SPECTRAL COHERENCE BASED APPROACH FOR IN-SILICO ENZYME CLASSIFICATION

By -

Payal Bhargava - 081512

Aahut Chandwani - 081506

Under the guidance of -

Dr. P. K. Naik

Thesis submitted in partial fulfillment of the

Degree of Bachelor of Technology

in

BIOINFORMATICS

MAY-2012

Department of Biotechnology & Bioinformatics

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TABLE OF CONTENTS

CERTIFICATE iii
ACKNOWLEDGEMENT iv
SUMMARY v
LIST OF TABLES vi
1. CHAPTER 1 – INTRODUCTION 1-14
1.1. Enzymes
1.1.1. About Enzymes
1.1.2 Physico-Chemical properties of enzymes
1.1.3. Importance of EC number
1.1.4. Scheme for the classification of enzymes and the generation of EC numbers
1.1.5. Need of prediction and classification of enzymes
1.2 Spectral Coherence
1.3 Current scenario of Insilico enzyme classification
1.4 Objective

2. CHAPTER 2 – TOOLS AND TECHNIQUES 15-24
2.1. Model Development
2.1.1 Statistica 9.0
2.1.2 Multiple Linear Regression- Statistica
2.1.3 Minitab
2.2. Calculation of Physico- Chemical properties of novel sequence and model validation
2.2.1 Active Perl (version 5)
2.3 Calculation of mean squared coherence values
2.3.1 Mean Squared Coherence
2.3.2 Integration of scales
2.3.2.1 Scalar product
2.3.2.2 Eigenvalues and Eigenvectors
2.3.2.3 Feature Extraction
2.3.2.4 Cross-Correlation Function
2.3.2.5 Auto-Correlation Function
2.3.2.6 Discrete-Fourier Transform
2.3.2.7 Mean Squared Coherence
2.4 Matlab
2.5 Development of classification model using weighted random forests in statistica
2.6 Web Interface Development

2.6.1 HTML / CSS

2.7. Validation Method

2.7.1. Accuracy Calculation

3. CHAPTER – 3 – METHODOLOGY **25-28**

3.1 Protocol for multiple linear regression model

3.2 Protocol for model development using mean squared coherence and weighted random forests.

RESULTS AND CONCLUSION **29-31**

Work action plan

Results

Conclusion

Appendix A: Perl Scripts **32-54**

Appendix B: Tables **55-56**

Appendix C: Matlab programs **57**

Appendix D: Visual basic code **58-63**

REFERENCES **64-65**

BRIEF PROFILE OF STUDENTS **66**

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

juit

विद्या तत्व ज्योतिसम

CERTIFICATE

This is to certify that the thesis entitled **"SPECTRAL COHERENCE BASED APPROACH FOR IN-SILICO ENZYME CLASSIFICATION "** submitted by **Payal Bhargava(081512)** and **Aahut Chandwani(081506)** to the Department of Biotechnology & Bioinformatics, Jaypee University of Information Technology, Waknaghat in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Bioinformatics** is a record of bona fide research work carried out by them under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Mr. P.K. Naik (Ph. D)
Associate Professor
Dept.of Biotechnology & Bioinformatics
Jaypee University of
Information Technology
Waknaghat-173234
Solan, Himachal Pradesh

Date: 26/5/2012

ACKNOWLEDGEMENT

This project would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost, we express our utmost gratitude to **Dr. P.K. Naik,** Associate Professor at Department of Biotechnology and Bioinformatics (JUIT), for his unfailing support as our project adviser, and **Mr. Dipankar Sengupta,** Associate lecturer at Department of Biotechnology and Bioinformatics (JUIT), as our project co-adviser for his patience and steadfast encouragement. They have been an inspiration as we hurdled through all the obstacles in this research work.
Working under both of them is an enriching experience.

Dr. R. S. Chauhan, Head of the Department of Biotechnology and Bioinformatics (JUIT), who had kind concern and consideration regarding our academic requirements. We thank **Mrs. Somlata Sharma** (Bioinformatics Laboratory In-charge) for providing her full cooperation with keen interest.

We would also like to thank **Mrs Shruti Jain**(Senior Lecturer, department of Electronics and Communication) for her extended support and guidance.

Our special thanks to Ms Pranshu Saxena, Ms Neha Sood, Ms Seneha Santoshi (Phd scholar), Ms Charu Suri(Phd scholar) for their guidance, encouragement and support.

Apart from these, countless events, countless people and several incidents have made a contribution to this project that is indescribable. We again express our gratitude to them. We are indebted to all those who provided reviews and suggestions for improving the results and topics covered in our project, and extend apologies to any one whom we have failed to recognize in this effort of ours. All copyrights and trademarks that are cited in this document remain the property of their respective owners.

Payal Bhargava

Payal Bhargava (081512)

Date: .26..|.5.|.. 2012

Aahut Chandwani(081506)

Date: 26|5|.2012

SUMMARY

With the advent of genomic technologies and High Throughput computing, there is an exponential growth of sequence data at both genomic and proteomic level and characterizing these sequences with traditional experimental methods remains a challenge if one considers the growth of data. When we are talking about proteins there is a specialized group of them which are performing very important activity of catalyzing various biochemical reactions. Almost 99% of processes in our body take help of enzymes. With discovery of every new protein and its sequencing completed, it's also expected to find out whether it has any catalytic activity associated with it or not. Thus it would be highly beneficial to explore *in-silico* means to classify newly found sequences into enzymatic and non enzymatic and if enzymatic then into their respective enzyme classes to gain an insight into its catalytic mechanism and their biological function.

We had an Enzyme Data Warehouse called EnHouse which contains a total of 2,02,000 enzyme sequences along with its various physiochemical properties. It also consists of information of 60,000 non enzyme sequences. Using the predicted properties of this enzymatic & non-enzymatic protein, data mining approach has been applied to develop a model which would predict whether a given protein sequence is an enzyme or not based on the predicted properties of the sequence. The data was divided into training and test data at 70:30 ratios and a model was developed by applying multiple regression technique using the training data. Validation of the model was done using 30% of the test data, which showed R^2 value of 0.82 and accuracy of 72% indicating the quality of the model developed.

Furthermore, we have translated the protein sequence into a set of 62 parameters utilizing hydrophobicity and hydrophilicity parameters of amino acids as well as by applying spectral coherence technique. These parameters were used to classify a protein sequence into different subclasses of enzymes. For an example we have apply this technique for the classification of enzyme sequences belonging to class 6 into different subclasses. Thus, we intend to develop a classification tool **EnClass** for enzymes in which when a user enters a sequence it would predict whether sequence is an enzyme or a non-enzyme, if enzyme then it belongs to which class and subclass.

LIST OF TABLES

Table No.	Title	Page No.
Table 1.	Classification of oxidoreductases into 21 subclasses	5
Table 2.	Classification of Transferases into 8 sub-classes.	7
Table 3.	Subclasses of Hydrolases enzymes and their function.	9
Table 4.	Subclasses of Lyases and their defined function.	10
Table 5.	Classification of Isomerases into 6 sub classes.	11
Table 6.	Classification of Ligases into 5 subclasses.	12
Table 7.	Accuracy of class 6.	30
Table 8.	Accuracy of class 4.	30

CHAPTER – 1
INTRODUCTION

1.1. Enzymes

1.1.1. About Enzymes

Enzymes are proteins that catalyze (*i.e.*, increase the rates of) biochemical reactions. In enzymatic reactions, the molecules at the beginning of the process are called substrates, and they are converted into molecules, called as products. Almost all processes in a biological cell need enzymes to occur at significant rates.

Most enzymes are much larger than the substrates they act on, and only a very small portion of enzyme (only 3-4 amino acids) is directly involved in catalysis. The region that contains these catalytic residues, binds the substrate, and then carries out the reaction is known as the active site. Enzymes can also contain sites that bind cofactors, which are needed for catalysis. Some enzymes also have binding sites for small molecules, which are often direct or indirect products or substrates of the reaction catalyzed.

1.1.2 Physico-Chemical properties of enzymes

Like all proteins, enzymes are made as long, linear chains of amino acids that fold to produce a three-dimensional product. Each unique amino acid sequence produces a unique structure, which has unique properties.

These properties are essential for us to classify the enzymes and on the basis of which a model could be designed for prediction of enzymatic class for a given protein.some of the properties are:

- **Length of sequence:**

The number of amino acids present in a sequence determines the length of the sequence.

- **Hydrophobicity:**

hydrophobicity is the physical property of a molecule (known as a hydrophobe) that is repelled from a mass of water. It is the prime descriptor being used for classification system. Amphiphilic protein structures are characterized by a structural segregation of hydrophobic and hydrophilic amino acid residues and therefore, to describe or to quantify the amphiphilicity of a given structure requires knowledge of the relative hydrophobicity of each residue contributing to that structure.

- **Molar percentage of amino acids:**

This is the number of specific amino acid residues per 100 residues in a protein

1.1.3. Importance of EC number

Traditionally the enzymes are classified into six major classes based on their EC Number. The **Enzyme Commission number (EC number)** is a numerical classification scheme for enzymes, based on the chemical reactions they catalyze. As a system of **enzyme nomenclature,** every EC number is associated with a recommended name for the respective enzyme [ExPASy].Every enzyme code consists of the letters "EC" followed by four numbers separated by periods. Those numbers represent a progressively finer classification of the enzyme.

For example, the tripeptide aminopeptidases have the code "EC 3.4.11.4", whose components indicate the following groups of enzymes:

- *EC 3* enzymes are hydrolases (enzymes that use water to break up some other molecule)
- *EC 3.4* are hydrolases that act on peptide bonds
- *EC 3.4.11* are those hydrolases that cleave off the amino-terminal amino acid from a polypeptide
- *EC 3.4.11.4* are those that cleave off the amino-terminal end from a tripeptide

Top Level EC numbers

Class	**Reaction catalyzed**	**Typical reaction**	**Enzyme example(s) with trivial name**
EC 1 ***Oxidoreductases***	To catalyze oxidation/reduction reactions; transfer of H and O atoms or electrons from one substance to another	AH + B → A + BH **(reduced)** A + O → AO **(oxidized)**	Dehydrogenase, oxidase
EC 2 ***Transferases***	Transfer of a functional group from one substance to another. The group may be methyl-, acyl-, amino- or phosphate group	AB + C → A + BC	Transaminase, kinase
EC 3 ***Hydrolases***	Formation of two products from a substrate by hydrolysis	AB + H_2O → AOH + BH	Lipase, amylase, peptidase
EC 4 ***Lyases***	Non-hydrolytic addition or removal of groups from substrates. C-C, C-N, C-O or C-S bonds may be cleaved	RCOCOOH → RCOH + CO_2	
EC 5 ***Isomerases***	Intramolecule rearrangement, i.e. isomerization changes within a single molecule	AB → BA	Isomerase, mutase
EC 6 ***Ligases***	Join together two molecules by synthesis of new C-O, C-S, C-N	X + Y+ ATP → XY + ADP + Pi	Synthetase

Basically all the enzymes available in the nature are classified into six major classes based on their mechanism of action. They are discussed as:

Class 1. Oxidoreductases.

To this class belong all enzymes catalysing oxidoreduction reactions. The substrate that is oxidized is regarded as hydrogen donor. The systematic name is based on donor:acceptor oxidoreductase. The common name will be dehydrogenase, wherever this is possible; as an alternative, reductase can be used. Oxidase is only used in cases where O_2 is the acceptor. The second figure in the EC-Number of the oxidoreductases, is 11, 13, 14 or 15, indicates the group in the hydrogen (or electron) donor that undergoes oxidation: 1 denotes a -CHOH- group, 2 a -CHO or -CO-COOH group or carbon monoxide, and so on, as listed in the key. The third figure, except in subclasses EC 1.11, EC 1.13, EC 1.14 and EC 1.15, indicates the type of acceptor involved: 1 denotes NAD(P)+, 2 a cytochrome, 3 molecular oxygen, 4 a disulfide, 5 a quinone or similar compound, 6 a nitrogenous group, 7 an iron-sulfur protein and 8 a flavin. In subclasses EC 1.13 and EC 1.14 a different classification scheme is used and subclasses are numbered from 11 onwards. It should be noted that in reactions with a nicotinamide coenzyme this is always regarded as acceptor, even if this direction of the reaction is not readily demonstrated. The only exception is the subclass EC 1.6, in which NAD(P)H is the donor; some other redox catalyst is the acceptor. Although not used as a criterion for classification, the two hydrogen atoms at carbon-4 of the dihydropyridine ring of nicotinamide nucleotides are not equivalent in that the hydrogen is transferred stereospecifically. The class oxidoreductases is further classified into 21 sub-classes as mentioned in the Table 1 based on more specilised mechanism.

Table 1. Classification of oxidoreductases into 21 subclasses.

Enzyme Classification number (E.C)	FUNCTION
EC 1.1	Acting on the CH-OH group of donors.
EC 1.2	Acting on the aldehyde or oxo group of donors.
EC 1.3	Acting on the CH-CH group of donors.
EC 1.4	Acting on the CH-NH(2) group of donors.
EC 1.5	Acting on the CH-NH group of donors.
EC1.6	Acting on Nadh or nadph.
EC 1.7	Acting on other nitrogenous compounds as donors.
EC 1.8	Acting on a sulfur group of donors.
EC 1.9	Acting on a heme group of donors.
EC 1.10	Acting on diphenols and related substances as donors.
EC 1.11	Acting on a peroxide as acceptor.
EC 1.12	Acting on hydrogen as donor.
EC 1.13	Acting on single donors with incorporation of molecular oxygen.
EC 1.14	Acting on paired donors, with incorporation or reduction of.
EC 1.15	Acting on superoxide as acceptor.
EC 1.16	Oxidizing metal ions.
EC 1.17	Acting on Ch or CH(2) groups.
EC 1.18	Acting on iron-sulfur proteins as donors.
EC 1.20	Acting on phosphorus or arsenic in donors.
EC 1.21	Acting on x-H and y-H to form an x-y bond.
EC 1.97	Other oxidoreductases.

Class 2. Transferases.

Transferases are enzymes transferring a group, e.g. a methyl group or a glycosyl group, from one compound (generally regarded as donor) to another compound (generally regarded as acceptor). The systematic names are formed according to the scheme donor: acceptor group transferase. The common names are normally formed according to acceptor grouptransferase or donor grouptransferase. In many cases, the donor is a cofactor (coenzyme) charged with the group to be transferred. A special case is that of the transaminases. Some transferase reactions can be viewed in different ways. For example, the enzyme-catalysed reaction :X-Y + Z = X + Z-Y; may be regarded either as a transfer of the group Y from X to Z, or as a breaking of the X-Y bond by the introduction of Z. Where Z represents phosphate or arsenate, the process is often spoken of as 'phosphorolysis' or 'arsenolysis', respectively, and a number of enzyme names based on the pattern of phosphorylase have come into use. These names are not suitable for a systematic nomenclature, because there is no reason to single out these particular enzymes from the other transferases, and it is better to regard them simply as Y-transferases. In the above reaction, the group transferred is usually exchanged, at least formally, for hydrogen, so that the equation could more strictly be written as:

X-Y + Z-H = X-H + Z-Y. Another problem is posed in enzyme-catalysed transaminations, where the -NH2 group and -H are transferred to a compound containing a carbonyl group in exchange for the = O of that group, according to the general equation:

R1-CH(-NH2)-R2 + R3-CO-R4 R1-CO-R2 + R3-CH(-NH2)-R4.

The reaction can be considered formally as oxidative deamination of the donor (e.g. amino acid) linked with reductive amination of the acceptor (e.g. oxo acid), and the transaminating enzymes (pyridoxal-phosphate proteins) might be classified as oxidoreductases. However, the unique distinctive feature of the reaction is the transfer of the amino group (by a well-established mechanism involving covalent substrate-coenzyme intermediates), which justified allocation of these enzymes among the transferases as a special subclass (EC 2.6.1, transaminases). The second figure in the code number of transferases indicates the group transferred; a one-carbon group in EC 2.1, an aldehydic or ketonic group in EC 2.2, an acyl group in EC 2.3 and so on. The third figure gives further information on the group transferred; e.g. subclass EC 2.1 is subdivided into methyltransferases (EC 2.1.1), hydroxymethyl- and formyltransferases (EC 2.1.2) and so on; only in subclass EC 2.7, does the third figure indicate the nature of the acceptor group. The class transferases is further classified into 8 sub-classes as mention in Table 2.

Table 2. Classification of Transferases into 8 sub-classes.

Enzyme Classification Number (E.C)	FUNCTION
EC 2.1	Transferring one-carbon groups.
EC 2.2	Transferring aldehyde or ketone residues.
EC 2.3	Acyltransferases.
EC 2.4	Glycosyltransferases.
EC 2.5	Transferring alkyl or aryl groups, other than methyl groups.
EC 2.6	Transferring nitrogenous groups.
EC 2.7	Transferring phosphorous-containing groups.
EC 2.8	Transferring sulfur-containing groups.

Class 3. Hydrolases.

These enzymes catalyse the hydrolytic cleavage of C-O, C-N, C-C and some other bonds, including phosphoric anhydride bonds. Although the systematic name always includes hydrolase, the common name is, in many cases, formed by the name of the substrate with the suffix -ase. It is understood that the name of the substrate with this suffix means a hydrolytic enzyme. A number of hydrolases acting on ester, glycosyl, peptide, amide or other bonds are known to catalyse not only hydrolytic removal of a particular group from their substrates, but likewise the transfer of this group to suitable acceptor molecules. In principle, all hydrolytic enzymes might be classified as transferases, since hydrolysis itself can be regarded as transfer of a specific group to water as the acceptor. Yet, in most cases, the reaction with water as the acceptor was discovered earlier and is considered as the main physiological function of the enzyme. This is why such enzymes are classified as hydrolases rather than as transferases. Some hydrolases (especially some of the esterases and glycosidases) pose problems because they have a very wide specificity and it is not easy to decide if two preparations described by different authors (perhaps from different sources) have the same catalytic properties, or if they should be listed under separate entries. An example is vitamin A esterase (formerly EC 3.1.1.12, now believed to be identical with EC 3.1.1.1). To some extent the choice must be arbitrary; however, separate entries should be given only when the specificities are sufficiently different. Another problem is that proteinases have 'esterolytic' action; they usually hydrolyse ester bonds in appropriate substrates even more rapidly than natural peptide bonds. In this case, classification among the peptide hydrolases is based on historical priority and presumed physiological function. The second figure in the code number of the hydrolases indicates the nature of the bond hydrolysed; EC 3.1 are the esterases; EC 3.2 the glycosylases, and so on. The third figure normally specifies the nature of the substrate, e.g. in the esterases the carboxylic ester hydrolases (EC 3.1.1), thiolester hydrolases (EC 3.1.2), phosphoric monoester hydrolases (EC 3.1.3); in the glycosylases the O-glycosidases (EC 3.2.1), N-glycosylases (EC 3.2.2), etc. Exceptionally, in the case of the peptidyl-peptide hydrolases the third figure is based on the catalytic mechanism as shown by active centre studies or the effect of pH. The class Hydrolases further classified into 10 subclasses (Table 3).

Table 3. Subclasses of Hydrolases enzymes and their function.

Enzyme Classification Number (E.C)	FUNCTION
3.1	Acting on ester bonds.
3.2	Glycosylases.
3.3	Acting on ether bonds.
3.4	Acting on peptide bonds (peptide hydrolases).
3.5	Acting on carbon-nitrogen bonds, other than peptide bonds.
3.6	Acting on acid anhydrides.
3.7	Acting on carbon-carbon bonds.
3.8	Acting on halide bonds.
3.11	Acting on carbon-phosphorus bonds.
3.13	Acting on carbon-sulfur bonds.

Class 4. Lyases.

Lyases are enzymes cleaving C-C, C-O, C-N, and other bonds by elimination, leaving double bonds or rings, or conversely adding groups to double bonds. The systematic name is formed according to the pattern substrate group-lyase. The hyphen is an important part of the name, and to avoid confusion should not be omitted, e.g. hydro-lyase not 'hydrolyase'. In the common names, expressions like decarboxylase, aldolase, dehydratase (in case of elimination of CO2, aldehyde, or water) are used. In cases where the reverse reaction is much more important, or the only one demonstrated, synthase (not synthetase) may be used in the name. Various subclasses of the lyases include pyridoxal-phosphate enzymes that catalyse the elimination of a b- or g-substituent from an a-amino acid followed by a replacement of this substituent by some other group. In the overall replacement reaction, no unsaturated end-product is formed; therefore, these enzymes might formally be classified as alkyl-transferases (EC 2.5.1...). However, there is ample evidence that the replacement is a two-step reaction involving the transient formation of enzyme-bound a,b(or b,g)-unsaturated amino acids. According to the rule that the first reaction is indicative for classification, these enzymes are correctly classified as lyases. Examples are tryptophan synthase (EC 4.2.1.20) and cystathionine b-synthase (EC 4.2.1.22). The second figure in the code number indicates the bond broken: EC 4.1 are carbon-carbon lyases, EC 4.2 carbon-oxygen lyases and so on. The third figure gives further information on the group eliminated (e.g. CO2 in EC 4.1.1, H2O in EC 4.2.1). The class Lyases is further categorised into 6 subclasses based on their specific function (Table 4).

Table 4. Subclasses of Lyases and their defined function.

Enzyme Classification Number (E.C)	Function
4.1	Carbon-carbon lyases
4.2	Carbon-oxygen lyases.
4.3	Carbon-nitrogen lyases.
4.4	Carbon-sulfur lyases.
4.6	Phosphorus-oxygen lyases.
4.99	Other lyases.

Class 5. Isomerases.

These enzymes catalyse geometric or structural changes within one molecule. According to the type of isomerism, they may be called racemases, epimerases, cis-trans-isomerases, isomerases, tautomerases, mutases or cycloisomerases. In some cases, the interconversion in the substrate is brought about by an intramolecular oxidoreduction (EC 5.3); since hydrogen donor and acceptor are the same molecule, and no oxidized product appears, they are not classified as oxidoreductases, even though they may contain firmly bound NAD(P)+. The subclasses are formed according to the type of isomerism, the sub-subclasses to the type of substrates. The Isomerases are further categorised into 6 sub-classes (Table 5).

Table 5. Classification of Isomerases into 6 sub classes.

Enzyme Classification Number(E.C)	FUNCTION
5.1	Racemases and epimerases.
5.2	Cis-trans-isomerases.
5.3	Intramolecular oxidoreductases.
5.4	Intramolecular transferases (mutases).
5.5	Intramolecular lyases.
5.99	Other isomerases.

Class 6. Ligases.

Ligases are enzymes catalysing the joining together of two molecules coupled with the hydrolysis of a diphosphate bond in ATP or a similar triphosphate. The systematic names are formed on the system X:Y ligase (ADP-forming). In earlier editions of the list the term synthetase has been used for the common names. Many authors have been confused by the use of the terms synthetase (used only for Group 6) and synthase (used throughout the list when it is desired to emphasis the synthetic nature of the reaction). Consequently NC-IUB decided in 1983 to abandon the use of synthetase for common names, and to replace them with names of the type X-Y ligase. In a few cases in Group 6, where the reaction is more complex or there is a common name for the product, a synthase name is used (e.g. EC 6.3.2.11 and EC 6.3.5.1). It is recommended that if the term synthetase is used by authors, it should continue to be restricted to the ligase group. The second figure in the code number indicates the bond formed: EC 6.1 for C-O bonds (enzymes acylating tRNA), EC 6.2 for C-S bonds (acyl-CoA derivatives), etc. Sub-subclasses are only in use in the C-N ligases. The enzyme Ligases are further classified into 5 subclasses.

Table 6. Classification of Ligases into 5 subclasses.

Enzyme Classification Number(E.C)	**Function**
6.1	Forming carbon-oxygen bonds.
6.2	Forming carbon-sulfur bonds.
6.3	Forming carbon-nitrogen bonds
6.4	Forming carbon-carbon bonds.
6.5	Forming phosphoric ester bonds.

1.1.5. Need for prediction and classification of enzymes:

Enzymes are substances that occur naturally in all living things, including the human body. If it's an animal or a plant, it has enzymes. Enzymes are critical for life. At present, researchers have identified more than 3,000 different enzymes in the human body. Every second of our lives these enzymes are constantly changing and renewing, sometimes at an unbelievable rate. Our body's ability to function, to repair when injured, and to ward off disease is directly related to the strength and numbers of our enzymes. That's why an enzyme deficiency can be so devastating. All life processes consist of a complex series of chemical reactions.

Using the protein engineering techniques, new enzymes are been created, ranging from food enzymes to the enzymes used for curing diseases. The large international genome sequence projects are gaining a great amount of public attention and huge sequence data bases are created it becomes more and more obvious that we are very limited in our ability to access functional data for the gene products - the proteins, in particular for enzymes. It seems quite improbable to experimentally determine function and structure of each candidate protein. So a revolutionary method is needed to solve this computation catastrophe. Primary sequence of these proteins are readily available, therefore a method using the sequence derived features will prove a much valuable and a cost effective process of determining and classifying these proteins into broader enzyme/non-enzyme and specifically into 6 major classes as defined by international enzyme commission.

1.2 Spectral Coherence:

The extraction of sequence order information is not simple as protein sequence data suffers from the issue of unequal cardinality (i.e. sequences are of different lengths). In our study we purpose a spectral coherence approach to extract sequence-order based on the amphiphilic and amphipathic characteristics of residues that are vital in the functional characterization of sequences. Coherence is the property of vectors that quantifies the degree interference. By interference we imply that if at least two random events are combined and depending on the relative phase between them, they can add constructively or subtract destructively.[7]

The major issues related to a protein are:

- the presence of an unbalanced number of proteins in different subclasses
- unequal length of the proteins

To solve these issues consistent cardinality of feature vector for such proteins should be discovered, which will enable uniformity in feature treatment by the classification schema.

1.3. Current Scenario of Insilico enzyme classification

Currently there is no enzyme classification tool existing independently with respect to all the enzymes that quickly classifies a protein into a particular enzyme class (if enzymatic nature is present). With respect to insilico classification of enzymes, previous studies have method being used which comprises of the following components - Integration of all the known hydrophobicity scales from the AAIndex ; then performing feature extraction using spectral coherence and carry out a supervised classification as validation for the goodness of using spectral coherence to capture sequence order information, which was a good effort with prediction accuracy but was only done for EC 1 , i.e. Oxidoreductase [2].

Taking the same as base of our study we are also trying to develop an insilico classification system up to 2nd levels.

1.4. Objective

To build an insilico Enzyme Classification Tool called ***EnClass*** which will perform the following functions:

1. Sequence Property calculations: To calculate the properties of the given sequence namely the length of sequence, Molar_aliphatic, Molar_aromatic, hydrophobicity, Molar_acidic.
2. Next, The tool would predict whether the entered protein sequence is an enzyme or not.
3. If the predicted sequence is an enzyme , the tool predict its class and subclass on the basis of the spectral coherence based method.

CHAPTER- 2
TOOLS AND TECHNIQUES

2.1. Model development

The model for predicting whether a sequence is an enzyme or not, was developed using the following softwares.

2.1.1. Statistica 9.0

Statistica is a product by Statsoft. STATISTICA provides the most comprehensive array of data analysis, data management, data visualization, and data mining procedures. Its techniques include the widest selection of predictive modeling, clustering, classification, and exploratory techniques in one software platform.

STATISTICA is a tried and true analytics platform. STATISTICA is provided in six basic categories of product lines:

1. **Enterprise:** STATISTICA products designed for use by multiple users across a site or an entire organization, including the use of STATISTICA through thin client (Web browser) architectures access across a Wide Area Network.
2. **Web-Based Analytic Applications**: STATISTICA products deployed in a highly scalable, Web-based architecture for customized, turnkey Web based analytic applications.
3. **Data Mining Solutions**: The most comprehensive and effective system of user-friendly tools for the entire data mining process - from querying databases to generating final reports.
4. **Desktop:** STATISTICA products designed for use on a single workstation.
5. **Connectivity and Data Integration Solutions:** The configurations to data sources in *STATISTICA Enterprise* are defined and managed centrally using the *STATISTICA Enterprise* Manager Administration tools.
6. **Power Solutions:** Power Solutions is a combination of products and consulting. These solutions, based on predictive data mining and analytics,

produce immediate, significant improvement, and are offered at a fraction of the cost of the respective hardware upgrades necessary to produce similar but often not as effective - outcomes. **[3]**

2.1.2 Multiple Linear Regression- Statistica

Multiple linear regression (MLR) is a method used to model the linear relationship between a dependent variable and one or more independent variables. The dependent variable is sometimes also called the predictand, and the independent variables the predictors. MLR is based on least squares: the model is fit such that the sum-of-squares of differences of observed and predicted values is minimized. In the process of fitting, or estimating, the model, statistics are computed that summarize the *accuracy* of the regression model.

Coefficient of determination- The explanatory power of the regression is summarized by its

"R-squared" value, computed from the sums-of-squares terms as

$$R^2 = 1 - \sum (\hat{Y}i - \overline{Y})^2 / \sum (Yi - \overline{Y})^2$$

This is also called the coefficient of determination.

It is important to keep in mind that a high R^2 does not imply causation. The relative sizes of the sums-of-squares terms indicate how "good" the regression is in terms of fitting the calibration data. If the regression is "perfect", all residuals are zero, SSE is zero, and R^2 is 1. If the regression is a total failure, the sum-of-squares of residuals equals the total sum-of-squares, no variance is accounted for by regression, and R^2 is zero.[8].

2.2. Calculation of Physico- Chemical properties of novel sequence and model validation

The properties of the novel sequence and the parameters for model validation were calculated using perl scripts.[*Appendix* A1]

2.3 Calculation of mean squared coherence values

2.3.1 Mean Squared Coherence

Our method comprises three main components: Integration of all the known hydrophobicity scales from the AAIndex [9]. We then perform feature extraction using spectral coherence and carry out a supervised classification as validation for the goodness of using spectral coherence to capture sequence order information.

2.3.2 Integration of scales

There are 34 different scales found in [9] to estimate the hydrophobicity of amino acids. Each of the 34 scales depicts different aspects of the intermolecular forces involved within the protein and the properties of the protein itself.

For the purpose of integration, each hydrophobicity scale is first normalized between zero and one using the following relation.

$$S'_{\alpha}(i) = (S_{\alpha}(i) - \min(S_{\alpha}))/(\max(S_{\alpha}) - \min(S_{\alpha})) \qquad (1)$$

where is the normalized property values of scale S of index α, i stands for the amino acid ,and min(S) and max(S) represent the minimum and maximum of the scale S respectively.

Given a 20x34 vector, each amino acid i is represented as a vector in a 34 dimensional continuous space, where the components are the normalized property values.

2.3.2.1 Scalar product:

The scalar product between two vectors and , where i and j refer to different amino acids is given by

$$Q_{ij} = \underline{S}'(i) \bullet \underline{S}'(j) \qquad (2)$$

$$Q_{ij} = \sum_{\alpha=1}^{34} S'_{\alpha}(i) \cdot S'_{\alpha}(j) \tag{3}$$

The positive symmetric 20x20 matrix consists of the scalar products of the property vectors and , where i=1..20 and j=1..20.

2.3.2.2 Eigenvalues and Eigenvectors:

The Eigenvectors E and eigenvalues λ of the matrix are calculated using Principle Component Analysis (PCA)[10].

$$Q.E = \lambda E \tag{4}$$

As is of order 20, we obtain 20 eigenvectors and eigenvalues λ and the smallest eigenvalue λ 20 is equal to 0 due to normalization of the properties. Thus if μ represents the index of eigenvalue and eigenvector, then the elements of the matrix can be equated to eigenvalues and eigenvectors as:

$$Q_{ij} = \sum_{\mu=1}^{20} \lambda_{\mu} E_i^{\mu} . E_j^{\mu} \tag{5}$$

The first two significant eigenvalues are selected for the representation of amino acids, thus can be written as:

$$Q_{ij} \approx \sum_{\mu=1}^{2} \lambda_{\mu} E_i^{\mu} . E_j^{\mu} \tag{6}$$

Each amino acid can be represented as a vector in the two dimensional space with each dimension orthogonal (perpendicular) to each other. The coordinates of the ith amino acid can be written as:

$$\sqrt{\lambda}_{\mu=1} E_i^{\mu=1}, \sqrt{\lambda}_{\mu=1} E_i^{\mu=2} \tag{7}$$

The resultant vectors of equation (7) are believed to be the combined representation of all 34 scales, with one vector best representing hydrophobicity and the other hydrophilicity respectively.

2.3.2.3 Feature extraction

Once the two eigenvectors , where μ= {1, 2} are extracted, our next immediate objective of merging the expression levels of both the eigenvectors for a given protein sequence.

To solve the issues related with the length cardinality of protein sequences we opted to use Magnitude(Mean) Squared Coherence (MSC) function [11].

Let *Pi*, be a protein sequence of arbitrary length, where *i* be the index of amino acids in the sequence *P* .Given eigenvectors E^{μ} , can be represented as two random process such that *x(Pi)* and *y(Pi)* are the representations of *Pi* when $\mu=1$ and *2* respectively.

Now, for *x(Pi)* and *y(Pi)*

$$x(Pi)= [C_{AA1}.x_1^{\mu=1} , C_{AA2}.x_2^{\mu=1} , ---------,C_{AA20}.x_{20}^{\mu=1}]$$

$$y(Pi)= [C_{AA1}.x_1^{\mu=2} , C_{AA2}.x_2^{\mu=2} , ---------,C_{AA20}.x_{20}^{\mu=2}]$$

where, C_{AA} is the composition of particular amino acid in protein *Pi.*
X1, x2, x3, ------, x20 are the components of eigen vector.

2.3.2.4 Cross-Correlation Function:

The discrete cross-correlation R at lag j for discrete signals x_n and y_n.

$$R_{xy}(j)=\sum x_n y_{n-j}$$

2.3.2.5 Auto-Correlation Function:

The discrete autocorrelation R at lag j for a discrete signal x_n is

$$R_{xx}(j) = \sum_n x_n \, \overline{x}_{n-j}.$$

Where j= 0, ±1, ±2, ±3, ………±20.

2.3.2.6 Discrete-Fourier Transform:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi \frac{k}{N} n}.$$

We have used N=64 and k=0, 1, 2, 3…………………64.

2.3.2.7 Mean Squared Coherence:

Coherence is the property of vectors that quantifies the degree of interference. By interference we imply that if at least two random events are combined and depending on the relative phase between them, they can add constructively or subtract destructively.

The coherence function is the normalized cross spectral density,

$$\gamma_{xy}(\omega) = \frac{\Phi_{xy}(\omega)}{\sqrt{\Phi_{xx}(\omega)\Phi_{yy}(\omega)}} \tag{8}$$

where $\Phi_{xy}(\omega)$ is the cross spectral density at frequency ω between two zero-mean eigenvector (E^{μ}) representations of *Pi, x(Pi) and y(Pi)* , with auto spectra $\Phi_{xx}(\omega)$ and $\Phi_{yy}(\omega)$.

In particular, the magnitude-squared coherence (MSC) is represented as follows.

$$|\gamma_{xy}(\omega)|^2 = \frac{|\phi_{xy}(\omega)|^2}{\phi_{xx}(\omega)\phi_{yy}(\omega)} \tag{9}$$

For computing coherence we used the Fast Fourier Transformation (FFT) along with equation (9) to generate the Spectral coherence for a given protein *Pi* , using Eigenvector E^{u} where *u=1* and *2* respectively.

The feature extraction was done using matlab

2.4 Matlab

MATLAB[12] (matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by Math Works, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran. Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

2.5 Development of classification model using weighted random forests in statistica:

A Random Forest consists of a collection or ensemble of simple tree predictors, each capable of producing a response when presented with a set of 64 predictor values. For classification problems, this response takes the form of a class membership, which associates, or classifies, a set of independent predictor values with one of the categories present in the dependent variable. Alternatively, for regression problems, the tree response is an estimate of the dependent variable given the predictors. The Random Forest algorithm was developed by Breiman. A Random Forest consists of an arbitrary number of simple trees, which are used to determine the final outcome. For classification problems, the ensemble of simple trees vote for the most popular class. In the regression problem, their responses are averaged to obtain an estimate of the dependent variable. Using tree ensembles can lead to significant improvement in prediction accuracy (i.e., better ability to predict new data cases). The response of each tree depends on a set of predictor values chosen independently (with replacement) and with the same distribution for all trees in the forest, which is a subset of the predictor values of the original data set. The optimal size of the subset of predictor variables is given by $\log_2 M+1$, where M is the number of inputs.

For classification problems, given a set of simple trees and a set of random predictor variables, the Random Forest method defines a margin function that measures the extent to which the average number of votes for the correct class exceeds the average vote for any other class present in the dependent variable.

This measure provides us not only with a convenient way of making predictions, but also with a way of associating a confidence measure with those predictions. For regression problems, *Random Forests* are formed by growing simple trees, each capable of producing a numerical response value. Here, too, the predictor set is randomly selected from the same distribution and for all trees.

Given the above, the mean-square error for a Random Forest is given by:

$$\textbf{mean error} = (\textbf{observed - tree response})_2$$

The predictions of the Random Forest are taken to be the average of the predictions of the trees:

Where, the index k runs over the individual trees in the forest.

Typically, Random Forests can flexibly incorporate missing data in the predictor variables. When missing data are encountered for a particular observation (case) during model building, the prediction made for that case is based on the last preceding (non-terminal) node in the respective tree. So, for example, if at a particular point in the sequence of trees a predictor variable is selected at the root (or other non-terminal) node for which some cases have no valid data, then the prediction for those cases is simply based on the overall mean at the root (or other non-terminal) node.

2.6 Web Interface Development

The front end of EnClass was developed using HTML/CSS scripts.

2.6.1. HTML / CSS

HTML, which stands for Hyper Text Markup Language, is the predominant markup language for web pages. HTML is the basic building-blocks of web pages. Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors and fonts. [6]

2.7. Validation Method

2.7.1. Accuracy Calculation

For multiple linear regression model- The accuracy of the model was calculated by comparing the predicted value from the model developed and the actual values, i.e. a flag was used where 0 was used to depict the non-enzymes and 1 depicts enzymes from *EnClass,* and the true positives ,true negatives ,false positives and false negatives were calculated using a perl script [*Appendix* A2] based on the following criteria from the table given below:

Actual **Predicted**	**Enzyme (1)**	**Non -Enzyme(0)**
Enzyme(1)	True positive	False negative
Non-Enzyme(0)	False positive	True negative

Where, for regression model

True positive indicates that predicted value >0.45

False negative indicates that predicted value >0.45

False positive indicates that predicted value <0.45

True negative indicates that predicted value <0.45

$$Accuracy = \frac{True\ positives + True\ negatives}{True\ positives + True\ negetives + False\ negetives + False\ positives}$$

For weighted random forests model- The accuracy was calculated by comparing the predicted value from the model developed and the actual values, if the two values are equal it counts true positive and if the values are unequal it counts false negative.this was done using a perl script [*Appendix* A5] .

CHAPTER – 3
METHODOLOGY

3.1. Protocol for multiple linear regression model

We had 2,02,000 enzyme sequences and 60,000 non enzyme sequences as our input data.[7] This data is split into training set and test set. The training set consists of the data on the basis of which the model will be trained. The test set consists of the data which would be used to check the robustness of the newly developed model and to check if it is giving high accuracy results. The data in the training and the test set is partitioned in a 70: 30 ratio.

To divide the data into test and training data, firstly we clubbed the 2,02,000 enzymes and 60,000 non-enzymes so that there will be no biasedness. After the data was partitioned, we applied our algorithm, Multiple Linear Regression, which would be used for building up of the model. Multiple Linear Regression, algorithm only takes into account the most efficient set of attributes which are bound to give the best accuracy.

STEP 1: Training:

The data that has been partitioned into training set(70%) is taken and Multiple Linear Regression was applied onto it. The algorithm took various parameters into account and generated a model.

STEP 2: Testing:

The model developed is given the 30% of the test data set, which was not used for training, to check for its robustness. We have given a flag value of 0 to the data that are non-enzymes and 1 to the data that are enzymes. The model predicts the values of this flag as a dependent variable. The actual flag values were compared with predicted flag values and in this way the robustness of the model generated is tested.

STEP 3: Validation:

The model has been developed, trained and tested. It is now that we need to know if the model can correctly classify any novel user entered sequence into its appropriate class. Validation is a method of measuring the performance of a previously developed quantitative method. It is an objective way of verifying that the development stage has been successful and the method is performing to its expectations.

The validation techniques we used to predict the robustness of our model were:

- ✓ Squared correlation coefficient R^2
- ✓ Accuracy prediction

Accuracy:

Accuracy is the proportion of true negatives and true positives that are correctly identified by the model. [*Appendix* A2]

After the multiple linear regression model was developed and validated, to improve the accuracy and to solve the two major problems related with the protein data , we applied mean spectral coherence and random weighted forests.

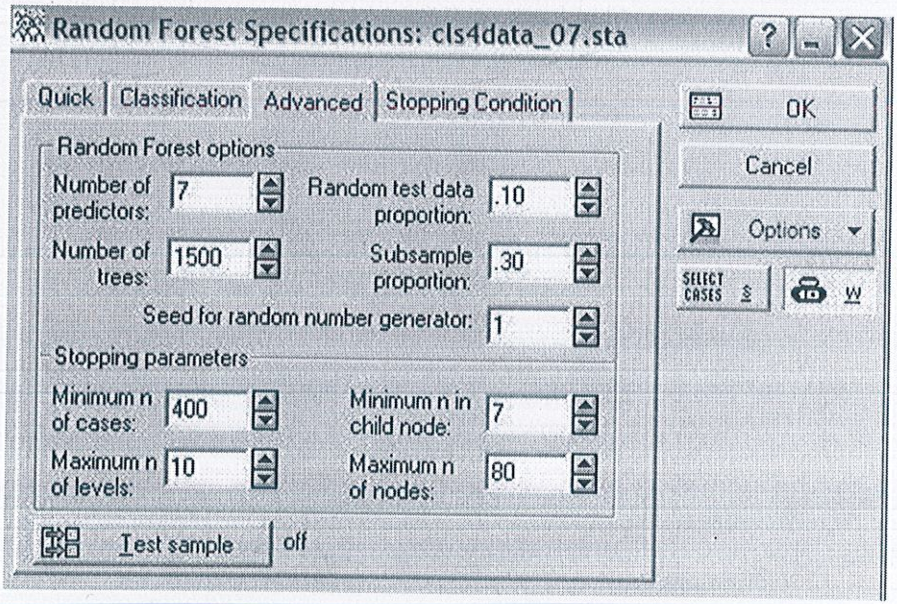
3.2 Protocol for development of model using mean squared coherence and weighted random forests

1. The scalar product was calculated using a perl script [*Appendix* A3] . The program takes the hydrophobicity scales [*Appendix* B1] values as input and returns the dot poduct matrix table [*Appendix* B2] as output. This dot product matrix values are used for calculation of eigenvectors and eigenvalues.
2. The eigenvectors and eigenvalues are calculated using a matlab program .
3. For each protein *Pi* , we have two vectors associated with it *x(Pi)* and *y(Pi)* or we can say that we have represented each protein into two vectors or directions.
4. These vectors were calculated using a perl script [*Appendix* A4] which takes into input the protein sequence and the perl script itself calculates the frequency of each amino acid in the sequence and the length of sequence and

gives the amino acid vectors , say x vector and y vextor for *x(Pi)* and *y(Pi)* where *Pi* is the protein sequence, as output. These vectors are used for calculation of MSC values for each protein sequence.

5. We get 39 values for both cross-correlation and auto-correlation using the matlab program [*Appendix* C1] .
6. The MSC was calculated using matlab program [Appendix C1] which takes the amino acid vectors i.e the x vector and y vector as input and returns 64 MSC values for a protein as an output.
7. When all these 64 values for all the proteins present in our database were calculated , we used those 64 values to develop a model using weighted random forests which will help in classification of a given sequence into its respective class and subclass.
8. The MSC values file for every class was imported to Statistica one by one and weighted random forest data mining technique was applied onto it.

The conditions applied for the classification are given in the snapshot from statistica



Random Forest Specifications: cls4data_07.sta

Random Forest options

Number of predictors: 7

Random test data proportion: .10

Number of trees: 1500

Subsample proportion: .30

Seed for random number generator: 1

Stopping parameters

Minimum n of cases: 400

Minimum n in child node: 7

Maximum n of levels: 10

Maximum n of nodes: 80

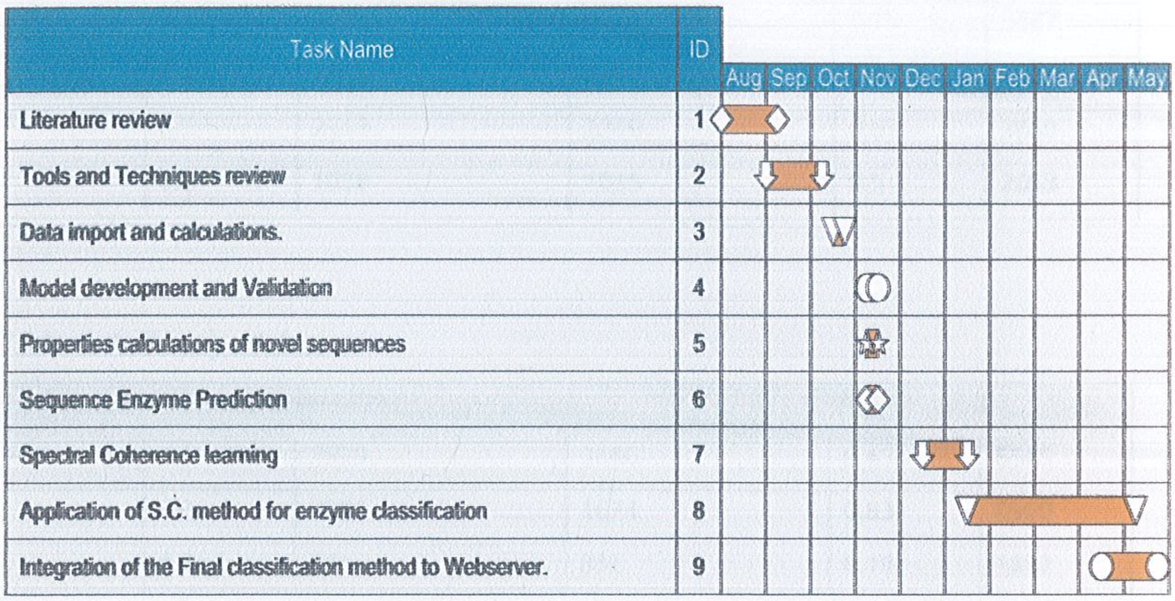
Test sample off

9. Using the above conditions the trees were built and the accuracy was calculated. To improve the accuracy more trees were built and the results were tabulated. [*Appendix* A5]

10. Visual basic code was generated for the developed models and these code can be used for classification purposes.[*Appendix* D]

RESULTS AND CONCLUSION

Work Plan



Using MLR the data has been divided into training and test data and the model was generated using the training data set. The model was tested on the test data set where the accuracy came out to be **72%** with the threshold value as 0.45 (if model predicts the value > 0.45, then the protein sequence is an enzyme and value <0.45 signifies entered protein sequence is a non-enzyme.) [*Appendix* A1]

The coefficient of determination came out to be **0.82** which is significant.

Using Weighted Random Forests the classification model was developed and for classes 6 and 4 the accuracy was achieved as follows:

Table 7. Accuracy of class 6.

class	subclass	correct predictions	wrong predictions	accuracy	total
6	6.1	8664	1562	0.84	10226
	6.2	423	1297	0.24	1720
	6.3	6990	2477	0.74	9467
	6.4	2893	1204	0.7	4097
	6.5	2341	1349	0.63	3690
	6.6	1038	1015	0.5	2053

Table 8. Accuracy of class 4.

class	subclass	Correct predictions	Wrong predictions	accuracy	total
4	4.1	2979	3666	0.45	6645
	4.2	7286	1674	0.83	8960
	4.3	211	891	0.19	1102
	4.4	10	434	0.022	444
	4.5	31	22	0.58	53
	4.6	169	320	0.345	489
	4.99	42	383	0.098	424

The accuracy can further be improved by generating more number of trees for each class.

Conclusion

As previously reported, the classes of enzymes are known to be correlated considerably with amino acid composition. Thus it is a challenge to find descriptors that are independent of the amino acid composition and yet capable of differentiating between the subfamilies of enzymes.

To this end, our ultimate goal was to introduce a unique method of extracting sequential order information based on the hydrophobic behavior of proteins using the statistical technique of Spectral Coherence. We have shown that this approach of feature extraction based on correlation between frequency amplitudes of two orthogonal profiles of the same sequence could identify unique patterns of amphiphilic residues. We validated our hypothesis; by training and testing our feature vector with a benchmark dataset to obtain comparable degrees of accuracies.

APPENDIX A
PERL SCRIPTS

A1.) To calculate the physico-chemical properties of the novel protein sequences and hence predicting whether the given sequence is an enzyme or non-enzyme.

```
#!/usr/bin/perl

print "\n**enter ur fyl name::";

$f=<stdin>;

open (MYFILE, "$f");

while (<MYFILE>)

{

        chomp $_;

        last if ($_ eq '');

        $seq=$_;

}

close (MYFILE);

@nwseq=split //,$seq;

print "@nwseq";

$len=@nwseq;

print $len;
print "\n";

$Molar_Aliphatic= 0;

$Molar_Aromatic=0;

$Molar_Acidic=0;

$hydrophobicity =0;

$count_of_A = 0.405767851;

$count_of_C = 0.602701239;

$count_of_D = 0.260904801;

$count_of_E =0.314868239;

$count_of_F = 0.643956039;

$count_of_G =0.319570698;

$count_of_H =0.365550068;

$count_of_I = 0.729315631;

$count_of_K = 0.322638396;

$count_of_L =0.641919103;

$count_of_M =0.569061629;

$count_of_N = 0.275187319;

$count_of_P = 0.405406469;

$count_of_Q =0.257203131;
```

```
$count_of_R = 0.33985266;

$count_of_S = 0.275544549;

$count_of_T =0.326643491;

$count_of_V = 0.633610582;

$count_of_W = 0.586783692;

$count_of_Y = 0.573512903;

foreach $base (@nwseq)

{

        if (($base eq 'A' ) || ($base eq 'I' ) ||($base eq 'L' ) || ($base eq 'V'))

                           {

                 ++$Aliphatic;

        }

        if (($base eq 'F' ) || ($base eq 'H' ) ||($base eq 'W' ) || ($base eq 'Y'))

          {

                  ++$Aromatic;

           }

        if (($base eq 'A' ) || ($base eq 'C' ) ||($base eq 'F' ) || ($base eq 'G')|| ($base
        eq 'I' )||      ($base eq 'L' ) ||($base eq 'M' ) || ($base eq 'P')|| ($base eq 'V'
        )||($base eq'W')||($base eq'Y'))

         {
              ++$Non_Polar;

   }

 if (($base eq 'D' ) || ($base eq 'E' ) ||($base eq 'H' ) || ($base eq 'K')|| ($base
 eq 'R' )|| ($base eq 'N' ) ||($base eq 'Q' ))

 {

              ++$Charged;

 }

 if(($base eq 'D' ) || ($base eq 'N' ) ||($base eq 'E' ) || ($base eq 'Q' ))

 {

              ++$Acidic;

 }

 if ( $base eq  'A' )

 {

              $hydrophobicity=$hydrophobicity+$count_of_A;

 }

 elsif ( $base eq 'C' )

 {

              $hydrophobicity=$hydrophobicity+$count_of_C;

 }

elsif ( $base eq 'D' )
```

```
{
        $hydrophobicity=$hydrophobicity+$count_of_D;
}
elsif ( $base eq 'E' )
{
        $hydrophobicity=$hydrophobicity+$count_of_E;
}
elsif ( $base eq 'F' )
{
        $hydrophobicity=$hydrophobicity+$count_of_F;
}
elsif ( $base eq 'G' )
{
        $hydrophobicity=$hydrophobicity+$count_of_G;
}
elsif ( $base eq 'H' )
{
        $hydrophobicity=$hydrophobicity+$count_of_H;
}
```

```
elsif ( $base eq 'I' )
{
        $hydrophobicity=$hydrophobicity+$count_of_I;
}
elsif ( $base eq 'K' )
{
        $hydrophobicity=$hydrophobicity+$count_of_K;
}
elsif ( $base eq 'L' )
{
        $hydrophobicity=$hydrophobicity+$count_of_L;
}
elsif ( $base eq 'M' )
{
        $hydrophobicity=$hydrophobicity+$count_of_M;
}
elsif ( $base eq 'N' )
{
        $hydrophobicity=$hydrophobicity+$count_of_N;
```

```
}

elsif ( $base eq 'P' )

{

        $hydrophobicity=$hydrophobicity+$count_of_P;

}

elsif ( $base eq 'Q' )

{

        $hydrophobicity=$hydrophobicity+$count_of_Q;

}

elsif ( $base eq 'R' )

{

        $hydrophobicity=$hydrophobicity+$count_of_R;

}

elsif ( $base eq 'S' )

{

        $hydrophobicity=$hydrophobicity+$count_of_S;

}

elsif ( $base eq 'T' )

{
```

```
                $hydrophobicity=$hydrophobicity+$count_of_T;

        }

        elsif ( $base eq 'V' )

        {

                $hydrophobicity=$hydrophobicity+$count_of_V;

        }

        elsif ( $base eq 'W' )

        {

                $hydrophobicity=$hydrophobicity+$count_of_W;

        }

        else{

                if ( $base eq 'Y' )

                  {

                    $hydrophobicity=$hydrophobicity+$count_of_Y;

                  }

            }

}

$Molar_Aliphatic= (($Aliphatic/$len)*100);

$Molar_Aromatic=(($Aromatic/$len)*100);
```

```
$Molar_Acidic=(($Acidic/$len)*100);

print "aliphatic = $Molar_Aliphatic\n";

print "aromatic=$Molar_Aromatic \n";

print "acidic =$Molar_Acidic\n";

print "hydrophobicity =$hydrophobicity\n";

$flag=(2.83+(0.00866*$Molar_Acidic)(10.7*$hydrophobicity)+(0.0384*$Molar
_Aromatic)   + (0.0578* $Molar_Aliphatic));

print "\n the predicted value is $flag\n";

if($flag>0.45)

        {

                print " \n the given sequence is an enzyme\n";

        }

else

        {

                print "\n the given sequence is a non enzyme\n";

        }

<stdin>;
```

A2.) To calculate the accuracy of the model .

```
#!/usr/bin/perl
```

```
print "enter ur file name::";

$file=<stdin>;

$g="data.txt";

open (OUTFYL, ">>$g");

my @data;

$j=0;

$k=0;

$accuracy=0;

open( FILE, $file ) or print "Can't open file '$file': $!";

while( <FILE> )

        {

                chomp;

                my @row = split;

                push @data, \@row;

        }

close( FILE );

$countn=0;

$countfp=0;

$counttp=0;

$countfn=0;

for($i=0; $i<10000; $i++)
```

```
    {
            print OUTFYL "$data[$i][7] and $data[$i][8]\n\n";
            if(($data[$i][7]==1 && $data[$i][8]>0.50) )
             {
                    $counttp++;
                    print OUTFYL " ::tp \n";
             }
            else{}
            if(($data[$i][7]=='0' && $data[$i][8]>0.50))
            {
                    $countfn++;
                     print OUTFYL " ::fn \n";
            }
            else{}
             if(($data[$i][7]==1 && $data[$i][8]<=0.50))
            {
                    $countfp++;
                    print OUTFYL " ::fp \n";
            }
            else{}
            if(($data[$i][7]=='0' && $data[$i][8]<=0.50))
```

```
                {

                        $counttn++;

                        print OUTFYL " ::tn \n";

                }

                else{}

        }

    print OUTFYL "false positives equals to $countfp\n\n";

    print OUTFYL "true positives equals to $counttp\n\n";

    print OUTFYL "false negatives equals to $countfn\n\n";

    print OUTFYL "true negatives equals to $counttn\n\n";

    $accuracy= (($counttp+$counttn)/($countfn+$counttp+$counttn+$countfp));

    print OUTFYL " \n\n the accuracy of the model is:  $accuracy";<stdin>;
```

A3) Program to calculate the dot(scalar) product of normalized hydrophobicity scales between two different amino acids.

```
#!/usr/bin/perl

print "enter ur file name::";

$file=<stdin>;

$g="data.txt";

open (OUTFYL, ">>$g");

my @data;
```

```
 $j=0;
 $k=0;
 $r=0;
 $nwmat=0;
 open( FILE, $file ) or print "############@#Can't open file '$file': $!";
 while( <FILE> ) {
  chomp;
  my @row = split;
  push @data, \@row;
 }
close( FILE );
for($i=0;$i<20;$i++)
{
 for($j=0;$j<20;$j++)
  {
  for($k=0;$k<34;$k++)
  {
  $new[$r]=$data[$k][$i]*$data[$k][$j];
 print "\n dot is equal to:: $new[$r]";
$nwmat=$nwmat+$new[$r];
$r++;
}
$nwmat1=$nwmat;
$nwmat=0;
print "\n\ngettin it:: $nwmat1";
$r=0;
$aa[$i][$j]=$nwmat1;
}
}
print "\n";
for($p=0;$p<34;$p++)
{
for($u=0;$u<20;$u++)
{
print OUTFYL "$aa[$p][$u] ";
}
print OUTFYL "\n";
}<stdin>;
```

A4) A program to calculate the amino acid vectors that represent a protein into two directions .

```
print "enter the name result file::";
```

```
$g=<stdin>;

open (OUTFYL, ">>$output_aminoacid_vector");

print "enter ur file name::";

$file=<stdin>;

$count_of_A=0;

$count_of_C=0;

$count_of_D=0;

$count_of_E=0;

$count_of_F=0;

$count_of_G=0;

$count_of_H=0;

$count_of_I=0;

$count_of_K=0;

$count_of_L=0;

$count_of_M=0;

$count_of_N=0;

$count_of_P=0;

$count_of_Q=0;

$count_of_R=0;

$count_of_S=0;

$count_of_T=0;
```

```
$count_of_V=0;

$count_of_W=0;

$count_of_Y=0;

$length=0;

open( FILE, $file ) or print "###########@#Can't open file '$file': $!";

while( <FILE> ) {

 chomp;

 my @row = split;

 push @data, \@row;

}

close( FILE );

#print OUTFYL "\n";

for($i=0; $i<3101; $i++)

{

$r=$i+1;

print   "\n\nS.no ($r.) \n";

$length=$data[$i][1];

print  "\nlength = $length\n";

$count_of_A=$data[$i][2];

print  "Adenine = $count_of_A\n";

$count_of_C=$data[$i][3];
```

```
print  "Cysteine = $count_of_C\n";
$count_of_D=$data[$i][4];
print  "Aspartic acid = $count_of_D\n";
$count_of_E=$data[$i][5];
print  "Glutamic acid = $count_of_E\n";
$count_of_F=$data[$i][6];
print  "Phenyl Alanine= $count_of_F\n";
$count_of_G=$data[$i][7];
print  "Glycine = $count_of_G\n";
$count_of_H=$data[$i][8];
print  "Histidine = $count_of_H\n";
$count_of_I=$data[$i][9];
print  "Isoleucine = $count_of_I\n";
$count_of_K=$data[$i][10];
print  "Lysine = $count_of_K\n";
$count_of_L=$data[$i][11];
print  "Leucine = $count_of_L\n";
$count_of_M=$data[$i][12];
print  "Methionine = $count_of_M\n";
$count_of_N=$data[$i][13];
print  "Aspargine = $count_of_N\n";
```

```
$count_of_P=$data[$i][14];

print  "Proline = $count_of_P\n";

$count_of_Q=$data[$i][15];

print  "Glutamine = $count_of_Q\n";

$count_of_R=$data[$i][16];

print  "Arginine = $count_of_R\n";

$count_of_S=$data[$i][17];

print  "Serine = $count_of_S\n";

$count_of_T=$data[$i][18];

print  "Threonine = $count_of_T\n";

$count_of_V=$data[$i][19];

print  "Valine = $count_of_V\n";

$count_of_W=$data[$i][20];

print  "Tryptophan = $count_of_W\n";

$count_of_Y=$data[$i][21];

print  "Tyrosine = $count_of_Y\n";

$xv[0]=($count_of_A*-0.2108)/$length;

$xv[1]=($count_of_L*-0.1464)/$length;

$xv[2]=($count_of_R*0.1220)/$length;

$xv[3]=($count_of_K*-0.1836)/$length;

$xv[4]=($count_of_N*0.0831)/$length;
```

```
$xv[5]=(-0.1665*$count_of_M)/$length;
$xv[6]=($count_of_D*0.1249)/$length;
$xv[7]=($count_of_F*0.3028)/$length;
$xv[8]=($count_of_C*0.3215)/$length;
$xv[9]=($count_of_P*0.2561)/$length;
$xv[10]=($count_of_Q*0.2683)/$length;
$xv[11]=($count_of_S*0.0148)/$length;
$xv[12]=($count_of_E*0.1675)/$length;
$xv[13]=($count_of_T*-0.4140)/$length;
$xv[14]=($count_of_G*-0.1323)/$length;
$xv[15]=($count_of_W*0.3258)/$length;
$xv[16]=($count_of_H*-0.0513)/$length;
$xv[17]=($count_of_Y*-0.0895)/$length;
$xv[18]=($count_of_I*-0.2054)/$length;
$xv[19]=($count_of_V*0.3559)/$length;

$yv[0]=($count_of_A*0.0947)/$length;
$yv[1]=($count_of_L*-0.1855)/$length;
$yv[2]=($count_of_R*-0.5651)/$length;
$yv[3]=($count_of_K*0.0315)/$length;
$yv[4]=($count_of_N*-0.0518)/$length;
```

```
$yv[5]=(0.0919*$count_of_M)/$length;

$yv[6]=($count_of_D*-0.3056)/$length;

$yv[7]=($count_of_F*0.0415)/$length;

$yv[8]=($count_of_C*-0.0518)/$length;

$yv[9]=($count_of_P*0.2485)/$length;

$yv[10]=($count_of_Q*-0.1628)/$length;

$yv[11]=($count_of_S*0.4513)/$length ;

$yv[12]=($count_of_E* -0.1637)/$length;

$yv[13]=($count_of_T*0.0403)/$length;

$yv[14]=($count_of_G*-0.2159)/$length ;

$yv[15]=($count_of_W*0.0012)/$length;

$yv[16]=($count_of_H*0.1197)/$length;

$yv[17]=($count_of_Y*0.1718)/$length;

$yv[18]=($count_of_I*-0.1360)/$length;

$yv[19]=($count_of_V*0.3132)/$length;

print OUTFYL "\n";

print OUTFYL "x(vector)::";

for($f=0; $f<20; $f++)

{

        print OUTFYL "  $xv[$f]  ";

}
```

```
print OUTFYL "\ny(vector)::";
for($q=0; $q<20; $q++)
{
        print OUTFYL "  $yv[$q]  ";
}
}<stdin>;
```

A5) Program to calculate the accuracy of the weighted random forests model.

```
print "enter no. of subclasses";
$subb=<stdin>;
sub classes()
{
   $file=0;
   print "\nenter ur input file name::";
   $file=<stdin>;
   $count=0;
   $g=0;
   print "\nenter ur output file name::";
   $g=<stdin>;
}
for ($loop=0;$loop<$subb; $loop++)
{
```

```
classes;
open (OUTFYL, ">>$g");
my @data;
$j=0;
$k=0;
$accuracy=0;
open( FILE, $file ) or print "###########@#Can't open file '$file': $!";
while( <FILE> ) {
chomp;
my @row = split;
push @data, \@row;
}
close( FILE );
$counttn=0;
$countfp=0;
$counttp=0;
$countfn=0;
print "\nenter no. of rows";
$end=<stdin>;
for($i=0; $i<$end; $i++)
{
    $count++;

    print OUTFYL "$count::::::: $data[$i][0] and $data[$i][1]\n\n";

    if(($data[$i][0]==$data[$i][1]) )

     {

        $counttp++;

        print OUTFYL " ::tp \n";

      }

      else{}

      if(($data[$i][0]!=$data[$i][1]))

       {

          $countfp++;

          print OUTFYL " ::fp \n";

        }

  }

  print OUTFYL "false positives equals to $countfp\n\n";

  print OUTFYL "true positives equals to $counttp\n\n";

  $accuracy= (($counttp+$counttn)/($countfn+$counttp+$counttn+$countfp));

  print OUTFYL " \n\n accuracy is equals to $accuracy";

}

<stdin>
```

APPENDIX B

TABLES

[1] NORMALISED HYDROPHOBICITY SCALE VALUES

	A	L	R	K	N	M	D	F	C	P	Q	S	E	T	G	W	H	Y	I	V
ARGP820101	0.23	0.58	0.23	0.43	0.02	0.45	0.17	0.76	0.40	0.74	0.00	0.02	0.18	0.02	0.03	1.00	0.23	0.71	0.84	0.50
CIDH920101	0.36	0.93	0.43	0.39	0.44	0.96	0.00	1.00	0.77	0.47	0.18	0.18	0.24	0.27	0.17	0.96	0.86	1.00	0.76	0.92
CIDH920102	0.35	0.72	0.34	0.34	0.17	0.72	0.17	0.80	0.58	0.37	0.26	0.11	0.00	0.20	0.13	1.00	0.49	0.80	0.76	0.67
CIDH920103	0.46	0.71	0.20	0.19	0.09	0.72	0.03	0.66	0.56	0.34	0.04	0.18	0.11	0.00	0.11	0.74	0.40	0.67	1.00	0.72
CIDH920104	0.42	0.66	0.15	0.18	0.09	0.55	0.05	0.76	0.75	0.30	0.00	0.11	0.00	0.18	0.19	0.83	0.29	0.57	1.00	0.74
CIDH920105	0.39	0.77	0.24	0.25	0.13	0.73	0.03	0.84	0.65	0.36	0.01	0.06	0.00	0.13	0.12	0.97	0.47	0.76	1.00	0.77
EISD840101	0.81	0.92	0.00	0.27	0.45	0.81	0.42	0.95	0.72	0.68	0.43	0.60	0.46	0.63	0.77	0.86	0.55	0.71	1.00	0.92
GOLD730101	0.25	0.80	0.25	0.50	0.23	0.43	0.00	0.88	0.33	0.87	0.20	0.00	0.00	0.15	0.00	1.00	0.00	0.95	0.98	0.57
JOND750101	0.23	0.58	0.23	0.44	0.02	0.44	0.18	0.76	0.40	0.73	0.00	0.02	0.18	0.02	0.03	1.00	0.23	0.71	0.84	0.50
MANP780101	0.44	0.83	0.18	0.10	0.12	0.73	0.00	0.65	0.78	0.11	0.19	0.08	0.21	0.17	0.33	0.63	0.27	0.53	0.99	1.00
PONP800101	0.35	0.77	0.16	0.00	0.05	0.83	0.04	0.62	0.97	0.09	0.11	0.11	0.09	0.20	0.28	0.50	0.48	0.58	0.93	1.00
PONP800102	0.36	0.70	0.21	0.00	0.09	0.79	0.09	0.63	1.00	0.18	0.18	0.23	0.13	0.26	0.31	0.52	0.41	0.54	0.82	0.89
PONP800103	0.41	0.69	0.27	0.00	0.12	0.85	0.14	0.73	1.00	0.27	0.27	0.39	0.15	0.35	0.35	0.59	0.36	0.54	0.77	0.88
PONP800104	0.61	0.69	0.07	0.22	0.29	0.55	0.00	0.71	0.80	0.12	0.07	0.06	0.36	0.46	1.00	0.25	0.14	0.38	0.83	0.43
PONP800105	0.60	1.00	0.31	0.32	0.00	0.94	0.42	0.51	0.87	0.49	0.05	0.33	0.38	0.58	0.51	0.45	0.76	0.63	0.49	0.96
PONP800106	0.15	0.62	0.22	0.00	0.18	1.00	0.06	0.66	0.83	0.14	0.35	0.25	0.35	0.12	0.20	0.29	0.42	0.31	0.60	0.78
PRAM900101	0.13	0.06	1.00	0.78	0.53	0.02	0.81	0.00	0.11	0.24	0.49	0.19	0.74	0.16	0.17	0.11	0.42	0.27	0.04	0.07
SWER830101	0.28	0.78	0.22	0.20	0.12	0.72	0.00	1.00	0.46	0.25	0.12	0.24	0.03	0.32	0.20	0.56	0.21	0.92	0.79	0.69
ZIMJ680101	0.27	0.82	0.27	0.52	0.03	0.46	0.21	0.90	0.48	0.88	0.00	0.05	0.21	0.18	0.03	0.10	0.36	0.97	1.00	0.58
PONP930101	0.47	0.73	0.32	0.00	0.16	0.60	0.02	0.67	0.76	0.01	0.18	0.15	0.09	0.26	0.27	0.68	0.33	0.59	1.00	0.86
WILM950101	0.33	0.82	0.20	0.09	0.35	0.35	0.29	1.00	0.39	0.42	0.36	0.23	0.30	0.41	0.35	0.64	0.00	0.59	0.81	0.54
WILM950102	0.47	0.79	0.36	0.03	0.15	0.00	0.02	1.00	0.31	0.24	0.12	0.14	0.22	0.40	0.16	0.74	0.19	0.37	0.61	0.44
WILM950103	0.13	0.33	0.00	0.07	0.33	0.60	0.32	0.15	1.00	0.51	0.26	0.39	0.35	0.44	0.11	0.47	0.83	0.45	0.50	0.30
WILM950104	0.06	0.40	0.45	0.44	0.57	0.35	0.25	0.54	0.78	0.28	0.31	0.48	0.42	0.31	0.19	0.64	0.00	0.04	1.00	0.49
JURD980101	0.65	0.93	0.00	0.10	0.17	0.73	0.16	0.82	0.79	0.33	0.15	0.48	0.20	0.46	0.46	0.48	0.20	0.40	1.00	0.97
WOLR790101	0.98	0.99	0.00	0.47	0.46	0.83	0.46	0.86	0.84	0.82	0.47	0.67	0.43	0.67	1.00	0.63	0.43	0.62	0.99	0.98
KIDA850101	0.38	0.14	1.00	0.95	0.69	0.24	0.69	0.04	0.15	0.24	0.78	0.53	0.80	0.64	0.41	0.00	0.54	0.29	0.23	0.34
COWR900101	0.66	1.00	0.18	0.06	0.31	0.85	0.43	0.98	0.76	0.77	0.32	0.40	0.47	0.49	0.56	0.91	0.00	0.68	1.00	0.89
BLAS910101	0.62	0.94	0.00	0.28	0.24	0.74	0.03	1.00	0.68	0.71	0.25	0.36	0.04	0.45	0.50	0.88	0.17	0.88	0.94	0.83
CASG920101	0.51	0.60	0.26	0.00	0.31	0.60	0.06	0.74	1.00	0.17	0.14	0.26	0.09	0.34	0.43	0.91	0.57	0.60	0.86	0.66
ENGD860101	0.13	0.06	1.00	0.78	0.53	0.02	0.81	0.00	0.11	0.24	0.49	0.19	0.74	0.16	0.17	0.11	0.42	0.28	0.04	0.07
FASG890101	0.59	0.14	0.82	1.00	0.71	0.24	0.75	0.14	0.00	0.68	0.76	0.78	0.84	0.69	0.61	0.27	0.48	0.51	0.12	0.26
HOPT810101	0.45	0.25	1.00	1.00	0.56	0.33	1.00	0.14	0.38	0.53	0.56	0.58	1.00	0.47	0.53	0.00	0.45	0.17	0.25	0.30
KUHL950101	0.28	0.08	1.00	0.57	0.66	0.17	0.79	0.00	0.07	0.20	0.65	0.48	0.88	0.52	0.19	0.21	0.47	0.48	0.00	0.04

[2] dot product values matix(20*20)

6.9777	9.8879	3.9262	4.2118	3.8433	8.5873	3.5936	9.6673	8.8561	6.0208	3.6469	4.4256	4.3259	5.3189	5.8026	8.4651	5.0954	8.2193	10.7362	9.7361
9.8879	16.7665	4.9337	5.1842	4.7367	14.3641	3.7932	16.8741	14.8523	9.4292	4.3513	5.5343	4.9935	7.1082	7.4829	14.6222	7.5895	13.8646	18.4535	16.2195
3.9262	4.9337	7.2815	6.188	4.672	4.5711	5.4143	4.7618	4.6906	4.2015	4.512	3.7639	5.9251	3.9992	3.2807	4.7528	4.6842	5.5144	5.5131	5.135
4.2118	5.1842	6.188	6.5359	4.4219	4.6152	5.1342	5.0554	4.3842	5.2712	4.1708	3.7889	5.4927	4.0085	3.6216	4.8758	4.3261	5.7829	5.8876	5.1157
3.8433	4.7367	4.672	4.4219	4.0689	4.3841	3.9823	4.6764	4.6466	3.8315	3.7618	3.5835	4.5623	3.8255	3.5794	4.4873	3.6556	4.6189	5.4115	4.7904
8.5873	14.3641	4.5711	4.6152	4.3841	13.6083	3.6221	14.1024	13.8642	7.9513	4.1783	5.2917	4.7324	6.307	6.7389	12.5402	7.5338	12.0664	15.9318	14.591
3.5936	3.7932	5.4143	5.1342	3.9823	3.6221	5.1756	3.3121	3.6861	4.0033	3.967	3.6871	5.3452	3.7634	3.3969	3.2607	3.7413	4.014	3.9896	3.901
9.6673	16.8741	4.7618	5.0554	4.6764	14.1024	3.3121	17.6612	14.5965	9.5965	4.1531	5.2479	4.5565	6.8151	7.1271	15.3073	7.1087	14.1735	18.935	16.0574
8.8561	14.8523	4.6906	4.3842	4.6466	13.8642	3.6861	14.5965	15.2522	7.8555	4.2098	5.4839	4.9256	6.659	7.134	13.0723	7.8326	12.0062	16.9615	15.1049
6.0208	9.4292	4.2015	5.2712	3.8315	7.9513	4.0033	9.5965	7.8555	7.6652	3.5622	4.0966	4.4178	4.7531	4.5616	8.7878	4.8495	8.8347	10.5006	8.7934
3.6469	4.3513	4.512	4.1708	3.7618	4.1783	3.967	4.1531	4.2098	3.5622	3.7947	3.5241	4.4259	3.6753	3.3894	3.8966	3.442	4.2581	4.769	4.5657
4.4256	5.5343	3.7639	3.7889	3.5835	5.2917	3.6871	5.2479	5.4839	4.0966	3.5241	4.0098	4.1776	4.1619	4.0097	4.7184	3.7488	4.7922	6.0647	5.774
4.3259	4.9935	5.9251	5.4927	4.5623	4.7324	5.3452	4.5565	4.9256	4.4178	4.4259	4.1776	5.9953	4.4108	4.1572	4.1365	4.3269	4.8276	5.3641	5.0885
5.3189	7.1082	3.9992	4.0085	3.8255	6.307	3.7634	6.8151	6.659	4.7531	3.6753	4.1619	4.4108	4.8709	4.7702	5.8886	4.3058	6.0728	7.5407	7.0065
5.8026	7.4829	3.2807	3.6216	3.5794	6.7389	3.3969	7.1271	7.134	4.5616	3.3894	4.0097	4.1572	4.7702	5.5943	5.872	3.9914	5.8763	8.0266	7.4485
8.4651	14.6222	4.7528	4.8758	4.4873	12.5402	3.2607	15.3073	13.0723	8.7878	3.8966	4.7184	4.1365	5.8886	5.872	15.0259	6.8403	12.8611	16.8529	14.1589
5.0954	7.5895	4.6842	4.3261	3.6556	7.5338	3.7413	7.1087	7.8326	4.8495	3.442	3.7488	4.3269	4.3058	3.9914	6.8403	6.1121	7.2127	8.1805	7.8929
8.2193	13.8646	5.5144	5.7829	4.6189	12.0664	4.014	14.1735	12.0062	8.8347	4.2581	4.7922	4.8276	6.0728	5.8763	12.8611	7.2127	12.9646	15.4194	13.3338
10.7362	18.4535	5.5131	5.8876	5.4115	15.9318	3.9896	18.935	16.9615	10.5006	4.769	6.0647	5.3641	7.5407	8.0266	16.8529	8.1805	15.4194	21.5647	18.0786
9.7361	16.2195	5.135	5.1157	4.7904	14.591	3.901	16.0574	15.1049	8.7934	4.5657	5.774	5.0885	7.0065	7.4485	14.1589	7.8929	13.3338	18.0786	16.3768

APPENDIX C

MATLAB PROGRAM

C1) Program to calculate the mean squared coherence for n number of proteins.

```
clear all

clc

load matlab.mat;

N = 64;          % number of samples

for i = 0:size(x,1)/2-1

  i

  a = x(2*i+1,:);

  b = x(2*i+2,:);

  phi_xy = xcorr(a,b);

  phi_x = xcorr(a);

  phi_y = xcorr(b);

  si(i+1,:) = abs (square(abs(fft(phi_xy,N)))./(fft(phi_x,N).*fft(phi_y,N)));

  max_sai(i+1)=max(si(i+1,:));

  end
```

APPENDIX D

VISUAL BASIC CODE

D1) Visual basic code for the model developed for class 4

```
Option Base 1

Sub Main

Dim newanalysis As Analysis

Set newanalysis = Analysis (scRandomForest, ActiveDataSet)

Dim oAD1 As STARandomForest.RandomForestStartup

Set oAD1 = newanalysis.Dialog

oAD1.TreeAnalysis = 0

newanalysis.Run

Dim oAD2 As STARandomForest.RandomForestSpecification

Set oAD2 = newanalysis.Dialog

oAD2.Variables = "65 |    | 1-64"

oAD2.ResponseCodes = "1-6 99"

oAD2.EqualMisclassificationCost = True

oAD2.UseEstimatedPriors = True

oAD2.NumberOfPredictorsInEachNode = 7

oAD2.NumberOfAdditiveTrees = 100
```

```
oAD2.SubsampleProportion = 0.5

oAD2.AutomaticTestDataProportion = 0.3

oAD2.MinimumNToStop = 500

oAD2.MinimumChildNodeSizeToStop = 5

oAD2.MaximumNumberOfNodes = 100

oAD2.EnableAdvanceStoppingCondition = True

oAD2.SeedForRandomNumberGenerator = 1

oAD2.PercentageDecrease = 5

oAD2.NumberOfCyclesForAverage = 10

oAD2.CrossValidation = "off"

oAD2.MaximumNumberOfLevelsInTree = 10

newanalysis.CaseWeightSource = scCWSourceSpreadsheet

With newanalysis.CaseWeight

            .Enabled = True

            .Variable = 66

      End With

oAD2.NumberOfAdditiveTrees = 1500

oAD2.SubsampleProportion = 0.3

oAD2.AutomaticTestDataProportion = 0.1

oAD2.MinimumChildNodeSizeToStop = 7

oAD2.MaximumNumberOfNodes = 80
```

```
newanalysis.Run
Dim oAD3 As STARandomForest.RandomForestResults
Set oAD3 = newanalysis.Dialog
oAD3.ResponseCategory = 7
oAD3.StartTreeNumber = 1
oAD3.EndTreeNumber = 1
oAD3.NumberOfTreesForModel = 560
oAD3.NumberOfMoreTreesToCreate = 100
oAD3.AnalysisDataSet = True
oAD3.LiftChartLiftValue = True
oAD3.ResponseCategory = 7
oAD3.CumulativeLiftchart = True
newanalysis.RouteOutput(oAD3.PredictedValues).Visible = True
oAD3.NumberOfTreesForModel = 1560
oAD3.NumberOfMoreTreesToCreate = 1000
newanalysis.RouteOutput(oAD3.CreateMoreTrees).Visible = True
newanalysis.RouteOutput(oAD3.PredictedValues).Visible = True
End Sub
```

D2) Visual basic code for the model developed for class 6

```
Sub Main
```

```
Dim newanalysis As Analysis
Set newanalysis = Analysis (scRandomForest, ActiveDataSet)
Dim oAD1 As STARandomForest.RandomForestStartup
Set oAD1 = newanalysis.Dialog
oAD1.TreeAnalysis = 0
newanalysis.Run
Dim oAD2 As STARandomForest.RandomForestSpecification
Set oAD2 = newanalysis.Dialog
oAD2.Variables = "65 |   | 1-64"
oAD2.ResponseCodes = "1-6"
oAD2.EqualMisclassificationCost = True
oAD2.UseEstimatedPriors = True
oAD2.NumberOfPredictorsInEachNode = 7
oAD2.NumberOfAdditiveTrees = 500
oAD2.SubsampleProportion = 0.5
oAD2.AutomaticTestDataProportion = 0.2
oAD2.MinimumNToStop = 975
oAD2.MinimumChildNodeSizeToStop = 7
oAD2.MaximumNumberOfNodes = 70
oAD2.EnableAdvanceStoppingCondition = True
oAD2.SeedForRandomNumberGenerator = 1
```

```
oAD2.PercentageDecrease = 5

oAD2.NumberOfCyclesForAverage = 10

oAD2.CrossValidation = "off"

oAD2.MaximumNumberOfLevelsInTree = 15

newanalysis.CaseWeightSource = scCWSourceSpreadsheet

With newanalysis.CaseWeight

                .Enabled = True

                .Variable = 66

        End With

newanalysis.Run

Dim oAD3 As STARandomForest.RandomForestResults

Set oAD3 = newanalysis.Dialog

oAD3.ResponseCategory = 6

oAD3.StartTreeNumber = 1

oAD3.EndTreeNumber = 1

oAD3.NumberOfTreesForModel = 1190

oAD3.NumberOfMoreTreesToCreate = 1500

oAD3.AnalysisDataSet = True

oAD3.LiftChartLiftValue = True

oAD3.ResponseCategory = 6

oAD3.CumulativeLiftChart = True
```

```
newanalysis.RouteOutput(oAD3.CreateMoreTrees).Visible = True

newanalysis.RouteOutput(oAD3.CodeGeneratorCLang).Visible = True

newanalysis.RouteOutput(oAD3.PredictedValues).Visible = True

End Sub
```

REFERENCES

[1] IUBMB Biochemical Nomenclature. —Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB) Internet: http://www.chem.qmul.ac.uk/iubmb/enzyme/rules.html, [August, 2011]

[2] S. Dua and P. Chowriappa, —Prediction of Enzyme Classes using Spectral based AA Information, Proc. of 12th Information Conference on Information Technology, Dec. 21-24, 2009

[3] STATISTICA Product Overview.
Internet http://www.statsoft.com/products/ , [September , 2011]

[4] Minitab
Internet: http:// en.wikipedia.org/wiki/Minitab , [October , 2011]

[5] About Perl.
Internet: http://www.perl.org/about.html, [September , 2011]

[6] Wikipedia, —HTML,
Internet: http://en.wikipedia.org/wiki/HTML, [October , 2011]

[7] EnClass: A DATA WAREHOUSE OF ENZYMES AND WRF BASED TOOL FOR HIERARCHIAL CLASSIFICATION OF ENZYMES- Pranshu Saxena ,Neha Sood [May-2011]

[8] Multiple Linear Regression www.ltrr.arizona.edu/~dmeko/notes_11.pdf [November ,2011]

[9] Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., and Kanehisa, M., "AAindex: Amino Acid Index Database, Progress Report 2008," Nucleic Acids Research, vol.36, D202- D205, 2008.

[10] Jackson, J. E., "A User's Guide to Principal Components," John Wiley and Sons, Hoboken, NJ, 1991, p. 592.

[11] Carter, G. C., Knapp, C. H., Nuttall, A. H., "Estimation of the Magnitude-Squared Coherence Function via Overlapped Fast Fourier Transform Processing," IEEE Transactions on Audio and Electroacustics, IEEE, New York, NY, 1973, pp. 331- 344.

[12] Matlab
en.wikipedia.org/wiki/MATLAB, [March 2012].

BREIF PROFILE OF STUDENTS

Payal Bhargava

She is pursuing her B.Tech in Bioinformatics from Jaypee University of Information Technology and will be completing her degree in June 2012. Her technical and research interests include data mining and datawarehousing, drug designing and chemoinformatics, perl programming, php,C++, HTML. She will be pursuing her higher studies in the field of Bioinformatics itself.

Aahut Chandwani

He is pursuing his B.Tech in Bioinformatics from Jaypee University of Information Technology and will be completing his degree in June 2012. His technical and research interests include data mining, perl programming. He will be pursuing his higher studies in the field of Bioinformatics itself.