be utilized in making decisions or predictions on unforeseen data. This process is extremely useful in situations where an analytic solution is lacking, but data abounds that can be utilized to construct an empirical solution. This concept of *learning from data* is one of the most widely adopted tools today by both researchers and practitioners in various fields of science and technology.

The available data that is exploited in the learning process is referred to as the *training data*. Based on the information in the training samples or data points, the process of learning can be *supervised* or *unsupervised*. In supervised learning, the training data contains explicit examples of what the correct output should be for given inputs. As the training examples are marked with the correct output, the data in this case is said to be *labeled*. It is then possible to empirically formulate a mapping from the input space to the output space, which can then be used to predict output for unforeseen input data.

On the contrary, in unsupervised learning, the input examples are not labeled with the correct outputs. Such data is called *unlabeled* data. Despite this lack of crucial information, it is still possible to extract patterns and clustering in the data. When a new data point (*query point*) comes in, the task is then to determine to which cluster the query point closely resembles and make decisions or predictions accordingly.

In this thesis, we have applied supervised learning in solving several protein attribute prediction problems. Therefore, let us now define the problem of supervised learning more formally. Let there be an input \mathbf{x} and an unknown target function $f : \mathcal{X} \rightarrow \mathcal{Y}$. Here \mathcal{X} is the input space and \mathcal{Y} is the output space. There is a data set \mathcal{D} of labeled examples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$, where $y_n = f(\mathbf{x}_n)$, $1 \leq n \leq N$. Then the problem of supervised learning is to learn from the data set \mathcal{D} a formula $g : \mathcal{X} \rightarrow \mathcal{Y}$ that approximates f . The choice of g is made from a set of candidate formulas under consideration, which we call the hypothesis set \mathcal{H} .

2.1.3 Classification and Regression

When the labels of input in a machine learning problem are discrete, it is called a classification problem. If there are only 2 possible labels, then it is called a binary classification problem. If there are more than 2 classes, then the problem is called a multi-class classification problem.

When the input labels take continuous values, then the learning problem is called a regression problem. Interestingly, regression analysis can also be used to solve what is an inherently binary classification problem. In this case, the regression process generates a score for each training sample. Then an optimal threshold is chosen to cluster the scores into 2 separate classes. The learnt model, including the threshold, can then be applied to a query data point to predict its class. In this thesis, we have used both binary classification and regression analysis.

2.1.4 Support Vector Machine

The Support Vector Machine (SVM) [32] algorithm formulates the supervised learning problem as an optimization problem. Given the labeled training samples, it tries to find an optimal separating hyperplane such that the distance from the hyperplane to the nearest data points is maximized. The larger this distance (i.e. “margin”), the lower the generalization error of the classifier. The classifier that it outputs is often referred to as the *maximum margin classifier*. The data points that are nearest to the hyperplane are called the *support vectors*.

SVM can be applied directly in the input space or in a transformed higher dimensional space. If the original problem has samples that are not linearly separable, then the original finite-dimensional space can be mapped into a much higher-dimensional, possibly infinite dimensional, space. This transformed space can yield the problem to be linearly separable and SVM then produces an optimal separating hyperplane in this space. To keep the computational load reasonable, the mappings are designed to ensure that dot products

may be computed easily in terms of the variables in the original space, by defining them in terms of a *kernel function*. Several kernel functions have been proposed and have successfully been used in predicting different protein attributes. These include linear kernel, radial basis function (RBF) kernel and polynomial kernel.

2.1.5 Random Forests

Random forests [34] is an ensemble learning method. An ensemble method uses several learning algorithms to obtain better predictive performance. Random forests algorithm constructs many decision trees at training time. A decision tree is a flowchart-like structure in which each internal node represents a comparison on an attribute. Each branch of the tree represents the outcome of the test. The leaf nodes represent class labels. The paths from root to leaf therefore represent all the classification rules.

Random forests algorithm constructs a multitude of decision trees and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. As the individual trees can overfit to the training data, averaging the result over the many decision trees regularizes the result and allows for a generalized classifier.

The importance of different attributes or features of the data points can be computed by permuting out-of-bag (OOB) data of random forests algorithm. This importance score indicates the global importance over all OOB cross-validated predictions and is very robust as it is averaged over all predictions for a given feature variable. Known as *mean decrease in accuracy*, this score can be used to rank the features and subsequently filter out irrelevant features. The larger this value is for a feature, the more important that feature is in the context of the prediction task.

2.1.6 Feature Selection

It is computationally expensive to work with a large feature vector, both during the learning phase and the prediction phase. Besides, all features may not always be effective in the learning model [117,242]. As such, after protein sample representation (also known as *feature extraction*), the next obvious step is to select a set of relevant features that will contribute to the learning model in improving accuracy.

Feature selection techniques can generally be divided into three categories: filter methods, wrapper methods and embedded methods. Filter methods rank the features based on some criteria. Then a subset of top ranked features are passed to train the classifier. These methods are thus independent of the choice of the classifier. Wrapper methods, on the other hand, search the feature space to find an optimal subset of features. The quality of the feature subset is measured by training and testing a specific classification model. Therefore such methods are tied to specific classification algorithms. Embedded methods are similar to wrapper methods. However, in this approach, the search for the optimal feature subset is inherently built into the classification algorithm.

2.1.7 Chou's General PseAAC

With the explosive growth of biological sequences in the post-genomic era, one of the most important, albeit difficult, problems in computational biology is how to express a biological sequence with a discrete model, yet capture considerable amount of sequence-order information. In a discrete model, each protein is represented by a fixed length feature vector that is independent of the protein sequence length. This model is preferred because all the existing machine-learning algorithms can only handle feature vectors but not sequence samples [63]. However, a vector defined in a discrete model may completely lose all the sequence-pattern information. To overcome this, the concept of Pseudo amino acid composition, or *PseAAC* in short, was proposed by Chou [59] in 2001. Since then, PseAAC has been widely used in nearly all the areas of computational proteomics (see, for example, [28, 147, 150, 195, 196, 250, 290] as well as a long list of references cited

in [64]). Because of its wide adoption, several open access softwares such as *propy* [38] and *PseAAC-General* [86] were established. Encouraged by the successes of using PseAAC to deal with protein/peptide sequences, the concept of PseKNC (Pseudo K -tuple Nucleotide Composition) [47] was developed for generating various feature vectors for DNA/RNA sequences that have proven very useful as well [45, 48, 165, 179]. Particularly, recently a very powerful web-server called *Pse-in-One* [171] and its updated version *Pse-in-One 2.0* [175] have been established that can be used to generate any desired feature vectors for protein/peptide and DNA/RNA sequences according to the need of users' studies. In this thesis, we have used Chou's general PseAAC to represent protein samples and developed various protein attribute prediction models.

Let a protein sequence P of length L be written as:

$$P = R_1 R_2 R_3 R_4 R_5 \dots R_L \quad (2.1)$$

where R_1 represents the first amino acid residue, R_2 the second residue and so on. The PseAAC of the protein can be represented as follows:

$$P = [\psi_1 \ \psi_2 \ \dots \ \psi_u \ \dots \ \psi_\Omega]^T \quad (2.2)$$

Here, the classical amino acid composition (AAC) is represented by subscripts $1 \leq u \leq 20$ and the subsequent features express sequence order information through one or more different schemes. The sequence order related features that we have utilized can largely be divided into two categories: position independent and position specific. Among the position independent features are Dipeptides (Dip), Tripeptides and n -Gapped-Dipeptides (nGDip). These features do not depend on any specific position in the amino acid sequence. These features have widely been used in the literature of protein attribute prediction. The position specific features, on the other hand, are introduced in this thesis. Below, we describe these feature schemes that constitute the generalized pseAAC in our work. In describing the feature types, we have followed the nomenclature from [241].

Amino Acid Composition (AAC)

Amino Acid Composition (AAC) of a protein sequence means the normalized frequencies of the 20 native amino acids. The frequencies are normalized by dividing those by the protein sequence length.

Dipeptides (Dip)

The normalized frequency of adjacent amino acids within the protein sequence can be used as features. This is called Dipeptides (Dip) or Dipeptide Composition (DPC) feature type. This feature type provides into the feature vector some sequence-order information and has been successfully used in several protein related studies [10, 30, 82, 114, 145, 166, 275]. Dip (DPC) contributes 400 features to PseAAC.

Tripeptides

We have similarly applied the notion of Tripeptides to extract another 8000 features. All these feature types are derived from the generalized form of *n-grams* feature type where frequencies of *n*-length peptides are used as feature vectors. Dong et al. [82] referred to it as *kmer*. In our study, we have extracted *n*-grams (*kmer*) features, for $n = 1, 2$ and 3 .

***n*-Gapped-Dipeptides (nGDip)**

In the *n*-Gapped-Dipeptides (nGDip) feature type, we count the frequency of amino acid dipeptides such that the amino acids are separated by n positions. The frequency is normalized, dividing it by the total number of nGDip. (i.e., $L - n - 1$ for a sequence of length L). For each specific gap, 400 features can be generated. The motivation for this feature type stems from the belief that the gap between any two amino acids may carry significant information about the protein [41]. Also known as Gapped Di-peptide Composition (GDPC), this feature extraction technique has recently become popular for protein classifications [184], protein structural class prediction [185], sub-Golgi protein identification [79, 80, 288], DNA binding protein prediction [177] etc.

Position Specific n -grams (PSN)

The Position Specific n -grams (PSN) represent whether specific n -grams occur in specific positions in the protein sequence. The value of each such feature in any sequence will therefore be either 0 or 1 (*on* or *off*). We have introduced this feature extraction technique in this thesis. We have considered PSN feature for $n = 1, 2$ and 3 . These can be referred to as position specific amino acids, dipeptides and tripeptides respectively.

As an example, consider the sequence “AAP TAA”. In the first position, we have the amino acid “A”, the dipeptide “AA” and the tripeptide “AAP”. Therefore, the position specific n -grams features (1, “A”), (1, “AA”) and (1, “AAP”) will be set (i.e. will have values of 1). Similarly, the other PSN features that will be “on” are (2, “A”), (2, “AP”), (2, “APT”), (3, “P”), (3, “PT”), (3, “PTA”), (4, “T”), (4, “TA”), (4, “TAA”), (5, “A”), (5, “AA”), (6, “A”).

2.1.8 Testing a Predictor

Several testing methods exist that can assess the quality of the learning model while it is being trained as well as after the training has been completed. These include jackknife cross-validation, 10-fold cross-validation test, independent test etc. We briefly describe these techniques below.

Jackknife Cross-validation

In jackknife cross-validation, one sample from the training set is set aside. The remaining part is used to train the predictor. Then the set-aside sample is used to test the model. This is repeated N times, where N is the size of the training set. In each iteration, the testing sample is always different from previous testing samples, so that all samples are considered once as the testing sample. Though this technique executes slowly compared to other testing techniques, it produces unique results. This technique has been used in this thesis. Since one sample is left out in each iteration, this technique is also widely known as *Leave-one-out* cross-validation technique.

Independent Testing

In independent testing, the testing dataset is completely different from training dataset. After the model is completely trained using the training set, independent testing is performed using the testing dataset. The distribution of the testing dataset should be similar to that of the training dataset. Otherwise, output of this testing strategy may be misleading [149].

10-fold Cross-validation

In 10-fold cross-validation technique, training dataset is divided into 10 equal parts. Among these 10 parts, one part is used for testing and other 9 parts are used to train the model. This is repeated 10 times so that each part gets to be used for testing exactly once.

2.1.9 Predictor Performance Metrics

As predictor performance metrics, we have used in this thesis accuracy, sensitivity, specificity and Matthew's Correlation Coefficient (MCC). These are well-established performance metrics in the literature [13, 225]. These metrics are calculated using a confusion matrix which can be generated based on true classes and predicted classes [148]. We have also analyzed the Area Under Receiver Operating Characteristic Curve (ROC-Curve) [95] and Area Under Precision-Recall Curve (PR-Curve) [73].

The samples in the dataset can be categorized into two classes: the positive class and the negative class. When the true class of a test sample is positive (negative) and the predicted class is also positive (negative), it is called a True Positive (True Negative). When true class of a testing sample is positive (negative) but predicted class is negative (positive) it is called False Negative (False Positive). Let P , N , TP , TN , FP , FN respectively denote the number of positives, negatives, true positives, true negatives, false positives and false negatives. Then we can define the relevant performance metrics by the

following set of equations:

$$\left\{ \begin{array}{l} \textit{Accuracy} = \frac{TP+TN}{P+N} \\ \textit{Sensitivity} = \frac{TP}{TP+FN} \\ \textit{Specificity} = \frac{TN}{FP+TN} \\ \textit{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}} \end{array} \right. \quad (2.3)$$

However, in the form of Equation 2.3, these metrics lack intuitiveness and is not easy-to-understand for most biologists. In particular, the interpretation of MCC is not at all intuitive in this form, although it is very important in measuring the stability of a prediction method. Therefore, we adopt the formulation based on Chou's symbols [60] that was recently proposed in [46, 287] as follows: Let N^+ (N^-) be the total number of positive (negative) samples in the dataset. Let N_+^+ (N_+^-) be the number of positive (negative) samples that were incorrectly predicted. The relationship between the symbols used in Equation 2.3 and Chou's symbols just introduced can be given by the following equation:

$$\left\{ \begin{array}{l} TP = N^+ - N_+^+ \\ TN = N^- - N_+^- \\ FP = N_+^- \\ FN = N_+^+ \end{array} \right. \quad (2.4)$$

And the performance metrics can then be redefined as:

$$\left\{ \begin{array}{l} \textit{Accuracy} = 1 - \frac{N_+^+ + N_+^-}{N^+ + N^-} \\ \textit{Sensitivity} = 1 - \frac{N_+^+}{N^+} \\ \textit{Specificity} = 1 - \frac{N_+^-}{N^-} \\ \textit{MCC} = \frac{1 - \left(\frac{N_+^+}{N^+} + \frac{N_+^-}{N^-}\right)}{\sqrt{\left(1 + \frac{N_+^- - N_+^+}{N^+}\right) \left(1 + \frac{N_+^+ - N_+^-}{N^-}\right)}} \end{array} \right. \quad (2.5)$$

From the definitions in Equation 2.5, the interpretation of each of the performance metrics is much more intuitive and easier-to-understand. For example, when all the

instances of the positive (negative) class are correctly predicted, we have $N_{-}^{+} = 0$ ($N_{+}^{-} = 0$) and thus sensitivity (specificity) of the classifier is 1. On the contrary, if all the positive (negative) instances are incorrectly predicted, then $N_{-}^{+} = N^{+}$ ($N_{+}^{-} = N^{-}$). Therefore, sensitivity (specificity) becomes 0. For a perfect classifier, we have $N_{-}^{+} = N_{+}^{-} = 0$ and both accuracy and MCC become 1 in this case. On the other hand when all the samples are misclassified (i.e. $N_{-}^{+} = N^{+}$ and $N_{+}^{-} = N^{-}$), then accuracy and MCC becomes 0 and -1 respectively. For a random predictor, we can expect $N_{-}^{+} = \frac{N^{+}}{2}$ and $N_{+}^{-} = \frac{N^{-}}{2}$, which results in an accuracy of 0.5 and an MCC of 0.

The advantages of Chou’s intuitive metrics have been analyzed and concurred by a series of studies published very recently (see, e.g., [44, 45, 51, 55, 97, 138, 139, 172, 173, 178, 187, 228–230]). It is important, however, to call out that the set of metrics, as described above, is valid only for the single-label systems (in which each protein only belongs to one functional class). For the multi-label systems (in which a protein might belong to several functional classes), whose existence has become more frequent in systems biology [51–54], systems medicine [55, 56] and biomedicine [230], a completely different set of metrics as defined in [62] is needed.

As mentioned earlier, in addition to the performance metrics described above, we have also analyzed the area under ROC and PR curves. The ROC-Curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. TPR is *sensitivity*, while $(1 - \textit{specificity})$ gives FPR. When ROC-Curve gets close to the left upper corner in the graph, it indicates better performance [95]. In this case, we get higher values for Area Under ROC-Curve (auROC). The PR-curve, on the other hand, is created by plotting the *precision* against the *recall* at various threshold settings. Precision represents true positive accuracy. Recall, on the other hand, reports the true positive rate and therefore is identical to sensitivity. These metrics can formally be defined by the following set of equations:

$$\begin{cases} \textit{Precision} &= \frac{TP}{TP+FP} \\ \textit{Recall} &= \frac{TP}{TP+FN} \end{cases} \quad (2.6)$$

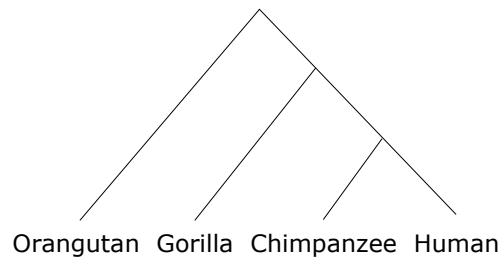


Figure 2.2: A phylogenetic tree relating four species: human, chimpanzee, gorilla and orangutan.

The closer the PR-curve is to the top right corner of the graph, the better is the performance of the predictor [73]. In this case, we get higher value for Area Under PR-Curve (auPR).

2.2 Phylogeny Reconstruction

We now discuss preliminary concepts about phylogeny reconstruction, which is the focus of Part II of this thesis. Phylogeny refers to the evolutionary relationships among a set of entities. Such entities may include species, genes, languages, etc. In our work, the first 2 entities are the most relevant. The entities amongst which an evolutionary relationship is being sought are referred to as taxa. Each such taxon is placed as a leaf in a phylogenetic tree and the tree topology represents the evolutionary history. The non-leaf (i.e. internal) nodes of the tree represent hypothetical ancestral taxa from which the present day taxa evolved. These ancestral taxa are believed to have existed in the past, but has become extinct. Notably, a tree T is a connected acyclic graph with a set of vertices V and a set of edges E . The vertex and edge sets are sometimes also shown as $V(T)$ and $E(T)$ respectively. An edge $e = (u, v) \in E$ represents an evolutionary relationship between the two taxa represented by the vertices u and v . The set of internal nodes is represented by $V_{int}(T)$, while the set of leaf nodes (i.e. the present day taxa set) by $L(T)$.

Figure 2.2 shows a sample phylogenetic tree among four species: human, chimpanzee, gorilla and orangutan. This evolutionary tree depicts that human and chimpanzee share a

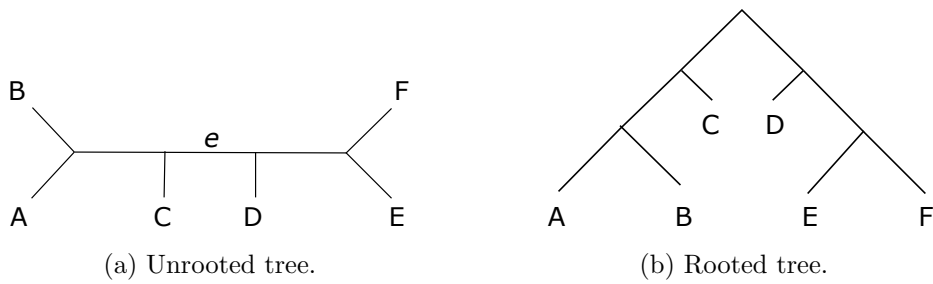


Figure 2.3: Rooted and unrooted trees.

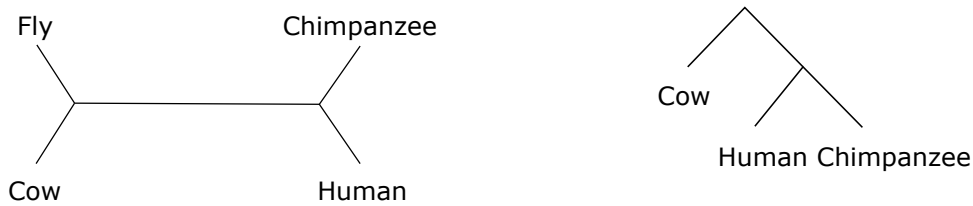


Figure 2.4: Phylogenetic tree on a set of mammalian species, with fly as the outgroup.

common ancestor. As such, we consider humans to be more closely related to chimpanzees than they are to gorillas and orangutans.

2.2.1 Rooted and Unrooted Trees

In a rooted phylogenetic tree, one vertex $r \in V$ is designated as the root of the tree. The root is generally denoted by $root(T)$. In an unrooted tree, on the other hand, there is no such designated vertex. Figure 2.3 shows samples of rooted and unrooted trees on a set of 6 taxa $\{A, B, C, D, E, F\}$. The rooted tree in the figure is obtained by rooting the unrooted tree on the edge e .

True evolutionary histories are better represented by a rooted tree. However, identifying the root of an estimated phylogenetic tree is generally a difficult task. It requires precise knowledge of the set of taxa under consideration. Another approach to tree rooting takes the assumption of a “molecular clock” which implies that DNA and protein sequences evolve at a rate that is relatively constant over time and among different organisms. However, such assumption seldom holds in real datasets. Therefore, a common approach that is applied to tree rooting uses an *outgroup*, which is a taxon known to have

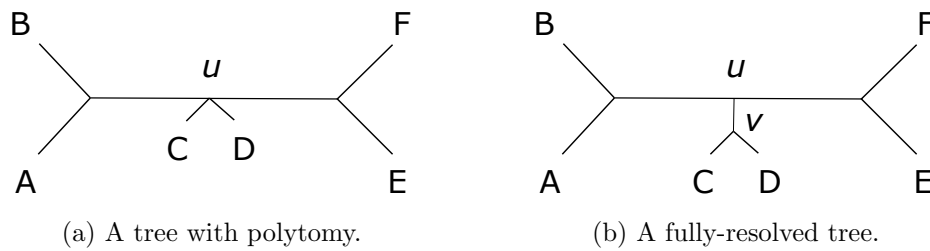


Figure 2.5: Binary and non-binary phylogenetic trees of 6 taxa $\{A, B, C, D, E, F\}$. In the left figure, u is a polytomy. In the right figure, the polytomy has been resolved to create a binary tree.

branched off before all other taxa under consideration. The outgroup is added to the set of taxa under study and an unrooted tree is estimated on this augmented set. This unrooted tree is then rooted by “picking up” the unrooted tree at the outgroup. This is shown in Figure 2.4. In the left part of the figure, an unrooted tree is estimated on a set of mammalian taxa (cow, chimpanzee and human) with addition of the outgroup fly. This unrooted tree is then picked up at the fly node to obtain the rooted tree that is depicted at the right part of the figure.

2.2.2 Binary and Non-binary Trees

A Phylogenetic tree can be binary or non-binary. A tree is called binary or fully-resolved if all internal nodes have degree at most three. In a non-binary tree, on the other hand, there is at least one node with degree greater than three. Such a node is known as a polytomy. In Figure 2.5a, the vertex u has degree 4. Therefore, u is a polytomy and the tree is a non-binary tree. The polytomy is resolved in Figure 2.5b, by introducing an additional vertex v , resulting in a fully-resolved tree.

2.2.3 Clade and Bipartition

In a rooted phylogenetic tree, each internal vertex defines a group of taxa that are more closely related to each other than they are to any other taxon in the tree. Such a group is called a clade. Formally, a clade in a phylogenetic tree T is a rooted subtree of T .

In case of unrooted trees, the similar concept of grouping is captured by bipartitions of the taxon set. For each edge e of a phylogenetic tree T , there is a bipartition π_e . Deleting the edge e from T creates two subtrees T_1 and T_2 , resulting in a bipartition of leaves $L(T_1)|L(T_2)$. Observe that, an edge incident on a leaf creates a bipartition in which the corresponding leaf is in one party and the remaining nodes are in the other party. Such a partitioning is called a trivial bipartition since it does not convey any information about the topology of the tree. Bipartitions corresponding to the internal edges, on the other hand, are called non-trivial bipartitions.

2.2.4 Branch Length

The length of an edge (or branch) in the phylogenetic tree is called the branch length. Branch length is a non-negative real number that may represent various quantities measured on a branch. For example, a branch length can represent the amount of evolutionary change or the amount of time between two nodes.

2.2.5 Gene Tree-Species Tree Discordance

Equipped with the basic concepts of phylogeny, we can now discuss the gene tree-species tree discordance and the reasons behind it. A species tree represents the evolutionary history among a set of species via the process of speciation. On the contrary, a gene tree represents the evolution of a particular “gene” within a group of species. When species are split by speciation, the gene copies within species are also split into separate lineages of descent. Within each such lineage, the gene trees continue branching and descending through time. Thus, the gene trees are contained within the branches of the species trees [192].

However, when gene copies are sampled from various species, the gene tree relating these copies might disagree with the species phylogeny. Figure 2.6 shows an example of discordance between a species tree and a gene tree. Here, species B and C are “sister”

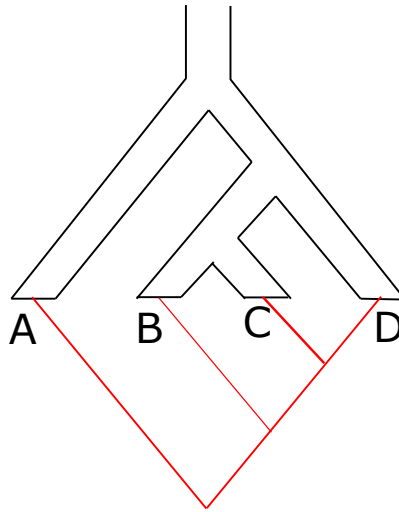


Figure 2.6: Gene tree-species tree discordance or incongruence. A species tree (given in block diagram) and a gene tree (given in line diagram) on the same set $\{A, B, C, D\}$ of taxa with different topologies.

species. However, in the gene lineage, C is closer to D than B . This discord can arise from horizontal transfer, incomplete lineage sorting, and gene duplication and extinction [192]. Additionally, an apparent gene tree-species tree discordance might simply be due to error introduced during the reconstruction of one or more gene trees [192].

As Maddison [192] philosophizes, “Perhaps it is misleading to view some gene trees as agreeing and other gene trees as disagreeing with the species tree; rather, all of the gene trees are part of the species tree, which can be visualized like a fuzzy statistical distribution, a cloud of gene histories. Alternatively, phylogeny might be (and has been) viewed not as a history of what happened, genetically, but as a history of what could have happened, i.e., a history of changes in the probabilities of interbreeding.”

Maddison [192] further writes, “When we take a sample from a population and try to understand a statistical distribution by calculating means and variances, we do not single out all of the samples whose values differ from the mean as disagreeing with the mean. They are simply part of the variance, part of the distribution. A simple phylogenetic tree diagram with sticklike branches represents only the mean or mode of a distribution. Phylogeny has a variance as well, represented by the diversity of trees of different genes.”

Horizontal Gene Transfer

Horizontal Gene Transfer (HGT) is the process that causes the genes to be transferred across species. If the native gene copy in the receiving species lineage goes extinct or is not sampled, then the gene tree will disagree with the species tree [192]. Also known as *lateral gene transfer*, this process might be accomplished by hybridization or a vector such as a virus or mite [71]. Successful transfer requires the transferred genes to become functioning members of the receiving genome [71]. HGT is expected to be less likely the more phylogenetically distant the original and receiving species are [192].

Incomplete Lineage Sorting

Incomplete lineage sorting (ILS) refers to the failure of two gene lineages to coalesce at their speciation point. Also known as deep coalescence, this process is best understood under the coalescent model [75, 76, 207, 264, 265]. The coalescent model explains evolutionary process by going backwards in time and connecting gene lineages to a common ancestor through a process of “coalescence” of lineage pairs. In this model, each species is treated as a population of individuals, having a pair of alleles for each gene. The present day variants of a gene (known as alleles) are then traced back in time across successive generations by following the ancestral alleles in the previous generation from which this given alleles evolved. Eventually a point is reached where two alleles coalesce (i.e., they find a common ancestor). The multi-species coalescent (MSC) model is the extension of this general coalescent framework where multiple randomly mating populations corresponding to multiple species are present.

Under the coalescent model, ILS can be a source of gene tree discordance, as the common ancestry of gene copies at a single locus can extend deeper than speciation events. The larger the effective population size and the shorter the branch length of the evolutionary tree, the greater the chance of ILS or deep coalescence to occur [192, 217].

Figure 2.7 shows an example of discordance due to ILS. The gene copies within species *B* and *C* first meet at their corresponding speciation point as we go back in time. The

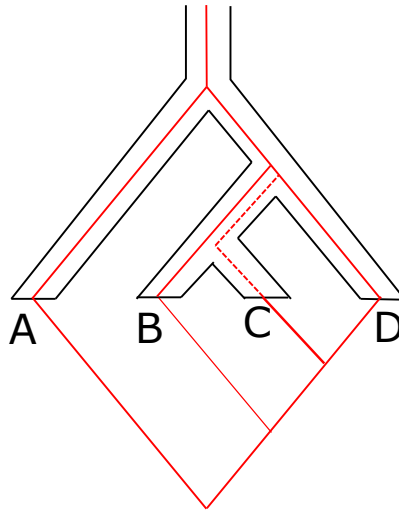


Figure 2.7: Example of gene tree-species tree discordance due to incomplete lineage sorting. Going back in time, the gene copies within species *B* and *C* first meet at their corresponding speciation point, but fail to coalesce. Both the lineages (dashed and solid black lines) exist on deeper ancestral branch. The gene from *C* first coalesces with the gene from species *D*, and subsequently with the gene from *B*.

speciation point is the most recent common ancestor of species *B* and *C*. However, the gene copies fail to coalesce here. Both of these copies go further back in time, resulting in two gene lineages on deeper ancestral branch. The extra lineage is shown by the dashed black lines in Figure 2.7. Then the gene from *C* first coalesces with the gene from species *D*, and subsequently with the gene from *B*.

Gene Duplication and Extinction

Gene duplication is a process that generates multiple gene lineages that coexists in a species lineage [215]. A gene duplication event creates a second locus, and both loci thereafter evolve independent of each other. This results in discordance between gene tree and the containing species tree [112]. Also, some of the gene lineages could go extinct if it decayed into a “pseudo-gene”, or if it evolved a new function and diverged [192]. This phenomenon is known as gene extinction (also known as gene loss) which too may result in gene tree incongruence.

2.2.6 Gene Tree Parsimony

Gene tree parsimony (GTP) is an optimization problem that estimates species trees from a set of gene trees. In this approach, various possible species trees are assessed and for each tree we determine what evolutionary events the species tree requires to explain the observed gene trees. The tree that results in the minimum number of evolutionary events is the most parsimonious tree [192].

As discussed earlier, there are three different classes of evolutionary processes by which discordance between gene trees and species trees arise: horizontal gene transfer, incomplete lineage sorting, and gene duplication and loss. For each of these processes, the parsimony criterion would be different. In case of horizontal transfer, the species tree that minimizes the number of transfer events is sought. In case of deep coalescence or ILS, we find the tree that minimizes the number of extra gene lineages that had to coexist along species lineages. For gene duplication, the parsimony criterion is to choose the tree minimizing duplication and/or extinction events [192].

It is plausible to construct a mixed method that allows for each of these discordance to occur. However, it is certainly difficult [192]. Therefore, typically only gene duplication and loss are considered in GTP [24].

2.2.7 Statistical Consistency

A species tree reconstruction method is said to be statistically consistent under a particular model of evolution if the probability of returning the true species tree converges to one as the amount of data increases. Here the increase in data refers to increase in both the number of sites (i.e. gene length) and the number of loci (i.e. the number of genes).

Let the set of genes in a study be $\mathcal{G} = \{g_1, g_2, \dots, g_k\}$. Let s_i be the number of sites in g_i ($1 \leq i \leq k$). Then A species tree estimation method is statistically consistent if the estimated species tree converges to the true species tree as $k \rightarrow \infty$, $\underset{1 \leq i \leq k}{s_i} \rightarrow \infty$.

2.2.8 Species Tree Estimation Methods

A species tree estimation method takes a collection of gene trees as input and attempts to estimate the species phylogeny. The various methods found in literature for estimating species trees can largely be divided into two categories. These are *concatenation* and *summary methods*.

In the concatenation approach (also known as *combined analysis*), alignments are estimated for each gene and then concatenated into a *supermatrix*. The supermatrix of alignments is then used to estimate the species tree. Concatenation does not consider gene tree discordance. As such, when the genes differ in evolutionary history, this approach can return incorrect trees with high confidence [74, 124, 153, 158, 159, 181].

Summary methods, on the other hand, construct species trees by summarizing a collection of gene trees. These methods take the reason of discordance into account. Gene tree parsimony methods such as estimating species trees by minimizing deep coalescence (MDC) and minimizing duplication and loss (MGDL) are examples of summary methods. Such methods are becoming more popular these days [124, 180, 181, 199, 200, 209, 236, 292]. Summary methods such as STEM [152], STAR [182], *BEAST [124], BUCKy-pop [158], GLASS [202], MP-EST [181] and ASTRAL [199, 200, 292] consider ILS as the reason for discordance and are statistically consistent. On the other hand, methods such as greedy consensus, minimize deep coalescence (MDC) [192], matrix representation with parsimony (MRP) [22], matrix representation with likelihood (MRL) [209] etc. are not statistically consistent, but perform well in practice.

2.2.9 Evaluation of Species Tree Estimation Methods

Here we describe the various standard ways of measuring species tree estimation error on simulated datasets. Since the ground truth (i.e. the model or true species tree) is known in a simulated dataset, we compare the species trees estimated by the methods of consideration with this true tree. We now describe the metrics that are widely used to quantify the species tree reconstruction error.

False negative (FN) rate

The False negative (FN) rate is the proportion of the edges present in the true tree but not present in the estimated tree. This is also known as the *missing branch rate*.

False positive (FP) rate

The FP rate is the proportion of the edges present in the estimated tree but not in the true tree. For a binary tree, the FP rate is identical to the FN rate. However, for non-binary trees, these quantities are not necessarily identical. The FP rate is not a good measure of accuracy in the latter case. For example, let us assume that a species tree reconstruction method produced a star (a tree with one internal node) for an arbitrary true tree. In this case, the FP rate is zero even though the estimated tree fails to reconstruct the internal edges.

Robinson-Foulds (RF) rate

The Robinson-Foulds (RF) rate is the ratio of the total number of false positive and false negative edges to the total number of internal edges in the two trees. When true and estimated trees are binary, the RF is identical to the FP and FN rates. While RF rate is the most commonly used error metric, this metric is not appropriate when the trees are not binary for the same reason described above.

Quartet support

The *quartet support score* [199] measures the similarity between a candidate tree T and the input gene trees, and is computed as follows. Each input gene tree is decomposed into its induced set of quartet trees (i.e., unrooted trees formed by picking four leaves). The quartet support score of a given candidate species tree T is the total, over all the input gene trees, of the number of induced quartet trees that T agrees with. As shown in [199], the tree that optimizes the quartet support score is a statistically consistent estimator of the true species tree under the multi-species coalescent model.

2.3 Conclusion

In this chapter we have provided an in depth technical background of various methods and concepts that are needed to comprehend the research conducted in this thesis. Next chapter begins Part I of this thesis that deals with a set of protein attribute prediction problems. In particular, in the next chapter, titled *sub-Golgi Protein Type Prediction*, we focus on building a new sequence based computational model for classification of sub-Golgi proteins.

Part I

Protein Attribute Prediction

Chapter 3

sub-Golgi Protein Type Prediction

The Golgi apparatus (GA) is a key organelle for protein synthesis within the eukaryotic cell. The main task of GA is to modify and sort proteins for transport throughout the cell. Proteins permeate through the GA on the ER (Endoplasmic Reticulum) facing side (*cis* side) and depart on the other side (*trans* side). Based on this phenomenon, we get two types of GA proteins, namely, *cis*-Golgi protein and *trans*-Golgi protein. Any dysfunction of GA proteins can result in congenital glycosylation disorders and some other forms of difficulties that may lead to neurodegenerative and inherited diseases like diabetes, cancer and cystic fibrosis. So, the exact classification of GA proteins may contribute to drug development which will further help in medication.

In this chapter, we focus on building a new computational model that not only introduces easy ways to extract features from protein sequences but also optimizes classification of *trans*-Golgi and *cis*-Golgi proteins. After feature extraction, we have employed random forests model to rank the features based on the importance score obtained from it. After

Much of the material in this chapter is taken without alteration from the following paper.

- Rahman, M. S., Rahman, M. K., Kaykobad, M., & Rahman, M. S. (2018). *isGPT: An optimized model to identify sub-Golgi protein types using SVM and Random Forest based feature selection*. Artificial Intelligence in Medicine 84 (2018) 90–100.

selecting the top ranked features, we have applied Support Vector Machine (SVM) to classify the sub-Golgi proteins. We have trained regression model as well as classification model and found the former to be superior. The model shows improved performance over all previous methods. As the benchmark dataset is significantly imbalanced, we have applied Synthetic Minority Over-sampling Technique (SMOTE) to the dataset to make it balanced and have conducted experiments on both versions. Our method, namely, *identification of sub-Golgi Protein Types (isGPT)*, achieves accuracy values of 95.4%, 95.9% and 95.3% for 10-fold cross-validation test, jackknife test and independent test respectively. According to different performance metrics, isGPT performs better than state-of-the-art techniques. The source code of isGPT, along with relevant dataset and detailed experimental results, can be found at <https://github.com/srautonu/isGPT>. A publicly accessible web interface has also been established at: <http://isgpt.research.buet.ac.bd/>.

3.1 Introduction

An eukaryotic cell is defined by a membrane-bound nucleus. All eukaryotic cells have a nucleus, a plasma membrane, ribosomes and cytoplasm [188]. Most of the eukaryotic cells have other small membrane-bound structures in cytoplasm called organelles and Golgi apparatus (GA) is one of them. It is a key organelle in protein synthesis along with some other elements of the cell [66]. It consists of disk like membranes called cisternae which are stacked together [157]. GA has three elements, namely, *cis*-Golgi, medial and *trans*-Golgi. *cis*-Golgi is responsible for receiving proteins, while *trans*-Golgi releases the synthesized proteins. The function of medial is to synthesize the received proteins from *cis*-Golgi (see Figure 3.1, image source: [284]). Endoplasmic Reticulum (ER) builds proteins and sends out to the cell through GA [284]. A side of GA facing ER (*cis*-side) captures those proteins (also called cargo proteins) for synthesis and send those out via the other side of GA facing the plasma membrane (*trans*-side). In the medial region, the cargo proteins get modified by the Golgi enzymes through addition or removal of sugars. Modifications may also occur through the addition of sulphate groups or phosphate groups [284].

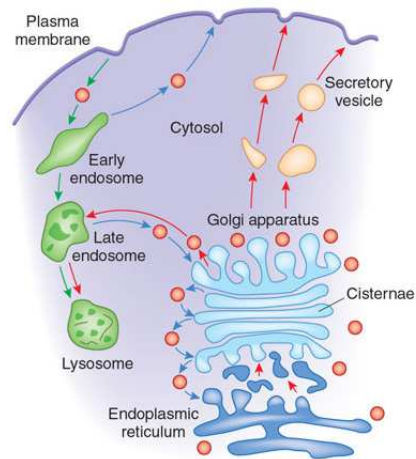


Figure 3.1: The Golgi apparatus and its synthesis process. (image source: [284])

Any functional deviation of GA may result in adaptable disorders during the synthesis process in medial which may further contribute to inheritable and neurodegenerative diseases such as diabetes [129], cancer [268], Parkinson’s disease [257] and Alzheimer’s disease [17]. It is necessary to identify any rambling and damage in a timely manner to better understand the problem of GA dysfunction. The current methods of treating patients having these diseases include neuroprotective therapies and anti-inflammation which are not able to provide a permanent solution [90]. Exact identification of sub-Golgi proteins can provide new insight for scientists to recognize the dysfunctions subscribed by Golgi proteins [267]. Thus, sub-Golgi (*cis*-Golgi vs. *trans*-Golgi) protein classification is very important for more effective drug-development.

Significant amount of research have been conducted during the last decade to build prediction tools for protein sub-cellular localization using machine learning methods [31, 92, 130, 169, 273, 291]. However, few tools have been developed for sub-Golgi protein classification. Nevertheless, researchers nowadays are focusing on this topic and trying to build efficient classification models. van Dijk et al. [269] did the pioneering work to predict sub-Golgi localization of type II membrane proteins. They used amino acid grouping in conjunction with string-based triads as well as 3D-structure based triads for protein representation. Here, Support Vector Machine (SVM) [32] with linear kernel was used as the classifier.

Ding et al. [80] used increment of Shannon entropy (IH) on amino acid compositions (AAC) and g-gap dipeptide compositions for protein representations. They then applied Modified Mahalanobis Discriminant (MD) algorithm to predict the Golgi-resident proteins. They achieved an accuracy of 74.7% using jackknife cross-validation test. Ding et al. [79] further continued their previous work and proposed a g-gap dipeptide based feature extraction technique. They used analysis of variance (ANOVA) test to select relevant features and applied SVM as the learner. This time, they obtained an accuracy of 85.4% using jackknife cross-validation.

Jiao et al. [140] presented a model which computes Positional Specific Physico-Chemical Properties (PSPCP) of a protein sequence. The PSPCP essentially integrates the Position Specific Scoring Matrix (PSSM) with the different physicochemical property values. ANOVA was applied for feature selection, while SVM with RBF kernel was used as the learner, to achieve an accuracy of 86.9%. In a subsequent study [141], they applied minimum Redundancy Maximum Relevance (mRMR) feature selection technique, instead of ANOVA, on the same feature space. This improved the accuracy to 91%.

Both Ding et al. [79,80] and Jiao et al. [140,141] used a small benchmark dataset, where there are only 150 GA proteins. In addition, the dataset is highly imbalanced - the number of *trans*-Golgi proteins is significantly lower than the number of *cis*-Golgi proteins. Yang et al. [288] have recently created an updated benchmark dataset where there are 304 sub-Golgi proteins for training and 64 sub-Golgi proteins for testing the classification model. They applied Synthetic Minority Over-sampling Technique (SMOTE) [43] to balance the dataset. They conducted experiments on both the imbalanced and balanced datasets and demonstrated improved accuracy with the balanced version. For feature selection, they used random forests [34] based recursive feature elimination method. They then applied random forests algorithm as the learning method as well. Their model shows accuracy values of 88.5%, 93.8% and 90.1% for jackknife cross-validation, independent testing and 10-fold cross-validation, respectively.

Very recently, Ahmad et al. [10] have also conducted similar kind of experiments though their feature construction, feature selection and learning algorithms are different. They have applied Fisher feature selection method to select relevant features and k-nearest neighbor (KNN) algorithm as the learner. The model proposed by Ahmad et al. reports accuracy values of 94.9%, 94.8% and 94.9% on the balanced benchmark dataset for jackknife cross-validation, independent testing and 10-fold cross-validation, respectively.

Exploring previous studies, we note that there is still room for improvement because even a small improvement in accuracy is highly demanding in bioinformatics tools. Improved accuracy can also contribute to better drug-development which is maintained by sensible computer-aided design [88,216].

There are three important tasks in the pathway of protein function predictions [241]. These include processing of datasets, construction of features from protein sequences and application of a suitable classification algorithm. In this chapter, we first construct a large set of features based on three feature construction techniques and then apply random forests algorithm on the constructed feature set. We select relevant features based on the importance score provided by the random forests model. Then, we apply SVM on the selected features for both classification and regression analyses. Our tool, named *identification of sub-Golgi Protein Types* or *isGPT* in short, is evaluated based on several well-established performance metrics and demonstrates superiority over existing methods.

Our overall contributions are summarized as follows:

- We present an easy and flexible method that produces several position specific as well as position independent features from protein sequences. Then feature selection is performed based on the importance score provided by the random forests model.
- We model the problem of sub-Golgi protein localization both as a classification problem and a regression problem. Using SVM, we train classification models as well as regression models on the benchmark (imbalanced) dataset as well as on the dataset, balanced with a celebrated balancing technique called SMOTE. We make a comparative analysis of the different models.

- Finally, through extensive experiments, we compare isGPT with the methods of [288] and [10] which are currently two state-of-the-art techniques. Our method shows superior results according to different performance metrics.

3.2 Material and Methods

In what follows, we describe our methodology in accordance with Chou’s 5-step procedure [61], which was briefly described in Section 2.1.1.

3.2.1 Benchmark Dataset

We have collected the training and testing benchmark datasets from Yang et al. [288], which have also been used by Ahmad et al. [10] recently to measure the performance of their method. The training dataset¹ contains 304 sub-Golgi protein sequences among which there are 87 sequences of *cis*-Golgi type and 217 sequences of *trans*-Golgi type. None of the proteins has more than 40% pairwise sequence identity with any other proteins in the dataset.

The testing dataset is used for independent testing and it contains 13 *cis*-Golgi protein sequences and 51 *trans*-Golgi protein sequences. This is the same set that was used in [10, 288] for independent testing and was first introduced by Ding et al. [79]. It is important to note here that the training and testing datasets are mutually exclusive.

We can easily observe that both training and testing datasets are highly imbalanced because these datasets contain 71.4% and 80% *trans*-Golgi protein sequences, respectively, among all sub-Golgi protein sequences.

¹Yang et al. constructed this dataset from Universal Protein Knowledge base (UniprotKB) [6]

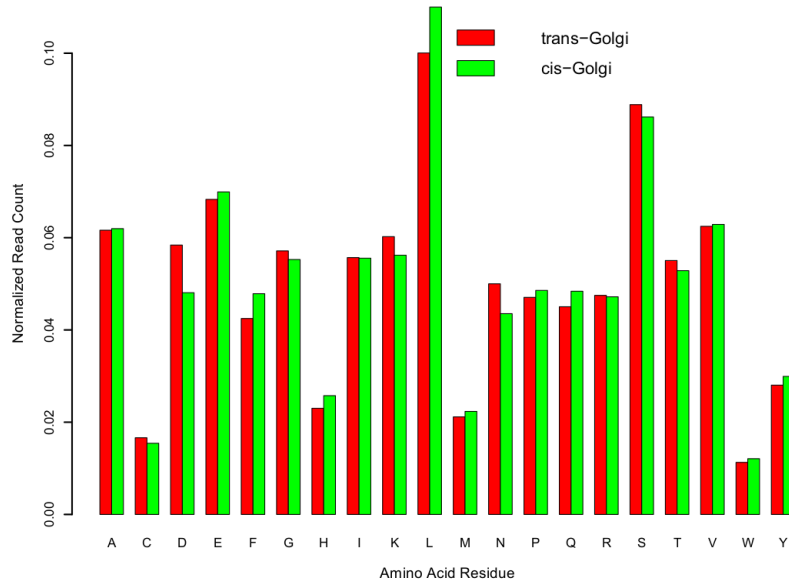


Figure 3.2: Amino Acid Composition (AAC), on average, for the different sub-Golgi protein classes in the training dataset.

3.2.2 Protein Sample Representation

A protein sample can be represented by its primary sequence, as shown in Equation 2.1. To represent each protein sample as a fixed length feature vector that is independent of the protein sequence length, we have utilized Chou’s general formulation of PseAAC [61] (described in Section 2.1.7). The generalized PseAAC of a protein, as defined in Equation 2.2, is as follows:

$$P = [\psi_1 \ \psi_2 \ \dots \ \psi_u \ \dots \ \psi_\Omega]^T$$

The classical AAC is represented by subscripts $1 \leq u \leq 20$ and the subsequent features express sequence order information through one or more different schemes. We have calculated the average AAC in the training dataset for *cis*-Golgi and *trans*-Golgi proteins (see Figure 3.2). We see that there is only slight difference in each amino acid ratio between *cis*-Golgi and *trans*-Golgi proteins. This indicates that AAC alone is unlikely to be able to categorize an unknown sub-Golgi protein sequence.

The sequence order related features that we have extracted can largely be divided into two categories: position independent and position specific. Among the position independent features, we have used Dipeptides (Dip), Tripeptides and n -Gapped-Dipeptides (nGDip). These features have widely been used in the literature. The position specific features, on the other hand, are something that are introduced in this thesis. All these feature extraction techniques have already been described in Section 2.1.7.

AAC, Dip and Tripeptides derive from the generalized form of n -grams feature type where frequencies of n -length peptides are used as feature vectors. Dip has also been successfully used in [10] for sub-Golgi protein type prediction. In our study, we extract a total of 8420 n -grams features, for $n = 1, 2$ and 3 . Note that, for some features, all the samples of sub-Golgi protein sequences may produce 0 frequency. Such features will naturally have no effect on the learning model. We have carefully removed these from the feature vector. Subsequently the n -grams feature count got reduced to 8348.

We have also applied the n -Gapped-Dipeptides (nGDip) feature extraction technique in this work. This technique has already been applied in the problem of sub-Golgi protein identification: both Ding et al. [79, 80] and Yang et al. [288] have utilized this feature scheme in their works. Yang et al. called it *g-Gap Dipeptide Composition* and used $g = 3$ only. In our work, rather than considering one specific gap, we have used nGDip (GDPC) feature type for gaps of upto 15 positions. Thus we get a total of $15 \times 400 = 6000$ n -Gapped-Dipeptides features.

Another type of feature scheme that we have used in this work is the Position Specific n -grams (PSN). As described in Section 2.1.7, PSN generates features which identify whether specific n -grams occur in specific positions in the protein sequence. The value of each such feature in any sequence will therefore be either 0 or 1 (*on* or *off*). If the maximum sequence length is L , the feature space size would be $L \times 20^n$. For small sample sizes, however, many of the features will not have discriminating scores. Such features that are on in all samples (or vice versa) can be excluded from the final feature vector. Thus the actual size may be considerably smaller than the theoretical maximum.

Like in the case of position independent features, we wanted to consider n -grams for $n = 1, 2$ and 3 in case of PSN as well. However, the feature space became too large for the computing power and the memory at the disposal of the machines we have used. As such, rather than considering each position of the sequence, we were motivated by the concept of Split Amino Acid Composition (SAAC), which was also used by Ahmad et al. [10]. In SAAC, a protein sequence is split into three parts: 25 residues at the N-terminus, the center part and the 25 residues at the C-terminus. Each portion is handled separately for feature extraction. In our case, we construct the PSN only from the N-terminus part. However, even with this part, the feature space is still too large. Therefore, we considered only the first 10 positions of the N-terminus part.

For a specific position, 20, 400 and 8000 PSN features can be generated for $n= 1, 2$, and 3 respectively. Since there are only 304 training samples, no more than 304 features can be generated for each of $n= 2$ and 3 , such that at least one sample has the respective feature on. Thus the number of features with discriminating scores cannot be larger than $(10 \times 20 + 9 \times 304 + 8 \times 304) = 5368$. Depending on the actual samples, this number may be less than this higher bound – some features will be on in many samples, whereas some will be off in all samples. The actual number of PSN features for our training set was 4492.

So, counting all types of features, we have extracted a total of $8348 + 6000 + 4492 = 18840$ features. Our combined feature space thus can be modeled as a version of Chou’s PseAAC as described below. For $1 \leq u \leq 20$, we have the amino acid composition in the feature vector. From $21 \leq u \leq 8348$, the dipeptide and tripeptide compositions are represented. From $8349 \leq u \leq 14348$, the features in this vector comes from the nGDip feature space. Finally, the PSN features construct the remaining portion of the PseAAC, from $14349 \leq u \leq 18840 = \Omega$.

3.2.3 Prediction Algorithm

It has been observed in the literature that there is similarity in Amino Acid Composition (AAC) [58] among *cis*-Golgi proteins and *trans*-Golgi proteins [288]. Thus, traditional computational methods using Basic Local Alignment Search Tool (BLAST) [15] is inefficient to distinguish between *cis*-Golgi and *trans*-Golgi proteins. Machine learning methods can be a wise alternative option, which we have pursued in this chapter.

A diagram of our model construction workflow has been shown in Figure 3.3. We process all sub-Golgi protein sequences through isGPT feature construction step. In this step, several position independent and position specific features are extracted from the training dataset. All the features are obtained directly from the sequences. Among the position independent features are n -grams and n -gapped dipeptide based features, which have widely been used in literature. All these features are combined together to make a hybrid feature space. As we already know that the benchmark dataset is significantly imbalanced, we conduct Synthetic Minority Oversampling Technique (SMOTE) [43] following previous methods to make a balanced dataset. In the feature selection step, features are ranked based on random forests based importance score and only a subset of the top ranked features are selected based on 10-fold cross validation performance. Finally, Support Vector Machine (SVM) is applied on the selected features to compute the final predictor.

As mentioned in Section 3.1, we have modeled the problem of sub-Golgi protein localization as a classification as well as a regression problem. The problem is inherently a binary classification problem where *cis*-Golgi represents the minority class and *trans*-Golgi represents the majority class. To model it as a regression problem, we give each *cis*-Golgi protein a score of 1 while each *trans*-Golgi protein gets a score of 0. SVM is then applied in regression mode, which is also known as Support Vector Regression (SVR). Finally a class discriminating threshold is identified to optimize the classification performance. The threshold can be tuned to increase the sensitivity or specificity to the desired level. This is where the benefit lies in modeling it as a regression problem.

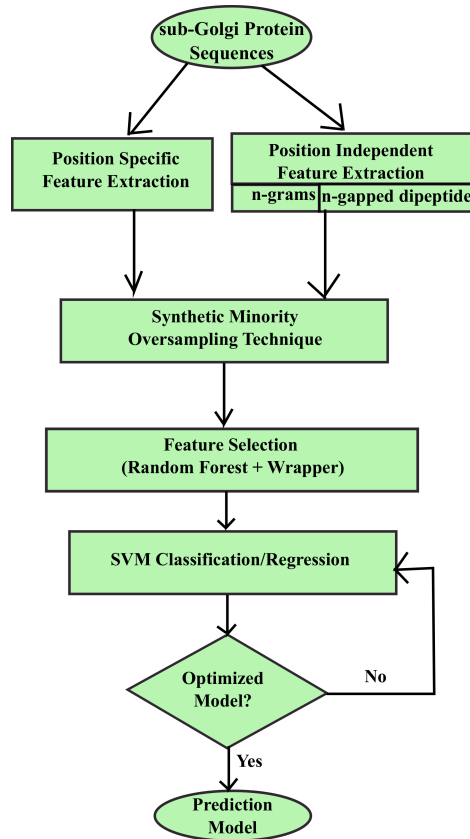


Figure 3.3: isGPT model construction. In the first step, position specific and position independent features are extracted from the sub-Golgi protein sequences. Among the position independent features are n -grams and n -gapped dipeptide based features. The whole dataset, along with extracted features, then goes through Synthetic Minority Oversampling Technique (SMOTE) to make a balanced dataset. To optimize the model, features are selected based on random forests based importance score followed by a wrapper method. Finally, SVM classification or regression method is applied on the selected features to compute the final predictor.

Feature Selection Technique

As mentioned in Section 3.2.2, we extracted a total of 18840 features to represent the protein sequences. It will be computationally expensive to work with such a large feature vector, both during the learning phase as well as in the prediction phase. Besides, not all features may always be effective in the learning model [117, 242]. As such, we need to

select a set of relevant features that will contribute to the learning model in improving accuracy. The feature selection step has been applied in previous studies of sub-Golgi protein type identification as well. For example, Yang et al. [288] used random Forests-Recursive Feature Elimination (RF-RFE) which is a wrapper method. Ahmad et al. [10], on the other hand, applied a filtering approach, using the Fisher selection technique. In this chapter, we have employed a composition of filter and wrapper approaches.

In the filtering phase, we apply random forests algorithm on the entire feature set to generate a model. Through this model creation, the random forests algorithm is able to set an importance score (*mean decrease in accuracy*) to each of the input features. This importance score indicates the global importance over all out-of-bag cross validated predictions and is very robust as it is averaged over all predictions for a given feature variable. The importance score is used to rank the features and subsequently filter out irrelevant features.

In Figure 3.4, we see that when we take all the features into account, the summation of importance score for *n*-Gapped-Dipeptides (nGDip) based features is quite high. However, for the *n*-grams based features as well as position specific *n*-grams (PSN) this sum is in fact negative. Overall, only the top 2985 features have positive importance score. From feature 2986 up to feature 15980, the importance score of each feature is 0. Beyond that, the scores actually become negative. Therefore, we further examine the top 3000 features. When we select this feature subset, the total importance scores for all three feature types are positive. Among these 3000 features, there are 2105 nGDip features, 52 PSNs and 843 *n*-grams.

Subsequently we apply the wrapper phase. Instead of directly selecting the 3000 features, we further experimented with the top 3500, 3000, 2500, 2000 and 1500 features by training SVM regression models on the benchmark dataset as well as on the dataset balanced with SMOTE. In Figure 3.5, we have reported the Receiver Operating Characteristics (ROC) curves from these experiments, as obtained using 10-fold cross validation. Since the dataset is imbalanced, ROC-Curve alone is not able to identify the significance of

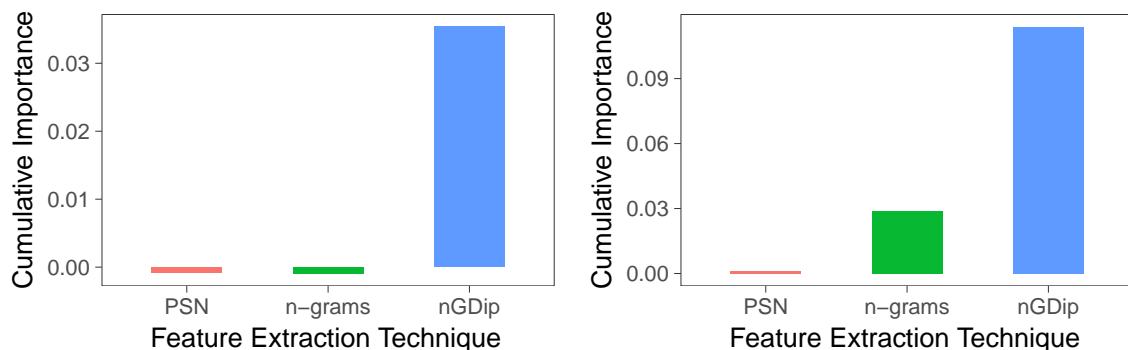


Figure 3.4: Categorized feature importance based on *mean decrease in accuracy*, as obtained from random forests model. The aggregate mean decrease in accuracy is better for top 3000 features (right) compared to all features (left). PSN: Position Specific n -grams, n -grams: Combination of AAC, dipeptide and tripeptide composition features, nGDip: n -Gapped Dipeptides.

selected features. In fact it has been argued in the literature that for imbalanced datasets, Precision Recall Curve (PR-Curve) is of more significance than ROC-Curve [73]. Thus, we also report PR-Curve in Figure 3.5.

The closer the ROC curve is to the top-left corner of the graph, the better is the performance of the model. On the other hand, the PR curve should be as close to the top-right corner as possible. Therefore, from the curves of Figure 3.5, it is clear that the performance with 3500 or 3000 features is much inferior to the other feature subsets. The curves further demonstrate the importance of balancing the dataset. Perhaps a better articulation of these points are in Table 3.1, where the area under the ROC and PR curves for the different settings are recorded.

Subsequently, we have further examined the feature space comprising the top-ranked 1500 to 2800 features. First, we ran SVM with 10-fold cross validation using the top 2800 features. The C parameter of the regularization term in the *Lagrange formulation* of SVM was varied from the set $\{0.3, 1, 3, 10, 30, 100\}$. Thus 6 different models were constructed and we evaluated their performances. Then, from the feature set, we eliminated the

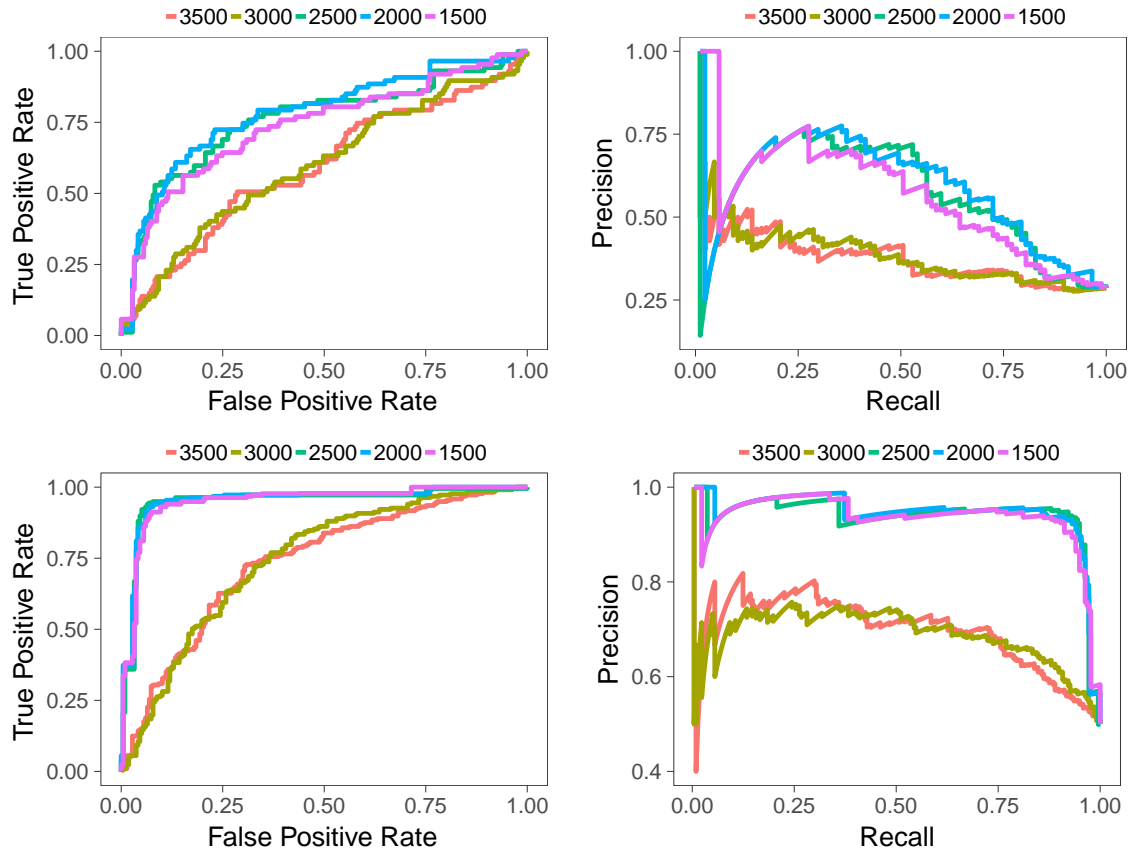


Figure 3.5: ROC-Curves (left) and PR-Curves (right): The curves are generated by regression analysis with 10-fold cross validation, using top 3500, 3000, 2500, 2000 and 1500 features, respectively. The benchmark (imbalanced) dataset was used to generate the top curves. For the bottom curves, the dataset was balanced using SMOTE.

Table 3.1: Area under ROC and PR curves for different number of top-ranked features selected.

Number of Features	Without SMOTE		With SMOTE	
	auROC	auPR	auROC	auPR
3500	0.55	0.33	0.73	0.68
3000	0.59	0.37	0.74	0.68
2500	0.75	0.53	0.95	0.95
2000	0.78	0.60	0.95	0.95
1500	0.75	0.57	0.95	0.95

least ranked 50 features, recomputed 6 more models in the same way and measured their performances. We repeated this process until the feature subset size was reduced to 1500. Thus a total of 162 models were evaluated. We finally selected the combination of C and feature subset that yielded the best performance. This wrapping step was applied independently both in classification and regression analysis with the native (imbalanced) dataset as well as with the dataset balanced with SMOTE.

3.2.4 Predictor Evaluation

To objectively measure the performance of isGPT, we have applied jackknife cross-validation, 10-fold cross-validation and independent testing. These methods have already been briefly described in Section 2.1.8. As performance metrics, we have used accuracy, sensitivity, specificity and Matthew's Correlation Coefficient (MCC). We have also analyzed the area under ROC curve (auROC) and PR curve (auPR). The performance metrics and curves have been described in Section 2.1.9.

For parameter tuning, van Dijk et al. [269] used a nested cross validation setup to avoid optimistic bias in the cross validation performance. However, recent works in sub-Golgi protein localization have not performed cross validation nesting. Since we compare our results with the recent methods, we have accordingly avoided nested cross validation. The concern of optimistic bias in cross validation is mitigated by measuring our performance on the independent dataset as well. The performance results, reported later in this chapter, indicate that our cross validation performance generalizes well in the independent testing.

We have conducted experiments using R language (version 3.2.1) on three different machines with the following configurations:

- A Desktop computer with Intel Core i3 CPU @ 1.90GHz x 4, Ubuntu 15.10 64-bit OS and 4 GB RAM.
- A Desktop computer with Intel Core i7 CPU @ 3.30GHz x 4, Windows 7, 64-bit OS and 8 GB RAM.

- A server machine with Intel Xeon CPU E5-4617 0 @ 2.90GHz x 6, Ubuntu 13.04 64-bit OS, 15 MB L3 cache and 64 GB RAM.

To construct the isGPT model, we have used random forests and SVM machine learning algorithms. These are available respectively from R packages *randomForest* and *e1071*. In the random forests algorithm, we have used the default parameters setting. In particular, the number of trees (*ntree*) was restricted to 500, while the number of variables tried at each split (*mtry*) was set to square root of the total number of features.

As discussed earlier, random forests model has been used for feature selection, while SVM is used to learn the model. Since our training set is relatively small, we have used linear kernel function in SVM to avoid overfitting. The *cost* parameter was varied as described in Section 3.2.3. Data were scaled internally to zero mean and unit variance as per the default behavior of the SVM implementation in *e1071* package.

All codes have been written in R language where we have used some available R packages. We have also used *ROCR* and *pracma* R packages for performance evaluation of our model. For balancing the dataset, we used an implementation of SMOTE from *Weka 3 Data Mining Software* [104, 120].

3.2.5 Predictor Availability

isGPT is freely available as an R script at <https://github.com/srautonu/isGPT>. Additionally, we have established a publicly accessible web server at <http://isgpt.research.buet.ac.bd/> to facilitate wide adoption.

3.3 Results

In this section, we describe several experiments and analyze their results. We have compared results from both the regression analysis and the binary classification. We have also compared the results of our proposed technique with state-of-the-art methods.

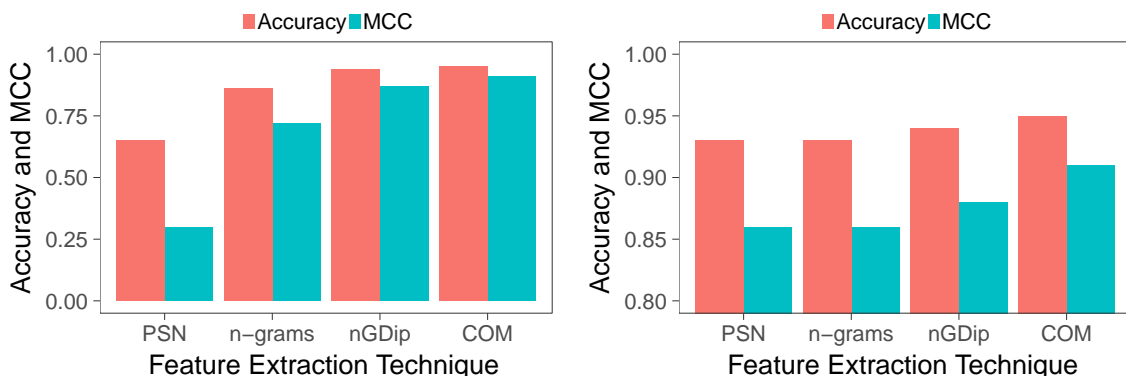


Figure 3.6: Accuracy and Matthew’s Correlation Coefficient (MCC) of different feature extraction techniques. The results are obtained from 10-fold cross validation of SVM regression model trained on the benchmark dataset balanced using SMOTE. PSN: Position Specific n -grams, n -grams: Combination of AAC, dipeptide and tripeptide composition. nGDip: n -Gapped-Dipeptides. COM: Combination of all the feature extraction techniques. The left figure is generated using the features of specific feature space that are within top 2500 positions in the combined space. In the right figure, for each feature space, corresponding top 2500 features are used.

3.3.1 Impact of Feature Extraction Techniques

To analyze the efficacy of the different feature extraction techniques, we take a closer look at the top 2500 features. In this subset, there are 1772 nGDip features, 34 PSN features and 694 n -grams features. With the SMOTE-balanced dataset, we trained three different SVM regression models using each of these three subsets of features. In another model, we trained with all the 2500 features. In Figure 3.6, the accuracy and MCC values from these four models are compared in the left side graph. The nGDip feature extraction technique is a clear winner over the other two, while the combination of all performs slightly better than that.

Note, however, that in the above comparison, the size of the feature vectors was widely different. Therefore, we conducted another experiment where we trained 3 different SVM regression models using top 2500 features of the 3 individual feature extraction techniques.

Table 3.2: Comparison of classification and regression models of isGPT. In the *Type* column, ‘C’ and ‘R’ are used to represent classification and regression respectively. The ‘w/ S’ prefix is added if the model was computed on the dataset balanced with SMOTE. Acc: Accuracy, Sn: Sensitivity, Sp: Specificity, MCC: Matthew’s Correlation Coefficient.

Type	10-fold Cross-Validation				Jackknife Cross-Validation				Independent Test			
	Acc	Sn	Sp	MCC	Acc	Sn	Sp	MCC	Acc	Sn	Sp	MCC
C w/ S	94.7	95.9	93.6	0.89	94.9	95.9	94.0	0.90	93.8	69.2	100	0.80
C	80.3	46.0	94.0	0.48	80.6	48.3	93.6	0.49	95.3	76.9	100	0.85
R w/ S	95.4	95.4	95.4	0.91	95.9	95.9	95.9	0.92	95.3	84.6	98.0	0.85
R	80.9	48.2	94.0	0.50	81.9	56.3	92.2	0.53	92.2	76.9	96.1	0.75

We compare the performance of these models to the combined model in the right side graph of Figure 3.6. The superiority of combined feature space over the individual feature spaces hold in this setting as well. PSN, n -grams and nGDip feature extraction techniques individually achieve accuracy values of 93%, 93% and 94%, respectively. When the combined feature space is used instead, the accuracy increases to 95%. Similarly the MCC increases from respective individual values of 0.86, 0.88, 0.86 to 0.91 for the combined feature space.

3.3.2 Impact of Data Imbalance in isGPT Learning Model

As mentioned earlier, the benchmark dataset is highly imbalanced. Both Yang et al. [288] and Ahmad et al. [10] reported that the dataset, balanced using Synthetic Minority Over Sampling Technique (SMOTE), performs better than the imbalanced dataset to classify the sub-Golgi proteins. To be consistent with their approach, we too have applied SMOTE to balance the data by increasing the number of *Cis*-Golgi data points to 217. To examine the impact of balancing, we have conducted experiments both before and after balancing and then compared the results. In both cases, we have run regression as well as classification models. As discussed earlier, we examined 162 different models in each experiment by varying the regularization parameter C and the feature vector

Table 3.3: Optimal parameters for classification and regression models of isGPT, based on 10-fold and jackknife cross-validation results. In the *Model Type* column, ‘C’ and ‘R’ are used to represent classification and regression respectively. The ‘w/ S’ prefix is added if the model was computed on the dataset balanced with SMOTE

Model Type	Number of Features	C	Threshold
C w/ S	2800	10	N/A
C	2050	1	N/A
R w/ S	2800	1	0.58
R	2250	10	0.44

size and measured performance using 10-fold cross validation. The models yielding the best accuracy were further validated using Jackknife cross validation. Subsequently, the best models, as determined by the jackknife accuracy, were applied to the separate test dataset for independent testing. In Table 3.2, we have summarized the results from our experiments. We have highlighted the best results in bold faced fonts. The impact of data balancing is clearly evident. Also, we see that the regression model performs better than the classification model. So, we have subsequently compared the results from the regression model on the SMOTE-balanced dataset with previous studies.

In Table 3.3, we have recorded the optimal parameters for each model. These include the number of features and the C constant of the regularization term in SVM. For regression models, the class discriminating threshold is also recorded. These values are obtained based on the 10-fold and jackknife cross validation results.

3.3.3 Comparison between isGPT and Existing Techniques

In Table 3.4, we have compared the performance of isGPT regression model with previous methods. The results reported for isGPT as well as for Yang et al. [288] and Ahmad et al. [10] are for the same benchmark dataset, after balancing was performed using SMOTE. The work of Ding et al. [79, 80] uses an earlier dataset of smaller size. We have reported results for jackknife cross-validation, independent test and 10-fold cross-validation, with

Table 3.4: Comparison of isGPT regression model with previous methods. Acc: Accuracy, Sn: Sensitivity, Sp: Specificity, MCC: Matthew’s Correlation Coefficient.

Tools	Jackknife Cross-Validation				Independent Testing				10-fold Cross-Validation			
	Acc	Sn	Sp	MCC	Acc	Sn	Sp	MCC	Acc	Sn	Sp	MCC
[80]	74.7	69.6	79.6	0.52	-	-	-	-	-	-	-	-
[79]	85.4	73.8	90.5	0.65	85.9	69.2	90.2	0.58	-	-	-	-
[288]	88.5	88.9	88.0	0.76	93.8	92.3	94.1	0.82	90.1	90.8	89.4	0.80
[10]	94.9	97.2	92.6	0.90	94.8	94.0	93.9	0.86	94.9	97.2	92.6	0.90
isGPT	95.9	95.9	95.9	0.92	95.3	84.6	98.0	0.85	95.4	95.4	95.4	0.91

the best results highlighted in bold faced fonts. 10-fold cross-validation results are absent in [79]; both independent testing results and 10-fold cross-validation results are absent in [80]. As such, we have marked the corresponding cells in the table by ‘-’ symbol.

We see that isGPT achieves an accuracy of 95.9%, 95.3% and 95.4% for jackknife cross-validation, independent testing and 10-fold cross-validation, respectively. In comparison, the previous best method (Ahmad et al. [10]) respectively achieved an accuracy of 94.9%, 94.8% and 94.9%. So, in all cases, isGPT shows improved performance. In terms of MCC, isGPT demonstrates superiority in jackknife and 10-fold cross-validation - compare isGPT’s respective scores of 0.92 and 0.91 to the previous best: 0.90 and 0.90. In case of independent testing, the MCC score of isGPT is slightly behind than that of [10]. However, we believe that the latter value might have been erroneously reported. This is elaborated in the ‘Discussion’ section. Overall, it is evident that isGPT performs better than all existing methods.

3.4 Discussion

In this section, we briefly discuss results we have obtained, previous results as well as key differences between our work and state-of-the-art methods. Table 3.5 compares isGPT with earlier works in terms of the different steps taken in building the prediction model. The novelty in isGPT lies in the addition of tripeptide composition and PSN features; and in the use of combination of random forests and SVM for feature selection.

Table 3.5: Comparison of of different steps in model building in *isGPT* vs. prior art.

<i>Tools</i>	<i>Benchmark Dataset</i>	<i>Feature Extraction Technique(s)</i>	<i>Feature Selection</i>	<i>Classifier</i>
van Dijk et al. [269]	[269]	Amino acid grouping String-based triads 3D-structure based triads	No	SVM (Linear)
Ding et al. [80]	[80]	Amino acid composition Gapped dipeptide composition Increment of Shannon entropy	No	Modified MD
Ding et al. [79]	[79]	Gapped dipeptide composition	ANOVA (83 features)	SVM (RBF)
Jiao et al. [140]	[79]	Position Specific Physico-Chemical Properties (PSPCP)	ANOVA	SVM (RBF)
Jiao et al. [141]	[79]	Position Specific Physico-Chemical Properties (PSPCP)	mRMR	SVM (RBF)
Yang et al. [288]	[288] SMOTE	Common Spatial Patterns (CSP) PSSM-Dipeptide Composition Bi-gram PSSM Evolutionary Difference-PSSM Gapped dipeptide composition	RF-RFE (55 features)	Random forests
Ahmad et al. [10]	[288] SMOTE	Dipeptide composition Split-PseAAC Bi-gram PSSM	Fisher (83 features)	KNN
isGPT	[288] SMOTE	Amino acid composition Dipeptide composition Tripeptide composition Gapped dipeptide composition Position specific n -grams	Random forests filter SVM wrapper (2800 features)	SVM (Linear) SVR (Linear)

3.4.1 Consistency Check of Earlier Results

During our comparative analysis with earlier works, we attempted to check the consistency of earlier results. As we know, the independent dataset has 13 *cis*-Golgi and 51 *trans*-

Golgi proteins. Since *cis*-Golgi class has lesser data, conventionally it should be the positive class in a binary classification model. Therefore, using the symbols introduced in Section 2.1.9, $P = 13$ and $N = 51$. From the accuracy, sensitivity and specificity values, we can now compute the TP, TN in the earlier works, using Equation 2.3.

From the data reported by Yang et al. [288], since sensitivity = TP/P , we find that $TP = 11.99 \approx 12$. Therefore, $FP = 1$. Similarly, from the specificity data, we can find, $TN = 47.99 \approx 48$. From accuracy data, we can find that $TP + TN = 60$, which is consistent with the already obtained values of TP and TN . Now, we can further compute that $FN = P - TP = 1$ and $FP = N - TN = 3$. Plugging these values into MCC equation gives us 0.82. So, Yang et al.'s data is consistent.

Now, let us complete the same exercise for the results reported by Ahmad et al. [10]. From the sensitivity data, we can find that $TP = 12.22 \approx 12$. If we accept it to be 12 then the sensitivity should have been 92.3, not 94. Similarly, from the specificity data, we can find that $TN = 47.8948 \approx 48$. If we take it as 48 then specificity should have been 94.1, not 93.9. So, in both cases we find some inconsistency. Perhaps, Ahmad et al. took *trans*-Golgi to be the positive class. In that case, we should have $P = 51$ and $N = 13$. We can then calculate $TP = 47.94 \approx 48$ and $TN = 12.21 \approx 12$. Like before, the rounding off error seems too high. In both scenarios, plugging the values into the MCC equation yields, 0.82. But, the value reported in the paper is 0.86. Thus, some inconsistency has been introduced in the reported data of Ahmad et al. In fact, they made another minor reporting error: in their paper the data from [80] and [79] have been swapped.

3.4.2 Choice of Class Discriminating Threshold in isGPT

Now onto a discussion about the class discriminating threshold in the isGPT regression mode. In the regression model, trained with the imbalanced dataset, the accuracy and MCC values in independent testing is the best when the threshold is between 0.41 to 0.47. The optimal threshold (0.44), as chosen by the cross validation methods, does fall in this range. This is not the case in the regression model trained with the SMOTE-

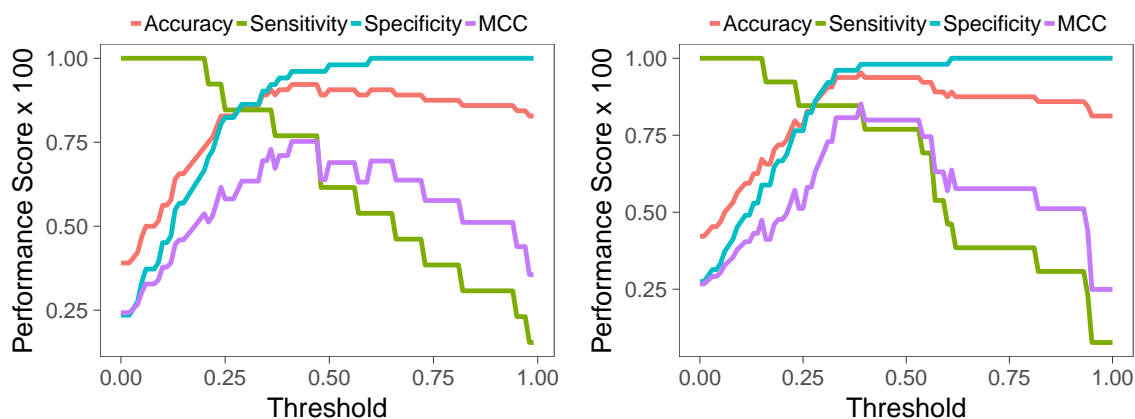


Figure 3.7: Response of different performance metrics against variation of class discriminating threshold. The measurements are done on independent testing using a regression model trained on the imbalanced dataset (**Left**) as well as the SMOTE-balanced dataset (**Right**).

balanced dataset. In this case, the 0.58 threshold did not yield the best performance in the independent testing. Instead, we had to set the threshold to 0.40.

To better analyze the impact of the threshold, the response of different performance metrics in the independent testing while changing the threshold has been plotted in Figure 3.7. The Left side graph therein does confirm that the thresholds in the range 0.41 to 0.47 produce the maximum MCC as well as accuracy for the independent testing in case of the regression model trained using imbalanced benchmark dataset. The right side graph is plotted using a model trained with the SMOTE-balanced dataset. In this case, we observe a good range of threshold between 0.33 to 0.53, where both MCC and accuracy values are very high, with a peak MCC observed for the threshold of 0.39. While the peak value (0.85) is very satisfactory, in the remaining parts of this plateau, MCC remains competitive, between 0.80 to 0.81. Therefore, in our final predictor, we have set a default threshold of 0.50. The 10-fold cross validation with this threshold yields an accuracy of 94.7% and MCC of 0.90 which are competitive with state-of-the-art predictors. Biologists using isGPT can tweak the threshold to further meet their experimental goal (i.e. increased sensitivity or specificity).

3.4.3 Large Feature Space Independent of PSSM

Finally, we discuss a key point that distinguishes our work from the state-of-the-art methods. We explored a large feature space, comprising 18840 features and then selected 2800 features for training the model. The size of the selected feature set is way higher than earlier studies. Ahmad et al. [10] used 83 dimensional feature vector, while Yang et al. [288] selected 55 features. However, it is important to note that both of the above mentioned methods use features derived from the Position Specific Scoring Matrix (PSSM). The PSSM can be computed from PSI-BLAST [15] by searching the non-redundant protein database using at least three iterations. As such, this is a time consuming step. Our approach, on the other hand, can extract all the necessary features from a target protein in a single pass along the sequence and then use the classifier to predict its class. On the server machine with Intel Xeon CPU E5-4617 0 @ 2.90GHz x 6, 64 GB RAM, PSSM generation for the smallest sequence (116 residues) in the test set (Accession Id: O95183) took around 10 minutes 30 seconds. For the largest sequence (Accession Id: Q55EI3, 4241 residues), almost 28 minutes were needed. In contrast, isGPT completed the prediction for the entire test set in less than two and a half minutes.

Besides, the PSSM based representation of protein is heavily dependent on the database being searched for homology information. If the target protein does not have enough homologous sequences in the database, then the generated PSSM cannot describe the protein well. Therefore, any prediction model dependent on PSSM information will produce wrong predictions in such a case [164]. The work by Ahmad et al. and Yang et al. are susceptible to this issue. isGPT, on the other hand, is completely resilient to it as it does not have any dependence on PSSM.

3.5 Conclusion

In this chapter, we present isGPT, an optimized model to identify sub-Golgi protein types. As the training dataset is significantly imbalanced, we use SMOTE to balance

the dataset. We apply a combination of sequence based feature extraction techniques followed by a random forests based novel feature selection technique. Finally, Support Vector Machine (SVM) is employed to train a prediction model that can distinguish between *trans*-Golgi and *cis*-Golgi proteins. Our approach outperforms state-of-the-art techniques according to different performance metrics. Our predictor is available as an R script that can readily be applied to target protein sequences, without dependency on any other services or pre-processing (e.g. computation of PSSM). Since user-friendly and publicly accessible web-servers represent the future direction for developing practically more useful models [115, 162, 167, 186, 266, 293], we have thus made isGPT available as a web based predictor. We hope the appeal of isGPT, in its simple model and ease of access, will attract biologists in applying this new predictor in their relevant research projects.

In the next chapter, *DNA-binding protein prediction*, we build a classification model to predict whether a given protein sequence would bind to a DNA or not. Like isGPT, our work for DNA-binding protein prediction too puts emphasis on sequence based features for protein sample representation.

Chapter 4

DNA-binding Protein Prediction

A DNA-binding protein (DNA-BP) is a protein that can bind and interact with a DNA. DNA-BPs regulate and effect various cellular processes like transcription, DNA replication, recombination, repair and modification. As such, these proteins can potentially be used for drug development in treating genetic diseases and cancers. This is why identification DNA-BPs is a very important task. As the experimental methods of this important task are expensive as well as time consuming, fast and accurate computational methods are sought for predicting whether a protein can bind with a DNA or not. In this chapter, we focus on building a new computational model to identify DNA-binding proteins in an efficient and accurate way. Our model extracts meaningful information directly from the protein sequences, without any dependence on functional domain or structural information. After feature extraction, we have employed random forests model to rank the features. Afterwards, we have used Recursive Feature Elimination (RFE) method to extract an optimal set of features and trained a prediction model using Support Vector Machine (SVM) with linear kernel. Our proposed method, named as *DNA-binding*

Much of the material in this chapter is taken without alteration from the following paper.

- Rahman, M. S., Shatabda, S., Saha, S., Kaykobad, M., & Rahman, M. S. (2018). *DPP-PseAAC: A DNA-binding protein prediction model using Chou's general PseAAC*. Journal of theoretical biology, 452, 22–34.

Protein Prediction model using Chou's general PseAAC (DPP-PseAAC), demonstrates superior performance compared to the state-of-the-art predictors on standard benchmark dataset. DPP-PseAAC achieves accuracy values of 93.21%, 95.91% and 77.42% for 10-fold cross-validation test, jackknife test and independent test respectively. The source code of DPP-PseAAC, along with relevant dataset and detailed experimental results, can be found at <https://github.com/srautonu/DNABinding>. A publicly accessible web interface has also been established at: <http://dpp-pseaac.research.buet.ac.bd>.

4.1 Introduction

A DNA-binding protein (DNA-BP) is a protein that can bind and interact with a DNA. Such a protein is composed of DNA binding domains that include transcription factors, nucleases and histones. The transcription factors modulate the process of transcription, while the nucleases can cleave DNA molecules. Histones, on the other hand, are involved in chromosome packaging in the cell nuclei. Figure 4.1 shows examples of protein DNA binding interactions: in the left figure, a transcription factor is bound to a DNA, while in the right figure, the restriction enzyme EcoRV is interacting with its target DNA.

The DNA-BPs thus perform two main functions: firstly, they organize and compact the DNA and secondly, they regulate and affect various cellular processes like transcription, DNA replication, recombination, repair and modification. Therefore, the DNA-BPs can potentially be used for drug development in treating genetic diseases and cancers [116,161]. This is why developing efficient and highly accurate methods to identify DNA-BPs is a very important research problem in the field of molecular biology.

Traditionally, the DNA-BPs have been identified through different experimental methods. These include filter binding assays [126], genetic analysis [105], X-ray crystallography [57], chromatin immunoprecipitation on microarrays [35] etc. However, these experimental methods are costly and time consuming. On the contrary, the number of sequence-known proteins has grown exponentially in recent years due to the rapid devel-

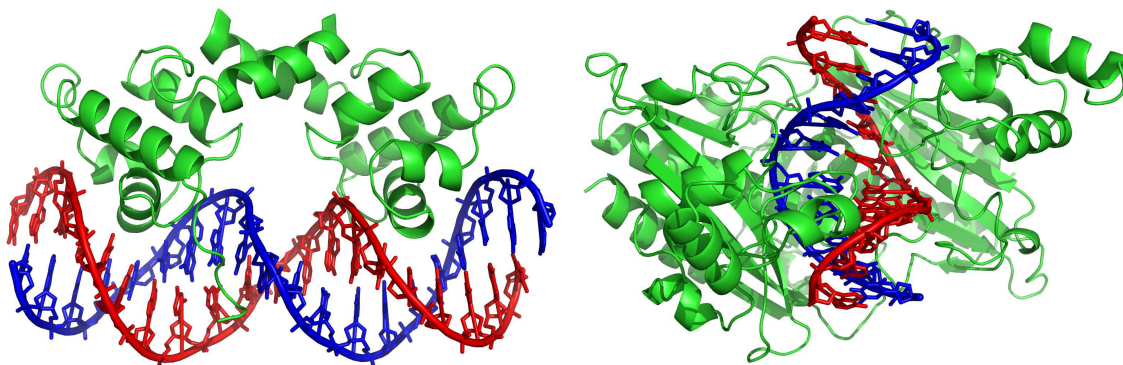


Figure 4.1: DNA-binding proteins bound to respective target DNAs. **(Left)** The lambda repressor helix-turn-helix transcription factor bound to its DNA target. Created from PDB 1LMB. Image source: [1]. **(Right)** The restriction enzyme EcoRV in a complex with its substrate DNA. Created from PDB 1RVA. Image source: [4].

opment of fast sequencing technologies. To catch up, researchers have started to rely on computational methods to identify DNA-binding proteins. These methods can largely be categorized into two groups: structure based methods and sequence based methods.

Structure-based methods depend on the structural information of the protein sequences. These include high-resolution 3D structure, accessible surface area, torsion angles, structure motifs etc. Stawiski et al. [256] did the pioneering work in identifying DNA-BPs using structural information. They extracted 12 parameters from the detailed atomic structure of the protein. The calculation of these parameters requires analysis of electrostatic patches, surface clefts and conservation analysis of the sequence. A three-layer artificial neural network (ANN) was used for the classification task. Ahmad et al. [11], on the other hand, used a two-layer neural network with parameters calculated solely from bulk electrostatic properties.

Szilágyi et al. [261] subsequently identified a flaw in the way Ahmad et al. constructed their dataset. They also proposed a fast and efficient method to predict DNA-BPs from only the amino acid sequences and low-resolution, C^α -only protein models. Available as a web based predictor called DNABIND, their predictor uses logistic regression (LR) as the classifier, with only 10 features, calculated from proportion of certain amino acid residues,

spatial asymmetry of certain other residues and dipole moment of the entire molecule. Gao et al. [106] proposed another structure based predictor, *DBD-Hunter*, that applies structural alignment and evaluation of a statistical potential to identify DNA-BPs. In *DBD-Hunter*, first, the target structure is matched against a template library of DNA-protein complex structures for structural similarity. For templates with matching scores better than a threshold, the statistical potential energy between the target protein and the template DNA is calculated by evaluating contacts within the structurally aligned regions. Gao et al. [107] subsequently proposed another predictor, *DBD-Threader*, for the prediction of DNA-binding domains and associated DNA-binding protein residues. While this method also uses a template library composed of DNA-protein complex structures, it requires only the target protein's sequence for its classification. This independence from structural information makes the predictor very useful, while its performance remains comparable with *DBD-Hunter*.

Examples of other structure-based methods can be found in [213,260,294,296]. However, structure-based predictors are applicable only when the structural information of a candidate protein is known. While the post-genomic era witnesses a rapid growth in sequence known proteins, the structure of many of these proteins still remain undiscovered. The predictors that solely rely on structural information of proteins are thus limited in their use. Sequence based methods, on the other hand, attempt to identify the DNA-BPs from the amino acid sequence by extracting various discriminating features. Some predictors may additionally rely on some structural features for improved prediction accuracy when the protein structure is known. Examples of prominent sequence based predictors of DNA-BPs can be found in [65,82,93,135,154,155,168,174,176,191,203,206,219,245,251,275,277,285,286,295,297].

Kumar et al. [155] used evolutionary information from the Position Specific Scoring Matrix (PSSM) for protein representation. The PSSM profile of each protein was generated from PSI-BLAST [15] by searching the non-redundant (nr) protein database using three iterations with e-value cutoff set to 0.001. They calculated the probability of occurrences of each type of amino acid corresponding to each type of amino acid in the protein

sequence. From each column of the PSSM, 20 features are thus generated. Deemed as *PSSM-400*, this feature scheme generates a total of $20 \times 20 = 400$ features. They then applied Support Vector Machine (SVM) [32] as the learner. Their predictor is available as a webtool, *DNAbinder*. In addition to comparing their predictor with prior art, they demonstrated the effectiveness of the PSSM based features over amino acid composition, di-peptide composition and 4-parts amino acid compositions.

The performance of *DNAbinder* depends on the quality of PSSM profiles, which is heavily dependent on the database being searched for homology information. To eliminate this dependency, *DNA-Prot* was proposed by another group Kumar et al. [154]. This predictor used features such as frequency of amino acid residues and groups, predicted secondary structure (PredSS) information from PSIPRED [194], physico-chemical properties from AAIndex database [146]. They also used sliding 10 residue windows in the protein sequence to represent short peptides and calculated the composition of hydrophobic, hydrophilic and neutral amino acid rich peptides. The total number of features was 116. Subsequently correlation-based feature subset selection method (CFSS) was applied to finally select a subset of 20 features. Finally, random forests [34] was applied as the learner.

Lin et al. [168] incorporated the Grey model [143] parameters in the general form of Chou's PseAAC [61] for protein sequence representation. They then trained their model, *iDNA-Prot*, using random forests algorithm. Lou et al. [191] introduced a predictor called *DBPPred*, where amino acid composition, PSSM scores, PredSS and predicted relative solvent accessibility (PredRSA) were used as features. The PredRSA and PredSS features were derived by SPINE-X program [94]. They then used random forests algorithm to rank the features and applied wrapper based feature selection based on the best-first forward search strategy. They used Gaussian Naïve Bayes (GNB) as the final classifier. Notably, they also achieved good performance using SVM with the Radial Basis Function (RBF) kernel. However, GNB was finally chosen due to its simplicity. They compared their predictor with prior ones using an independent dataset called PDB186, comprising equal number of DNA-binding and non DNA-binding proteins. This dataset has subsequently been used in performance evaluation of many other predictors.

Liu et al. [177] used amino acid distance-pair coupling information into Chou's general form of PseAAC [61]. To reduce the dimension of the feature vector and to speed up the prediction process, they also used amino acid reduced alphabet profile [222]. They then applied SVM with RBF kernel on 602 features to produce the prediction tool called *iDNA-Prot|dis*. To train and assess their predictor using cross-validation, they prepared a stringent balanced dataset of 1075 protein samples. This benchmark dataset has subsequently been referred to as PDB1075 and has been widely used in literature for cross-validation. We have also used this dataset in our work and provide a detailed description of the dataset later in this chapter. In addition to preparation of the benchmark dataset, a key contribution of Liu et al.'s work was re-implementation of major earlier predictors and measuring their cross-validation performance using this benchmark dataset. This paved the way for subsequent predictors to be compared with prior art in an apple for apple comparison.

In 2015, Liu et al. [174] presented another predictor called *iDNAPro-PseAAC*. They used profile-based representation of the protein sequence and then used PseAAC with the 3rd order sequence-order effect. Their predictor thus uses a total of 23 features. Their model was trained using SVM with RBF kernel. Dong et al. [82] used Auto-Cross Covariance (ACC) transformation with amino acid k-mer compositions and physicochemical properties. They then used SVM to train the predictor, widely known as *Kmer1 + ACC*. Liu et al. [176] proposed yet another predictor called *PseDNA-Pro*. It uses overall amino acid composition (OAAC), pseudo amino acid composition (PseAAC) and physicochemical distance transformation (PDT) based features for protein representation. The predictor was trained using SVM with RBF kernel.

Waris et al. [275] employed feature extraction techniques such as dipeptide composition (DPC), split amino acid composition (SAAC) and PSSM. They experimented with these techniques independently as well as in combination. As learners, they utilized K -nearest neighbor (KNN), probability neural network, SVM and random forests. The best results were obtained using PSSM and SVM.

Wei et al. proposed *Local-DPP* [277], where local pseudo position specific scoring matrix (Local Pse-PSSM) features have been used. The locally conserved protein information is captured by fragmenting the PSSMs into several equally sized sub-PSSMs. The local features are then computed from each sub-PSSM. Finally, all the local features are combined to form the final feature vector. Random forests algorithm is then used to learn the model.

Very recently, Chowdhury et al. [65] developed iDNAProt-ES, that utilizes both the evolutionary profile and structure information of proteins to identify their DNA-binding functionality. From the PSSM profile, they extracted features like amino acid composition [58], Dubchak features [87], bigram, auto-covariance, segmented distribution etc. To extract structural features, they used SPIDER2 [289], a freely available software that provides information on accessible surface area, torsion angles, structure motifs in each amino acid residue position. From this information, they extracted features like secondary structure composition and occurrence, accessible surface area composition, torsional angles bigram and auto-covariance, structural probabilities bigram and auto-covariance etc. They subsequently used recursive feature elimination to extract an optimal set of features and used SVM with linear kernel to learn the model. Their proposed method significantly outperforms the existing state-of-the-art predictors on standard benchmark dataset in cross-validation testing.

While significant amount of work has been done in this field, there is still room for improvement in different ways. Firstly, the prediction performance could be improved further. Secondly, many of the existing predictors use feature extraction techniques that are time consuming, some use sophisticated prediction models. In this chapter, we therefore propose a DNA binding protein predictor that extracts features from the protein sequence alone, that has a fast and simple prediction model and that outperforms the existing predictors.

We have followed Chou's 5-step procedure [61] for establishing our predictor. As briefly described in Section 2.1.1, the steps in this process include dataset preparation, construc-

tion of features from protein sequences, applying a powerful classification algorithm, objectively evaluating the predictor and finally making the predictor widely available. We have collected a benchmark dataset from literature and then applied general formulation of Chou’s PseAAC [61] for discrete model representation of the protein. In addition to amino acid composition (AAC), we have used three different sequence based feature construction techniques to fill up the remaining portion of the general PseAAC vector. Each of these features provides some sequence-order information into the discrete model. We thus created a large feature vector, whereby feature selection became necessary. Random forests algorithm was then applied to rank the features. We have then applied SVM in combination with recursive feature elimination to identify an optimal subset of features and to train the classifier. Our tool, *DNA-binding Protein Prediction model using Chou’s general PseAAC*, or *DPP-PseAAC* in short, is evaluated based on several well-established performance metrics. DPP-PseAAC convincingly demonstrated superior predictive performance compared to its predecessors. It has been made available publicly as an web interface for wide adoption.

4.2 Material and Methods

In what follows, we describe our methodology in accordance with Chou’s 5-step procedure [61], which was briefly described in Section 2.1.1.

4.2.1 Benchmark Dataset

As mentioned in the Introduction of this chapter, Liu et al. [177] prepared a stringent balanced dataset of 1075 protein samples. This dataset is known as *PDB1075* and has been widely used in literature for cross-validation. As described in their paper, the DNA-binding proteins were extracted from Protein Data Bank (PDB), December, 2013 version, by searching the *mmCIF* keyword of ‘DNA binding protein’ through the advanced search interface. The resulting proteins were filtered further as follows. Proteins shorter than

50 residues were excluded. Proteins containing the residue ‘X’ were removed because they contained unknown residue. Less than 25% sequence similarity between any protein pair was ensured by using PISCES [274]. A set of 525 DNA-binding proteins was thus obtained. The negative set of 550 proteins was prepared by randomly selecting from other proteins in PDB. The same strict filtering criteria, as mentioned above, was also applied to this negative set. Thus the benchmark dataset had a total of $525+550 = 1,075$ protein samples.

We have also used another smaller benchmark dataset for independent testing. Lou et al. [191] prepared this dataset of 93 DNA-binding and 93 non DNA-binding proteins. The dataset is widely known as the PDB186 dataset. All the sequences in this set are guaranteed to be no smaller than 60 residues and they do not contain any ‘X’ character. Pairwise sequence identity of no more than 25% was ensured in this dataset using BLASTCLUST [15].

4.2.2 Protein Sample Representation

A protein sample can be represented by its primary sequence, as shown in Equation 2.1. To represent each protein sample as a fixed length feature vector that is independent of the protein sequence length, we have utilized Chou’s general formulation of PseAAC (described in Section 2.1.7). The generalized PseAAC of a protein, as defined in Equation 2.2, is as follows:

$$P = [\psi_1 \ \psi_2 \ \dots \ \psi_u \ \dots \ \psi_\Omega]^T$$

The classical AAC is represented by subscripts $1 \leq u \leq 20$ and the subsequent features express sequence order information through one or more different schemes. The sequence order related features that we have extracted can largely be divided into two categories: position independent and position specific. Among the position independent features, we have used Dipeptides (Dip), Tripeptides and n -Gapped-Dipeptides (nGDip). All these feature extraction techniques have already been described in Section 2.1.7.

AAC, Dip and Tripeptides derive from the generalized form of *n-grams* feature type where frequencies of *n*-length peptides are used as feature vectors. Dong et al. [82] referred to it as *kmer* and used it in their DNA-BP predictor. Dip has also been successfully used in [275] for DNA-BP prediction. In our study, we extract a total of 8420 *n*-grams (kmer) features, for $n = 1, 2$ and 3 . For some features, all the samples of the training set may produce 0 frequency. Such features will naturally have no effect on the learning model. We have carefully removed these from the feature vector. Subsequently the *n*-grams feature count reduced to 8383.

We have also applied the *n*-Gapped-Dipeptides (nGDip) feature extraction technique in this work. Liu et al. have used it in building the predictor *iDNA-Prot|dis* [177]; however they called it *distance-pairs* and used a gap (distance) of 3 only. In our work, we have considered upto 25 position gaps. Thus we get a total of $25 \times 400 = 10000$ *n*-Gapped-Dipeptides features.

We have also used the Position Specific *n*-grams (PSN) feature scheme. As described in Section 2.1.7, PSN represent whether specific *n*-grams occur in specific positions in the protein sequence. The value of each such feature in any sequence will therefore be either 0 or 1 (*on* or *off*). We have considered *n*-grams for $n = 1, 2$ and 3 in case of PSN as well. However, to avoid feature space explosion, we considered only the first 10 positions of the N-terminus part for extracting the PSN features. This produced 11296 features.

Thus we have extracted a total of $8383 + 10000 + 11296 = 29679$ features. These features are represented in Chou's PseAAC as follows: For $1 \leq u \leq 20$, we have the amino acid composition in the feature vector. From $21 \leq u \leq 8383$, the dipeptide and tripeptide compositions are represented. From $8384 \leq u \leq 18383$, the features in this vector comes from the nGDip feature space. Finally, the PSN features construct the remaining portion of the PseAAC, from $18384 \leq u \leq 29679 = \Omega$.

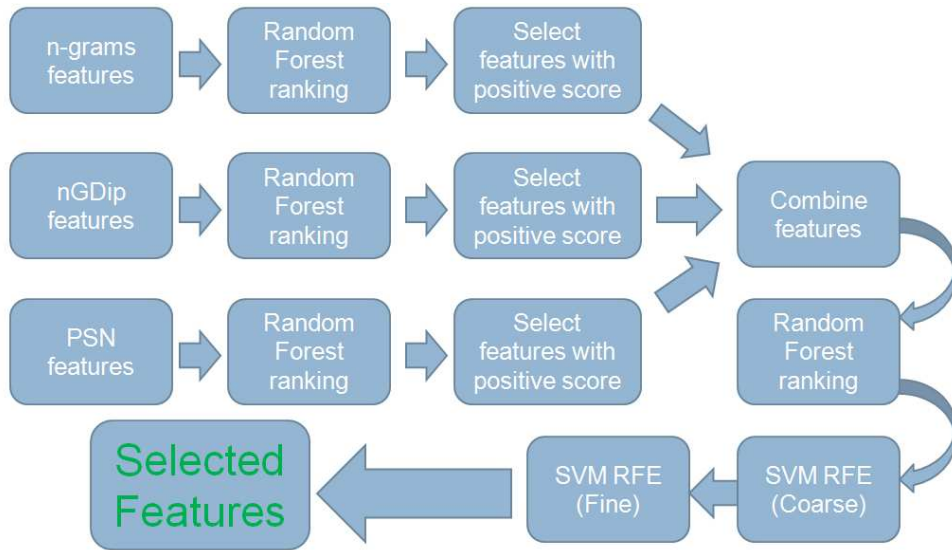
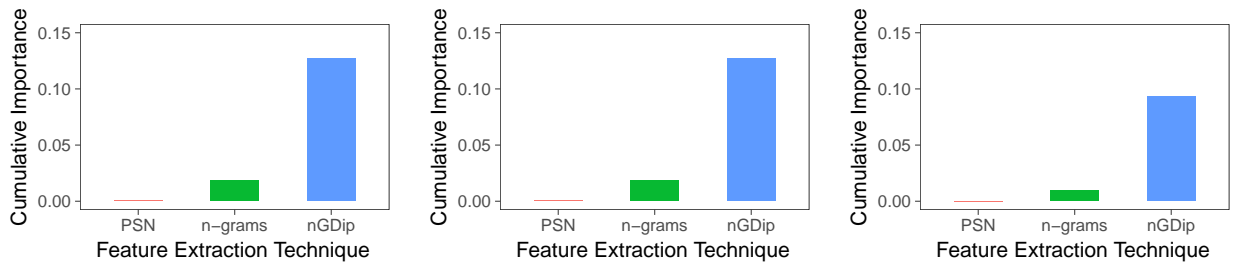


Figure 4.2: Steps in feature selection.

4.2.3 Prediction Algorithm

To reduce the computational burden of dealing with such a large feature vector, feature selection is applied as the first step in our prediction algorithm. We applied a random forests model based feature ranking, followed by multiple steps of SVM based Recursive Feature Elimination (SVM-RFE).



(a) Top 5000 features.

(b) Top 7000 features.

(c) All features.

Figure 4.3: Categorized feature importance based on random forests model based ranking. The aggregate ranking score is better for subsets of top-ranked features, compared to all features. PSN: Position Specific n -grams, n -grams: Combination of AAC, dipeptide and tripeptide composition features, nGDip: n -Gapped Dipeptides.

Feature Ranking Using Random Forests

Filter methods rank the features based on some criteria. Then a subset of top ranked features are selected. To achieve this, we would like to run random forests on the full set of features and take the *mean decrease in accuracy* as the ranking score of each feature. Note that random forests algorithm is not used as the classifier at this point, rather it is exploited as a means to generating the feature ranking. Random forests based filtering approach has also been used in *DBPPred* [191]. However, our feature space is very large, getting the feature ranking using random forests algorithm itself is a difficult task. Our attempt to get the ranking scores of such a big feature space with the best server machine in our computing laboratories did not finish even after one month's of execution.

To solve this problem, we therefore followed the approach as detailed in Figure 4.2. Firstly, we computed random forests model based rankings in each individual feature space. This was manageable, since the size of the largest feature space was around 11000. Generation of each of the three random forests (and therefore the respective ranking scores) took less than a day. Based on these 3 rankings, we selected the features with positive score in each feature space. We thus obtained 3200 *n*-grams features, 5522 nGDip features and 1214 PSN features which totals to 9936 features. These 9936 features were re-ranked using another iteration of Random forests algorithm. In this ranking, 4566 features had positive mean decrease accuracy scores, 2379 features had 0 scores and remaining features had negative scores. This is why the cumulative scores of the features are almost identical for top 5000 and 7000 features, demonstrated in Figure 4.3. And they are superior compared to the cumulative scores when all features are considered, as observed in the same figure.

Feature Ranking Using SVM-RFE

After obtaining the random forests model based ranking, we apply a *wrapper* phase. Wrapper methods search the feature space to find an optimal subset of features. The quality of the feature subset is measured by training and testing a specific classification

model. We have used SVM based Recursive Feature Elimination (RFE) approach in our work. SVM-RFE wrapper method was first introduced by [118]. Chowdhury et al. [65] have used it in building *iDNAProt-ES*. Using SVM-RFE, we re-ranked the top 7000 features as follows: SVM was first run on the entire feature set and the technique described in [118] was applied to rank the features. In the recursive step, 25 least ranked features were removed, SVM was run on the reduced feature space, and feature ranking was recomputed. The recursion was repeated until all the features are eliminated. Thus a new feature ranking is obtained. We call it the SVM-RFE (coarse) ranking.

Using this new ranking, we constructed different SVM models by varying the number of features and found the top 600 features to be most promising. (The exact details of how the number of features were varied is discussed in Section 4.3. A second round of SVM-RFE was applied in this feature space, but this time with steps of 1 feature elimination (instead of 25 features). This gives a more reliable ranking of the top 600 features, which we have called SVM-RFE (fine) ranking. Using this final ranking, we again explored several models of different feature count and found the model with 289 features to be the best model.

4.2.4 Predictor Evaluation

We have utilized jackknife cross-validation, 10-fold cross-validation test and independent test for assessing the performance of our DPP-PseAAC. These methods have already been briefly described in Section 2.1.8. As performance metrics, we have used in this work accuracy, sensitivity, specificity and Matthew's Correlation Coefficient (MCC). We have also analyzed the Area Under Receiver Operating Characteristic Curve (auROC). The reader is referred to Section 2.1.9 for details about these metrics.

Experimental Setup and Packages

We have conducted experiments using R language (version 3.2.3 or above) on three different machines with the following configurations:

- A Desktop computer with Intel Core i5 CPU @ 3.00GHz x 4, Windows 7, 64-bit OS and 4 GB RAM.
- A Desktop computer with Intel Core i5 CPU @ 3.20GHz x 4, Ubuntu 16.04, 64-bit OS and 8 GB RAM.
- A server machine with Intel Xeon CPU E5-4617 0 @ 2.90GHz x 6, Ubuntu 13.04 64-bit OS, 15 MB L3 cache and 64 GB RAM.

Random forests and Support Vector Machine (SVM) machine learning algorithms were used for feature ranking and model learning. These are available respectively from the R packages, *randomForest* and *e1071*. In the random forests algorithm, we have used the default parameters setting. In particular, the number of trees (*ntree*) was restricted to 500, while the number of variables tried at each split (*mtry*) was set to square root of the total number of features.

In addition to pre-installed packages in R, we have also used *ROCR* and *pracma* packages for performance evaluation of our model. *ggplot2* package was used for plotting relevant graphs. All of our source code, experimental results, cross-validation and independent datasets are available at the following link: <https://github.com/srautonu/DNABinding>.

4.2.5 Predictor Availability

DPP-PseAAC is freely available as an R script at <https://github.com/srautonu/DNABinding>. Additionally, we have established a publicly accessible web server at <http://dpp-pseaac.research.buet.ac.bd> to facilitate wide adoption.

4.3 Results

In this section, we describe several experiments and analyze their results. We measure the impact of different aspects in the performance of our model. Such factors of influence

include number of features, combination of the feature extraction techniques etc. We use 10-fold cross validation testing in these experiments. We also run experiments to compare DPP-PseAAC with state-of-the-art methods. For these experiments we have used jackknife cross validation and independent testing.

4.3.1 Impact of Number of Features

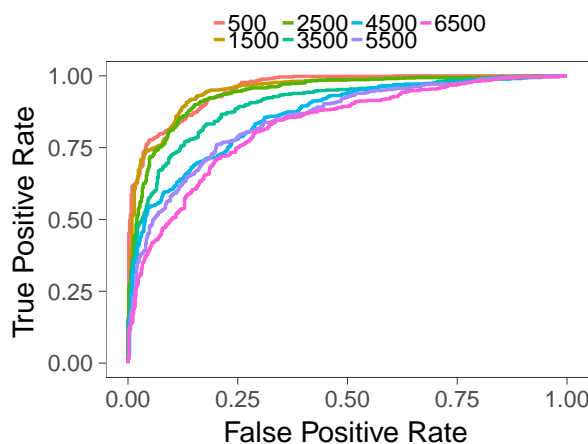


Figure 4.4: ROC-Curves of prediction models with varying number of features, generated by 10-fold cross validation on the PDB1075 dataset.

We have run several experiments varying the number of features and analyzed the impact on the classification model. The analysis was done in terms of the various performance metrics discussed in the earlier section. In this process, we were able to identify the right number of features for our model.

The ROC curve for a number of SVM models of different number of features is shown Figure 4.4. The close a ROC curve is to the top-left corner of the graph, the better is the performance of the corresponding model. Therefore, from the curves of Figure 4.4, it is clear that the performance with 500, 1500 and 2500 features is much better compared to the other feature subsets. This same conclusion can be made from Figure 4.5a.

In Figure 4.5 we plot the area under ROC curve, accuracy, sensitivity, specificity and MCC of models that are created with varying number of top-ranked features. We first

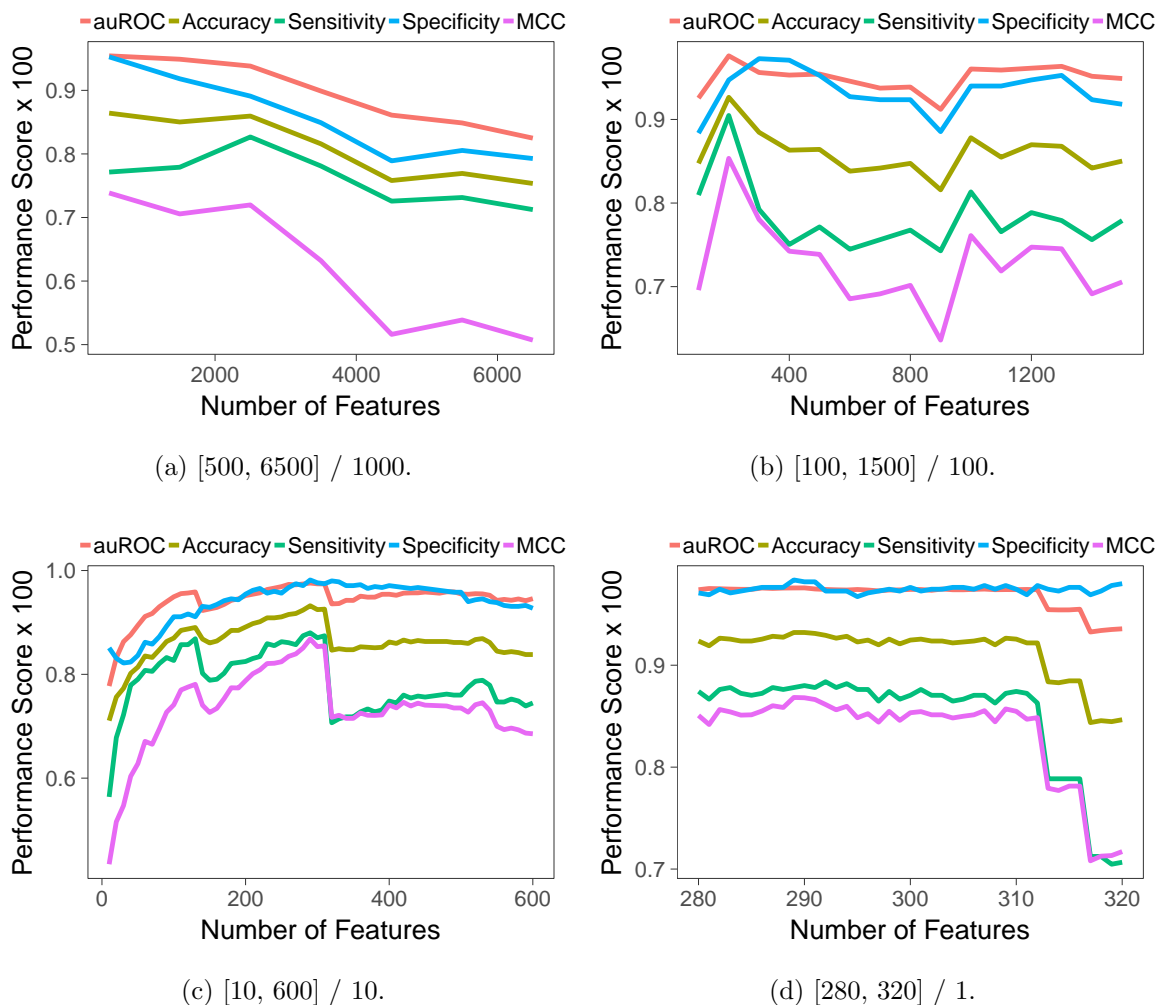


Figure 4.5: Area under ROC curve (auROC), accuracy, sensitivity, specificity and MCC of models with varying number of features, generated by 10-fold cross validation on the PDB1075 dataset. The $[x, y]/z$ style annotation of each sub-figure means that, the experiment started with x top-ranked features. Then a model was trained with z more features and the performance scores were recomputed. This process continued until the feature count became y .

explore a large feature space, albeit with coarse granularity. That means, the number of features that are added (removed) between experiments is large. As an example, Figure 4.5a is generated by starting with a model with 500 top-ranked features. The coarse grain SVM-RFE feature ranking was used in this case. Then 1000 next ranked features were added in each iteration. Based on the curves, the feature space range [100,

1500] seems promising. Therefore, more models are generated in this space, however the change of features in each step becomes finer: 100 features. We thus examine 15 models, whose performances are recorded in Figure 4.5b. Moving on this way, we keep zooming in the interesting terrain of the feature space and increase our thoroughness in investigating the narrowed-down spaces. Figure 4.5c examines 60 models, with 10 feature increase steps. The fine grain SVM-RFE feature ranking was used in this and subsequent experiment. From this Figure, the range [280, 320] seems most interesting. So, this space is investigated, with single feature increase steps, yielding 41 different models. Based on the performance comparison of these models (in Figure 4.5d), the model built with 289 features was chosen to be our final classifier. Among the features, there were 102 n -grams, 126 nGDip and 61 PSN features.

4.3.2 Impact of Feature Extraction Techniques

To analyze the contribution of the different feature extraction techniques in building the model, we have run some experiments with the top 100 features. In this subset, there are 35 n -grams features, 54 nGDip features and 11 PSN features. We trained three different SVM models using each of these three subsets of features. In another model, we trained with all the 100 features. In Figure 4.6a, the accuracy, sensitivity, specificity and MCC values from these four models are compared. The nGDip feature extraction technique is a clear winner over the other two, while the combination of all performs slightly better than that.

The size of the feature vectors in the above comparison was widely different. Therefore, we conducted another experiment where we trained 3 different models using top 100 features of the 3 individual feature extraction techniques. We compare the performance of these models to the combined model in Figure 4.6b. The superiority of combined feature space over the individual feature spaces hold in this setting as well. PSN, n -grams and nGDip feature extraction techniques individually achieve accuracy values of 62%, 74% and 84%, respectively. When the combined feature space is used instead, the accuracy

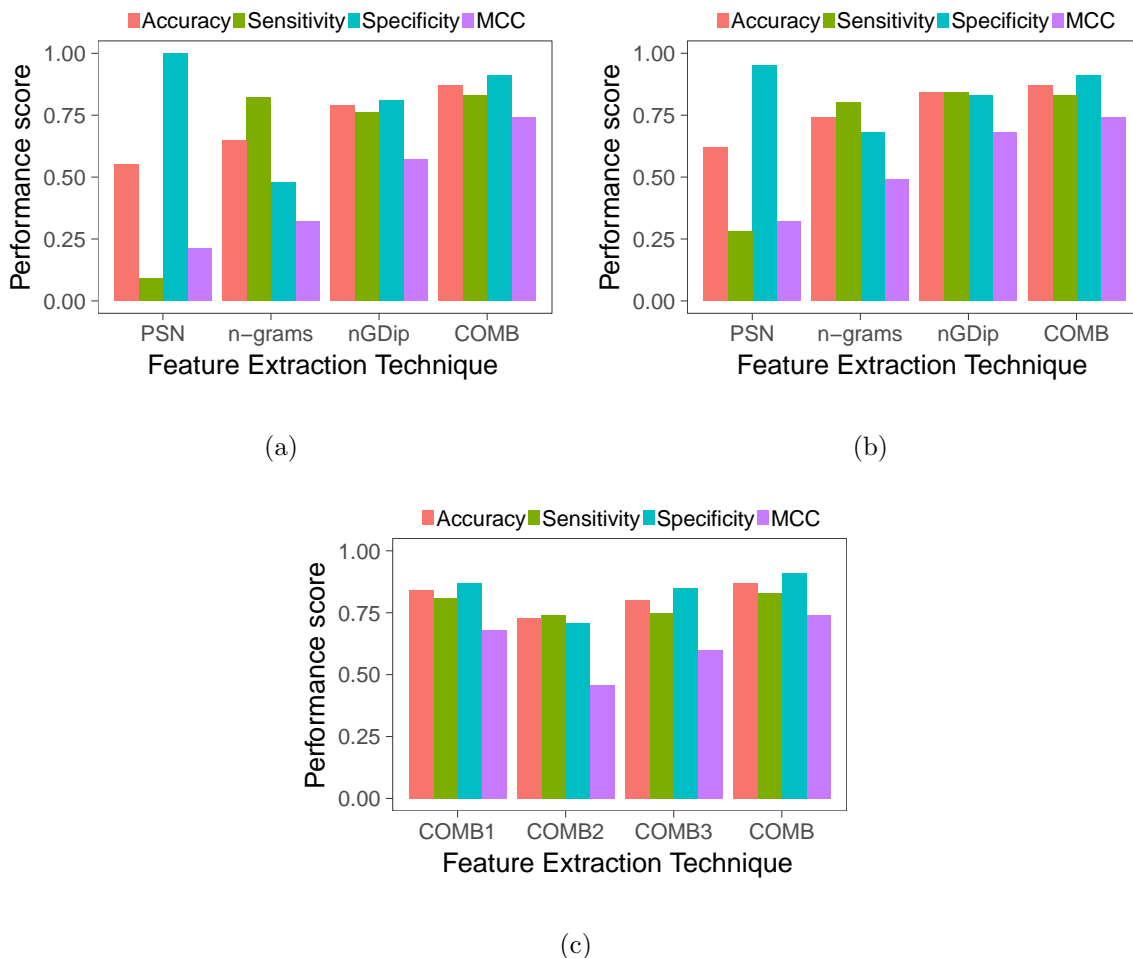


Figure 4.6: Performance of different feature extraction techniques. The results are obtained from 10-fold cross validation on the PDB1075 dataset. PSN: Position Specific n -grams, n -grams: Combination of AAC, dipeptide and tripeptide composition. nGDip: n -Gapped-Dipeptides. COM: Combination of all the feature extraction techniques. COM1: Combination of n -grams and nGDip. COM2: Combination of n -grams and PSN. COM3: Combination of nGDip and PSN.

increases to 87%. Similarly the MCC increases from respective individual values of 0.32, 0.49, 0.68 to 0.74 for the combined feature space.

Another observation is worth noting from these experiments. The PSN only classifier is extremely biased towards the negative class. The accuracy of the positive class in the model built with top 100 PSN features is only 28%, while that of the negative class is 95%.

n -grams feature space, on the other hand, provides a slight bias towards the positive class. The nGDip feature space produces the most balanced classifier. Its ability to predict the positive class is in fact slightly better than that of the model constructed on the combined feature space. However, the specificity of the latter is much better, resulting in a superior overall accuracy and MCC scores.

From the above discussion, it is clear that each of the feature spaces have contribution in improving the classification performance with the combined feature space. We ran one more experiment to check whether this is indeed the case. In this experiment, we used combination of two feature spaces, leaving the other feature space out. We chose the top 100 features to construct the model. We compared the performance of the three generated models with that of the model created using the combination of all 3 feature spaces. The results are shown in Figure 4.6c. The composition of each combination is tabulated below:

Id.	Feature spaces	n -Grams	nGDip	PSN
COMB1	n -grams, nGDip	38%	62%	-
COMB2	n -grams, PSN	67%	-	33%
COMB3	nGDip, PSN	-	75%	25%
COMB	n -grams, nGDip, PSN	35%	54%	11%

It is clear from Figure 4.6c that among the 2 feature space combinations, the combination of n -grams and nGDip is the best. Nonetheless, adding the PSN feature space clearly adds value - the model constructed with combination of all 3 feature spaces is superior to models built with 2 feature space combinations in terms of each performance metric we have used.

4.3.3 Discriminant Visualization

To study the discriminant power of different features, we calculated the discriminant weight vector in the feature space. This vector is also needed during the RFE step and is calculated following the steps used in [118]. The discriminative weights of top 25 features

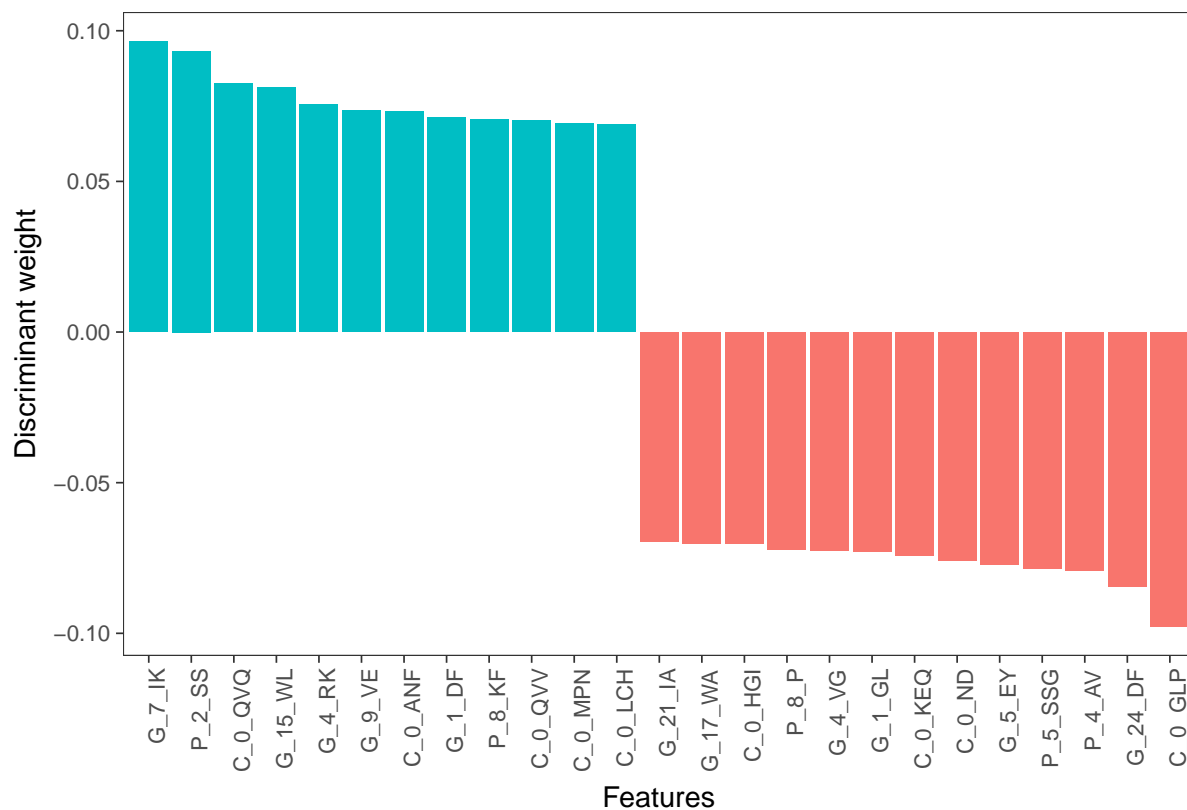


Figure 4.7: The discriminative weights of top 25 features.

are shown in Figure 4.7. The feature names are encoded as follows:

- A feature starting with the prefix “G_” is an nGDip feature. The integer that follows is the particular gap being considered. The dipeptide in question is given as the suffix. Therefore, the feature “G.7_IK” represents the normalized frequency of dipeptide “IK”, such that the residues ‘I’ (Isoleucine) and ‘K’ (Lysine) are separated from each other by 7 residues.
- A feature starting with the prefix “C_0_” is an n -grams feature. The suffix represents the particular n -gram. Therefore, the feature “C_0_QVQ” represents the normalized frequency of the tripeptide “QVQ”.
- A feature starting with the prefix “P_” is a PSN feature. The integer that follows is the particular position. The n -gram in question is given as the suffix. Therefore,

the feature “P_2_SS” represents whether the dipeptide “SS” occurs in the second position of the protein sequence.

There are 12 features with positive scores and 13 features with the negative scores. The absolute weights of both set of features are in the same tight range of [0.07, 0.10]. The decrease of importance is gradual as we move to lesser ranked features, and the pattern of the decrease is almost identical for both set of features. The features with positive (negative) scores contribute in prediction of the positive (negative) class.

4.3.4 Comparison between DPP-PseAAC and Existing Techniques

As discussed earlier, we conducted several 10-fold cross validation tests using PDB1075 dataset. We varied the number of features to identify the best model. The model with 289 top-ranked feature demonstrated the best performance. While comparing with the state-of-the-art predictors, DPP-PseAAC will actually refer to the model with these 289 top-ranked features. The 10-fold cross validation accuracy, sensitivity, specificity, MCC and area under ROC curve scores of the model respectively were 93.21%, 87.81%, 98.36%, 0.87 and 0.98. Subsequently, we have compared the performance of DPP-PseAAC with prominent prediction tools from literature, using jackknife cross-validation approach. The results are recorded in Table 4.1, the best values having been highlighted in bold faced fonts. The results for DNAbinder, DNA-Prot, iDNA-Prot, iDNA-Prot|dis were collected from [177]. For the other predictors, the cross-validation results with the same benchmark dataset was available in the respective research papers. However, PseDNA-Pro used a benchmark dataset other than PDB1075.

DPP-PseAAC demonstrates superiority over all the earlier predictors in terms of each of the performance metrics used. Since PDB1075 is a stringent dataset which guarantees that pairwise sequence similarity is no more than 25%, any concerns of overestimation in jackknife approach is mitigated [61].

Table 4.1: Comparison of DPP-PseAAC with previous methods using jackknife cross-validation on the PDB1075 dataset.

Method	Accuracy	Sensitivity	Specificity	MCC	auROC
DNAbinder (dimension 21)	73.95	68.57	79.09	0.48	0.8140
DNAbinder (dimension 400)	73.58	66.47	80.36	0.47	0.8150
DNA-Prot	72.55	82.67	59.76	0.44	0.7890
iDNA-Prot	75.40	83.81	64.73	0.50	0.7610
iDNA-Prot dis	77.30	79.40	75.27	0.54	0.8310
PseDNA-Pro	76.55	79.61	73.63	0.53	-
iDNAPro-PseAAC	76.76	75.62	77.45	0.53	0.8392
Kmer1 + ACC	75.23	76.76	73.76	0.50	0.8280
Local-DPP	79.20	84.00	74.50	0.59	-
iDNAProt-ES	90.18	90.38	90.00	0.80	0.9412
DPP-PseAAC	95.91	94.10	97.64	0.92	0.9884

Next we compare performance of DPP-PseAAC with state-of-the-art predictors using independent testing approach. The PDB186 dataset is used in this case. However, if there is significant sequence similarity between proteins of the training set and that of the testing set, then the independent test results will be over estimated. To avoid this, proteins of PDB1075 that had more than 25% sequence identity to any protein in the PDB186 dataset were removed using BLASTCLUST [15]. The prediction model was then rebuilt using this reduced PDB1075 dataset. This protocol was introduced by Liu et al. [177] and has subsequently been followed in independent testing of other DNA-BP predictors. The reduced PDB1075 contained 487 positive samples, 548 negative samples; the total size of the training set became 1035.

The independent test results of DPP-PseAAC and state-of-the-art predictors are recorded in Table 4.2. The results for DNABIND, DNAbinder, DNA-Threader, DNA-Prot, iDNA-Prot and DBPPred were obtained from [191]. As the newer predictors had adopted this dataset for independent testing, the test results for these predictors were obtained from the respective research papers.

Table 4.2: Comparison of DPP-PseAAC with previous methods using independent test.

Method	Accuracy	Sensitivity	Specificity	MCC	auROC
DNABIND	67.70	66.70	68.80	0.355	0.6940
DNAbinder	60.80	57.00	64.50	0.216	0.6070
DBD-Threader	59.70	23.70	95.70	0.279	-
DNA-Prot	61.80	69.90	53.80	0.240	-
iDNA-Prot	67.20	67.70	66.70	0.344	-
DBPPred	76.90	79.60	74.20	0.538	0.7910
iDNA-Prot dis	72.00	79.50	64.50	0.445	0.7860
iDNAPro-PseAAC	69.89	77.41	62.37	0.402	0.7754
Kmer1 + ACC	70.96	82.79	59.13	0.431	0.7520
Local-DPP	79.00	92.50	65.60	0.625	-
DPP-PseAAC	77.42	83.87	70.97	0.553	0.7986

From the results, we can see that DPP-PseAAC performs better than all prior predictors, except for Local-DPP. If Local-DPP is left out of the comparison, then DPP-PseAAC has the best accuracy, sensitivity, MCC and area under ROC curve. DBD-Threader has the best specificity, but its sensitivity is extremely poor. DBPPred also has better specificity than our method. But ours outperforms DBPPred in terms of sensitivity. The accuracy and MCC values are similar for both approaches, albeit DPP-PseAAC has a slight edge. Now, let us compare DPP-PseAAC with Local-DPP method. Local-DPP has the highest sensitivity among all the methods, a commendable score of 92.5%. Its sensitivity, however, is only 65.60%. So, it is skewed considerably towards the positive class. DPP-PseAAC has a better sensitivity and is more balanced in its predictive performance in contrast.

To summarize, DPP-PseAAC shows best performance in each of the performance metrics in the jackknife cross-validation testing. In case of independent testing, its performance is also commendable, remaining behind of only Local-DPP.

Table 4.3: Structure based predictors at a glance. ANN: Artificial Neural Network, LR: Logistic Regression.

<i>Tools</i>	<i>Feature Extraction Technique(s)</i>	<i>Feature Selection</i>	<i>Classifier</i>
Stawiski et al. [256]	Analysis of electrostatic patches, surface clefts, Conservation analysis of the sequence.	12 features	ANN (3 layers)
Ahmad et al. [11]	Bulk electrostatic properties.		ANN (2 layers)
DNABIND [261]	Proportion of certain amino acid residues, Spatial asymmetry of amino acid residues, Dipole moment of the entire molecule.	10 features	LR
DBD-Hunter [106]	Library of DNA-protein complex structures, Structural alignment, Evaluation of a statistical potential, Matching score thresholding.		
DBD-Threader [107]	Library of DNA-protein complex structures, Target protein’s sequence, Matching score thresholding.		

4.4 Discussion

In this section, we present brief discussion on several aspects relevant to our work.

4.4.1 Differentiation between DPP-PseAAC and Existing Predictors

To give a clear picture of differentiation between DPP-PseAAC and prior art, Tables 4.3 and 4.4 show the different steps taken in building these prediction models. As can be seen, the novelty in DPP-PseAAC lies in the addition of tripeptide composition and PSN features into Chou’s general PseAAC. The combination of random forests algorithm for feature ranking followed by recursive feature elimination using SVM is also a new approach in this prediction problem.

Another distinguishing factor is that we explored a large feature space, comprising 29679 features and then selected 289 features for training the model. Even the selected feature set’s size is larger than the number of features used in most of the earlier works. The most recent predictors, iDNAProt-ES [65] and Local-DPP [277] respectively used 86 and 120 features. However, it is important to note that both of the above-mentioned methods use PSSM based features, extraction of which take time. Our approach, on the other hand, can extract all the necessary features from a target protein in a single pass along the sequence and then use the classifier to predict its class. Additionally, if the target protein does not have enough homologous sequences in the database, the generated PSSM cannot describe the protein adequately. Therefore, any prediction model dependent on PSSM information will produce wrong predictions in such a case [164].

4.4.2 Some Errors in Results of Earlier Predictors

In the independent testing, we have not compared DPP-PseAAC with iDNAProt-ES [65], which was the best predictor so far in terms of both jackknife and independent testing. DPP-PseAAC outperformed it in the jackknife cross-validation test. And we found a flaw in the independent testing of iDNAProt-ES. As discussed earlier, the protocol followed by Liu et al. [177] was to eliminate the sequences in PDB1075 that had more than 25% pairwise similarity with the independent test set (PDB186), and then retrain the predictor with this reduced set. This was followed by subsequent authors as well. Unfortunately, this important step was missed in the independent testing of iDNAProt-ES. Therefore, the performance scores reported for that tool are over estimations. As such, we have excluded it in our independent test comparisons. Notably, we have notified Chowdhury et al. through private communication about the error in their independent test process and they are currently in the process of rerunning their experiments.

Another minor error is observed in the MCC score of independent test of Local-DPP [277]. In explaining the error, we use the symbols that were defined in Section 2.1.9. As we know P and N values of PDB186 (93 each), the TP, TN, FP, FN values can easily

be computed from the accuracy, sensitivity and specificity data. When we plug these values into Equation 2.3, we get an MCC of 0.602. However, the reported value in [277] is 0.625, which is over estimated.

4.4.3 Unavailability of BLASTCLUST in the Latest Version of Standalone BLAST

Liu et al. [177] created the reduced PDB1075 dataset using BLUSTCLUST [15]. Subsequent authors have followed the same steps. However, the reduced PDB1075 is not made publicly available by any of the authors. So, we needed to follow the same steps to generate this reduced training set. However, we were not able to find the BLUSTCLUST tool in the latest version of standalone BLAST software downloadable from NCBI [2]. Also, some discussion forums suggested that it was deprecated [3]. While we found an older version (version 2.2.14) from NCBI that contained BLUSTCLUST, we could not make it work. For example, we tried to check how many clusters are there in the PDB186 data set with a 25% cut off, but all proteins showed up in single cluster, which seemed wrong. As such, we reached out to Wei et al. [277] and they kindly shared their reduced PDB1075 dataset with us.

4.4.4 Jackknife Cross-validation vs. Independent Testing

We have shown that DPP-PseAAC has the best performance in terms of jackknife testing. However, it came second in independent testing. Also, there was quite a fall in the performance scores. Since PDB1075 has less than 25% pairwise sequence similarity, the jackknife cross-validation results should be trusted. We think, the lesser performance in the independent test can easily be explained by the protocol that was used. As discussed earlier, the PDB1075 was reduced in size to eliminate sequence similarity of this set with sequences in the PDB186. This eliminated 40 samples from the training set. More importantly, 38 of these samples were positive samples. Therefore, data imbalance was

introduced, resulting in a model that is inferior to the original model. We did not handle the imbalance on purpose. This is because this was not mentioned in the protocol followed in the earlier literature. Besides, handling the imbalance by under sampling or over sampling would introduce a significant difference in the model construction process and as such the independent test results would not be representative at all of our original model.

In general, we have preferred jackknife cross-validation results over independent test results, following Chou’s argument that sheds doubt on the objectivity of the independent testing [61]:

“The way of how to select the independent proteins to test the predictor could be quite arbitrary unless the number of independent proteins is sufficiently large. This kind of arbitrariness might result in completely different conclusions. For instance, a predictor achieving a higher success rate than the other predictor for a given independent testing dataset might fail to keep so when tested by another independent testing dataset. Accordingly, the independent dataset test is not a fairly objective test method although it was often used to demonstrate the practical application of a predictor.”

4.5 Conclusion

In this chapter, we present DPP-PseAAC, a machine learning based predictor for DNA-binding proteins. We apply several sequence based feature extraction techniques on a benchmark dataset called PDB1075. Random forests and SVM-RFE methods are then applied on the proteins, as represented by the extracted features, to obtain a reliable ranking of the features. Finally, SVM with linear kernel is employed to train a prediction model. Our approach outperforms state-of-the-art techniques according to different performance metrics in jackknife cross-validation. The independent test results are also found to be satisfactory. Our predictor is available as an R script that can readily be applied to target protein sequences, without dependency on any other services or pre-processing

(e.g. computation of PSSM or structural information etc.). DPP-PseAAC is also available as a publicly accessible web based predictor. We hope the simple to use web interface, combined with the good performance, will lead to wide adoption of DPP-PseAAC.

In the next chapter, *Protective Antigen Prediction*, we focus on building a new computational model to identify protective antigens in an efficient and accurate way. Our model extracts meaningful information directly from the protein sequences, without any dependence on functional domain or structural information.

Table 4.4: Sequence based predictors at a glance. SVM: Support Vector Machine, RF: Random Forests, GNB: Gaussian Naïve Bayes.

<i>Tools</i>	<i>Feature Extraction Technique(s)</i>	<i>Feature Selection</i>	<i>Classifier</i>
DNAbinder [155]	PSSM-400	400 features	SVM
DNA-Prot [154]	Frequency of amino acid/amino acid groups, hydrophobic, hydrophilic, neutral residues, PredSS from PSIPRED, Amino acid physico-chemical properties, Split sliding 10 residue windows.	CFSS (20 features)	RF
iDNA-Prot [168]	AAC, coefficients of the second order Grey differential equation with one variable.	23 features	RF
DBPPred [191]	AAC, PredSS, PredRSA Auto-correlation coefficients of PSSM. Percentile values of PSSM scores.	RF filter GNB Wrapper (56 features)	GNB
iDNA-Prot dis [177]	Amino acid distance-pair coupling Amino acid reduced alphabet profile	602 features	SVM (RBF)
iDNAPro-PseAAC [174]	Profile-based protein representation. PseAAC ($\lambda = 3$).	23 features	SVM (RBF)
Kmer1 + ACC [82]	ACC, kmer composition, Physico-chemical properties.		SVM
PseDNA-Pro [176]	OAAC, PseAAC, PDT	573 features	SVM (RBF)
Waris et al. [275]	DPC, SAAC, PSSM.		SVM
Local-DPP [277]	Local Pse-PSSM	120 features	RF
iDNAProt-ES [65]	AAC, bigram, auto-covariance from PSSM, Dubchak features, Structural features from SPIDER2.	SVM-RFE (86 features)	SVM (Linear)
DPP-PseAAC	AAC, dipeptide and tripeptide comp., Gapped dipeptide composition, Position specific features.	RF filter SVM-RFE (289 features)	SVM (Linear)

Chapter 5

Protective Antigen Prediction

An antigen is a protein capable of triggering an effective immune system response. Protective antigens are the ones that can invoke specific and enhanced adaptive immune response to subsequent exposure to the specific pathogen or related organisms. Such proteins are therefore of immense importance in vaccine preparation and drug design. However, the laboratory experiments to isolate and identify antigens from a microbial pathogen are expensive, time consuming and often unsuccessful. This is why *Reverse Vaccinology* has become the modern trend of vaccine search, where computational methods are first applied to predict protective antigens or their determinants, known as epitopes. In this chapter, we focus on building a new computational model that can identify protective antigens by extracting meaningful information solely from the protein sequences. Our prediction model does not need any functional domain or structure specific features, nor does it depend on any predicted features from other predictors. We have used random forests algorithm as well as SVM-RFE to select an optimal set of features. Random forests was also used to train the classifier. Named as *Antigenic*, our proposed model demon-

Much of the material in this chapter is taken without alteration from the following manuscript.

- Rahman, M. S., Rahman, M. K., Saha, S., Kaykobad, M., & Rahman, M. S. *Antigenic: An improved prediction model of protective antigens*. (Under review) Artificial Intelligence in Medicine

strates superior performance compared to the state-of-the-art predictors on a benchmark dataset. Antigenic achieves accuracy, sensitivity and specificity values of 78.04%, 78.99% and 77.08% in 10-fold cross-validation testing respectively. In jackknife cross-validation, the corresponding scores are 80.03%, 80.90% and 79.16% respectively. The source code of Antigenic, along with relevant dataset and detailed experimental results, can be found at <https://github.com/srautonu/Antigenic>. A publicly accessible web interface has also been established at: <http://antigenic.research.buet.ac.bd/>.

5.1 Introduction

An antigen is a protein that is capable of triggering a measurable immune system response [102]. Antigens can be subdivided into overlapping subclasses such as serodiagnostic, crossreactive and protective antigens [193]. Serodiagnostic antigens are associated with a differential humoral antibody response between naive and exposed individuals. Such antigens are important for diagnostics purposes. Cross-reactive antigens are associated with a strong humoral antibody response in both naive and exposed individuals. Protective antigens, on the other hand, are the ones that can stimulate protective immunity against pathogens. That is, these antigens can invoke specific and enhanced adaptive immune response to subsequent exposure to the specific pathogen or related organisms. Protective antigens are of immense importance in vaccine preparation and drug design [110, 189, 232].

Vaccines are molecular or supramolecular agents that can stimulate protective immunity against microbial pathogens. They can prevent, or at least improve, the effects of infection [218]. Vaccination has been the most effective method of preventing infectious diseases such as influenza, smallpox, varicella, diphtheria, tetanus, polio, hepatitis, rotavirus and more [19, 42, 100, 163, 235, 279]. However, the battle against many infectious diseases is far from complete. It is still difficult to develop safe and effective vaccines against tuberculosis, HIV, malaria and so on [278].

Vaccines are prepared from killed or attenuated microorganisms, or subunits purified from them [9,218]. While vaccines based on attenuated pathogens can be highly effective, this technique is seldom used in modern vaccinology due to safety concerns and technical reasons [12]. Subunit vaccines, on the other hand, use only the protective antigens, instead of the entire microorganism. This reduces the chance of any adverse reaction to the vaccine [233]. The hepatitis B vaccine, containing the surface antigen HbsAg, is an example of one of the most successful subunit vaccines [262,263]. The advent of recombinant DNA technology (rDNA) has conceived the idea of *multiepitopic* vaccines [136]. In this technique, several protective epitopes (parts of an antigen that is recognized by the immune system) are included in a single molecule, immunodominant but non-protective epitopes are discarded. Epitopes exerting adjuvant effects can also be included to enhance the protective response. This opens up the possibility of designing highly efficient, multi-target vaccines [253].

The modern trend in vaccine preparation has therefore been towards creating subunit vaccines or epitope vaccines containing only full or partial protective antigens. As a result, identification of protective antigens or their determinants is a key step in any vaccine development project [83]. The microbiological approach for antigen identification comprises several steps. At first, the target pathogen is cultivated under laboratory conditions. It is then purified and dissected into the constituent proteins. The proteins are then assayed in cascades of *in vitro* and *in vivo* assays. Finally, the proteins which display requisite protective immunity are identified [282]. While this process requires many hours of expensive and laborious tasks, it does not always yield fruitful results. For example, it is not always possible to cultivate a particular pathogen outside of the host organism. Also, as many proteins are only expressed transiently during the course of an infection, the antigens expressed *in vivo* may not always express during *in vitro* cultivation [102]. These limitations of the laboratory experiments, coupled with wide availability of whole genome sequences of pathogens, have led researchers explore techniques that are based on computational genomics and thus a new paradigm known as Reverse Vaccinology has emerged.

Reverse Vaccinology (RV) [224, 233] is a computational pipeline for identification of protective antigens or epitopes against microorganisms from their genome sequences. In this approach, all proteins of a pathogen proteome are first screened *computationally* for their vaccine potential. Computationally predicted protective antigens are then tested *in vivo* and *in vitro* for their immunogenicity. This approach dramatically cuts down the cost and increases the speed of progress in vaccine discovery. RV was first applied to the development of a vaccine against serogroup B *Neisseria meningitidis* (MenB), which causes sepsis and meningitis in children and young adults [224]. This has eventually led to the approval of the first MenB vaccine, BEXSERO[®], for use in Europe [270], and United States [103]. This is a milestone for rational vaccine design using RV. This principle for vaccine development has successfully been applied against many other pathogens, including *Helicobacter pylori* [40], *Streptococcus pneumoniae* [280], *Porphyromonas gingivalis* [240], *Chlamydia pneumoniae* [201], *Bacillus anthracis* [18] and *Mycobacterium tuberculosis* [20].

Over the years, researchers have developed many computational techniques for protective antigen prediction. Some of these techniques are focused on specific pathogen models, while some are more generic. Some techniques use concepts of sequence alignment, while other ones leverage statistical tools or machine learning methods. In this chapter, we propose a protective antigen predictor that is based on the latter approach. Based on features extracted from the primary sequence of the protein, our method provides a fast and simple prediction model that outperforms the existing predictors. Prior to presenting the details of our predictor, we briefly review the literature of protective antigen prediction here.

For a sequence-alignment based approach to be useful, sequences of many extant antigens must be available in a database. Sequence searching programs such as BLAST [14], FASTA [220], PSORT [204] etc., can then be applied to identify similar sequences in the target genome. However, such an approach will fail to discover truly novel protective antigens which lack any sequence similarity with the repository of known protective antigens.

Another criterion, that has frequently been used to screen for potential antigens, is the likelihood of a protein containing a signal sequence. SignalP [210] has widely been used in this regard. It originally employed neural networks to predict the presence and location of signal peptide cleavage site [211]. Subsequently a hidden Markov model (HMM) was implemented which is able to discriminate uncleaved signal anchors from cleaved signal peptides [212]. Several updates to this predictor have been made in recent years [29, 221]. One of the limitations of SignalP, however, is overprediction, as it cannot reliably discriminate between several very similar yet distinct signal sequences [102].

Vivona et al. [272] developed a system for antigen discovery, called NERVE, that works in several stages as follows. Firstly, the target protein's subcellular localization is predicted. Then whether the protein is an adhesin is determined. This is followed by the identification of transmembrane domains. The protein is then compared against human and pathogen proteomes. Finally it is assigned a suggestive score. However, the system requires software download and database setup and does not include precomputed data of vaccine target prediction, which makes its use inconvenient and time consuming [123].

Doytchinova et al. [84] proposed the first alignment-free approach for antigen prediction. They trained the predictor for three different models: bacteria, virus and tumor. Each model was trained with a balanced dataset of 100 known protective antigens and 100 non-antigens. The principal amino acid properties were represented by z descriptors, originally derived by Hellberg et al. [125]. A transformation using auto cross covariance (ACC) [281] was then applied to produce a uniform vector of 45 terms for each protein sequence. Then a two-class discriminant analysis was performed using the partial least squares technique (DA-PLS). The cross-validation accuracy of their predictor was 82% for the bacterial model, 87% for the viral model and 85% for the tumor model. The models were implemented in a server called VaxiJen [7], which has since been widely used. However, the dataset used to create VaxiJen was rather small. Additionally, several of the sequences in the non-antigen set were subsequently predicted as antigens by other methods [137]; some were also experimentally discovered as such [160, 190].

In a subsequent work, Doytchinova et al. [85] added parasite and fungal models to the VaxiJen predictor. For this purpose, 117 parasitic and 33 fungal antigens were identified from the literature. For each antigen, a non-antigen protein was randomly selected from the same species. The same features and learning algorithms were used as before. The parasite model achieved an accuracy of 78% while the fungal model obtained 97% accuracy.

Ansari et al. [16] developed AntigenDB, a database compiling more than 500 antigens, from 44 important pathogenic species. This database maintains information regarding the sequence, structure, origin, etc. of antigens. B and T-cell epitopes, MHC binding, function, gene-expression and post translational modifications are also available for some antigens. He et al. [123] introduced Vaxign, another web-based vaccine design system that can predict protein subcellular location, transmembrane helices, adhesin probability, conservation to human and/or mouse proteins etc. The precomputed Vaxign database contains prediction of vaccine targets for more than 70 genomes.

Magnan et al. [193] developed another predictor for protective antigens, called ANTI-GENpro. Unlike VaxiJen's approach of pathogen specific prediction models, they created a generic classifier of antigens from any pathogen. To train their classifier, they first collected known protective antigens from literature. They then augmented this set using human immunoglobulin reactivity data obtained from protein microarray analyses. ANTI-GENpro achieved 76% accuracy in 10-fold cross-validation experiments. Unfortunately, ANTI-GENpro server [5] restricts queries to only one protein sequence per submission. This makes its use on a genome-wide scale quite impractical [89].

El-Manzalawy et al. [89] proposed another predictor called BacGen which can classify antigens for bacteria model only. They used amino acid moment descriptors (AAMD) [246] as features. After applying Haar wavelet transform (HWT) [119], they used random forests [34] as the classifier. Finally they combined the prediction of random forests algorithm with SignalP [210] prediction. Their approach produced results that are competitive with ANTI-GENpro. However, while BacGen was implemented as a web server (<http://ailab.cs.iastate.edu/bacgen/>), it does not seem to be in service anymore.

Jaiswal et al. [137] also developed a web-based predictor, for protein vaccine candidates (PVCs) for bacterial pathogens. Called Jenner-Predict, the predictor targets host-pathogen interactions by considering known functional domains from various protein classes. Altindis et al. [12] examined the structural and functional features recurring in known bacterial protective antigens to define “protective signatures” which can be used for protective antigen discovery. They applied their approach to *Staphylococcus aureus* and Group B *Streptococcus* and were able to identify two new protective antigens, in addition to re-discovering the already known protective antigens. Ong et al. [214] in a recent publication verified the critical role of adhesins, subcellular localization, peptide signaling, in predicting secreted extracellular or surface-exposed protective antigens. They also found a significant negative correlation of transmembrane α -helix to antigen protectiveness in Gram-positive and Gram-negative pathogens. Their findings can be used to extract relevant features from the protein secondary structure to discriminate between protective antigens and non-antigens.

While significant amount of work has been done in protective antigen prediction, the performance of the current predictive tools has left a lot of room for improvement. Also, some of the state-of-the-art predictors use feature extraction techniques that are time consuming, some use sophisticated prediction models which are susceptible to the overfitting problem. In this chapter, we therefore propose a protective antigen predictor that extracts features from the protein sequence alone, that has a fast and simple prediction model and that outperforms the existing predictors. We have followed Chou’s 5-step procedure [61] for establishing our predictor. The steps include dataset preparation, extracting relevant features from protein sequences, learning the classification model using a powerful algorithm, objectively evaluating the predictor and finally making the predictor available through a web server for wide adoption. We have collected a benchmark dataset from literature and then applied a fixed length vector representation of the protein. In addition to amino acid composition (AAC), we have used three different sequence based feature construction techniques to create the feature vector. Each of these features provides some sequence-order information. As we created a large feature vector, feature

selection became necessary. Random forests [34] algorithm was then applied to rank the features. We have then applied Support Vector Machine (SVM) [32] in combination with Recursive Feature Elimination (RFE) to identify an optimal subset of features. Finally random forests algorithm was used again, but this time to train the classifier. Named as *Antigenic*, our predictor has been evaluated based on several well-established performance metrics. Antigenic convincingly demonstrated superior predictive performance compared to its predecessors. Therefore, it has been made available publicly as an web interface for wide adoption.

5.2 Material and Methods

In what follows, we describe our methodology in accordance with Chou’s 5-step procedure [61], which was briefly described in Section 2.1.1.

5.2.1 Benchmark Dataset

Table 5.1: Size and composition of the six protein sets used as the training set.

Protein set	Size	Antigenic	Non-antigenic
PAntigens	213	213	0
Brucella	206	70	136
Burkholderia	17	5	12
Candida	13	3	10
Malaria	333	114	219
Tuberculosis	542	171	371
Total	1324	576	748

In order to create a robust predictor, there needs to be a reliable training dataset of relatively large size. For our study we have collected the benchmark dataset from [193]. This dataset, prepared by Magnan et al., was not available publicly. However, they kindly provided us with the dataset upon request through private communication. Below, we provide a brief description of the dataset and how it was prepared.

Magnan et al. [193] argued that mere literature review did not generate a satisfactory collection of protective antigens. Therefore, they prepared the benchmark dataset based on protein microarray data analysis for training and testing their predictor. They leveraged a high-throughput technology [72] to study the humoral immune response to pathogen infection using protein microarrays. In this approach, proteins of a pathogen genome are expressed by a proprietary *in vitro* expression system. These expressed proteins can then be probed with sera from naive, exposed and vaccinated individuals. The resulting reactivity data gives a reliable estimate of the humoral immune response. The protein microarray data can thus be used to prepare a dataset of antigens and non-antigens to train a predictor. Although the microarray data does not directly provide information about whether or not a particular antigen is protective, Magnan et al. [193] hypothesized that the actual protective antigens are significantly overrepresented among the set of antigens for which the protected individuals elicit a significant antibody response, and the unprotected individuals do not. They have validated this hypothesis in their work.

The benchmark dataset contains a training set as well as a testing set. The training set consisted of 6 subsets. Of these, 5 subsets were curated from protein microarray data analysis for pathogens *Candida albicans*, *Plasmodium falciparum*, *Brucella melitensis*, *Burkholderia pseudomallei* and *Mycobacterium tuberculosis*. Each of these subsets contained some antigens as well as non-antigens. The other (6th) subset, on the other hand, contained only protective antigens collected from literature and public databases. This subset is referred to as *PAntigens*.

Any redundancy or considerable pairwise sequence similarity in the training dataset may hamper the quality of the model being trained. The cross-validation results may also get overestimated. To mitigate this concern, BLASTCLUST [15] was run with a 30% similarity threshold after combining the data from the five pathogens in the training set and redundant sequences were removed. The *PAntigens* set was similarly processed. It is possible, however, that some antigens in the *PAntigen* set may have redundancy with the proteins in the pathogens set. As such, proteins in the merged pathogen set with more than 30% sequence similarity with any protein in *PAntigens* were also removed. The composition of the training set, after all processing, is shown in Table 5.1.

It is noteworthy here that earlier works used much smaller datasets and did not have validated non-antigens. Instead, proteins selected at random and having very little sequence similarity with known protective antigens were tagged as non-antigens. In the benchmark dataset of [193], however, the non-antigens are curated by selecting proteins with low seroreactivity according to the protein microarray experiments.

The testing set was constructed from protein microarray data analysis for the pathogen *Bartonella henselae*. This dataset consists of 1463 proteins of which 73 were antigenic. The remaining 1390 were non-antigens.

For details of the microarray data analysis and protocols followed to prepare the benchmark dataset, the reader is referred to [193].

5.2.2 Protein Sample Representation

Like in previous chapters, we have resorted to Chou’s general formulation of PseAAC (described in Section 2.1.7) to represent the protein samples as fixed length feature vectors. The generalized PseAAC of a protein, as defined in Equation 2.2, is as follows:

$$P = [\psi_1 \ \psi_2 \ \dots \ \psi_u \ \dots \ \psi_\Omega]^T$$

The classical AAC is represented by subscripts $1 \leq u \leq 20$ and the subsequent features express sequence order information through one or more different schemes. The feature schemes that we have used are Dipeptides (Dip), Tripeptides, n -Gapped-Dipeptides (nGDip) and Position Specific n -grams (PSN). All these feature extraction techniques have already been described in Section 2.1.7.

AAC, Dip and Tripeptides derive from the generalized form of n -grams feature type, for $n = 1, 2$ and 3 . The total number of n -grams features we have extracted are 8409. For the nGDip feature type, we have considered upto 25 position gaps, thus producing a total of $25 \times 400 = 10000$ nGDip features. To avoid feature space explosion, we have extracted PSN features only for the first 10 positions of the primary sequence. This resulted in 14058

PSN features. We have thus extracted a total of $8409 + 10000 + 14058 = 32467$ features. These features can be represented in Chou's PseAAC as follows: For $1 \leq u \leq 20$, we have the amino acid composition in the feature vector. From $21 \leq u \leq 8409$, the dipeptide and tripeptide compositions are represented. From $8410 \leq u \leq 18409$, the features in this vector comes from the nGDip feature space. Finally, the PSN features construct the remaining portion of the PseAAC, from $18410 \leq u \leq 32467 = \Omega$.

5.2.3 Prediction Algorithm

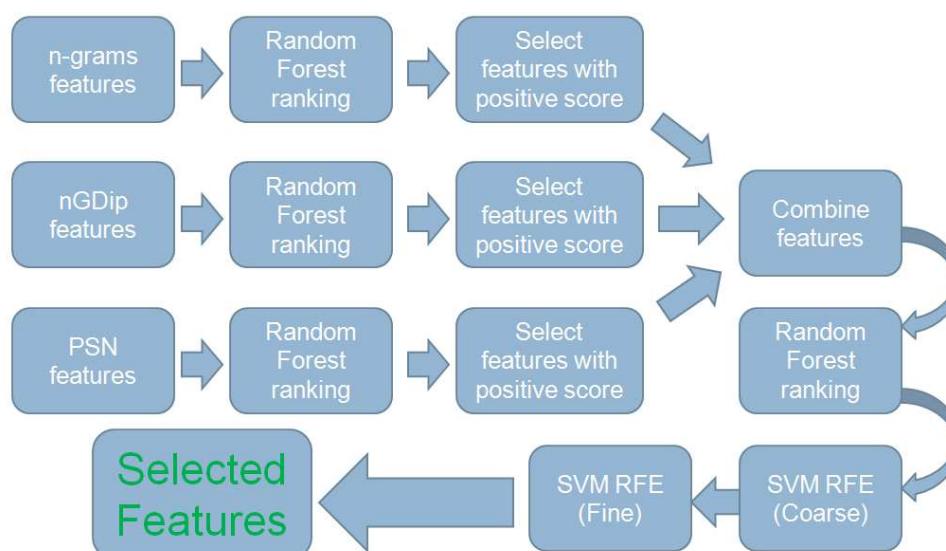
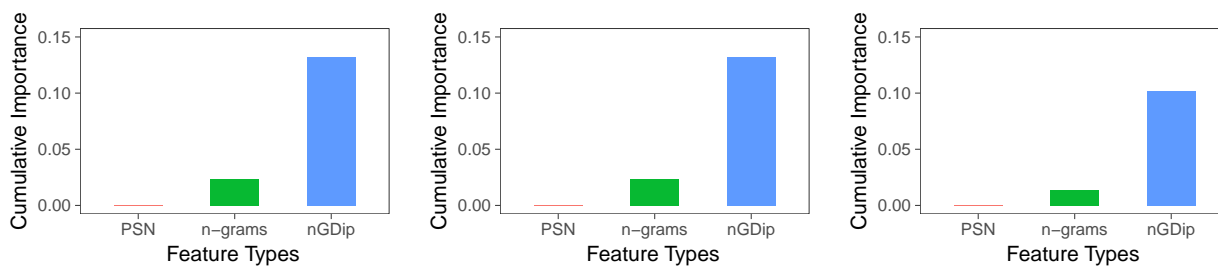


Figure 5.1: Steps in feature selection.

The first step in our prediction algorithm is *feature selection*. In this step we apply several techniques to reduce the size of the feature vector. As our protein samples are represented by a total of 32467 features, it would be computationally infeasible to train a classifier with the amount of computing power and memory we have at our disposal. The motivation behind the feature selection step obviously is to reduce the cardinality of the feature vector to a manageable size. Besides, a set of relevant features must be selected that is able to express the intrinsic difference between antigens and non-antigens. We have selected a suitable subset of the extracted features by applying a random forests algorithm based feature filtering, followed by multiple steps of SVM-RFE [118].



(a) Top 5500 features.

(b) Top 7500 features.

(c) All features.

Figure 5.2: Categorized feature importance based on random forests model based ranking. The aggregate ranking score is better for subsets of top-ranked features, compared to all features. PSN: Position Specific n -grams, n -grams: Combination of AAC, dipeptide and tripeptide composition features, nGDip: n -Gapped Dipeptides.

Feature Filtering Using Random Forests

For each feature, the *mean decrease in accuracy* computed from the random forests model can be used as a ranking score. The larger this value is for a feature, the more important that feature is in the context of the prediction task. However, as our feature space is quite large, getting the feature ranking using random forests itself is a difficult task. Our attempt to get the ranking scores of such a big feature space with the best server machine in our computing laboratories did not finish even after one month of execution.

To solve this problem, we followed the same steps that were used in the previous chapter and captured in Figure 4.2. For ease of reference, we have depicted the same pipeline in Figure 5.1 as well. Minor modifications were made in this process to account for the imbalance in the training dataset. Firstly, a random forests model is trained in each individual feature space. Using these models, features in each respective spaces are ranked locally. Generation of each of the three random forests models (and therefore the respective ranking scores) took less than a day. Based on these 3 rankings, we selected the features with positive score in each feature space. We thus obtained 3625 n -grams features, 5811 nGDip features and 1363 PSN features which totals to 10799 features. These 10799 features were re-ranked using another iteration of random forests algorithm.

In this ranking, 5196 features had positive mean decrease accuracy scores, 2283 features had 0 scores and remaining features had negative scores. This is why the cumulative scores of the features are almost identical for top 5500 and 7500 features, demonstrated in Figure 5.2. And they are superior compared to the cumulative scores when all features are considered, as observed in the same figure.

Feature Ranking Using SVM-RFE

After getting the feature ranking as above, the top-ranked 10000 features were re-ranked using SVM based Recursive Feature Elimination (SVM-RFE) [118] as follows. SVM was first run on the top-ranked 10000 features and the technique described in [118] was used to get a new ranking of these features. In the recursive step, 25 least ranked features were removed, SVM was run on the reduced feature space, and feature ranking was recomputed. The recursion was repeated applied until all the features are eliminated. Thus a new feature ranking is obtained. We call it the SVM-RFE (coarse) ranking.

Using this new ranking, different prediction models were constructed using the random forests algorithm. As we have an imbalanced dataset, we balanced it by undersampling the larger (negative) class randomly. After this step of random under sampling, we generated several prediction models by varying the number of features and compared their performances. We found the top 600 features to be most promising. (The exact details of how the number of features were varied is discussed in the Results section). To obtain a more reliable ranking of the top 600 features, a second round of SVM-RFE was applied in this feature space, but this time with steps of 1 feature elimination (instead of 25 features). We have referred to this new ranking as SVM-RFE (fine) ranking. Using this final ranking, we again explored several models of different feature count and found the model with 490 features to be the best model.

5.2.4 Predictor Evaluation

To evaluate the predictive performance of Antigenic, we have utilized jackknife cross-validation, 10-fold cross-validation test and independent test. These methods have already been briefly described in Section 2.1.8. In addition, another cross-validation technique, known as *leave one protein set out* was also used in this work. In this technique, one protein subset is left out and the predictor is trained with the remaining samples. Then the predictor performance is assessed using the subset that was left out. Thus each of the 6 subsets were used as testing set in 6 different iterations.

As performance metrics, we have used in this work accuracy, sensitivity, specificity and Matthew's Correlation Coefficient (MCC). We have also analyzed the Area Under Receiver Operating Characteristic Curve (auROC) and Precision Recall Curve (auPR). The reader is referred to Section 2.1.9 for details about these metrics.

Experimental Setup and Packages

We have conducted all our experiments using R language (version 3.2.3 or above). We used three different machines with the following configurations:

- A Desktop computer with Intel Core i5 CPU @ 3.00GHz x 4, Windows 7, 64-bit OS and 4 GB RAM.
- A Desktop computer with Intel Core i5 CPU @ 3.20GHz x 4, Ubuntu 16.04, 64-bit OS and 8 GB RAM.
- A server machine with Intel Xeon CPU E5-4617 0 @ 2.90GHz x 6, Ubuntu 13.04 64-bit OS, 15 MB L3 cache and 64 GB RAM.

As discussed earlier, random forests and Support Vector Machine (SVM) algorithms were used for ranking the features and learning the model. These are available from the R packages *randomForest* and *e1071* respectively. In both algorithms, default parameters setting was used. In particular, in random forests algorithm, the number of trees (*ntree*)

was restricted to 500, while the number of variables tried at each split (*mtry*) was set to the square root of the total number of features.

In addition to pre-installed packages in R, we have also used *ROCR* and *pracma* packages for performance analysis of our model. For plotting different graphs, we have leveraged *ggplot2* package. All of our source code, experimental results, cross-validation and independent datasets are available at: <https://github.com/srautonu/Antigenic>.

5.2.5 Predictor Availability

Antigenic is freely available as an R script at <https://github.com/srautonu/Antigenic>. Additionally, we have established a publicly accessible web server at <http://antigenic.research.buet.ac.bd/> to facilitate wide adoption. We hope our predictor will be beneficial to researchers working in the field of reverse vaccinology.

5.3 Results

In this section, we describe several experiments and analyze their results. We have conducted 10-fold cross-validation testing to assess the influence of number of features, combination of the feature extraction techniques etc. in the performance of our prediction model. We have also run experiments to compare Antigenic with VaxiJen and ANTI-GENpro, the two most widely used alignment-free predictors of protective antigens.

As the benchmark dataset is imbalanced, using it directly to learn the classifier may create a bias towards the majority class. Therefore we have balanced the dataset by random undersampling of the majority class, following [193]. Dittman et al. [81] has recently shown, however, when random forests algorithm is used to train the learning model, the increase in performance due to balancing the training set using random undersampling is not statistically significant. Hence, in many of our experiments we have used two different models - *Antigenic**, a model that was trained directly on the entire training set, and *Antigenic*, a model that was trained with a balanced (reduced by random undersampling) set.

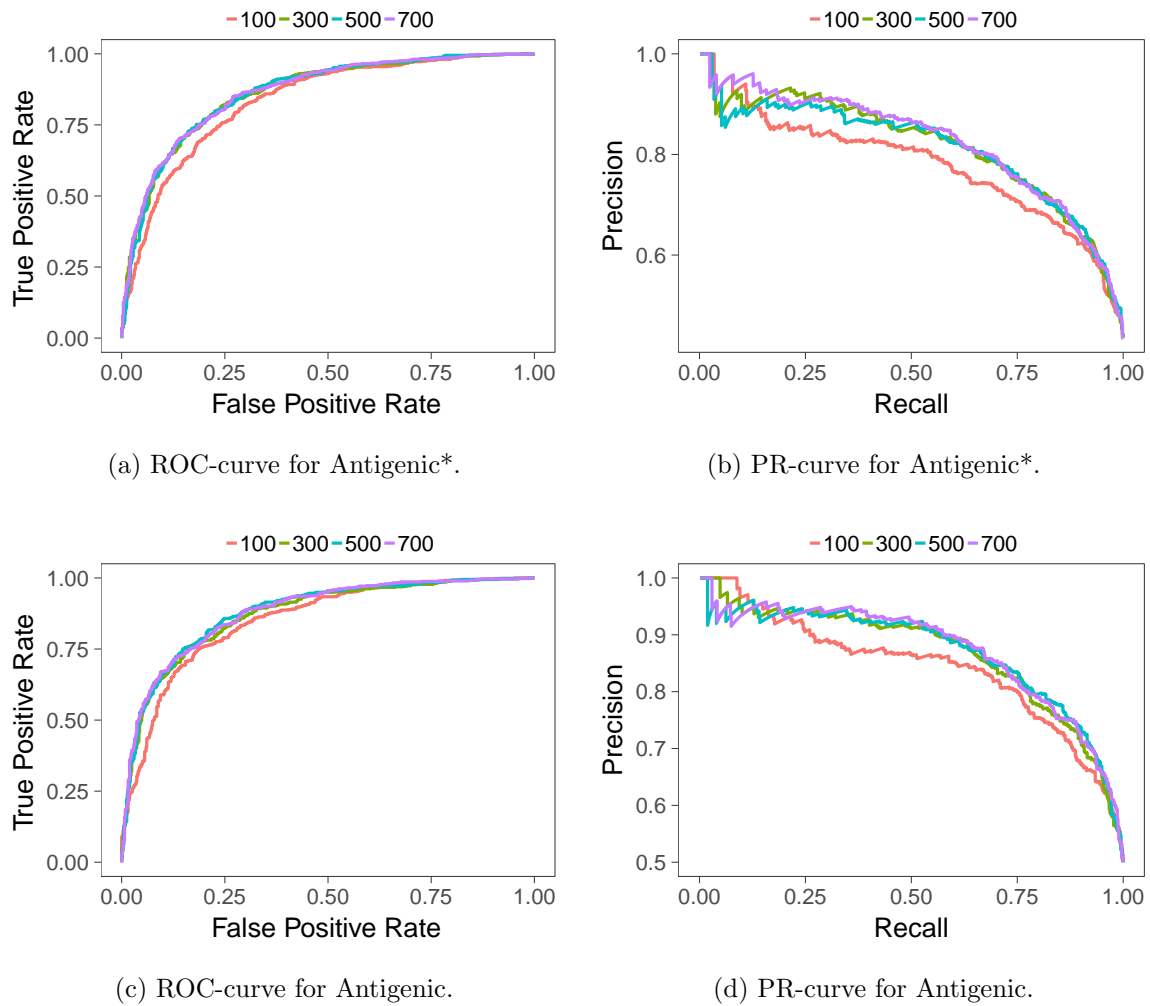


Figure 5.3: ROC and PR curves of prediction models with varying number of features, generated by 10-fold cross validation on the training dataset.

5.3.1 Impact of Number of Features

To find the ideal number of features, we ran several experiments varying the number of features and analyzed the impact on the classification model. The analysis was based on the various performance metrics discussed in the earlier section. In Figure 5.3, ROC and PR curves have been plotted for Antigenic and Antigenic*. In each case, 4 different curves are generated for models trained with best 100, 300, 500 and 700 features respectively. The closer a ROC curve is to the top-left corner of the graph, the better is the performance of the corresponding model. Therefore, it is clear that as the number of features is increased

beyond 100, the performance continues to improve at a good rate. The curves for 300, 500 and 700 features, on the other hand, lie very close to each other.

When the dataset is not balanced, ROC curve alone is not able to identify the relevance of selected features. Precision Recall (PR) curve is of more significance in this case [73]. We have plotted the PR curves for *Antigenic** and *Antigenic* in Figures 5.3b and 5.3d respectively. The closer a PR curve is to the top-right corner of the graph, the better is the performance of the corresponding model. From this analysis too, we observe that the performance increases with the increased number of features up to a certain point. As the number of features increases, the return on the performance gradually diminishes.

In Figure 5.4 we plot the auROC, auPR, accuracy, sensitivity, specificity and MCC of models that are created with varying number of top-ranked features. We have considered the unbalanced training set (Figures 5.4a, 5.4b and 5.4c) as well as training set balanced using random undersampling (Figures 5.4d, 5.4e and 5.4f). As both sets of experiments yielded similar pattern of results, we only describe the experiments with the balanced training set. At first a large feature space was explored. During this phase, the number of features that are added (removed) between experiments was also large. For example, Figure 5.4d is generated by starting with a model with 500 top-ranked features, accordingly to the SVM-RFE (coarse) ranking. We subsequently added 1000 next-ranked features in each iteration. Based on the curves, the feature space range [100, 1500] seemed promising. Therefore, more models were generated in this space, however the change of features in each step was made finer: 100 features. We thus examine 15 models, whose performances are recorded in Figure 5.4e. Moving along, we kept zooming in the interesting terrain of the feature space and increased our thoroughness in investigating the narrowed-down spaces. Figure 5.4f examines 60 models, with 10 feature increase steps. Based on the performance comparison of these models, the model built with 490 features was chosen to be our final classifier (*Antigenic*). Among the features, there were 181 *n*-grams, 170 nGDip and 139 PSN features. On the other hand, the final model trained with the unbalanced dataset, *Antigenic**, consisted of 500 features.

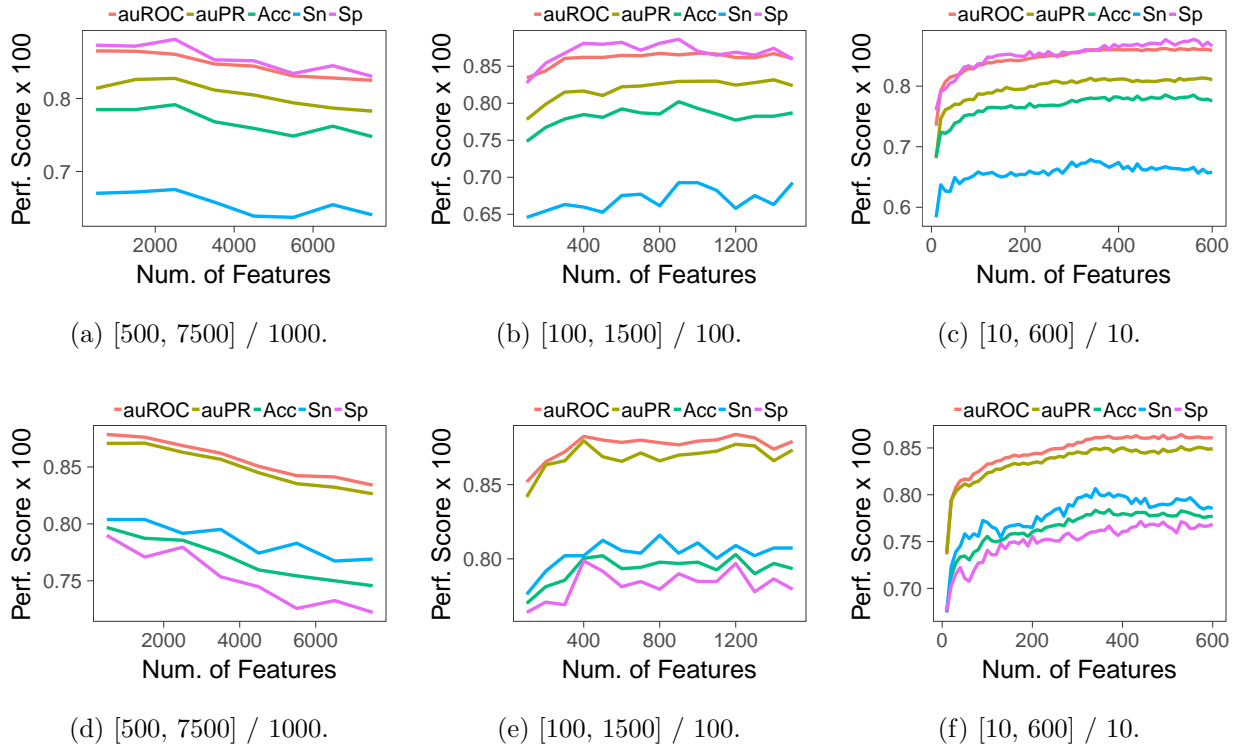


Figure 5.4: Area under ROC-curve (auROC), Area under PR-curve (auPR), Accuracy (Acc), Sensitivity (Sn), and Specificity (Sp) of models with varying number of features. The models were generated by 10-fold cross validation. For the top curves, the entire (unbalanced) training set was used (*Antigenic** models). For the bottom curves, training set was balanced with random undersampling (*Antigenic* models). The $[x, y]/z$ style annotation of each sub-figure means that, the experiment started with x top-ranked features. Then a model was trained with z more features and the performance scores were recomputed. This process continued until the feature count became y .

Another important observation comes out of these experiments. That is, balancing the training dataset helps make the classifier behave in a more balanced fashion. The model trained this way does not show any bias towards a particular class. *Antigenic** models, on the other hand, are clearly biased towards the negative class. The specificity is much higher compared to the sensitivity in these models. The overall accuracy is naturally dictated by the specificity and is somewhat misleading. Therefore, it is reasonable to claim that balancing the training dataset has a clear positive impact on the overall performance

of our predictor. While [81] claims that the data balancing results in an improvement that is not statistically significant, the authors there analyzed performance solely based on auROC. However, when dealing with data imbalance, analyzing the PR curve is more important [73]. From Figures 5.4b and 5.4e, the minimum, average and maximum increase in auPR, due to balancing the dataset, were respectively 4%, 6% and 8%.

Notably, in the above experiments, we have reported scores that are averaged over 5 different runs. As 10-fold cross-validation results may vary based on how the data is partitioned, 5 runs with different data partitioning were conducted and the average score was taken to have more confidence on the result.

5.3.2 Impact of Feature Extraction Techniques

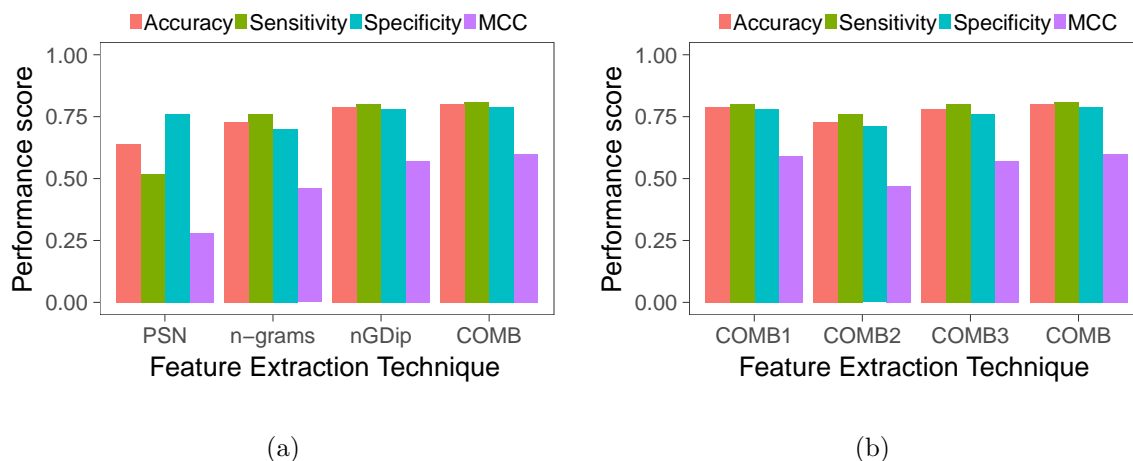


Figure 5.5: Performance of different feature extraction techniques. The results are obtained from 10-fold cross validation after balancing the training dataset with random undersampling. PSN: Position Specific n -grams, n -grams: Combination of AAC, dipeptide and tripeptide composition. nGDip: n -Gapped-Dipeptides. COM: Combination of all the feature extraction techniques. COM1: Combination of n -grams and nGDip. COM2: Combination of n -grams and PSN. COM3: Combination of nGDip and PSN.

To analyze the contribution of the different feature extraction techniques in building the model, we ran some experiments with the top 500 features. We trained a model

with these features after balancing the training set with random undersampling. Using the same balanced data, we trained 3 more models using top 500 features of the 3 individual feature extraction techniques separately. In Figure 5.5a, the accuracy, sensitivity, specificity and MCC values from these four models are compared. The nGDip feature extraction technique is a clear winner over the other two, while the combination of all performs slightly better than that.

In yet another experiment, we used combination of two feature spaces, leaving the other feature space out. Like before, we chose the top 500 features to construct the model. We compared the performance of the three generated models with that of the model created using the combination of all 3 feature spaces. The results are shown in Figure 5.5b. The composition of each combination is tabulated below:

Id.	Feature spaces	n -grams	nGDip	PSN
COMB1	n -grams, nGDip	258	242	-
COMB2	n -grams, PSN	283	-	217
COMB3	nGDip, PSN	-	269	231
COMB	n -grams, nGDip, PSN	186	181	133

The combination of n -grams and nGDip is the best performer among all the 2 feature space combinations. Adding the PSN feature space still adds value - the model constructed with combination of all 3 feature spaces is superior to models built with 2 feature space combinations in terms of each performance metric we have used.

5.3.3 Feature Importance Visualization

The importance of different features can be computed by permuting Out-of-bag (OOB) data of random forests algorithm. First, the prediction error on the OOB portion of the data is recorded for each tree. Afterwards, each predictor variable is permuted and the error is recalculated. The difference between the two errors is then averaged over all trees and then normalized by the standard deviation of the differences. This is called *Mean Decrease in Accuracy*. The larger this value is for a feature, the more important

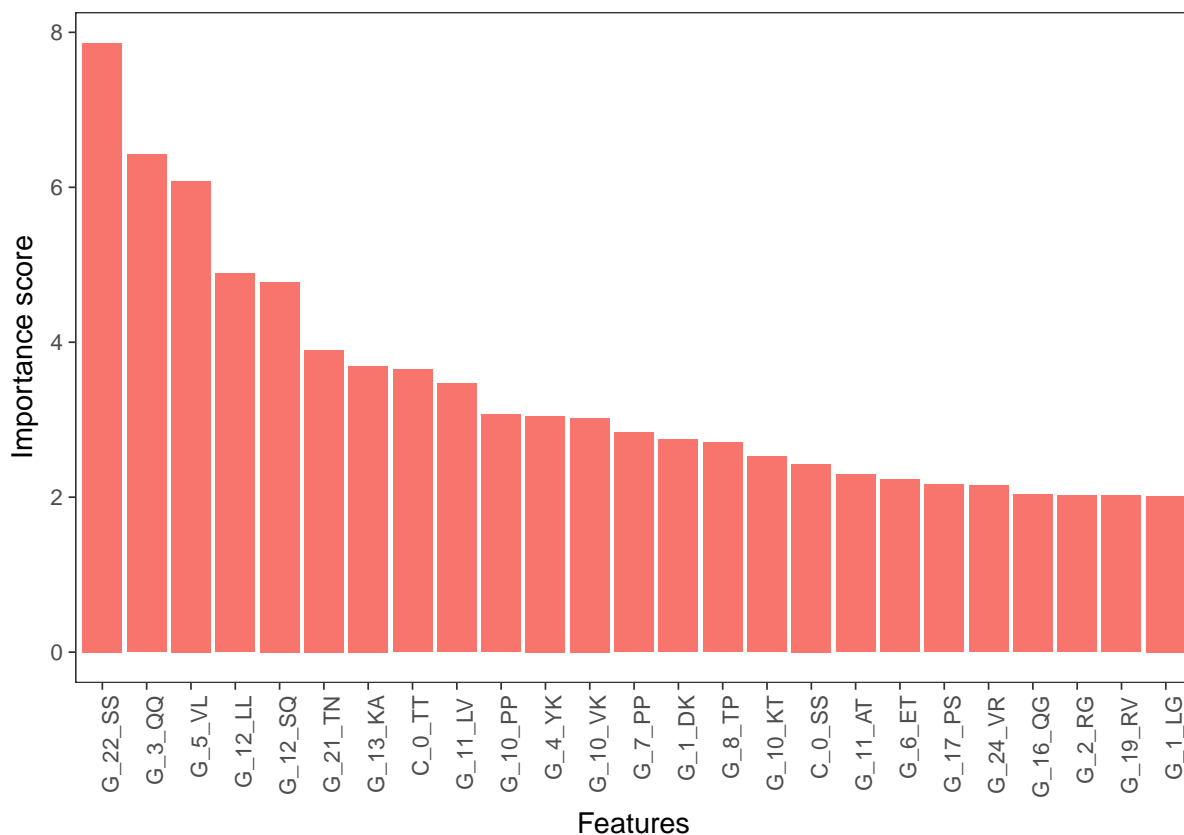


Figure 5.6: The importance score of top 25 features.

that feature is in the context of the prediction task. We have used the mean in decrease accuracy to rank the features and have demonstrated the importance score of top 25 features in Figure 5.6. The feature names are encoded following the convention that was used in Section 4.3.3.

5.3.4 10-fold Cross-validation Results

Table 5.2 records the performance of Antigenic and Antigenic* in 10-fold cross validation tests. In both cases, we used a decision threshold of 0.5. The results are averaged over 5 runs. The standard deviation (SD) in the different performance scores are shown after the \pm sign. The fact that the SD in each metric is very small vouches for the reliability of the average score. The performance of VaxiJen and ANTIGENpro, as obtained from [193] are

Table 5.2: Comparison of Antigenic with VaxiJen and ANTIGENpro based on 10-fold cross-validation. Although it is apparent that Antigenic* beats all the methods, including Antigenic, in accuracy and specificity, the lack of balance between the class-wise performance of the former is evident from the huge difference between the second and third column. The same lack of balance is present in Vaxijen, albeit more strikingly.

Method	Accuracy	Sensitivity	Specificity	MCC	auROC
Vaxijen	59.48 \pm 0.140	89.69 \pm 0.000	25.85 \pm 0.742	0.20 \pm 0.008	0.67 \pm 0.006
ANTIGENpro	75.51 \pm 0.992	75.88 \pm 1.937	75.14 \pm 1.480	0.51 \pm 0.020	0.81 \pm 0.012
Antigenic*	78.55 \pm 0.005	66.70 \pm 0.010	87.67 \pm 0.005	0.56 \pm 0.011	0.86 \pm 0.002
Antigenic	78.04 \pm 0.008	78.99 \pm 0.004	77.08 \pm 0.018	0.56 \pm 0.017	0.86 \pm 0.003

also recorded in the same table. The best values are highlighted in bold faced fonts. It is clear from this tabulated data that our models outperform the state-of-the-art predictors. Antigenic has superior accuracy, sensitivity, specificity, MCC and auROC compared to ANTIGENpro. It also performs better than VaxiJen in all metrics except for sensitivity. VaxiJen has a commendable sensitivity of almost 90%. However, it has poor specificity (26%), thus rendering itself as a predictor very much biased towards the positive class. This also means that it has poor precision. Antigenic, on the other hand, provides a balanced performance (79% sensitivity vs. 77% specificity). This is why Antigenic is better than Antigenic* as well, albeit both demonstrate similar performance in terms of MCC and auROC. But Antigenic* is biased towards the negative class (67% sensitivity vs. 88% specificity). Though it has the best accuracy among the lot, the accuracy is overestimated due to its bias towards the majority (negative) class, as it was trained with unbalanced dataset. For models trained with unbalanced datasets, the auPR is a good metric for performance comparison [73]. But, neither VaxiJen nor ANTIGENpro reports this metric. We nonetheless compare our own models in terms of auPR - while Antigenic* has an auPR of 0.81 \pm 0.004, Antigenic’s auPR is a superior 0.85 \pm 0.009. Therefore, based on the performance scores reported in Table 5.2 and the qualitative arguments given above, we can conclude that Antigenic is the best predictor.

5.3.5 Leave One Protein Set Out Cross-validation Results

Table 5.3: Comparison of accuracy between Antigenic and ANTIGENpro based on leave one protein set out cross-validation.

Test set	Test set size	ANTIGENpro	Antigenic*	Antigenic
PAntigens	213	81.60	29.77 ± 0.016	80.28 ± 0.019
Brucella	140	70.00	72.71 ± 0.034	73.00 ± 0.014
Burkholderia	10	66.00	70.00 ± 0.071	64.00 ± 0.055
Candida	6	66.67	43.33 ± 0.253	46.67 ± 0.075
Malaria	228	59.96	52.72 ± 0.008	51.93 ± 0.007
Tuberculosis	342	68.30	69.94 ± 0.019	70.00 ± 0.009

Magnan et al. [193] conducted another interesting cross-validation to assess the performance of ANTIGENpro. Recall that the training dataset consisted of antigens and non-antigens from 5 different pathogens and another subset of antigens obtained from literature. In this cross-validation approach, they left one subset out, trained the predictor with the remaining samples, then tested its performance using the subset that was left out. Thus each of the 6 subsets were used as testing set in 6 different iterations. The training set (i.e. the combination of remaining 5 subsets) was balanced using random undersampling before the training step. In addition, the testing set was also balanced using random undersampling. This was done to ensure a fair estimation of the predictor performance. An exception however was made when PAntigens subset was used as the testing set. Since this set does not have any non-antigens at all, undersampling cannot be done. Therefore, this set was used unaltered during testing. We have followed the same approach to measure the performance of our models and have a comparison with ANTIGENpro. We refer to this as *Leave one protein set out* cross-validation. In case of Antigenic, the training data was balanced using random undersampling. For the Antigenic* model, the training data was not balanced. In all the experiments, we kept the decision threshold at 0.5.

The results of this cross-validation approach are recorded in Table 5.3. For our models, the accuracy scores are averaged over 5 runs, with the standard deviation recorded after

the \pm sign. Once again, the small standard deviations gives confidence on the average scores. For ANTIGENpro, the average scores were obtained from [193], but the corresponding standard deviations were unavailable. When PAntigens is used as the testing set, ANTIGENpro has an formidable accuracy of 81.60%. Antigenic is not far behind, logging an average accuracy score of 80.28%. In fact, in 2 runs the accuracy scores were 82.63% and 81.69%, which are better than the reported value of ANTIGENpro. The performance on the PAntigens test set demonstrates that our featuring scheme, combined with the prediction algorithm, is able to predict protective antigens by learning solely from protein microarray data.

When the Brucella subset was used, Antigenic produced the best performance among the 3 predictors. For Burkholderia and Candida test sets, Antigenic* and ANTIGENpro respectively demonstrated the best performance. However, since these testing sets were very small, no conclusions should be made based on these results. For the Malaria test set, ANTIGENpro performed significantly better than both of our models. For the Tuberculosis testset, Antigenic is the winner with 70% accuracy, but ANTIGENpro is not far away (68.3% accuracy).

Another observation that we can make from these experiments is regarding the benefit of balancing the training dataset. In case of the PAntigens test set, Antigenic* has a poor accuracy of around 30%. As argued earlier, this model is quite biased towards the negative class. And since the testing set did not have any negative instances at all, the accuracy merely reflects the sensitivity of the predictor, which is poor. In this setting the negative class is twice as large as the positive class. On the other hand, when the full training dataset was used during 10-fold cross-validation (Table 5.2), the imbalance ratio was 1:1.3, yielding 67% sensitivity.

5.3.6 Jackknife Cross-validation Results

Neither VaxiJen nor ANTIGENpro has reported jackknife cross-validation results. However, for completeness and to enable comparison with future predictors, we report the

Table 5.4: Jackknife cross-validation performance of Antigenic* and Antigenic.

Method	Accuracy	Sensitivity	Specificity	Precision	MCC	auROC	auPR
Antigenic*	79.15	67.71	87.97	81.25	0.57	0.87	0.82
Antigenic	80.03	80.90	79.16	79.52	0.60	0.88	0.87

jackknife cross-validation performance of Antigenic and Antigenic* in Table 5.4. Like before, the best scores are highlighted in bold-face. Between the 2 models, Antigenic prevails as superior.

As discussed in Section 2.1.8, jackknife cross-validation always produces unique result, which is a key advantage of this technique. 10-fold cross validation results, on the other hand, may vary depending on how folds are constructed. In the 10-fold cross validation, Antigenic had the better sensitivity between the two. However, Antigenic* recorded superior accuracy and both method logged the same MCC and auROC. In jackknife testing, on the contrary, Antigenic demonstrated superior accuracy, sensitivity, MCC, auROC and auPR. Since the jackknife test cannot be biased by any particular way of splitting the data for cross-validation, the performance results obtained in this testing should therefore be given preference over the 10-fold cross-validation results. Also, since the training dataset in our case guarantees that pairwise sequence similarity is no more than 30%, any concerns of overestimation in jackknife approach is reasonably mitigated [61].

5.3.7 Independent Test Results

Table 5.5: Comparison of Antigenic with VaxiJen and ANTIGENpro based on independent testing.

Method	Accuracy	Sensitivity	Specificity
VaxiJen	39.71	72.60	37.99
ANTIGENpro	56.94	65.75	56.47
Antigenic*	61.18	61.64	61.15
Antigenic	46.27	76.71	44.68

In Table 5.5, independent testing performance of different predictors are recorded.

The entire proteome of *Bartonella Henselae* pathogen has been used for independent testing. As mentioned earlier, it contains 1463 proteins, of which only 73 are protective antigens. The FASTA file containing the proteome was easily uploaded to the VaxiJen server [7]. The prediction results were obtained in a response webpage within minutes of the query. The publicly available ANTIGENpro web tool [5] is less friendly for bulk queries. Single protein sequence can be pasted in a form and submitted. After some time the prediction results are provided via an email response. We wrote a simple Java program to automatically query the tool for each sequence of the proteome. Between queries, one minute waiting time was added so that the server does not get flooded with a lot of queries in a short period of time. The response emails were also processed through code written in Java. Getting the ANTIGENpro predictions for the *Bartonella* proteome this way took approximately 2 days. The results were obtained in a few minutes in case of Antigenic* and Antigenic.

In each case, we have considered the default class discriminating threshold. For VaxiJen, it is 0.4, for all others it is 0.5. As seen from Table 5.5, Antigenic is more sensitive than all other tools (even VaxiJen) at the default threshold. Its specificity is better than that of VaxiJen, but worse than that of ANTIGENpro and Antigenic*. Since the proteome is extremely imbalanced, the inferior specificity also impacts the overall accuracy. Surprisingly, Antigenic* demonstrates the most balanced performance in the independent testing.

Figure 5.7 shows the ROC curve for VaxiJen, ANTIGENpro, Antigenic* and Antigenic. Since Antigenic* is built using an imbalanced dataset, the PR curves of all the tools are also drawn in the same figure for better comparative assessment. Performance of ANTIGENpro is the best, while VaxiJen is the least performing. The area under ROC curve for ANTIGENpro, as recorded in Table 5.6, is marginally less than that of Antigenic* and slightly larger than that of Antigenic. It has the best auPR score (0.143). Antigenic* and Antigenic are not too far behind, however. VaxiJen, on the other hand, has a modest 0.074 unit area under PR curve.

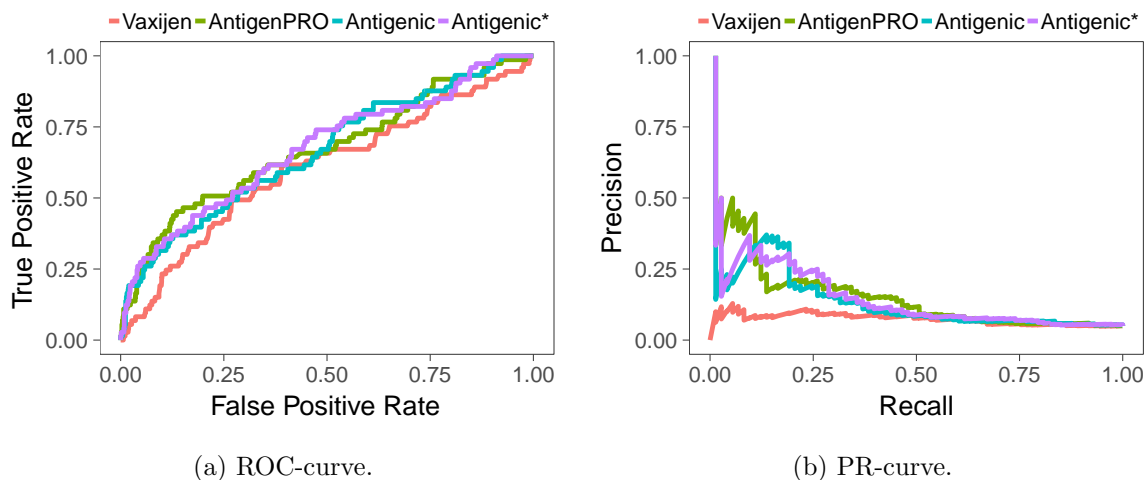


Figure 5.7: ROC and PR curves on the independent test for different prediction tools.

In 10-fold cross-validation testing, Antigenic demonstrated superiority over the state-of-the-art predictors. Its superior performance was also corroborated by the jackknife cross-validation testing. While it did not produce the best results in the independent testing, we have, as argued by Chou [61], preferred the cross-validation results over the independent test results and concluded Antigenic to be the superior predictor.

Table 5.6: Area under ROC and PR curves for different predictors on the Bartonella dataset.

Method	auROC	auPR
VaxiJen	0.603	0.074
ANTIGENpro	0.671	0.143
Antigenic*	0.674	0.136
Antigenic	0.662	0.125

The success of reverse vaccinology relies heavily on how efficiently and precisely the predictors can find protective antigens [142,234]. This is why the predictors are assessed in terms of yet another metric, known as *Enrichment*. It is the ratio of number of protective antigens among a top ranked subset to the number of protective antigens in the entire proteome. The expected enrichment of a random predictor is thus 1.0. For any good predictor, enrichment would be much higher than 1.0 in any top ranked subsets. Table 5.7

compares the enrichment of our predictors in independent testing (i.e. on the Bartonella dataset) with that of VaxiJen, ANTIGENpro and SignalP in top ranked 2%, 5%, 10% and 25% subsets. The data for SignalP was obtained from [193]. The data for VaxiJen and ANTIGENpro were computed based on the prediction scores, as obtained from their respective servers [5, 7]. While the scores for VaxiJen matched what was reported in [193], there was slight variation in the scores for ANTIGENpro. The enrichment for top ranked 5% turned out to be 4.1 instead of 4.4; and for top ranked 25% turned out to be 2.0 instead of 2.1. Both Antigenic* and Antigenic had superior enrichment in the top ranked 2% subset. In particular, Antigenic scored 6.9 which is much higher than ANTIGENpro’s 5.5. This means, if a practitioner ranks all the proteins of a new pathogen using Antigenic and selects only one protein at random from the top 2% for wet lab testing, his chance of identifying a protective antigen is almost 7 times higher than if he were to select one protein at random from the entire proteome. Therefore, our predictor seems quite suitable for wide adoption in reverse vaccinology based projects.

Table 5.7: Enrichment among top ranked proteins of Bartonella dataset, ranked by different predictors.

Method	SignalP	VaxiJen	ANTIGENpro	Antigenic*	Antigenic
Top ranked 2%	1.7	2.1	5.5	6.2	6.9
Top ranked 5%	2.7	1.6	4.1	4.9	3.8
Top ranked 10%	2.9	1.9	3.4	3.3	3.0
Top ranked 25%	2.2	1.6	2.0	2.2	1.8

5.4 Discussion

Antigenic demonstrated superior performance compared to other predictors in 10-fold cross-validation. It showed good performance in the leave one protein set out cross-validation as well. Like ANTIGENpro, it too demonstrated the ability to recognize protective antigens by learning the classifier from a training dataset that is prepared solely from the protein microarray data. In case of independent testing, it showed superior sen-

sitivity. It also showed better enrichment in the top ranked 2% subset. It did not hold the best auROC or auPR in the independent test. The best auPR was obtained by ANTI-GENpro, which also demonstrated good sensitivity. However, preference should be given on the cross-validation results over the independent test results in comparing different predictors [61]. Since Antigenic outperforms ANTIGENpro in the 10-fold cross-validation on the same training dataset, we consider Antigenic to be the superior predictor.

While VaxiJen has widely been adopted in various reverse vaccinology projects, it was able to correctly classify only 59.48% of the bacterial and viral proteins in the benchmark training dataset. The other antigens (around 25% of the antigens in the training set) could not be tested since no prediction model is available for these pathogens. This is a clear shortcoming of Vaxijen. Antigenic, on the other hand, provides a generic classification model for any pathogens and is able to demonstrate superior prediction performance.

Additionally, Antigenic has a fairly simple prediction model. The features it rely on are extracted from the protein's primary sequence directly. On the other hand ANTI-GENpro has a relatively complex model. It uses eight different feature sets, six of which are frequencies of amino acid monomers and dimers using three different amino acid alphabets. The remaining two feature sets are computed and predicted features. The computed features include sequence length, turn-forming residues fraction, absolute charge per residue, molecular weight, GRAVY Index [156], Aliphatic index [134]. Predicted features are obtained from external predictors like SSpro [49], DOMpro [50], ACCPro [49], and TMHMM [151]. Using these eight feature sets, forty distinct primary classifiers are then trained using one algorithm from Naive Bayes, C4.5, k-nearest neighbors, neural networks and SVMs. The 40 probability estimates thus obtained are then fed as input to a second stage SVM classifier. Perhaps it is because of the external dependency that ANTIGENpro limits query submissions to a single protein at a time. Antigenic, on the other hand, can handle large files with multiple protein sequences, making it convenient to use for whole proteome analysis.

The novelty of Antigenic lies in the addition of gapped dipeptides, tripeptide composition and PSN features into Chou’s general PseAAC. The combined application of random forests algorithm followed by SVM-RFE for feature selection is also a new approach in this prediction problem. Another distinguishing factor is that we explored a large feature space, comprising 32467 features and then selected 490 features for training the model. In contrast, VaxiJen used only 45 features, while ANTIGENpro used a total of 768 features.

5.5 Conclusion

In this chapter, we have presented Antigenic, a machine learning based predictor for protective antigens. We applied three different feature extraction techniques on a benchmark dataset that was primarily prepared from protein microarray data. Represented in a discrete model known as Chou’s general PseAAC, the proteins were then subjected to random forests and SVM-RFE methods to obtain a reliable ranking of the features. Finally, random forests algorithm was employed to learn a prediction model using a top-ranked feature subset. As the training dataset was not balanced, random undersampling was performed to balance the data. We trained models with both the unbalanced and balanced dataset and found the latter to be superior. Our approach outperforms state-of-the-art techniques according to different performance metrics in 10-fold cross-validation. The independent test results are also found to be satisfactory. Our predictor is available as an R script that can readily be applied to target protein sequences, without dependency on any other services or pre-processing. Antigenic is also available as a publicly accessible web based predictor. We hope the simple to use web interface, combined with the good performance, will lead to wide adoption of Antigenic. At the same time, we hope that our simple and lightweight framework will trigger further research using this in similar other domains.

Next chapter takes us to Part II of this thesis which is focused on phylogeny reconstruction. In particular, in the next chapter, titled *Gene Tree Estimation Using Absent Words*, we explore the idea of using minimal and relative absent words to compute the dis-

tance between two biological sequences. We also demonstrate how the pairwise distance matrix thus produced can be used to reconstruct the gene phylogeny.

Part II

Phylogeny Reconstruction

Chapter 6

Gene Tree Estimation Using Absent Words

An absent word with respect to a sequence is a word that does not occur in the sequence as a factor. A minimal absent word (MAW) is a word that is absent in a sequence but all its proper factors occur in that sequence. On the other hand, a relative absent word (RAW) is a word that occurs in a target sequence but is absent in a reference sequence. A RAW is minimal if none of its proper factors are RAW for the same pair of target and reference sequences. In this chapter we explore the idea of using MAW and RAW to compute the distance between two biological sequences. The motivation of our work comes from the potential advantage of being able to extract as little information as possible from large genomic sequences to reach the goal of comparing sequences in an alignment-free manner. For a collection of gene sequences, we demonstrate how the pairwise distance matrix thus produced can be used to reconstruct the gene phylogeny. We provide recommendations

Much of the material in this chapter is taken without alteration from the following paper.

- Rahman, M. S., Alatabbi, A., Athar, T., Crochemore, M., & Rahman, M. S. (2016). *Absent words and the (dis) similarity analysis of DNA sequences: an experimental study*. BMC research notes, 9(1), 186.

to use the best distance measure based on our analysis. In particular, our analysis reveals that the best performers are: the length weighted index of minimal RAW sets, the length weighted index of the symmetric difference of the MAW sets, and the Jaccard distance between the MAW sets. We also show that considering the reverse complement strands along with the input gene sequences during computation of the absent words improves the quality of the gene tree.

6.1 Introduction

Recently, the concept of minimal absent word (MAW) has been used to compute the distance between two species [39]. Similar effort has also been made to investigate the variation in number and content of minimal absent words within a species using four human genome assemblies [108]. This concept along with the related notions of absent words, also known as nullomers and forbidden words, have received significant attention in the relevant literature (e.g., [21, 25–27, 68, 98, 197, 198, 283]) and have been shown to be useful in applications like text compression [69, 70]. Perhaps the most significant use of this concept is in the field of computational biology. Hampikian and Andersen have studied nullomers, i.e., the shortest words that do not occur in a given genome, and primes, i.e., the shortest words that are absent from the entire known genetic data with a motivation to discover the constraints on natural DNA and protein sequences [121]. Acquisti et al. [8] have studied nullomers and the cause of absent words in the human genome. Herold et al. [127] have presented a method to compute the shortest absent words in genomic sequences. Pinho et al. [223] on the other hand focused on minimal absent words that form a set smaller than the set of absent words. Subsequently, Garcia and Pinho have studied four human genome assemblies from the perspective of minimal absent words [108]. Very recently, Silva et al. [247] coined the notion of minimal relative absent words (RAW) for differential identification of sequences that are derived from a pathogen genome but absent from its host. They applied this concept in analyzing Ebola virus genome from the 2014 outbreak [37] and discovered the presence of short DNA sequences

in the Ebola virus genome that appear nowhere in the human genome. The pathogen-specific signatures identified from such analysis can be useful for quick and precise action against the infectious agents.

The main focus of this chapter is to study and analyze possible functions that can be used with MAW and RAW sets to establish an alignment-free distance measure, which can then be utilized to develop a sequence-based gene tree estimation method. The study of gene phylogeny not only helps identify the historical relationships among a group of organisms, but also aids in other biological research such as drug and vaccine design, protein structure prediction and so on [170].

In sequence-based methods of gene phylogeny reconstruction, the input is a set of homologous sequences from different species. After obtaining an alignment of these sequences, different methods are applied to extract the phylogenetic information. In distance-based methods, a distance matrix is computed from the alignment that gives the pairwise distances among the sequences under consideration. This distance matrix is then used to estimate the gene tree using standard clustering methods or specially tailored methods. Examples of this approach include Neighbor Joining (NJ) [243], BIONJ [109], RapidNJ [248], FastME [78], QuickTree [128], Clearcut [91]. Another approach uses heuristics for either Maximum-Likelihood (ML) [96] or Maximum-Parsimony (MP) [101] which are two NP hard optimization problems. The most popular tools of gene tree estimation to date, RAxML [254,255] and FastTree [226,227], both use heuristics for ML, so does PhyML [113]. Yet another approach, Bayesian Markov Chain Monte Carlo (MCMC), produces not just a single gene tree but a probability distribution of the trees or aspects of the evolutionary history [33,124,158]. All these methods rely on sequence alignment, which is a time consuming task. Also, any error in the alignment significantly affects the downstream processes, resulting in poor estimation of the gene tree.

While the most popular distance-based methods compute the distance matrix from sequence alignment, it is technically possible to use these approaches without the alignment step, so long as the distance measure is able to reflect the number of substitutions per site,

which underlies classical alignment-based phylogeny reconstruction [122]. In fact, several such alignment-free gene tree estimation methods are found in literature [122,144,249,271]. Also, as mentioned in the beginning of this section, the concept of minimal absent word (MAW) has been used to compute the distance between two species. For example, in [39], Chairungsee and Crochemore have proposed a distance measure based on the set of minimal absent words and have used that distance measure to construct a gene tree among 11 species, following an experimental setup of Liu and Wang [183]. And, in [108], Garcia and Pinho have explored the potential of the minimal absent words from the perspective of similarities and differences among 4 human genome assemblies.

While the use of MAW and RAW sets as a distance measure seems interesting and useful, to the best of our knowledge there exists no attempt in the literature to identify the best approach to extract distance measures from these sets. Indeed, Chairungsee and Crochemore [39] chose to employ Length weighted index on the symmetric difference of two MAW sets but without any discussion on the rationale behind their choice. While it is likely that the potential advantage of MAW set would encourage researchers and practitioners to use this as a distance measure in the context of sequence comparison and phylogeny reconstruction, the lack of any directions on which approach to use with it may remain as an obstacle. This is where our current research work fits in. In this work we conduct an experimental study on the same setting of [183] and [39] to analyze and identify the best function to use with the MAW and RAW sets to infer the pairwise sequence distances. In our experiments we have analyzed all the functions that are already used in the literature. Additionally we have used some well-studied functions for the first time as a distance measure using minimal absent words. Table 6.1 lists and comments on the functions and concepts considered in this chapter. In the sequel, based on our analysis and comparison among the different methods studied, we have presented some recommendations with a goal to aid the researchers to select a suitable distance matrix for gene tree estimation in an alignment free manner.

Table 6.1: Functions used and compared in this chapter as distance measures.

Index	Comment
Length weighted Index (LWI)	Applied in [39] on the symmetric difference of MAW sets. Here we use LWI on symmetric difference and intersection of MAW sets, as well as on RAW sets.
Jaccard Distance	Used in this chapter.
Total Variation Distance (TVD)	Used in [108] to analyze similarity on 4 human genome assemblies.
GC Content	Used in [108] to analyze similarity on 4 human genome assemblies. Here we use GC Content on symmetric difference and intersection of MAW sets, as well as on RAW sets.

6.2 Methods

A string $x = x_1x_2 \dots x_n$ is a sequence of characters of length n from a finite alphabet Σ , i.e., $x_i \in \Sigma, 1 \leq i \leq n$. An empty string is denoted by ϵ . A string y is a factor or substring of a string x iff there exist strings u, v such that $x = uyv$; if $u \neq \epsilon$ or $v \neq \epsilon$, then, y is a proper factor of x . We use the term *word* and *string* synonymously. Below, we describe the concepts of MAW and different distance measures based on the MAW sets of a pair of sequences. We subsequently focus on another recently coined absent word based concept, known as relative absent word (RAW).

6.2.1 Minimal Absent Word (MAW)

An absent word in a string is a word that does not occur in the given string. More formally, a string y is an absent word in a string x if it is not a factor of x . Additionally, if all its proper factors are factors of x , then y is said to be a minimal absent word. For example, aaa , aba , and bbb are examples of minimal absent words for the string $x = abbaab$. But, $aaab$ is an absent word but not a minimal absent word of x . Given a string x , we will use

MAW_x to denote the set of minimal absent words of x .

Given a set, $\mathcal{S} = \{s_1, s_2, \dots, s_k\}$ of k sequences, we employ the following methodology:

Step 1: For each sequence $s_i, 1 \leq i \leq k$, we compute MAW_{s_i} .

Step 2: We compute distance matrix $\mathcal{M}_{\mathcal{S}}^{\mathcal{D}}$ for the set \mathcal{S} using a distance measure \mathcal{D} based on $MAW_{s_i}, 1 \leq i \leq k$. For all $1 \leq i, j \leq k$, we have $\mathcal{M}_{\mathcal{S}}^{\mathcal{D}}[i, j] = \mathcal{D}[s_i, s_j]$. Because the distance measure is symmetric, we only focus on the upper triangle of the matrix $\mathcal{M}_{\mathcal{S}}^{\mathcal{D}}$.

Step 3: We build a phylogenetic tree $\mathcal{T}_{\mathcal{A}}^{\mathcal{D}}(\mathcal{S})$ on the set \mathcal{S} based on the distance measure \mathcal{D} applying clustering algorithm \mathcal{A} on $\mathcal{M}_{\mathcal{S}}^{\mathcal{D}}$ for phylogeny reconstruction.

6.2.2 Distance Measures

We apply a number of distance measures discussed below. In what follows we will consider two sequences x and y and their MAW sets, MAW_x and MAW_y .

Length Weighted Index

In [39], the length weighted index (LWI) has been studied and experimented. There, this measure has been applied on the symmetric difference of the MAW sets. In our study we apply intersection operation as well. Formally:

$$LWI_{\Delta}(x, y) = \sum_{u \in MAW_x \Delta MAW_y} \frac{1}{|u|^2} \quad (6.1)$$

$$LWI_{\cap}(x, y) = - \sum_{u \in MAW_x \cap MAW_y} \frac{1}{|u|^2} \quad (6.2)$$

Here, Δ and \cap refer to the set symmetric difference and set intersection operations respectively. Note that, the intersection operation between two sets can be seen as a similarity measure and hence we use negation in Equation 6.2.

Jaccard Distance

Jaccard index is a statistical measure to use as a similarity coefficient between sample sets. Because we are interested in a distance matrix we use the following equation (based on Jaccard index) for computing the Jaccard distance.

$$J(x, y) = 1 - \frac{|MAW_x \cap MAW_y|}{|MAW_x \cup MAW_y|} \quad (6.3)$$

Total Variation Distance (TVD)

Garcia and Pinho [108] used total variation distance (TVD) to assess pairwise variance. The definition of TVD is as follows:

$$TVD(P, Q) = \frac{1}{2} \sum_i |P(i) - Q(i)| \quad (6.4)$$

where P and Q are two probability measures over a finite alphabet, and the term $\frac{1}{2}$ corresponds to the normalization by the two probability distributions [77]. This distance measure has values in the interval $[0, 1]$ with higher values implying greater dissimilarity or difference. To calculate $TVD(x, y)$, i.e., TVD between two sequences x and y we first count the number of MAWs in MAW_x and MAW_y for each word size and then transform this histogram in a normalized version that can be interpreted as a probability distribution. Subsequently, TVD is computed according to Equation 6.4.

GC Content

The above-mentioned indexes are based on the number statistics of the MAW sets. Inspired by the work of [108], we make an effort to suggest a measure that is more related to the content of the minimal absent words. In particular we focus on the compositional bias or GC content of the MAW sets. The GC content is the overall fraction of G plus C nucleotides in each set. We compute the GC content considering both symmetric dif-

ference and intersection. Assume that $NUM_\alpha(P)$ provides the number of a particular character $\alpha \in \Sigma$ in the members of the set P and $NUM_\Sigma(P)$ provides the number of all characters in the members of the set P . Then, formally:

$$GCC_\Delta(x, y) = \frac{NUM_G(MAW_x \Delta MAW_y) + NUM_C(MAW_x \Delta MAW_y)}{NUM_\Sigma(MAW_x \Delta MAW_y)} \quad (6.5)$$

$$GCC_\cap(x, y) = 1 - \frac{NUM_G(MAW_x \cap MAW_y) + NUM_C(MAW_x \cap MAW_y)}{NUM_\Sigma(MAW_x \cap MAW_y)} \quad (6.6)$$

6.2.3 Relative Absent Word (RAW)

The concept of Relative Absent Word (RAW) has been defined by Silva et al. [247] in the context of a target sequence x and a reference sequence y . Suppose $W_k(x)$ denotes the set of all length- k factors of x . Then $\overline{W_k(y)}$ denotes the set of all length- k words that are not present in y . Therefore, the set of all length- k relative absent words that exist in x but do not exist in y is defined as:

$$R_k(x, \bar{y}) = W_k(x) \cap \overline{W_k(y)} \quad (6.7)$$

A RAW is minimal if none of its proper factors are in the RAW set for the same pair of target and reference sequences. Formally, the set of length- k minimal relative absent words for target sequence x and reference sequence y is defined as:

$$M_k(x, \bar{y}) = \{\alpha \in R_k(x, \bar{y}) : W_{k-1}(\alpha) \cap M_{k-1}(x, \bar{y}) = \emptyset\} \quad (6.8)$$

Silva et al. [247] used RAW for differential identification of sequences that are derived from a pathogen genome (i.e., EBOLA virus) but absent from its host (i.e., Human). This inspires us to use RAW to compute the distance between two species in our study. Here we have used their software called EAGLE to compute the set of RAWs considering each

species in turn as the reference and the remaining species as targets. To elaborate, recall that we have a set, $\mathcal{S} = \{s_1, s_2, \dots, s_k\}$ of k sequences. For a particular pair of sequences $s_i, s_j \in \mathcal{S}$, we first compute RAW_{s_i, s_j} (RAW_{s_j, s_i}), i.e., the set of RAWs considering s_i (s_j) as the reference and s_j (s_i) as the target sequence. Then we compute the Length Weighted Index (LWI) (discussed above) of both RAW_{s_i, s_j} and RAW_{s_j, s_i} . This gives us two distance values for a particular pair of species. We then take the average of these two distance measures. Similarly, we also apply the *GC* content measure on the RAW sets.

6.2.4 Gene Tree Estimation Algorithms

A gene tree represents the evolution of a particular gene within a group of species (taxa). In sequence-based methods of gene tree reconstruction, the input is a set of homologous sequences. The distance based phylogeny reconstruction methods start by computing a matrix that gives us the pairwise distances between the sequences under consideration. This distance matrix is then used to estimate the tree using standard clustering methods or specially tailored methods to reconstruct the phylogeny from the distance matrix. The distance measures described above have also been used to reconstruct gene phylogeny using two well-known methods, namely, Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [258] and Neighbor Joining (NJ) [243]. UPGMA builds the tree by clustering similar taxa iteratively, and it works by building the phylogenetic tree bottom up from its leaves. NJ method on the other hand starts with a star tree as the initial and construct a modified distance matrix in which the separation between each pair of nodes is adjusted on the basis of their average divergence from all other nodes. Very briefly, the tree is constructed by linking the least-distant pair of nodes in this modified matrix.

6.3 Experiments

We have used the same datasets used in [183] and [39]. In particular, we have conducted our experiments on the first exon sequences of β -globin genes from 11 species, namely,

Human, Goat, Gallus, Opossum, Lemur, Mouse, Rabbit, Rat, Bovine, Gorilla, and Chimpanzee. Because the gene family of β -globin has a significant biological role in oxygen transport in organisms, it is used to analyze DNA and the first exon of the β -globin gene is an example for many DNA studies instead of computing similarity/dissimilarity of the whole genomes [39]. Inspired by the experimental setup of Garcia and Pinho [108], we consider two scenarios: the original sequence itself and the original sequence concatenated with its reversed complement (artificial words across the boundary between both sequences are ignored). The former will be referred to as the noRC setting and the latter as the RC setting. The motivation for using the reverse complement is to take into consideration words that might occur in the reverse complement strand but that might be absent from the direct strand.

We have used the algorithm of [21] to compute the MAW sets using their implementation, which is available at: <http://github.com/solonas13/maw>. In this implementation, there are two parameters k and K , respectively representing the minimum and maximum length of MAWs to be generated. Since we wanted to generate all possible MAWs, we set k to 2 and K to one less than the length of the sequence. To compute the RAW sets, we have used EAGLE software [247] available at: <http://bioinformatics.ua.pt/software/eagle/>. For each pair of sequences, we generated RAWs of length between 2 and 28. The code to compute the distance matrices and analyze the results were written in C++ language and can be found at: <https://github.com/srautonu/AWordS>. We have also implemented a related web-based tool with limited capacity here: <http://77.68.43.135/AWordS>. It is planned that this web-tool will be improved with more functionalities in near future.

We have considered the four distance measures described in Section 6.2.2 based on the MAW sets. For a pair of sequences, LWI and GC content have been applied on the symmetric difference as well as the intersection of the MAW sets. We have also applied these 2 measures on the RAW sets. With noRC and RC settings, this gives us a total of 16 distance matrices. All the distance matrices are given in Appendix A.

6.4 Results

Following the methodology of [183] we have carefully analyzed the computed distance matrices based on the real biological phenomena that are also considered in [183]:

- It is believed that Gorilla and Chimpanzee are most similar to Human [REL 1];
- Similarly, among these 11 species, Goat and Bovine should be similar [REL 2] as are Rat and Mouse [REL 3];
- Gallus and Opossum should be remote from the other species because Gallus is the only non-mammalian representative in this group [REL 4] and Opossum is the most remote species from the remaining mammals [REL 5];
- Besides gallus and Opossum, lemur is more remote from the other species relatively [REL 6].

We have analyzed the distance measures based on the above-mentioned 6 expected relations (REL 1 - REL 6). Among these 6 relations we give higher importance on REL 1 through REL 3 in the sense that when all of these are captured we look into the rest for further comparison. Below we discuss several interesting points from our analysis.

- As is evident from our analysis, unfortunately, the GC Content measure does not do very well in comparison to the other metrics despite that it is more related to the content of the minimal absent words. In particular, in most cases this measure is unable to capture the expected relationships (REL 1 - REL 6) mentioned above. However, despite the overall relative poor performance, except for the cases when intersection operation has been used, GC Content measure is at least able to capture the close relation among Human, Gorilla and Chimpanzee, i.e. REL 1. For intersection operation however, GCC fails miserably to capture any of the important relationships among REL 1 REL 2 and REL 3.

Table 6.2: The distance matrix based on the Length Weighted Index on RAW sets (on RC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		23.39	26.94	28.34	27.82	23.49	19.31	27.88	4.77	21.60	7.26
goat			28.71	24.16	25.89	25.52	24.33	27.43	21.77	8.73	24.26
opossum				29.55	31.23	29.21	26.69	30.52	26.90	28.16	28.44
gallus					28.66	30.22	26.27	30.89	28.25	26.21	30.51
lemur						30.21	27.63	30.96	27.77	25.91	30.27
mouse							24.09	26.43	20.98	23.17	23.29
rabbit								29.19	19.02	22.28	21.50
rat									28.37	27.95	30.21
gorilla										19.48	9.62
bovine											21.97
chimp											

- The total variation distance also fails to be highly impressive. It has been able to capture some of the relations but not all. However, it definitely seems better than the GC Content measure. In particular, it has been able to capture REL 1 and in most cases it also captures REL 2. However, it fails to capture REL 3 in both RC and NoRC settings.
- Among the distance measures one of the best (if not the best) performers turns out to be the length weighted index applied on the RAW sets. The result is better when RC setting is used. In particular, Table 6.2 (also see Table 6.3) has all the desired relations (REL 1 through REL 6) mentioned above.
- Jaccard distance has also turned out to be a very good measure in our experiments. In particular, in Table 6.4 (also see Table 6.5) we can identify almost all desired relations (REL 1 through REL 6).
- Length Weighted Index (LWI) for Symmetric Difference under the RC setting also performs very well in conserving relations REL 1 through REL 5. This measure seems quite good under the NoRC setting as well. However, it is worth-mentioning that under the latter setting it fails to capture the close relation between Rat and Mouse (REL 3).

Table 6.3: The sorted list of each species from a particular species (left most column of each row) according to the computed distance based on the Length Weighted Index on RAW sets (on RC setting).

human	→gorilla	→chimp	→rabbit	→bovine	→goat	→mouse	→opossum	→lemur	→rat	→gallus
goat	→bovine	→gorilla	→human	→gallus	→chimp	→rabbit	→mouse	→lemur	→rat	→opossum
opossum	→rabbit	→gorilla	→human	→bovine	→chimp	→goat	→mouse	→gallus	→rat	→lemur
gallus	→goat	→bovine	→rabbit	→gorilla	→human	→lemur	→opossum	→mouse	→chimp	→rat
lemur	→goat	→bovine	→rabbit	→gorilla	→human	→gallus	→mouse	→chimp	→rat	→opossum
mouse	→gorilla	→bovine	→chimp	→human	→rabbit	→goat	→rat	→opossum	→lemur	→gallus
rabbit	→gorilla	→human	→chimp	→bovine	→mouse	→goat	→gallus	→opossum	→lemur	→rat
rat	→mouse	→goat	→human	→bovine	→gorilla	→rabbit	→chimp	→opossum	→gallus	→lemur
gorilla	→human	→chimp	→rabbit	→bovine	→mouse	→goat	→opossum	→lemur	→gallus	→rat
bovine	→goat	→gorilla	→human	→chimp	→rabbit	→mouse	→lemur	→gallus	→rat	→opossum
chimp	→human	→gorilla	→rabbit	→bovine	→mouse	→goat	→opossum	→rat	→lemur	→gallus

Table 6.4: The distance matrix based on the Jaccard distance on MAW sets (on RC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		0.70	0.82	0.80	0.76	0.70	0.61	0.80	0.15	0.69	0.26
goat			0.84	0.74	0.74	0.77	0.77	0.79	0.69	0.36	0.71
opossum				0.85	0.87	0.91	0.84	0.90	0.82	0.85	0.82
gallus					0.81	0.82	0.79	0.85	0.80	0.81	0.80
lemur						0.83	0.81	0.81	0.76	0.72	0.77
mouse							0.78	0.78	0.64	0.74	0.68
rabbit								0.81	0.63	0.75	0.65
rat									0.80	0.82	0.82
gorilla										0.67	0.15
bovine											0.69
chimp											

Table 6.5: The sorted list of each species from a particular species (left most column of each row) according to the computed distance based on the Jaccard distance on MAW sets (on RC setting).

human	→gorilla	→chimp	→rabbit	→bovine	→mouse	→goat	→lemur	→gallus	→rat	→opossum
goat	→bovine	→gorilla	→human	→chimp	→lemur	→gallus	→rabbit	→mouse	→rat	→opossum
opossum	→chimp	→human	→gorilla	→rabbit	→goat	→gallus	→bovine	→lemur	→rat	→mouse
gallus	→goat	→rabbit	→human	→gorilla	→chimp	→bovine	→lemur	→mouse	→opossum	→rat
lemur	→bovine	→goat	→gorilla	→human	→chimp	→rabbit	→rat	→gallus	→mouse	→opossum
mouse	→gorilla	→chimp	→human	→bovine	→goat	→rat	→rabbit	→gallus	→lemur	→opossum
rabbit	→human	→gorilla	→chimp	→bovine	→goat	→mouse	→gallus	→lemur	→rat	→opossum
rat	→mouse	→goat	→human	→gorilla	→rabbit	→lemur	→chimp	→bovine	→gallus	→opossum
gorilla	→human	→chimp	→rabbit	→mouse	→bovine	→goat	→lemur	→gallus	→rat	→opossum
bovine	→goat	→gorilla	→human	→chimp	→lemur	→mouse	→rabbit	→gallus	→rat	→opossum
chimp	→gorilla	→human	→rabbit	→mouse	→bovine	→goat	→lemur	→gallus	→opossum	→rat

- In general it seems that the results are better for the RC setting which is expected because this setting takes into consideration words that might occur in the reverse complement strand but that might be absent from the direct strand.

6.4.1 Estimated β -globin Gene Trees

As discussed in Section 6.2.4 all the distance measures analyzed in this chapter have been used to estimate gene trees using two well-known methods, namely, Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [258] and Neighbor Joining (NJ) [243]. The reconstructed β -globin gene trees based on the different distance measures are presented in Appendix A. Here we only present the gene trees reconstructed using Neighbor Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the RAW sets (Figure 6.1), the length weighted index of the symmetric difference of the MAW sets (Figure 6.2) and the Jaccard distance (Figure 6.3) considering RC setting. Notably, these three indexes are the best performers according to our analysis. Finally, in Figure 6.4 we present the phylogenetic tree constructed using NJ algorithm on the distance matrix proposed in [183] for a visual comparison.

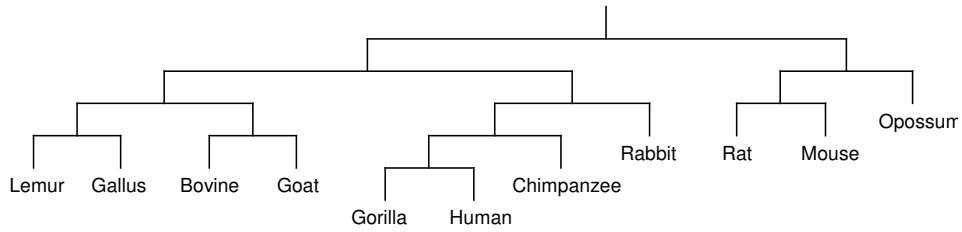


Figure 6.1: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the RAW sets (on RC setting).

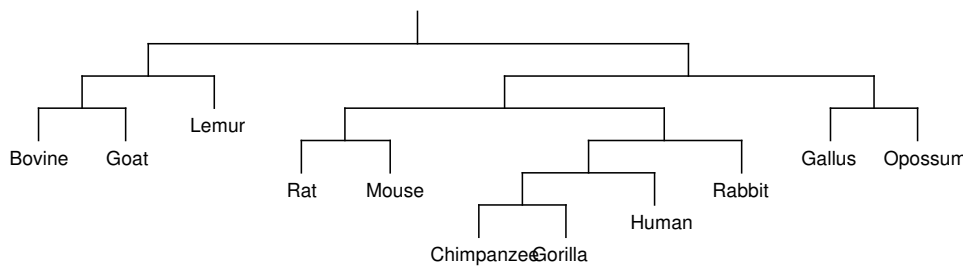


Figure 6.2: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on Symmetric Difference of the MAW sets (on RC setting).

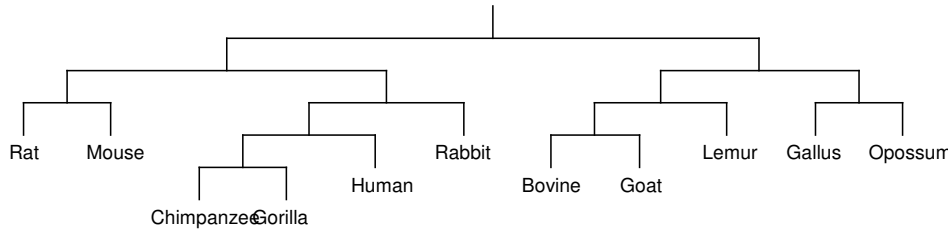


Figure 6.3: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Jaccard distance on the MAW sets (on RC setting).

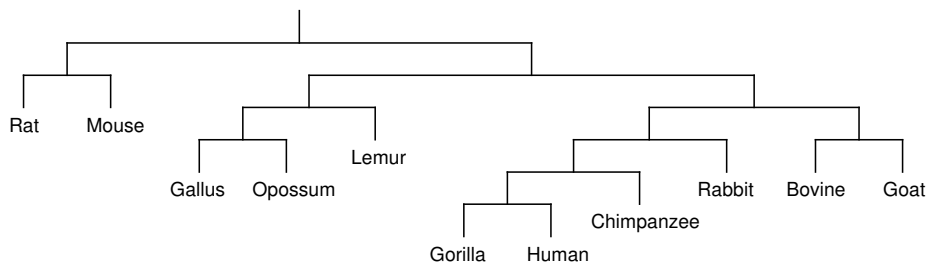


Figure 6.4: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix of [183].

6.5 Conclusion

In this chapter we have experimentally studied a number of distance measures based on the concept of absent words to estimate the distance among different biological sequences. Our main motivation has been to provide the research community an alignment free method of gene tree estimation. Our work is inspired by the previous work with similar goals as in [39] and [183]. In the sequel we present a comparison among the different distance functions we have studied with a goal to aid the researchers in choosing a suitable method for such dissimilarity analysis and phylogeny reconstruction. Based on our analysis we recommend 3 distance measures. These are Length weighted index (LWI) applied on the RAW sets, LWI applied on Symmetric Difference of the MAW sets and Jaccard distance of the MAW sets. When computing these distance measures, the MAW and RAW sets should be extracted considering both the direct and the reverse complement strand of the gene sequences. This is supported by the natural assumption that this setting takes into consideration words that might occur in the reverse complement strand but that might be absent from the direct strand. Finally, Neighbor Joining algorithm should be applied on the distance matrix thus computed to capture the gene phylogeny. We have established a publicly accessible web interface that can be used to compute the different distance measures described in this chapter. The resulting distance matrices can be used with any clustering algorithm of researchers' choice to produce gene trees in an alignment free manner. We hope the appeal of our alignment free approaches, combined with the public availability of our tool, will attract researchers to apply our methods in their respective projects.

In the next chapter, *Species Tree Estimation using DCM Boosted QFM*, we focus on species tree estimation, particularly when the gene tree discordance is modeled by incomplete lineage sorting (ILS) or deep coalescence.

Chapter 7

Species Tree Estimation using DCM Boosted QFM

In the previous chapter, we developed a distance based scheme for gene tree estimation. In this chapter, we shift our focus to species tree estimation, particularly when the gene tree discordance is modeled by incomplete lineage sorting (ILS) or deep coalescence. When the genes evolve down different tree topologies due to ILS, coalescent-based methods need to be applied to estimate the species tree. These methods provide statistical guarantees of returning the true tree with high probability, as the number of genes in the study is increased. One such method is Quartet FM (QFM), which is highly accurate but does not scale to large number of taxa. In this chapter, we apply disk-covering methods (DCMs) to boost the scalability and performance of QFM. Experiments with a simulated dataset of 37 taxa shows that DCM boosted QFM outperforms ASTRAL, a highly popular, accurate and fast coalescent-based species tree estimation method that is statistically consistent under the multi-species coalescent model.

The results presented in this chapter are not published yet.

7.1 Introduction

In a typical approach to species tree estimation, one would use multiple loci, concatenate alignments for each locus into a super-matrix and then use it to estimate the species tree [23]. However, when the genes evolve down different tree topologies due to gene duplication and extinction, horizontal gene transfer, or incomplete lineage sorting (ILS), this approach can return incorrect trees with high confidence [153]. A superior approach to the concatenated analyses is the summary methods that take the reason of discordance into account. In particular, when the discordance is modeled by ILS, the coalescent-based summary methods provide statistical guarantees of returning the true tree with high probability, as the number of genes in the study is increased [124, 152, 158, 180–182, 199, 200, 202, 236, 292]. One of these coalescent-based approaches, QFM [236], is the focus of this chapter. Here we make an attempt to improve the running time of QFM to allow it to process larger number of species or taxa.

QFM or Quartet FM [236] is a quartet-based phylogeny reconstruction algorithm. QFM employs a divide and conquer approach. At each recursive step of the divide phase, the input set of taxa is partitioned into 2 disjoint subsets using a heuristic bipartition algorithm that is inspired by the Fiduccia and Mattheyses (FM) bipartition algorithm [99]. The algorithm starts with an initial partition and applies a heuristic search iteratively to find a better partition. Each partition is scored by the number of satisfied quartets (i.e. quartet support), less the number of violated quartets. To aid in combining 2 trees in the conquer phase, whenever a bipartition is formed, both the partitions are augmented with a unique dummy (artificial) taxon. With each bipartition of taxa, the set of quartets is also divided into 2 subsets accordingly. During recursion, if the quartet set becomes empty or the number of taxa becomes less than 4, then a depth one tree (i.e. a star) is returned. In the conquer phase, as the recursion unwinds, at each step, two trees are rerooted at the dummy taxon. Then the dummy taxon is removed from each tree and the two roots are joined by an internal edge. Experiments conducted by Reaz et al. [236] with both simulated and biological datasets demonstrate that QFM is highly accurate.

However, QFM does not scale well as the number of taxa in the study increases. To mitigate this, the authors in [236] sampled the quartets from the input set of gene trees, instead of considering all the quartets. In a phylogenetic study of n species (taxa), the number of quartets obtained from an input gene tree (with no missing taxa) will be $\binom{n}{4}$, which is on the order of n^4 . However, Reaz et al. [236] samples only $\mathcal{O}(n^{2.8})$ quartets. We, on the other hand, take a different approach in an effort to scale QFM. Rather than sub-sampling the input set of quartets, we apply *disk-covering methods* which was shown to improve the run time as well as accuracy of MP-EST in [23].

Disk-covering methods (DCMs) are meta-methods that employ divide-and-conquer and iteration to boost the performance of the existing phylogenetic reconstruction methods [132,133,205,239]. In the first step, the dataset is decomposed into overlapping subsets of taxa. Then species trees are estimated on these subsets using a coalescent-based species tree method. Finally the species trees on the subsets of taxa are merged to get a tree on the full set of taxa. In this study, we will apply DCM to boost the performance of QFM and compare its performance with ASTRAL [292], which is a fast, accurate and highly popular coalescent-based species tree estimation method.

7.2 Methods

In order to boost QFM using the DCM approach, we need to first decompose the dataset into overlapping subsets of taxa. We use DACTAL [208] based decomposition with a target subset size of 15 and padding size of 4. The target subset size represents the maximum size of a subset, while the padding size represents the number of overlapping taxa in the subsets. Both these parameters are treated as targets rather than hard constraints.

The DACTAL decomposition requires an initial (guide) tree on the full set of taxa. We use Matrix Representation with Parsimony (MRP) [22,231] to obtain the guide tree. MRP is a widely used supertree method for phylogeny reconstruction that is fast but less accurate. MRP encodes all the small trees into a matrix using the characters 0, 1 and ?. Then it uses Maximum-Parsimony (MP) [101] to get a tree from the data matrix.

In the second step, species trees are estimated on these subsets using QFM. For each subset, we restrict the input gene trees to the species present in the subset (each such gene tree is called a subset gene tree). Then the quartets induced by these subset gene trees are extracted and weighted by their respective frequencies. The set of quartets, along with their weights, are then passed to QFM to estimate the species tree on the taxa subsets.

In the final step, we combine the subset species trees using SuperFine+MRL [209]. Thus a species tree on the full set of taxa is obtained. This can now be used as the new guide tree and the entire process can be repeated. Several iterations of these steps can be performed, with the species tree produced in one iteration being used as the guide tree for the next iteration. We experiment with 2 and 5 iterations, following the methodology of [23].

The quality of the species tree produced in each iteration is measured in terms of quartet support score, as defined in Section 2.2.9. The species tree with the highest quartet support, across all the iterations of boosting, is returned as the final estimated tree.

7.3 Experiments

We compare the performance of boosted versions of QFM [236] with the latest version of ASTRAL [199, 200, 292] on a collection of simulated datasets. We choose to compare our results with ASTRAL since it has already been shown to be more accurate than MP-EST [181] and BUCKy-pop [158] under different model conditions [199]. ASTRAL has a clear advantage over concatenation when ILS levels are at least moderate, while concatenation having an advantage when ILS levels are low.

We measure the tree error using the missing branch rate, also known as the false negative (FN) rate. As defined in Section 2.2.9, FN rate is the percentage of the internal edges in the model tree that are missing in the estimated tree.

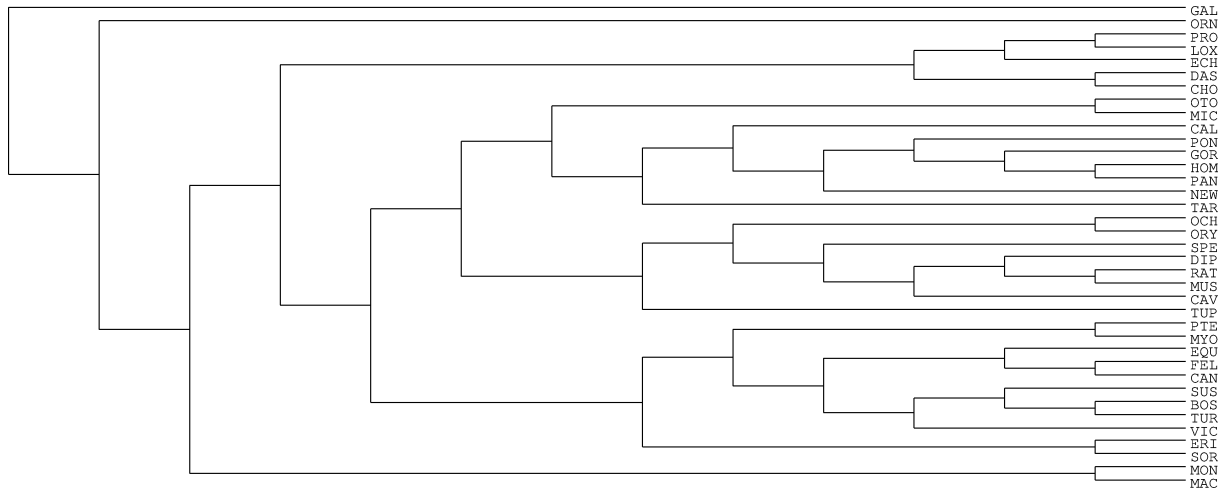


Figure 7.1: The model species tree for the 37-taxon mammalian dataset of [252].

7.3.1 Mammalian Simulated Datasets

We have used the mammalian simulated dataset that was prepared by Mirarab et al. [199] and was also used in [23]. For completeness, we provide a brief description here on how the dataset was constructed.

The simulated dataset was constructed based on a 37-taxon mammalian dataset with 447 genes that was studied by Song et al. [252]. Mirarab et al. [199] re-analyzed the data and observed 2 clear outliers in terms of pairwise distance of the gene trees. They also identified 21 genes with mislabeled sequences, which was subsequently also confirmed by the authors of [252]. The outliers as well as the mislabeled sequences were therefore removed and the dataset with the remaining 424 genes were then used to estimate a species tree using MP-EST. This tree was then used as a model species tree, with branch lengths in coalescent units. This tree is shown in Figure 7.1.

The branch lengths of the model species tree were rescaled to vary the amount of ILS and create different model conditions. Shorter branch lengths increases the amount of ILS and vice versa. The model condition with reduced ILS was created by uniformly doubling (2X) the branch lengths, and two model conditions with higher ILS were generated by uniformly dividing the branch lengths by two (0.5X) and five (0.2X). The amount of ILS obtained without adjusting the branch lengths is referred to as “moderate ILS”. Using

the multi-species coalescent model, gene trees were generated from each model species tree. The gene tree branch lengths were then modified to deviate from the strict molecular clock, and sequences were simulated down each gene tree under the GTRGAMMA model. Maximum likelihood (ML) gene trees were estimated on each sequence alignment using RAxML [254] under the GTRGAMMA model, with 200 bootstrap replicates to produce bootstrap support on the branches.

In the biological data, the average bootstrap support (BS) was 71%. Therefore, the sequence lengths in the simulated dataset were set to produce estimated gene trees with average BS bracketing that value - 500 bp alignments produced estimated gene trees with 63% average BS and 1000 bp alignments produced estimated gene trees with 79% average BS. The number of genes was varied from the set {50, 100, 200, 400, 800}. Each model condition was identified by 3 aspects - ILS level, the number of genes and the sequence length. In total, there were 11 model conditions. For each model condition, 20 replicate datasets were simulated. Notably, all the gene trees in all the model conditions were bifurcating.

7.3.2 Species Tree Estimation Tools

We have used ASTRAL version 5.6.1, downloaded from <https://github.com/smirarab/ASTRAL>. This version incorporates the original algorithm of ASTRAL as well as subsequent improvements, as published in [199, 200, 244, 292]. We run the heuristic version of ASTRAL in our comparisons. In this version, ASTRAL constrains the search space to reduce the running time. This heuristic version is statistically consistent under the multi-species coalescent model [199].

To create the MRP trees, the MRP matrices were generated using a custom Java program. The parsimony problem was solved heuristically using the default approach available in PAUP* (v.4.0b10) [259]. PAUP* generates an initial tree through random sequence addition and then performs Tree Bisection and Reconnection (TBR) moves until it reaches a local optimum. This process is repeated 1000 times, and at the end the

most parsimonious tree is returned. If there are multiple trees with the same maximum parsimony score, the “extended majority consensus” of those trees is returned.

The source code for QFM was obtained from Reaz et al. [236]. The code and packages for DCM boosting were obtained from Bayzid et al. [23].

7.4 Results

In this section, we present the results of our experiments. We compare the running time of DCM boosted QFM with that of native implementation of QFM. Our results indicate that the scalability of QFM is improved through boosting. We then focus on comparing the performance of our approach against ASTRAL in terms of missing branch rate (i.e. FN rate). As mentioned earlier, our approach requires a starting/guide species tree. We generated the guide tree using MRP which is very fast but less accurate. Since the MRP tree is available to us, we also measure its accuracy and compare it with trees generated by ASTRAL and our approach. We examine 2 versions of our approach. In one version, we perform 2 iterations of boosting (DCM2-QFM) and in another version, 5 iterations are applied (DCM5-QFM).

7.4.1 Running Time

As mentioned earlier, each model condition in the simulated dataset is identified by 3 aspects - ILS level, the number of genes and the sequence length. In total, there are 11 model conditions. In each of these model conditions, we have averaged the running time of QFM over 20 replicates. We then performed 5 iterations of DCM boosting (DCM5-QFM) and averaged the running time of boosted QFM over the same 20 replicates. The results are recorded in Table 7.1. Our current implementation of DCM5-QFM is sequential. However, after DACTAL decomposition, the steps needed to estimate species trees on each subset of taxa are embarrassingly parallel. Therefore, in a parallel implementation, the running times of these parallel steps will not add up. Instead only the maximum time

Table 7.1: Average running time (minutes) of boosted and native QFM for various model conditions. The average is taken over 20 replicates. The running time of the parallel version of DCM5-QFM is estimated.

Model condition	QFM	DCM5-QFM (Sequential)	DCM5-QFM (Parallel)
0.2X,200gt,500bp	15.00	5.14	0.72
0.5X,200gt,500bp	14.19	4.52	0.57
1X,200gt,500bp	14.80	4.53	0.58
2X,200gt,500bp	9.89	3.84	0.57
1X,50gt,500bp	8.44	2.08	0.32
1X,100gt,500bp	9.27	3.00	0.45
1X,400gt,500bp	11.22	5.00	0.65
1X,800gt,500bp	12.54	7.97	1.03
1X,200gt,250bp	12.59	4.00	0.59
1X,200gt,1000bp	10.80	3.82	0.57
1X,200gt,true gene tree	10.32	3.71	0.56

taken by any of these parallel steps will contribute to the total running time. Assuming negligible time to bootstrap the parallel platform, we can thus estimate the running time of the parallel implementation, though we have not actually implemented it.

As can be seen from Table 7.1, the improvement in running time in the sequential implementation of DCM5-QFM is 1.5 to 4 times over the native implementation of QFM among the different model conditions. When we take a look at the estimated running time of the parallel version of DCM5-QFM, the improvement ratio lies between 12 to 26.5 times. Therefore, boosting has clearly succeeded in increasing the scalability of QFM.

7.4.2 Topological Accuracy

While boosting improved the running time of QFM, it is important to ensure that this does not adversely impact the topological accuracy of the estimated species tree. Therefore, we have compared boosted QFM with ASTRAL in the same 11 model conditions. We

Table 7.2: Average FN rate of different species trees for various model conditions. The average is taken over 20 replicates.

Model condition	MRP	ASTRAL	DCM2-QFM	DCM5-QFM
0.2X,200gt,500bp	0.1471	0.0861	0.1029	0.0945
0.5X,200gt,500bp	0.0544	0.0485	0.0471	0.0441
1X,200gt,500bp	0.05	0.0485	0.0471	0.0426
2X,200gt,500bp	0.0294	0.025	0.0206	0.0206
1X,50gt,500bp	0.0941	0.075	0.0897	0.0838
1X,100gt,500bp	0.0706	0.0706	0.0647	0.0632
1X,400gt,500bp	0.0412	0.0147	0.0147	0.0147
1X,800gt,500bp	0.0176	0.0074	0.0029	0.0029
1X,200gt,250bp	0.0794	0.0632	0.0544	0.0485
1X,200gt,1000bp	0.0338	0.0221	0.0162	0.0162
1X,200gt,true gene tree	0.0294	0.0132	0.0088	0.0088

summarize the average FN rates of the different species tree estimation methods in these model conditions in Table 7.2. The best result in each model condition is highlighted using bold-faced font. As expected, MRP has inferior FN rate compared to the other methods in all the model conditions. In 8 of the 11 model conditions, boosted QFM outperforms ASTRAL. In another case, ASTRAL and boosted QFM had identical average FN rates. Of these 9 cases, 5 cases observed the benefit of boosting with only 2 iterations. In the remaining 4 cases, 5 iterations were needed to achieve better performance. In what follows, we have examined the comparative behavior more thoroughly as the length of gene sequences or the amount of ILS or the number of gene trees vary.

In Figure 7.2, we plot the FN rate of trees generated by MRP, ASTRAL, DCM2-QFM and DCM5-QFM against varying sequence length. The box plots are drawn based on 20 replicates. The model condition had 200 genes and moderate level of ILS. In one experiment, we have used the true gene trees that were generated from the model species tree. In other experiments we simulate the gene sequences of different lengths (250, 500 and 1000 base pairs) from the true gene trees and then use the gene trees estimated from

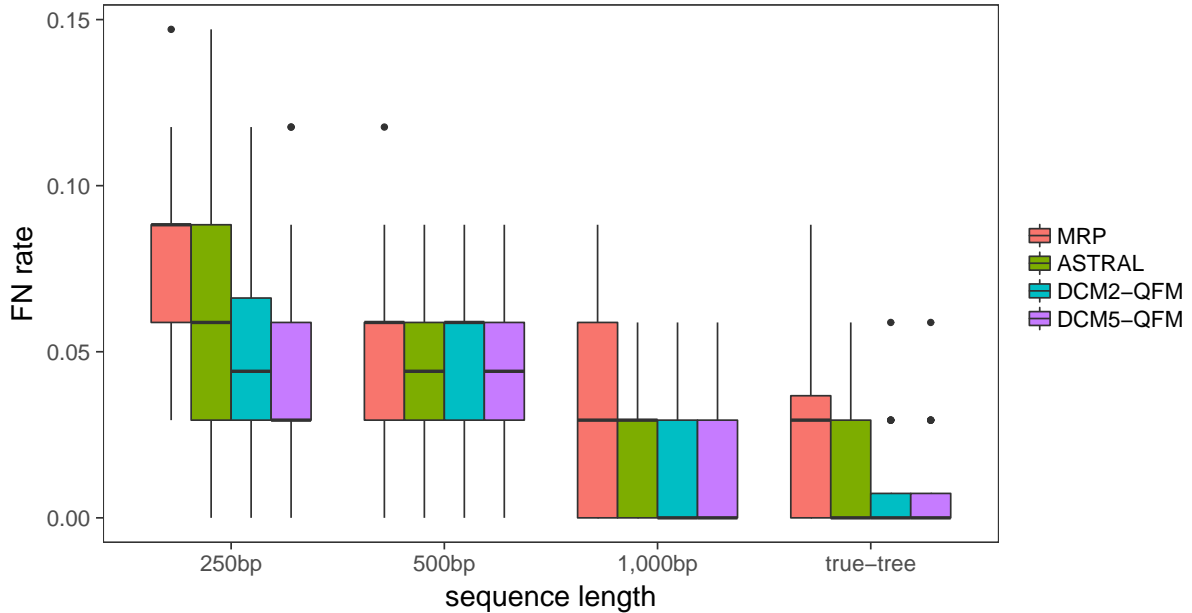


Figure 7.2: FN rates of MRP, ASTRAL and boosted versions of QFM on the simulated mammalian datasets with varying sequence length (200 genes, moderate amount of ILS). For the boosted versions of QFM, we show the FN rates of the best trees, with respect to the quartet support, after two and five iterations of DACTAL-based boosting.

these simulated sequences. These estimated gene trees are expected to have errors that decreases with the increased length of the simulated sequences.

As shown in Figure 7.2, in each case the MRP tree is inferior compared to the other trees, which is expected. For smaller sequences (250bp), 2 iterations of boosting sufficed for QFM to outperform ASTRAL. Nonetheless, 5 iterations improved the performance further. For the 500bp sequences, DCM2-QFM was worse than ASTRAL. Increasing the number of iterations (DCM5-QFM) made QFM perform at par with ASTRAL. For the 1000bp sequences, both versions of boosted QFM had similar performance, which was

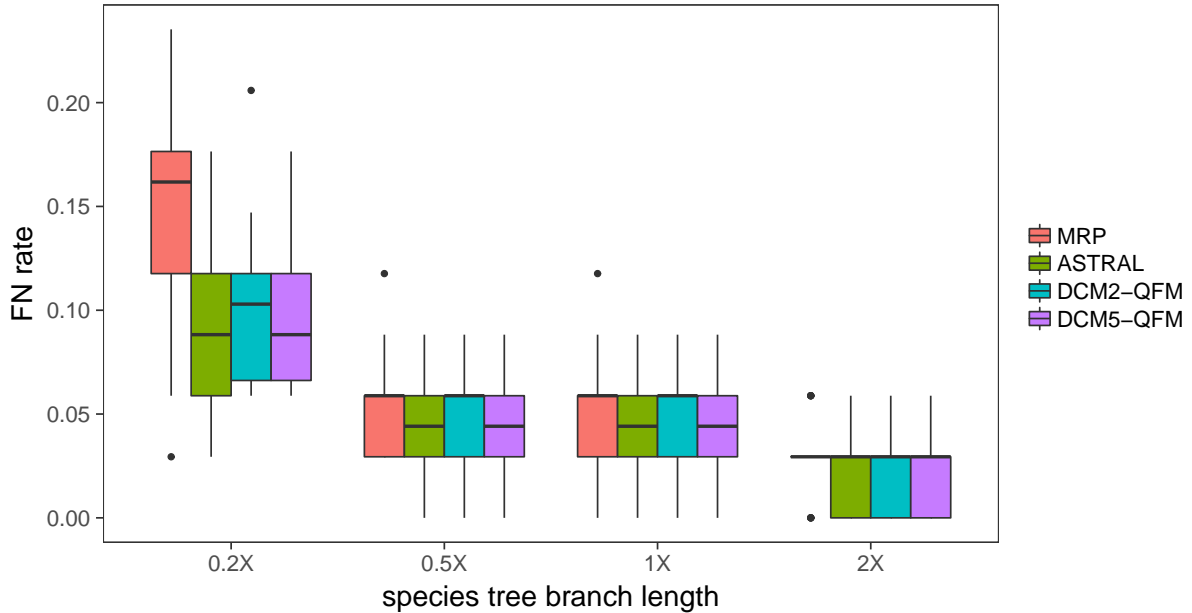


Figure 7.3: FN rates of MRP, ASTRAL and boosted versions of QFM on the simulated mammalian datasets with varying amounts of ILS. The number of genes and sequence length were fixed to 200 instances and 500 bp respectively. 2X model condition contains the lowest amount of ILS while 0.2X refers to the model conditions with the highest amount of ILS. For the boosted versions of QFM, we show the FN rates of the best trees, with respect to the quartet support, after two and five iterations of DACTAL-based boosting.

better than ASTRAL’s performance. In case of the true-tree, ASTRAL and boosted QFM had the same median FN rate. However, FN rate of QFM in the 20 replicates spanned a very small range, with a few outliers.

Figure 7.3 shows the FN rates of MRP, ASTRAL and boosted QFM in the face of varying amounts of ILS. 200 genes of 500 bp each were used to estimate the gene trees. In

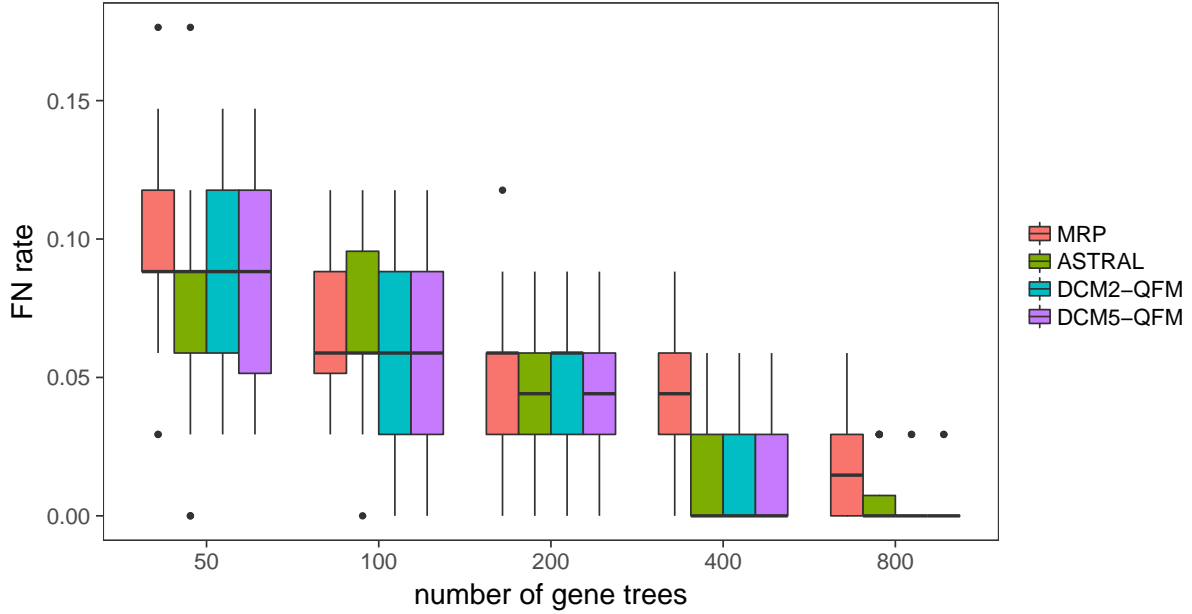


Figure 7.4: FN rates of MRP, ASTRAL and boosted versions of QFM on the simulated mammalian datasets for different number of gene trees. The number of genes was varied from 50 to 800. The amount of ILS was set to 1X level, while the sequence length was set to 500 bp. For the boosted versions of QFM, we show the FN rates of the best trees, with respect to the quartet support, after two and five iterations of DACTAL-based boosting.

the model conditions with increased ILS (0.5X and 0.2X), DCM5-QFM was comparable with ASTRAL in terms of FN rate; DCM2-QFM was slightly worse. The same observation holds in the moderate ILS condition. In low ILS level (2X), all 4 methods had the same median FN rate.

Figure 7.4 shows the FN rates of MRP, ASTRAL and boosted QFM as the number of input gene trees is varied from the set $\{50, 100, 200, 400, 800\}$. Moderate level of ILS (1X) and gene sequences of 500 bp were used. All the methods show performance improvement

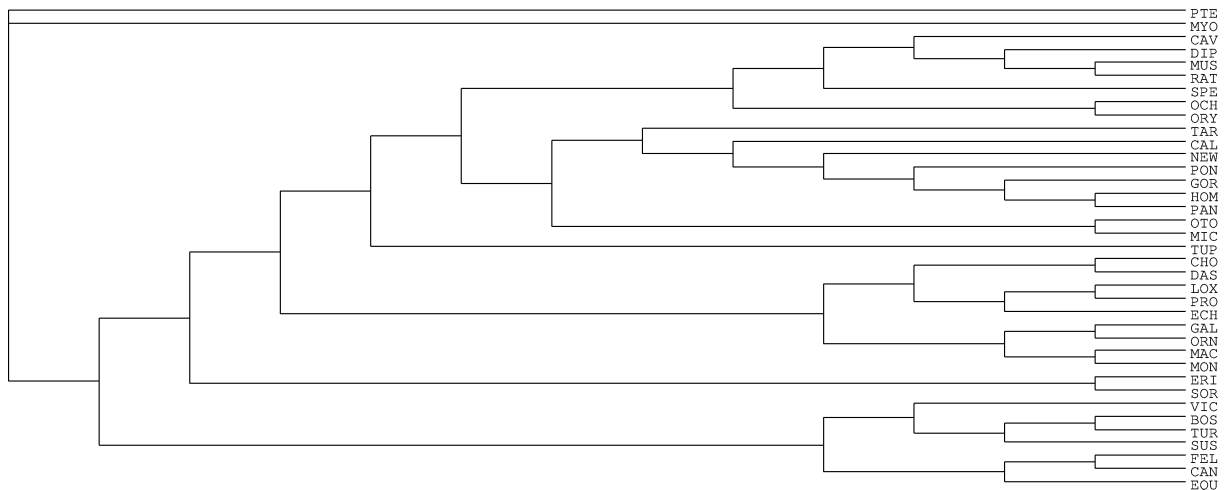


Figure 7.5: Species tree generated by DCM boosted QFM on the simulated dataset with the 37 taxa used in this study. The model condition used to generate this tree had 1X level of ILS, 200 true gene trees. The first replicate (out of the 20 replicates) was used. Boosting with 2 and 5 iterations produced the same tree.

with increased number of gene trees. Boosted QFM shows comparable performance with ASTRAL in all cases, marginally outperforming ASTRAL when 800 gene trees are used.

With the superiority of boosted QFM established in our experiments, we are now compelled to visually depict the species trees estimated by boosted QFM. We have drawn the species tree for each of the 11 model conditions in Appendix B. In Figure 7.5, we only show the species tree generated for the model condition with moderate ILS with 200 true gene trees.

7.5 Conclusion

QFM is a highly accurate quartet amalgamation method for species tree estimation in presence of ILS. However, QFM does not scale to large number of taxa. In this chapter, we have applied DCM boosting to scale QFM to datasets with large number of taxa. We then compare the boosted version of QFM with ASTRAL, a highly accurate, statistically consistent species tree estimation method which has become extremely popular. In our

experiments with different model conditions with varying amount of ILS, number of gene trees and gene lengths, we find DCM boosted QFM to estimate species trees that are better or as good as the species trees estimated by ASTRAL. We have thus offered the biologists an alternate tool for accurate estimation of species trees on large number of taxa.

In the next chapter, we conclude this thesis by summarizing the major contributions of this research, followed by directions for future research.

Chapter 8

Conclusion

In this final chapter, we draw the conclusion of our thesis by describing the major contributions of this research work along with some interesting discussion that may be seen as the main message of this thesis. Subsequently we follow that up with some directions for future research.

8.1 Protein Attribute Prediction

In Part I of this thesis, we focused on solving several protein attribute prediction problems. These included Golgi Apparatus (GA) resident protein type (or sub-Golgi protein type) prediction, DNA-binding protein (DNA-BP) prediction and protective antigen prediction. In solving each of these problems, we applied machine learning based approaches where the class-discriminating features were extracted solely from the primary sequence of the proteins.

Given a GA protein sequence as input, we built a classifier that can predict whether it is a *cis*-Golgi or *trans*-Golgi protein. In building this predictor, we have employed Random forests [34] algorithm for feature ranking and Support Vector Machine (SVM) [32] to train the classification model. We have applied Synthetic Minority Over-sampling Technique (SMOTE) [43] for data balancing. Our method, *identification of sub-Golgi Protein Types*

(*isGPT*), outperformed state-of-the-art predictors by achieving accuracy values of 95.4%, 95.9% and 95.3% for 10-fold cross-validation test, jackknife test and independent test respectively. We subsequently established a publicly accessible web interface of *isGPT* at <http://isgpt.research.buet.ac.bd/>. Notably, the relevant paper has been published in *Artificial Intelligence in Medicine*.

Next we worked to build a predictor for DNA-BPs. Like before, we employed Random forests algorithm to rank the features, after extracting them from the primary sequence. As the dataset was balanced in this problem, there was no need for a data balancing step. However, before applying the prediction algorithm, we applied SVM based Recursive Feature Elimination (SVM-RFE) [118] method to select an optimal set of features. We then trained the prediction model using SVM with linear kernel. Our proposed method was named *DNA-binding Protein Prediction model using Chou's general PseAAC*, or *DPP-PseAAC* in short. *DPP-PseAAC* outperformed all the state-of-the-art DNA-BP predictors in cross-validation testing, by achieving accuracy values of 93.21%, 95.91% for 10-fold cross-validation test and jackknife test respectively. In independent testing, it obtained an accuracy of 77.42% which was also better than all other methods except for Local-DPP [277]. Based on the commendable performance of *DPP-PseAAC*, we have established a publicly accessible web interface for its wide adoption at <http://dpp-pseaac.research.buet.ac.bd/>. Informatively, the relevant paper has been published in *Journal of Theoretical Biology*.

We also proposed a new protective antigen predictor. The steps that were applied in the process of building *DPP-PseAAC* were also followed in solving the problem of protective antigen prediction. However, since the training dataset had significant imbalance, it was balanced using random undersampling before the learning algorithm was applied. Named as *Antigenic*, our proposed model achieved accuracy, sensitivity and specificity values of 78.04%, 78.99% and 77.08% respectively in 10-fold cross-validation testing on the benchmark dataset. In jackknife cross-validation, the corresponding scores were 80.03%, 80.90% and 79.16% respectively. *Antigenic* has been made publicly available through the web interface at <http://antigenic.research.buet.ac.bd/>.

Through the design and development of these three predictors, we have generalized a framework for feature extraction and selection that can be applied to any protein attribute prediction problem, which can be seen as a key contribution of this thesis. The steps of this framework can be summarized as follows.

- Represent the proteins in terms of sequence based features such as n -grams, n -gapped-dipeptide and position specific n -grams features.
- Rank the features in each individual feature space separately by applying Random forests algorithm. The *mean decrease in accuracy*, as calculated by the Random forests algorithm, is used as the ranking score of each feature.
- In each individual feature space, remove all the features with ranking score less than or equal to 0.
- Combine the remaining features from all the feature spaces. Re-rank the combined set of features using another iteration of Random forests algorithm.
- Keep all the features with non-negative ranking scores. Apply SVM-RFE to re-rank these features as follows. SVM is first run on the entire feature set to rank the features. In the recursive step, 25 least ranked features are removed, SVM is then run again on this reduced feature space, and feature ranking is recomputed. The recursion is repeated until all the features are eliminated. Thus a new feature ranking is obtained. We call it the SVM-RFE (coarse) ranking.
- Using this new ranking, construct different prediction models (using a suitable prediction algorithm) by varying the number of features. A large feature space is first explored, albeit with coarse granularity. That means, the number of features that are added or removed between model constructions is large. After examining various performance curves, zoom into the interesting terrain of the feature space and reduce the jumps in feature count in generating new models. As an example, in case of DPP-PseAAC, we started with a model with 500 top-ranked features, using

the SVM-RFE (coarse) ranking. We subsequently added 1000 next ranked features in each iteration, until the feature count became 6500. Based on the performance curves, the feature space range [100, 1500] seemed promising. Therefore, more models were generated in this space, however the change of features in each step became finer: 100 features.

- When the feature space under investigation is significantly narrowed down, apply a second round of SVM-RFE in this feature space, but this time with steps of 1 feature elimination (instead of 25 features). This gives a more reliable ranking, which we have called SVM-RFE (fine) ranking. Using this final ranking, explore prediction models with very granular feature count jumps to close in on the best model. For example, in case of DPP-PseAAC, the SVM-RFE (fine) ranking was generated for the top 600 features. Based on this new ranking, we first generated a prediction model using the top 10 features. We generated and compared more models, adding 10 next ranked features in each iteration, until the feature count became 600.

Figures 4.2 (in Chapter 4) and 5.1 (in Chapter 5) depicted the steps of this general framework of feature extraction and selection. A distinct and note-worthy property of this framework is essentially the focus on only the primary sequence. This is in sharp contrast to the modern trend of focusing on features from structural and functional domains. There has also been an increased interest recently in features extracted from the proteins' evolutionary information. However, this evolutionary information, in the form of a Position Specific Scoring Matrix (PSSM), takes time to generate, requiring at least three iterations of PSI-BLAST [15] against the non-redundant protein database. Besides, lack of enough homologous sequences in the searched database may render a PSSM that cannot describe the target protein adequately, thus resulting in wrong attribute predictions [164]. Our proposed framework therefore offers a relief, empirically promising the potential of only focusing on the primary sequence, which is light-weight and less time consuming. While we dare suggest that focusing on the primary sequence should be enough to capture the underlying structural and functional information encoded therein,

we still lack sufficient theory behind it. So, our promising quantitative results should be followed through by qualitative investigation to elucidate the biological insight behind our suggestion, which by the way is out of the scope of this thesis that basically focuses on computational biology rather than biology itself.

8.2 Phylogeny Reconstruction

In Part II of this thesis, we focused on phylogeny reconstruction. We first explored the idea of using minimal and relative absent words to compute the distance between two biological sequences (proteins, genes, RNAs etc.). We proposed several distance measures for a pair of biological sequences and recommended the best distance measure based on our analysis. We have also implemented a related web-based tool with limited capacity here: <http://77.68.43.135/AWordS>. For a collection of gene sequences, we demonstrated how the pairwise distance matrix thus produced can be used to reconstruct the gene phylogeny in an alignment-free manner. The relevant paper has been published in *BMC Research Notes*.

We subsequently focused on species tree estimation, particularly when the gene tree discordance is modeled by incomplete lineage sorting (ILS) or deep coalescence. We applied disk covering methods (DCMs) [132,133,205,239] to boost the scalability and performance of Quartet FM (QFM) [236], a coalescent-based summary method for species tree estimation. Experiments with a simulated dataset of 37 taxa demonstrated superiority of DCM boosted QFM over ASTRAL [199,200,292], a highly accurate and popular species tree estimation method that is statistically consistent under the multi-species coalescent model.

8.3 Future Research Directions

The work in this thesis has introduced further research opportunities. These new research directions can be summarized as follows.

- In any protein attribute prediction problem, the first step is to prepare or obtain a stringent benchmark dataset [61]. To avoid homology bias, the datasets should contain proteins with pairwise sequence similarity no more than a certain cutoff or threshold (e.g. 25%) [61]. In the sub-Golgi protein type prediction problem, we used a benchmark dataset from [288] where CD-HIT [131] software was used to restrict pairwise sequence similarity to less than 40%. In the DNA-BP and protective antigen prediction problems, homology bias was reduced in the benchmark datasets using PISCES [274] and BLASTCLUST [15] software packages respectively. These latter 2 tools are old, specially BLASTCLUST is deprecated [3]. On the contrary, CD-HIT is a relatively new tool and have been gaining popularity of late. Therefore, new benchmark datasets could be prepared for the DNA-BP and protective antigen prediction problems, using CD-HIT to reduce the homology bias. The state-of-the-art predictors along with our proposed predictors should be re-trained and evaluated using this new dataset. This can pave the way for future researchers to innovate new and improved prediction techniques, using a refreshed benchmark dataset.
- In all the prediction problems we have worked on, the position specific n -grams (PSN) features were extracted only from the N-terminus of the protein sequences. PSNs from the C-terminus could also be extracted and combined with the other features in pursuit of further improvement in performance of our predictors. Other sequence based features such as amino acid physico-chemical properties could also be experimented with.
- We proposed several pairwise distance measures for biological sequences and demonstrated their use in phylogeny reconstruction. These distance measures could potentially be used in the pipeline of protein attribute prediction problems as well. Firstly, they can be applied during dataset preparation to avoid homology bias.

As mentioned above, CD-HIT [131] is currently the most popular software in this regard. Experiment with different datasets from different protein attribute prediction problems should be designed to examine the efficacy of our proposed distance measures in reducing homology bias in any benchmark dataset. Secondly, for a collection of protein sequences, the pairwise distance matrix, based on our proposed distance measures, can be computed and used as a custom kernel with SVM to produce prediction models for different protein attribute prediction problems. For the distance matrix to qualify as a valid kernel, Mercer's condition must be satisfied [67]. We welcome theoretical research to validate whether the condition is satisfied for any or all of our proposed distance measures. Even if our custom kernels do not satisfy Mercer's condition, a given training dataset can possibly result in a positive semidefinite Hessian, in which case the training will converge perfectly well, even though the theoretical basis for the maximum margin classifier may be lacking [36]. Therefore, another research avenue that we propose is to experimentally evaluate the performance of our proposed custom kernels in several representative protein attribute prediction problems.

- The gene tree estimation methods using minimal and relative absent words were evaluated on a dataset of 11 gene sequences. An obvious future direction in this work is to test our methods with a larger dataset (with more and longer sequences).
- The DCM boosted QFM technique for species tree estimation was evaluated using a simulated dataset from [199] that was also used in [23]. This dataset does not have any polytomy in the input gene trees. Since QFM utilizes quartets induced from the gene trees, any polytomy in the input must be resolved before QFM can proceed. How best to resolve the input polytomy? Would arbitrary resolution suffice? One option could be to enumerate all possible resolutions, estimate the species tree in each case and then choose from these the best species tree based on the criteria of quartet support. However, will this approach be computationally feasible? Can an alternate approach be devised which enumerates and utilizes only a few of the

possible resolutions based on some optimality criteria? These questions need to be investigated both through theoretical analysis and experimental studies.

- In the final step of DCM boosted QFM, we combine the subset species trees using a *supertree* method called SuperFine+MRL [209]. While this method attempts to produce a fully resolved tree, it may not always be successful. In fact, we observed non-binary species trees generated in many runs of our method in the simulated dataset. Therefore, further research is required to resolve the polytomy in the output species trees as well.
- We have used the native implementation of QFM algorithm from [236] and then applied the techniques of boosting. Reaz et al. [236] commented that their native implementation of QFM was not very efficient. While debugging the source code, we too observed several possible improvement opportunities. As an example, work could be done in applying suitable data structures to make the partitioning algorithm of QFM run faster.
- The DCM boosted QFM should be tested with even larger datasets, as used in [200]. An exciting future work thus remains that is to examine its performance in studies with taxa set as large as 1000 and optimize different steps in its pipeline to scale the method even better. Work remains as well to build a simple to use package for the biologists. This latter work is very crucial for wide adoption of any new species tree estimation method.

Overall, our research in this thesis empirically asserts the natural belief that a protein's functional and structural information are intrinsically encoded within its primary sequence. This assertion culminates in generalizing a framework for sequence based feature extraction and selection that can be applied to any protein attribute prediction problem. Our efforts in phylogeny reconstruction makes good strides in making different parts of the phylogenomics pipeline scale to larger datasets. And finally, complementing these significant contributions, our research opens up several directions for important future research in the fields of protein attribute prediction and phylogeny reconstruction.

Appendices

Appendix A

Supplementary Materials for Gene Tree Estimation Using Absent Words

These supplementary materials provide all distance matrices and estimated gene trees resulting out of our experiments described in Chapter 6. Notably, we have considered the four distance measures described in Section 6.2.2. These are Length-weighted Index (LWI), Jaccard Distance (JD), Total Variation Distance (TVD) and GC Content. The TVD and JD measures are directly applied on a pair of MAW sets. LWI and GC content measures have been applied on the symmetric difference as well as the intersection of the MAW sets. We have also applied these 2 measures on the RAW sets. With noRC and RC settings, this gives us a total of 16 distance matrices. From each of these distance matrices gene trees were estimated using two well-known methods, namely, Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [258] and Neighbor Joining (NJ) [243].

A.1 Distance Matrices

Table A.1: The distance matrix based on the Length Weighted Index on Symmetric Difference of MAW sets (on RC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		12.79	16.83	17.17	15.46	12.95	11.65	17.10	1.95	13.08	4.35
goat			16.06	13.49	14.21	13.62	13.48	17.19	12.45	4.67	13.71
opossum				17.22	17.95	20.30	16.93	20.20	16.66	17.27	17.48
gallus					18.01	17.04	15.25	20.94	17.19	16.27	17.96
lemur						17.35	16.96	18.58	15.04	13.25	16.25
mouse							15.54	16.96	11.61	13.11	13.72
rabbit								17.58	12.40	13.68	13.41
rat									17.28	17.81	18.55
gorilla										12.57	2.56
bovine											13.84
chimp											

Table A.2: The distance matrix based on the Length Weighted Index on Intersection of MAW sets (on RC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		-7.15	-4.63	-5.38	-6.91	-7.67	-7.99	-6.19	-13.21	-7.11	-12.55
goat			-4.29	-6.49	-6.81	-6.61	-6.34	-5.42	-7.24	-10.59	-7.14
opossum				-4.13	-4.44	-2.77	-4.12	-3.41	-4.64	-3.79	-4.76
gallus					-5.34	-5.33	-5.89	-3.96	-5.29	-5.21	-5.44
lemur						-5.84	-5.70	-5.82	-7.04	-7.40	-6.97
mouse							-5.92	-6.14	-8.27	-6.97	-7.74
rabbit								-5.49	-7.54	-6.35	-7.56
rat									-6.02	-5.21	-5.92
gorilla										-7.28	-13.36
bovine											-7.18
chimp											

Table A.3: The distance matrix based on the Length Weighted Index on RAW sets (on RC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		23.39	26.94	28.34	27.82	23.49	19.31	27.88	4.77	21.60	7.26
goat			28.71	24.16	25.89	25.52	24.33	27.43	21.77	8.73	24.26
opossum				29.55	31.23	29.21	26.69	30.52	26.90	28.16	28.44
gallus					28.66	30.22	26.27	30.89	28.25	26.21	30.51
lemur						30.21	27.63	30.96	27.77	25.91	30.27
mouse							24.09	26.43	20.98	23.17	23.29
rabbit								29.19	19.02	22.28	21.50
rat									28.37	27.95	30.21
gorilla										19.48	9.62
bovine											21.97
chimp											

Table A.4: The distance matrix based on the GCC Index on Symmetric Difference of MAW sets (on RC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		0.53	0.54	0.56	0.54	0.56	0.53	0.54	0.40	0.52	0.40
goat			0.55	0.57	0.56	0.57	0.55	0.54	0.55	0.62	0.53
opossum				0.57	0.54	0.54	0.53	0.53	0.55	0.54	0.54
gallus					0.55	0.56	0.58	0.54	0.57	0.58	0.55
lemur						0.54	0.54	0.52	0.56	0.55	0.54
mouse							0.55	0.54	0.58	0.56	0.56
rabbit								0.52	0.53	0.55	0.52
rat									0.54	0.53	0.53
gorilla										0.53	0.41
bovine											0.52
chimp											

Table A.5: The distance matrix based on the GCC Index on Intersection of MAW sets (on RC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		0.41	0.48	0.41	0.47	0.46	0.43	0.48	0.44	0.41	0.43
goat			0.46	0.40	0.47	0.45	0.41	0.45	0.42	0.45	0.42
opossum				0.46	0.52	0.44	0.43	0.51	0.48	0.43	0.48
gallus					0.41	0.39	0.42	0.39	0.40	0.40	0.41
lemur						0.46	0.44	0.46	0.48	0.45	0.48
mouse							0.41	0.46	0.46	0.44	0.46
rabbit								0.41	0.42	0.40	0.42
rat									0.47	0.43	0.47
gorilla										0.41	0.44
bovine											0.41
chimp											

Table A.6: The distance matrix based on the GCC Index on RAW sets (on RC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		0.58	0.54	0.60	0.55	0.55	0.55	0.55	0.50	0.56	0.50
goat			0.56	0.61	0.57	0.58	0.58	0.56	0.59	0.62	0.59
opossum				0.57	0.53	0.54	0.54	0.53	0.55	0.55	0.55
gallus					0.58	0.59	0.60	0.58	0.60	0.60	0.60
lemur						0.55	0.55	0.54	0.56	0.56	0.56
mouse							0.56	0.54	0.56	0.56	0.56
rabbit								0.55	0.56	0.57	0.56
rat									0.55	0.55	0.55
gorilla										0.58	0.59
bovine											0.58
chimp											

Table A.7: The distance matrix based on the Jaccard Distance of MAW sets (on RC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		0.70	0.82	0.80	0.76	0.70	0.61	0.80	0.15	0.69	0.26
goat			0.84	0.74	0.74	0.77	0.77	0.79	0.69	0.36	0.71
opossum				0.85	0.87	0.91	0.84	0.90	0.82	0.85	0.82
gallus					0.81	0.82	0.79	0.85	0.80	0.81	0.80
lemur						0.83	0.81	0.81	0.76	0.72	0.77
mouse							0.78	0.78	0.64	0.74	0.68
rabbit								0.81	0.63	0.75	0.65
rat									0.80	0.82	0.82
gorilla										0.67	0.15
bovine											0.69
chimp											

Table A.8: The distance matrix based on the Total Variation Distance of MAW sets (on RC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		0.09	0.11	0.17	0.09	0.13	0.08	0.12	0.04	0.06	0.03
goat			0.15	0.15	0.08	0.10	0.09	0.07	0.11	0.07	0.09
opossum				0.25	0.18	0.06	0.12	0.14	0.10	0.09	0.12
gallus					0.13	0.19	0.16	0.11	0.18	0.18	0.18
lemur						0.14	0.10	0.10	0.11	0.10	0.13
mouse							0.10	0.08	0.11	0.08	0.13
rabbit								0.10	0.07	0.09	0.10
rat									0.13	0.08	0.13
gorilla										0.08	0.06
bovine											0.07
chimp											

Table A.9: The distance matrix based on the Length Weighted Index on Symmetric Difference of MAW sets (on NoRC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		8.34	10.55	11.18	10.61	8.21	8.44	10.54	1.09	8.22	2.65
goat			10.37	9.23	9.00	8.50	8.63	10.21	8.06	2.78	8.83
opossum				11.01	12.34	12.60	10.24	12.08	10.45	10.87	11.00
gallus					11.08	10.93	9.17	12.55	11.46	10.22	11.76
lemur						10.69	10.93	11.29	10.24	8.76	11.11
mouse							10.11	10.32	7.43	8.11	8.50
rabbit								10.53	8.58	9.06	9.00
rat									10.60	10.41	11.71
gorilla										7.93	1.64
bovine											8.71
chimp											

Table A.10: The distance matrix based on the Length Weighted Index on intersection of MAW sets (on NoRC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		-3.73	-2.35	-2.40	-3.16	-4.01	-3.77	-2.92	-7.85	-3.73	-7.35
goat			-1.96	-2.90	-3.48	-3.38	-3.19	-2.61	-3.89	-5.97	-3.78
opossum				-1.73	-1.54	-1.06	-2.11	-1.39	-2.41	-1.65	-2.41
gallus					-2.53	-2.26	-3.01	-1.52	-2.28	-2.33	-2.40
lemur						-2.85	-2.60	-2.62	-3.36	-3.54	-3.20
mouse							-2.67	-2.76	-4.42	-3.52	-4.16
rabbit								-2.53	-3.71	-2.91	-3.78
rat									-2.91	-2.44	-2.62
gorilla										-3.89	-7.87
bovine											-3.77
chimp											

Table A.11: The distance matrix based on the Length Weighted Index on RAW sets (on NoRC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		12.94	14.16	15.55	15.08	12.50	10.06	14.79	2.06	11.85	3.35
goat			16.08	12.84	13.68	13.08	13.26	15.34	11.69	4.32	13.18
opossum				16.61	17.03	15.56	13.86	16.33	14.18	15.57	14.90
gallus					15.48	15.95	13.93	17.10	15.21	14.03	16.75
lemur						15.53	14.73	16.56	14.85	13.43	16.19
mouse							12.41	14.34	11.27	11.82	12.54
rabbit								15.96	9.60	11.81	11.09
rat									14.86	15.46	15.86
gorilla										10.33	8.89
bovine											11.82
chimp											

Table A.12: The distance matrix based on the GCC Index on Symmetric Index of MAW sets (on NoRC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		0.55	0.55	0.59	0.54	0.57	0.54	0.55	0.39	0.54	0.40
goat			0.56	0.58	0.56	0.59	0.56	0.56	0.57	0.63	0.55
opossum				0.58	0.53	0.54	0.54	0.53	0.55	0.54	0.54
gallus					0.57	0.58	0.60	0.57	0.59	0.58	0.58
lemur						0.54	0.54	0.52	0.55	0.55	0.54
mouse							0.57	0.56	0.60	0.57	0.58
rabbit								0.54	0.55	0.56	0.53
rat									0.55	0.54	0.54
gorilla										0.55	0.41
bovine											0.54
chimp											

Table A.13: The distance matrix based on the GCC Index on intersection of MAW sets (on NoRC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		0.40	0.47	0.40	0.44	0.45	0.41	0.44	0.42	0.40	0.41
goat			0.49	0.34	0.47	0.47	0.39	0.44	0.42	0.45	0.41
opossum				0.46	0.51	0.44	0.45	0.49	0.46	0.43	0.45
gallus					0.41	0.37	0.43	0.37	0.40	0.36	0.41
lemur						0.45	0.42	0.42	0.46	0.46	0.45
mouse							0.44	0.46	0.47	0.45	0.47
rabbit								0.41	0.41	0.41	0.41
rat									0.44	0.42	0.43
gorilla										0.41	0.43
bovine											0.41
chimp											

Table A.14: The distance matrix based on the GCC Index on RAW sets (on NoRC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		0.58	0.54	0.59	0.55	0.55	0.55	0.55	0.50	0.56	0.52
goat			0.55	0.60	0.56	0.57	0.58	0.56	0.59	0.61	0.59
opossum				0.57	0.52	0.53	0.54	0.52	0.55	0.54	0.55
gallus					0.57	0.59	0.59	0.57	0.60	0.60	0.60
lemur						0.54	0.55	0.53	0.56	0.55	0.56
mouse							0.56	0.54	0.56	0.56	0.56
rabbit								0.55	0.56	0.57	0.57
rat									0.55	0.55	0.56
gorilla										0.57	0.62
bovine											0.58
chimp											

Table A.15: The distance matrix based on the Jaccard Distance of MAW sets (on NoRC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		0.73	0.85	0.84	0.81	0.72	0.68	0.82	0.14	0.71	0.26
goat			0.88	0.81	0.77	0.80	0.80	0.82	0.72	0.38	0.74
opossum				0.89	0.92	0.94	0.86	0.93	0.84	0.89	0.84
gallus					0.85	0.88	0.81	0.90	0.85	0.86	0.85
lemur						0.85	0.84	0.85	0.80	0.76	0.82
mouse							0.83	0.82	0.66	0.78	0.70
rabbit								0.85	0.69	0.80	0.70
rat									0.83	0.84	0.85
gorilla										0.69	0.16
bovine											0.72
chimp											

Table A.16: The distance matrix based on the Total Variation Distance of MAW sets (on NoRC setting).

Species	human	goat	opossum	gallus	lemur	mouse	rabbit	rat	gorilla	bovine	chimp
human		0.09	0.12	0.18	0.09	0.14	0.08	0.14	0.03	0.06	0.03
goat			0.16	0.16	0.07	0.12	0.09	0.09	0.11	0.06	0.08
opossum				0.26	0.19	0.07	0.12	0.15	0.11	0.10	0.14
gallus					0.13	0.19	0.16	0.11	0.18	0.17	0.19
lemur						0.14	0.10	0.11	0.11	0.10	0.11
mouse							0.09	0.08	0.13	0.09	0.14
rabbit								0.10	0.07	0.07	0.09
rat									0.14	0.09	0.14
gorilla										0.08	0.06
bovine											0.07
chimp											

A.2 Estimated β -globin Gene Trees

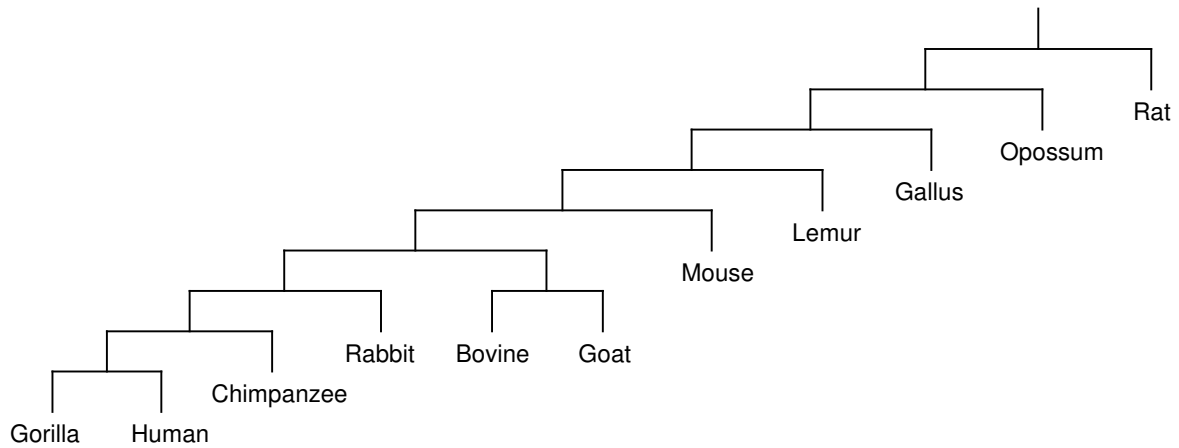


Figure A.1: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Length Weighted Index on the Symmetric Difference of the MAW sets (on RC setting).

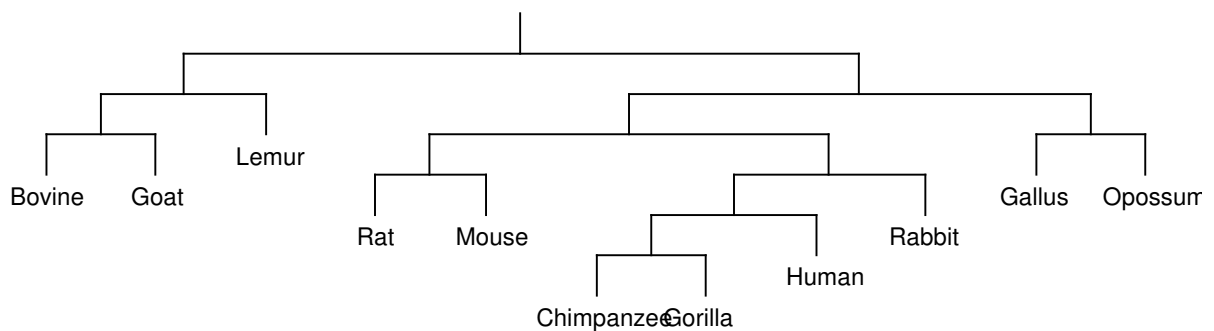


Figure A.2: The β -globin gene tree for the 11 species computed using Neighboring Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the Symmetric Difference of the MAW sets (on RC setting).

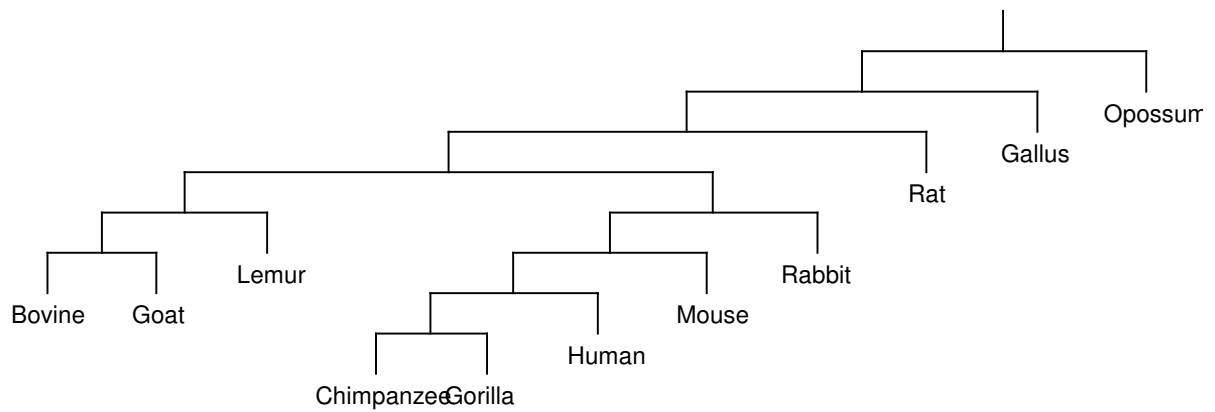


Figure A.3: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Length Weighted Index on the Intersections of the MAW sets (on RC setting).

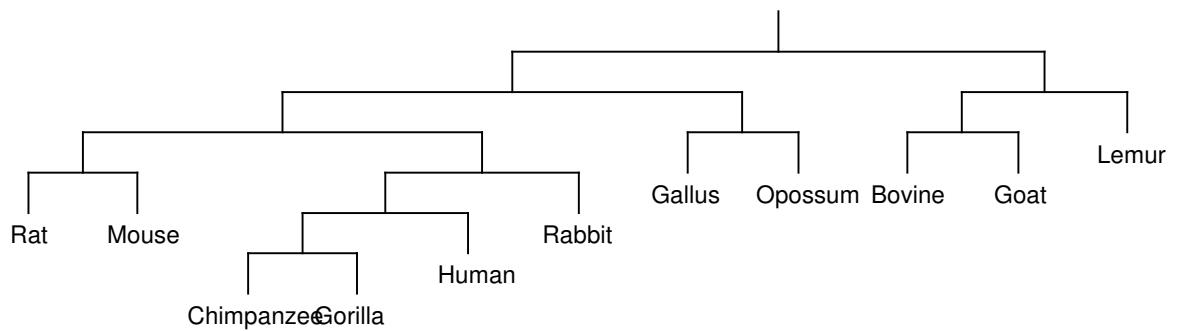


Figure A.4: The β -globin gene tree for the 11 species computed using Neighboring Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the Intersection of the MAW sets (on RC setting).

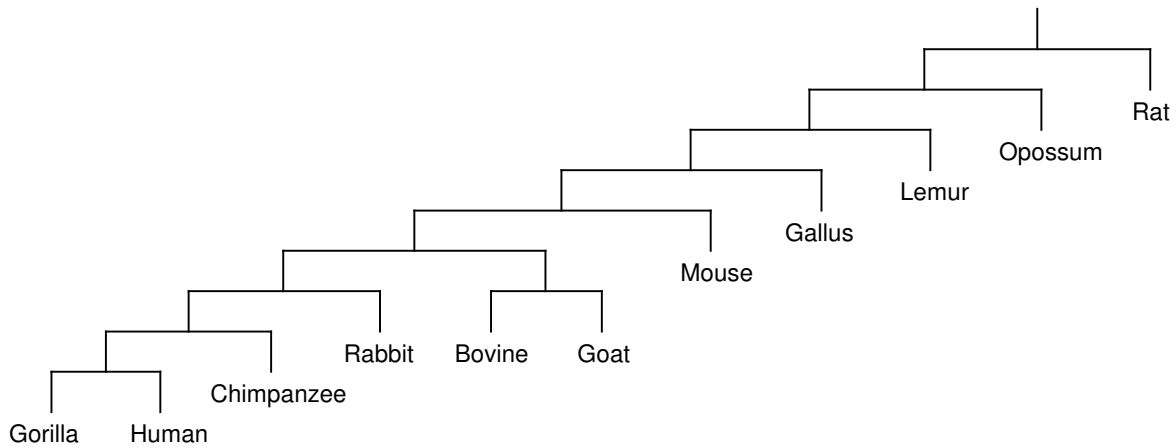


Figure A.5: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Length Weighted Index on the RAW sets (on RC setting).

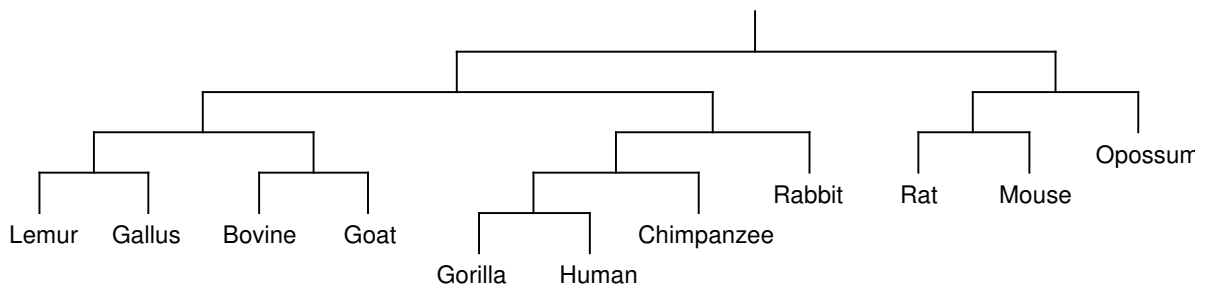


Figure A.6: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the RAW sets (on RC setting).

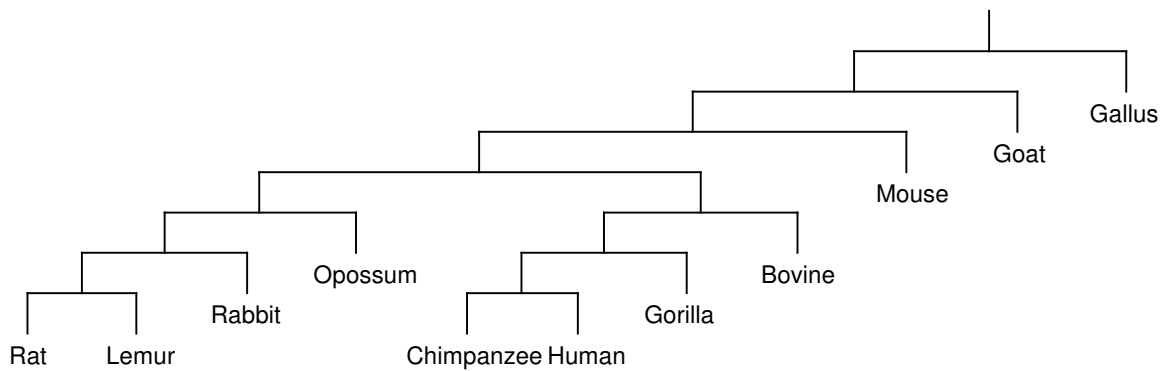


Figure A.7: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the GC content on the Symmetric Difference of the MAW sets (on RC setting).

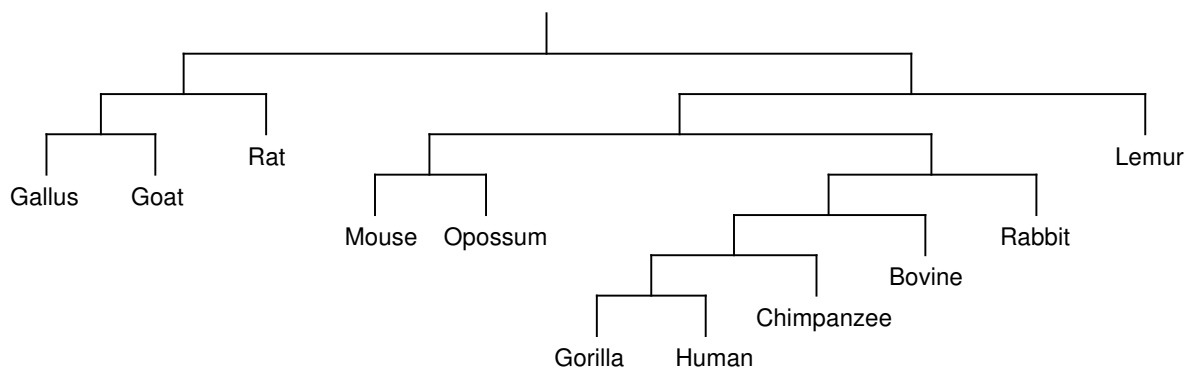


Figure A.8: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the GC content on the Symmetric Difference of the MAW sets (on RC setting).

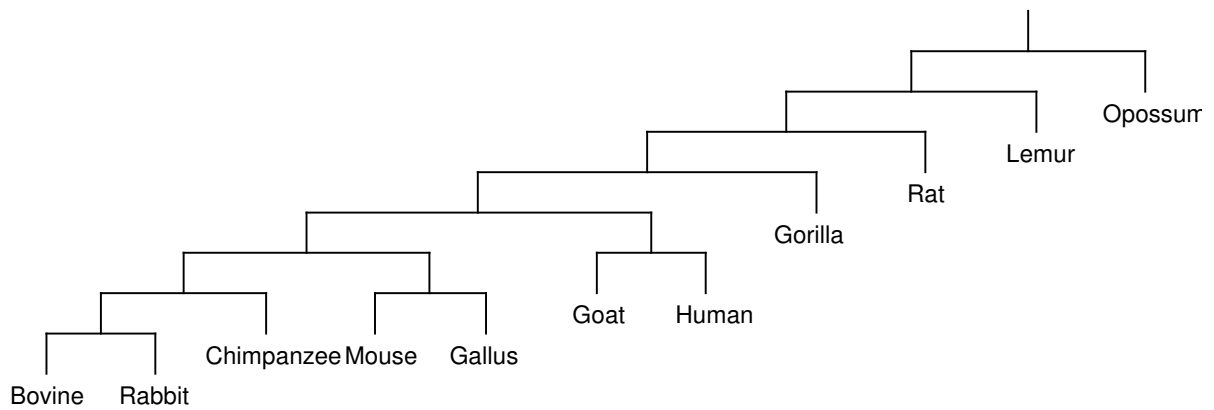


Figure A.9: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the GC content on the Intersection of the MAW sets (on RC setting).

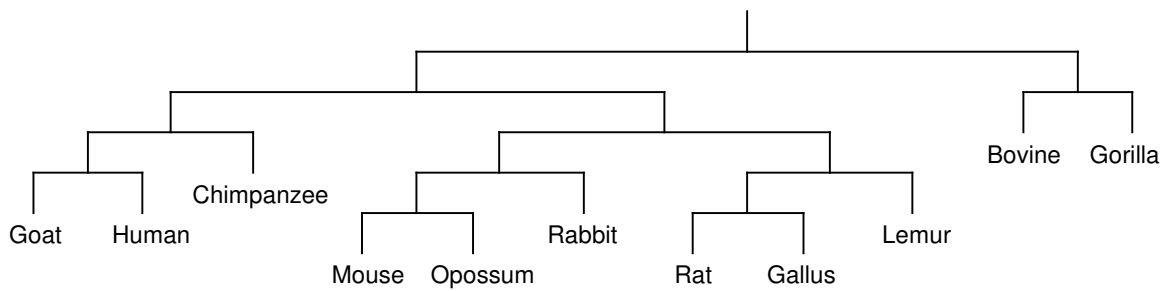


Figure A.10: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the GC content on the Intersection of the MAW sets (on RC setting).

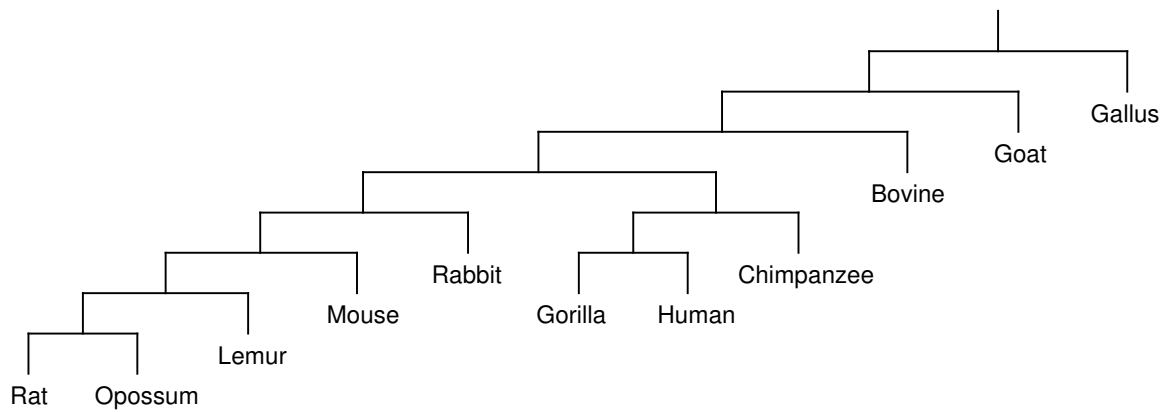


Figure A.11: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the GC content on the RAW sets (on RC setting).

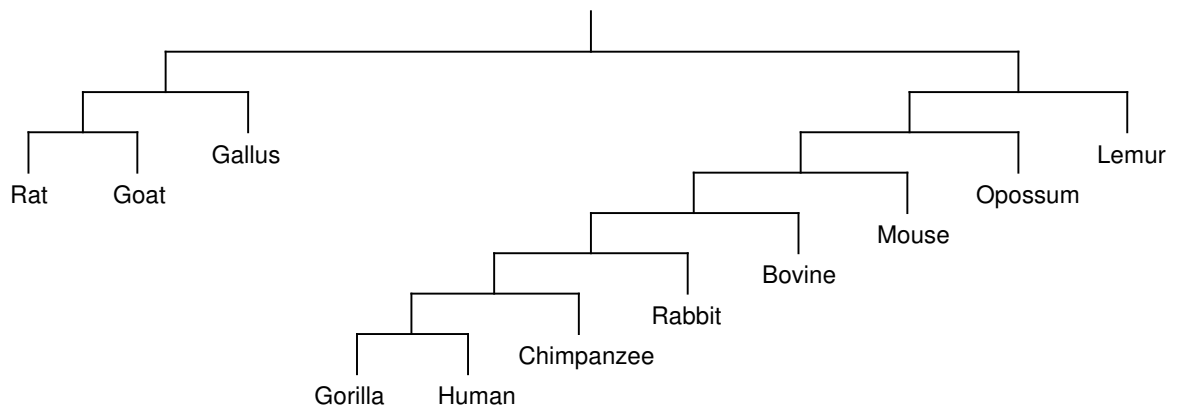


Figure A.12: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the GC content on the RAW sets (on RC setting).

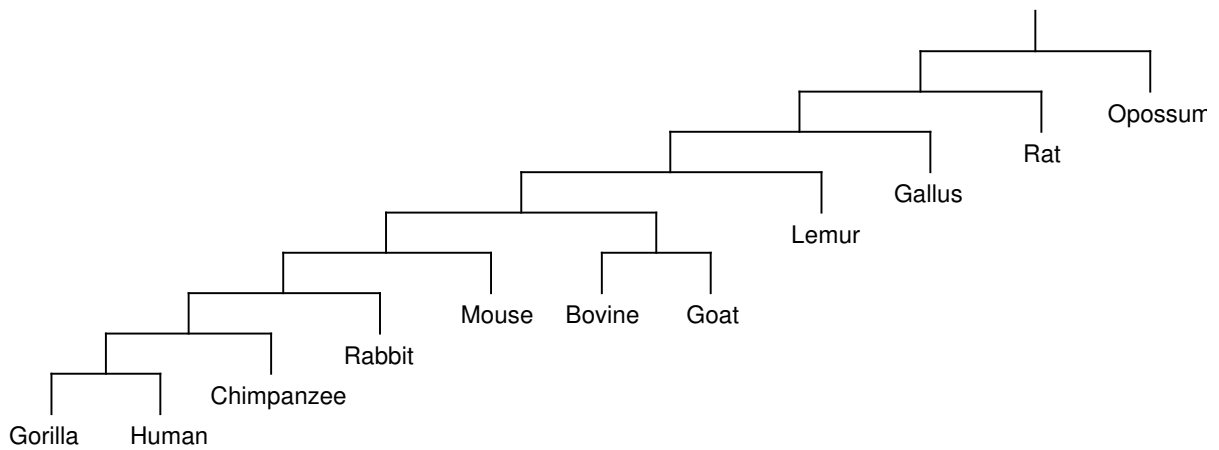


Figure A.13: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Jaccard Distance of the MAW sets (on RC setting).

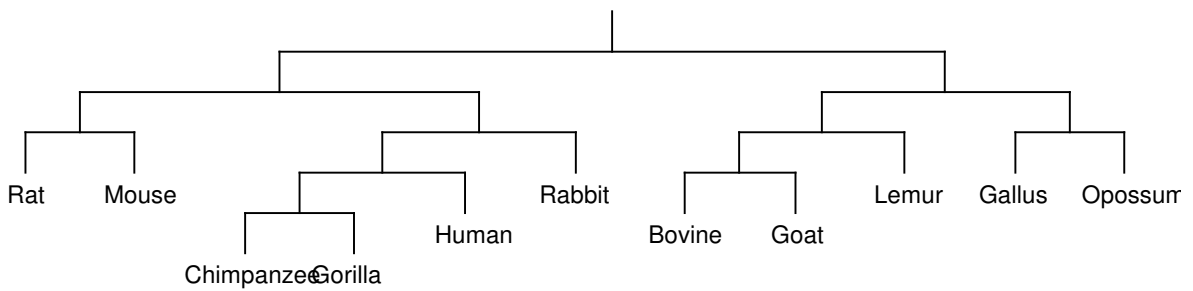


Figure A.14: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Jaccard Distance of the MAW sets (on RC setting).

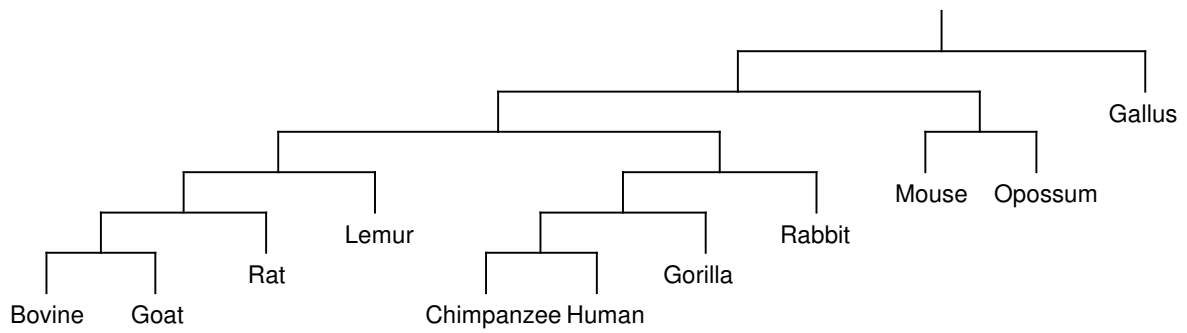


Figure A.15: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Total Variation Distance of the MAW sets (on RC setting).

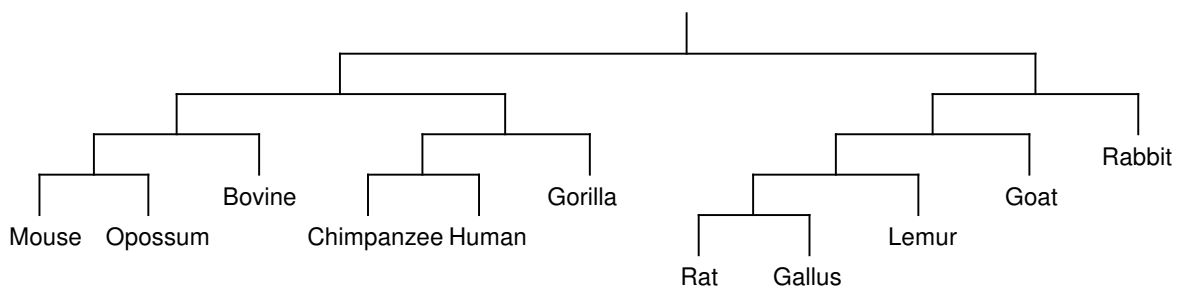


Figure A.16: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Total Variation Distance of the MAW sets (on RC setting).

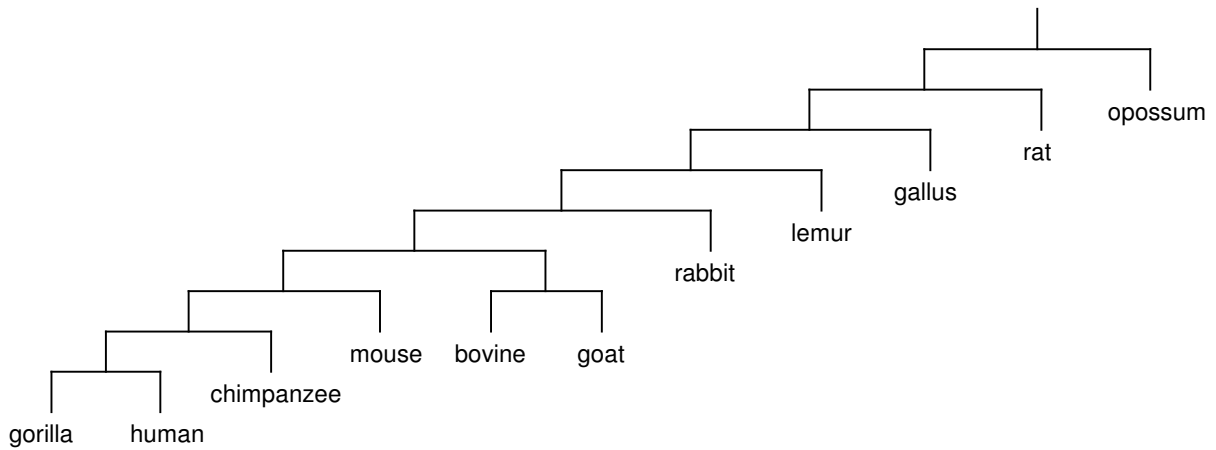


Figure A.17: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Length Weighted Index on the Symmetric Difference of the MAW sets (on NoRC setting).

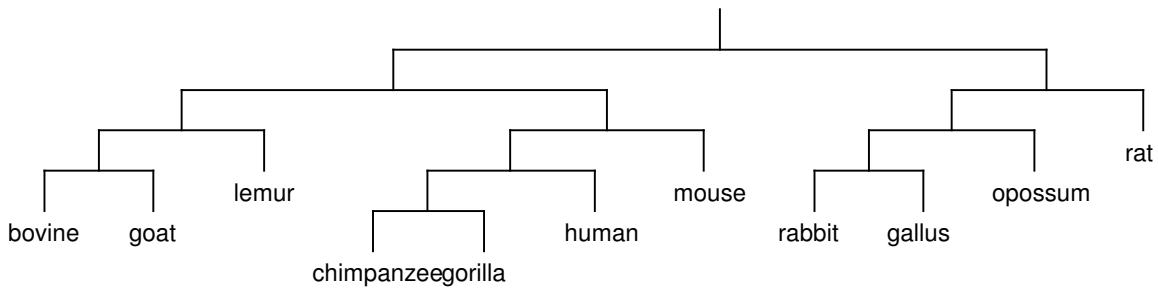


Figure A.18: The β -globin gene tree for the 11 species computed using Neighboring Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the Symmetric Difference of the MAW sets (on NoRC setting).

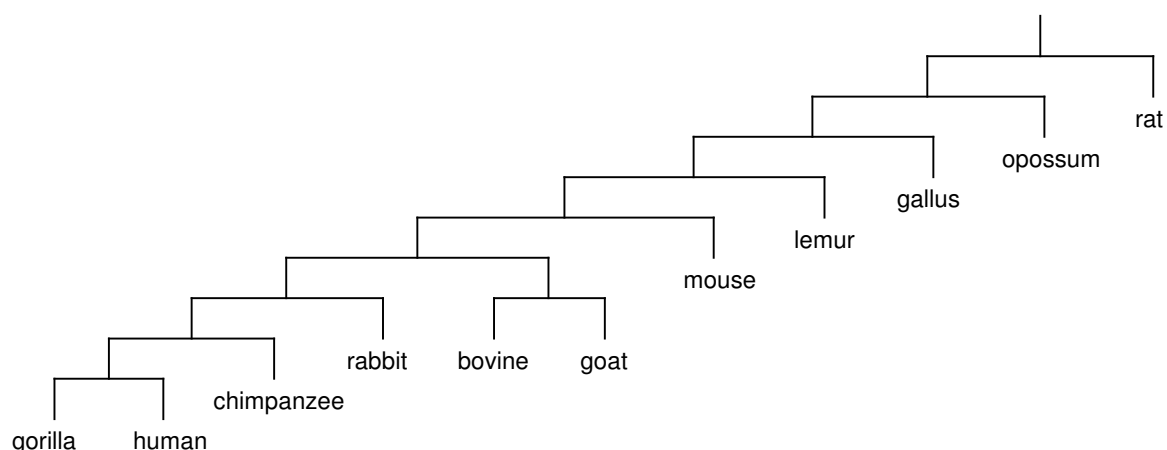


Figure A.21: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Length Weighted Index on the RAW sets (on NoRC setting).

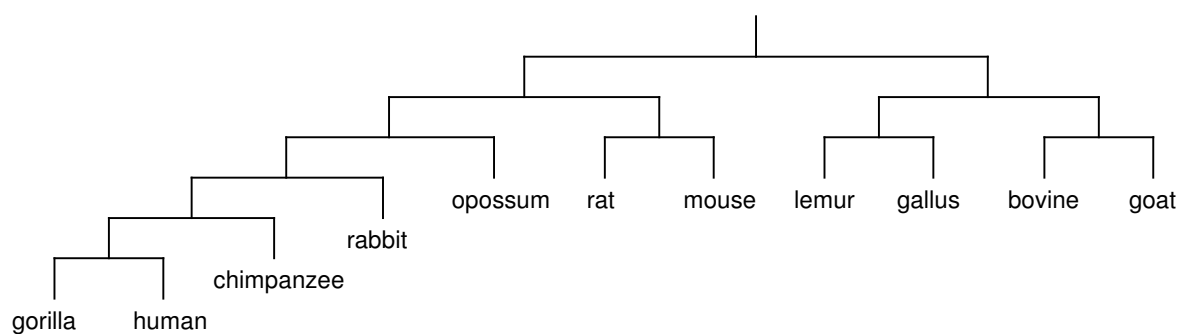


Figure A.22: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Length Weighted Index on the RAW sets (on NoRC setting).

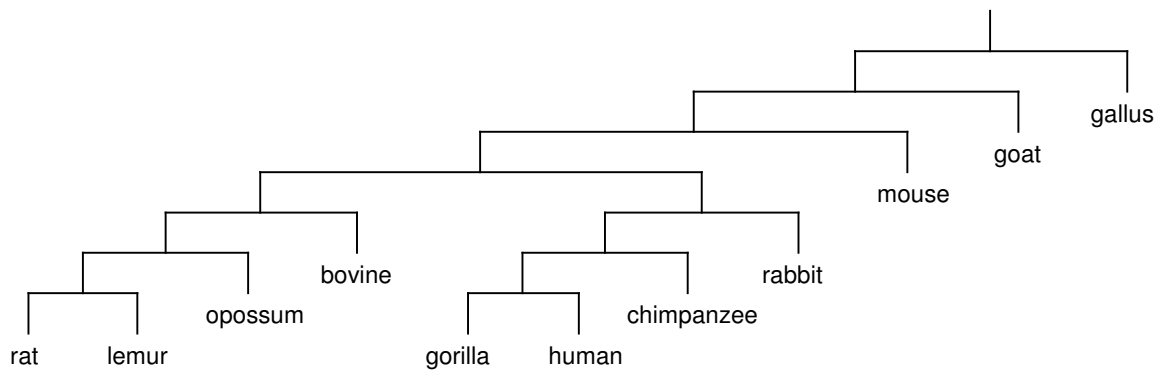


Figure A.23: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the GC content on the Symmetric Difference of the MAW sets (on NoRC setting).

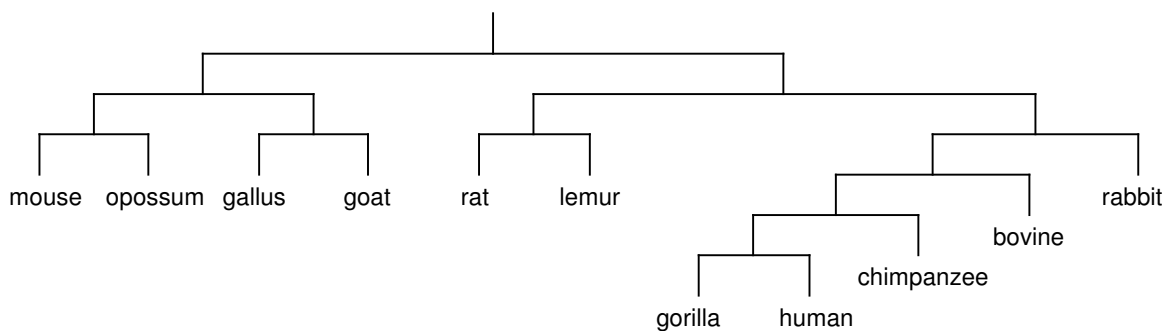


Figure A.24: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the GC content on the Symmetric Difference of the MAW sets (on NoRC setting).

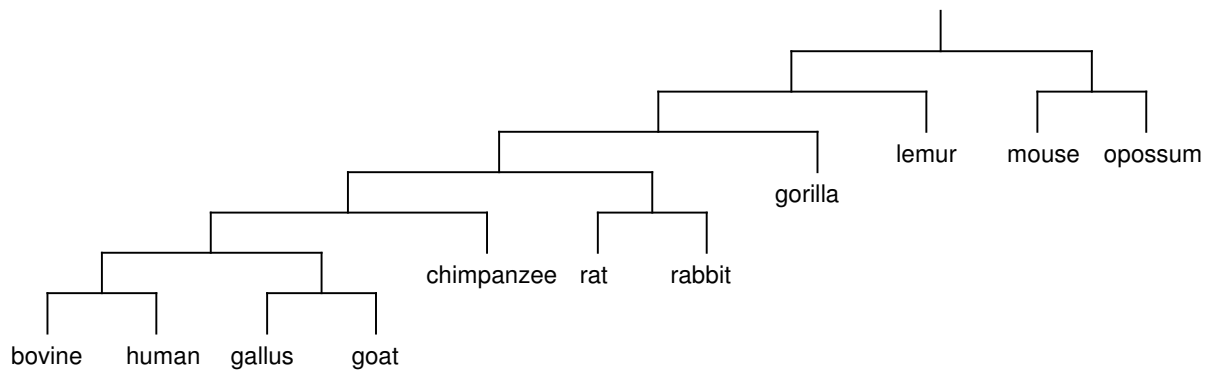


Figure A.25: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the GC content on the Intersection of the MAW sets (on NoRC setting).

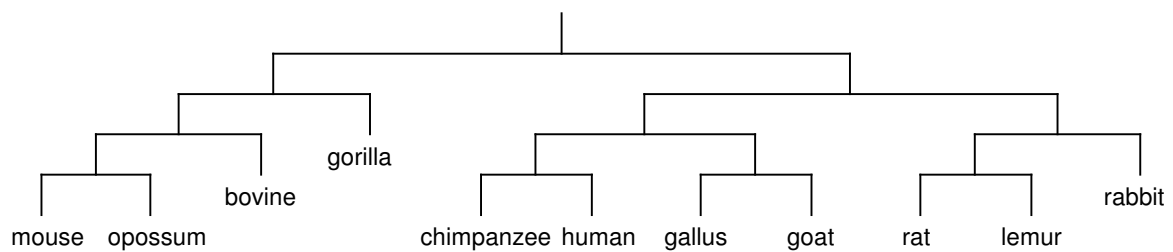


Figure A.26: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the GC content on the Intersection of the MAW sets (on NoRC setting).

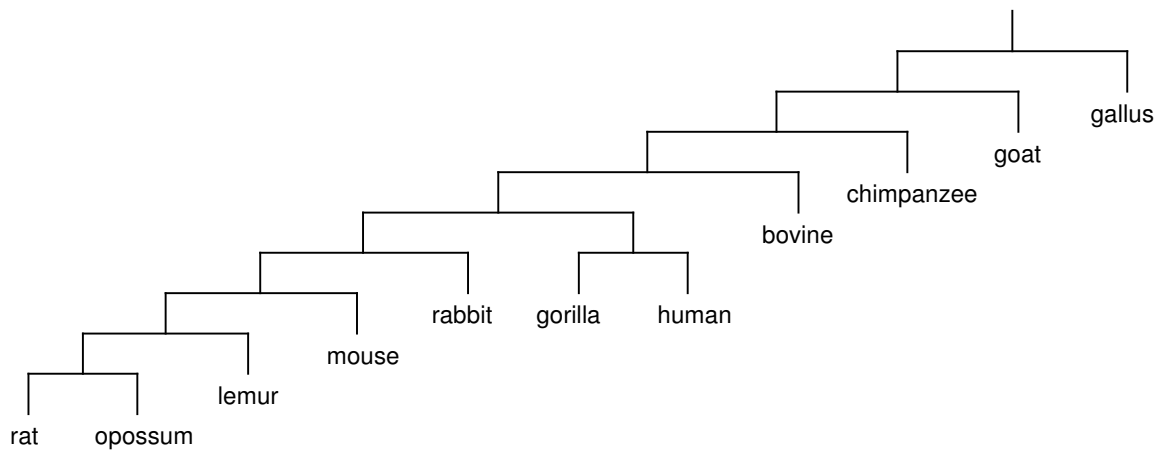


Figure A.27: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the GC content on the RAW sets (on NoRC setting).

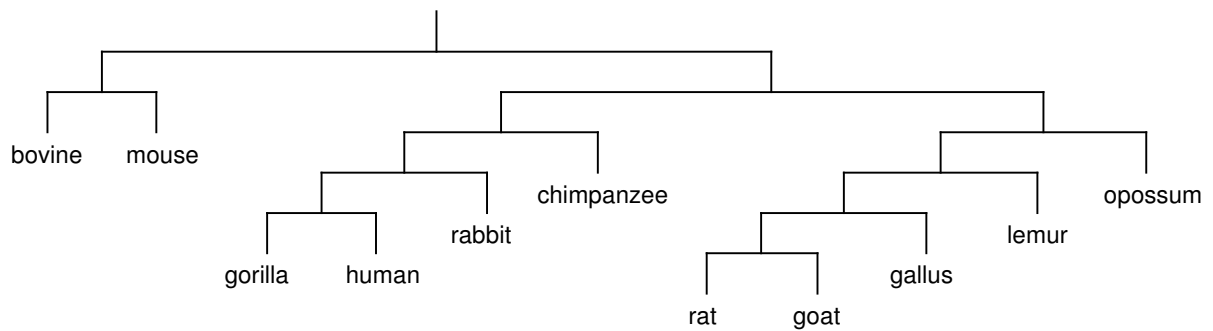


Figure A.28: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the GC content on the RAW sets (on NoRC setting).

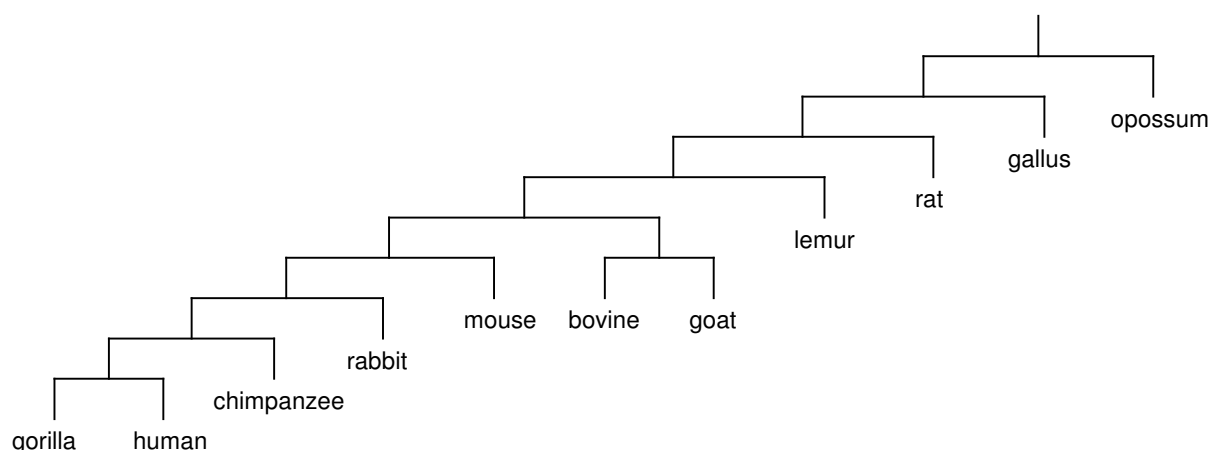


Figure A.29: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Jaccard Distance of the MAW sets (on NoRC setting).

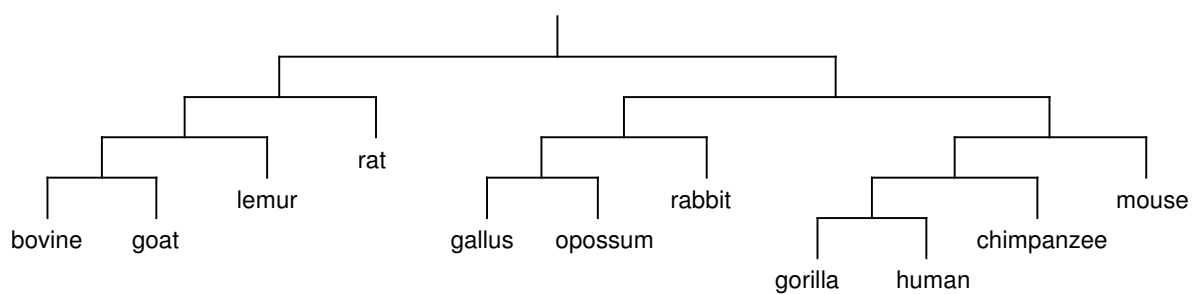


Figure A.30: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Jaccard Distance of the MAW sets (on NoRC setting).

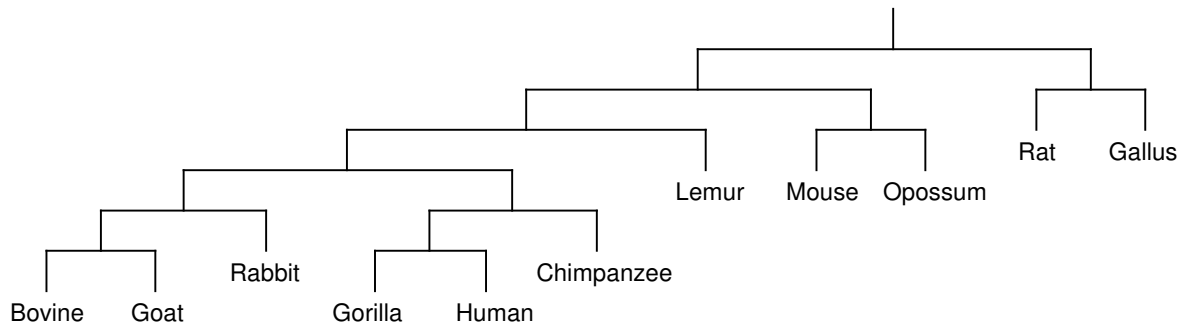


Figure A.31: The β -globin gene tree for the 11 species computed using UPGMA algorithm applied on the distance matrix computed based on the Total Variation Distance of the MAW sets (on NoRC setting).

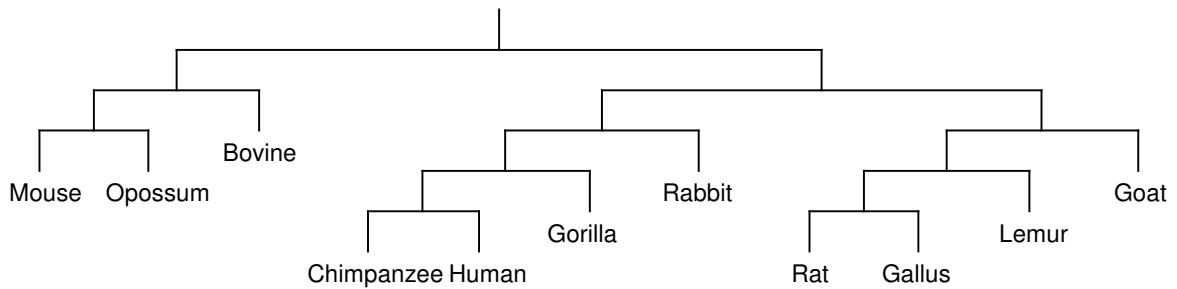


Figure A.32: The β -globin gene tree for the 11 species computed using Neighbor Joining algorithm applied on the distance matrix computed based on the Total Variation Distance of the MAW sets (on NoRC setting).

Appendix B

Supplementary Materials for Species Tree Estimation Using DCM Boosted QFM

These supplementary materials provide estimated species trees in different model conditions resulting out of our experiments described in Chapter 7. Notably, we have used the mammalian simulated dataset that was prepared by Mirarab et al. [199] and was also used in [23]. The dataset contains 20 replicates each for 11 different model conditions. We have drawn the species tree estimated by DCM Boosted QFM for the first replicate for each of the 11 model conditions.

B.1 Species Trees on the 37 Taxa Mammalian Dataset

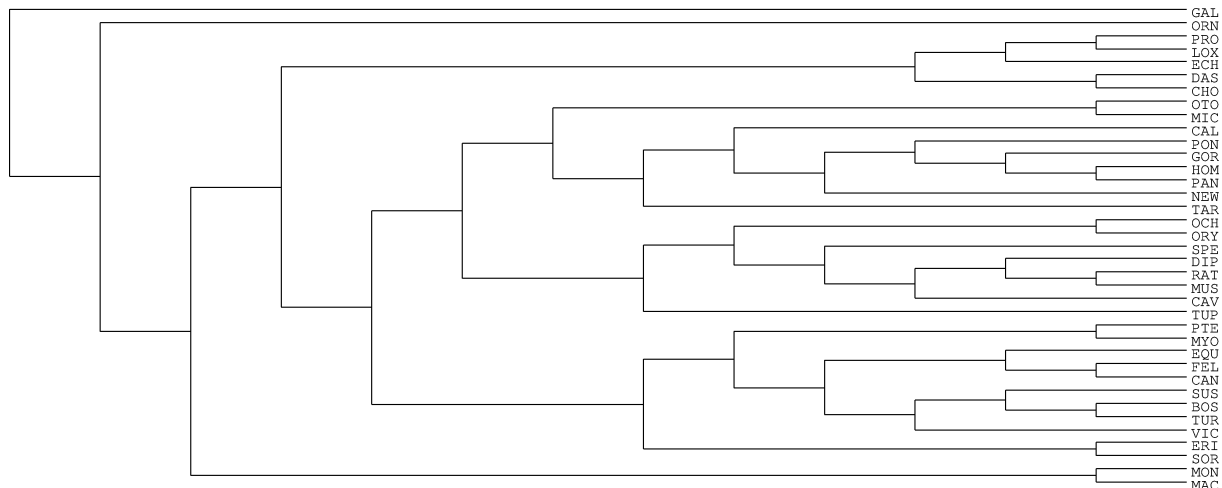


Figure B.1: The model species tree for the 37-taxon mammalian dataset used in this study.

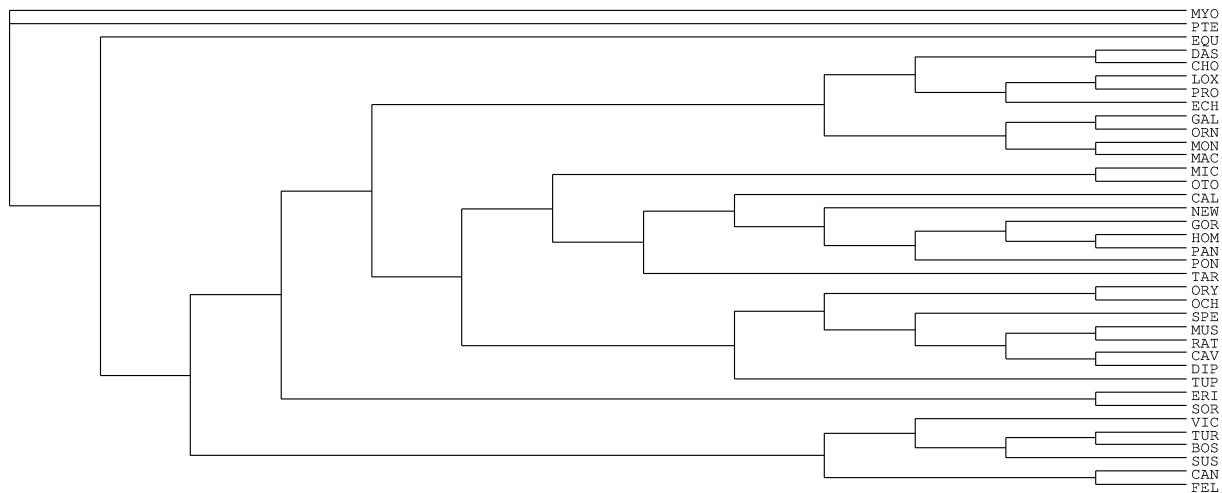


Figure B.2: Species tree generated by DCM boosted QFM on the simulated dataset with the 37 taxa used in this study. The model condition used to generate this tree had 0.2X level of ILS, 200 genes of 500 bp each. The first replicate (out of the 20 replicates) was used. Boosting with 2 and 5 iterations produced the same tree.

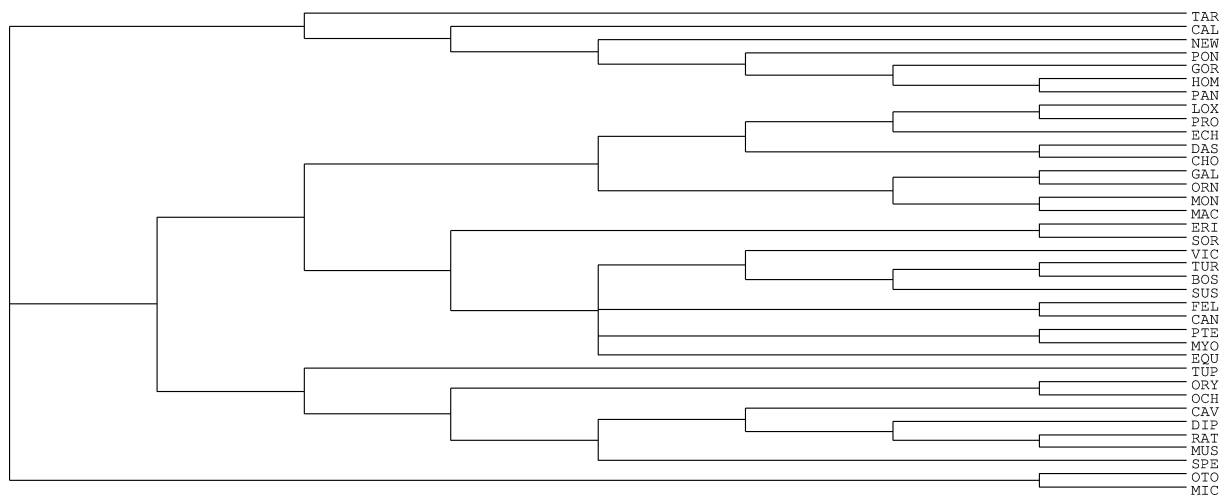


Figure B.3: Species tree generated by DCM boosted QFM on the simulated dataset with the 37 taxa used in this study. The model condition used to generate this tree had 0.5X level of ILS, 200 genes of 500 bp each. The first replicate (out of the 20 replicates) was used. Boosting with 2 and 5 iterations produced the same tree.

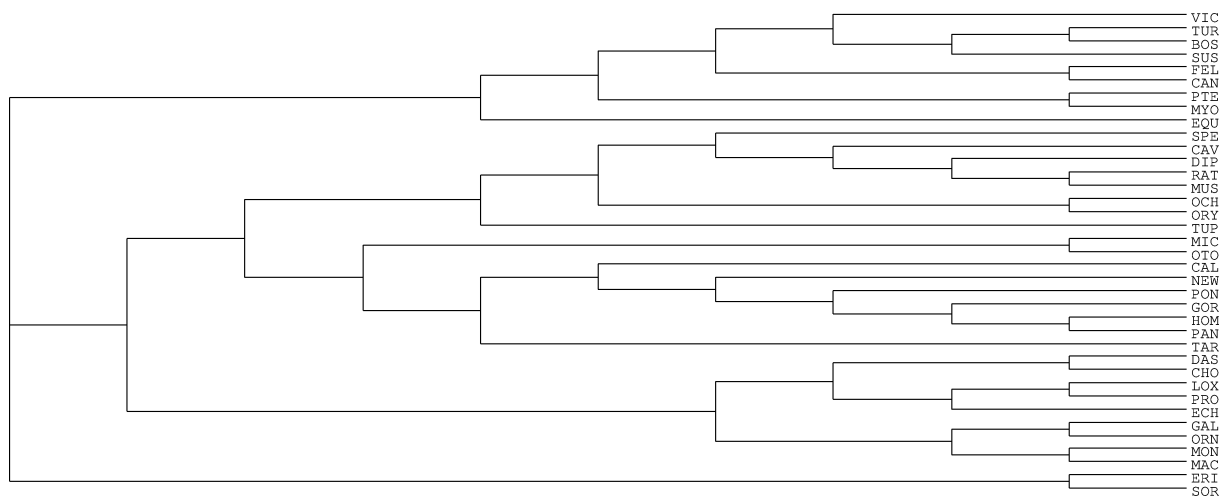


Figure B.4: Species tree generated by DCM boosted QFM on the simulated dataset with the 37 taxa used in this study. The model condition used to generate this tree had 1X level of ILS, 200 genes of 500 bp each. The first replicate (out of the 20 replicates) was used. Boosting with 2 and 5 iterations produced the same tree.

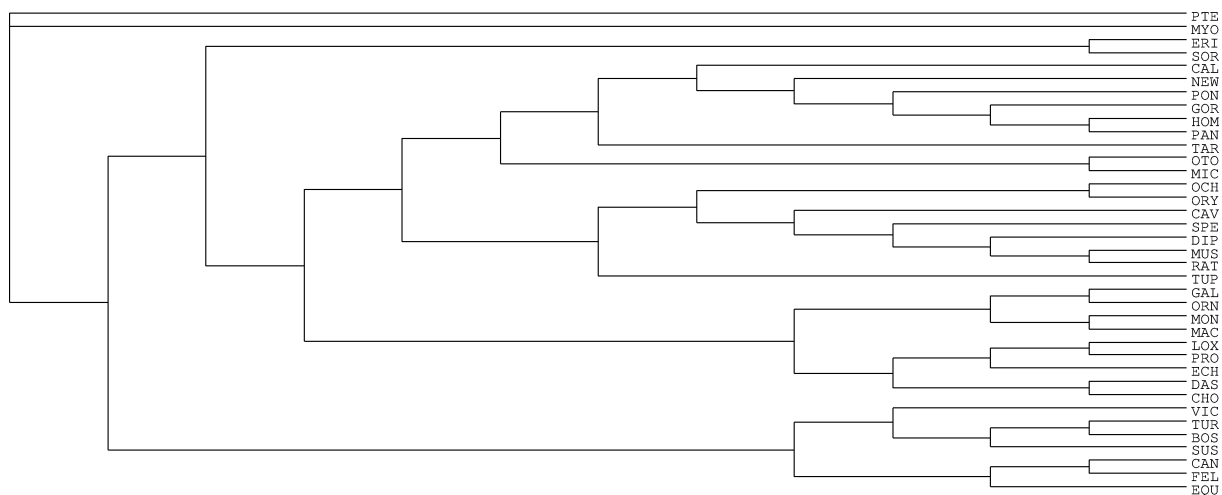


Figure B.5: Species tree generated by DCM boosted QFM on the simulated dataset with the 37 taxa used in this study. The model condition used to generate this tree had 2X level of ILS, 200 genes of 500 bp each. The first replicate (out of the 20 replicates) was used. Boosting with 2 and 5 iterations produced the same tree.

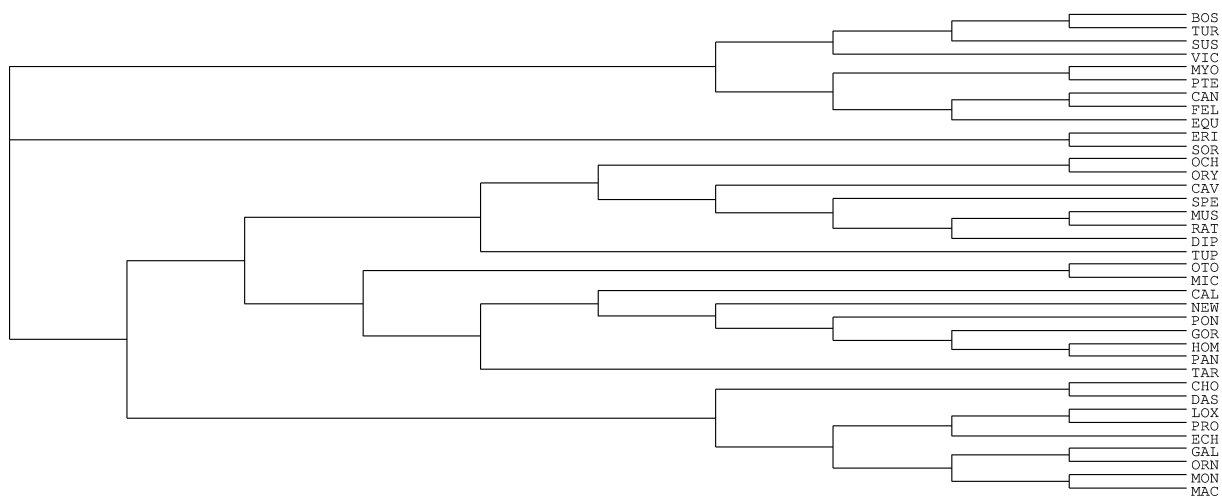


Figure B.6: Species tree generated by DCM boosted QFM on the simulated dataset with the 37 taxa used in this study. The model condition used to generate this tree had 1X level of ILS, 50 genes of 500 bp each. The first replicate (out of the 20 replicates) was used. Boosting with 2 and 5 iterations produced the same tree.

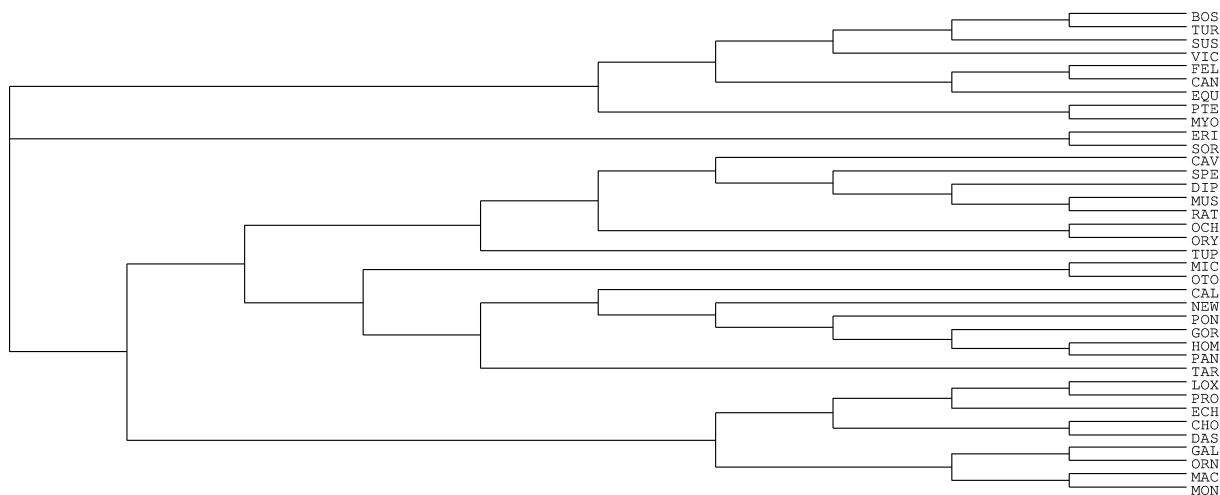


Figure B.7: Species tree generated by DCM boosted QFM on the simulated dataset with the 37 taxa used in this study. The model condition used to generate this tree had 1X level of ILS, 100 genes of 500 bp each. The first replicate (out of the 20 replicates) was used. Boosting with 2 and 5 iterations produced the same tree.

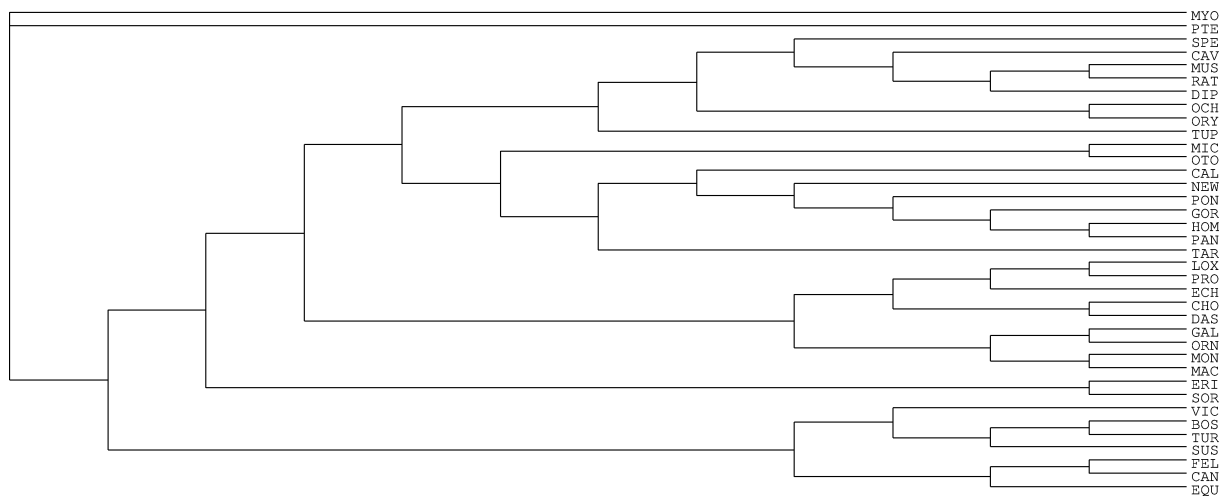


Figure B.8: Species tree generated by DCM boosted QFM on the simulated dataset with the 37 taxa used in this study. The model condition used to generate this tree had 1X level of ILS, 400 genes of 500 bp each. The first replicate (out of the 20 replicates) was used. Boosting with 2 and 5 iterations produced the same tree.

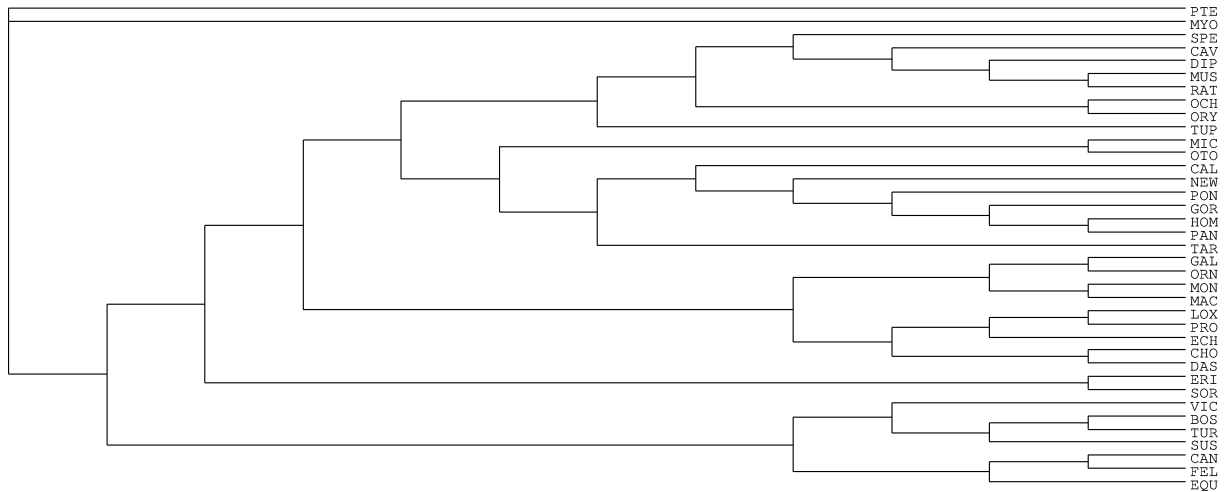


Figure B.9: Species tree generated by DCM boosted QFM on the simulated dataset with the 37 taxa used in this study. The model condition used to generate this tree had 1X level of ILS, 800 genes of 500 bp each. The first replicate (out of the 20 replicates) was used. Boosting with 2 and 5 iterations produced the same tree.

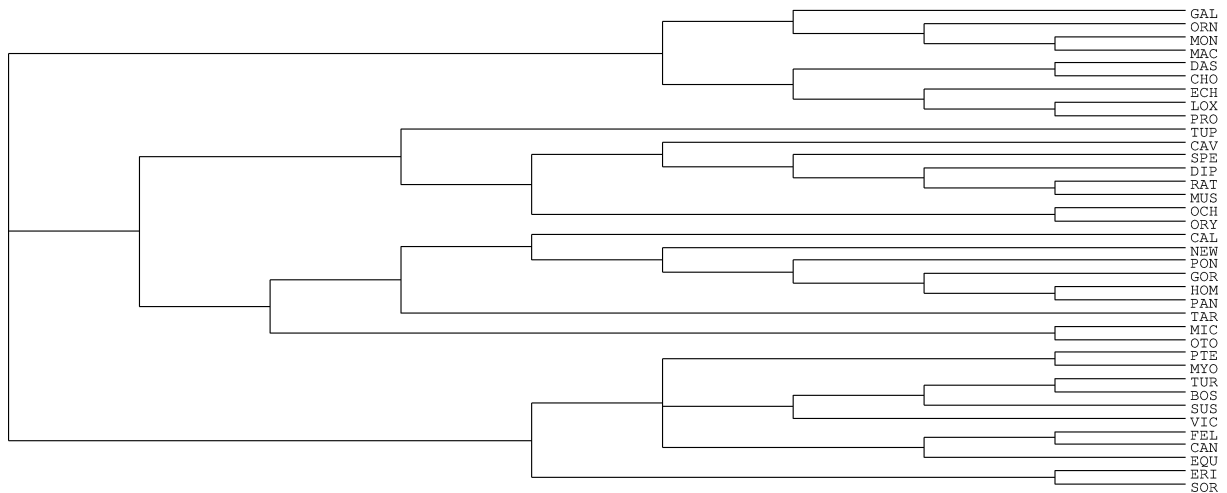


Figure B.10: Species tree generated by DCM boosted QFM on the simulated dataset with the 37 taxa used in this study. The model condition used to generate this tree had 1X level of ILS, 200 genes of 250 bp each. The first replicate (out of the 20 replicates) was used. Boosting with 2 and 5 iterations produced the same tree.

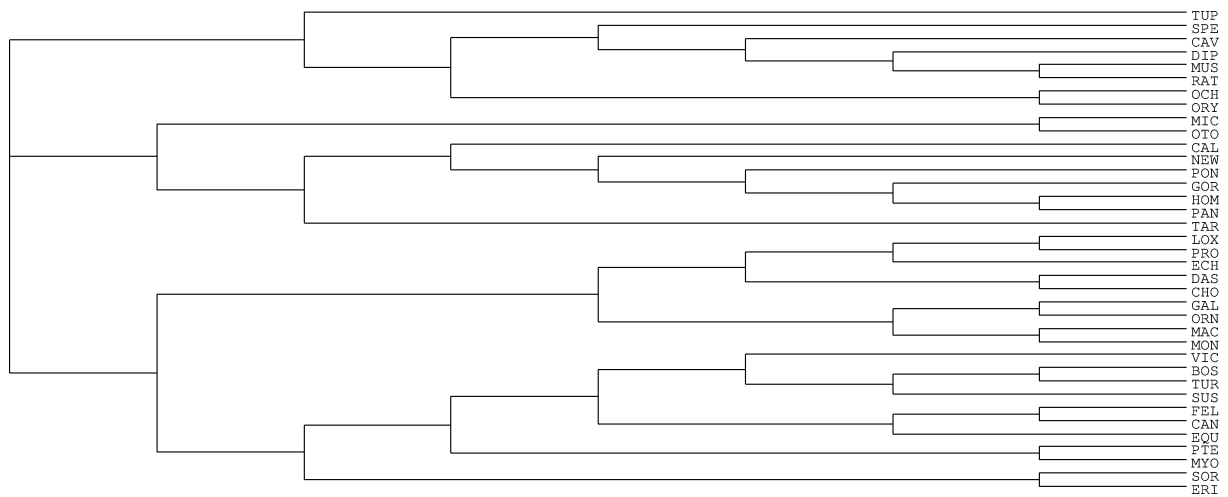


Figure B.11: Species tree generated by DCM boosted QFM on the simulated dataset with the 37 taxa used in this study. The model condition used to generate this tree had 1X level of ILS, 200 genes of 1000 bp each. The first replicate (out of the 20 replicates) was used. Boosting with 2 and 5 iterations produced the same tree.

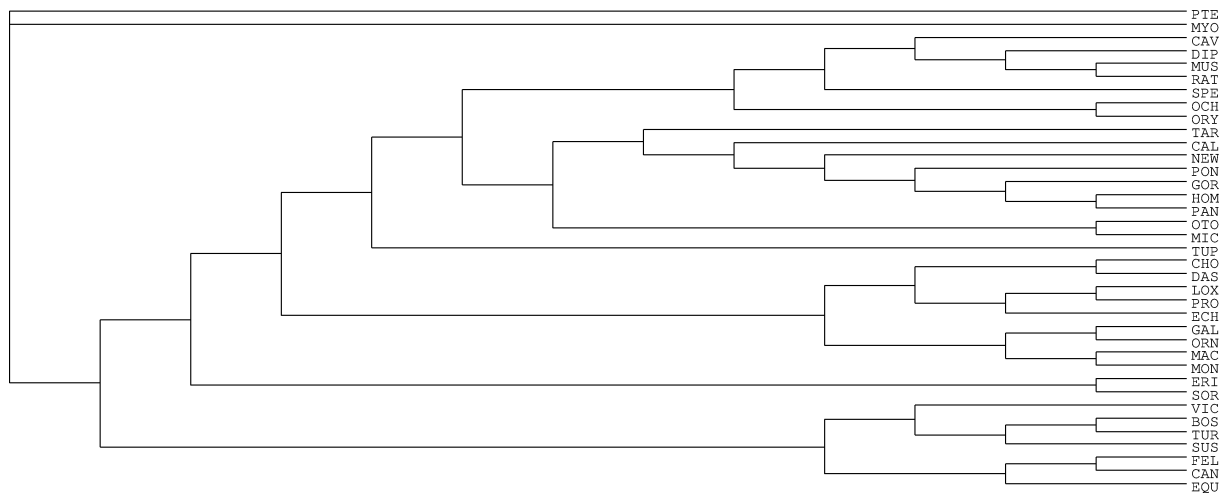


Figure B.12: Species tree generated by DCM boosted QFM on the simulated dataset with the 37 taxa used in this study. The model condition used to generate this tree had 1X level of ILS, 200 true genes. The first replicate (out of the 20 replicates) was used. Boosting with 2 and 5 iterations produced the same tree.

Bibliography

- [1] Lambda repressor (PDB 1LMB), by Zephyris at the English language Wikipedia, CC BY-SA 3.0. <https://commons.wikimedia.org/w/index.php?curid=2426895>. [Last accessed on 26-Feb-2018].
- [2] National Center for Biotechnology Information Search database. <https://www.ncbi.nlm.nih.gov/>. [Last accessed on 06-Mar-2018].
- [3] Question: Blastclust Standalone Download Address? <https://www.biostars.org/p/92324/>. [Last accessed on 06-Mar-2018].
- [4] Restriction enzyme, by Zephyris at English Wikipedia - Transferred from en.wikipedia to Commons., CC BY-SA 3.0. <https://commons.wikimedia.org/w/index.php?curid=2426900>. [Last accessed on 26-Feb-2018].
- [5] SCRATCH Protein Predictor. http://scratch.proteomics.ics.uci.edu/cgi-bin/new_server/sql_predict.cgi. [Last accessed on 25-Mar-2018].
- [6] Uniprot Database. <http://www.uniprot.org/>. [Last accessed on 06-Jun-2018].
- [7] VaxiJen v2.0. <http://www.ddg-pharmfac.net/vaxijen/VaxiJen/VaxiJen.html>. [Last accessed on 25-Mar-2018].
- [8] C. Acquisti, G. Poste, D. Curtiss, and S. Kumar. Nullomers: really a matter of natural selection? *PloS one*, 2(10):e1022, 2007.
- [9] G. Ada. The traditional vaccines: an overview. *New generation vaccines*, pages 12–23, 1997.

- [10] J. Ahmad, F. Javed, and M. Hayat. Intelligent computational model for classification of sub-Golgi protein using oversampling and fisher feature selection methods. *Artificial Intelligence in Medicine*, 78:14–22, 2017.
- [11] S. Ahmad and A. Sarai. Moment-based prediction of DNA-binding proteins. *Journal of molecular biology*, 341(1):65–71, 2004.
- [12] E. Altindis, R. Cozzi, B. Di Palo, F. Necchi, R. P. Mishra, M. R. Fontana, M. Soriani, F. Bagnoli, D. Maione, G. Grandi, et al. Protectome analysis: a new selective bioinformatics tool for bacterial vaccine candidate discovery. *Molecular & Cellular Proteomics*, 14(2):418–429, 2015.
- [13] D. G. Altman and J. M. Bland. STATISTICS NOTES-DIAGNOSTIC-TESTS-1-SENSITIVITY AND SPECIFICITY. 3., 1994.
- [14] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.
- [15] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.
- [16] H. R. Ansari, D. R. Flower, and G. Raghava. AntigenDB: an immunoinformatics database of pathogen antigens. *Nucleic acids research*, 38(suppl_1):D847–D853, 2009.
- [17] T. Arendt, H. Zveuintshva, and T. Lkontovich. Dendritic changes in the basal nucleus of Meynert and in the diagonal band nucleus in Alzheimer’s diseasea quantitative Golgi investigation. *Neuroscience*, 19(4):1265–1278, 1986.
- [18] N. Ariel, A. Zvi, H. Grosfeld, O. Gat, Y. Inbar, B. Velan, S. Cohen, and A. Shaferman. Search for potential vaccine candidate open reading frames in the Bacillus anthracis virulence plasmid pXO1: in silico and in vitro screening. *Infection and immunity*, 70(12):6817–6827, 2002.

- [19] N. Arinaminpathy, O. Ratmann, K. Koelle, S. L. Epstein, G. E. Price, C. Viboud, M. A. Miller, and B. T. Grenfell. Impact of cross-protective vaccines on epidemiological and evolutionary dynamics of influenza. *Proceedings of the National Academy of Sciences*, 109(8):3173–3177, 2012.
- [20] S. L. Baldwin, V. A. Reese, D. H. Po-wei, E. A. Beebe, B. K. Podell, S. G. Reed, and R. N. Coler. Protection and long-lived immunity induced by the ID93/GLA-SE vaccine candidate against a clinical Mycobacterium tuberculosis isolate. *Clinical and Vaccine Immunology*, 23(2):137–147, 2016.
- [21] C. Barton, A. Heliou, L. Mouchard, and S. P. Pissis. Linear-time computation of minimal absent words using suffix array. *BMC bioinformatics*, 15(1):388, 2014.
- [22] B. R. Baum and M. A. Ragan. The MRP method. In *Phylogenetic supertrees*, pages 17–34. Springer, 2004.
- [23] M. S. Bayzid, T. Hunt, and T. Warnow. Disk covering methods improve phylogenomic analyses. *BMC genomics*, 15(6):S7, 2014.
- [24] M. S. Bayzid and T. Warnow. Gene tree parsimony for incomplete gene trees: addressing true biological loss. *Algorithms for Molecular Biology*, 13(1):1, 2018.
- [25] M.-P. Béal, M. Crochemore, F. Mignosi, A. Restivo, and M. Sciortino. Computing forbidden words of regular languages. *Fundamenta Informaticae*, 56(1-2):121–135, 2003.
- [26] M.-P. Béal, F. Fiorenzi, and F. Mignosi. Minimal forbidden patterns of multi-dimensional shifts. *International journal of algebra and computation*, 15(01):73–93, 2005.
- [27] M.-P. Béal, F. Mignosi, and A. Restivo. Minimal forbidden words and symbolic dynamics. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 555–566. Springer, 1996.

- [28] M. Behbahani, H. Mohabatkar, and M. Nosrati. Analysis and comparison of lignin peroxidases between fungi and bacteria using three different modes of Chous general pseudo amino acid composition. *Journal of theoretical biology*, 411:1–5, 2016.
- [29] J. D. Bendtsen, H. Nielsen, G. von Heijne, and S. Brunak. Improved prediction of signal peptides: SignalP 3.0. *Journal of molecular biology*, 340(4):783–795, 2004.
- [30] M. Bhasin and G. P. Raghava. Classification of nuclear receptors based on amino acid composition and dipeptide composition. *Journal of Biological Chemistry*, 279(22):23262–23266, 2004.
- [31] J. X. Binder, S. Pletscher-Frankild, K. Tsafou, C. Stolte, S. I. ODonoghue, R. Schneider, and L. J. Jensen. COMPARTMENTS: unification and visualization of protein subcellular localization evidence. *Database*, 2014:bau012, 2014.
- [32] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.
- [33] B. Boussau, G. J. Szöllósi, L. Duret, M. Gouy, E. Tannier, and V. Daubin. Genome-scale coestimation of species and gene trees. *Genome research*, 23(2):323–330, 2013.
- [34] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [35] M. J. Buck and J. D. Lieb. ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics*, 83(3):349–360, 2004.
- [36] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.
- [37] D. Butler and L. Morello. Ebola by the numbers: The size, spread and cost of an outbreak. *Nature*, 514(7522), 2014.

- [38] D.-S. Cao, Q.-S. Xu, and Y.-Z. Liang. propy: a tool to generate various modes of Chous PseAAC. *Bioinformatics*, 29(7):960–962, 2013.
- [39] S. Chairungsee and M. Crochemore. Using minimal absent words to build phylogeny. *Theoretical Computer Science*, 450:109–116, 2012.
- [40] D. N. Chakravarti, M. J. Fiske, L. D. Fletcher, and R. J. Zagursky. Application of genomics and proteomics for identification of bacterial gene products as potential vaccine candidates. *Vaccine*, 19(6):601–612, 2000.
- [41] J.-M. Chang, E. C.-Y. Su, A. Lo, H.-S. Chiu, T.-Y. Sung, and W.-L. Hsu. PSLDoc: Protein subcellular localization prediction based on gapped-dipeptides and probabilistic latent semantic analysis. *Proteins: Structure, Function, and Bioinformatics*, 72(2):693–710, 2008.
- [42] Y. Chang, N. T. Brewer, A. C. Rinas, K. Schmitt, and J. S. Smith. Evaluating the impact of human papillomavirus vaccines. *Vaccine*, 27(32):4355–4362, 2009.
- [43] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- [44] W. Chen, P. Feng, H. Yang, H. Ding, H. Lin, and K.-C. Chou. iRNA-AI: identifying the adenosine to inosine editing sites in RNA sequences. *Oncotarget*, 8(3):4208, 2017.
- [45] W. Chen, P. Feng, H. Yang, H. Ding, H. Lin, and K.-C. Chou. iRNA-3typeA: identifying 3-types of modification at RNAs adenosine sites. *Molecular Therapy-Nucleic Acids*, 2018.
- [46] W. Chen, P.-M. Feng, H. Lin, and K.-C. Chou. iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition. *Nucleic acids research*, 41(6):e68–e68, 2013.

- [47] W. Chen, T.-Y. Lei, D.-C. Jin, H. Lin, and K.-C. Chou. PseKNC: a flexible web server for generating pseudo K-tuple nucleotide composition. *Analytical biochemistry*, 456:53–60, 2014.
- [48] W. Chen, H. Lin, and K.-C. Chou. Pseudo nucleotide composition or PseKNC: an effective formulation for analyzing genomic sequences. *Molecular BioSystems*, 11(10):2620–2634, 2015.
- [49] J. Cheng, A. Z. Randall, M. J. Sweredoski, and P. Baldi. SCRATCH: a protein structure and structural feature prediction server. *Nucleic acids research*, 33(suppl_2):W72–W76, 2005.
- [50] J. Cheng, M. J. Sweredoski, and P. Baldi. DOMpro: protein domain prediction using profiles, secondary structure, relative solvent accessibility, and recursive neural networks. *Data Mining and Knowledge Discovery*, 13(1):1–10, 2006.
- [51] X. Cheng, X. Xiao, and K.-C. Chou. pLoc-mEuk: Predict subcellular localization of multi-label eukaryotic proteins by extracting the key GO information into general PseAAC. *Genomics*, 2017.
- [52] X. Cheng, X. Xiao, and K.-C. Chou. pLoc-mPlant: predict subcellular localization of multi-location plant proteins by incorporating the optimal GO information into general PseAAC. *Molecular BioSystems*, 13(9):1722–1727, 2017.
- [53] X. Cheng, X. Xiao, and K.-C. Chou. pLoc-mVirus: predict subcellular localization of multi-location virus proteins via incorporating the optimal GO information into general PseAAC. *Gene*, 628:315–321, 2017.
- [54] X. Cheng, S.-G. Zhao, W.-Z. Lin, X. Xiao, and K.-C. Chou. pLoc-mAnimal: predict subcellular localization of animal proteins with both single and multiple sites. *Bioinformatics*, 33(22):3524–3531, 2017.

- [55] X. Cheng, S.-G. Zhao, X. Xiao, and K.-C. Chou. iATC-mISF: a multi-label classifier for predicting the classes of anatomical therapeutic chemicals. *Bioinformatics*, 33(3):341–346, 2016.
- [56] X. Cheng, S.-G. Zhao, X. Xiao, and K.-C. Chou. iATC-mHyb: a hybrid multi-label classifier for predicting the classification of anatomical therapeutic chemicals. *Oncotarget*, 8(35):58494, 2017.
- [57] C.-C. Chou, T.-W. Lin, C.-Y. Chen, and A. H.-J. Wang. Crystal structure of the hyperthermophilic archaeal DNA-binding protein Sso10b2 at a resolution of 1.85 Angstroms. *Journal of bacteriology*, 185(14):4066–4073, 2003.
- [58] K.-C. Chou. A novel approach to predicting protein structural classes in a (20–1)-D amino acid composition space. *Proteins: Structure, Function, and Bioinformatics*, 21(4):319–344, 1995.
- [59] K.-C. Chou. Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins: Structure, Function, and Bioinformatics*, 43(3):246–255, 2001.
- [60] K.-C. Chou. Prediction of signal peptides using scaled window. *peptides*, 22(12):1973–1979, 2001.
- [61] K.-C. Chou. Some remarks on protein attribute prediction and pseudo amino acid composition. *Journal of theoretical biology*, 273(1):236–247, 2011.
- [62] K.-C. Chou. Some remarks on predicting multi-label attributes in molecular biosystems. *Molecular Biosystems*, 9(6):1092–1100, 2013.
- [63] K.-C. Chou. Impacts of bioinformatics to medicinal chemistry. *Medicinal chemistry*, 11(3):218–234, 2015.
- [64] K.-C. Chou. An unprecedented revolution in medicinal chemistry driven by the progress of biological science. *Current topics in medicinal chemistry*, 17(21):2337–2358, 2017.

- [65] S. Y. Chowdhury, S. Shatabda, and A. Dehzangi. iDNAProt-ES: Identification of DNA-binding proteins using evolutionary and structural features. *Scientific Reports*, 7(1):14938, 2017.
- [66] G. M. Cooper and R. E. Hausman. *The cell*. Sinauer Associates Sunderland, 2000.
- [67] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [68] M. Crochemore, F. Mignosi, and A. Restivo. Automata and forbidden words. *Information Processing Letters*, 67(3):111–117, 1998.
- [69] M. Crochemore, F. Mignosi, A. Restivo, and S. Salemi. Text compression using antidictionaries. In *International Colloquium on Automata, Languages, and Programming*, pages 261–270. Springer, 1999.
- [70] M. Crochemore and G. Navarro. Improved antidictionary based compression. In *Computer Science Society, 2002. SCCC 2002. Proceedings. 22nd International Conference of the Chilean*, pages 7–13. IEEE, 2002.
- [71] M. P. Cummings. Transmission patterns of eukaryotic transposable elements: arguments for and against horizontal transfer. *Trends in ecology & evolution*, 9(4):141–145, 1994.
- [72] D. H. Davies, X. Liang, J. E. Hernandez, A. Randall, S. Hirst, Y. Mu, K. M. Romero, T. T. Nguyen, M. Kalantari-Dehaghi, S. Crotty, et al. Profiling the humoral immune response to infection by using proteome microarrays: high-throughput vaccine and diagnostic antigen discovery. *Proceedings of the National Academy of Sciences of the United States of America*, 102(3):547–552, 2005.
- [73] J. Davis and M. Goadrich. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

- [74] M. DeGiorgio and J. H. Degnan. Fast and consistent estimation of species trees using supermatrix rooted triples. *Molecular biology and evolution*, 27(3):552–569, 2009.
- [75] J. H. Degnan and N. A. Rosenberg. Discordance of species trees with their most likely gene trees. *PLoS genetics*, 2(5):e68, 2006.
- [76] J. H. Degnan and L. A. Salter. Gene tree distributions under the coalescent process. *Evolution*, 59(1):24–37, 2005.
- [77] A. Dembo and S. Karlin. Poisson approximations for r-scan processes. *The Annals of Applied Probability*, pages 329–357, 1992.
- [78] R. Desper and O. Gascuel. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. *Journal of computational biology*, 9(5):687–705, 2002.
- [79] H. Ding, S.-H. Guo, E.-Z. Deng, L.-F. Yuan, F.-B. Guo, J. Huang, N. Rao, W. Chen, and H. Lin. Prediction of Golgi-resident protein types by using feature selection technique. *Chemometrics and Intelligent Laboratory Systems*, 124:9–13, 2013.
- [80] H. Ding, L. Liu, F.-B. Guo, J. Huang, and H. Lin. Identify Golgi protein types with modified mahalanobis discriminant algorithm and pseudo amino acid composition. *Protein and peptide letters*, 18(1):58–63, 2011.
- [81] D. J. Dittman, T. M. Khoshgoftaar, and A. Napolitano. The effect of data sampling when using random forest on imbalanced bioinformatics data. In *Information Reuse and Integration (IRI), 2015 IEEE International Conference on*, pages 457–463. IEEE, 2015.
- [82] Q. Dong, S. Wang, K. Wang, X. Liu, and B. Liu. Identification of DNA-binding proteins by auto-cross covariance transformation. In *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*, pages 470–475. IEEE, 2015.

- [83] I. A. Doytchinova and D. R. Flower. Identifying candidate subunit vaccines using an alignment-independent method based on principal amino acid properties. *Vaccine*, 25(5):856–866, 2007.
- [84] I. A. Doytchinova and D. R. Flower. VaxiJen: a server for prediction of protective antigens, tumour antigens and subunit vaccines. *BMC bioinformatics*, 8(1):4, 2007.
- [85] I. A. Doytchinova and D. R. Flower. Bioinformatic approach for identifying parasite and fungal candidate subunit vaccines. *Open Vaccine J*, 1(1):4, 2008.
- [86] P. Du, S. Gu, and Y. Jiao. PseAAC-General: fast building various modes of general form of Chous pseudo-amino acid composition for large-scale protein datasets. *International journal of molecular sciences*, 15(3):3495–3506, 2014.
- [87] I. Dubchak, I. B. Muchnik, and S.-H. Kim. Protein folding class predictor for SCOP: approach based on global descriptors. In *ismb*, pages 104–107, 1997.
- [88] J. D. Durrant and J. A. McCammon. Molecular dynamics simulations and drug discovery. *BMC biology*, 9(1):71, 2011.
- [89] Y. El-Manzalawy, D. Dobbs, and V. Honavar. Predicting protective bacterial antigens using random forest classifiers. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, pages 426–433. ACM, 2012.
- [90] D. D. Elsberry and M. T. Rise. Techniques for treating neurodegenerative disorders by infusion of nerve growth factors into the brain, Mar. 28 2000. US Patent 6,042,579.
- [91] J. Evans, L. Sheneman, and J. Foster. Relaxed neighbor joining: a fast distance-based phylogenetic tree construction method. *Journal of molecular evolution*, 62(6):785–792, 2006.
- [92] G.-L. Fan and Q.-Z. Li. Predicting protein submitochondria locations by combining different descriptors into the general form of Chous pseudo amino acid composition. *Amino Acids*, 43(2):545–555, 2012.

- [93] Y. Fang, Y. Guo, Y. Feng, and M. Li. Predicting DNA-binding proteins: approached from Chou’s pseudo amino acid composition and other specific sequence features. *Amino acids*, 34(1):103–109, 2008.
- [94] E. Faraggi, T. Zhang, Y. Yang, L. Kurgan, and Y. Zhou. SPINE X: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *Journal of computational chemistry*, 33(3):259–267, 2012.
- [95] T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.
- [96] J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of molecular evolution*, 17(6):368–376, 1981.
- [97] P. Feng, H. Ding, H. Yang, W. Chen, H. Lin, and K.-C. Chou. iRNA-PseColl: Identifying the occurrence sites of different RNA modifications by incorporating collective effects of nucleotides into PseKNC. *Molecular Therapy-Nucleic Acids*, 7:155–163, 2017.
- [98] G. Fici, F. Mignosi, A. Restivo, and M. Sciortino. Word assembly through minimal forbidden words. *Theoretical Computer Science*, 359(1-3):214–230, 2006.
- [99] C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *Papers on Twenty-five years of electronic design automation*, pages 241–247. ACM, 1988.
- [100] A. E. Fiore, C. B. Bridges, and N. J. Cox. Seasonal influenza vaccines. In *Vaccines for Pandemic Influenza*, pages 43–82. Springer, 2009.
- [101] W. M. Fitch. Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Biology*, 20(4):406–416, 1971.

- [102] D. R. Flower, I. K. Macdonald, K. Ramakrishnan, M. N. Davies, and I. A. Doytchinova. Computer aided selection of candidate vaccine antigens. *Immunome research*, 6(2):S1, 2010.
- [103] T. Folaranmi, L. Rubin, S. W. Martin, M. Patel, and J. R. MacNeil. Use of serogroup B meningococcal vaccines in persons aged ≥ 10 years at increased risk for serogroup B meningococcal disease: recommendations of the Advisory Committee on Immunization Practices, 2015. *MMWR. Morbidity and mortality weekly report*, 64(22):608–612, 2015.
- [104] E. Frank, M. Hall, and I. H. Witten. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques". 2016.
- [105] K. Freeman, M. Gwadz, and D. Shore. Molecular and genetic analysis of the toxic effect of RAP1 overexpression in yeast. *Genetics*, 141(4):1253–1262, 1995.
- [106] M. Gao and J. Skolnick. DBD-Hunter: a knowledge-based method for the prediction of DNA–protein interactions. *Nucleic acids research*, 36(12):3978–3992, 2008.
- [107] M. Gao and J. Skolnick. A threading-based method for the prediction of DNA-binding proteins with application to the human genome. *PLoS computational biology*, 5(11):e1000567, 2009.
- [108] S. P. Garcia and A. J. Pinho. Minimal absent words in four human genome assemblies. *PLoS One*, 6(12):e29344, 2011.
- [109] O. Gascuel. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Molecular biology and evolution*, 14(7):685–695, 1997.
- [110] P. Gilchuk, F. C. Knight, J. T. Wilson, and S. Joyce. Eliciting Epitope-Specific CD8+ T Cell Response by Immunization with Microbial Protein Antigens Formulated with α -Galactosylceramide: Theory, Practice, and Protocols. In *Vaccine Adjuvants*, pages 321–352. Springer, 2017.

- [111] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.
- [112] M. Goodman, J. Czelusniak, G. W. Moore, A. E. Romero-Herrera, and G. Matsuda. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Biology*, 28(2):132–163, 1979.
- [113] S. Guindon and O. Gascuel. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Systematic biology*, 52(5):696–704, 2003.
- [114] J. X. Guo and N. N. Rao. The influence of dipeptide composition on protein folding rates. In *Advanced Materials Research*, volume 378, pages 157–160. Trans Tech Publ, 2012.
- [115] S.-H. Guo, E.-Z. Deng, L.-Q. Xu, H. Ding, H. Lin, W. Chen, and K.-C. Chou. iNuc-PseKNC: a sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition. *Bioinformatics*, 30(11):1522–1529, 2014.
- [116] K. Gurova. New hopes from old drugs: revisiting DNA-binding small molecules as anticancer agents. *Future oncology*, 5(10):1685–1704, 2009.
- [117] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.
- [118] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.
- [119] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.
- [120] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

- [121] G. Hampikian and T. Andersen. Absent sequences: nullomers and primes. In *Biocomputing 2007*, pages 355–366. World Scientific, 2007.
- [122] B. Haubold. Alignment-free phylogenetics and population genetics. *Briefings in bioinformatics*, 15(3):407–418, 2013.
- [123] Y. He, Z. Xiang, and H. L. Mobley. Vaxign: the first web-based vaccine design program for reverse vaccinology and applications for vaccine development. *BioMed Research International*, 2010, 2010.
- [124] J. Heled and A. J. Drummond. Bayesian inference of species trees from multilocus data. *Molecular biology and evolution*, 27(3):570–580, 2009.
- [125] S. Hellberg, M. Sjoestroem, B. Skagerberg, and S. Wold. Peptide quantitative structure-activity relationships, a multivariate approach. *Journal of medicinal chemistry*, 30(7):1126–1135, 1987.
- [126] R. Helwa and J. D. Hoheisel. Analysis of DNA–protein interactions: from nitro-cellulose filter binding assays to microarray studies. *Analytical and bioanalytical chemistry*, 398(6):2551–2561, 2010.
- [127] J. Herold, S. Kurtz, and R. Giegerich. Efficient computation of absent words in genomic sequences. *BMC bioinformatics*, 9(1):167, 2008.
- [128] K. Howe, A. Bateman, and R. Durbin. QuickTree: building huge Neighbour-Joining trees of protein sequences. *Bioinformatics*, 18(11):1546–1547, 2002.
- [129] S. Hoyer. Is sporadic Alzheimer disease the brain type of non-insulin dependent diabetes mellitus? A challenging hypothesis. *Journal of neural transmission*, 105(4):415–422, 1998.
- [130] C. Huang and J.-Q. Yuan. Predicting protein subchloroplast locations with both single and multiple sites via three different modes of Chou’s pseudo amino acid compositions. *Journal of theoretical biology*, 335:205–212, 2013.

- [131] Y. Huang, B. Niu, Y. Gao, L. Fu, and W. Li. CD-HIT Suite: a web server for clustering and comparing biological sequences. *Bioinformatics*, 26(5):680–682, 2010.
- [132] D. H. Huson, S. M. Nettles, and T. J. Warnow. Disk-covering, a fast-converging method for phylogenetic tree reconstruction. *Journal of Computational Biology*, 6(3-4):369–386, 1999.
- [133] D. H. Huson, L. Vawter, and T. J. Warnow. Solving large scale phylogenetic problems using DCM2. In *ISMB*, volume 99, page 1, 1999.
- [134] A. Ikai. Thermostability and aliphatic index of globular proteins. *The Journal of Biochemistry*, 88(6):1895–1898, 1980.
- [135] J. Im, N. Tuvshinjargal, B. Park, W. Lee, D.-S. Huang, and K. Han. PNIModeler: web server for inferring protein-binding nucleotides from sequence data. In *BMC genomics*, volume 16, page S6. BioMed Central, 2015.
- [136] M. W. Jackwood, L. Hickie, S. Kapil, R. Silva, K. Osterrieder, C. Prideaux, R. Schultz, and A. Bell. Vaccine development using recombinant DNA technology. 2008.
- [137] V. Jaiswal, S. K. Chanumolu, A. Gupta, R. S. Chauhan, and C. Rout. Jenner-predict server: prediction of protein vaccine candidates (PVCs) in bacteria based on host-pathogen interactions. *BMC bioinformatics*, 14(1):211, 2013.
- [138] J. Jia, Z. Liu, X. Xiao, B. Liu, and K.-C. Chou. iPPI-Esml: an ensemble classifier for identifying the interactions of proteins by incorporating their physicochemical properties and wavelet transforms into PseAAC. *Journal of theoretical biology*, 377:47–56, 2015.
- [139] J. Jia, Z. Liu, X. Xiao, B. Liu, and K.-C. Chou. iCar-PseCp: identify carbonylation sites in proteins by Monte Carlo sampling and incorporating sequence coupled effects into general PseAAC. *Oncotarget*, 7(23):34558, 2016.

- [140] Y.-S. Jiao and P.-F. Du. Predicting Golgi-resident protein types using pseudo amino acid compositions: Approaches with positional specific physicochemical properties. *Journal of theoretical biology*, 391:35–42, 2016.
- [141] Y.-S. Jiao and P.-F. Du. Prediction of Golgi-resident protein types using general form of Chou’s pseudo-amino acid compositions: Approaches with minimal redundancy maximal relevance feature selection. *Journal of theoretical biology*, 402:38–44, 2016.
- [142] D. Jones. Reverse vaccinology on the cusp, 2012.
- [143] D. Julong. Introduction to grey system theory. *The Journal of grey system*, 1(1):1–24, 1989.
- [144] S.-R. Jun, G. E. Sims, G. A. Wu, and S.-H. Kim. Whole-proteome phylogeny of prokaryotes by feature frequency profiles: An alignment-free method with optimal feature resolution. *Proceedings of the National Academy of Sciences*, 107(1):133–138, 2010.
- [145] R. Kaundal, R. Saini, and P. X. Zhao. Combining machine learning and homology-based approaches to accurately predict subcellular localization in Arabidopsis. *Plant physiology*, 154(1):36–54, 2010.
- [146] S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, and M. Kanehisa. AAindex: amino acid index database, progress report 2008. *Nucleic acids research*, 36(suppl.1):D202–D205, 2007.
- [147] M. Khan, M. Hayat, S. A. Khan, and N. Iqbal. Unb-DPC: Identify mycobacterial membrane protein types by incorporating un-biased dipeptide composition into Chou’s general PseAAC. *Journal of theoretical biology*, 415:13–19, 2017.
- [148] R. Kohavi and F. Provost. Confusion matrix. *Machine learning*, 30(2-3):271–274, 1998.

- [149] R. Kohavi, D. Sommerfield, and J. Dougherty. Data mining using/spl Mscr//spl Lscr//spl Cscr/++ a machine learning library in C++. In *Tools with Artificial Intelligence, 1996., Proceedings Eighth IEEE International Conference on*, pages 234–245. IEEE, 1996.
- [150] S. M. Krishnan. Using Chou’s general PseAAC to analyze the evolutionary relationship of receptor associated proteins (RAP) with various folding patterns of protein domains. *Journal of theoretical biology*, 445:62–74, 2018.
- [151] A. Krogh, B. Larsson, G. Von Heijne, and E. L. Sonnhammer. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes¹. *Journal of molecular biology*, 305(3):567–580, 2001.
- [152] L. S. Kubatko, B. C. Carstens, and L. L. Knowles. STEM: species tree estimation using maximum likelihood for gene trees under coalescence. *Bioinformatics*, 25(7):971–973, 2009.
- [153] L. S. Kubatko and J. H. Degnan. Inconsistency of phylogenetic estimates from concatenated data under coalescence. *Systematic Biology*, 56(1):17–24, 2007.
- [154] K. K. Kumar, G. Pugalenth, and P. Suganthan. DNA-Prot: identification of DNA binding proteins from protein sequence information using random forest. *Journal of Biomolecular Structure and Dynamics*, 26(6):679–686, 2009.
- [155] M. Kumar, M. M. Gromiha, and G. P. Raghava. Identification of DNA-binding proteins using support vector machines and evolutionary profiles. *BMC bioinformatics*, 8(1):463, 2007.
- [156] J. Kyte and R. F. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of molecular biology*, 157(1):105–132, 1982.
- [157] M. S. Ladinsky, D. N. Mastrorarde, J. R. McIntosh, K. E. Howell, and L. A. Staehelin. Golgi structure in three dimensions: functional insights from the normal rat kidney cell. *The Journal of cell biology*, 144(6):1135–1149, 1999.

- [158] B. R. Larget, S. K. Kotha, C. N. Dewey, and C. Ané. BUCKy: gene tree/species tree reconciliation with Bayesian concordance analysis. *Bioinformatics*, 26(22):2910–2911, 2010.
- [159] A. D. Leaché and B. Rannala. The accuracy of species tree estimation under simulation: a comparison of methods. *Systematic biology*, 60(2):126–137, 2010.
- [160] J. S. Lee, S. J. Shin, M. T. Collins, I. D. Jung, Y.-I. Jeong, C.-M. Lee, Y. K. Shin, D. Kim, and Y.-M. Park. Mycobacterium avium subsp. paratuberculosis fibronectin attachment protein activates dendritic cells and induces a Th1 polarization. *Infection and immunity*, 77(7):2979–2988, 2009.
- [161] C.-H. Leung, D. S.-H. Chan, V. P.-Y. Ma, and D.-L. Ma. DNA-Binding Small Molecules as Inhibitors of Transcription Factors. *Medicinal research reviews*, 33(4):823–846, 2013.
- [162] Z.-Y. Liang, H.-Y. Lai, H. Yang, C.-J. Zhang, H. Yang, H.-H. Wei, X.-X. Chen, Y.-W. Zhao, Z.-D. Su, W.-C. Li, et al. Pro54DB: a database for experimentally verified sigma-54 promoters. *Bioinformatics*, 33(3):467–469, 2017.
- [163] T. J. Liesegang. Varicella zoster virus vaccines: effective, but concerns linger. *Canadian Journal of Ophthalmology*, 44(4):379–384, 2009.
- [164] H. Lin, W. Chen, and H. Ding. AcalPred: a sequence-based tool for discriminating between acidic and alkaline enzymes. *PloS one*, 8(10):e75726, 2013.
- [165] H. Lin, E.-Z. Deng, H. Ding, W. Chen, and K.-C. Chou. iPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. *Nucleic acids research*, 42(21):12961–12972, 2014.
- [166] H. Lin and H. Ding. Predicting ion channels and their types by the dipeptide mode of pseudo amino acid composition. *Journal of theoretical biology*, 269(1):64–69, 2011.

- [167] H. Lin, Z.-Y. Liang, H. Tang, and W. Chen. Identifying sigma70 promoters with novel pseudo nucleotide composition. *IEEE/ACM transactions on computational biology and bioinformatics*, 2017.
- [168] W.-Z. Lin, J.-A. Fang, X. Xiao, and K.-C. Chou. iDNA-Prot: identification of DNA binding proteins using random forest with grey model. *PloS one*, 6(9):e24756, 2011.
- [169] W.-Z. Lin, J.-A. Fang, X. Xiao, and K.-C. Chou. iLoc-Animal: a multi-label learning classifier for predicting subcellular localization of animal proteins. *Molecular BioSystems*, 9(4):634–644, 2013.
- [170] C. Linder and T. Warnow. An overview of phylogeny reconstruction. In *Handbook of Computational Molecular Biology*. CRC Press, 2005.
- [171] B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, and K.-C. Chou. Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic acids research*, 43(W1):W65–W71, 2015.
- [172] B. Liu, R. Long, and K.-C. Chou. iDHS-EL: identifying DNase I hypersensitive sites by fusing three different modes of pseudo nucleotide composition into an ensemble learning framework. *Bioinformatics*, 32(16):2411–2418, 2016.
- [173] B. Liu, S. Wang, R. Long, and K.-C. Chou. iRSpot-EL: identify recombination spots with an ensemble learning approach. *Bioinformatics*, 33(1):35–41, 2016.
- [174] B. Liu, S. Wang, and X. Wang. DNA binding protein identification by combining pseudo amino acid composition and profile-based protein representation. *Scientific reports*, 5:15479, 2015.
- [175] B. Liu, H. Wu, and K.-C. Chou. Pse-in-One 2.0: an improved package of web servers for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Natural Science*, 9(04):67, 2017.

- [176] B. Liu, J. Xu, S. Fan, R. Xu, J. Zhou, and X. Wang. PseDNA-Pro: DNA-binding protein identification by combining Chous PseAAC and physicochemical distance transformation. *Molecular Informatics*, 34(1):8–17, 2015.
- [177] B. Liu, J. Xu, X. Lan, R. Xu, J. Zhou, X. Wang, and K.-C. Chou. iDNA-Prot|dis: identifying DNA-binding proteins by incorporating amino acid distance-pairs and reduced alphabet profile into the general pseudo amino acid composition. *PloS one*, 9(9):e106691, 2014.
- [178] B. Liu, F. Yang, and K.-C. Chou. 2L-piRNA: a two-layer ensemble classifier for identifying piwi-interacting RNAs and their function. *Molecular Therapy-Nucleic Acids*, 7:267–277, 2017.
- [179] B. Liu, F. Yang, D.-S. Huang, and K.-C. Chou. iPromoter-2L: a two-layer predictor for identifying promoters and their types by multi-window-based PseKNC. *Bioinformatics*, 34(1):33–40, 2017.
- [180] L. Liu. BEST: Bayesian estimation of species trees under the coalescent model. *Bioinformatics*, 24(21):2542–2543, 2008.
- [181] L. Liu, L. Yu, and S. V. Edwards. A maximum pseudo-likelihood approach for estimating species trees under the coalescent model. *BMC evolutionary biology*, 10(1):302, 2010.
- [182] L. Liu, L. Yu, D. K. Pearl, and S. V. Edwards. Estimating species phylogenies using coalescence times among sequences. *Systematic biology*, 58(5):468–477, 2009.
- [183] N. Liu and T.-m. Wang. A relative similarity measure for the similarity analysis of DNA sequences. *Chemical Physics Letters*, 408(4-6):307–311, 2005.
- [184] T. Liu, Y. Qin, Y. Wang, and C. Wang. Prediction of protein structural class based on gapped-dipeptides and a recursive feature selection approach. *International journal of molecular sciences*, 17(1):15, 2015.

- [185] T. Liu, X. Zheng, and J. Wang. Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile. *Biochimie*, 92(10):1330–1334, 2010.
- [186] W.-X. Liu, E.-Z. Deng, W. Chen, and H. Lin. Identifying the subfamilies of voltage-gated potassium channels using feature selection technique. *International journal of molecular sciences*, 15(7):12940–12951, 2014.
- [187] Z. Liu, X. Xiao, D.-J. Yu, J. Jia, W.-R. Qiu, and K.-C. Chou. pRNAm-PC: Predicting N6-methyladenosine sites in RNA sequences via physical–chemical properties. *Analytical biochemistry*, 497:60–67, 2016.
- [188] H. Lodish, D. Baltimore, A. Berk, S. Zipursky, P. Matsudaira, and J. Darnell. *Molecular Cell Biology*. Scientific American Books, New York. Technical report, ISBN 07167-2380-8, 1995.
- [189] R. J. Longley, B. R. Halbroth, A. M. Salman, K. J. Ewer, S. H. Hodgson, C. J. Janse, S. M. Khan, A. V. Hill, and A. J. Spencer. Assessment of the Plasmodium falciparum preerythrocytic antigen UIS3 as a potential candidate for a malaria vaccine. *Infection and immunity*, 85(3):e00641–16, 2017.
- [190] S. M. Loosmore, Y.-p. Yang, R. Oomen, J. M. Shortreed, D. C. Coleman, and M. H. Klein. The Haemophilus influenzae HtrA protein is a protective antigen. *Infection and immunity*, 66(3):899–906, 1998.
- [191] W. Lou, X. Wang, F. Chen, Y. Chen, B. Jiang, and H. Zhang. Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and Gaussian naive Bayes. *PLoS One*, 9(1):e86703, 2014.
- [192] W. P. Maddison. Gene trees in species trees. *Systematic biology*, 46(3):523–536, 1997.

- [193] C. N. Magnan, M. Zeller, M. A. Kayala, A. Vigil, A. Randall, P. L. Felgner, and P. Baldi. High-throughput prediction of protein antigenicity using protein microarray data. *Bioinformatics*, 26(23):2936–2943, 2010.
- [194] L. J. McGuffin, K. Bryson, and D. T. Jones. The PSIPRED protein structure prediction server. *Bioinformatics*, 16(4):404–405, 2000.
- [195] P. K. Meher, T. K. Sahu, V. Saini, and A. R. Rao. Predicting antimicrobial peptides with improved accuracy by incorporating the compositional, physico-chemical and structural features into Chous general PseAAC. *Scientific reports*, 7:42362, 2017.
- [196] J. Mei and J. Zhao. Prediction of HIV-1 and HIV-2 proteins by using Chous pseudo amino acid compositions and different classifiers. *Scientific reports*, 8(1):2359, 2018.
- [197] F. Mignosi, A. Restivo, and M. Sciortino. Forbidden factors and fragment assembly. *RAIRO-Theoretical Informatics and Applications*, 35(6):565–577, 2001.
- [198] F. Mignosi, A. Restivo, and M. Sciortino. Words and forbidden factors. *Theoretical Computer Science*, 273(1-2):99–117, 2002.
- [199] S. Mirarab, R. Reaz, M. S. Bayzid, T. Zimmermann, M. S. Swenson, and T. Warnow. ASTRAL: genome-scale coalescent-based species tree estimation. *Bioinformatics*, 30(17):i541–i548, 2014.
- [200] S. Mirarab and T. Warnow. ASTRAL-II: coalescent-based species tree estimation with many hundreds of taxa and thousands of genes. *Bioinformatics*, 31(12):i44–i52, 2015.
- [201] S. Montigiani, F. Falugi, M. Scarselli, O. Finco, R. Petracca, G. Galli, M. Mariani, R. Manetti, M. Agnusdei, R. Cevenini, et al. Genomic approach for analysis of surface proteins in *Chlamydia pneumoniae*. *Infection and immunity*, 70(1):368–379, 2002.

- [202] E. Mossel and S. Roch. Incomplete lineage sorting: consistent phylogeny estimation from multiple loci. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 7(1):166–171, 2010.
- [203] G. B. Motion, A. J. Howden, E. Huitema, and S. Jones. DNA-binding protein prediction using plant specific support vector machines: validation and application of a new genome annotation tool. *Nucleic acids research*, 43(22):e158–e158, 2015.
- [204] K. Nakai and P. Horton. PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization, 1999.
- [205] L. Nakhleh, U. Roshan, K. St. John, J. Sun, and T. Warnow. Designing fast converging phylogenetic methods. *Bioinformatics*, 17(suppl_1):S190–S198, 2001.
- [206] L. Nanni and A. Lumini. An ensemble of reduced alphabets with protein encoding based on grouped weight for predicting DNA-binding proteins. *Amino acids*, 36(2):167–175, 2009.
- [207] M. Nei. *Molecular evolutionary genetics*. Columbia university press, 1987.
- [208] S. Nelesen, K. Liu, L.-S. Wang, C. R. Linder, and T. Warnow. DACTAL: divide-and-conquer trees (almost) without alignments. *Bioinformatics*, 28(12):i274–i282, 2012.
- [209] N. Nguyen, S. Mirarab, and T. Warnow. MRL and SuperFine+MRL: new supertree methods. *Algorithms for Molecular Biology*, 7(1):3, 2012.
- [210] H. Nielsen. Predicting secretory proteins with SignalP. *Protein Function Prediction: Methods and Protocols*, pages 59–73, 2017.
- [211] H. Nielsen, J. Engelbrecht, S. Brunak, and G. von Heijne. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein engineering*, 10(1):1–6, 1997.

- [212] H. Nielsen and A. Krogh. Prediction of signal peptides and signal anchors by a hidden Markov model. In *Ismb*, volume 6, pages 122–130, 1998.
- [213] G. Nimrod, M. Schushan, A. Szilágyi, C. Leslie, and N. Ben-Tal. iDBPs: a web server for the identification of DNA binding proteins. *Bioinformatics*, 26(5):692–693, 2010.
- [214] E. Ong, M. U. Wong, and Y. He. identification of new Features from Known Bacterial Protective Vaccine antigens enhances rational Vaccine Design. *Frontiers in immunology*, 8, 2017.
- [215] R. D. Page. Genes, organisms, and areas: the problem of multiple lineages. *Systematic Biology*, 42(1):77–84, 1993.
- [216] T. Pahikkala, A. Airola, S. Pietilä, S. Shakyawar, A. Szwajda, J. Tang, and T. Aittokallio. Toward more realistic drug–target interaction predictions. *Briefings in bioinformatics*, page bbu010, 2014.
- [217] P. Pamilo and M. Nei. Relationships between gene trees and species trees. *Molecular biology and evolution*, 5(5):568–583, 1988.
- [218] A. Patronov and I. Doytchinova. T-cell epitope vaccine design by immunoinformatics. *Open biology*, 3(1):120139, 2013.
- [219] I. Paz, E. Kligun, B. Bengad, and Y. Mandel-Gutfreund. BindUP: a web server for non-homology-based prediction of DNA and RNA binding proteins. *Nucleic acids research*, 44(W1):W568–W574, 2016.
- [220] W. R. Pearson. Rapid and sensitive sequence comparison with FASTP and FASTA. 1990.
- [221] T. N. Petersen, S. Brunak, G. von Heijne, and H. Nielsen. SignalP 4.0: discriminating signal peptides from transmembrane regions. *Nature methods*, 8(10):785, 2011.

- [222] E. L. Peterson, J. Kondev, J. A. Theriot, and R. Phillips. Reduced amino acid alphabets exhibit an improved sensitivity and selectivity in fold assignment. *Bioinformatics*, 25(11):1356–1362, 2009.
- [223] A. J. Pinho, P. J. Ferreira, S. P. Garcia, and J. M. Rodrigues. On finding minimal absent words. *BMC bioinformatics*, 10(1):137, 2009.
- [224] M. Pizza, V. Scarlato, V. Massignani, M. M. Giuliani, B. Aricò, M. Comanducci, G. T. Jennings, L. Baldi, E. Bartolini, B. Capecchi, et al. Identification of vaccine candidates against serogroup B meningococcus by whole-genome sequencing. *Science*, 287(5459):1816–1820, 2000.
- [225] D. M. Powers. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. 2011.
- [226] M. N. Price, P. S. Dehal, and A. P. Arkin. FastTree: computing large minimum evolution trees with profiles instead of a distance matrix. *Molecular biology and evolution*, 26(7):1641–1650, 2009.
- [227] M. N. Price, P. S. Dehal, and A. P. Arkin. FastTree 2 – approximately maximum-likelihood trees for large alignments. *PloS one*, 5(3):e9490, 2010.
- [228] W.-R. Qiu, S.-Y. Jiang, Z.-C. Xu, X. Xiao, and K.-C. Chou. iRNAm5C-PseDNC: identifying RNA 5-methylcytosine sites by incorporating physical-chemical properties into pseudo dinucleotide composition. *Oncotarget*, 8(25):41178, 2017.
- [229] W.-R. Qiu, B.-Q. Sun, X. Xiao, D. Xu, and K.-C. Chou. iPhos-PseEvo: Identifying Human Phosphorylated Proteins by Incorporating Evolutionary Information into General PseAAC via Grey System Theory. *Molecular informatics*, 36(5-6), 2017.
- [230] W.-R. Qiu, B.-Q. Sun, X. Xiao, Z.-C. Xu, and K.-C. Chou. iPTM-mLys: identifying multiple lysine PTM sites and their different types. *Bioinformatics*, 32(20):3116–3123, 2016.

- [231] M. A. Ragan. Matrix representation in reconstructing phylogenetic relationships among the eukaryotes. *Biosystems*, 28(1-3):47–55, 1992.
- [232] A. Rana and Y. Akhter. A multi-subunit based, thermodynamically stable model vaccine using combined immunoinformatics and protein structure based approach. *Immunobiology*, 221(4):544–557, 2016.
- [233] R. Rappuoli. Reverse vaccinology. *Current opinion in microbiology*, 3(5):445–450, 2000.
- [234] R. Rappuoli and A. Aderem. A 2020 vision for vaccines against HIV, tuberculosis and malaria. *Nature*, 473(7348):463, 2011.
- [235] R. Rappuoli, M. Pizza, G. Del Giudice, and E. De Gregorio. Vaccines, new opportunities for a new society. *Proceedings of the National Academy of Sciences*, 111(34):12288–12293, 2014.
- [236] R. Reaz, M. S. Bayzid, and M. S. Rahman. Accurate phylogenetic tree reconstruction from quartets: A heuristic approach. *PLoS One*, 9(8):e104008, 2014.
- [237] A. S. Rose, A. R. Bradley, Y. Valasatava, J. M. Duarte, A. Prlić, and P. W. Rose. Web-based molecular graphics for large complexes. In *Proceedings of the 21st International Conference on Web3D Technology*, pages 185–186. ACM, 2016.
- [238] A. S. Rose and P. W. Hildebrand. NGL Viewer: a web application for molecular visualization. *Nucleic acids research*, 43(W1):W576–W579, 2015.
- [239] U. W. Roshan, T. Warnow, B. M. Moret, and T. L. Williams. Rec-I-DCM3: a fast algorithmic technique for reconstructing phylogenetic trees. In *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*, pages 98–109. IEEE, 2004.
- [240] B. C. Ross, L. Czajkowski, D. Hocking, M. Margetts, E. Webb, L. Rothel, M. Patterson, C. Agius, S. Camuglia, E. Reynolds, et al. Identification of vaccine can-

- didate antigens from a genomic analysis of *Porphyromonas gingivalis*. *Vaccine*, 19(30):4135–4142, 2001.
- [241] J. S Bernardes. A review of protein function prediction under machine learning perspective. *Recent patents on biotechnology*, 7(2):122–141, 2013.
- [242] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.
- [243] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4):406–425, 1987.
- [244] E. Sayyari and S. Mirarab. Fast coalescent-based computation of local branch support from quartet frequencies. *Molecular biology and evolution*, 33(7):1654–1668, 2016.
- [245] X. Shao, Y. Tian, L. Wu, Y. Wang, L. Jing, and N. Deng. Predicting DNA-and RNA-binding proteins from sequences with kernel methods. *Journal of Theoretical Biology*, 258(2):289–293, 2009.
- [246] J. Shi, S. Zhang, Y. Liang, and Q. Pan. Prediction of protein subcellular localizations using moment descriptors and support vector machine. In *International Workshop on Pattern Recognition in Bioinformatics*, pages 105–114. Springer, 2006.
- [247] R. M. Silva, D. Pratas, L. Castro, A. J. Pinho, and P. J. Ferreira. Three minimal sequences found in Ebola virus genomes and absent from human DNA. *Bioinformatics*, 31(15):2421–2425, 2015.
- [248] M. Simonsen, T. Mailund, and C. N. Pedersen. Rapid neighbour-joining. In *International Workshop on Algorithms in Bioinformatics*, pages 113–122. Springer, 2008.
- [249] G. E. Sims, S.-R. Jun, G. A. Wu, and S.-H. Kim. Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. *Proceedings of the National Academy of Sciences*, 106(8):2677–2682, 2009.

- [250] J. Song, Y. Wang, F. Li, T. Akutsu, and N. Rawlings. iProt-Sub: a comprehensive package for accurately mapping and predicting protease-specific substrates and cleavage sites. *Briefings in Bioinformatics*, 2018.
- [251] L. Song, D. Li, X. Zeng, Y. Wu, L. Guo, and Q. Zou. nDNA-prot: identification of DNA-binding proteins based on unbalanced classification. *BMC bioinformatics*, 15(1):298, 2014.
- [252] S. Song, L. Liu, S. V. Edwards, and S. Wu. Resolving conflict in eutherian mammal phylogeny using phylogenomics and the multispecies coalescent model. *Proceedings of the National Academy of Sciences*, 109(37):14942–14947, 2012.
- [253] R. E. Soria-Guerra, R. Nieto-Gomez, D. O. Govea-Alonso, and S. Rosales-Mendoza. An overview of bioinformatics tools for epitope prediction: implications on vaccine development. *Journal of biomedical informatics*, 53:405–414, 2015.
- [254] A. Stamatakis. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688–2690, 2006.
- [255] A. Stamatakis. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313, 2014.
- [256] E. W. Stawiski, L. M. Gregoret, and Y. Mandel-Gutfreund. Annotating nucleic acid-binding function based on protein structure. *Journal of molecular biology*, 326(4):1065–1079, 2003.
- [257] L. J. Su, P. K. Auluck, T. F. Outeiro, E. Yeger-Lotem, J. A. Kritzer, D. F. Tardiff, K. E. Strathearn, F. Liu, S. Cao, S. Hamamichi, et al. Compounds from an unbiased chemical screen reverse both ER-to-Golgi trafficking defects and mitochondrial dysfunction in Parkinsons disease models. *Disease models & mechanisms*, 3(3-4):194–208, 2010.
- [258] W.-K. Sung. *Algorithms in bioinformatics: A Practical Introduction*. CRC Press, 2011.

- [259] D. Swofford. PAUP*: phylogenetic analysis using parsimony (* and other methods). Sunderland, MA, 2003. Version 4.
- [260] A. Szabóová, O. Kuželka, F. Železný, and J. Tolar. Prediction of DNA-binding propensity of proteins by the ball-histogram method using automatic template search. In *BMC bioinformatics*, volume 13, page S3. BioMed Central, 2012.
- [261] A. Szilágyi and J. Skolnick. Efficient prediction of nucleic acid binding function from low-resolution protein structures. *Journal of molecular biology*, 358(3):922–933, 2006.
- [262] W. Szmunes, W. Oleszko, C. Stevens, and A. Goodman. PASSIVE. ACTIVE IMMUNISATION AGAINST HEPATITIS B: IMMUNOGENICITY STUDIES IN ADULT AMERICANS. *The Lancet*, 317(8220):575–577, 1981.
- [263] W. Szmunes, C. E. Stevens, E. J. Harley, E. A. Zang, P. E. Taylor, and H. J. Alter. The immune response of healthy adults to a reduced dose of hepatitis B vaccine. *Journal of medical virology*, 8(2):123–129, 1981.
- [264] F. Tajima. Evolutionary relationship of DNA sequences in finite populations. *Genetics*, 105(2):437–460, 1983.
- [265] N. Takahata. Gene genealogy in three related populations: consistency probability between gene and population trees. *Genetics*, 122(4):957–966, 1989.
- [266] H. Tang, Z.-D. Su, H.-H. Wei, W. Chen, and H. Lin. Prediction of cell-penetrating peptides with feature selection techniques. *Biochemical and biophysical research communications*, 477(1):150–154, 2016.
- [267] D. Ungar. Golgi linked protein glycosylation and associated diseases. In *Seminars in cell & developmental biology*, volume 20, pages 762–769. Elsevier, 2009.
- [268] J. M. van den Elsen, D. A. Kuntz, and D. R. Rose. Structure of Golgi α -mannosidase II: a target for inhibition of growth and metastasis of cancer cells. *The EMBO Journal*, 20(12):3008–3017, 2001.

- [269] A. D. van Dijk, D. Bosch, C. J. ter Braak, A. van der Krol, and R. C. van Ham. Predicting sub-Golgi localization of type II membrane proteins. *Bioinformatics*, 24(16):1779–1786, 2008.
- [270] G. Vernikos and D. Medini. Bexsero® chronicle. *Pathogens and global health*, 108(7):305–316, 2014.
- [271] S. Vinga and J. Almeida. Alignment-free sequence comparison – a review. *Bioinformatics*, 19(4):513–523, 2003.
- [272] S. Vivona, F. Bernante, and F. Filippini. NERVE: new enhanced reverse vaccinology environment. *BMC biotechnology*, 6(1):35, 2006.
- [273] S. Wan, M.-W. Mak, and S.-Y. Kung. HybridGO-Loc: Mining hybrid features on gene ontology for predicting subcellular localization of multi-location proteins. *PLoS One*, 9(3):e89545, 2014.
- [274] G. Wang and R. L. Dunbrack. PISCES: recent improvements to a PDB sequence culling server. *Nucleic acids research*, 33(suppl_2):W94–W98, 2005.
- [275] M. Waris, K. Ahmad, M. Kabir, and M. Hayat. Identification of DNA binding proteins using evolutionary profiles position specific scoring matrix. *Neurocomputing*, 199:154–162, 2016.
- [276] T. Warnow. *Computational phylogenetics: An introduction to designing methods for phylogeny estimation*. Cambridge University Press, 2017.
- [277] L. Wei, J. Tang, and Q. Zou. Local-DPP: An improved DNA-binding protein prediction method by exploring local evolutionary information. *Information Sciences*, 384:135–144, 2017.
- [278] WHO. MDG 6: Combat HIV/AIDS, malaria and other diseases, 2014. Geneva, WHO.

- [279] WHO, UNICEF, and World Bank. State of the Worlds Vaccines and Immunization, 2009. 3rd edition, Geneva, WHO.
- [280] T. M. Wizemann, J. H. Heinrichs, J. E. Adamou, A. L. Erwin, C. Kunsch, G. H. Choi, S. C. Barash, C. A. Rosen, H. R. Masure, E. Tuomanen, et al. Use of a whole genome approach to identify vaccine molecules affording protection against *Streptococcus pneumoniae* infection. *Infection and immunity*, 69(3):1593–1598, 2001.
- [281] S. Wold, J. Jonsson, M. Sjöström, M. Sandberg, and S. Rännar. DNA and peptide sequences and chemical processes multivariately modelled by principal component analysis and partial least-squares projections to latent structures. *Analytica Chimica Acta*, 277(2):239–253, 1993.
- [282] G. Woodrow. An overview of biotechnology as applied to vaccine development. *New generation vaccines*, 25, 1997.
- [283] Z.-D. Wu, T. Jiang, and W.-J. Su. Efficient computation of shortest absent words in a genomic sequence. *Information Processing Letters*, 110(14-15):596–601, 2010.
- [284] D. Xu and J. D. Esko. A Golgi-on-a-chip for glycan synthesis. *Nature chemical biology*, 5(9):612–613, 2009.
- [285] R. Xu, J. Zhou, B. Liu, Y. He, Q. Zou, X. Wang, and K.-C. Chou. Identification of DNA-binding proteins by incorporating evolutionary information into pseudo amino acid composition via the top-n-gram approach. *Journal of Biomolecular Structure and Dynamics*, 33(8):1720–1730, 2015.
- [286] R. Xu, J. Zhou, B. Liu, L. Yao, Y. He, Q. Zou, and X. Wang. enDNA-Prot: identification of DNA-binding proteins by applying ensemble learning. *BioMed research international*, 2014, 2014.
- [287] Y. Xu, X.-J. Shao, L.-Y. Wu, N.-Y. Deng, and K.-C. Chou. iSNO-AAPair: incorporating amino acid pairwise coupling into PseAAC for predicting cysteine S-nitrosylation sites in proteins. *PeerJ*, 1:e171, 2013.

- [288] R. Yang, C. Zhang, R. Gao, and L. Zhang. A Novel Feature Extraction Method with Feature Selection to Identify Golgi-Resident Protein Types from Imbalanced Data. *International journal of molecular sciences*, 17(2):218, 2016.
- [289] Y. Yang, R. Heffernan, K. Paliwal, J. Lyons, A. Dehzangi, A. Sharma, J. Wang, A. Sattar, and Y. Zhou. Spider2: A package to predict secondary structure, accessible surface area, and main-chain torsional angles by deep neural networks. In *Prediction of Protein Secondary Structure*, pages 55–63. Springer, 2017.
- [290] B. Yu, L. Lou, S. Li, Y. Zhang, W. Qiu, X. Wu, M. Wang, and B. Tian. Prediction of protein structural class for low-similarity sequences using Chous pseudo amino acid composition and wavelet denoising. *Journal of Molecular Graphics and Modelling*, 76:260–273, 2017.
- [291] D. Yu, X. Wu, H. Shen, J. Yang, Z. Tang, Y. Qi, and J. Yang. Enhancing membrane protein subcellular localization prediction by parallel fusion of multi-view features. *IEEE transactions on nanobioscience*, 11(4):375–385, 2012.
- [292] C. Zhang, M. Rabiee, E. Sayyari, and S. Mirarab. ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees. *BMC bioinformatics*, 19(6):153, 2018.
- [293] T. Zhang, P. Tan, L. Wang, N. Jin, Y. Li, L. Zhang, H. Yang, Z. Hu, L. Zhang, C. Hu, et al. RNALocate: a resource for RNA subcellular localizations. *Nucleic acids research*, 45(D1):D135–D138, 2017.
- [294] H. Zhao, Y. Yang, and Y. Zhou. Structure-based prediction of DNA-binding proteins by structural alignment and a volume-fraction corrected DFIRE-based energy function. *Bioinformatics*, 26(15):1857–1863, 2010.
- [295] J. Zhou, Q. Lu, R. Xu, L. Gui, and H. Wang. CNNsite: Prediction of DNA-binding residues in proteins using Convolutional Neural Network with sequence features. In *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*, pages 78–85. IEEE, 2016.

- [296] W. Zhou and H. Yan. Prediction of DNA-binding protein based on statistical and geometric features and support vector machines. In *Proteome science*, volume 9, page S1. BioMed Central, 2011.
- [297] C. Zou, J. Gong, and H. Li. An improved sequence based prediction protocol for DNA-binding proteins using SVM and comprehensive feature analysis. *BMC bioinformatics*, 14(1):90, 2013.

Index

- K*-nearest neighbor (KNN), 72
- n*-Gapped-Dipeptide (nGDip), 3, 22, 23, 48, 75, 106, 165
- n*-gram, 3, 48, 76, 106, 165
- 10-fold cross-validation, 25, 87, 115, 117
- AAIndex, 4
- Absent word, 9, 131
 - Minimal absent word (MAW), 10, 131, 132
 - Relative absent word (RAW), 10, 131, 132
- Accuracy, 25, 79, 87, 110
- Amino acid, 1, 14, 15, 23, 69
 - Amino acid composition (AAC), 3, 22, 23, 44, 47, 71, 103
 - Amino acid grouping, 43
 - Amino acid physico-chemical properties, 4
 - Non-standard amino acid, 14
 - Pseudo amino acid composition (PseAAC), 3, 21
 - Split amino acid composition (SAAC), 49, 72
 - Standard amino acid, 14
- Antigen, 97, 98
 - Cross-reactive antigen, 98
 - Protective antigen, 1, 8, 97, 98, 163
 - Serodiagnostic antigen, 98
- Antigenic, 8, 97, 104, 164
- ANTIGENpro, 102
- Artificial neural network (ANN), 4, 69
- ASTRAL, 36, 149, 151
- auPR, 25, 110
- auROC, 25, 79, 87, 110
- Bayesian Markov Chain Monte Carlo (MCMC), 6, 133
- Benchmark dataset, 16, 46, 74, 104
- Bipartition, 30
- BLAST, 2, 50, 92, 100
- BLASTCLUST, 92, 168
- Branch length, 31
- CD-HIT, 168
- Chou's general PseAAC, 21, 47, 72, 106
- Clade, 30
- Classification, 19, 42
- Coalescent-based method, 1, 11, 149
- Concatenation, 6, 36
- DACTAL, 151
- DCM2-QFM, 155
- DCM5-QFM, 155
- Deep coalescence, 6, 149, *see also* Incomplete lineage sorting (ILS)

Dipeptide (Dip), 3, 22, 23, 48, 72, 75, 106
 Disk-covering methods (DCMs), 7, 9, 149, 151
 Distance matrix, 6, 9
 Distance-based method, 6
 DPP-PseAAC, 7, 68, 164
 Embedded method, 21
 Enrichment, 123
 False negative (FN) rate, 37
 False positive (FP) rate, 37
 FASTA, 2, 100
 FastTree, 133
 Feature selection, 21, 51, 107
 Filter method, 21
 GC content, 135, 137
 Gene duplication and extinction, 5, 34, 150
 Gene tree, 5, 132, 149
 Gene tree discordance, 6, 31
 Gene tree estimation method, 1, 6, 9, 139
 Gene tree parsimony (GTP), 35
 Golgi apparatus (GA), 41, 163
 Hidden Markov model (HMM), 101
 Horizontal gene transfer (HGT), 5, 33, 150
 Incomplete lineage sorting (ILS), 5, 6, 33, 149, 150
 Independent test, 25, 88, 121
 isGPT, 7, 42, 164
 Jaccard distance, 132, 135, 137
 Jackknife cross-validation, 24, 87, 120
 Kernel function, 20
 Linear kernel, 20, 67, 164
 Polynomial kernel, 20
 RBF kernel, 20, 71
 Leave one protein set out cross-validation, 119
 Length weighted index (LWI), 132, 135, 136
 Logistic regression (LR), 69
 Machine learning (ML), 2, 18, 163
 Matthew's correlation coefficient (MCC), 25, 79, 87, 110
 Maximum margin classifier, 19
 Maximum-Likelihood (ML), 6, 133
 Maximum-Parsimony (MP), 6, 133
 Mean decrease in accuracy, 20, 108
 Minimal absent word (MAW), 135
 Minimizing deep coalescence (MDC), 36
 Minimizing duplication and loss (MGDL), 36
 MP-EST, 36
 MRL, 36
 MRP, 36
 Multi-species coalescent (MSC) model, 9, 149
 Neighbor joining (NJ), 133, 139, 144
 Phylogeny, 5, 9, 13, 131
 Phylogeny reconstruction, 1, 5, 28, 150, 167
 PISCES, 168
 Polytoomy, 30
 Position specific n -gram (PSN), 24, 48, 76, 106, 165

Position specific scoring matrix, 4, 44, 70
 Prediction algorithm, 17, 50, 77, 107
 Predictor availability, 17, 56, 80, 111
 Predictor evaluation, 17, 55, 79, 110
 propy, 22
 Protein, 1, 13, 14, 41, 97, 98
 cis-Golgi protein, 2, 7, 41, 163
 trans-Golgi protein, 2, 7, 41, 163
 DNA-binding protein (DNA-BP), 1, 67, 68, 163
 sub-Golgi protein, 1, 7, 42, 163
 Protein attribute prediction, 14, 163
 Protein sample representation, 17, 47, 75, 106
 Protein structure
 Primary structure, 14
 Secondary structure, 14
 α -helix, 15
 β -sheets, 15
 Tertiary structure, 15
 Pse-in-One, 22
 PseAAC-General, 22
 PseKNC, 22
 PSI-BLAST, 4
 PSORT, 2, 100
 PSSM, 4, 64, 70, 72, *see also* Position specific scoring matrix
 Quartet FM (QFM), 9, 149, 150
 Quartet support, 37
 Random forests, 4, 7, 20, 41, 67, 97, 104, 163
 Random undersampling, 111
 RAxML, 133
 Recursive feature elimination (RFE), 7, 67, 104, 164
 Regression, 19, 42
 Relative absent word (RAW), 138
 Reverse complement (RC), 132
 Reverse vaccinology (RV), 97, 100
 Robinson-Foulds (RF) rate, 37
 Sensitivity, 25, 79, 87, 110
 SignalP, 101
 SMOTE, 7, 42, 50, 163
 Species tree, 5, 149
 Species tree estimation method, 1, 9, 36
 Specificity, 25, 79, 87, 110
 SPIDER2, 3, 73
 Statistical consistency, 9, 35
 Summary method, 6, 36, 150
 Supervised learning, 18
 Support vector, 19
 Support vector machine (SVM), 4, 7, 19, 42, 50, 67, 163
 Taxa, 5, 28, *see also* Taxon
 Taxon, 28
 Total variation distance (TVD), 135, 137
 Training data, 18
 Tree, 28
 Binary tree, 30
 Non-binary tree, 30

- Phylogenetic tree, 28
 - Rooted tree, 29
 - Unrooted tree, 30
- Tripeptide, 22, 23, 48, 75, 106
- Unsupervised learning, 18
- UPGMA, 139, 144
- Vaccine, 98
 - Multiepitopic vaccine, 99
 - Subunit vaccine, 99
- VaxiJen, 101
- Wrapper method, 21

Publications

The research conducted as part of this thesis has resulted in the following publications.

1. Rahman, M. S., Alatabbi, A., Athar, T., Crochemore, M., & Rahman, M. S. (2016). *Absent words and the (dis) similarity analysis of DNA sequences: an experimental study*. BMC research notes, 9(1), 186.
2. Rahman, M. S., Rahman, M. K., Kaykobad, M., & Rahman, M. S. (2018). *isGPT: An optimized model to identify sub-Golgi protein types using SVM and Random Forest based feature selection*. Artificial Intelligence in Medicine 84 (2018) 90–100.
3. Rahman, M. S., Shatabda, S., Saha, S., Kaykobad, M., & Rahman, M. S. (2018). *DPP-PseAAC: A DNA-binding protein prediction model using Chou's general PseAAC*. Journal of theoretical biology, 452, 22–34.
4. Rahman, M. S., Rahman, M. K., Saha, S., Kaykobad, M., & Rahman, M. S. *Antigenic: An improved prediction model of protective antigens*. (Under review) Artificial Intelligence in Medicine