USING PROTEIN STRUCTURE PREDICTION TO UNDERSTAND THE EVOLUTION OF GENETIC CODING

By

Mohsen Khorshid

THESIS WORK
Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE
BIOINFORMATICS

Chalmers University of Technology
Gothenburg, Sweden

Supervised by:

Peter R. Wills, Ph.D.
Alexei J. Drummond, Ph.D.
New Zealand Bioinformatics Institute
University of Auckland
Auckland, New Zealand

Examined by:

Graham J.L. Kemp, Ph.D.
Chalmers University of Technology
Gothenburg, Sweden

ABSTRACT

The mechanism of emergence of genetic coding is one of the processes in prebiotic evolution that is not yet very well understood. Genetic Coding is a regular mapping from the set of tri-nucleotide codons onto the 20 standard amino acids. The mapping is mediated by enzymes, such as aminoacityl-tRNA synthetases (Aminoacityl-tRNA Synthetase) which catalyze the assignment of a particular amino acid to its set of cognate codons. Some indication of the path of evolution of the current system of genetic coding from simpler systems could be found in the structure of the Aminoacityl-tRNA Synthetases. The aim of this project is to use protein structure prediction of aminoacityl-tRNA Synthetase enzymes to understand the evolution of genetic coding. I have used a non-redundant set of experimentally determined Aminoacityl-tRNA Synthetase structures (PDB files) as basis for predicting structures. I applied THREADER on these structures against approximately 3000 Aminoacityl-tRNA Synthetase sequences with unknown structures to choose the most plausible folding templates among these known structures. Then by using MODELLER and Python coding, I have generated the PDB formatted files for each Aminoacityl-tRNA Synthetase sequence by applying multiple template prediction method on the corresponding results from THREADER. The efficiency of this combination for 3D structure modeling of these enzymes has been self-tested on experimentally determined Aminoacityl-tRNA Synthetase structures. To predict the catalytic domain of the predicted structures for each Aminoacityl-tRNA Synthetase sequence, I have developed a customized multiple sequence alignment with structural considerations, using algorithms in STRAP and BioJava libraries. By applying this alignment to my data the catalytic domains of the Aminoacityl-tRNA Synthetases have been identified, extracted and aligned for two distinct classes I, II of this type of enzymes. Finally, Tree-construction methodologies have been applied to reconstruct the evolution of specificity in amino acid recognition needed for Aminoacityl-tRNA Synthetase function. That is, given the alignment of catalytic domains, I have provided Maximum Likelihood (ML) trees. The tree topologies of these trees have been evaluated and furthermore Bayesian Inference using BEAST has been applied to find the posterior distribution of these predicted Aminoacityl-tRNA Synthetase catalytic domains. From the results, we concluded that the ancestry of Aminoacityl-tRNA Synthetases catalytic domains is preceding the last universal common ancestral (LUCA) of organism. That is, the evolution of catalytic domain for each type of Aminoacityl-tRNA Synthetase has been convergent in diverse organisms. We still could not find common pattern of ancestry between Classes I, II. The phylogeny of catalytic domain in different Aminoacityl-tRNA Synthetases types follows relationship between the Eukaryotes, Bacterial and Archaea (canonical phylogenetics pattern). In the end, we concluded that using 3D structure protein modeling seems to be successful for the Aminoacityl-tRNA Synthetase catalytic domain evolutionary analysis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Prof. Peter R. Wills and Dr. Alexei J. Drummond. They made this project possible and interesting. Their support for me is so valuable. IN this short time, I learned so many things - not just science - by working under their supervision. Without their comments and discussions, the thesis would have been a lonely endeavor indeed. I should also thank Dr. Graham J. Kemp for his remarks and explanations during the project work that always guided me remain in the right track. I am so grateful to the New Zealand's Bioinformatics Institute at University of Auckland for funding this work and for hosting me where I enjoyed a lovely work place. I appreciate the support for providing me the computing and library facilities.

I have to acknowledge the help of Prof. Allen Rodrigo, Dr. Howard Ross, the PhD students and my colleagues at bioinformatics institute for their helpful feedbacks and comments. I am grateful for Dr. Derrick J. Zwickl and Dr. Stéphane Guindon for their supportive and constructive role in my project.

I would like to thank my parents and family for their support that made the studying abroad possible. I would also like to show appreciations to the Bioinformatics program staff at Chalmers University of Technology.

Mohsen Khorshid

1 Introduction

Common to all life on earth are the mechanisms of genetic encoding. They are the linchpin of translation, the link between the worlds of protein and nucleic acid. The aminoacyl-tRNA synthetases (aaRS) are the key proteins involved in setting the genetic code in all living organisms. They are found in all three domains of life Bacteria, Archaea and Eukarya. It is not only the structure-function aspect of these enzymes, which has captured the Biologist's imagination, but the possibility that they could tell us the secrets of the genetic code. In this project, I try to focus on these enzymes in order to get some insight on how Genetic Coding evolved.

1.1 Problem

Understanding aminoacyl-tRNA synthetases in standard molecular terms is to add one more piece, a most important one, to the puzzle of what the cell is and how it works. The RNA world hypothesis suggests that the modern biological world evolved from a form of life that was mostly RNA-based. These ancient proteins are found in all extant organisms, and their inception likely predates the root of the universal phylogenetic tree. The evolution of these proteins is of particular interest for understanding the evolution of translation and the transition from the RNA world to the modern form of life dominated by protein-enzymes and DNA genomes.

Let us assume that all molecular biological systems involve the translation of genetic information because the proteins that catalyze the coding are themselves products of that synthesis. Let us also consider the genetic coding as a molecular information processing system where information represented as genes. A question that arises at this time, is how such a complex self-contained information-processing system can maintain physical stability and accuracy. More basic question is how such a complex system has emerged from molecular disorder in the first place. Some works have been done on modeling the characteristics of an autocatalytic system that can achieve stability and can self-organize to the type of complexity seen in modern genetic systems. Self-organization occurs when the accuracy of symbol replication is above a certain system-determined threshold. The idea has been developed by proposing characteristics models for such systems.

Orgel (1963) Hoffman(1974) and Füchslin and McCaskill (2001) developed more complex models for self-organization by adding parameters like better error catastrophe threshold or error-prone replication process. On the other hand, Peter R. Wills (1993:2004) suggested stepwise evolution of molecular biological coding.

The idea is to look at the genetic coding system as an information system. The current work of Markowitz, Wills and Drummond for Simulation Model of Prebiotic

Evolution of Genetic Coding[1] is another step in the area of biomolecular information systems where the genes and proteins presented by bits of data and the starting point is the randomly selected sequences of bits as catalytic centers for protein Synthetase. They have demonstrated an autocatalytic system that achieves stability and that self-organizes a coding system that is a start towards that seen in modern genetic systems. It remains to determine more fully specific constraints on stability in such a system, and to demonstrate how such a system can bootstrap itself to greater, biologically plausible levels of complexity.

One way to develop the model is to use practical protein sequences as starting point of the evolution simulation. A superior candidate for this purpose could be the ancient proteins such as Aminoacityl-tRNA Synthetases. The catalytic domain of these sequences is a good catch for this purpose since it is in particular responsible for charging their cognate tRNA with the amino acid that will subsequently be incorporated on the ribosome into the growing protein chain or in general catalyzing the protein synthetic machinery. At this time, a question that comes to mind is how to find/predict these catalytic domain sequences?

The 3D structures of the catalytic domain in Aminoacityl-tRNA Synthetase sequences are highly conserved perhaps due to their specific function in different organisms. The phylogenetic history of the Aminoacityl-tRNA Synthetase structures would help us to find the early plausible sequences for catalytic domain.

Most common methods of phylogenetic analysis use only information derived from the sequences to build the tree. The null hypothesis about the evolution of these proteins states that many considered proteins are diverged before the last universal common ancestral state. They have evolved independently since then. They have a very low level of sequence identity. In fact, many of these enzymes within the same class of Aminoacityl-tRNA Synthetase have no more sequence identity than would be expected at random (8–10%).

In this work, protein 3D structure prediction methods and phyloinformatics applied in order to find evolution of the catalytic domain sequences of Aminoacityl-tRNA Synthetase. The problem with 3D structure prediction of Aminoacityl-tRNA Synthetases is the lack of experimentally determined structure in Protein Data Bank for structure templates. Phylogenetic tree reconstruction also applied to propose how the catalytic domain of these enzymes evolved which could address Genetic Coding evolution.

1.2 Objectives, scope and overview of the project

In this section, it is defined what this project will and what will not attempt to do in this research. It describes how the work has been broken into small steps and how the

[1] http://www.cs.auckland.ac.nz/careers/index.php/Computer_Simulation_Models_of_Prebiotic_Evolution_of_the_Genetic_Code

objectives of the project achieved by following the procedure of these steps. Figure 1.1 gives overview of this project.

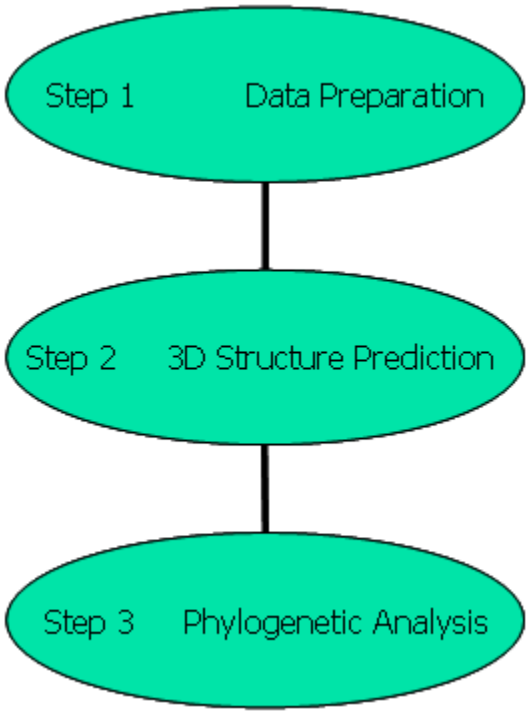

Figure 1-1 The Project Overview

1.2.1 Objectives

The main aim of the project is to propose the evolutionary view of Aminoacityl-tRNA Synthetases catalytic domains with 3D structure considerations by using different phylogenetic analysis methods. The catalytic domain might reflect the evolution of genetic coding because of its function. It is also aims to propose a combinational method for protein structure prediction for the sequences with very low sequence identity where there is structure conservation. Doing this project provide a survey the different methods of evolutionary analysis as well to study its strengths and weak points. The alignment of predicted catalytic domain could be used as initial sequences of amino acids that have catalytic functions in protein synthetic machinery for the work in Simulation Model of Prebiotic Evolution of Genetic Coding[2].

1.2.2 Scope and overview of the project

As shown in the Figure 1.1, this project is divided into three major steps. The first step relates to preparing the superposed structures of Aminoacityl-tRNA Synthetase catalytic domain. The common structural core of catalytic domains (Figure 3.2) is prepared by structurally aligning the catalytic domain of all non-redundant structures of experimentally determined of Aminoacityl-tRNA Synthetase from the Protein Data Bank (PDB) [20] in each class. In this step, it is also associated to collecting the Aminoacityl-

[2] http://bioinf.cs.auckland.ac.nz/index.php/sidney-markowitz/

tRNA Synthetase sequences classifying them by their taxonomic information. The second step is related to protein threading and modeling. It is also discussed about providing the Aminoacityl-tRNA Synthetase threading libraries. These threading libraries have been used in threading step to find a plausible template for each sequence. Subsequently, it is mentioned how these sequences has been modeled in a PDB formatted files containing the estimated coordination information of each atoms in amino acid sequences and finally how each of these PDB files aligned to the common structural core of catalytic domains (Figure 3.2) (conserved regions) provided in the first phase to predict the catalytic domains. The final phase is related to different phylogenetic analysis of catalytic domains, which has been extracted using multiple sequence alignment with structural considerations in the previous phase.

2 Background

2.1 An overview of Aminoacityl-tRNA Synthetase

Aminoacityl-tRNA Synthetases are among ancient proteins and would be among the first ones that have been evolved. In evolutionary prospective there are two distinct class of Aminoacityl-tRNA Synthetase. Common characteristic domain structures and sequence homologies define each class, but the two have nothing in common except the biochemistry of the reactions they catalyze. Between the two classes, proteins show no structural resemblance, have almost no common motifs. Two classes of Aminoacityl-tRNA Synthetase might reflect a bifurcated origin of translation itself, which are the results of fusion between two different primitive processes or the two classes are the surviving traces of an ancient evolutionary battle between emerging tRNA-charging mechanisms as biology evolved beyond the RNA world [1]. Figure 2.1 shows samples from different structure motifs in both classes of Aminoacityl-tRNA synthetases. On the right, the PDB entry is shown, 1ASZ, Aspartyl-tRNA Synthetase from class II together with tRNA. On the left is a sample from class I, 1GAX, Valyl-tRNA Synthetase together with tRNA.

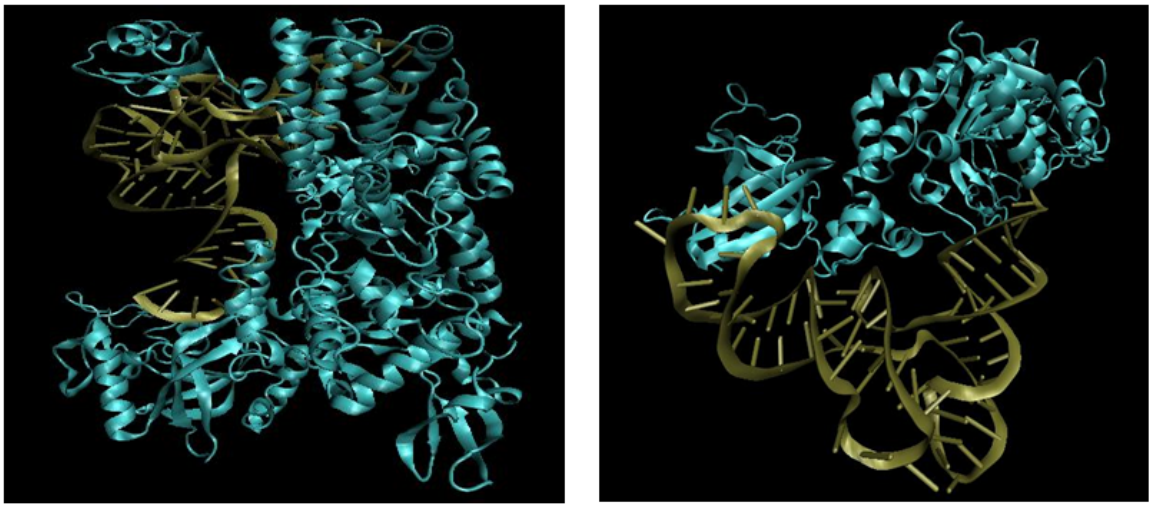

Figure 2-1 Samples from different structure motifs in both classes of Aminoacityl-tRNA synthetases. On the right, the PDB entry, 1ASZ, Aspartyl-tRNA Synthetase from class II together with tRNA is shown. On the left, the PDB entry, 1GAX, Valyl-tRNA Synthetase from class I together with tRNA.

The aminoacyl-tRNA synthetases are distributed between the classes according to specific rules. There are 10 amino acid's enzymes out of 20 in each class. The so-called "class rule" suggests that examples of a given amino acid's synthetase are of the same class. The "monophyletic" rule proposes that within a class, all synthetases associated with a given amino acid are specifically related to one another to the exclusion of the Aminoacityl-tRNA Synthetases associated with any other amino acids. A third rule, independent to the class of the enzyme for each organism, all tRNAs assigned to a given

amino acid (so-called isoacceptors) can be charged by a single synthetase, a rule that holds even for amino acids such as serine with two distinct sets of codons. [1]

2.2 Protein 3D Structure Prediction

Protein structure prediction is one of the important objectives pursued by bioinformaticians. It aims to predict the tertiary structure (3D structure) based on primary structure (Sequence). The performance of current methods is assessed in the CASP experiment. There are a lot of world wide efforts going on in structural genomics however the output of experimentally determined protein structures using X-ray crystallography or NMR spectroscopy is lagging far behind the output of protein sequences and on the other hands these methods are typically time-consuming and relatively expensive.

This attracts scientists to look for computational methods for predictions but too many parameters and factors exist in the problem of sequence-structure mapping that make protein structure prediction a very difficult and computationally intensive task. Since the number of possible protein structures is extremely large, and that the physical basis of protein structural stability is not fully understood, any protein structure prediction method needs a way to explore the space of possible structures efficiently (a search strategy), and a way to identify the most plausible structure something like energy function.

Comparative modeling approaches identify a known structure that is predicted to be similar. It assumes that there is a good chance that a protein that with a similar structure to the target protein has already been studied. It predicts structurally conserved regions, and locations of insertions and deletions (sometimes called ‘‘indels’’). Generally, comparative methods build model backbone structure, then copy predicted conserved main chain regions from template structure, and remodel loops with insertions or deletions. Then it adds side chains to the modeled main chain. In the end, they evaluate and refine model based on some criteria. In the following subsections, some methods of modeling are discussed.

2.2.1 Protein threading (Fold recognition)

Since protein structure and function are more conserved than protein sequence, the identification of correspondences between novel sequences and known structures would greatly assist in the characterization of these sequences. A huge field, known as fold recognition, has developed to tackle this problem. There are many approaches, but the unifying theme is to try to find folds that are compatible with a particular sequence. Unlike sequence-only comparison, these methods take advantage of the extra information made available from known 3D structures. In effect, this turns the protein-folding problem on it is head: rather than predicting how a sequence will fold, they predict how well a structure will fit a sequence.

2.2.2 Homology Modeling

Homology modeling involves taking a known sequence with an unknown structure and mapping it onto a known structure of one or several similar (homologous) proteins. It would be expected that two proteins of similar evolutionary origin and function would have reasonable similar structure. Therefore, it is possible to use the known structure as a template for modeling the structure of the target protein. All homology-modeling approaches consist of three steps:

1-Finding PDB files that contains homologous structures

2-Constructing an alignment, using pairwise or multiple sequence alignments (if more than one known structure is involved, sometimes the known structures are aligned together, then the unknown sequence aligned with the group; this helps ensure a better alignment. During alignment process, gap opening, gap extension, and secondary structure weighting is introducing.

3-Structure calculation and model refinement

The quality of the homology model is dependent on the quality of the sequence alignment and template structure. Homology modeling can produce high-quality structural models when the target and template are closely related, which has inspired the formation of a structural genomics consortium dedicated to the production of representative experimental structures for all classes of protein structures [2].

2.3 Structure superposition

The need for fast and accurate structure comparison algorithms has become more and more crucial. In general, there are two types of comparisons for protein tertiary structures. The first is to compare protein structures/models with well defined equivalence between pairs of residues (such equivalence can be provided by sequence or threading algorithms, for example). The most commonly used metric in this category is the root-mean-square distance, RMSD, in which the root-mean-square distance between corresponding residues is calculated after an optimal transformation of one structure to another. Since the RMSD weights the distances between all residues pairs equally, a small number of local structural deviations could result in a high RMSD, even when the global topologies of the compared structures are similar. The RMSD is rendered the unweighted distances between all residue-pairs, if there were small number of residue-pairs with large distance (local structural deviations) then this small part of structure could affect RMSD even when the global topologies of the compared structures are similar. The recently proposed TM-score [12] overcomes these problems by exploiting a variation weighting factors that weights the residue pairs at smaller distances relatively stronger than those at larger distances. Therefore, the TM-score is more sensitive to the global topology than to the local structural variations. Moreover, the value of the TM-score is normalized in a way that the score magnitude relative to random structures is not dependent on the protein's size, with a value of 0.17 for an average pair of randomly related structures.

The second type of structure comparison compares a pair of structures where the alignment between equivalent residues is not a priori given. Different approaches have been proposed in this area. For example in Distance-matrix ALIgnment, DALI [3] the idea is to construct a distance matrix using alpha carbon atoms of the main chain. When two proteins' distance matrices share the same or similar features in approximately the same positions, they can be said to have similar folds with similar-length loops connecting their secondary structure elements. DALI's actual alignment process requires a similarity search after the two proteins' distance matrices are built; this is normally conducted via a series of overlapping sub-matrices. Sub-matrix matches are then reassembled into a final alignment via a standard score-maximization algorithm. Other examples of approaches for structure comparisons are STRUCTAL [26] and SAL [27].

Profile alignments could be applied in comparative modeling. The accuracy of an alignment in comparative modeling between two protein sequences can be improved by including other detectably related sequences in the comparison. Some different protocols for creating and comparing profiles corresponding to the multiple sequence alignments have been proposed in some approaches e.g. SALIGN command in MODELLER [2]. This is discussed in detail in section 4.2.1.

2.4 Phylogenetic Analysis

The basic idea in phylogenetics is to study evolutionary history of the taxa e.g. organism, proteins, sequence, structures or any kind of molecular complexes based on similarity. Evolution is not always discrete with clearly defined boundaries that identify the origin of a new species [17]. The concept of similarity is the criterion for classification. In general, the result of phylogenetic analysis is a tree structure diagram, which depicts the hypothetical phylogeny of the taxa under consideration. In dealing with phylogenetics analysis of genetic data and/or amino acid sequences, there are some methods that are used for constructing the tree.

- **UPGMA**

A simple but popular clustering algorithm for distance data is Unweighted Pair Group Method using Arithmetic averages (UPGMA) [4], [5]. This method works by initially having all sequences in separate clusters and continuously joining these clusters together. The tree is constructed by considering all initial clusters as leaf nodes in the tree, and each time two clusters are joined, a node is added to the tree as the parent of the two chosen nodes. The clusters to be joined are chosen as those with minimal pairwise distance. The branch lengths are set corresponding to the distance between clusters, which is calculated as the average distance between pairs of sequences in each cluster. The algorithm assumes that the distance data has the so-called molecular clock property i.e. the divergences of sequences occur at the same constant rate at all parts of the tree. This means that the leaves of UPGMA trees all line up at the extant sequences and that a root is estimated as part of the procedure.

- **Neighbour Joining**

The neighbor joining algorithm [6], on the other hand, builds a tree where the evolutionary rates are free to differ in different lineages, i.e., the tree does not have a particular root. Some programs always draw trees with roots for practical reasons, but for neighbor joining trees, no particular biological hypothesis is nominated by the placement of the root. The method works very much like UPGMA. The main difference is that instead of using pairwise distance, this method subtracts the distance to all other nodes from the pairwise distance. This is done to take care of situations where the two closest nodes are not neighbors in the "real" tree. The neighbor joining algorithm is generally considered fine and is widely used. Algorithms that improve its cubic time performance exist. The improvement is only significant for quite large datasets.

- **Maximum Likelihood**

Maximum likelihood is probabilistic methods of inference. Both have the pleasing properties of using explicit models of molecular evolution and allowing for rigorous statistical inference. However, both approaches are very computer intensive. A stochastic model of molecular evolution is used to assign a probability (likelihood) to each phylogeny, given the sequence data of the operational taxonomical units (OTUs). Maximum likelihood inference [7] then consists of finding the tree that assigns the highest probability to the alignment.

- **Bayesian Inference**

The objective of Bayesian phylogenetic inference is not to infer a single "correct" phylogeny, but rather to obtain the full posterior probability distribution of all possible phylogenies. This is obtained by combining the likelihood and the prior probability distribution of evolutionary parameters. The vast number of possible trees means that Bayesian phylogenetics must be performed by approximate Monte Carlo based methods [8].

- **Bootstrapping**

A popular way of evaluating the reliability of an inferred phylogenetic tree is bootstrap analysis. The first step in a bootstrap analysis is to re-sample the alignment columns with replacement. For example, in the re-sampled alignment, a given column in the original alignment may occur two or more times, while some columns may not be represented in the new alignment at all. The re-sampled alignment represents an estimate of how a different set of sequences from the same genes and the same species may have evolved on the same tree. If a new tree reconstruction on the re-sampled alignment results in a tree similar to the original one, this increases the confidence in the original tree. If, on the other hand, the new tree looks very different, it means that the inferred tree is unreliable. By re-sampling a number of times, it is possibly to put reliability weights on each internal branch of the inferred tree. If the data was bootstrapped a 100 times, a bootstrap score of 100 means that the corresponding branch occurs in all 100 trees made from re-sampled alignments. Thus, a high bootstrap support score is a sign of greater reliability.

2.5 Available tools and solutions

In this section, knowledge and ideas that have been established on this topic will be conveyed and their relative strengths and weaknesses on different part of the projects from the data preparation, 3D structure prediction to the multiple sequence alignment with structural considerations – discussed in section 5.2.1- and phylogenetic analysis.

Let us start with 3D structure prediction methods and tools that will be used in the methodologies of the project. In section 2.2, it mentioned why scientist are interested to use computational methods for predict the 3D structure of a molecule specially Proteins. It also mentioned about comparative modeling and in particular Homology Modeling as a method of 3D structure prediction. MODELLER is an available and suitable tool that used for homology or comparative modeling of protein three-dimensional structures [2]. The user provides an alignment of a sequence to be modeled with known related structures and MODELLER automatically calculates a model containing all non-hydrogen atoms. Therefore, by providing such alignment, the user is suggesting the similarity and homology of a sequence (target protein) in respect to other sequences (protein template).

Such alignment could be constructed using different tools however; Structure Alignment (SALIGN) library in MODELLER can help in this manner. SALIGN can be used to generate multiple protein structures/sequences alignments or to align two blocks of sequences/structures that are in memory. However, this method is still in development, and has not yet been fully benchmarked. As with any other alignment method, generated alignments should be assessed for quality.

Broadly classifying, three different types of protein alignment categories are tackled by SALIGN:

-Multiple structure alignments
-Aligning a structure block to a sequence block
-Multiple and pair-wise protein sequence alignment

Based on this alignment, MODELLER implements comparative protein structure modeling. The proposed model for target protein should then satisfy some verification tests by checking some criteria (i.e. spatial constraints of residues, bond torsion angles, etc. Such tasks include de novo modeling of loops in protein structures, optimization of various models of protein structure with respect to a flexibly defined objective function, protein multiple sequences alignment and/or structures, clustering, searching of sequence databases, comparison of protein structures. The quality of alignment has the most important effect on the result that is the coordinates of atoms in a Protein Data Bank (PDB) format. That is, suggesting the homologous proteins to the target sequence should be addressed very careful.

Threading is one of the very few methods available that can predict the structure for a protein in the absence of an evolutionary relationship. If the exolutionary

relationships between target and template sequences are known, i.e. it is already known, which protein(s) is closely related to the target, and then perhaps it is the better idea to use other methods such as a simple PSI-BLAST search. Threading is an approach to fold recognition that used a detailed 3-D representation of protein structure. Jones et al. 1992 idea is to "thread" a sequence of amino acid side chains onto a backbone structure (a fold) and to evaluate this proposed 3-D structure using a set of pair potentials and (importantly) a separate solvation potential. The program that implemented this method was called THREADER [18]. THREADER is a convenient program run with UNIX OS. The results could easily be viewed and modified with simple UNIX commands. However, threading could be very time consuming process depending on the sequence length and number of templates that, a target is going to be threaded against them (threading library).

One major problem with threading is how to eliminate false positive matches from true positives. In THREADER User's Manual, this method is addressed as following:

"…A very long practiced technique in basic sequence analysis is to compare the match scores for a real sequence with those generated by random sequences of the same length and amino acid composition. Most commonly, these random sequences are produced by shuffling the original sequence. By using Z-scores calculated by shuffling experiments, false positive matches can be identified, and the true positive matches that produce the most accurate alignments are highlighted. As the results, one can just simply rank the proteins threading results based on their Z-score and choose the most plausible ones as candidates of homologous to the target."

-Taken from THREADER User's Manual

In THREADER User's Manual,[3] it is suggested how to evaluate and interpret the z-scores as following:

"…Z > 4.0 Very significant - probably a correct prediction
Z > 3.5 Significant - good chance of being correct
2.7 < Z < 3.5 Borderline significant - possibly correct
2.0 < Z < 2.7 Poor score - could be right, but needs other confirmation
Z < 2.0 Very poor score - probably there are no suitable folds in the library
Z = -9.99 Too little of the sequence or template fold have been aligned."[4]

Another feature about THREADER, that become popular more or less in other programs e.g. MODELLER is the new way to identify structures. The new identifiers are always six characters long e.g. 2fb4H1. This is interpreted as Domain 1 of Chain H of PDB entry 2FB4. Where no domains are defined, a default domain identifier of '0' is used. For example, the code 1mbd00 refers to the entire PDB entry 1MBD (sperm whale myoglobin) which has a single chain and no structural domains. Figure 2.2 demonstrates the description of the proposed by Jones et al. 1992 [18].

[3] http://bioinf.cs.ucl.ac.uk/downloads/threader/manual.pdf
[4] *Taken from THREADER User's Manual*

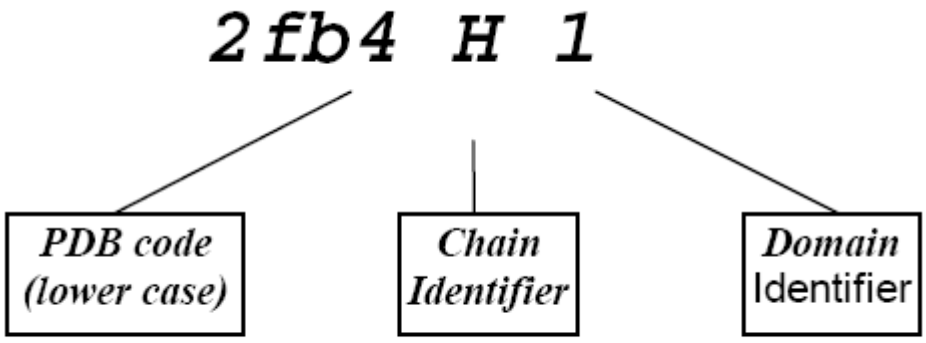

Figure 2-2 The New identifier for referring to chains and domains in a PDB file proposed by Jones et al 1992 [18]. The diagram is from the THREADER 3 User's Manual

THREADER has the benefit of simple user manual that describes how to work with the package and how to use THREADER in a research purpose.

Let us discuss a little bit about the other subject related to the projects. The use of nucleotide sequence differences in a single gene to investigate evolutionary relationships was first widely applied by Woese and Fox [24]. They recognized that sequence differences in a conserved gene, ribosomal RNA, could be used to infer phylogenetic relationships. The tree of life elucidated by Woese is noteworthy for its demonstration of the overwhelming diversity of microbial lineages; single-celled organisms represent the vast majority of the biospheres genetic, metabolic, and ecosystem diversity. Woese divided life into 23 main divisions based upon genetic relationships rather than obvious morphological similarities. All branches incorporated within three domains: Bacteria, Archaea, and Eukarya. Archaea are neither Bacteria nor Eukaryotes. In other words, Archaea are Prokaryotes that are not Bacteria [9]. He also looked at the evolution of Aminoacityl-tRNA Synthetase enzymes as a superb indicator of the evolutionary dynamics in general [1].

The evolutionary picture painted by the synthetases, however, is a world apart from this canonical pattern. Not only do the phylogenies fail to yield the canonical pattern in a number of cases, but also they typically violate the accepted taxonomic structure within the organism domains.

Furthermore, the molecular phylogenies inferred from the synthetases of different amino acid types tend not to agree with one another. However, this is the telling point that Aminoacityl-tRNA Synthetases are in essence modular components of the cell; they function in isolation from the rest of the translation apparatus and from the rest of the cell, except for their individual contacts in each case with a small subset of the tRNAs. Because of this and because of their universality, the Aminoacityl-tRNA Synthetases can function in a wide spectrum of cellular environments, often without disadvantage to the host. In other words, the Aminoacityl-tRNA Synthetases are ideal candidates for widespread horizontal gene transfer [1].

In Woese's work [1], the evolutionary profile of each 20 Aminoacityl-tRNA Synthetase has been examined to determine in which Aminoacityl-tRNA Synthetase profile conforms the canonical phylogenetic pattern. He looked at different organisms and the sequence similarity of these organisms in each Aminoacityl-tRNA Synthetase profile.

O'Donoghue et al. [10] also examined the Aminoacityl-tRNA Synthetases [10]. They looked at the structural conservation of Aminoacityl-tRNA Synthetase in respect to their sequence similarity. In order to investigate the structural evolutionary relationships between the Aminoacityl-tRNA Synthetases, they applied the unweighted pair group method with arithmetic averages (UPGMA) for clustering analysis. This method performs agglomerative clustering based on a pairwise distance measure that can be represented as a tree. Since the sequence identity distribution is tightly peaked near 10%, this measure is inappropriate for constructing trees of such distantly related proteins. Instead, the structural homology measure Q_H (Quality of Homology) was used as a pairwise similarity measure [10].

The STAMP algorithm has been used for constructing a multiple alignment based on the pairwise structural alignments of a set of Aminoacityl-tRNA Synthetases [11]. The Q_H measure is designed to include the effects of the gaps on the aligned portion: $Q_H = \$(q_{aln}+q_{gap})$, where \$ is the normalization form and q_{aln} account for the structurally aligned regions. The q_{gap} term accounts for the structural deviations induced by insertions in each protein in an aligned pair of structures. q_{aln}, computes the un-normalized fraction of C_{alpha} -C_{alpha} pair distances that are the same or similar between two aligned structures. Q_H ranges from 0 to 1 where $Q_H = 1$ refers to identical proteins. If there are no gaps in the alignment, then Q_H becomes Qaln = \$ q_{aln}.

The STAMP algorithm for protein alignment used in VMD [11] uses RMSD. Since the RMSD weights the distances between all residues pairs equally, a small number of local structural deviations could result in a high RMSD, even when the global topologies of the compared structures are similar. Furthermore, the average RMSD of randomly related proteins depends on the length of compared structures, which renders the absolute magnitude of RMSD meaningless. This can be problematic in dealing with Aminoacityl-tRNA Synthetases because the different types of Aminoacityl-tRNA Synthetase normally have different structure also the structures might be varied in the same type of Aminoacityl-tRNA Synthetase but in different organisms.

Zheng et al. [12] showed this problem using aligning two Aminoacityl-tRNA Synthetase structures. They show a typical example of a structural comparison between 1ATZ chain A and 1AU0 chain A, which have a sequence identity of 16% and share a similar alpha-beta-alpha sandwich fold. While 1ATZ chain A has five beta-strands and three alpha helices on each side, 1AU0 chain A has seven beta-strands in the middle and two alpha helices in the left and four alpha helices in the right side, shown in Figure 2.3. The latter has also a unique long beta-turn on the right side. An ideal structure alignment, therefore, should match two alpha helices on the left, five beta-strands in the middle and three alpha helices on the right side of the two structures. They proposed a method of aligning the structures called TM-Align. The algorithm overcomes these problems

(having regions small pairs of residues with large distance that will affect goodness of fitness value) by exploiting a variation weighting factors that weights the residue pairs at smaller distances relatively stronger than those at larger distances. Therefore, the TM-score is more sensitive to the global topology than to the local structural variations. Moreover, the value of the TM-score is normalized in a way that the score magnitude relative to random structures is not dependent on the protein's size, with a value of 0.17 for an average pair of randomly related structures. The alignment algorithm works much faster as well [12]. Figure 2.3 illustrates the values and the results corresponding to the problem.

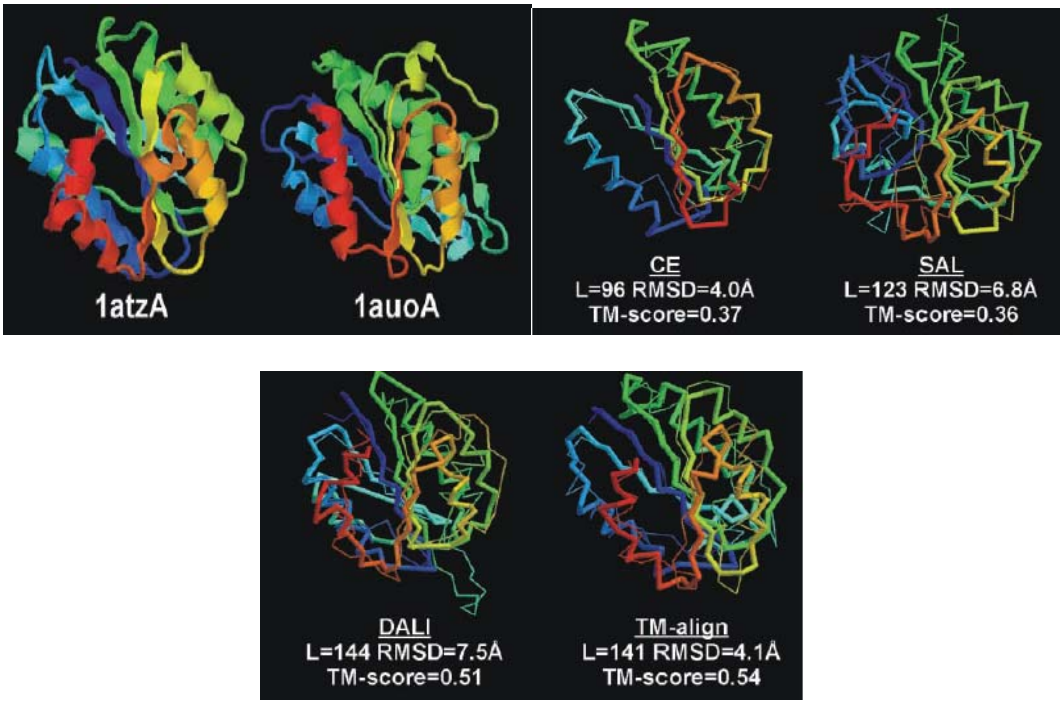

Figure 2-3 Example from Zheng et al. [12] regarding structure alignments by different alignment methods for 1atzA and 1auoA. The first row is the ribbon diagram of the native structures of 1atzA (184 residues) and 1auoA (218 residues), which have a sequence identity 16% and adopt the common alpha-beta-alpha sandwich topology. The second and third rows are the structure superposition between the aligned residues by CE [13] and SAL [14], DALI [15] and TM-align algorithms, respectively. The thick and thin backbones denote the aligned residues from 1atzA and 1auoA, respectively. The indicated numbers are the length of aligned residues, the RMSD between the aligned residues, and the TM-score normalized by the length of 1atzA. All the pictures are generated by RASMOL (http://www.umass.edu/microbio/rasmol) with blue to red running from the N- to C-terminus [12]. The figure has been taken from [12]

Gille [21] provided the STRAP package, which is an editor for in STructural Alignment of Proteins. STRAP package makes use of different protein alignment algorithms such as TM-Align. STRAP is a comfortable and extensible tool for the generation and refinement of multiple alignments of protein sequences. A wide range of functions related to protein sequences and protein structures are accessible with an intuitive graphical interface. Recent features include mapping of mutations and polymorphisms onto structures and production of high quality figures for publication.

STRAP is developed in Java™, which makes it portable. It uses simple to use interface called KISS that aims to Keep the user Interface for STRAP Simple. KISS renders STRAP extendable to bioscientists as well as to Bioinformaticians. Scientists with basic computer skills are capable of implementing statistical methods or embedding existing bioinformatical tools in STRAP by themselves. There are implemented functions for exporting the generated multiple sequence alignments in the convenient formats such as FASTA. Therefore, it seems to be a good candidate when we deal with generating, editing and exporting protein structure alignment. This is exactly what it is needed to generate multiple sequence alignment for the final step of the project (phylogenetic analysis). Essentially the tree results are based on this multiple alignment.

PAUP*[5] is a tool for inferring and interpreting evolutionary trees by David Swofford. There are many features in this package for evolutionary analysis. There are various methods for estimating the distance matrix using different type of empirical substitutions matrices. It is possible to construct the evolutionary trees using neighbor joining, bootstrapping or even complex methods, such as Maximum Likelihood (ML) estimation.

By using Model Test[6], it is possible to find out which type of substitution matrix fits the best to your alignment in order to use Maximum Likelihood methods. Although the ML methods could be quite slow on amino acids sequences considering 20*20 substitution matrix on large number of taxa.

For large number of taxa on solution is to use PhyML[7] by Guindon et al. [16] that is a simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood [16]. When very accurate branch lengths are not in consideration of the evolutionary analysis then one possibility is to use GARLI. It is a work of Derrick J. Zwickl. It is fast because of using Genetic Algorithm for estimating the branch length of the tree.

GARLI[8] performs phylogenetic searches on aligned nucleotide datasets using the maximum likelihood criterion. Available models of nucleotide substitution include the General Time Reversible (GTR) model and its more common substitution models. Gamma distributed rate heterogeneity (with a specified number of rate categories) and estimation of the proportion of invariable sites is included. GARLI uses a genetic algorithm approach to find the tree topology, branch lengths and model parameters that maximize the Log-likelihood probability (lnL) simultaneously. This involves the evolution of a population of solutions termed individuals, with each individual encoding a tree topology, a set of branch lengths and a set of model parameters. Each individual is assigned a fitness based on its lnL score. In each generation, random mutations are applied to some of the components of the individuals, and their fitnesses are recalculated.

[5] http://paup.csit.fsu.edu/
[6] http://darwin.uvigo.es/software/modeltest.html
[7] http://atgc.lirmm.fr/phyml/
[8] http://www.zo.utexas.edu/faculty/antisense/garli/Garli.html

The individuals are then chosen to be the parents of the individuals of the next generation, in proportion to their fitnesses.

This process is repeated many times, and the population of individuals evolves toward higher fitness solutions. Note that the highest fitness individual is automatically maintained in the population, ensuring that it is not lost due to chance (Genetic Drift). The mutation types used by GARLI are divided into three types: topological mutations, model parameter mutations and branch-length mutations.

Topological mutations consist of the standard nearest neighbor interchange (NNI) and stepwise addition plus subtree pruning regrafting (SPR) rearrangement types, as well as a localized form of SPR in which the pruned clade may only be reattached to branches within a certain radius of its former location. Topological mutations are followed by some degree of rough branch-length optimization. Model mutations simply choose one of the model parameters and multiply it by a gamma-distributed variable with mean 1.0. When branch-length mutations are performed, a number of branches are chosen and each has its current length multiplied by a different gamma-distributed variable.

GARLI can read either PHYLIP or Nexus formatted datasets. It is very easy to tell GARLI what to do or in other words to configure it. All directions to the program are provided through a text-based configuration file, which by default is named “garli.conf”. It is just needed to open this file in a text editor to make the configuration changes. In this file, all the specifications will be set. These specifications are for example the dataset, file name and the parameters such as the initial topology (random or from an existing tree file), empirical substitution matrix(WAG, mtREV, Dayhoff, JTT, VT, Blosum62, CpREV, RtREV, MtMam, MtArt, HIVb, and HIVw) population of individuals specifications and number of generations.

The problem that might come up with maximum likelihood tree search is that after running from different initial tree topology, we might not end with the same likelihood score that means the trees are not converging to a consensus tree topology. It might also end up with relatively similar log-likelihood scores but different topologies. So that, the tree space for such alignment suggests different trees with the same probability to the alignment or the problem might suggest that the tree space has. This is perhaps because of short sequence lengths that are not sufficient for suggesting a unique tree with the highest log-likelihood.

The solution for this problem is to try to find the posterior distribution of the tree space based on Bayesian Inference to get the idea whether the trees are converging to a consensus with maximum credibility (the best likelihood probability) or not. This is the idea behind Bayesian Evolutionary Analysis Sampling Trees or BEAST by Drummond and Rambaut [17]. BEAST uses Monte Carlo Markov Chain (MCMC) to average over tree space, so that each tree is weighted proportional to its posterior probability. It includes a simple to use user-interface program for setting up standard analyses and a suite of programs for analyzing the results. BEAUti (Bayesian Evolutionary Analysis Utility) is a graphical software package that allows the creation of BEAST XML input

files. The exact instructions for running BEAUti differ depending on which computer system you are operating. For further information how to work with BEAUti/BEAST package, you might want to refer to BEAST home page[9]

However, like GARLI it is possible to specify some parameters for evolutionary analysis. It would be useful to point out a few of parameters description that were considered in order to evaluate the results based on them. They have been taken from BEAST user's manual available in the BEAST homepage and mentioned in Appendix C.

[9] http://beast.bio.ed.ac.uk/

3 Data Preparation

In this chapter, the materials, methods and results for the first step are mentioned. Prior to this, the programming languages used in this project are listed.

3.1 Programming languages

Perl was used for all kinds of text modification and pattern matching in sequences. It is used for extracting data from THREADER and the sequences of catalytic domains of Aminoacityl-tRNA Synthetase that is discussed in the section 5.2.1. Python was used for configuring the MODELLER in order to read the results from THREADER and predict a PDB formatted files for sequences of Aminoacityl-tRNA Synthetase.

Java™'s platform is used to make a customized protein aligner using the libraries of STRAP package in a way to construct a multiple sequence alignment from a predicted structure of an Aminoacityl-tRNA Synthetase sequence (result from MODELLER) against the superposed catalytic domains of known Aminoacityl-tRNA Synthetase structures. Using convenient shells in UNIX based operating systems; it was possible to write scripts that performs the similar tasks for each sequence in turn for several thousand of files. These shell scripts have been used in many steps in the project such as threading, 3D structure modeling, sequence/structure alignment and eventually running BEAST for constructing the trees

3.2 Materials

Here is the list of tools and databases have been used to construct the superposed catalytic domains of known Aminoacityl-tRNA Synthetase structures. The data gained in from these step, then have been used in the next steps of the project. The entries from all Aminoacityl-tRNA Synthetase experimentally known structures available in Protein Data Bank PDB format have been used to construct these common structural core of catalytic domains (Figure 3.2) that later will be used in finding the catalytic domain of the predicted structure models.

For finding the catalytic domain of the known structures, Visual Molecular Dynamics (VMD with Multiseq 2.0 package) is used. The Multiseq 2.0 package has been also used in O'Donoghue's and Luthey-Schulten's work on studying Aminoacityl-tRNA Synthetase evolution [10]. VMD is designed for the visualization and analysis of biological systems such as proteins, nucleic acids, lipid bi-layer assemblies, etc. It may be used to view more any molecules. VMD can read standard Protein Data Bank (PDB) files and visualizes the contained structure.

In particular, VMD[10] can act as a graphical front end for an external Molecular Dynamics program by displaying and animating a molecule undergoing simulation on a remote computer. MultiSeq 2.0 plug-in is a major extension of the Multiple Alignment tool that is provided as part of VMD, a structural visualization program for analyzing molecular dynamics simulations. Both are freely distributed by the NIH Resource for Macromolecular Modeling and Bioinformatics and MultiSeq is included with VMD

[10] http://www.ks.uiuc.edu/Research/vmd/

starting with version 1.8.5. Sequences are downloaded from AMINOACITYL-TRNA SYNTHETASE database [19] and NCBI Entrez Nucleotide database [11][22].

3.3 Methods

3.3.1 How to prepare the sequences and the materials

Two different motifs or classes for Aminoacityl-tRNA Synthetase have been introduced in previous works [1], [10]. The project started with providing the annotated/categorized sequences as Aminoacityl-tRNA Synthetase types (for example Ala-RS, Gly-RS, Ser-RS, etc.). The sequences of Aminoacityl-tRNA Synthetase with unknown structure were downloaded from the protein sequence database in Aminoacityl-tRNA Synthetase database[12] [19]. However, almost all Aminoacityl-tRNA Synthetase sequences in the database are from bacteria and Archaea. Therefore, protein sequence databases have been searched another time to obtain some sequences from Eukaryotic organisms.

The sequences in these databases are not annotated by their taxonomy information. It makes two problems; first is that the taxonomy of each sequence is point of interest in order to make the final tree analysis based on diversity of organisms. The second problem is that Lys-RS are the only one among the Aminoacityl-tRNA Synthetase types that is found in both motifs therefore making decision by certain that based on which structural motif (class I or II), the comparative modeling should be performed. In the end, Lys-RS sequences were decided to add in to the both classes of structures and let the poor THREADING results give me the hint that each sequence belongs to the class I or to the class II.

The sequences are in FASTA format. The file names are the accession number in nucleotide database that is like NP_XXXXXX where X represents a number. One way to annotate the file is to use this accession number and use fastacmd tool in BLAST package from NCBI, which is a part of BLAST search package. “Fastacmd” has a “-T” option that helps the user to retrieving taxonomic information for a given sequence:

$>fastacmd -d nt -s 555 -T
NCBI sequence id: gi|555|emb|X65215.1|BTMISATN
NCBI taxonomy id: 9913
Common name: cow
Scientific name: Bos Taurus

Using this feature and Perl script, the sequence files were renamed with the following format:

“[Aminoacityl-tRNA Synthetase type]_[Accesstion_No]_[Name_Species]”
For example, *“Met-RS-NP_310947-Escherichia_coli.fasta*”.

[11] http://www.ncbi.nlm.nih.gov/sites/entrez?db=nucleotide
[12] http://rose.man.poznan.pl/Aminoacityl-tRNA Synthetase/

It was also essential to convert “fasta” formatted files to “pir” format in order to use them for 3D structure prediction tool (MODELLER). A sequence in PIR format consists of:

1- **One line** starting with
2- A ">" (greater-than) sign, followed by
3- A **two-letter code** describing the sequence type (P1, F1, DL, DC, RL, RC, or XX), followed by
4- A **semicolon**, followed by
5- The sequence **identification code** (the database ID-code).
6- **One line** containing a textual description of the sequence.
7- **One or more lines** contain the sequence itself. The **end of the sequence** is marked by a "*****" (asterisk) character.

A file in PIR format may comprise more than one sequence however just each files consists of just one sequence because the code script for using MODELLER accepts one sequence to do the structure prediction every time it runs. The entire sequences corresponding to each type of Aminoacityl-tRNA Synthetase are in a single file in the polish database so one needs to separate them as well. The PIR format is also often referred to as the NBRF format.

3.3.2 How to find common structural core of catalytic domains of known Aminoacityl-tRNA Synthetase structures

In order to find a common structural core of catalytic domain of known Aminoacityl-tRNA Synthetase structures, at first it is essential to download their PDB files. The entire files were downloaded using a Perl script that calls a Linux bash command “wget” to retrieve a file from Protein Data Bank’s FTP server[13]. The lists of downloaded PDB files for both classes are given in Appendix D.

The next step is to discard the redundant backbone structures from the set of experimentally determined 3D structures for Aminoacityl-tRNA Synthetase. In many downloaded PDB files, there are often identical chains e.g. Chain A, B, C, etc. that all correspond to the same Aminoacityl-tRNA Synthetase complex structure and all of these chains contain the catalytic domain. The identical chains were discarded for each PDB file. This is done by looking at different chains in each PDB files and superposing these chains using the STAMP algorithm in VMD multiseq 2.0. After superposing, the quality of homology (Q_H) value was considered to assess whether they were identical structures or not. If they were identical then just one of them was kept in the library of known structures. These structures were saved as separated PDB files named with the THREADER Identification naming format mentioned in section 2.5. By now, a list of non-redundant PDB files is provided for each class. These are listed in Appendix D.

[13] ftp://ftp.rcsb.org/pub/pdb/data/structures/all/pdb/

Subsequently, all corresponding known structures in class I where grouped together in one category and those for class II in a different category. In both groups, the structures were aligned using VMD with multiseq 2.0 plug-in that contains the STAMP algorithm for structure alignment. STAMP algorithm was used for superposing the structures all together. This task could be time consuming. Most of the time, when all group of different types of Aminoacityl-tRNA Synthetases (although they are from the same class) are going to be superposed in VMD, the process will not be successful and often give an error because they are not similar structures. However, because it is assumed that the structure of the catalytic domain is conserved, I decided to look at structures of each type and truncate the non-conserved parts in order to find the structure of catalytic domain. Again, the Q_H is a good clue to find such conserved regions (catalytic domains). At first, the truncation was done on structures from just one type of Aminoacityl-tRNA Synthetase to find the conserved regions of that type. Then all structures related to another type of Aminoacityl-tRNA Synthetases was added to the previous common structural core and again superposition and truncation were performed on this set to find conserved regions to these two types. This iterative process lasts until all structures from all types of Aminoacityl-tRNA Synthetase are added, superposed, and truncated. This iterative process was done for all non-redundant structures of each class separately. The output of this process represents the common structural core of catalytic domain of Aminoacityl-tRNA Synthetases. At this point, the profiles of Aminoacityl-tRNA Synthetase catalytic domains are available for both classes. In section 3.4, the screenshots illustrating the catalytic domain of two classes will be presented (Figure 3.2).

3.4 Results

Table 3.1 summarizes the sequences frequency for each type of Aminoacityl-tRNA Synthetase downloaded from the databases [19]. The sequences are mostly from Bacteria. However, having done more searches in NCBI Entrez Nucleotide database [22] it is managed to find 93 sequences of eukaryotes including modern organisms such as insects, animals and *Homo Sapiens*. These 93 sequences are for all types of Aminoacityl-tRNA Synthetase and in both classes.

By August 2007, having removed the identical structures from the PDB files of known structures, I managed to find 25 non-redundant structures associated with class I and 30 structures for class II to be superposed on each other. By using VMD MultiSeq 2.0, a superposition of these catalytic domains of known Aminoacityl-tRNA Synthetase structures was constructed. Then, the common structural core in the catalytic domains was used for multiple sequence alignment of catalytic domains of Aminoacityl-tRNA Synthetases for predicted structures (discussed in section 5.2.1). See Appendix D for details.

Figure 3.1 shows a snapshot from VMD that illustrates the comparison between poor sequence identity and conserved structure in the catalytic domain among the Aminoacityl-tRNA Synthetases.

Table 3-1 the frequency of different Aminoacityl-tRNA Synthetase sequences. The class I Aminoacityl-tRNA Synthetases are highlighted in white and the class II are in Blue. Lys-RS could be in class I and II and highlighted as green

Type	Frequency
Ala-RS	153
Arg-RS	138
Asn-RS	13
Asp-RS	231
Cys-RS	152
Gln-RS	54
Glu-RS	148
Gly-RS	206
His-RS	167
Ile-RS	151
Leu-RS	150
Lys-RS	176
Met-RS	153
Phe-RS	296
Pro-RS	134
Ser-RS	151
Thr-RS	154
Trp-RS	153
Tyr-RS	150
Val-RS	138
TOTAL	3068
CLASS I	1563
CLASS II	1681

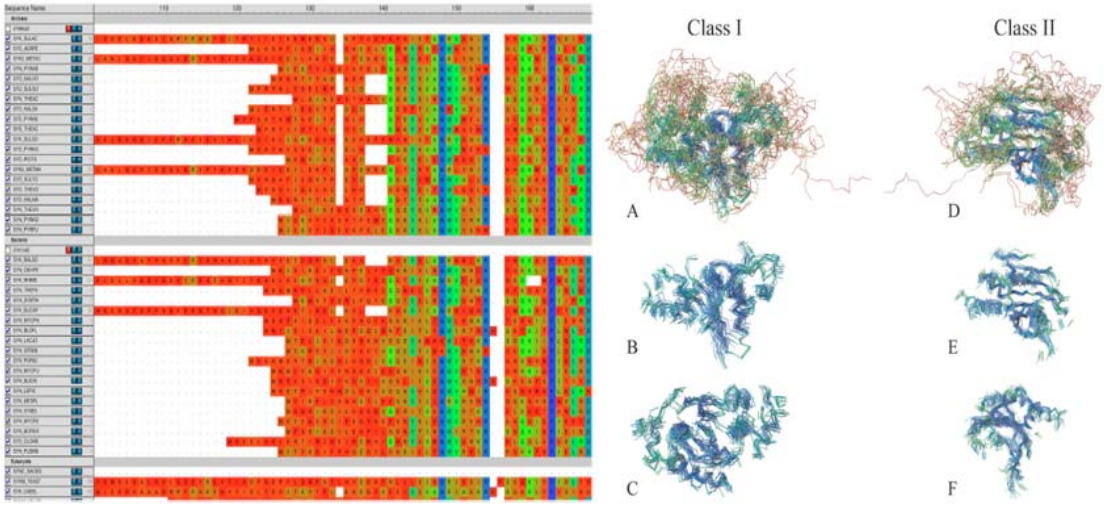

Figure 3-1 The sequence identity (left) vs. structure conservation (right). The picture on the right is form [10]

The structure conservation illustrated in Figure 3.1 motivates superimposition of the catalytic domains of known Aminoacityl-tRNA Synthetase structures using STAMP (Figure 3.2).

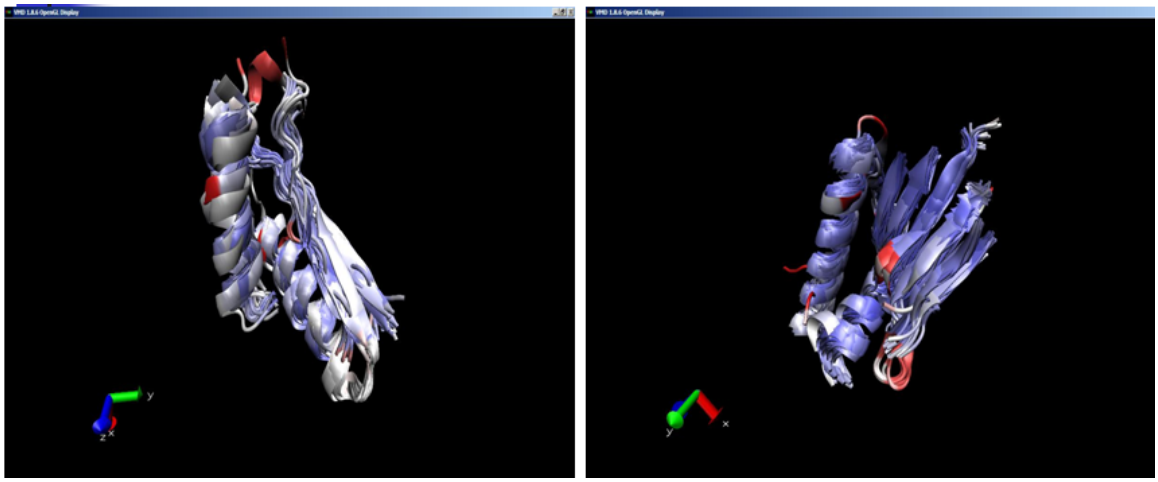

Figure 3-2 The superposition of conserved regions (superimposed catalytic domains of known Aminoacityl-tRNA Synthetase structures) in two classes of Aminoacityl-tRNA Synthetases. On the left, the catalytic domain of the experimentally determined Aminoacityl-tRNA Synthetase structures aligned using STAMP algorithm is shown and on the right is the same superposition of conserved region structures for class II. The structural motifs are suggesting different structural motifs.

The lengths of conserved regions in the catalytic domain of these known structures are relatively short. In class I, the length of the truncated structure that is shown in the Figure 3.2 (left) is 72 residues and for class II the length of the truncated conserved structures that is shown in Figure 3.2 (right) is 89 residues.

The short lengths of these conserved regions suggest that the alignment discussed in section 5.2.1 will also have the same lengths because the predicted structures will be superposed on these conserved regions in order to find the catalytic domain in the predicted models and extract the sequences of these regions. Perhaps they are so short for doing phylogenetic analysis. Due to low sequence identity, then there would be diversity of different aminoacids in each site or column in the alignment therefore the results interpretation and conclusions should be based on the assumption that the results are restricted to the catalytic domain not the whole Aminoacityl-tRNA Synthetase structures.

4 3D Structure Prediction

In this chapter, the materials, methods and results for the second step were pointed out. The results for modeling are many PDB files. Therefore, just the statistics were given instead of showing the files. Finally, the trees constructed by the different methods were shown.

4.1 Materials

THREADER [18] Version 3.5 was used in order to find modeling templates for a sequence with unknown structure. Subsequently MODELLER [2] applied. The MODELLER has been configured with Python codes. An aligner was developed in order to perform multiple sequence alignment with structural considerations by employing the libraries implemented in STRuctural Alignment Program (STRAP) [21] and BioJava.

4.2 Methods

4.2.1 How to predict the 3D structures of Aminoacityl-tRNA Synthetase sequences

This section describes how the results, from threading were combined with comparative modeling for prediction the 3D structure of Aminoacityl-tRNA Synthetase molecules from their sequences. As mentioned in section 1.1, the sequence identity among the Aminoacityl-tRNA Synthetase is low. This is the case even among the sequences from the same type. Because Homology modeling could only suggest a high-quality structural models when the target and template are closely related then using homology modeling on such datasets seems to be unrealistic.

Therefore, a question arises: that how to use MODELLER as a homology modeling tool when it is known not to have a clue about a homolog template for modeling a sequence with unknown structure. The answer is that to try to find the template by using another method called Threading that is used to recognize such templates from the library of structures in Protein Data Bank for a sequence where there is no evolutionary information available [18].

It is important to find out whether threading is the best choice for the application. If we are dealing with many conserved sequences that are show good correspondence to the structures, then generating models from threading is actually not as good a choice as simply generating sequence alignments with, e.g., BLAST and making homology models. In this project since the sequence identity with any known structure is low (see section 1.1), it is not possible to use comparative modeling directly.

That means THREADER tries to suggest a plausible structure among the known PDB structures to be template(s) for such sequence. More to the point is that THREADER is going to be applied to give such suggestion among the library of Aminoacityl-tRNA Synthetase structures that have been provided previously. However, the results of structures will bias to the set of known structures that had been downloaded

from PDB that are mostly from Bacteria. That means predicted structure for eukaryotes might be unrealistic and incorrect or even it might not be possible to model some of these sequences because THREADER could not find a template with a high Z score so it is hard to believe in the accuracy of the structure with a template that is not actually a good homolog template

THREADER was used to search through a subset of non-redundant structures of Aminoacityl-tRNA Synthetases that are provided before (see section 3.3.2). THREADER has its own library files called Threading Data Bank files (TDB), which are updated often and contained many structures. However, it is possible to make your own TDB file using the web service available THREADER home page[14]. This web service was used to build TDB files for all of the Aminoacityl-tRNA Synthetases structures listed in D.2 and D.4 in Appendix D. This set of TDB files were used as the threading library in this study.

After threading, the output file of THREADER were sorted (using the "sort" command in UNIX) by the z-score results and the top 10 template suggestions were saved in a file. Then using a Perl script, the z-scores were considered in order to select the template names in the six character format (section 2.5) only if they meet a certain z-score threshold. The selected template names were saved in a text file. Using UNIX shell features, this procedure was performed for all sequences in both classes. At this point, the template structure candidate(s) for doing homology modeling is available.

Now it is MODELLER's turn to do its job. It reads the list of templates from a text file, the associated PDB files of the templates and of course, a sequence of Aminoacityl-tRNA Synthetase of templates in PIR format as inputs. It provides the prediction in a PDB formatted file as output. It makes its prediction based on the templates and then evaluates them due to some evaluation criteria such as Free Energy, residues appear in the protein surface and which are the buried residues or quality of prediction based on the Ramachandran plot, etc.

A python program was written that applies the multiple template method for protein structure prediction. For more information about the multiple template method, please see MODELLER's homepage[15]. The code simply reads the template PDB files and then uses structure alignment function SALIGN multiple times (each time on a pairs of template structures) in order to generate an initial profile alignment and then improve upon it by using more information (sequences from other template(s)). The default values for the parameters such as gap penalties (e.g. gap penalty in sequence alignment and gap penalty in structure) were applied. For further information about different types of gap penalties is available at MODELLER's tutorial.

Using the python program, MODELLER tries to suggest (if possible) up to three 3D structure predictions for each target protein. It returns a score for each model called Discrete Optimized Protein Energy (DOPE) score [23]. DOPE score makes it easy to evaluate the suggestions by choosing the prediction with the best DOPE score.

[14] http://bioinf.cs.ucl.ac.uk/threader/maketdbform.html
[15] http://salilab.org/modeller/

DOPE is implemented in Python and is run within the MODELLER environment. The DOPE method is generally used to assess the quality of a structure model as a whole. Alternatively, DOPE can also generate a residue-by-residue energy profile for the input model, making it possible for the user to spot the problematic region in the structure model. Finally, a PDB files for the predicted (sequence with unknown structure) is provided as output.

4.2.2 Self-Testing of the 3D structure prediction accuracy

The accuracy of combining threading and homology modeling for predicting the 3D structure of these enzyme sequences has been self-tested on experimentally determined Aminoacityl-tRNA Synthetase structures. That means, a known structure of an Aminoacityl-tRNA Synthetase is taken and assumed that its structure is not known. Therefore, the sequence of this structure is considered as target (target = Aminoacityl-tRNA Synthetase with unknown structure). Then this sequence was threaded against the list of TDB libraries. It is obvious that the best hit will be actual name of this structure. Therefore, the best hit were discarded and the rest with z-score more than the threshold (>3) were considered because my purpose is to evaluate how precise a prediction could be using templates, other than itself.

The python code with MODELLER was used to predict the structure of this target using the threading results. In the end, I used VMD with Multiseq 2.0 to align the predicted structure and the real original one and see how similar these structures are. I had done this self-testing for all type of Aminoacityl-tRNA Synthetase known structures in both classes to see whether it works or not.

Figure 4.1 illustrates one example of self-testing from Class I and one from Class II. In the top left it shows the original structure of 1GAX, Valyl-tRNA Synthetase chain A. In the bottom left corner, it shows the structure of Class one Aminoacityl-tRNA Synthetase, 1GAX, Valyl-tRNA Synthetase chain A aligned to its predicted version. Figure 4.1 shows relatively accurate structure prediction for the tested enzymes.

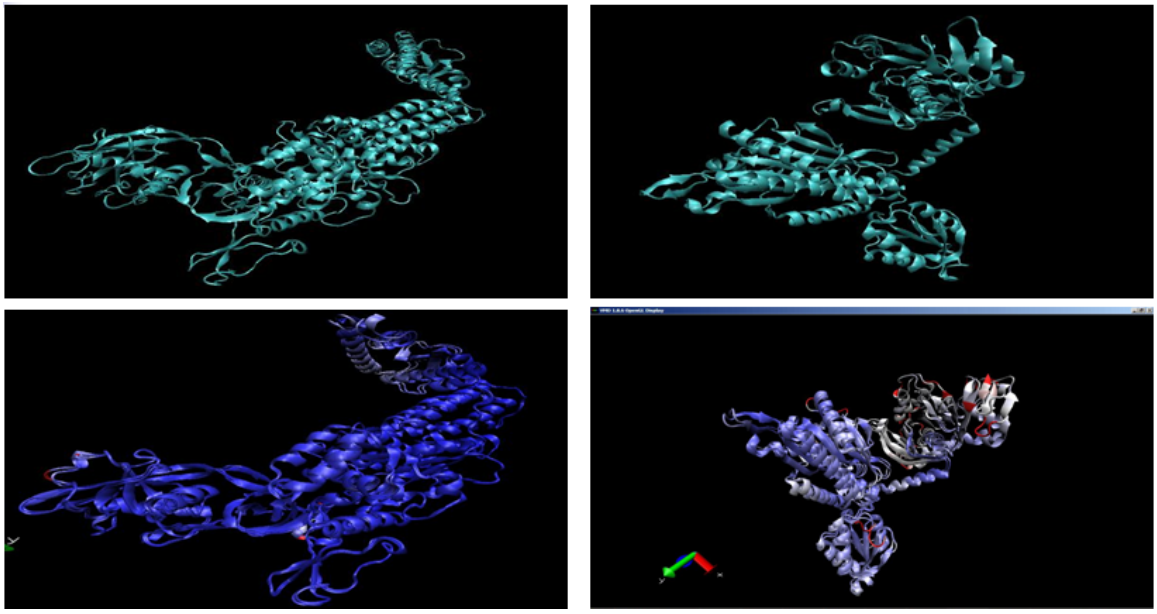

Figure 4-1 Example of self-testing from Class I and one from Class II. On the top left it shows the original structure of 1GAX, Valyl-tRNA Synthetase chain A. On the bottom left corner, it shows the structure of Class one Aminoacityl-tRNA Synthetase, 1GAX, a Valyl-tRNA Synthetase chain A aligned to its predicted version. It shows relatively accurate structure prediction for the tested enzymes. just very few structures are different and the catalytic domain is modeled very well. On the top right it shows the original structure of 1QF6, Threonyl-tRNA Synthetase chain A. On the bottom left corner, it shows the structure of Class II Aminoacityl-tRNA Synthetase, 1QF6, Threonyl-tRNA Synthetase chain A aligned to its predicted version. It shows how accurate the structure and model are from each other and how the catalytic domain are predicted

4.3 Results

In this section, the statistics regarding the number of successfully modeled sequences are given. As mentioned in section 4.2, THREADER was used to provide the list of templates for homology modeling. For some sequences, either no templates were found because of to poor threading z- score or MODELLER rejected the prediction after assessing the PDB files against some criteria such as Ramachandran Plots violation, etc.

The python code developed for prediction the 3D structures using MODELLER is given in the Appendix A. Table 4.1 summarizes the numbers of models that were provided after reliability assessments of predicted protein structure models using DOPE score [23] in MODELLER.

Having obtained the predicted structures, a multiple sequence aligner using STRAP's TM-Align algorithm was developed in order to identify the common structural core (Figure 3.2) in the predicted models and extract the sequences of these regions for each target protein. The output for this aligner is a multiple sequence alignment in NEXUS format. This procedure is described in section 5.2.1. By aligning these predicted structures, it is possible to predict the regions corresponding to the catalytic domains. Furthermore, it is then easy to construct the multiple sequence alignment of catalytic domains. Multiple sequence alignment is essential material for evolutionary analysis.

Figure 4.2 represents the sequence logo of these multiple sequence alignment of predicted catalytic domain sequences.

Table 4-1 Predicted structures after performing homology modeling using MODELLER. Lys-RS sequences were included in both classes and the analysis of threading and 3D structure prediction were performed twice on this type of Aminoacityl-tRNA Synthetase because it is not known that either in reality a Lys-RS is belong to class I or class II structural motifs.

Aminoacityl-tRNA Synthetases	Total No. of Sequences	No. of Models (Threshold =3)
Class I (+Lys-RS)	1536	517
Class II (+Lys-RS)	1681	917
Total	3068	1434

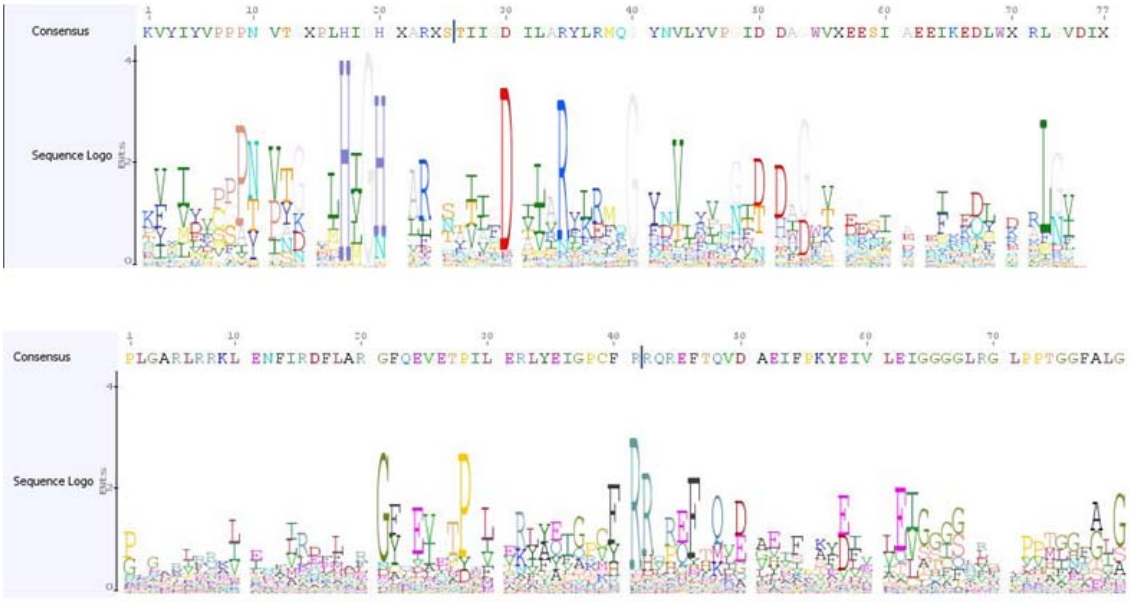

Figure 4-2 Sequence logos of the predicted catalytic domain of class I (top) and class II (bottom) Aminoacityl-tRNA Synthetase (generated using GENEIOUS http://www.geneious.com/)

5 Phylogenetic analysis

5.1 Materials

PAUP* have been used for finding distance matrix and generating neighbor joining evolutionary trees based on the multiple sequence alignment. GARLI has been applied to construct the evolutionary trees using Maximum Likelihood estimation. PhyML is also used to compare the topologies of maximum likelihood estimation trees with different tools like GARLI. In the end, BEAST has been brought into play to find the maximum credibility tree using Bayesian inference. BEAST uses Monte Carlo Markov Chain (MCMC) to average over tree space, so that each tree is weighted proportional to its posterior probability.

5.2 Methods

The methods for this step of the project is divided to two phase as well. Since phylogenetic analysis is mostly done on the multiple sequence alignment then it is essential to provide such multiple sequence alignment from the conserved regions (catalytic domain) of the predicted models to go back from 3D structure to primary structure (sequence) level. The first phase is to construct this multiple sequence alignment. The second phase is two apply different phylogenetic analysis approaches on this multiple sequence alignment.

5.2.1 How to align the predicted models with the common structural core of catalytic domains

Having generated the PDB files corresponding to predicted 3D structure of Aminoacityl-tRNA Synthetase sequences then next step is to find the catalytic domain sequences of these structures. One way is to align the predicted structures against the superposed catalytic domains of known Aminoacityl-tRNA Synthetase structures that have been discussed in section 3.4 and showed in Figure 3.2.

In section 3.3.2, it was shown that the common structural core in catalytic domains is consisting of a set of truncated PDB files containing residues of conserved regions (catalytic domain). Therefore, the question is how to superpose these common structural cores of catalytic domains from all types of Aminoacityl-tRNA Synthetase in each class.

As mentioned in section 2.5, STRAP [21] is convenient tool for such alignment. It is also provides programming interface for scientists. People with basic computer skills are capable of implementing statistical methods or embedding existing bioinformatical tools in STRAP by themselves. In STRAP, there are different algorithms for alignment. TM-Align algorithm was used, which simply tries to find the common structural core of catalytic domain somewhere in a predicted Aminoacityl-tRNA Synthetase PDB files (the results from MODELLER).

The code obtains all the PDB files containing catalytic domains of known Aminoacityl-tRNA Synthetase and a predicted structure, and then tries to align the structure against the conserved regions (Figure 3.2). I have developed a Java™ program

which using STRAP and Bio-Java libraries for multiple structurally alignment. That is, by running this code on the common structural core of catalytic domains (Figure 3.2) (conserved regions) will be aligned to different parts of a predicted structure that will be referred as the catalytic domain for the predicted Aminoacityl-tRNA Synthetase structures. The output will be in FASTA format, which then be exported as NEXUS format for phylogenetic analysis. Using UNIX shell scripting, this multiple sequence aligner was run for each predicted structure. One another reason for running one by one, is that it is not desired to align a predicted model to another predicted model. That means it is not possible to put all predicted and common structural of catalytic domains together at the same time, and then superimpose them once. Instead, it is essential to align each predicted model to the superposed catalytic domains of known Aminoacityl-tRNA Synthetase structures, separately. Another reason for using STRAP was that it is possible to save the sequence alignment of the common structural core of catalytic domains (Figure 3.2) in memory (i.e. caching the data in memory) so that the alignment of common structural catalytic domains is fixed and just the predicted model is varying for each run. It suggests that the common structural motif is going to be found in different predicted Aminoacityl-tRNA Synthetase. Figure 5.1 shows an example of the outputs from this aligner.

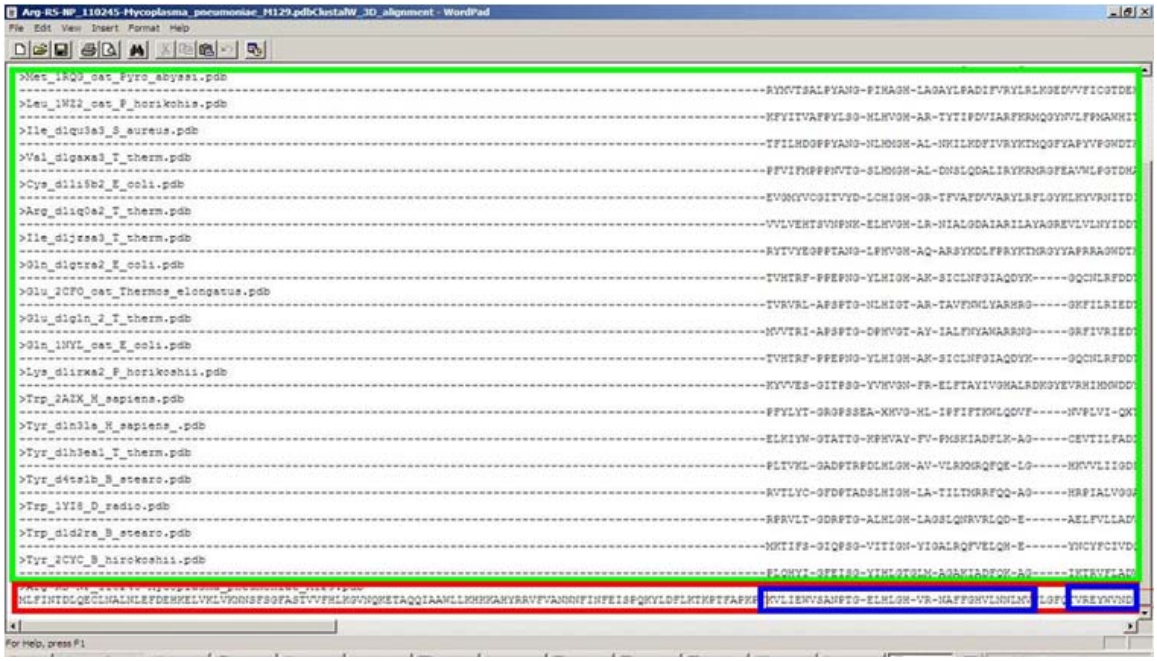

Figure 5-1 an example output of the aligner. The output is a multiple sequence alignment. The dash symbol "-"represents gap, Illustrates a sample output of the aligner. The output is a multiple sequence alignment. This alignment is fixed by caching the alignment in the memory. The Red box shows the predicted Aminoacityl-tRNA Synthetase model. The Blue box represents the part of predicted model, which has been aligned to the common structural core of catalytic domains (Figure 3.2). The sequence inside the blue box could be referred as predicted catalytic domain on Aminoacityl-tRNA Synthetase sequence

At this point, the multiple sequence alignment of predicted structures against the superimposed catalytic domains of known Aminoacityl-tRNA Synthetase structures is available (common structural core of catalytic domains (Figure 3.2)). By using a PERL

script, the sequences of predicted catalytic domain for each Aminoacityl-tRNA Synthetase are identified. In other words, the sequences of aligned regions to the common structural core of catalytic domains (Figure 3.2) in the alignments are identified and then put together in a single file to construct a multiple sequence alignment. After providing a multiple sequence alignment of predicted catalytic domain of Aminoacityl-tRNA Synthetase in a single file in each class, the sequences of the catalytic domains from the common structural core are also added to this multiple sequence alignment manually to start the final analysis, which is evolutionary analysis. Many tools used in the evolutionary analysis work with NEXUS file format as an input so the multiple alignment file exported to a Nexus format before starting the phylogenetic analysis.

5.2.2 How to reconstruct the evolutionary tree for the extracted catalytic domains of Aminoacityl-tRNA Synthetases

The phylogenetic analysis started with constructing the tree with the simplest method of UPGMA by using PAUP*. It is simple and fast to generate to generate tree using UPGMA and therefore this method is often first choice. The results were bootstrapped to answer how different the UPGMA trees are from each other and on the other hand, which clades in the tree found with more confident in the whole set of UPGMA trees.

As might be expected, due to low sequence identity, the topologies were quite different and the bootstrap values did not support the idea of using UPGMA to generate trees be reliable. Therefore maximum likelihood analysis was applied using two packages GARLI and PhyML. To do so, it is essential to specify an empirical substitution matrix. They have been run the maximum likelihood analysis with different empirical substitution matrix. Looking at the final likelihood probability score then the empirical substitution matrix was chosen and then the further runs just performed using this substitution matrix. In order to set the parameter “number of generations”, GARLI was run multiple times (with different values). Then the final liog-likelihood have been done and the final likelihood score were considered to see whether the log-likelihood score converge to a certain value. Finally, by trial and error, the number of generations was chosen 1 000 000 so that we can have more confident that the log-Likelihood values converge. In Maximum Likelihood estimation analysis, it is always recommended to run the dataset multiple times on the other hands GARLI was a stochastic program. Therefore, GARLI were run on both classes around 10 times (Figure 5.2).

Then the similarity of tree topologies were compared considering log Likelihood score. In order to find out whether the trees are similar or not the Symmetric Distance Index is used. This index for each A, B trees represents the number of clades in tree A that are not present in tree B in addition to the number of clades in tree B that are not present in tree A. Let us consider a random tree with 0.5 probability of having a shared split between two trees. Let us also consider “n” as number of taxa in the tree. Then “2n-3” is the number of splits in an unrooted tree. By decreasing the number of symmetric distance in one of the trees then the number of shared splits is achieved. Finally, by dividing this number by the total number of splits again, the proportion of shared splits in one tree is calculated. In formal way:

n: number of taxa

d_s: Symmetric Distance Index

2n-3=A, where A is the number of splits in an unrooted tree

2n - 3 - (d_s)/2=B, where B is the number of shared split in one of the trees to the other one

B/A=C, where C is the proportion of shared splits

The value of C then is used to see how similar the two topologies are. If they are not showing the similar topologies perhaps, it means because of to large number of taxa there are so many trees topologies with relatively same probability given to the alignment but different topologies, which is not guiding the analysis to a consensus one. At this point one can think to use a method that is more sophisticated. For example, to find the posterior probability of trees given an alignment, search through the trees, and find the one with the highest log likelihood score as the maximum credibility tree. BEAST was used for this purpose because it uses MCMC to average over tree space, so that each tree is weighted proportional to its posterior probability. It is a very computationally intensive task. Therefore, the numbers of taxa were decreased. To do so I decided to choose samples from a family of close organisms to decrease the number of taxa.

5.3 Results

The trees constructed using neighbor joining and UPGMA methods did not show solid topology similarity. The bootstrap supports for branches in these trees were around 50-60% even for the branch points close to the root of tree. There were many ambiguities in the tree topologies. For example, some taxa from one type of Aminoacityl-tRNA Synthetases clustered to another Aminoacityl-tRNA Synthetases group. Nevertheless, considering its homolog from the same family of Aminoacityl-tRNA Synthetases and a related organism, the homologs are clustered in different clade. By looking at some of trees manually, it found out that the trees are not showing a solid evolutionary view. That is, looking at UPGMA trees, they suggest that all Aminoacityl-tRNA Synthetases are distinguished from each other in very beginning in the evolution and started to evolve separately from other types of Aminoacityl-tRNA Synthetases that looks contradictory if to the structural conservation. Such ambiguities with poor bootstrap support that are close to random tree are suggesting that results UPGMA is not reliable and the results are close to a random tree.

As mentioned in section 3.2.5, it is essential to find out which empirical substitution matrix for amino acids fits better to the multiple alignment and gives a better log likelihood score. Having done test on different matrices such as JTT, DAYHOFF, WAG, etc. it turned out that WAG is the best choice for both classes. Figure 5.2 shows the trace files on class one taxa using GARLI with WAG as substitution matrix. They illustrates the trace of the log likelihood estimation of trees against the states (no. of generations = 1 000 000) using GARLI on Aminoacityl-tRNA Synthetases predicted catalytic domains in class I and class II. It is comparison of maximum likelihood tree search for different runs starting from dissimilar random topology

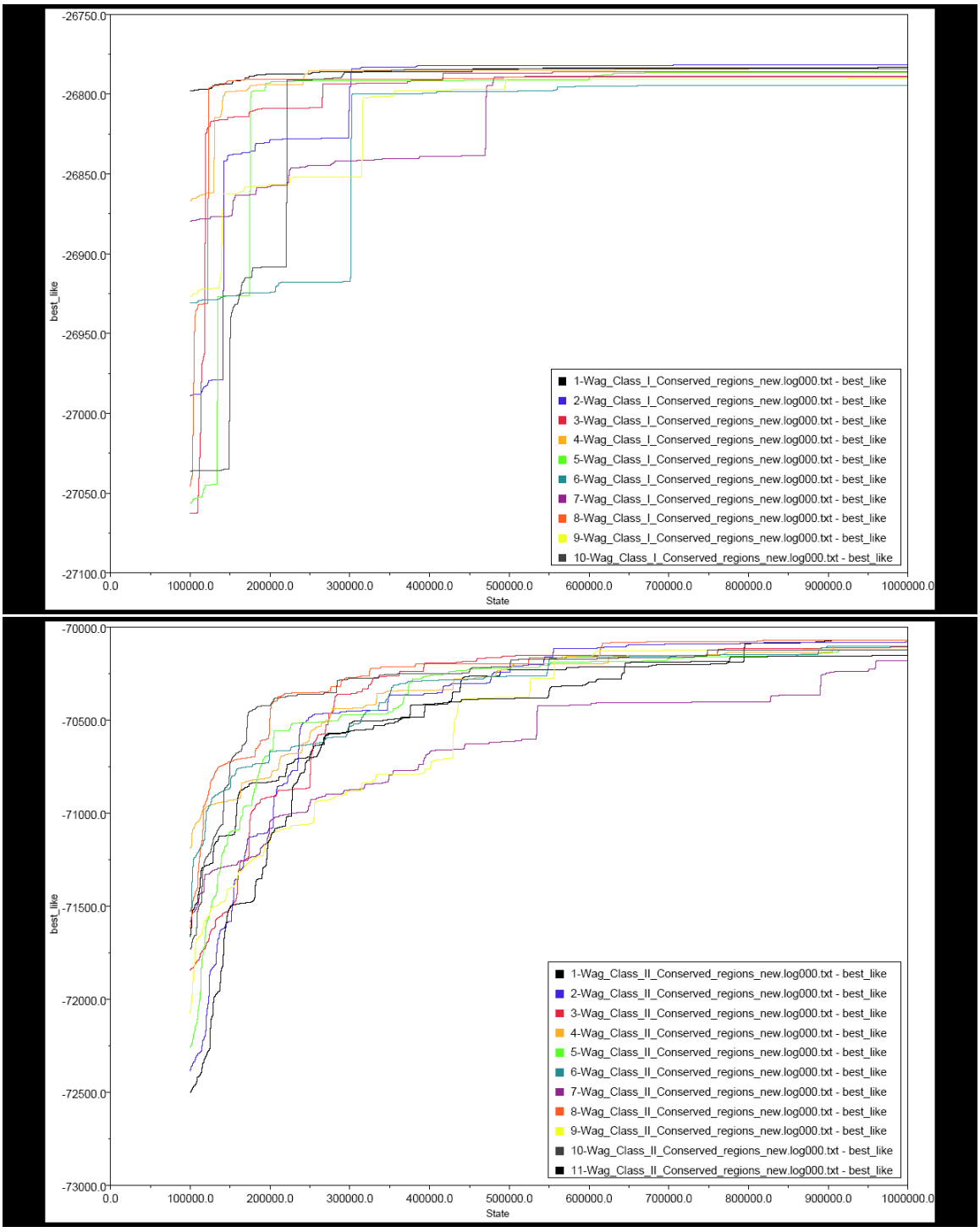

Figure 5-2 on the top it illustrates the trace of likelihood estimation of trees against the states (no. of generations) using GARLI on Aminoacityl-tRNA Synthetases predicted catalytic domains in class I. It is comparison of maximum likelihood tree search for 10 different runs starting from dissimilar random topology. On the bottom it illustrates the trace of likelihood estimation of trees against the states (no. of generations) using GARLI on Aminoacityl-tRNA Synthetases predicted catalytic domains in class II. It is comparison of maximum likelihood tree search for 11 different runs starting from dissimilar random topology

Having sorted the trees from each run based on the final log-likelihood scores, the top trees with the least score were picked up. They were considered to check out how similar these topologies are. Table 5-3 shows the summary of the results for testing the GARLI best trees in both classes:

Table 5-1 the results for testing the GARLI best trees in both classes

CLASS I (Number of taxa 485):

Final Likelihood scores:
1st tree: -26783.91017
2nd tree: -26785.58076
3rd tree: -26781.57472
4th tree: -26782.72109

Symmetric Distance Index (d_s):

Tree	1	2	3	4
1	0			
2	304	0		
3	266	***244***	0	
4	324	256	252	0

Shared splits between the closest topologies (Trees 2 and 3, highlighted in blue): ***87%***

CLASS II (Number of taxa 901):

Final Likelihood scores:
1st tree: -70099.86373
2nd tree: -70078.51920
3rd tree: -70069.63931

Symmetric Distance Index (d_s):

Tree	1	2	3
1	0		
2	772	0	
3	***758***	820	0

Shared splits between the closest topologies (Trees 1 and 3, highlighted in blue): ***79%***

Due to large number of taxa, PhyML package performed slowly to find the Maximum likelihood trees. Although having looked at the symmetric distance index and the shared splits in the trees generated by GARLI, it is decided to discard following Maximum Likelihood approach because there are many trees with approximately the same probability given to my alignment but different topologies as the symmetric distance index suggests. However having similar likelihood probability for trees was suggesting it is getting closer to find a consensus tree that describes this alignment (alignment of the predicted Aminoacityl-tRNA Synthetases catalytic domain sequence). To generate trees using BEAST, it is essential to decrease the number of taxa. Because it will take a lot of time to find the posterior probability distribution with effective sample size > 1000 (ESS value in beast) for large amount of taxa[16]. For species-level phylogenies, coalescent priors are generally inappropriate. In this case, authors of BEAST suggest that to use the Yule tree prior. Yule Process has been chose as prior distribution. The site heterogeneity model in BEAST was set to Gamma Distribution with Invariant sites. Please refer to Appendix C where the different site heterogeneity models from the BEAST User's manual are presented. BEAST searches through the tree

[16]For more information how to interpret the results from BEAST using ESS, please refer to BEAST home page http://beast.bio.ed.ac.uk/ or Appendix C

space, changes the branch and clades in any generations, and calculates the likelihood of the tree given the alignment then by Bayesian Inference, it calculates the Posterior probability. In the Figure 5.3 illustrates the posterior distribution of the tree spaces based on their probability to the alignments in class I(left) and class II(right). The Y-axis represents the frequency and on the x-axis, represents posterior probability. The statistics of the analysis is shown in the table 5.2.

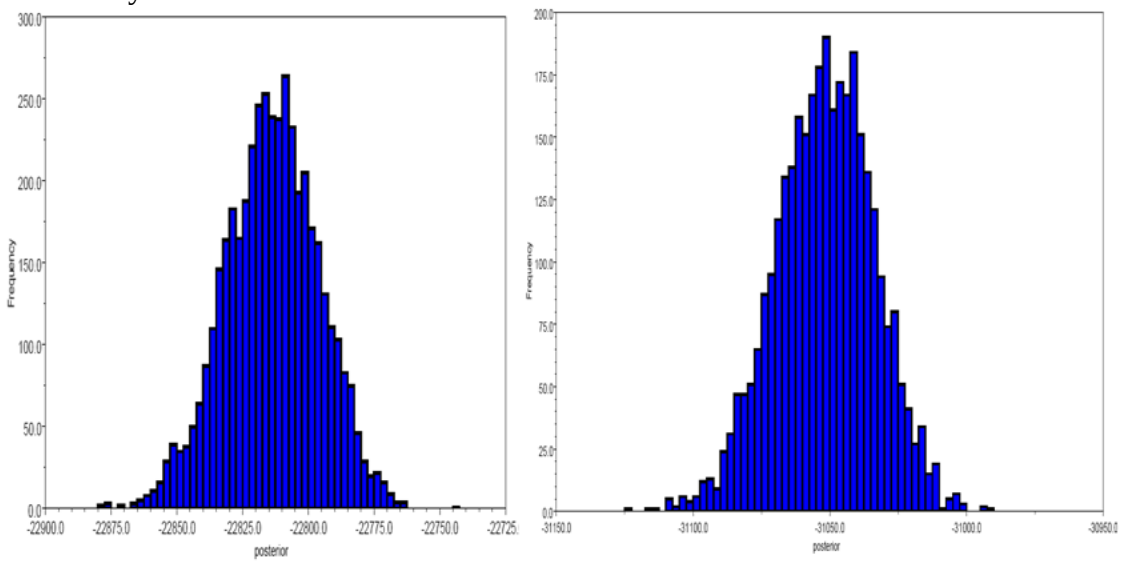

Figure 5-3 The posterior distribution of the tree spaces based on their probability to the alignments in class I(left) and class II(right). The Y-axis represents the frequency and on the x-axis, represents posterior probability.. The Prior distribution is YuleProcess discussed in the Appendix C section and the likelihood distribution is being calculated during the analysis given to alignment. Having Prior and Likelihood distribution then the posterior distribution is handy to calculate.

Table 5-2 the statistics of the analysis in BEAST for class I (top) and class II (bottom)

Summary Statistic for Class I	*Likelihood*
Mean	-22810.00
Standard deviation of mean	0.73
Median	22810.00
95% HPD lower	-22850.00
95% HPD upper	-22780.00
Auto-correlation time (ACT)	38730.00
Effective sample size (ESS)	571.55
Summary Statistic for Class II	*Likelihood*
Mean	-31050.00
Standard deviation of mean	1.01
Median	-31050.00
95% HPD lower	-31090.00
95% HPD upper	-31020.00
auto-correlation time (ACT)	51660.00
effective sample size (ESS)	318.05

Due to the large number of taxa in both classes, it is hard to browse in a page. Therefore, I decided to put the files containing the tree topologies online[17]. From the link, the maximum credibility trees for both taxa corresponding to class I or to class II Aminoacityl-tRNA Synthetases catalytic domains are available. These trees suggest a very good separation among the different types of Aminoacityl-tRNA Synthetases catalytic domains. This good separation is supported with high posterior probability in the branch points that makes us more confident that these clades, which contain one type of Aminoacityl-tRNA Synthetases, are always together. Another good feature of the trees generated in BEAST was that the known structures are clustered with their corresponding Aminoacityl-tRNA Synthetases type.

In order to see whether these trees follow the canonical pattern of life proposed by Woese et al. [24], BEAST was run separately on taxa in each clade. It would also give a more accurate image on the evolution of each type of Aminoacityl-tRNA Synthetases catalytic domain. Each clade corresponds to type of Aminoacityl-tRNA Synthetases predicted catalytic domains. The results are shown in Figures. Figure 5.4-9 show the trees for each type of Aminoacityl-tRNA Synthetases predicted catalytic domains in class I. There is no tree for Lys-RS (Class I) because there were no predictions for this type. In chapter 6, we discuss why that could happen. Since there were few predicted models in Trp-RS and Tyr-RS and all of them were related to Bacteria, no separated trees were generated for these types. That is because the trees from these few taxa do not show reliable evolutionary picture. Figure 5.10-18 show the trees for one type of Aminoacityl-tRNA Synthetases predicted catalytic domains in class II.

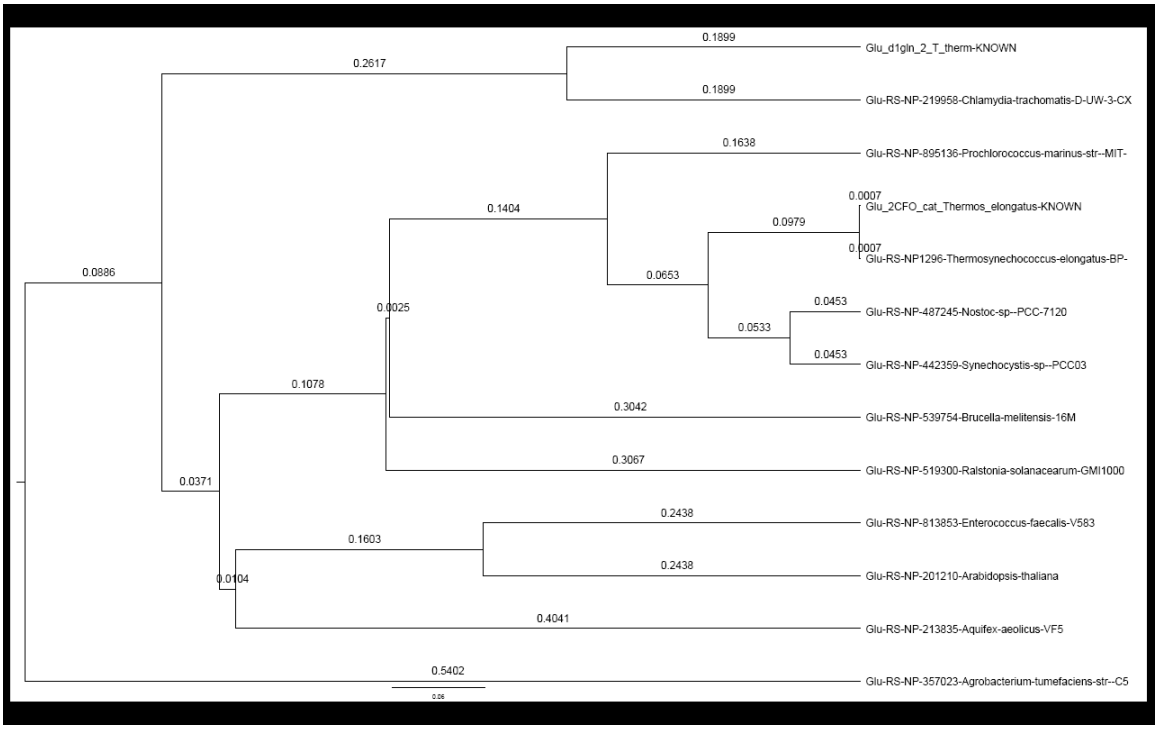

Figure 5-4 the tree related to Glu-RS predicted catalytic domain regions structures. The numbers represents the posterior probability of each branch point

[17] /chalmers/users/khorshid/Master_Thesis/Supplementary/Trees

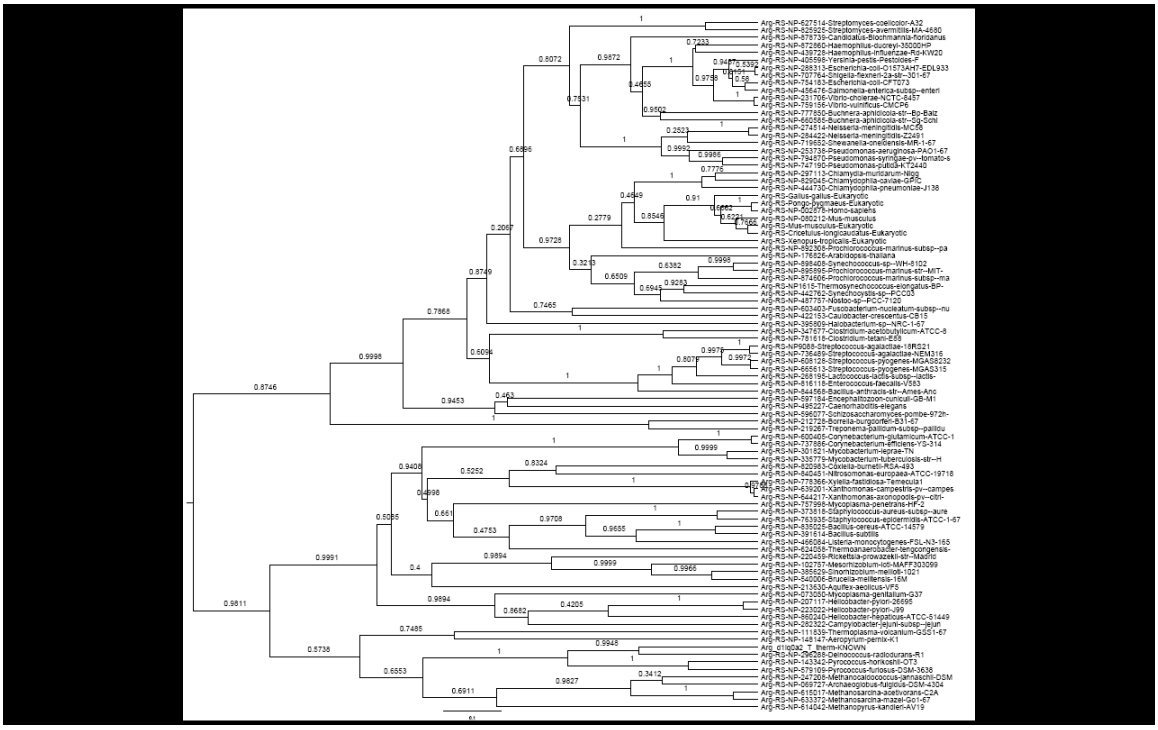

Figure 5-5 the tree related to Arg-RS predicted catalytic domain regions structures. The numbers represents the posterior probability of each branch point

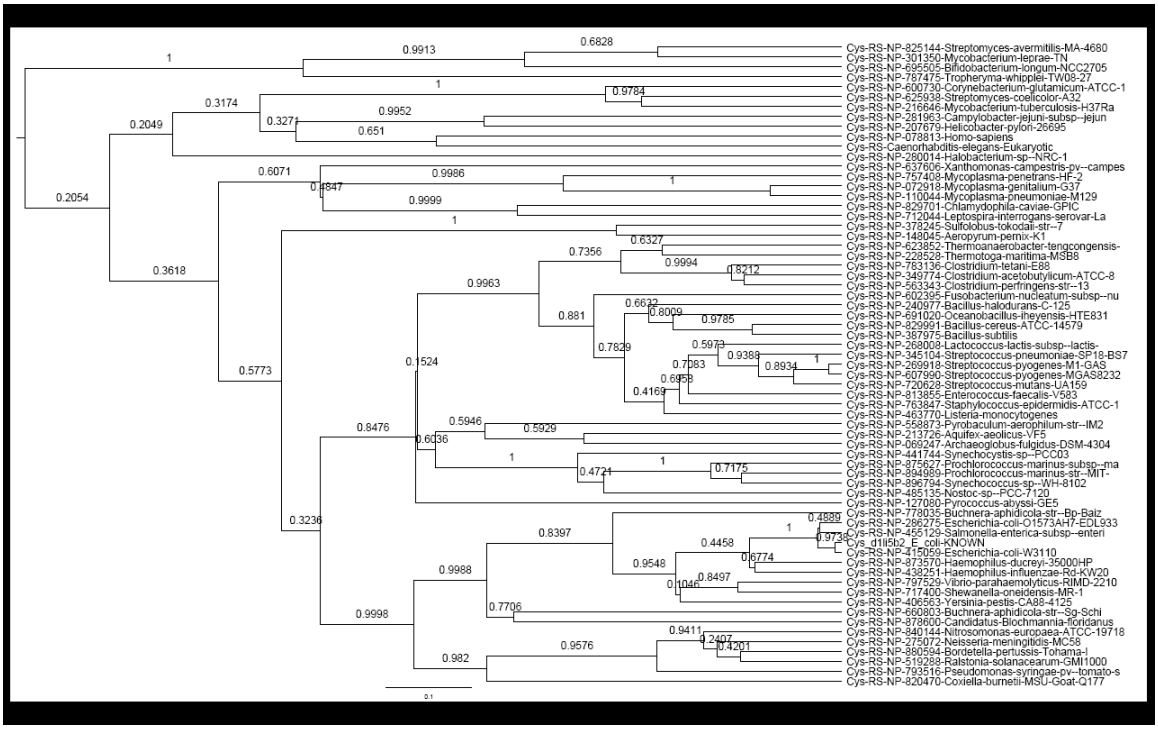

Figure 5-6 the tree related to Cys-RS predicted catalytic domain regions structure. The numbers represents the posterior probability of each branch point

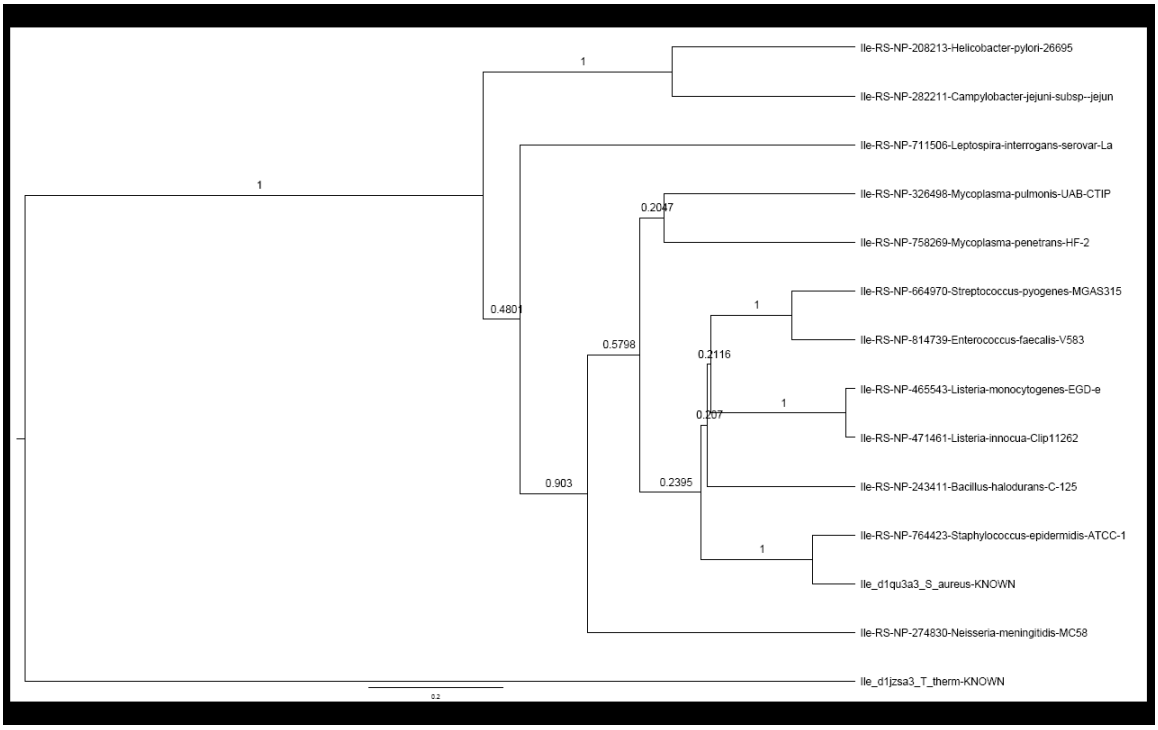

Figure 5-7 the tree related to Ile-RS predicted catalytic domain regions structure. The numbers represents the posterior probability of each branch point

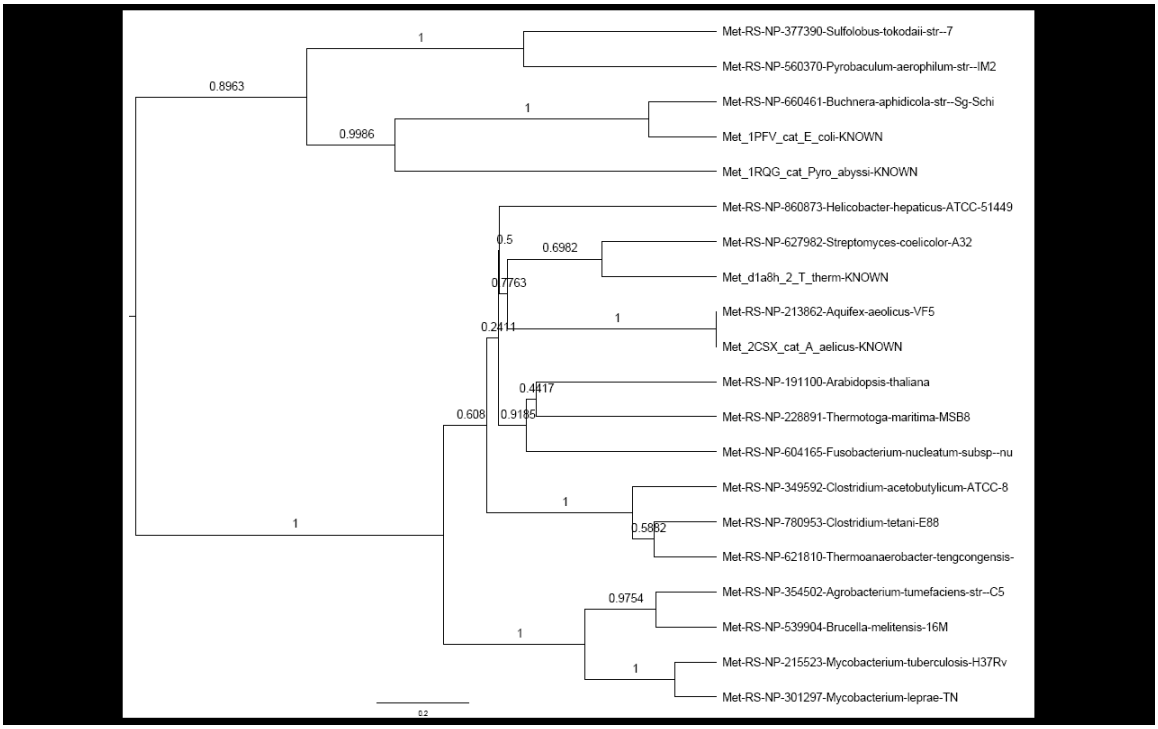

Figure 5-8 the tree related to Met-RS predicted catalytic domain regions structure. The numbers represents the posterior probability of each branch point

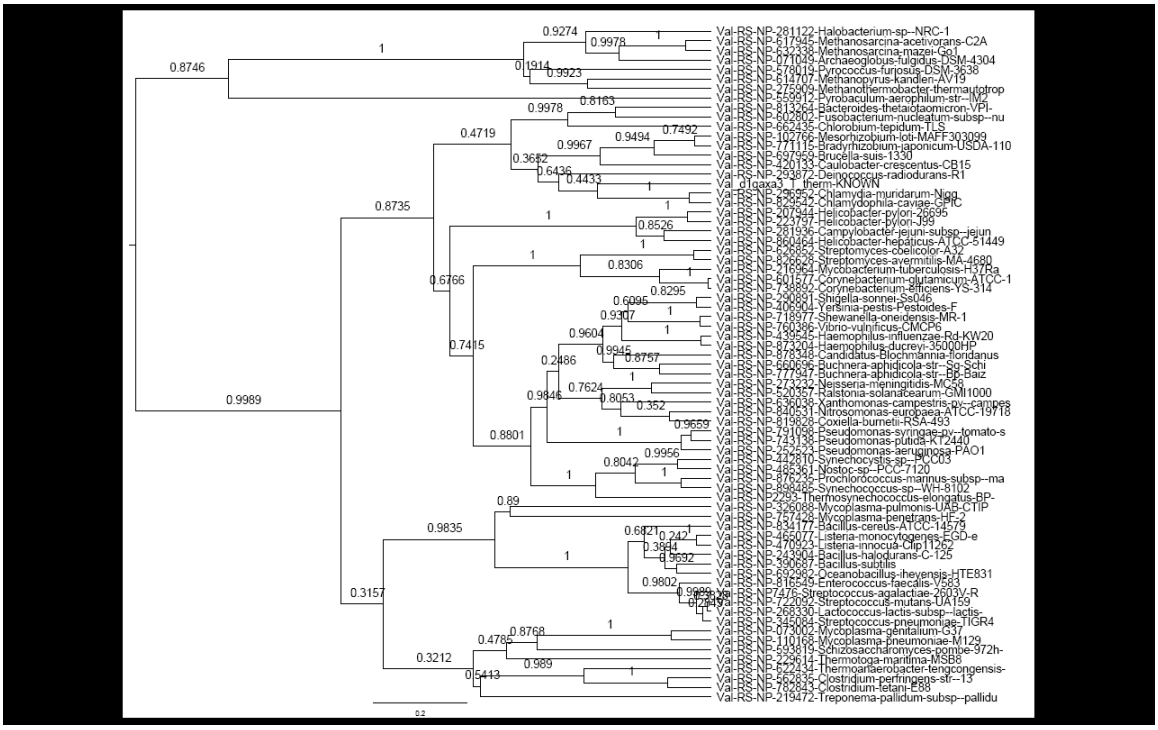

Figure 5-9 the tree related to Val-RS predicted catalytic domain regions structure. The numbers represents the posterior probability of each branch point

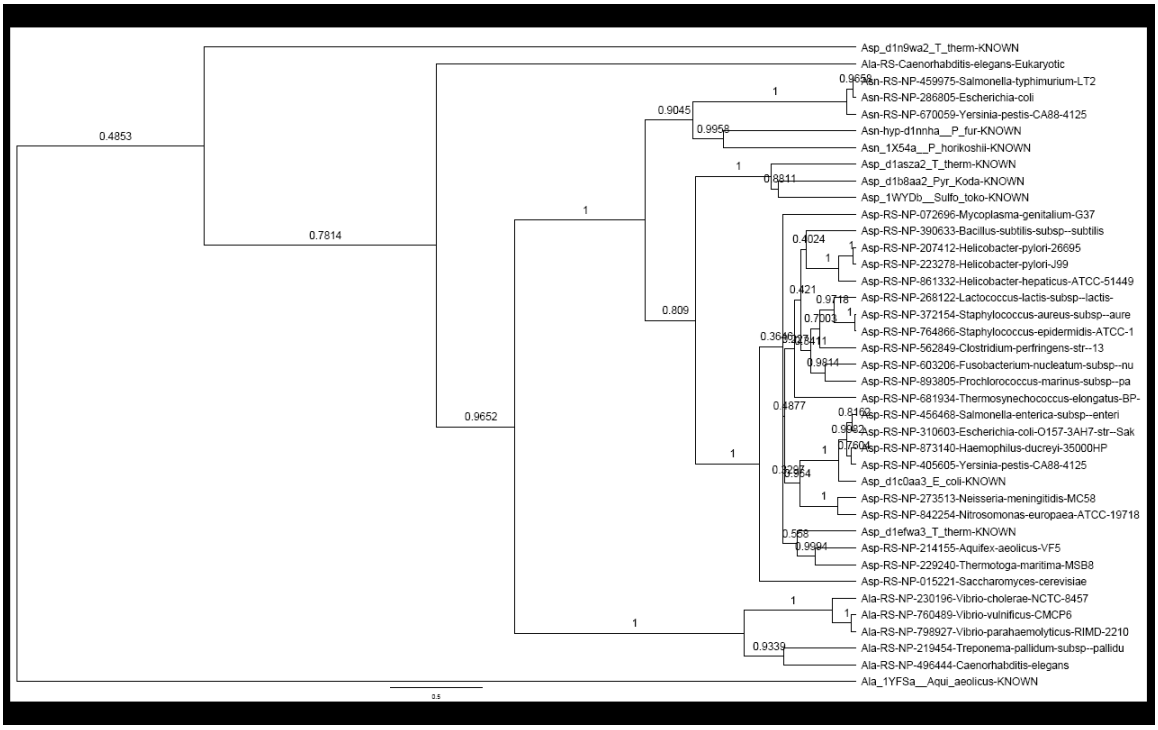

Figure 5-10 the tree related to Ala-RS, Asp-RS and Asn-RS predicted catalytic domain regions structure. The numbers represents the posterior probability of each branch point

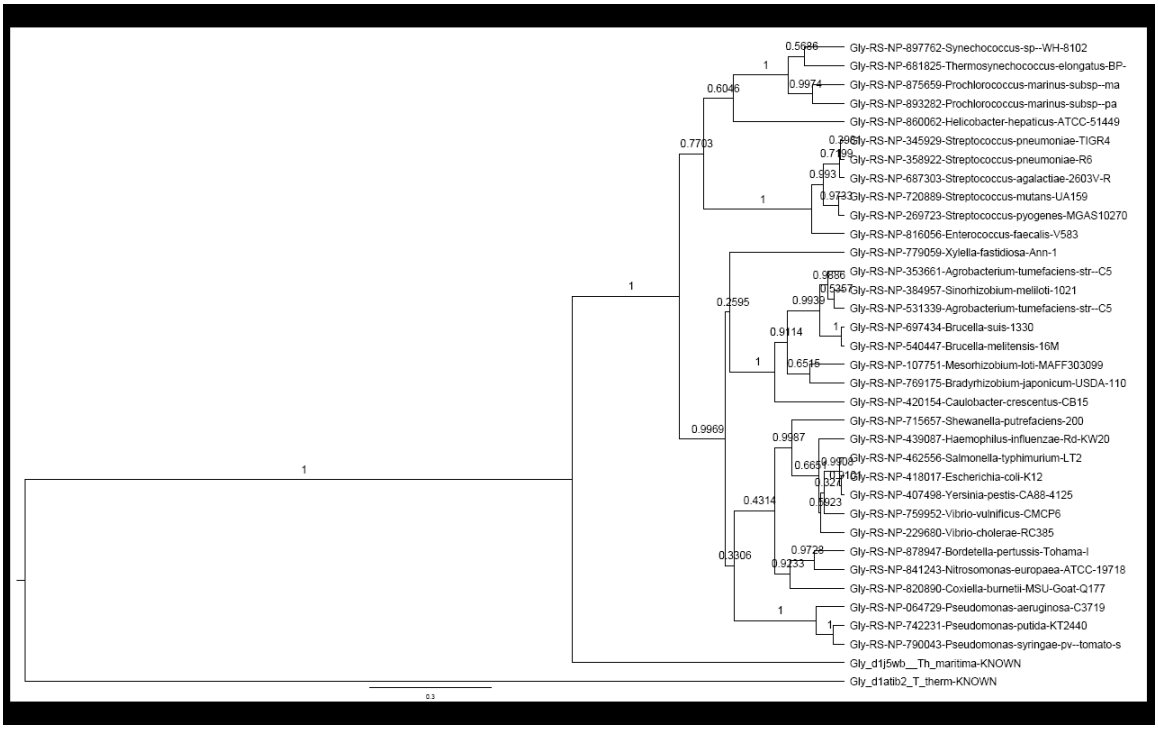

Figure 5-11 the tree related to Gly-RS predicted catalytic domain regions structure. The numbers represents the posterior probability of each branch point

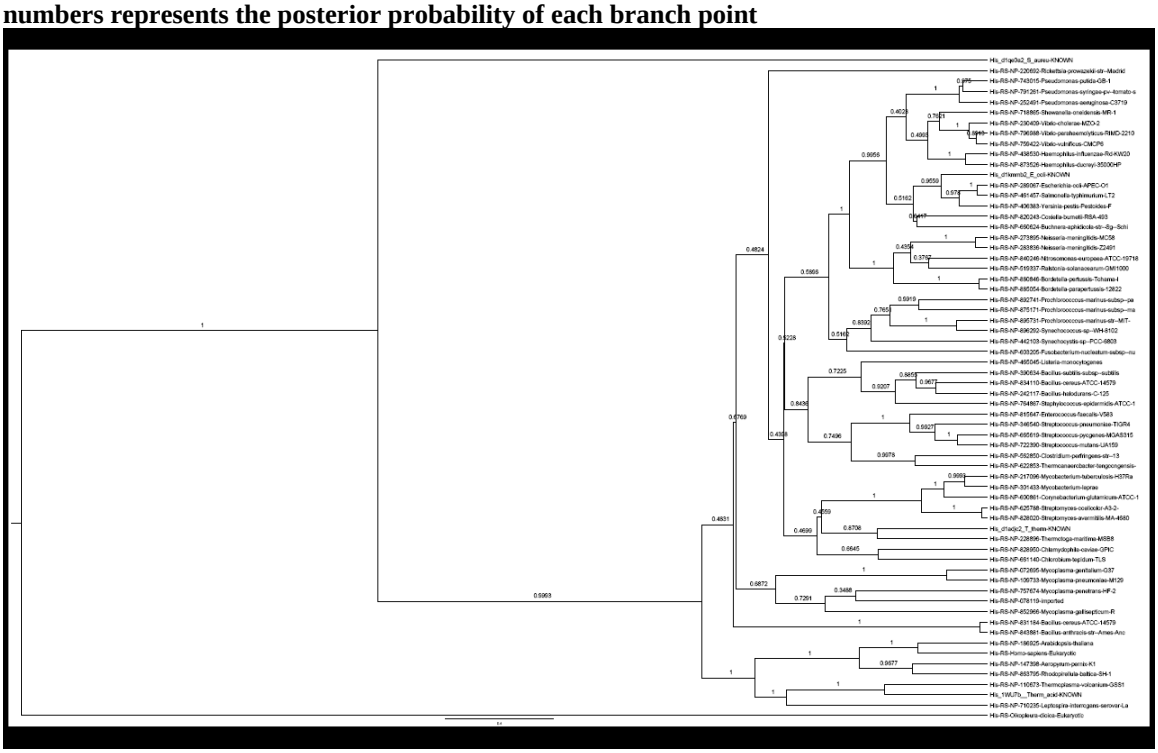

Figure 5-12 the tree related to His-RS predicted catalytic domain regions structure. The numbers represents the posterior probability of each branch point

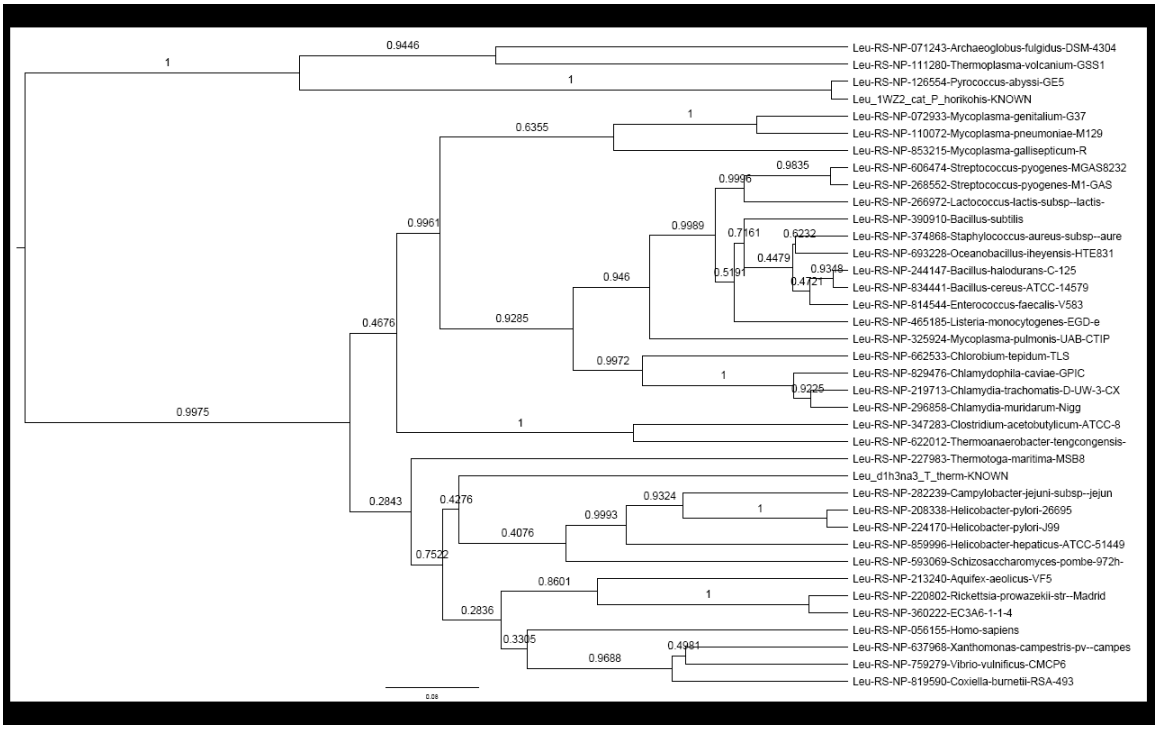

Figure 5-13 the tree related to Leu-RS predicted catalytic domain regions structure. The numbers represents the posterior probability of each branch point

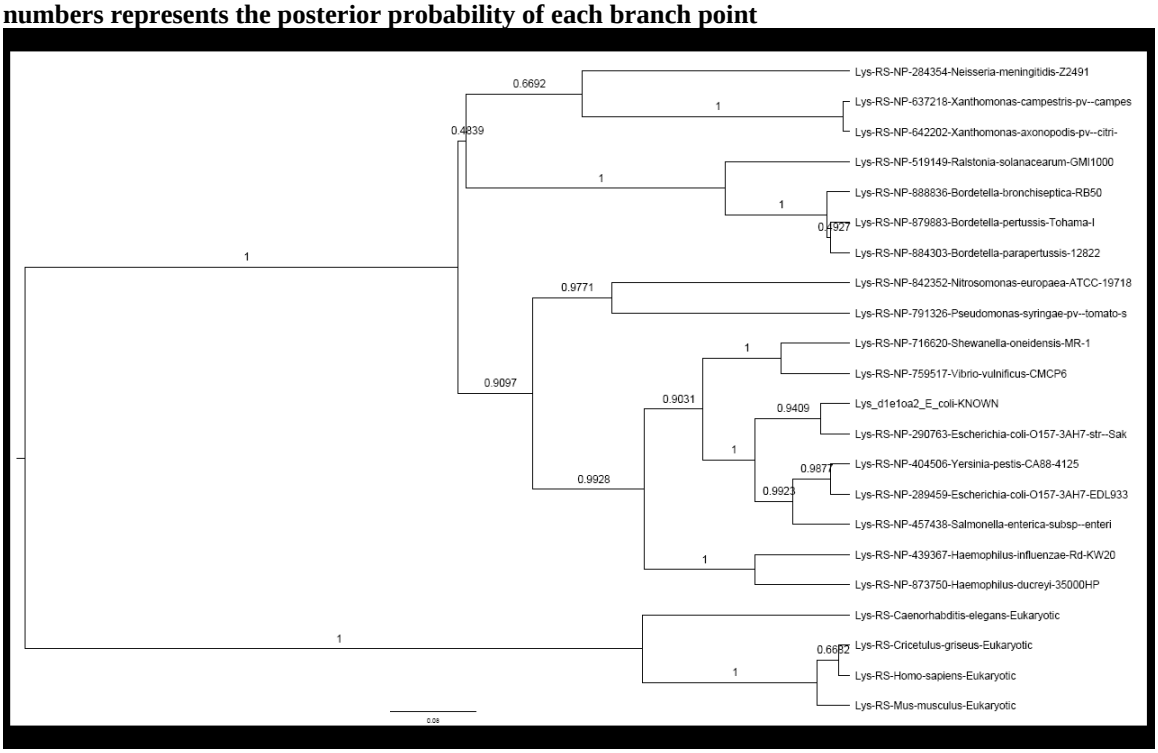

Figure 5-14 the tree related to Lys-RS predicted catalytic domain regions structure. The numbers represents the posterior probability of each branch point

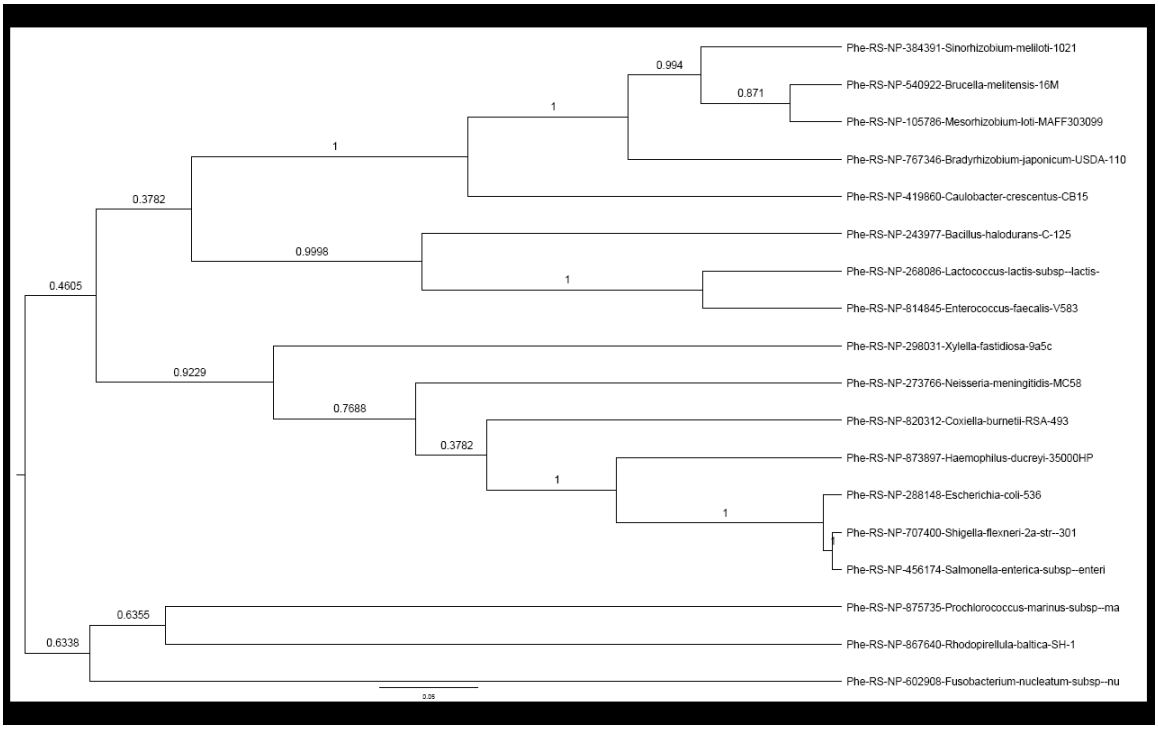

Figure 5-15 the tree related to Phe-RS predicted catalytic domain regions structure. The numbers represents the posterior probability of each branch point

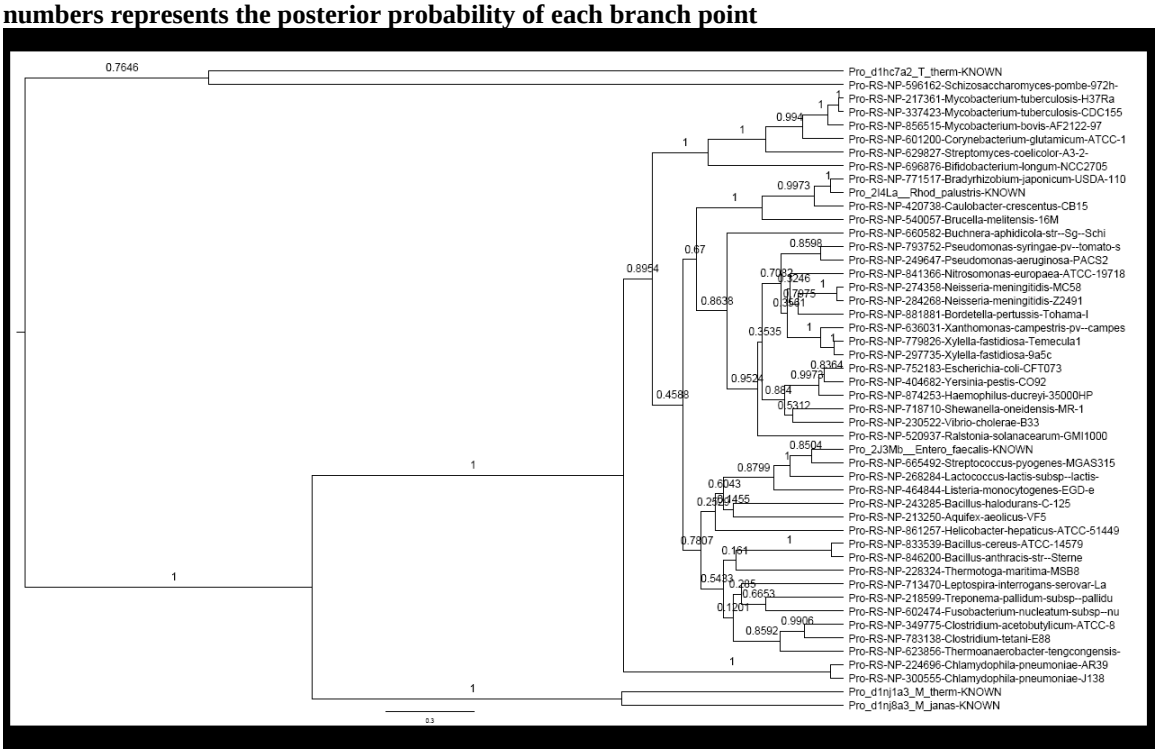

Figure 5-16 the tree related to Pro-RS predicted catalytic domain regions structure. The numbers represents the posterior probability of each branch point

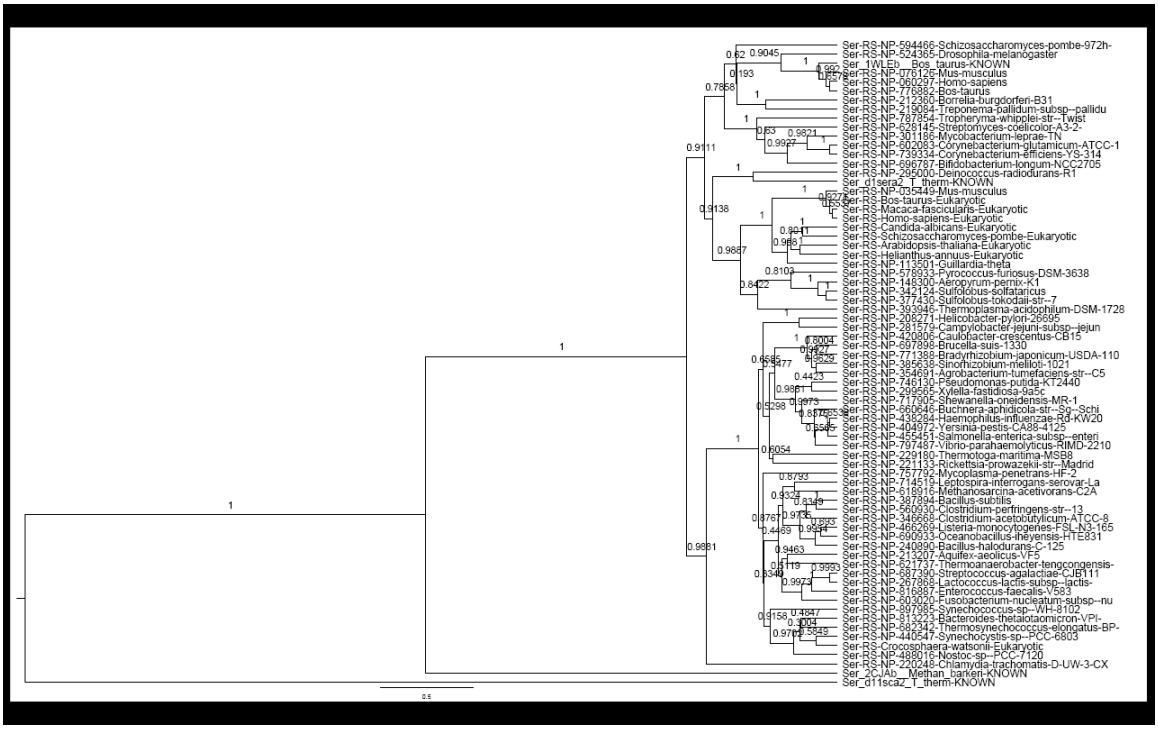

Figure 5-17 the tree related to Ser-RS predicted catalytic domain regions structure. The numbers represents the posterior probability of each branch point

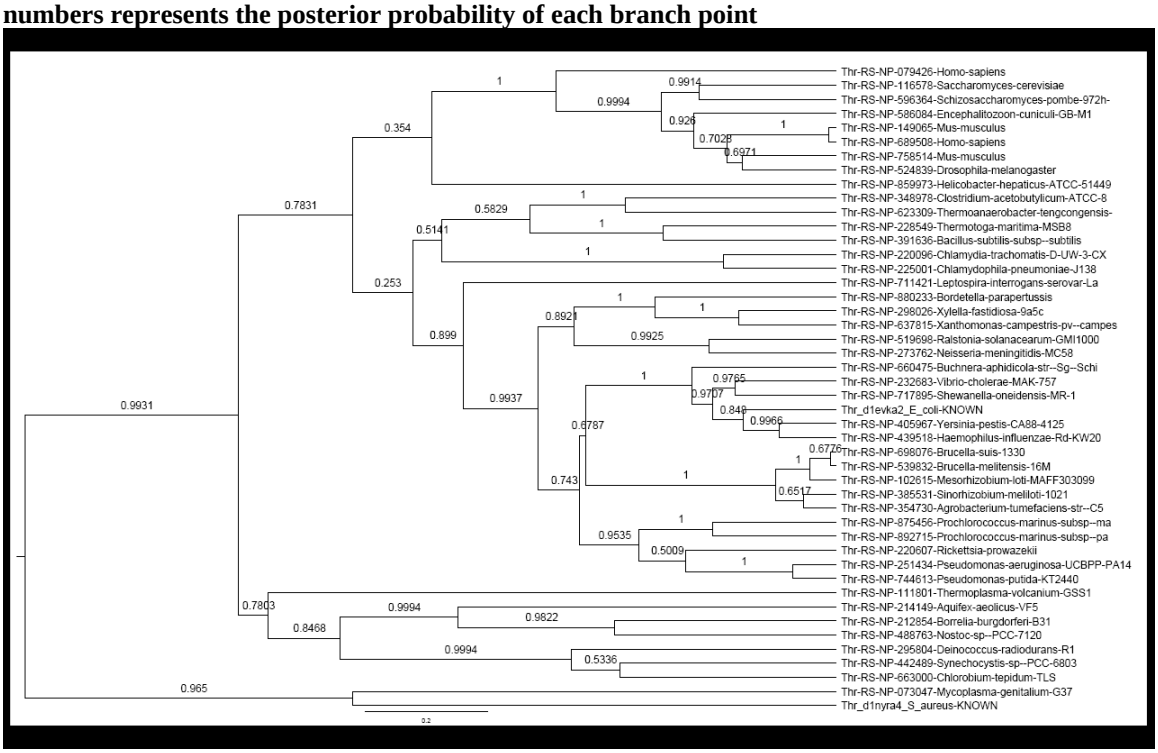

Figure 5-18 the tree related to Ser-RS predicted catalytic domain regions structure. The numbers represents the posterior probability of each branch point

6 Discussion

6.1 Discussion

The STAMP algorithm is very sensitive in superposition of with multiple proteins with different global structural motif (such as the set of non redundant structures used in section 3.3.2) The aim was to find a highly conserved region (catalytic domains) among different types of Aminoacityl-tRNA Synthetase structures – which share the conserved catalytic domain structure and differ in other parts. Therefore, perhaps using the TM-Align algorithm overcomes these problems by exploiting weighting factors that weight the residue pairs at smaller distances more strongly than those at larger distances. The Weighted approach for calculating the goodness of fit perhaps help to end up with longer sequence common structural core of catalytic domains of Aminoacityl-tRNA Synthetase, which has dramatic effect on the credibility of the evolutionary analysis in the final step.

The overall Aminoacityl-tRNA Synthetases types known structures have different structure to each other but all share the catalytic domains with conserved structure. This is the reason that using STAMP was time consuming since it should be checked manually which type of remaining Aminoacityl-tRNA Synthetase should be added to the superposed catalytic domains of known Aminoacityl-tRNA Synthetase structures and truncate the structures again. On other hand, it should be avoided to make STAMP get confused in superposition of structures otherwise it stops and gives an error regarding not-similar structures.

Short length multiple alignment may affect the accuracy of evolutionary analysis inference. It might be good to use another type of algorithm for superposition of catalytic domains of known structures (catalytic domain structures) instead of STAMP. One possibility might be to use TM-Align – the same used in the section 5.2.1- that is using weighted approach for calculating the goodness of fit. It perhaps might help to end up with longer sequence as conserved regions in the catalytic domains for the profile, which has a dramatic influence to the credibility of the evolutionary analysis in the final step.

THREADER predicted a list of homologous structure for around 34% for sequences in class I and around 54% for sequences in class II out of the overall sequences, which were downloaded in the beginning. There are several reasons for this. First of all most of the experimentally known structures available in the Protein Data Bank are from Bacteria so finding a candidates among bacteria structures for all other organisms might not be successful. It might also be due to the lack of diverse experimentally known structures of in the Protein Data Bank for example the method did not work out to predict any 3D structures for class I Lys-RS. The idea also comes from when it was only possible to model about 20 sequences in total from sequences of Aminoacityl-tRNA Synthetase from animals (modern eukaryotic organisms) for all types in both classes. Although some of the 3D structure prediction models were also considered as poor models because having done sequence alignment with structural consideration – described in section 5.2.1 - with the known structures, the alignment

looked fuzzy with many gaps suggesting a poor sequence alignment. Therefore such models were ignored as well because such an alignment with many gaps did not describe any evolutionary track.

In the threading of class I sequences against their cognate library of TDB files, it was inevitable to decrease the z-score threshold from 3 to the 2.7 in order to have at least some samples from such types of Aminoacityl-tRNA Synthetases i.e. for Gln-RS in class I. Although, having performed the 3D modeling prediction, there were still very few predicted structures. It might be because of lack of available experimentally known structures for these types of Aminoacityl-tRNA Synthetases.

Sequence logos of the extracted catalytic domain sequences from predicted 3D structures could give us the regions that show us strong sequence conservation and double check the quality of prediction with the previous works [1][10] on the Aminoacityl-tRNA Synthetases catalytic domains. The sequence logos of the conserved regions also agree with the previous work Aminoacityl-tRNA Synthetases [10]. For example, in [10], they show the "HIGH/HVGH" regions with the very high sequence and structural conservation in the class I catalytic domain. This similarity of results would be supportive that the protein structure prediction could be applied successfully.

Another way to assess this success is to look at the self-testing mentioned in the section 4.2.2. It also showed that, even though sometimes there inaccurate structure predictions of other parts of the target protein were found, the catalytic domain is predicted satisfactorily.

The TM-align algorithm seems to work well for aligning for the procedure mentioned in section 5.2.1. It is possible to cache the multiple sequence alignment containing common conserved regions in the machine's memory in order to speed up the process. Perhaps it is possible to use this algorithm for preparing the profile of conserved regions from experimentally determined Aminoacityl-tRNA Synthetases because perhaps it could guide us to longer sequences corresponding to common structural core. As the result, it might help us to perform a better phylogenetic analysis.

The results from BEAST on the sequences from different types corresponding to their own class are interesting[18]. The main feature of the trees is that all predicted catalytic regions from the same types are clustered together. There are no vague groupings in these trees. For the tree corresponding to class II target proteins, the posterior probabilities of the branch points are very interesting. They are remarkably high (close to 1) in many branch points, specially for the splits that start to distinguish

[18] Tree for Class I Aminoacityl-tRNA Synthetases predicted catalytic domains is available at: /chalmers/users/khorshid/Master_Thesis/Supplementary/Trees/ Class_I_seperated/
Tree for Class II Aminoacityl-tRNA Synthetases predicted catalytic domains is available at: /chalmers/users/khorshid/Master_Thesis/Supplementary/Trees/ Class_II_seperated

particular Aminoacityl-tRNA Synthetases' predicted catalytic regions from other types of enzyme in class II. That is very promising for relying on the tree and its topology.

Considering the multiple sequence alignment, there were no common structural or sequence relation found between class I and class II Aminoacityl-tRNA Synthetases catalytic domains. That means, the evolutionary analysis described in the project still left the question of a bifurcated origin of translation machinery among domains of life unanswered.

Another special characteristic of these tree topologies is that they suggest that the emergence of different types of Aminoacityl-tRNA Synthetases catalytic domains predates the origin of divergence of today's organisms.

The topologies show that the Aminoacityl-tRNA Synthetases catalytic domain has been remained conserved throughout evolution while there are substantial differences in other parts of the Aminoacityl-tRNA Synthetases structure in diverse organisms. Figure 6.1 and 6.2 shows this conclusion in both classes.

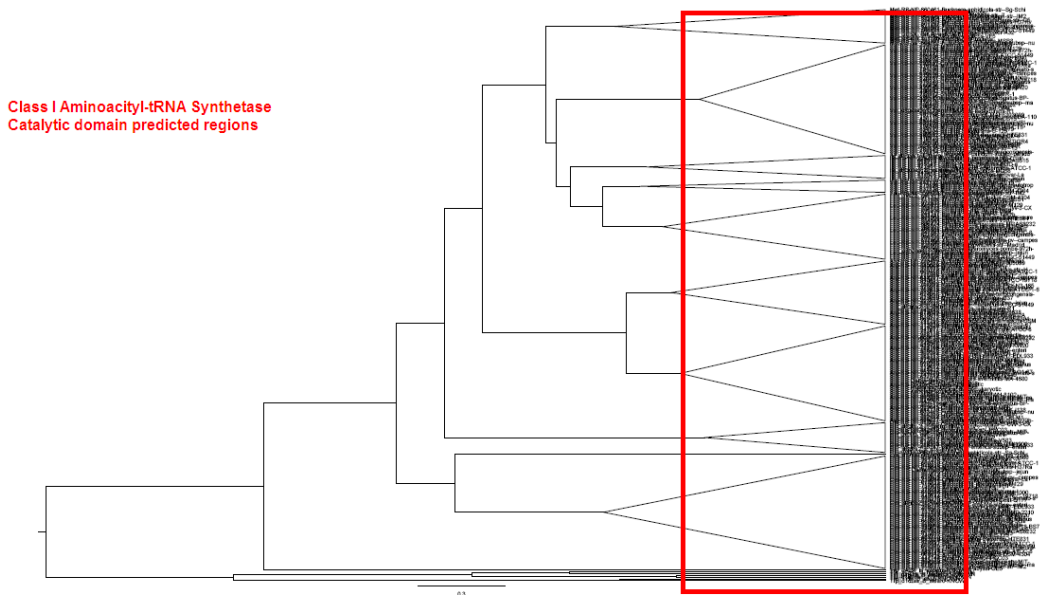

Figure 6-1 Phylogenetic tree for class I. The red box is suggesting the times that in the beginning, all Aminoacityl-tRNA Synthetases predicted catalytic domain have been specified and then the different organism started to diverge.

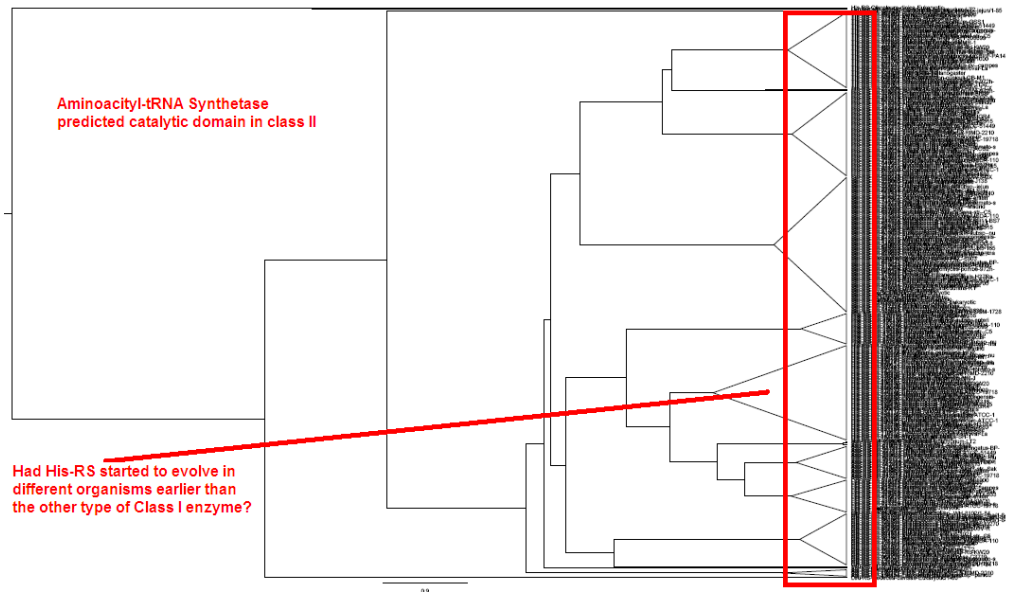

Figure 6-2 Phylogenetic tree for class II Aminoacityl-tRNA Synthetases predicted catalytic domain. The red box is suggesting the times that in the beginning, all Aminoacityl-tRNA Synthetases catalytic domain have been specified and then the different organism started to diverge. In class II, His-RS looks that started to develop earlier than all other type of enzymes in his class

The red boxes in Figures 6.1 and 6.2 highlight the time of divergence for organisms is almost the same in all different types of Aminoacityl-tRNA Synthetases in each class. Although His-RS in class II looks strange.

The next step is to look at each type separately then it might give us more clear view for each clade. Therefore, the analysis concentrated on each clade that is discussed in more details in the following subsections in sections 6.1.1 and 6.1.2.

6.1.1 Aminoacityl-tRNA Synthetase in Class I

Let us look at the tree for Class I containing all type of Aminoacityl-tRNA Synthetases' predicted catalytic domains. In the following, the results from each type were discussed separately. It is suggested to look at the tree in this section available online[19] or the corresponding trees to each type, Figures 5.4-9:

- **Trp-RS & Tyr-RS:**

In the trees, these Trp-RS and Tyr-RS form a well-separated clade except for the Trp-RS from *Homo sapiens* is an outlier. Perhaps Trp-RS from *Homo sapiens* has not been modeled properly because there were few known structures available. Ideally, we would like to find others of these Aminoacityl-tRNA Synthetases to align and redo the process.

[19] /chalmers/users/khorshid/Master_Thesis/Supplementary/Trees/ class_I_max_creadibility.pdf

- **Glu-RS & Gln-RS:**

Glu-RS structures also form a neat clade. The THREADER results suggest that these should mostly be modeled based on 2CFO chain A and some based on 1GLN chain A. The clade does not follow the canonical pattern in phylogenetics. Most of Glu-RS taxa are from the Bacteria domain except a Glu-RS *Arabidopsis Thaliana* from Eukaryotes. However, the posterior probability for this clade is not solid (0.23 * 0.4 =0.09) which does not provide strong evidence for inferring whether they follow canonical phylogenetic pattern among three domains of life. Therefore, Glu-RS *Arabidopsis Thaliana* is not well positioned. On the other hand, almost Glu-RS from Bacteria from the same family are close to each other.

Regarding Gln-RS, perhaps it would be a good idea to try a new technique for modeling Gln-RS to get some more structures since there are just four structures picked for BEAST analysis.

- **Arg-RS:**

The predicted structures show a trim clade. The modeling was successful for this type perhaps because more structures were available. The clade also shows very nice grouping of close related organisms. They are also supported with relatively high posterior probability.

- **Cys-RS**

These form a nice separate clade. The relationship of Cys-RS with the other types is unclear (branch confidence only 0.065). O'Donoghue et al. [10] have grouped Cys-RS differently by considering known Cys-RS structure.

- **Met-RS , Ile-RS**

Both Met-RS and Ile-RS show a trim clade. In Ile-RS, the structures are mostly predicted based on known structures 1IVS chain A, 1Qu3 chain A or 1JZS chain A. Perhaps that is the reason they are grouped together.

- **Val-RS**

There are two clades of Val-RS, but this had already been found by Woese et al. [1]. They noticed that the archaean enzymes were well separated. We find that the archaean enzymes are closer to Leu (and Ile and Met) than they are to the rest (bacterial and eukaryotic) of the Val-RSs. For this reason, Val-RS would be divided into 2 subgroups separated from each other , but I think it is better to recognize the two clades than to force the known structure from *Thermus thermophilus* into an artificial position more like the archaean position (as I think has probably happened in Fig. 9 of O'Donoghue[10]). Actually, this could be of considerable significance and may mean that the Val functionality has appeared under 2 different circumstances, once as an early split from a Val/[Met/Ile/Leu] ancestor (in bacteria and eukaryotes) and once as a late split from a Val/Leu ancestor (in Archaea).

- **Lys-RS(Class I)**

We do need to find some more known structures of Lys-RS in order to be able to predict structures using THREADER and MODELLER or perhaps other structure prediction techniques.

6.1.2 Aminoacityl-tRNA Synthetase in Class II

Let us look at the tree for Class II containing all type of Aminoacityl-tRNA Synthetases predicted catalytic domains. In the following, the results from each type were discussed separately. It is suggested to look at the tree in this section available online[20] and Figures 5.10-18:

- **Gly-RS:**

Gly (alpha-2 motif): The Gly-RS from *Thermus thermophilus* we find in together with Thr-RS, Pro-RS and Ser-RS and close to His-RS. O'Donoghue et al. [10] have found the same results. It would be nice to find some other Gly-RS structures (or sequences) aligned. We have the same problem as with the Trp-RS and Tyr-RS in Class I.

Gly (alpha-beta-2 motif): The threading results suggest that these are constructed based on 1J5W chain B. Therefore, we could infer that all of the organisms in this clade have the alpha-beta-2 form of the enzyme.

- **ASP-RS, ASN-RS, Ala-RS:**

Regarding Ala-RS, the clade at the bottom of the dendogram looks trim, neat and separated from other types of Aminoacityl-tRNA Synthetases in class I. They grouped around the known structures 1RIQ chain A and 1YFS chain A in the clade.

Regarding Asn-RS, all the "unknown" structures seem related to the *Thermus thermophilus* (d1efwa3) and *E. coli.* (d1c0aa3) structures. The other three structures form a more distant subclade is surprising, because (d1asza2) is from *Thermus thermophilus*, the same as d1efwa3 except that it has the tRNA attached in the crystal structure. Therefore, it is unknown why d1asza2 forms a distant subclade.

- **Thr-RS & His-RS & Phe-RS:**

His-RS and Thr-RS form a well-separated clade around their own known structures. Phe-RS also shows a trim and cut-off clade but it would be useful to have more known structures from different domains of life.

- **Lys-RS(class II)**

These form an isolated clade. There was just one known structure for Lys-RS, Lys_d1e1oa from *E-Coli*. Therefore, the results are biased towards this structure. For Lys-RS (class I) we have the same problem regarding lack of known structures so we could not have any predicted structures.

- **Ser-RS**

These form a trim and isolated clade, but the known structure 2CJA chain B from *M. barkeri* is a wide outlier. The threading results suggest that they are mostly based on

[20] /chalmers/users/khorshid/Master_Thesis/Supplementary/Trees/ class_II_max_creadibility.pdf

1WLE Chain B and/or 1SER, 1SRY and 1DQ3 PDB files. Perhaps that is the reason for 2CJA B to become an outlier.

7 Conclusions and Future Work

7.1 Conclusions

To sum up the discussion, some conclusions and inferences about the applied methodologies and the results are listed in the following:

Considering the trees in each class, it could be concluded that 3D structure protein modeling seems to be successful for the Aminoacityl-tRNA Synthetases catalytic domain identification because different types of Aminoacityl-tRNA Synthetases were separated form each other and constructed a cluster for their own.

In general, phylogenies of catalytic domain Aminoacityl-tRNA Synthetases follow the "canonical pattern" for three domains of life: (Eukaryotic, (Bacteria, Archaea)). There are some types did not follow the pattern that could be because of horizontal gene transfer phenomena suggested in previous works.

Another special characteristic of these tree topologies is that they suggest that the emergence of different types of Aminoacityl-tRNA Synthetases catalytic domains predates the origin of divergence of today's organisms.

The topologies show that the Aminoacityl-tRNA Synthetases catalytic domain has remained conserved throughout evolution while there are substantial differences in other parts of the Aminoacityl-tRNA Synthetases' structure in diverse organisms

Considering the multiple sequence alignment, there were no common structural or sequence relation found between classes I and class II Aminoacityl-tRNA Synthetases catalytic domains. That means, the evolutionary analysis described in the project still left the question of a bifurcated origin of translation machinery among domains of life unanswered.

7.2 Future Work

Several ways to extend this project are suggested here.

When more Aminoacityl-tRNA Synthetases' structures have been determined experimentally, especially for those types where only a few (or no)experimentally determined structure are available today, the steps performed in this project should be repeated. This will enable us to make more, and better, resulting phylogenetic analysis.

Using different methods of structural alignment to prepare structural superposition of catalytic domain might lead to longer alignments. A longer multiple sequence alignment containing such a common structural core will make the multiple sequence alignment of catalytic domains longer that will help better evolutionary analysis.

It is suggested to use the use Molecular Clock for more accurate trees in BEAST better understanding in BEAST because these sequences are ancient and it might be unrealistic to consider a fix substitution rate for the whole analysis because of their low sequence identity.

One could think of using faster method of Bayesian Inference. One can think of using Machine Learning methods for speeding up BEAST tree search it could help BEAST how to change the branches intelligently in the tree search that will affect the time complexity and could help to do the analysis with larger number of taxa.

REFERENCES

1. Woese Carl R., Olse Gary J., Ibba Michael and Soll Ditter, (March 2000) "*Aminoacyl-tRNA Synthetases, the Genetic Code, and the Evolutionary Process*" Microbiology and Molecular Biology Reviews, p. 202-236, Vol. 64, No. 1
2. Eswar N., Marti-Renom M. A., Webb B., Madhusudhan M. S., Eramian D., Shen M., Pieper U. and Sali A. (2000) "*Comparative Protein Structure Modeling With MODELLER. Current Protocols in Bioinformatics"* John Wiley & Sons, Inc., Supplement 15, 5.6.1-5.6.30
3. Holm L and Sander C. (1993) "*Protein structure comparison by alignment of distance matrices*". J. Mol. Biol., 233, 123–138.
4. Michener, C. and Sokal, R. (1957) "*A quantitative approach to a problem in classification. Evolution*" 11:130—16
5. Sneath, P. and Sokal, R. (1973) "*Numerical Taxonomy*". Freeman, San Francisco.
6. Saitou, N. and Nei, M. (1987) "*The neighbor-joining method: a new method for reconstructing phylogenetic trees*" Mol Biol Evol, 4(4):406--425.
7. Felsenstein, J. (1981) *"Evolutionary trees from DNA sequences: a maximum likelihood approach*". J Mol Evol, 17(6):368--376
8. Larget, B. and Simon, D. (1999) *"Markov chain monte carlo algorithms for the bayesian analysis of phylogenetic trees*". Mol Biol Evol, 16:750--759.
9. Woese C, Kandler O and Wheelis M. (1990)*"Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eukarya."* Proc Natl Acad Sci USA 87 (12): 4576-9. PMID 2112744,
10. O'Donoghue P. and Luthey-Schulten Z. (Dec. 2003) "*On the Evolution of Structure in Aminoacyl-tRNA Synthetases"* Microbiology and Molecular Biology Reviews, p. 550-573, Vol. 67, No. 4
11. Russell, R. B. and Barton G. J. (1992) "*Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confi-dence levels*" Proteins Structure Function Genet. 14:309–323
12. Zhang Y. and Skolnick J. (2005) "*TM-align: a protein structure alignment algorithm based on the TM-score*" Nucleic Acids Research, Vol. 33, No. 7: 2302-2309
13. Shindyalov I.N. and Bourne P.E. (1998) "*Protein structure alignment by incremental combinatorial extension (CE) of the optimal path*" Protein Eng., 11, 739–747.
14. Kihara,D. and Skolnick,J. (2003) "*The PDB is a covering set of small protein structures*" J. Mol. Biol., 334, 793–802,
15. Holm,L. and Sander,C. (Nov 1995) "*Dali: a network tool for protein structure comparison"* Trends Biochem Sci.;20(11):478-80
16. Guindon S and Gascuel O. (2003) "*A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood*" Systematic Biology. 52(5):696-704.
17. Drummond Alexei J, Rambaut A 2007 "*BEAST: Bayesian evolutionary analysis by sampling trees*" BMC Evolutionary Biology 7:214
18. Jones, D.T, Taylor, W.R. and Thornton, J.M. (1992) "*A new approach to protein fold recognition*" Nature Magazine. 358, 86-89
19. Maciej Szymanski, Marzanna A. Deniziak and Jan Barciszewski, (2001) "*Aminoacyl-tRNA synthetases database*" Nucleic Acids Res. 29:288-290
    http://rose.man.poznan.pl/Aminoacityl-tRNA Synthetase/

20. H.M.Berman, J.Westbrook, Z.Feng, G.Gilliland, T.N.Bhat, H.Weissig, I.N.Shindyalov, P.E.Bourne (2000) "*The Protein Data Bank*" Nucleic Acids Research, 28 pp. 235-242, http://www.rcsb.org/pdb/home/home.do
21. Christoph Gille (2006) "*Structural interpretation of mutations and SNPs using STRAP-NT*" Protein Sci. 15: 208-210;
22. Entrez Nucleotide database http://www.ncbi.nlm.nih.gov/sites/entrez?db=nucleotide
23. M.-y. Shen and A. Sali. (2006) "*Statistical potential for assessment and prediction of protein structures*". Protein Science 15, 2507-2524,
24. Carl Woese, George E. Fox (November 1977) "*Phylogenetic structure of the prokaryotic domain: The primary kingdoms*" Proc. Natl. Acad. Sci. USA Vol. 74, No. 11, pp. 5088-5090,
25. Kei Takahashi and Masatoshi Nei, (2000) "*Efficiencies of Fast Algorithms of Phylogenetic Inference Under the Criteria of Maximum Parsimony, Minimum Evolution, and Maximum Likelihood When a Large Number of Sequences Are Used*" Molecular Biology and Evolution 17:1251-1258
26. Kihara D. and Skolnick J. (2003) "*The PDB is a covering set of small protein structures."* J. Mol. Biol., 334, 793–802.
27. Levitt M. and Gerstein M. (1998) *"A unified statistical framework for sequence comparison and structure comparison"* Proc. Nat'l Acad. Sci. USA, 95, 5913–5920

ENDORSEMENT

Following the author’s supervisor, Prof. Peter. R. Wills, it is the author's desire that no agency should ever derive military or purely financial benefit from the publication of this report. Authors who cite this work in support of their own are requested similarly to qualify the availability of their results.

Appendix A MODELLER Python Code[21]

In this appendix, the commented source code applied in MODELLER 9v1 is brought.

```
#Work Flow: modeling of a protein sequence with unknown 3D structure with comparative modeling method using MODELLER package
from modeller import *
from modeller.automodel import *
from modeller.scripts import complete_pdb
import sys

log.verbose()    # request verbose output

#assigns arguments to variables target_sequence and model_segments
# Protein sequence name which is available in PIR formatted file
target_file = sys.argv[1]
Aminoacityl-tRNA Synthetases_index = target_file[0:6]
target_seq = target_file[7:(len(target_file)-4)]

#================================================================
# illustrates the SALIGN multiple structure/sequence alignment

env = environ()
env.io.atom_files_directory = './:../atom_files/'

aln = alignment(env)
template_file = sys.argv[2]
my_inputfile = open(template_file, 'r')
#preparing known folds for next steps
known_folds=""
for line in my_inputfile:
    line=line.rstrip('\n')
    code,chain = line.split(' ')
    known_folds= known_folds + " ' " + code + chain + " ',"
    mdl = model(env,
                file=code,
                model_segment=('FIRST:'+chain, 'LAST:'+chain))

    aln.append_model(mdl, atom_files=code,
                     align_codes=code+chain)
known_folds=known_folds.strip(",")
for (weights, write_fit, whole) in (((1., 0., 0., 0., 1., 0.),
False, True),
                                    ((1., 0.5, 1., 1., 1., 0.),
False, True),
                                    ((1., 1., 1., 1., 1., 0.),
True, False)):
    aln.salign(rms_cutoffs=(3.5, 6., 60, 60, 15, 60, 60, 60, 60,
60, 60),
               normalize_pp_scores=False,
               rr_file='$(LIB)/as1.sim.mat',
               overhang=30,
               gap_penalties_1d=(-450, -50),
               gap_penalties_3d=(0, 3),
               gap_gap_score=0,
               gap_residue_score=0,
               dendrogram_file='template_alignment.tree',
```

[21] Available online at: /chalmers/users/khorshid/Master_Thesis/Supplementary/MODELLER CONFIGURATION FILE

```
                        alignment_type='tree', # If 'progressive', the tree is not
                            # computed and all structures will be
                            # aligned sequentially to the first
                feature_weights=weights, # for a multiple sequence alignment only
                            # the first feature needs to be non-zero
                        improve_alignment=True,
                        fit=True,
                        write_fit=write_fit,
                        write_whole_pdb=whole,
                        output='ALIGNMENT QUALITY')

      template_alignment =  'template_alignment'

      aln.write(file= template_alignment + '.pap',
                alignment_format='PAP')
      aln.write(file= template_alignment + '.ali',
                alignment_format='PIR')

      aln.salign(rms_cutoffs=(1.0, 6., 60, 60, 15, 60, 60, 60, 60, 60,
60),
                 normalize_pp_scores=False,
                 rr_file='$(LIB)/as1.sim.mat',
                 overhang=30,
                 gap_penalties_1d=(-450, -50),
                 gap_penalties_3d=(0, 3),
                 gap_gap_score=0,
                 gap_residue_score=0,
                 dendrogram_file='Tree.tree',
                 alignment_type='progressive',
                 feature_weights=[0]*6,
                 improve_alignment=False,
                 fit=False,
                 write_fit=True,
                 write_whole_pdb=False,
                 output='QUALITY')

#=============================================================================
#2D-Aligning of the template multiple alignments with the target sequence
      log.verbose()     # request verbose output
      align_2d_env = environ()

# read topology
      align_2d_env.libs.topology.read(file='$(LIB)/top_heav.lib')
# read parameters
      align_2d_env.libs.parameters.read(file='$(LIB)/par.lib')
# Read aligned structure(s):
      target_aln = alignment(align_2d_env)
      target_aln.append(file= template_alignment + '.ali',
                        align_codes='all')
      aln_block = len(target_aln)
# Read aligned sequence(s):
      target_aln.append(file= target_file,
                        align_codes= target_seq )
# Structure sensitive variable gap penalty sequence-sequence alignment:
      target_aln.salign(output='',
                        max_gap_length=20,
                        gap_function=True,
# To use structure-dependent gap penalty
                        alignment_type='PAIRWISE',
                        align_block=aln_block,
                        feature_weights=(1., 0., 0., 0., 0., 0.),
                        overhang=0,
```

```
                        gap_penalties_1d=(-450, 0),
                  gap_penalties_2d=(0.35, 1.2, 0.9, 1.2, 0.6, 8.6, 1.2, 0.,
0.),
                        similarity_flag=True)
      target_template_align_file = target_file[0:(len(target_file)-4)] +
'_multi_templates_alignment'
      target_aln.write(file= target_template_align_file + '.ali',
                       alignment_format='PIR')
      target_aln.write(file= target_template_align_file + '.pap',
                       alignment_format='PAP')

#=================================================================
# Making Models based on the template_alignment.ali
      log.verbose()   # request verbose output
      modelling_env = environ()

      auto_model = automodel(modelling_env,
                             alnfile=    target_template_align_file    +
'.ali',
                             knowns=(known_folds),
                             sequence= target_seq,
                             assess_methods=(assess.DOPE, assess.GA341))
# Proposing 3 PDB files if possible
      auto_model.starting_model = 1
      auto_model.ending_model = 3
      auto_model.make()

#=================================================================
# Assessment for proposed models of target structure prediction
# Get a list of all successfully built models from a.outputs
      ok_models    =    filter(lambda    x:    x['failure']    is    None,
auto_model.outputs)

# Rank the models by DOPE score
      key = 'DOPE score'
      ok_models.sort(lambda a,b: cmp(a[key], b[key]))

# Get top model
      m = ok_models[0]
      print "Top model: %s (DOPE score %.3f)" % (m['name'], m[key])
```

Appendix B Protein Profile Aligner source code[22]

The commented source code used for superposing the predicted models on superimposed catalytic domains of known Aminoacityl-tRNA Synthetase structures is brought here:

```
/**
 *
 */
import charite.christo.strap.extensions.ClustalW_3D;
import charite.christo.strap.StrapProtein;
import charite.christo.strap.extensions.AlignmentWriterFasta;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

/**
 * @author Mohsen Khorshid
 *
 * Sequence Alignment is one of the basic methods in Bioinformatics and
many different implementations exist.
 * The aim of this code is to provide sequence alignment of some
proteins against a superposed common core structures
(sequence/structure multiple alignment) in a FASTA formatted file
 * Java my_Aligner shows how proteins are aligned using automatic
procedures. The alignment of superposed conserved regions of proteins
will be cached so that it is fixed for all model proteins that you want
to do sequence/structure multiple alignment.
 * It make use of the interface in STRAP called "KISS".
 * for more information about KISS please check (Christoph Gille et al.
2003) KISS for STRAP: user extensions for a protein alignment editor
 *
 * This code is mainly using libraries of "STructural Alignment program
for Protein" or STRAP developed by Christoph Gille
 * Institut für Biochemie, Charité Group for Computational Biochemistry
Group Leader: Prof. H. Holzhütter http://www.charite.de/sysbio
 *
 * If you are going to run this code on a linux machine terminal using
remote connection from a windows machine please note that you might
need to install X-11 servers on your windows machine...e.g. Excess
 *
 * USAGE:
 * >java -cp [file Path for STRAP.JAR]:. my_aligner args[0..4]
 *
       * @param args
       * args[0]=known_path (Path address for the common structural
core' PDB files)
```

[22] Available online at: /chalmers/users/khorshid/Master_Thesis/Supplementary/Aligner

```
       * args[1]=known_filename (a file contains list of your common
structural core PDB format proteins' names e.g.  )
       * args[2]=model_path (Path address for the PDB format proteins
which are going to align against the common structural core)
       * args[3]=model_filename
       * args[4]=destination_path
       *
       *
       *
 */
public class my_aligner {


      public static void main(String[] args) {
// TODO Auto-generated method stub
            String path= args[0];

            List known_list = new ArrayList();
            List models_list = new ArrayList();

            int protein_index=0;
//          reading the PDB file names
            try{
                  String line=null;
                  String protein_list= path+args[1];
                  FileReader file_reader = new
FileReader(protein_list);
                  BufferedReader buffered_reader = new
BufferedReader(file_reader);
                  while((line=buffered_reader.readLine())!= null){
                        known_list.add(path+line);

      System.out.println(known_list.get(protein_index).toString());
                        protein_index++;
                  }
            }
            catch(IOException e){
                  System.out.println(e);
            }
//          Filling the known Proteins in the list
            int pp_size =known_list.size();
            StrapProtein[] pp = new StrapProtein[pp_size];
            for(int i=0;i<known_list.size();i++){
                  pp[i]= StrapProtein.newInstance(new
File(known_list.get(i).toString()));
            }
            String models_path = args[2];
            String model_list_files= models_path+args[3];
            try{
                  String line=null;
                  FileReader models_reader = new
FileReader(model_list_files);
                  BufferedReader buffered_reader = new
BufferedReader(models_reader);
                  while((line=buffered_reader.readLine())!= null)
                        models_list.add(line);
```

```
            }
            catch(IOException e){
            System.out.println(e);
            }
//                Providing the StrapProtein objects array which is
//consist of the Proteins in the PDB files
                  for(int
model_index=0;model_index<models_list.size();model_index++){
//                            adding one Model to the Known PDB set

      System.out.println(models_path+models_list.get(model_index).toStr
ing());
                            pp[pp_size-
1]=StrapProtein.newInstance(new
File(models_path+models_list.get(model_index).toString()));
//                            Make an Instance and set the sequences
                            ClustalW_3D aligner= new
charite.christo.strap.extensions.ClustalW_3D();
                            aligner.setProteins(pp);
//                            start computation
                            aligner.compute();
//                            print the results
                            String
ss[]=aligner.getAlignedSequences();
                            for(int i=0;i<pp_size;i++){

      pp[i].setGappedSequence(aligner.getAlignedSequences()[i]);
                            }
                            for(int i=0;i<ss.length; i++){
                                  System.out.println(ss[i]);
                            }
                            //Configuring the Alignment writer
//settings
                            AlignmentWriterFasta aw = new
AlignmentWriterFasta();
                            aw.setCharForCterminalBlank('-');
                            aw.setCharForNterminalBlank('-');
                            aw.setResiduesPerLine(2500);
                            aw.setProteins(pp);
                            StringBuffer sb=new StringBuffer();
                            aw.getText(sb);
                            System.out.println(sb);
                            try{
                                  FileWriter fw =new
FileWriter(args[4]+models_list.get(model_index).toString()+"ClustalW_3D
_alignment.fasta");
                                  BufferedWriter bw =new
BufferedWriter(fw);
                                  bw.write(sb.toString());
                                  bw.flush();
                            }
                            catch(IOException e){
                                  System.out.println(e);
                            }
                  }
                  System.exit(0);
      }
```

```
}
```

Appendix C Useful information for configuration of BEAST

In Appendix C, declarations of some parameters are taken from BEAST User's Manual in order to configure the package. It is also useful in order to interpret the results.

- ***"…Empirical Substitution Model (Matrix):***

Substitution models describe the process of one nucleotide or amino acid being substituted for another. Such as WAG, mtREV, Dayhoff, JTT, VT, Blosum62, CpREV, RtREV, MtMam, MtArt, HIVb, and HIVw.

- ***Site heterogeneity model***

This allows the refinement of the substitution model to allow different sites in the alignment to evolve at different rates. The "None", "Gamma", "Invariant Sites" and "Gamma + Invariant Sites" options in this menu help explain among site rate heterogeneity within your data.

Selecting "None" specifies a model in which all sites are assumed to evolve at the same rate. For most data sets, this will not be the case, however for some alignments, there is very little variation and the equal rates across sites model cannot be rejected.

Selecting "Gamma" will permit substitution rate variation among sites within your data (i.e., the substitution rate is allowed to vary so that some sites evolve slowly and some quickly). The shape parameter "alpha" of the Gamma distribution specifies the range of the rate variation among sites. Small alpha values (< 1) result in L shaped distributions, indicating that your data has extreme rate variation such that most sites are invariable but a few sites have high substitution rates. High alpha values result in a bell shaped curve, indicating that there is little rate variation from site to site in your sequence alignment. When alpha reaches infinity, all sites have the same substitution rate (i.e., equivalent to "None").

If the analysis concerns protein coding DNA sequences, the estimated gamma distribution will generally be L-shaped. If the codons are however partitioned into 1st, 2nd and 3rd positions, 1st and 2nd will generally have a lower alpha value than the 3rd.

Selecting "Invariant Sites" specifies a model in which some sites in your data never undergo any evolutionary change while the rest evolve at the same rate. The parameter introduced by this option is the proportion of invariant sites within your data. The starting value of this parameter must be less than 1.0, or BEAST will fail to run. Finally, selecting "Gamma and Invariant Sites" will combine the two simpler models of among-site rate heterogeneity so that there will be a proportion of invariant sites and the rates of the remaining sites are assumed to be distributed.

• ***Priors panel***

The Priors panel allows the user to specify informative priors for all the parameters in the model. It can be useful because relevant knowledge such as fossil calibration points within a phylogeny can be incorporated into the analysis. However when no obvious prior distribution for a parameter exists, it is your responsibility to ensure that the prior selected is not inadvertently influencing the posterior distribution of the parameter of interest.

Choosing the appropriate prior is important. When sequences have been collected from a population where Individuals from every area of the population can be mutually selected to produce offspring then various coalescent tree priors that can be used to model the population size changes through time. All the demographic models are parametric priors on the ages of nodes in the tree, in which the hyper-parameters (e.g., population size and growth rate in the case of the exponential growth model) can be sampled and estimated.

The Yule tree prior assumes a constant speciation rate per lineage. An evolutionary lineage is a sequence of species that form a line of descent, each new species the direct result of speciation from an immediate ancestral species. The Yule prior has a single parameter (yule.birthRate) that represents the average net rate of lineage birth. Under this prior, the branch lengths are expected to be exponentially distributed with a mean of yule.birthRate^{-1}. "[23]

-Taken from BEAST User's Manual

[23] *Taken from BEAST User's Manual*

Appendix D List of PDB files[24]

In this appendix, the list of PDB files for both Class I and Class II are listed.

D.1 List of all known structure entries for Class I in Protein Data Bank

1A8H, 1BS2, 1D2R, 1EUQ, 1EUY, 1EXD, 1F1O, 1F4L, 1F7U, 1F7V, 1FFY, 1FYJ, 1G59, 1GAX, 1GLN, 1GSG, 1GTR, 1GTS, 1H3E, 1H3F, 1H3N, 1I6K, 1I6L, 1I6M, 1ILE, 1IQ0, 1IVS, 1IYW, 1J09, 1J1U, 1JH3, 1JII, 1JIJ, 1JIK, 1JIL, 1JZQ, 1JZS, 1LI5, 1LI7, 1M83, 1MAU, 1MAW, 1MB2, 1MEA, 1MED, 1MKH, 1N3L, 1N75, 1N77, 1N78, 1NNH, 1NTG, 1NYL, 1O0B, 1O0C, 1O5T, 1OBC, 1OBH, 1P7P, 1PFU, 1PFV, 1PFW, 1PFY, 1PG0, 1PG2, 1PYB, 1Q11, 1QQT, 1QRS, 1QRT, 1QRU, 1QTQ, 1QU2, 1QU3, 1R6T, 1R6U, 1RQG, 1SCA, 1TYA, 1TYB, 1TYC, 1TYD, 1U0B, 1U7D, 1U7X, 1UDZ, 1UE0, 1ULH, 1VBM, 1VBN, 1WK8, 1WK9, 1WKA, 1WKB, 1WNY, 1WNZ, 1WOY, 1WQ3, 1WQ4, 1WZ2, 1X54, 1X8X, 1Y42, 1YI8, 1YIA, 1YID, 1ZH0, 1ZH6, 1ZJW, 2A4M, 2AG6, 2AJG, 2AJH, 2AJI, 2AKE, 2AZX, 2BTE, 2BYT, 2CFO, 2CSX, 2CT8, 2CUZ, 2CV0, 2CV1, 2CV2, 2CYA, 2CYB, 2CYC, 2D54, 2D5B, 2DJV, 2DLC, 2DR2, 2DXI, 2G36, 2HGZ, 2HQT, 2HRA, 2HRK, 2HSM, 2HSN, 2HZ7, 2O5R, 2OV4, 2TS1, 3TS1, 4TS1

D.2 List of non-redundant entries of protein data bank for class I

Arg_d1bs2a2_S_cerev, Glu_2CFO_cat_Thermos_elongatus, Leu_d1h3na3_T_therm, Met_d1a8h_2_T_therm, Tyr_2CYC_B_hirokoshii, Arg_d1iq0a2_T_therm, Glu_d1gln_2_T_therm, Lys_d1irxa2_P_horikoshii, Tyr_d1h3ea1_T_therm, Cys_d1li5b2_E_coli, Ile_d1jzsa3_T_therm, Met_1PFV_cat_E_coli, Trp_1YI8_D_radio, Tyr_d1n3la_H_sapiens, Gln_1NYL_cat_E_coli, Ile_d1qu3a3_S_aureus, Met_1RQG_cat_Pyro_abyssi, Trp_2AZX_H_sapiens, Tyr_d4ts1b_B_stearo, Gln_d1gtra2_E_coli, Leu_1WZ2_cat_P_horikohis, Met_2CSX_cat_A_aelicus, Trp_d1d2ra_B_stearo, Val_d1gaxa3_T_therm,

D.3 List of all known structure entries for Class II in Protein Data Bank

1ADJ, 1ADY, 1ASY, 1ASZ, 1ATI, 1B70, 1B76, 1B7Y, 1B8A, 1BBU, 1BBW, 1C0A, 1E1O, 1E1T, 1E22, 1E24, 1EFW, 1EIY, 1EOV, 1EQR, 1EVK, 1EVL, 1FYF, 1FYJ, 1G51, 1GGM, 1H4Q, 1H4S, 1H4T, 1H4V, 1HC7, 1HTT, 1IL2, 1IRX, 1J5W, 1JJC, 1KMM, 1KMN, 1KOG, 1KRS, 1KRT, 1L0W, 1LYL, 1N9W, 1NJ1, 1NJ2, 1NJ5, 1NJ6, 1NJ8, 1NNH, 1NYQ, 1NYR, 1PYS, 1QE0, 1QF6, 1RIQ, 1SER, 1SES, 1SET, 1SRY, 1TJE, 1TKE, 1TKG, 1TKY, 1V4P, 1V7O, 1WDV, 1WLE, 1WNU, 1WU7, 1WWT, 1WXO, 1WYD, 1X54, 1X55, 1X56, 1X59, 1Y2Q, 1YFR, 1YFS, 1YFT, 1YGB, 2AKW, 2ALY, 2AMC, 2CIM, 2CJ9, 2CJA, 2CJB, 2CX5, 2CXI, 2DQ0, 2DQ1, 2DQ2, 2DQ3, 2DU7, 2DXA, 2E1B, 2HGZ, 2HKZ, 2HL0, 2HL1, 2HL2, 2I4L, 2I4M, 2I4N, 2I4O, 2IY5, 2J3L, 2J3M, 2PME, 2PMF

[24] Available Online at:
/chalmers/users/khorshid/Master_Thesis/Supplementary/PDB Files/

D.4 List of non-redundant entries of protein data bank for class II

ALA_1YFS_A, ASN_1NNH_A_Hyp, ASN_1X54_A, ASP_1ASZ_A, ASP_1C0A_A, ASP_1EFW_A, ASP_1N9W_A, ASP_1WYD_A, GLY_1ATI_B, GLY_1J5W_B_, HIS_1ADJ_C, HIS_1KMM_B, HIS_1QE0_A, HIS_1WU7_A, HIS_1WU7_B, LYS_1E10_A, PHE_2ALY_A, PRO_1HC7_A, PRO_1NJ1_A, PRO_1NJ8_A, PRO_2I4L_A, PRO_2J3M_B, SER_1SER_A, SER_1WLE_B, SER_2CJA_A, SER_2CJA_B, SER_2DQ3_A, THR_1EVK_A, THR_1NYR_A, THR_1NYR_B,