APPLICATION OF EFFICIENCY BASED TOOLS TO EXPLORE THE RELATIONSHIP OF PROTEIN SEQUENCE AND STRUCTURE

DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF TECHNOLOGY IN INFORMATION TECHNOLOGY

(BIO-INFORMATICS)

Under the Supervision of
Dr. T. Lahiri
Assistant Professor
IIIT-Allahabad

Submitted By
Anil Kumar
M.Tech. IT (Bio-Informatics)
IIIT-Allahabad
MB200503

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
ALLAHABAD, U.P., INDIA-211002
2007

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY ALLAHABAD

(A University Established under Sec.3 of UGC Act, 1956 vide notification No.F.9-4/99-U.3 Dated 04.08.2000 of the Govt. of India)

(A Centre of Excellence in Information Technology Established by Govt. of India)

Date: _____

WE DO HEREBY RECOMMEND THAT THE THESIS WORK PREPARED UNDER OUR SUPERVISION BY *Mr. Anil Kumar* ENTITLED "Application of efficiency based tools to explore the relationship of protein sequence and structure" BE ACCEPTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF TECHNOLOGY IN INFORMATION TECHNOLOGY (BIOINFORMATICS) FOR EXAMINATION.

प्रज्ञानम् ब्रह्म

_____

(Prof. U. S. Tiwari)

DEAN (ACADEMICS)

Indian Institute of Information Technology,

Allahabad

_____

(Dr. T.Lahiri)

Assistant Professor, Bioinformatics Division,

Indian Institute of Information Technology,

Allahabad

_____

THESIS ADVISOR

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY ALLAHABAD

(A University Established under Sec.3 of UGC Act, 1956 vide notification No.F.9-4/99-U.3 Dated 04.08.2000 of the Govt. of India)

(A Centre of Excellence in Information Technology Established by Govt. of India)

*CERTIFICATE OF APPROVAL**

The foregoing thesis is hereby approved as a creditable study in the area of Information Technology carried out and presented in a manner satisfactory to warrant its acceptance as a pre-requisite to the degree for which it has been submitted. it is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it is submitted.

COMMITTEE ON FINAL EXAMINATION FOR EVALUATION OF THE THESIS _____

_____

_____

*Only in case the recommendation is concurred in

DECLARATION

This is to certify that this thesis work entitled **"Application of efficiency based tools to explore the relationship of protein sequence and structure"** which is submitted by me in partial fulfillment of the requirement for the completion of M.Tech in Information Technology (specialization in Bioinformatics) to Indian Institute of Information Technology, Allahabad comprises only my original work and due acknowledgement has been made in the text to all other material used.

I understand that my thesis may be made electronically available to the public.

Anil Kumar

Acknowledgement

I am highly grateful to the honorable Director, IIIT-Allahabad, **Prof. M. D. Tiwari**, for his ever helping attitude and encouraging us to excel in studies, besides, he has been a source of inspiration during my entire period of M.Tech at IIITA.

I am thankful to **Prof. U.S. Tiwari**, Dean Academics IIIT-Allahabad for providing all the necessary requirements and for his moral support for this dissertation work as well during the whole course of M.Tech.

The most notable source of guidance was my advisor, **Dr. T. Lahiri**, Assistant Professor. I owe him a great deal of thanks for taking me under his wing and allowing me to soak up some of his knowledge and insight. He has not only made us to work but guided us to orient towards research.

It gives me great pleasure to express my deep sense of gratitude and my humble gratefulness to **Prof. Krishna Mishra** for her honest dedication towards our education and carrier and for being with us in various levels of academic pursuits.

I am also grateful to **Dr. C.M. Bhandari** Professor**, Dr. C.V.S. Siva Prasad** Assistant Professor, **Mr. Vikram Katju**, **Mr. Pritish Varadwaj** and **Dr. B. S. Sanjeev**, Faculty Associates, M.Tech. Information Technology (Bioinformatics) program, IIIT- Allahabad for their support and motivation through out my research project work.

I would specially like to thanks **Mr. Ojas Wandalkar**, **Mr. Shrikant Mantri** and **Dr. Parikshit Totawar** for being a continuous source of inspiration during my two year tenure in IIIT-A.

I would like to mention thanks to my friends **Mr. D.V. Singh**, **Dr. Ashutosh Khanna**, **Mr. Manish Mishra** and **Mr. Kapil singh** for not only helping me in studies but also for making this batch a house of learning through their hard work and dedication.

I am also thankful to rest of the classmates for their cooperation during my work. I am also thankful to them for helping me in my project work and also some kind of discussion regarding my work which helps me to understand the concept regarding my work.

This acknowledgement will not complete until I pay my respectful homage to my family especially my parents, whose enthusiasm to see this work complete was an infectious as their inspiration.

Finally, I also acknowledge Ministry of Human Resource and Development (MHRD), Govt. of India for their financial support at IIIT- Allahabad.

Anil Kumar

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

HMM	Hidden markov Model
ANN	Artificial Neural Network
DSSP	Dictionary of Protein Secondary Structure
DNA	Deoxy Ribonucleic Acid
RNA	Ribonucleic Acid
3D	3 Dimensional
SD	Standard Deviation
Q3 Score	Score for three hidden states

1. ABSTRACT

The idea of this project is to study the protein structure and sequence relationship using the hidden markov model and artificial neural network. In this context we have assumed two hidden markov models. In first model we have taken protein secondary structures as hidden and protein sequences as observed. In second model we have taken protein sequences as hidden and protein structures as observed. The efficiencies for both the hidden markov models have been calculated. The results show that the efficiencies of first model is greater that the second one .These efficiencies are cross validated using artificial neural network. This signifies the importance of protein secondary structures as the main hidden controlling factors due to which we observe a particular amino acid sequence. This also signifies that protein secondary structure is more conserved in comparison to amino acid sequence.

2. INTRODUCTION

The information for life of most of the organisms is stored in genes by the four different types of nucleotides. Proteins are the other important macromolecules that do all important tasks in all organisms, like catalysis of biochemical reactions, transport of nutrients, recognition and transmission of signals. In this way genes are the blueprint of life and it can be said that the proteins are the functional unit or the machinery of life. Proteins are made up of amino acids joined in a long stretched chain. This long stretched chain is called as primary sequence, a translation of the genes into 20 types of amino acids. In water, the chain folds up to a specific three-dimensional (3D) structure [1]. The main driving force is the necessity to pack residues for which a contact with water is energetically unfavourable into the interior of the molecule.

The three dimensional structure of a protein determines its function and it is well known that the details of the three dimensional structure are uniquely determined by the specificity of the sequence [2]. In principle it can be said that the code could by cracked by calculating the physico-chemical force fields determining the fold. But the required computer time to calculate the three dimensional structure based on this principle is many orders of magnitude. However, it is of practical importance to know the three dimensional structure.

The exchange of a few amino acids can already destabilise a protein this means that if the length of a protein sequence is N the possible number of protein structures will be 20 raised to power N. But nature has not created this number of variety. Random errors in the DNA level information lead to a different translation of proteins. These 'errors' are the basis of evolution. Mutations resulting in a structural change are not likely to be accepted because of it the protein cannot perform its task, if protein can perform its task the mutation will be accepted.

The evolutionary pressures to conserve [3, 4] function and the discontinuity of the universe of structures have the result that the structure is evolutionarily more conserved than sequence.

The idea in this project is to explore the relationship of protein sequence and structure using statistical methods like hidden markov models and artificial neural network.

To explore the relationship two hidden markov models have been assumed:

- In the first model protein secondary structure has been assumed the main controlling factor to determine the protein sequence.
- In the second model protein sequence has been assumed the main controlling factor to determine the protein secondary structure.

The efficiencies for both the models will be calculated and compared. These efficiencies are cross validated using artificial neural network. If the first model will have greater efficiencies in comparison to second model it will show that protein secondary structure is the main hidden controlling factor to determine the protein sequence thus it will state that protein secondary structure is more conserved than protein sequence. If the second model will have greater efficiencies the conclusions will be just opposite.

3. LITERATURE SURVEY

3.1. INTRODUCTION TO PROTEINS

Proteins are very common in all organisms and are fundamental unit of life. The diversity of protein structure underlies the very large range of their function [5] like enzyme catalysis, transport and storage, coordinated motions, mechanical supports, immune protection, generation and transmission of nerve impulses and control of growth and differentiation.

1. Commonly all the biological reactions are catalyzed by the enzymes. Enzymes are well known to increase the rate of the biological reactions up to the 1 million fold. The number of enzymes identified till today reaches up to several thousands.

2. Proteins play a significant role in transport of the small molecules that are important for the physiological system, for example the transport of the oxygen to the tissues is carried out by the protein hemoglobin. There are many drug molecules that partially bound to serum albumins in the plasma.

3. Muscles are mostly composed of proteins and the mechanism of muscle contraction is mediated by the actin and myosin filaments those are proteins.

4. Proteins also take part to give mechanical support as skin and bone are strengthened by the protein collagen.

5. Proteins are important for the immune system as antibodies are proteins and they are responsible for reacting with specific foreign substances in the body.

6. Some amino acids work as neurotransmitters which transmit electrical signals from one nerve cell to another. Receptors for neurotransmitters, drugs, etc. are

protein in nature acetylcholine receptor is a good example for this which is a protein structure that remains embedded in postsynaptic neurons.

7. Proteins play a critical role in the control of growth, cell differentiation and expression of DNA. For example, repressor proteins may bind to specific segments of DNA. In this way it can prevent the expression and thus the formation of the product of that particular DNA segment. Also, many growth factors and hormones that regulate cell function are proteins such as insulin or thyroid stimulating hormone.

3.2. PROTEIN STRUCTURE

Proteins are macromolecules made up from 20 different L-α-amino acids which fold into a particular three-dimensional structure that is unique to each protein. This three-dimensional structure is responsible for the function of a protein. Thus in order to understand the details of protein functions it is necessary to understand protein structure. Protein structure is discussed in terms of four levels of organization.

Primary structure [6, 7] is the amino acid sequence of its polypeptide chains. Proteins are large polypeptides of defined amino acid sequence. The particular sequence of amino acids in protein is determined by the gene that encodes it. The gene is transcribed into a messenger ribonucleic acid (mRNA) and the mRNA is translated into a protein with the help of ribosome.

Primary structure is also called as the covalent structure of proteins because all of the covalent bonding within proteins defines the primary structure except disulfide bonds. In contrast, the higher level organizations of protein structure like secondary, tertiary and quaternary structure involve mainly noncovalent interactions.

Secondary structure is the spatial arrangement of the polypeptide ignoring the confirmation of the side chains. So secondary structure is local ordered structure brought

about by hydrogen bonding mainly within the peptide backbone. The most common secondary structure elements in proteins are the alpha helix and the beta sheet.

Tertiary structure is the three dimensional structure of the entire polypeptide it can be said the global folding of a single polypeptide chain. Hydrophobic effect is one of the major driving forces in determining the tertiary structure of globular proteins. The polypeptide chain folds in such a way that the side chains of the nonpolar amino acids are hidden within the structure and the side chains of the polar residues are exposed on the outer surface. Hydrogen bonding plays an important role in stabilizing tertiary structure. In some proteins disulfide bonds between cysteine residues are involve in stabilizing the tertiary structure.

Quaternary structure refers to the three dimensional structure of proteins that are composed of two or more polypeptide chains. The spatial arrangement of these subunits is the quaternary structure of the protein. The forces that hold sub-units together are the same weak bonds as those that stabilize the tertiary structure of proteins like van der Waals, hydrogen bonds and salt bridges. The contact region between sub-units resembles the interior of a protein. The sub-units may be identical or non-identical.

The quaternary structure is not required for all proteins to be functional; many proteins may have only secondary or tertiary structure.

3.3. HIDDEN MARKOV MODEL

3.3.1. INTRODUCTION TO HIDDEN MARKOV MODEL

A hidden Markov Model (HMM) [8, 9, 10] is a statistical model where the system being modeled is assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden parameters from the observed parameters. The extracted model parameters can then be used to perform further analysis, for example for prediction of patterns of a system.

In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters of concern. Whereas in hidden Markov model, the state is not directly visible, but variables influenced by the state are visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states.

3.3.2. ELEMENTS OF AN HMM

An HMM is characterized by the following:

(1) N, the number of states in the model. Although the states are hidden, for many practical applications there is often some physical significance attached to the states or to sets of states of the model. Generally the states are interconnected in such a way that any state can be reached from any other state. We denote the individual states as $S = \{S_1, S_2, \ldots S_N\}$, and the state at time t as qt.

(2) M, the number of distinct observation symbols per state, i.e., the discrete alphabet size. The observation symbols correspond to the physical output of the system being modeled. We denote the individual symbols as $V = \{v_1, v_2, \ldots, v_M\}$

(3) The state transition probability distribution $A = \{a_{ij}\}$

where

$a_{ij} = p[q_{t+1} = Sj | qt = Si]$, $1 <= i, j <= N$.

For the special case where any state can reach any other state in a single step, we have $a_{ij} > 0$ for all i, j. For other types of HMMs, we would have $a_{ij} = 0$ for one or more (i,j) pairs.

4) The observation symbol probability distribution in state j, $B = \{b_{j(k)}\}$,

where

$b_{j(k)} = P[v_k \text{ at } t | q_t = S_j],\ 1 <= j <= N$

$1 <= k <= M$

5) The initial state distribution $\pi = \{\pi_i\}$ where

$\pi_i = p[q_l = S_i],\ 1 <= i <= N.$

Given appropriate values of N, M, A, B, and π, the HMM can be used as a generator to give an observation sequence $O=O_1,O_2,....O_T$.

(where each observation Ot is one of the symbols from V, and T is the number of observations in the sequence) as follows:

1) Choose an initial state $q_1 = S_i$ according to the initial state distribution π.

2) Set t = 1.

3) Choose $O_t = v_k$ according to the symbol probability distribution in state S_i, i.e., $b_i(k)$.

4) Transit to a new state $q_{t+1} = S_j$ according to the state transition probability distribution for state S_i, i.e., a_{ij}

5) Set t = t + 1; return to step 3) if t < T; otherwise terminate the procedure.

The above procedure can be used as both a generator of observations, and as a model for how a given observation sequence was generated by an appropriate HMM.
It can be seen that a complete specification of an HMM requires specification of two model parameters (N and M), specification of observation symbols, and the specification of the three probability measures A, B, and π. For convenience, we use the compact notation

$$\lambda = (A, B, \pi)$$

to indicate the complete parameter set of the model.

3.3.3. THE BASIC PROBLEMS FOR HMM

There are three basic problems of interest that must be solved for the model to be useful in real world applications. These problems are the following:

Problem 1: Given the observation sequence $O=O_1,O_2,....O_T$ and a model $\lambda = (A, B, \pi)$, how to efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model.

Problem 2: Given the observation sequence $O=O_1,O_2,....O_T$ and the model λ how to choose a corresponding state sequence $Q = q_1\ q_2\ .\ .\ .\ q_T$ which is optimal in some meaningful sense.

Problem 3: How to adjust the model parameters $\lambda = (A, B, \pi)$, to maximize $P (O|\lambda)$.

Problem 1 is the evaluation problem, namely given a model and a sequence of observations, how do we compute the probability that the observed sequence was produced by the model. We can also view the problem as one of scoring how well a given model matches a given observation sequence. The latter viewpoint is extremely useful. For example, if we consider the case in which we are trying to choose among several competing models, the solution to Problem 1 allows us to choose the model which best matches the observations.

Problem 2 is the one in which we attempt to uncover the hidden part of the model, i.e., to find the "correct" state sequence.

Problem 3 is the one in which we attempt to optimize the model parameters so as to best describe how a given observation sequence comes about. The observation sequence used to adjust the model parameters is called a training sequence since it is used to "train" the HMM. The training problem is the crucial one for most applications of HMMs, since it allows us to optimally adapt model parameters to observed training data-i.e., to create best models for real phenomena.

3.3.4. SOLUTIONS TO THE BASIC PROBLEMS OF HMM

A. Solution to Problem 1.

The Forward-Backward Procedure is used to solve this problem.
Consider the forward variable $\alpha_{t(i)}$ defined as

$$\alpha_{t(i)} = P(O1, O2 \ldots O_t, q_t = S_i|\lambda)$$

i.e., the probability of the partial observation sequence, $O_1, O_2 \ldots O_t$,(until time t) and state S_i at time t ,given the model λ. We can solve for $\alpha_t(i)$ inductively, as follows:

1) Initialization:

$$\alpha_{1(i)} = \pi_i.b_{i(O1)}, \quad 1 <= i <= N.$$

2) Induction:

$$\alpha_{(t+1)(j)} = [\sum_{i=1}^{N} \alpha_t(i).a_{ij}].b_{j(Ot+1)}, \quad 1 <= t <= T-1$$

$$1<=j <= N.$$

3) Termination:

$$P(Q|\lambda) = [\sum_{i=1}^{N} \alpha_{T(i)}]$$

Step(l) initializes the forward probabilities as the joint probability of state S_i and initial observation O_1. The induction step, which is the heart of the forward calculation, is illustrated in fallowing figure.

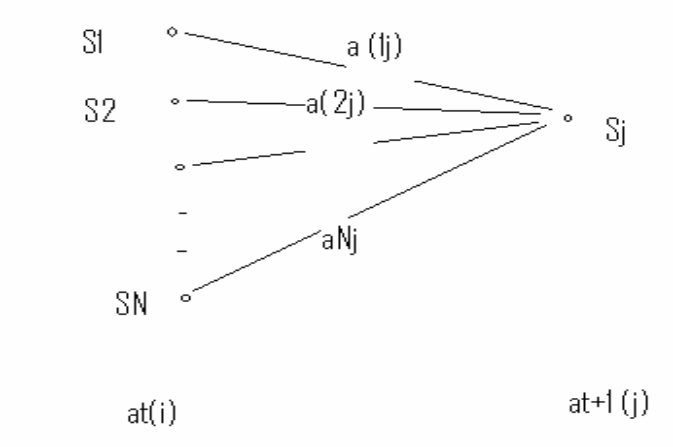


Figure (1) Illustration of the sequence of operations required for the computation of the forward variable $\alpha_{t+1(j)}$ [8].

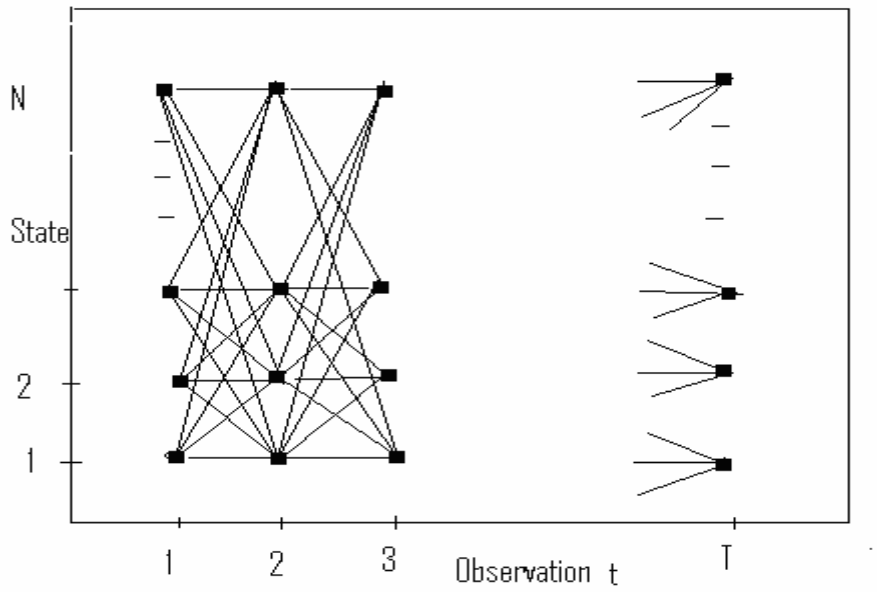


Figure (2) Implementation of the computation of $\alpha_{t\,(i)}$ in terms of a lattice of observations t, and states i [8].

Above figure shows how state S_j can be reached at time t + 1 from the N possible states, S_i, 1 <= i <= N, at time t. Since $\alpha_{t(i)}$ is the probability of the joint event that O_1 , O_2 . . . O_t, are observed, and the state at time t is S_i, the product $\alpha_{t(i)}$. a_{ij} is then the probability of the joint event that are observed, and state S_j is reached at time t + 1 via state S_i at time t. Summing this product over all the N possible states S_i, 1 <= i <= N at time t results in the probability of slat time t + 1 with all the accompanying previous partial observations. Once this is done and S_j is known, it is easy to see that $\alpha_{t+l(j)}$ is obtained by accounting for observation O_{t+1} in state j, i.e., by multiplying the summed quantity by the probability $b_j(O_{t+l})$. The computation of $\alpha_{(t+1)(\ j\ )}$ is performed for all states j, 1 <= j <= N, for a given t; the computation is then iterated for t = 1,2, . . . , T - 1. Finally, step 3) gives the desired calculation of $P(O|\lambda)$ as the sum of the terminal forward variables $\alpha_{T(i)}$. This is the case since, by definition,

$$\alpha_{T(i)} = P(O_1 , O_2 \ . \ . \ . \ O_T, q_T = S_i \mid \lambda)$$

and hence $P(O|\lambda)$ is just the sum of the $\alpha_{T(i)}$'s.

In a similar manner, we can consider a backward variable $\beta_{t(i)}$ defined as

$$\beta_{t(i)} = P(O_{t+1},\ O_{t+2} \ . \ . \ . \ O_T \mid q_t = S_i, \lambda\ )$$

i.e., the probability of the partial observation sequence from t + 1 to the end, given state S_i at time t and the model λ, again we can solve for $\beta_{t(i)}$ inductively, as follows:

1) Initialization:

$$\beta_{T(i)} = 1, \qquad 1 <= i <= N.$$

2) Induction:

$$\beta_{t(i)} = \sum_{i=1}^{N} a_{ij} .b_j(O_{t+1})\ \beta_{t+1(j)}\ ,\ t = T-1,\ T-2,...,1$$

$$1<=j <= N.$$

The initialization step (1) arbitrarily defines $\beta_{T(i)}$ to be 1 for all i. Step(2), shows that in order to have been in state S_i at time t, and to account for the observation sequence from time t + 1 on, we have to consider all possible states S_j at time t + 1, accounting for the transition from S_i to S_j (the a_{ij} term), as well as the observation in state j (the $b_{j(Ot+1)}$ term), and then account for the remaining partial observation sequence from state j (the $\beta_{t+1(j)}$ term).

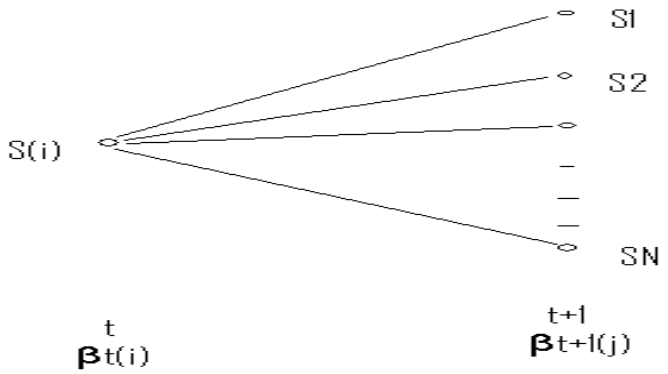


Figure (3) Illustration of the sequence of operations required for the computation of the backward variable $\beta_t(i)$ [8].

B. Solution to Problem 2.

There are several possible ways of finding the "optimal" state sequence associated with the given observation sequence. The difficulty lies with the definition of the optimal state sequence; i.e., there are several possible optimality criteria. For example, one possible optimality criterion is to choose the states qt which are individually most likely. This optimality criterion maximizes the expected number of correct individual states. To implement this solution, we define the variable

$$\gamma_{t(i)} = P(qt = S_i | O, \lambda)$$

i.e., the probability of being in state S_i at time t, given the observation sequence O, and the model λ. Above equation can be expressed simply in terms of the forward-backward variables, i.e.,

$$\gamma_{t(i)} = \alpha_{t(i)}.\beta_{t(i)} / P(O|\lambda) = \alpha_{t(i)}.\beta_{t(i)} / \sum_{i=1}^{N} \alpha_{t(i)}.\beta_{t(i)}.$$

since $\alpha_{t(i)}$ accounts for the partial observation sequence $O_1, O_2, \ldots O_t$, and state S_i at t, while $\beta_{t(i)}$ accounts for the remainder of the observation sequence $O_{t+1}, O_{t+2} \ldots O_T$, given state S_i at t. The normalization factor

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_{t(i)}.\beta_{t(i)}$$

makes $\gamma_{t(i)}$ a probability measure so that

$$\sum_{i=1}^{N} \gamma_{t(i)} = 1$$

Using $\gamma_{t(i)}$, we can solve for the individually most likely state qt at time t, as

$$q_t = \underset{1 <= i <= N}{\text{argmax}} [\gamma_{t(i)}], \qquad 1 <= t <= T$$

Although above equation maximizes the expected number of correct states (by choosing the most likely state for each t), there could be some problems with the resulting state sequence. For example, when the HMM has state transitions which have zero probability ($a_{ij} = 0$ for some i and j), the"optima1" state sequence may, in fact, not even be a valid state sequence. This is due to the fact that the above solution simply determines the most likely state at every instant, without regard to the probability of occurrence of sequences of states.

C. Solution to problem 3.

The third and the most difficult problem of HMMs are to determine a method to adjust the model parameters (A, B, π) to maximize the probability of the observation sequence given the model. There is no known way to analytically solve for the model which maximizes the probability of the observation sequence. In fact, given any finite observation sequence as training data, there is no optimal way of estimating the model parameters. We can, however, choose λ = (A, B, π) such that P (O|λ) is locally maximized using an iterative procedure such as the Baum-Welch method (or equivalently the EM (expectation-modification) method), or using gradient techniques.

3.3.5. APPLICATIONS OF HMMs

HMM's have applications in the many areas of computational biology like gene finding and prediction, Protein- Profile HMMs and Prediction of protein secondary structure.

3.3.5.1. Gene finding and prediction

The gene-prediction HMMs can be used to predict the structure of the gene. Our objective is to find the coding and non-coding regions of an unlabeled string of DNA nucleotides.

The motivation behind this is to assist in the annotation of genomic data produced by genome sequencing methods and to gain insight into the mechanisms involved in transcription, splicing and other processes

A string of DNA nucleotides containing a gene will have separate regions, introns (non-coding regions within a gene) and exons (coding regions). These regions are separated by functional sites start and stop codons and splice sites (acceptors and donors). In the process of transcription, only the exons are left to form the protein sequence.

Many problems in biological sequence analysis have a grammatical structure . HMMs are very useful in modeling grammar. The input to such a HMM is the genomic DNA sequence and the output, in the simplest case is a parse tree of exons and introns on the DNA sequence.

3.3.5.2. Protein- Profile HMMs

Protein structural similarities make it possible to create a statistical model of a protein family which is called a **profile**. The idea is, given a single amino acid target sequence of unknown structure, we want to infer the structure of the resulting protein. The profile HMM is built by analyzing the distribution of amino-acids in a training set of related proteins. This HMM in a natural way can model positional dependant gap penalties. Profile HMM's can also be used for the following purpose:

Scoring a sequence

We can calculate the probability of a sequence given a profile by simply multiplying emission and transition probabilities along the path.

Classifying sequences in a database

Given a HMM for a protein family and some unknown sequences, we are trying to find a path through the model where the new sequence fits in or we are tying to 'align' the sequence to the model. Alignment to the model is an assignment of states to each residue in the sequence. There are many such alignments and the Viterbi's algorithm is used to give the probability of the sequence for that alignment.

Creating multiple sequence alignment

HMMs can be used to automatically create a multiple alignment from a group of unaligned sequences. By taking a close look at the alignment, we can see the history of evolution. One great advantage of HMMs is that they can be estimated from sequences,

without having to align the sequences first. The sequences used to estimate or train the model are called the training sequences, and any reserved sequences used to evaluate the model are called the test sequences. The model estimation is done with the forward-backward algorithm. It is an iterative algorithm that maximizes the likelihood of the training sequences.

3.3.5.3. HMM for protein secondary structure

While applying HMM to predict protein secondary structures from protein sequences, researchers commonly take the amino acid sequence as observed and seek to recover the hidden, secondary protein structure from this sequence.

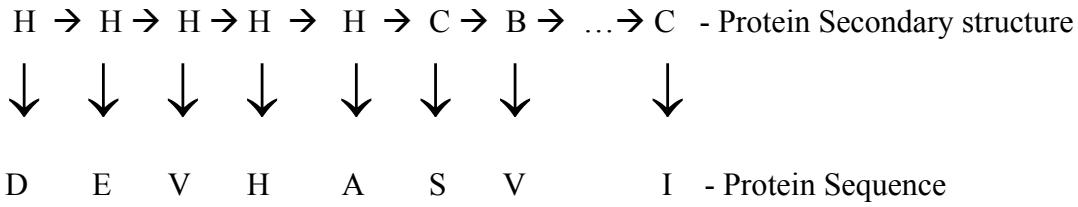


Figure (4) Emission of protein sequence (observed) from protein secondary structure (hidden) [10]

Figure (4) is translating the structure prediction problem into a hidden Markov model. The hidden process (upper line) is the succession of secondary structures: H for a α-helix, B for a β-strand, and C for a coil. The observed sequence, D, E, V, H, A, S, V, I, is the amino acid sequence (lower line). Horizontal arrows symbolize the hidden process's first-order dependence. Vertical arrows indicate the dependence between the observed sequence and hidden process. Successive amino acids are independent given the hidden process.

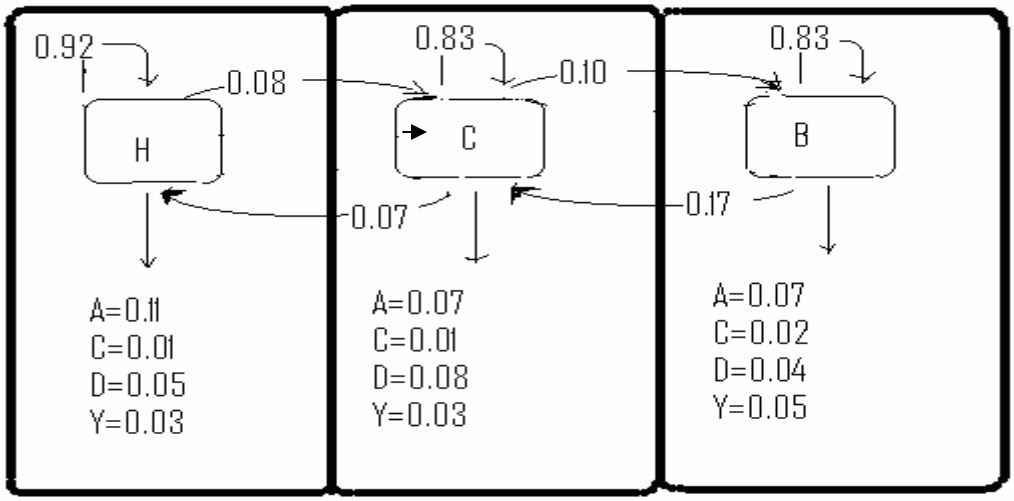


Figure (5) Three-state HMM for secondary-structure prediction [10]

Figure (5) shows a three-state HMM for prediction of secondary structures H (helix), C (coil) and B (beta sheet) from their emission relationship with the amino acids: A, C, D, and Y. The numbers next to each amino acids represent their emission probabilities. The numbers associated with state transitions are given with arrows.

3.3.6. ADVANTAGES OF HMMs

HMM's can accommodate variable-length sequence, because most biological data has variable-length properties, machine learning techniques which require a fixed-length input, such as neural networks or support vector machines, are less successful in biological sequence analysis.

HMM's allow position dependant gap penalties. HMM's treat insertions and deletions is a statistical manner that is dependant on position.

3.3.7. LIMITATIONS OF HMMs

HMM is a linear Model so they are unable to capture higher order correlations among amino-acids. In HMM we take markov Chain assumption of independent events. It means probabilities of states are supposed to be independent which is not true of biology.

In the training problem, we need to watch out for local maxima and so model may not converge to a truly optimal parameter set for a given training set. Secondly, Since the model is only as good as your training set, this may lead to over-fitting.

3.3.8. OPEN AREAS FOR RESEARCH IN HMMS IN BIOLOGY

1. Integration of structural information into profile HMMs: Despite the almost obvious application of using structural information on a member protein family when one exists to better the parameterization of the HMM, this has been extremely hard to achieve in practice.

2. Model architecture: The architectures of HMMs have largely been chosen to be the simplest architectures that can fit the observed data. Is this the best architecture to use? Can one use protein structure knowledge to make better architecture decisions, or, in limited regions, to learn the architecture directly from the data? Will these implied architectures have implications for our structural understanding?

3. Biological mechanism: In gene prediction, the HMM's may be getting close to replicating the same sort of accuracy as the biological machine (the HMM's have the additional task of finding the gene in the genomic DNA context, which is not handled by the biological machine that processes the RNA). What constraints does our statistical model place on the biological mechanism- in particular, can we consider a biological mechanism that could use the same information as the HMM.

3.4. ARTIFICIAL NEURAL NETWORK

3.4.1. INTRODUCTION TO ARTIFICIAL NEURAL NETWORK

The relationship between an amino acid sequence and the structure of the protein it forms is currently unknown. Researchers do not understand the folding process which causes this transformation and have termed this the protein folding problem. An artificial neural network (ANN) approach may be successful in solving the problem by implementing an ANN to predict protein structure from the amino acid sequence [11, 12, 13].

The protein folding problem articulates the challenge of determining how a protein is formed from its primary structure. Proteins are synthesized within the body. They begin as a chain of amino acids. This chain then kinks and folds as complex interactions take place between the molecules. Within seconds, the process is complete and a protein structure has formed. Only a single structure will form from a given amino acid sequence. Thus, the original sequence uniquely determines the resulting structure. How this happens remains unknown to researchers. The rules governing folding remain a mystery. Because we do not understand how protein folding occurs, we must determine a posteriori the protein's structural feature. This information is often much harder to get than the amino acid sequence. The difficulties encountered in determining protein structure have stifled progress in protein engineering and gene therapy research. Our inability to predict protein structure is often viewed as the last major hurdle of molecular biology. With an algorithm for predicting protein folding, we would be in a much better position to harness the benefits of the Human Genome Project and hasten the arrival of intelligent drug design.

As the protein folding problem is such a pressing issue, it has received enormous attention from researchers [14, 15]. Most research has focused on deterministic methods. That is, an algorithm gives the rules for turning an input sequence into an output structure. The research lies in figuring out the rules to be followed.

One strategy for predicting structure is to look at overall trends and deduce from them a general set of rules. For example, we may observe a particular structure and note that it always comes from a class of amino acid sequences. Thus we may deduce that the class of sequences always gives rise to the observed structure. These observations are maintained in a database. When a new sequence is encountered, it is checked against the database for homologies. These homologies then give clues about the final structure.

The other strategy is to try combining various laws of chemistry and physics to predict the folding mechanism. Folding occurs as new chemical bonds are formed and others are broken. At the molecular level, quantum mechanical effects also become important. Thus, folding must occur according to the principles of physics and chemistry. The problem with this strategy is the risk of oversimplifying the folding process. There is still much about quantum effects that we do not understand. Perhaps this is why no algorithm has yet been discovered that successfully predicts structure. An alternative way to predict folding is to use an ANN. Such computer programming paradigms are esteemed for their ability to learn on their own the implicit relationship between input and output data. The network will figure out for itself what rules to follow.

Artificial neural networks draw their inspiration from biological neural networks [16, 17, 18, 19]. The basic computing unit of the brain is the neuron. The neuron gathers inputs through its many dendrites and sends them to the soma.

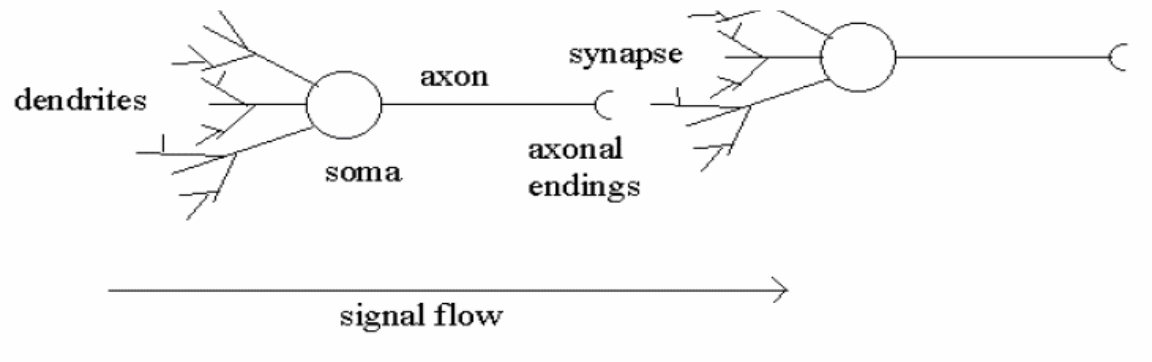


Figure (6) Schematic representation of biological neuron: The dendrites receive the input signal. The soma processes the signal with a nonlinear function. The axon transmits the signal to the axonal endings. The signal is transmitted to the subsequent neuron via the synapse [55]

A nonlinear function is then applied to the signals within the soma. In essence, the soma sums the dendritic inputs together and evaluates the result according to a threshold function, if the result is larger than the threshold an action potential fires. This action potential then propagates away from the soma, down the axon. The output signal leaves the neuron through the axonal endings. This simplified model of the neuron serves as the basis for the artificial neuron.

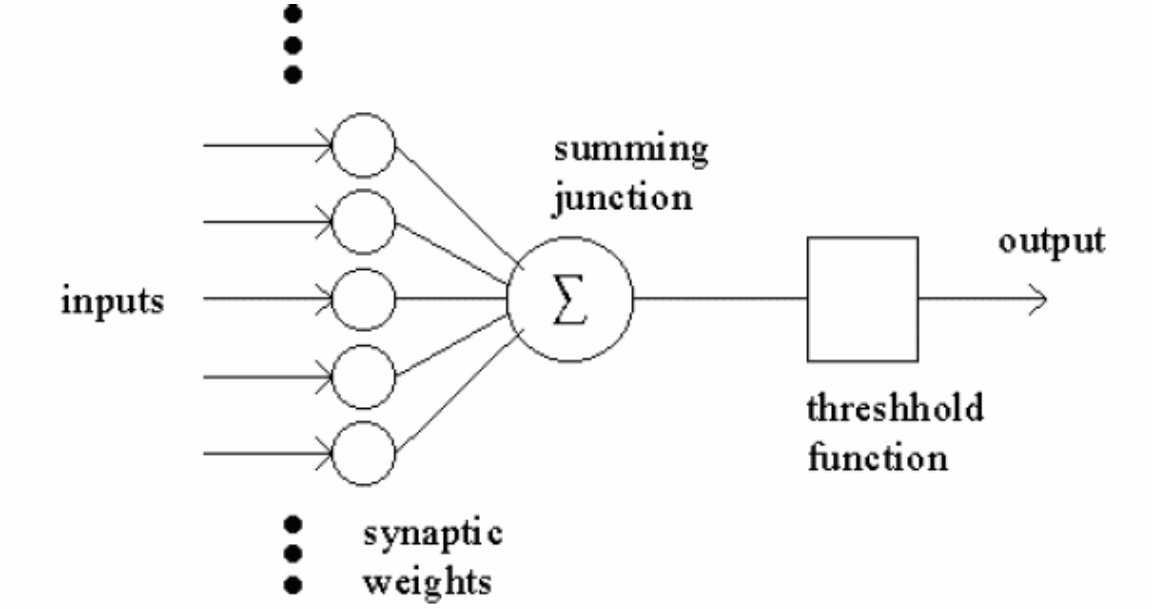


Figure (7) Schematic representation of biological neuron: The artificial neuron preserves the computational function of the biological neuron [55]

The brain is composed of networks of these biological neurons. The output of one neuron serves as input to one or more other neurons. However, the axon of one neuron is not directly connected to the dendrite of another. Rather a gap, known as the synapse, exists between them. There are many factors within the synapse which affect how the signal from the output neuron is transmitted to the input neuron. So while the connection between neurons determines which neurons influence other neurons, the nature of the synapse determines the extent of this influence. The nature of the synapse is constantly changing with each experience the synapse adapts so connections are either strengthened or weakened. Thus, the synapse is believed to be the primary source of learning. The state

of the synapse encapsulates our acquired knowledge and controls our information processing. In an ANN, the synapse is modeled by a weight which indicates how much influence one artificial neuron has on another to which it is connected. Using this scheme the artificial neuron performs a weighted sum of the inputs, compares this to a threshold and outputs the result to other neurons down the line.

For ANNs, there are various architectures in which neurons can be arranged to specify their connections. The neurons are first clumped into layers. The interlayer connections are primarily divided into two groups: feedforward and feedback. A feedforward network has only unidirectional connections. In this way, signals propagate forward from the input layer to the output layer. In feedback networks, a particular layer may be connected to the next layer or any of the previous layers. This arrangement leads to a feedback loop for the signals. We must design the architecture of the network. The number of neurons in each layer must be decided; as well inter- and intra-layer connections must be hardwired. While the connections are hardwired, the weights between neurons can be changed by the network. This changing of weights causes the network to learn a solution to a problem. The optimal weights are determined by the network through training. There are various learning paradigms which guide this process. In a supervised learning scheme, the network is presented with an input and the correct output. The network compares its prediction with the true result and adjusts its weights accordingly. If the network is not presented with the desired output it must learn to organize its weights without being able to measure its predictive success and minimize its error. In such an unsupervised scheme, neurons compete for the opportunity to update their weights, resulting in self-organization.

For the case of protein structural prediction [20, 21, 22] we are primarily concerned with supervised learning [23]. The network is presented with an amino acid sequence and the corresponding structure. The ANN updates its weights as it minimizes the error between its prediction and the true structure. Learning takes place with each new case as the network is trained to predict the protein structure from the amino acid sequence. On its own, the network discovers the underlying rules governing folding. The extent of

learning is tested by supplying the network with a previously unseen sequence and comparing its prediction with the known structure.

To solve the protein folding problem is to know the rules governing folding [24, 25, 26, 27, 28, 29]. A viable approach to the problem must work towards this end. The criteria by which one can measure the capability of their approach to reach this goal are as follows:
Firstly, a solution to the protein folding problem must accurately predict protein structure from an amino acid sequence. This is a challenging problem because of the huge number of forces at work. The electrostatic forces of repulsion and attraction, chemical bonding and quantum effects must all be combined. Thus to solve the problem requires immense computational power and the ability to conceptualize and account for all these factors.
Secondly, the approach must be credible. It must provide a solution that agrees with the basic principles of chemistry and physics, is applicable to all proteins (not just a class), and is independent of those factors not scientifically relevant to the outcome. Essentially, we are asking that the solution comes as a scientific theory: compatible with other truths, broad in scope, and bounded in independent variables.

We can use the above criteria to evaluate if it is even possible to use ANNs to solve the protein folding problem. In what follows, we use these criteria to examine this possibility and to compare ANNs and deterministic algorithms as methods for approaching our problem. To begin this evaluation it is important to note that we have not yet been successful in our efforts to solve the protein folding problem. This suggests that the scope of the problem may be too large for the human brain to conceive (at least at this point in time). In all our efforts to simplify the problem, we disregard critical influences of the folding process. The result is an inaccurate prediction and an unsuccessful algorithm. On the other hand, this is precisely the kind of task at which ANNs excel. ANNs are very effective at combining multiple non-linear factors, such as those affecting folding. They exceed human computing power for detecting trends embedded deep within the data. This exceptional ability to conceptualize allows them to predict the resulting structure with much more accuracy. The inadequacy of ANNs is their inherent dependency on their training set. One neural net, having been trained on certain data, may produce a

prediction quite different from that obtained from another neural net, trained on different data. This poses problems not only for accuracy but also to the credibility of ANNs as tools for scientific research. This is currently the largest drawback for using ANNs. Deterministic approaches can avoid this problem with credibility. However, we should not hasten to dismiss ANNs based on the credibility issue. New research has shown that it is possible to extract useful information from the synaptic weights of the network. If we were able to extract the rules from the network, in this or a similar manner, then we could generalize them to make them compatible with other networks. Once formulated as a deterministic algorithm, the issues of credibility would be obsolete. Thus, with more research in this area, ANNs could be used to conceive a deterministic algorithm for solving the protein folding problem. Speed at which a program executes is determined by the specific design rather than the solution methodology. For deterministic programs, execution speed is proportional to the complexity of the algorithm. Improved accuracy usually comes with increased complexity. Thus, any hope for an accurate deterministic program would be very slow. With ANNs, speed is dependent on architecture. The more layers we have, the more time we need. As well, feedback is much slower than feedforward [30]. An ideal implementation would increase speed while retaining accuracy.

Based on the previous evaluation, the ANN approach is a feasible alternative for addressing the protein folding problem [31, 32, 33, 34]. Although the ANN approach is not without its faults, it remains a strong contender for solving the protein folding problem. Keeping this in mind, we should dedicate ourselves to improving the accuracy and credibility of ANNs. To improve accuracy, we can experiment with various architectures and training sets. As well, we should try to give the network more biological information. Information such as size, electrical properties, chemical reactivity and hydrogen bonding capacity of the amino acids involved may be quite useful to the network.

To address the credibility issue, we must figure out how to extract information (in the form of rules) from the network. These rules should then be reformulated to be

compatible with other networks, applicable to all sequences and compliant with the truths of chemistry and physics. This is a focal point for further research and should be thoroughly explored.

3.4.2. INTRODUCTION TO FEED-FORWARD NETS

Feed-forward nets are the most well-known and widely-used class of neural network. The popularity of feed-forward networks derives from the fact that they have been applied successfully to a wide range of information processing tasks in such diverse fields as speech recognition, financial prediction, image compression, medical diagnosis and protein structure prediction[35, 36, 37 , 38]; new applications are being discovered all the time.

In common with all neural networks, feed-forward networks are trained, rather than programmed, to carry out the chosen information processing tasks. Training a feed-forward net involves adjusting the network so that it is able to produce a specific output for each of a given set of input patterns [39, 40, 41, 42]. Since the desired inputs are known in advance, training a feed-forward net is an example of supervised learning.

3.4.3. THE FEED-FORWARD ARCHITECTURE

Feed-forward networks [43, 44] have a characteristic layered architecture, with each layer comprising one or more simple processing units called artificial neurons or nodes. Each node is connected to one or more other nodes by real-valued weights (parameters), but not to nodes in the same layer. All feedforward nets have an input layer and an output layer. A net with only an input and an output layer is called a single layer net or single layer perceptron (not a two-layer net, because the input layer - which presents a given input pattern to the net but serves no computational function - is not counted).

Feed-forward nets are generally implemented with an additional node - called the bias unit - in all layers except the output layer. Typically the output of each bias unit is 1.0 for

all patterns in the data set. The output *y* of each (non-input) node in the network (for a given pattern *p*) is simply the weighted sum of its inputs, i.e.

$$a_{i,p} = \sum_j w_{ij} y_{j,p}$$

$$y_{i,p} = \int a_{i,p}$$

The squashing function *f(x)*, which is required to be both monotonic and differentiable, is typically the sigmoid or logistic function, given by

$$\int(x) = \frac{1}{1+e^{-x}}$$

3.4.4. TRAINING A FEED-FORWARD NET

Feed-forward nets are trained using a set of patterns known as the training set for which the desired outputs are known in advance - a process known in the neural network literature as supervised learning. Every pattern must have the same number of elements as the net has input nodes, and every target the same number of elements as the net has output nodes. Taken together, a training pattern and its associated target are known as a training pair. Prior to training, the network weights are initialised to small random values. A training algorithm is then used to progressively reduce the total network error by iteratively adjusting the weights. The best-known and simplest training algorithm for feed-forward networks is backpropagation, based on the venerable classical optimisation method steepest descent.

When training a neural network, it is important not to lose sight of the underlying purpose, which is *not* to learn the training set to the highest degree of accuracy. Rather, the aim is to generate a network that is good at classifying patterns similar to, but not

identical to, patterns in the training set - i.e. a network that has the ability to generalise [54]. A considerable amount of neural network research has been concerned with specifying the conditions necessary to generate a network that will generalise well. A variety of factors have been identified, including:

1. *The number of training patterns versus the number of network weights.* If the number of network weights is too large compared with the number of training patterns, there is a risk of *over-fitting*. One 'rule of thumb' asserts that there should be at least 20 times as many patterns at network weights.

2. *The number of hidden nodes.* In case of too few hidden nodes the network will be unable to learn a given tasks and in case of too many its generalisation will be poor. In fact, the 'basic' secondary structure prediction task can be learned by a single layer perceptron, i.e. by a feed-forward net with zero hidden nodes.

3. *The number of training iterations*. In case of too few training iterations network will be unable to extract important features from the training set; and in case of too many net will begin to learn the details of the training set to the detriment of its ability to abstract general features - a process known as over-training.

There are seven target structural classes given in DSSP (Dictionary of Protein Secondary Structure) which include G,H,I,T,E,B,S [G is 3-turn helix (3_10 helix) minimum length 3 residues, H is 4-turn helix (alpha helix) minimum length 4 residues, I is 5-turn helix (pi helix) minimum length 5 residues, T is hydrogen bonded turn (3, 4 or 5 turn), E is beta sheet in parallel and/or anti-parallel sheet conformation (extended strand) minimum length 2 residues, B is residue in isolated beta-bridge (single pair beta-sheet hydrogen bond formation). S is bend (the only non-hydrogen-bond based assignment)]. In DSSP residues which are not in any of the above conformations is designated as ' ' (space).

Both the residues and target classes are encoded in binary format [for example Alanine (A): 0 0 0 0 1 Helix (H): 0 0 1 etc.]. Thus each pattern presented to the network

comprises *n**5 inputs for a window of size *n*. The advantage of this sparse encoding scheme is that it does not introduce an artificial ordering; each amino acid and secondary structure type is given equal weight. The main disadvantage is that it entails a large number of network parameters.

3. 4.5. MEASURING PERFORMANCE

The most popular statistical measure of performance is simply the percentage of correctly classified residues, known as *Q3*.The main problem with *Q3* as a measure is that it fails to penalise the network predictions (e.g. non-helix residues predicted to be helix) or under-predictions (e.g. helix residues predicted to be non-helix).

3.4.6. DRAWBACKS WITH THE BASIC APPROACH

Predictions are based on a limited local context (the window size). No account is taken of non-local factors, yet there is considerable evidence that long-range interactions constrain the formation of protein secondary structure. Predictions are based on a limited amount of biological information. For example, the network is not presented with any evolutionary information or with information about the physico-chemical properties of proteins. No account is taken of what we know about the principles underlying protein structure, e.g. knowledge about what constitutes a 'protein-like' prediction. The predictions are uncorrelated, i.e. each prediction is made in isolation, taking no account of the predictions for neighbouring residues.

4. MATERIALS AND METHODS

Dataset for study:

For our study we have collected protein sequence-structure data set from the Jpred distribution list by the Barton Group at University of Dundee [45]. We have selected 507 data from the database available as free distribution with above group. All sequences in this set have been compared pairwise, and are non redundant to a 5 SD cut-off. For training and testing we have taken 407 and 100 protein sequence-structure pair respectively.

4.1. APPROACH FOR HIDDEN MORKOV MODEL

General approach to build the HMM based first model:

The work has been divided in two models. First model is related with prediction of protein secondary structures [46] from protein sequences and have taken Protein sequence as observed state and Protein secondary structure as hidden state. Second model is related with prediction of protein sequence from protein secondary structures and have taken protein secondary structure as observed state and protein sequence as hidden state.

It has been assumed that N, the number of states in the model. Although the states are hidden, for many practical applications there is often some physical significance attached to the states or to sets of states of the model. Generally the states are interconnected in such a way that any state can be reached from any other state. We denote the individual state as $S = \{S_1, S_2, \ldots\ldots S_N\}$, and the state at time t as q_t.

For first model we have done the following steps for the protein sequence-structure pairs.

Step i) Calculation of initial probability for hidden state.

As transition is occurring among the protein secondary structures (hidden states) so initial probability, π_i for i-th protein secondary structures will be the ratio of " how many times that particular state occurred at first position in the training dataset(m)" to " how many times any state occurred at first position i.e., total number of training protein sequence structure pairs(M) ".

$$\pi_i = \frac{m}{M}$$

where $1 \leq i \leq N$.

Step ii) Calculation of transition probability for training dataset.

Here transition is occurring among the hidden states (protein secondary structure). As we are using DSSP structural symbols [47], the transition matrix for training dataset will be of 8 × 8 matrix.

Transition probability from one state to a particular state will be the ratio of "how many times transition occur from one state, ***i*** to that particular state, ***j*** (n)" to "total transition from one state to any other states (N)"

$$a_{ij} = p(q_t + 1 = S_j \mid q_t = S_i) = \frac{n}{N}$$

for $1 \leq \{i, j\} \leq N$ where N =8 for our case.

Step iii) Calculation of emission probability for training dataset.

Emission is occurring when transition occur among hidden states. As any hidden state can emit any one of the twenty amino acids, so the emission matrix will be of 8 × 20 matrix.

Emission probability of an observed state by a particular hidden state will be the ratio of " how many times that observed state, ***k*** is emitted by that particular hidden state, ***j***" (r) to " total number of emission of any observed state by that particular hidden state. ***j***" (R).

$$b_j(k) = p(v_k \text{ at } t \mid q_t = S_j) = \frac{r}{R}$$

$$\text{for } 1 \le j \le 8 \text{ and } 1 \le k \le 20.$$

Step iv) Calculation of forward, backward and forward-backward variables:

Using transition probability and emission probability we have calculated the forward variable (α_i) and backward variable (β_i) for i-th hidden state within a hidden sequence of state [8]. Then we have calculated the forward-backward variable (γ_i) for i-th hidden state using following formula:

$$\gamma_i = \frac{\alpha_i \beta_i}{\sum_i \alpha_i \beta_i}$$

Step v) Calculation of prediction efficiency:

For comparison of predicted and actual protein secondary structure for the test data, we calculated the number of matches for the predicted protein secondary structures and calculated the efficiency for three hidden states (Q_3 score) for each test data as

$$Q_3 = (N_{match} / N_{total}) \text{ X } 100$$

Total efficiency for test dataset is calculated as the mean of the Q_3 scores.

Second model: Prediction of protein sequences by protein secondary structures

Similarly we designed a model for protein sequence prediction [48] from protein secondary structures. In this model protein secondary structures and sequences are taken as observed and hidden states respectively.

Efficiencies of both the model were compared to explore and compare the conservation of protein sequence and structure.

4.2. APPROACH FOR ARTIFICIAL NEURAL NETWORK

General approach to build the ANN based first model:

Step1) Representation of sequence and secondary structures into binary format:

We converted sequence and secondary structures into a binary number format. To represent 20 amino acids at least 5 binary digits are needed. For example, if the number of residue in a sequence is 10, it will create a 10 by 5 matrix of binary digits whose each row signifies each residue.

To represent eight secondary structure elements at least 3 binary digits are needed. For example, if the number of conformation in a secondary structure sequence is 10, it will create a 10 by 3 matrix of binary digits whose each row signifies each conformation (`'GHITEBSU'`).Binary format of sequence and secondary structure were used as input and out put respectively to train the feed forward net.

Step 2) Training feed-forward network to get adjusted weights:

We implemented weight-updating following feed-forward ANN architecture and using sigmoidal function y = 1/ (1+exp (-a)); where a is the net at the output nodes of the network Weight adjustment will be following 200 iterations per epoch i.e., per residue position. For example, for a sequence of size 40 it runs 40x200 times.

Step 3) Prediction of secondary structure:

Once the adjusted weights will be obtained, adjusted weights and 13 residue window is used to get 3 digit binary numbers. These three digit binary number will give the predicted structure.

Step 4) Calculation of Q3score:

Calculation of the Q3score (i.e., the percentage of matched structural motifs) given test structure and resultant structure using the following formula

$$Q_3 = (N_{match} / N_{total}) \text{ X } 100$$

General approach to build the ANN based second model:

In the second model binary format of the secondary structure is used as input and sequence is used as output. Using the adjusted weights given by feed forward net test sequence is predicted after this calculation of Q3score will be done.

Once we got the efficiency results for artificial neural network, the conclusions made on the basis hidden markov model is cross checked if both the model are supporting for the same conclusions.

5. RESULTS AND DISCUSSIONS

Hidden Markov Model

Model-1: Prediction of protein secondary structures from protein sequences.

Observed state: Protein sequence

Hidden state: Protein secondary structure

Table-1: Efficiencies of HMM model 1 for different training and test dataset

Training dataset index	Test dataset index	Efficiency
101 to 507	1 to 100	47.0800
1 to 100 and 201 to 507	101 to 200	46.7600
1 to 200 and 301 to 507	201 to 300	45.9400
1 to 300 and 401 to 507	301 to 400	46.5500
1 to 400 and 501 to 507	401 to 500	44.8800

Model-2: Prediction of protein sequence from protein secondary structures.

Observed state: Protein secondary structure

Hidden state: Protein sequence

Table-2: Efficiencies of HMM model 2 for different training and test dataset

Training dataset index	Test dataset index	Efficiency
101 to 507	1 to 100	13.1000
1 to 100 and 201 to 507	101 to 200	12.8600
1 to 200 and 301 to 507	201 to 300	13.3100
1 to 300 and 401 to 507	301 to 400	13.6300
1 to 400 and 501 to 507	401 to 500	13.6600

Artificial neural network

Model-1: Prediction of protein secondary structures from protein sequences.

Input: Protein sequence

Out put: Protein secondary structure

Table-3: Efficiencies of ANN model 1 for different training and test dataset

Training dataset index	Test dataset index	Efficiency
101 to 507	1 to 100	42
1 to 100 and 201 to 507	101 to 200	41
1 to 200 and 301 to 507	201 to 300	41
1 to 300 and 401 to 507	301 to 400	42
1 to 400 and 501 to 507	401 to 500	42

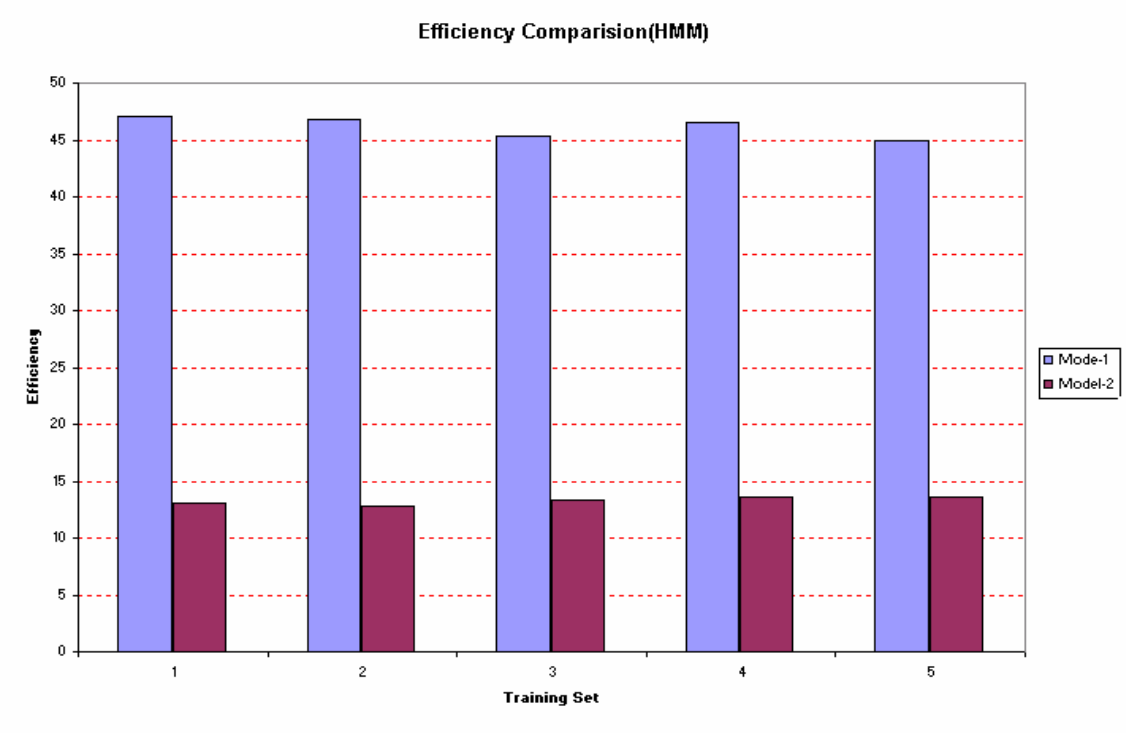
Model-2: Prediction of protein sequence from protein secondary structures.

Input: Protein secondary structure

Output: Protein sequence

Table-4: Efficiencies of ANN model 2 for different training and test dataset

Training dataset index	Test dataset index	Efficiency
101 to 507	1 to 100	10
1 to 100 and 201 to 507	101 to 200	10
1 to 200 and 301 to 507	201 to 300	10
1 to 300 and 401 to 507	301 to 400	10
1 to 400 and 501 to 507	401 to 500	09



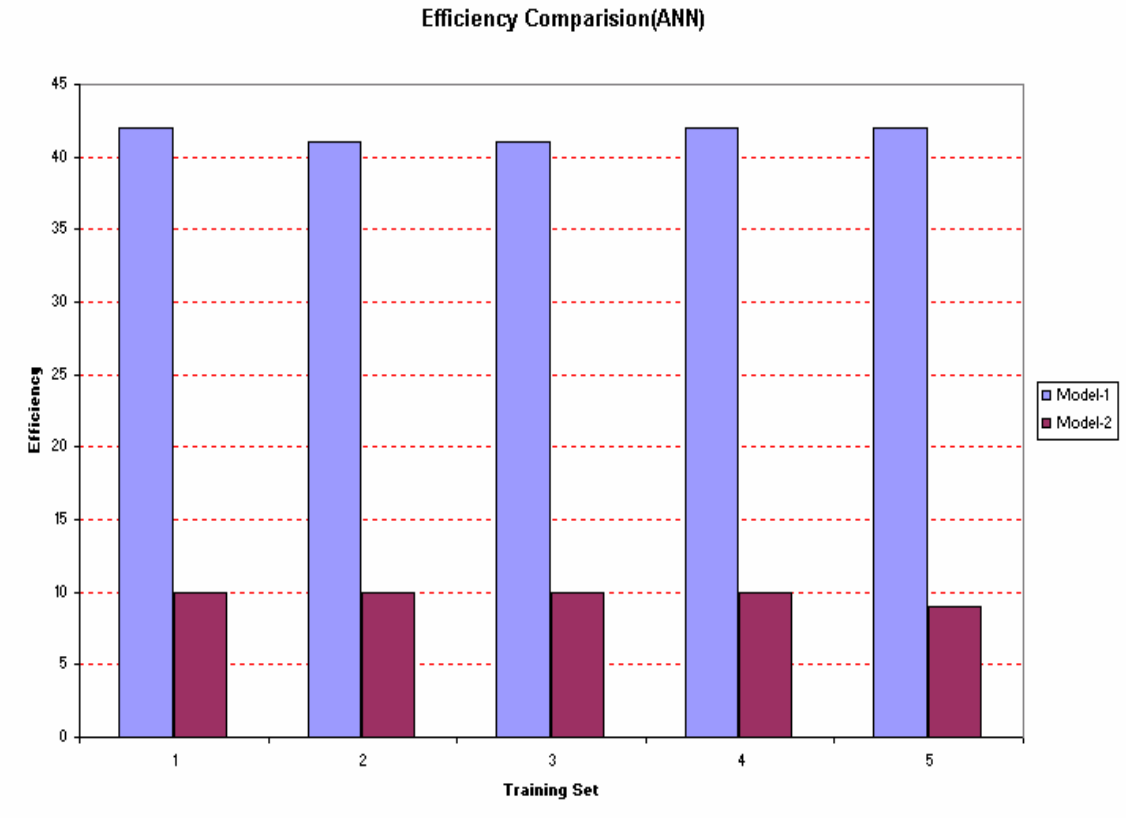


Figure (8) Efficiency comparison of HMM and ANN

From the work on HMM we find that the model designed on the basis of "hidden secondary structure giving rise to emission of observed protein primary structure i.e., its sequence" is giving better efficiency. From the HMM based model we get the following natural inferences:

1) To draw better efficiency in secondary structure prediction we should start with observed or known state as its primary structure i.e., its sequence. That means the protein sequence should be known (i.e., observed) to build such a prediction model.
2) The hidden pattern of protein secondary structure is in some way controlling the emission of observed pattern of protein primary structure (i.e., protein sequence).

To substantiate the above hypothesis, we performed an experiment with the help of simple feed forward ANN model of computing with starting point of two types:

i) Protein sequence is known (or observed in the jargon of HMM) which is fed into the ANN model as input AND protein secondary structure is unknown (or hidden in the jargon of HMM) which is fed into the ANN model as output.
ii) Reverse of the above.

From the work on ANN we find that the first type of starting point is giving significantly better efficiency than its second type – thus confirming the finding we obtained from HMM based model.

The two computing models employed by us are giving us the hint about the fact that "hidden pattern of protein secondary structure is organized earlier than its sequence" is more probable than its reverse one.

6. CONCLUSIONS

The pattern analysis above and the result shown there in above suggest that, the efficiency of the model-1 is better due to the fact that irrespective of the sequence variation the conservation is maintained in structural class of protein. The figure (8) shows the efficiency of both models over five classes of training sets, randomly generated from the total 507 dataset. In each class of training set of size 407, the model-1 outperforms model-2. This result support the fact that hidden pattern of protein secondary structure is organized earlier than its sequence and the patterns in sequence are less conserved in comparison to the structural pattern conservation so sometimes structures from the same family have less than 10% of sequence identity, yet are structurally similar.

Additionally, since structure is more conserved than sequence [49], structural data can be used to reconstruct many of the deeper evolutionary branches that would be difficult or impossible to determine with sequence data alone [50, 51, 52, 53]. This result can be supported in light of the structural constraints on different secondary structure elements, and their role in protein structural stabilization and topology.

7. FUTURE WORK

Only using the evolutionary information it is not possible to make any statistical model which can give enough protein secondary structure prediction accuracy on which we can rely on. Commonly the best accuracy given by these models is up to 70 %. This much accuracy is not good enough to use the predicted secondary structure for further analysis. So our prime object will be to improve the prediction accuracy by feeding the chemical properties of the amino acids to the model.

There can be no doubt that artificial neural networks are very effective at combining multiple non-linear factors, such as those affecting folding. They have been applied successfully to medical diagnosis, financial prediction, and face recognition. Given their success in other areas, it is very likely that with more research, neural nets will prove themselves worthy for protein structural prediction and will solve the protein folding problem.

8. REFERENCES

[1] E. E. Lattman and G. D. Rose, "Protein folding-what's the question?" Proc. Natl. Acad. Sci. U.S.A. 90, 439-441, 1993.

[2] Hin Hark Gan, Rebecca A. Perlow, Sharmili Roy,Joy Ko, Min Wu, Jing Huang, Shixiang Yan,Angelo Nicoletta, Jonathan Vafai, Ding Sun, Lihua Wang, Joyce E. Noah, Samuela Pasquali, and Tamar Schlick, "Analysis of Protein Sequence/Structure Similarity Relationships" Biophysical Journal ,Volume 83,2781–2791, 2002.

[3] Pierre Baldi, Soren Brunik, Paolo Frasconi, Giovanni Soda and Gianluca Pollastri , "Exploiting the past and the future of protein secondary structure prediction"Bioinformatics,Vol.15,937-946, 1999.

[4] Einat Sitbon and Shmuel Pietrokovski 2007, "Occurrence of protein structure elements in conserved sequence regions", BMC Structural Biology, 7:3, 2007.

[5] P. E. Wright and H. J. Dyson, "Intrinsically unstructured proteins: re-assessing the protein structure-function paradigm",J. Mol. Biol. 293 , 321-331, 1999.

[6] Biochemistry by Jeremy M. Berg, John L. Tymoczko, and Lubert Stryer.

[7] Lehninger Principles of Biochemistry, Fourth Edition by David L. Nelson and Michael M. Cox.

[8] Lawrence R. Rabiner February, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition" Proceedings of the IEEE.77 (2), 257–286, 1989.

[9] Juliette Martin, Jean-François Gibrat, and François Rodolphe, "Choosing the Optimal Hidden Markov Model for Secondary-Structure Prediction" Intelligent Systems IEEE. 20(6),19 – 25, Nov.-Dec. 2005.

[10] Juliette Martin, Jean-François Gibrat and François Rodolphe, "Analysis of an optimal hidden Markov model for secondary structure prediction", BMC Structural Biology 2006, 6-25, 2006.

[11] A. Krogh and S. K. Riis, "Hidden neural networks", Neural Comput 11, 541-563, 1999.

[12] C. Pasquier and S. J. Hamodrakas, "An hierarchical artificial neural network system for the classification of transmembrane proteins", Prot. Engin. 12 , 631-634,1999.

[13] A. J. Shepherd, D. Gorse and J. M. Thornton, "Prediction of the location and type of beta-turns in proteins using neural networks", Prot. Sci. 8, 1045-1055, 1999.

[14] P. Baldi, S. Brunak, P. Frasconi, G. Soda and G. Pollastri, "Exploiting the past and the future in protein secondary structure prediction", Bioinformatics 15, 937-946, 1999.

[15] I. Jacoboni, P. L. Martelli, P. Fariselli, M. Compiani and R. Casadio, "Predictions of protein segments with the same aminoacid sequence and different secondary structure: A benchmark for predictive method", Proteins 41, 535-544, 2000.

[16] F. Sasagawa and K. Tajima, "Prediction of protein secondary structures by a

neural network", CABIOS 9 (1993) 147-152.

[17] R. Casadio, P. Fariselli, C. Taroni and M. Compiani, "A predictor of transmembrane a-helix domains of proteins based on neural networks" European Journal of Biophysics submitted, 8/94, 1994.

[18] G. W. Dombi and J. Lawrence, "Analysis of protein transmembrane helical regions by a neural network", Prot. Sci. 3 ,557-566, 1994.

[19] B. Rost and C. Sander, "Conservation and prediction of solvent accessibility in protein families", Proteins 20, 216-226, 1994.

[20] G. J. Barton, "Protein secondary structure prediction", Curr. Opin. Str. Biol., 5 , 372-376, 1995.

[21] B. Rost and C. Sander, "Bridging the protein sequence-structure gap by structure predictions", Annu. Rev. Biophys.Biomol. Struct, 25, 113-136, 1996.

[22] B. Rost, "Protein secondary structure prediction continues to rise", J. Struct. Biol., 134, 204-218, 2001.

[23] B. Rost and S. I. O'Donoghue, "Sisyphus and prediction of protein structure", CABIOS 13, 345-356, 1997.

[24] B. Rost, R. Casadio, P. Fariselli and C. Sander, "Prediction of helical transmembrane segments at 95% accuracy", Prot.Sci. 4 ,521-533, 1995.

[25] B. Rost, R. Casadio and P. Fariselli, "Topology prediction for helical transmembrane proteins at 86% accuracy" Prot. Sci. 5, 1704-1718, 1996.

[26] B. Rost and C. Sander, "Improved prediction of protein secondary structure by use of sequence profiles and neural networks", Proc. Natl. Acad. Sci. U.S.A. 90 , 7558-7562, 1993.

[27] B. Rost and C. Sander, Secondary structure prediction of all-helical proteins in two states. Prot. Engin. 6 ,831-836,1993.

[28] B. Rost and C. Sander, "Combining evolutionary information and neural networks to predict protein secondary structure", Proteins 19, 55-72, 1994.

[29] B. Rost, C. Sander and R. Schneider, "PHD - an automatic server for protein secondary structure prediction" CABIOS 10, 53-60, 1994.

[30] J. D. Hirst and M. J. E. Sternberg, "Prediction of ATP-binding motifs a comparison of a perceptron-type neural network and a consensus sequence method", Prot. Engin., 4, 615-623, 1991.

[31] Chris H.Q. Ding and Inna Dubchak, **"**Multi-class protein fold recognition using upport vector machines and neural networks**"** Bioinformatics Vol. 17, 349-358, 2000.

[32] Eisenhaber F, Frommel C and Argos P, "Prediction of secondary structural content of proteins from their amino acid composition alone. II. the paradox with secondary structural class", Proteins, 25(2), 169-79, 1996.

[33] S. R. Presnell and F. E. Cohen, "Artificial neural networks for pattern recognition in biochemical sequences", Annu. Rev.Biophys. Biomol. Struct, 22, 283-298, 1993.

[34] B. Rost, "PHD: predicting one-dimensional protein structure by profile based

neural networks" Meth. Enzymol, 266, 525-539, (1996).

[35] Rost B, Sander C, "Prediction of protein secondary structure at better than 70% accuracy", Journal of Molecular Biology, 232, 584-599, 1993.

[36] C. H. Wu, "Artificial neural networks for molecular sequence analysis", Comput.Chem., 21, 237-256, 1997.

[37] B. Rost and C. Sander, "Prediction of protein secondary structure at better than 70% accuracy", J. Mol. Biol. 232, 584-599, 1993.

.

[38] B. Rost and C. Sander, "Jury returns on structure prediction", Nature, 360, 540, 1992.

[39] B. Rost, C. Sander and R. Schneider, "Progress in protein structure prediction?", TIBS 18, 120-123, 1993.

[40] B. Rost and C. Sander, "Structure prediction of proteins - where are we now?" Curr. Opin. Biotech, 5, 372-380, 1994.

[41] B. Rost and C. Sander, "Third generation prediction of secondary structure", Meth. Mol. Biol., 143, 71-95, 2000.

[42] X. Zhang, J. P. Mesirov and D. L. Waltz, "Hybrid system for protein secondary structure prediction", J. Mol. Biol., 225, 1049-1063, 1992.

[43] L. Howard Holley and Martin Karplus, "Protein secondary structure prediction with a neural network", Biophysics Vol. 86, pp. 152-156, 1989.

[44] T. W. Barlow, "Feed-forward neural networks for secondary structure prediction" J. Mol. Graph. 13 , 175-183, 1995.

[45] http://www.compbio.dundee.ac.uk/~www-jpred/data/

[46] Pietro Lio, Nick Goldman, Jeffrey L. Throne and David T. Jones, "PASSML: combining evolutionary inference and protein secondary structure prediction", Bioinformatics, Vol.14, 726-733, 1998.

[47] Wolfgang Kabsch and Cristian Sander, "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features" Biopolymers. 22(12), 2577-637, 1983.

[48] Kausik Raha, Andrew M., Wollacott, Michael J.Italia and John R.Desjarlais, "Prediction of amino acid sequence from structure", Protein Science, 9:1106–1119, 2000.

[49] Chothia C, Lesk AM, "The relation between the divergence of sequence and structure in proteins" EMBO J 1986, 5(4):823-826.

[50] O'Donoghue P, Luthey-Schulten Z, "On the evolution of structure in the aminocyl-tRNA synthetases", Microbiol Mol Bio, 67:550-573, Rev 2003.

[51] O'Donoghue P, Luthey-Schulten Z, "Evolutionary profiles derived from the QR factorization of multiple structural alignments gives an economy of information", J Mol Biol, 346(3):875-894, 2005.

[52] Sethi A, O'Donoghue P, Luthey-Schulten Z, "Evolutionary profiles from the QR factorization of multiple sequence alignments", Proc Natl Acad Sci USA,

102(11):4045-4050, 2005.

[53] O'Donoghue P, Sethi A, Woese CR, Luthey-Schulten ZA, "The evolutionary history of Cys-tRNA^{Cys} formation", Proc Natl Acad Sci USA, 102(52):19003-19008, 2005.

[54] Nowlan S.J. and Hinton, G.E., "Simplifying neural networks by soft weight sharng" Neural Computation, 4, 173-193, 1992.

[55] D. Klerfors, "Artificial Neural Networks," [Online document], 1998 Nov, [cited 2001. Apr 10]

9. APPENDIX-A

A SUBSET OF DATA SET (from 1 to 20):

```
seq{1}='MFKVYGYDSNIHKCVYCDNAKRLLTVKKQPFEFINIMPEKGVFDDEKIAELLTKLGRDTQIGL
TMPQVFAPDGSHIGGFDQLREYFK'
str{1}='UEEEEEUUTTTSUUHHHHHHHHHHHHHTTUUEEEEESUSBTTBUUHHHHHHHHHHHHTUSUSSSU
USUEEEUTTSUEEESHHHHHHHTU'

seq{2}='APAFSVSPASGASDGQSVSVSVAAAGETYYIAQCAPVGGQDACNPATATSFTTDASGAASFSF
TVRKSYAGQTPSGTPVGSVDCATDACNLGAGNSGLNLGHVALTFG'
str{2}='UUEEEEEUUSSUUSSUEEEEEEESUUSEEEEEEEUEETTEEUUUTTTUUEEEUUSSUUUEEEE
EUUSEEEEEUTTSUEEEEEETTTSUUEEEEEUSSUUUUUUUUUUU'

seq{3}='TPAFNKPKVELHVHLDGAIKPETILYFGKKRGIALPADTVEELRNIIGMDKPLSLPGFLAKFD
YYMPVIAGCREAIKRIAYEFVEMKAKEGVVYVEVRYSPHLLANSKVDPMPWNQTEGDVTPDDVVDLVNQGL
QEGEQAFGIKVRSILCCMRHQPSWSLEVLELCKKYNQKTVVAMDLAGDETIEGSSLFPGHVEAYEGAVKNG
IHRTVHAGEVGSPEVVREAVDILKTERVGHGYHTIEDEALYNRLLKENMHFEVCPWSSYLTGAWDPKTTHA
VVRFKNDKANYSLNTDDPLIFKSTLDTDYQMTKKDMGFTEEEFKRLNINAAKSSFLPEEEKKELLERLYRE
YQ'
str{3}='UUSUUSUEEEEEEEGGGSUUHHHHHHHHHHHHTUUUSUSSHHHHHHHHSUSSUUUHHHHTTGGG
GTHHHHTTUHHHHHHHHHHHHHHHHHHTTEEEEEEEEUSGGGUSSSUSSUGGGUUUUSUUHHHHHHHHHHHHH
HHHHHHHUUEEEEEEEEETTUTTTHHHHHHHHHHTBTTTEEEEEEESUTTSTTGGGUHHHHHHHHHHHHHT
UEEEEEESSSSUHHHHHHHHHTTUUSEEEEUGGGGGSHHHHHHHHHTTUEEEEUHHHHHHTSSSUTTSUUH
HHHHHHTTUEEEEUUBUHHHHTUUHHHHHHHHHHTTTUUHHHHHHHHHHHHHHTSUUUHHHHHHHHHHHHHH
TU'

seq{4}='SEACPLILDYHVALDNAREKARGAKAIGTTGRGIGPAYEDKVARRGLRVGDLFDKETFAEKLK
EVMEYHNFQLVNYYKAEAVDYQKVLDDTMAVADILTS'
str{4}='UTTUBBUUHHHHHHHHHHHHHHHUSUUUUUSSTTHHHHHHHHHTTUUUBGGGGSUHHHHHHHHH
HHHHHHHHHHHHTSUUUUUUHHHHHHHHHHHHHHHT'

seq{5}='VPSLATISLENSWSGLSKQIQLAQGNNGIFRTPIVLVDNKGNRVQITNVTSKVVTSNIQLLLN
TRNI'
str{5}='UUUHHHHHHHHHHHHHHHHHHHHTTTTTEEEEEEEEUUSSSSUEEEEETTSHHHHHTBUUBUU
GGGU'

seq{6}='TPEMPVLENRAAQGNITAPGGARRLTGDQTAALRNSLSDKPAKNIILLIGDGMGDSEITAARN
YAEGAGGFFKGIDALPLTGQYTHYALNKKTGKPDYVTDSAASATAWSTGVKTYNGALGVDIHEKDHPTILE
MAKAAGLATGNVSTAELQDATPAALVAHVTSRKCYGPSATSQKCPGNALEKGGKGSITEQLLNARADVTLG
GGAKTFAETATAGEWQGKTLREEAEARGYQLVSDAASLNSVTEANQQKPLLGLFADGNMPVRWLGPKATYH
GNIDKPAVTCTPNPQRNDSVPTLAQMTDKAIELLSKNEKGFFLQVEGASIDKQDHAANPCGQIGETVDLDE
AVQRALEFAKKEGNTLVIVTADHAHASQIVAPDTKAPGLTQALNTKDGAVMVMSYGNSEEDSQEHTGSQLR
IAAYGPHAANVVGLTDQTDLFYTMKAALGLK'
str{6}='UUUUUUUUUUSUUSUTTSTTTTUUUSSUUHHHHHHHUUUSUUSEEEEEEETTUUHHHHHHHHH
HHHUTTUUUTTGGGUUEEEEEEUUEEUTTTUSEESSUUHHHHHHHHHHSUUUUTTUBSBUTTUUBUUUHHH
HHHHTTUEEEEEEEEETTSHHHHTTTUUBSUTTUUSHHHHHHHUGGGUTTTTUSUUHHHHHHHHUUSEEEE
EUGGGGGUBBSSGGGBTSBHHHHHHHHTUEEEUSHHHHHHUUUUBTTBUEEEEUUSSSUUUSEEUUUUEET
HHHHSUUBUUEEUTTSUSSSUUHHHHHHHHHHHHHHTUTTUEEEEEEEUHHHHHHHTTUHHHHHHHHHHHH
HHHHHHHHHHHHSSEEEEEEUSSBUSUEEEUTTUUUSSEEEEEEUTTSSEEEEEEUSUSSSSUUUBUUUEE
EEEESTTGGGGSEEEEHHHHHHHHHHHHTUU'
```

```
seq{7}='SAISFHSGYSGLVATVSGSQQTLVVALNSDLGNPGQVASGSFSEAVNASNGQVRVWR'
str{7}='UEEEEUTTSSSEEEEEEUSSUEEEEEESUUUUUGGGTUUSUUUEEEEETTTTEEEEU'

seq{8}='MPPITQQATVTAWLPQVDASQITGTISSLESFTNRFYTTTSGAQASDWIASEWQALSASLPNA
SVKQVSHSGYNQKSVVMTITGSEAPDEWIVIGGHLDSTIGSHTNEQSVAPGADDDASGIAAVTEVIRVLSE
NNFQPKRSIAFMAYAAEEVGLRGSQDLANQYKSEGKNVVSALQLDMTNYKGSAQDVVFITDYTDSNFTQYL
TQLMDEYLPSLTYGFDTCGYACSDHASWHNAGYPAAMPFESKFNDYNPRIHTTQDTLANSDPTGSHAKKFT
QLGLAYAIEMGSATG'
str{8}='UUUUUUHHHHHHHGGGUUHHHHHHHHHHHHTSSUUUTTSHHHHHHHHHHHHHHHHHHTTSTTE
EEEEEUUTTSUUUEEEEEEUUSSEEEEEEEEEEEUUUUSSTTUUTTUUUUUTTTTHHHHHHHHHHHHHHHH
TTUUUSEEEEEEEESUGGGTSHHHHHHHHHHHHTTUEEEEEEEUUUUSUUUSSSSEEEEUSSSUHHHHHHH
HHHHHHHUTTSUEEEEUUSSUUSTHHHHHHTTUUEEUEESSUGGGSUTTTTSTTUUGGGSUTTUHHHHHHH
HHHHHHHHHHHSUUU'

seq{9}='MYGNWGRFIRVNLSTGDIKVEEYDEELAKKWLGSRGLAIYLLLKEMDPTVDPLSPENKLIIAA
GPLTGTSAPTGGRYNVVTKSPLTGFITMANSGGYFGAELKFAGYDAIVVEGKAEKPVYIYIKDEHIEIRDA
SHIWGKKVSETEATIRKEVGSEKVKIASIGPAGENLVKFAAIMNDGHRAAGRGGVGAVMGSKNLKAIAVEG
SKTVPI'
str{9}='UUSSUSEEEEEETTTTEEEEEEUUHHHHHHHUSHHHHHHHHHHHHSUTTSUTTSTTSUEEEEE
UTTTTSSSTTUUUEEEEEEUTTTSSEEEEEEUSSHHHHHHHHTUSEEEEESUUSSUEEEEEETTEEEEEEU
TTTTTUUHHHHHHHHHHHTTUSSUEEEEUUHHHHTTUTTBUEEETTTEEEUSSSHHHHHHHTTEEEEEEEU
UUUUUU'

seq{10}='VAGGGLPKYGTAVLVNIINENGLYPVKNFQTGVYPYAYEQSGEAMAAKYLVRNKPCYACPIG
CGRVNRLPTVGETEGPEYESVWALGANLGINDLASIIEANHMCDELGLDTISTGGTLATAMELYEKGHIKD
EELGDAPPFRWGNTEVLHYYIEKIAKREGFGDKLAEGSYRLAESYGHPELSM'
str{10}='HHHTHHHHHUGGGHHHHHHHTTUUUBTTTTBSUUTTGGGGSHHHHHHHTEEEEEUUTTUSSU
UEEEEEETTTEEEEUUUHHHHHHHHTGGGTUUUHHHHHHHHHHHHHHHTBUHHHHHHHHHHHHHHHHTTSSUH
HHHTTSUUUUTTUTHHHHHHHHHHHTTUTTHHHHTTUHHHHHHTTTUGGGUU'

seq{11}='SQIRHYKWEVEYMFWAPNCNENIVMGINGQFPGPTIRANAGDSVVVELTNKLHTEGVVIHWH
GILQRGTPWADGTASISQCAINPGETFFYNFTVDNPGTFFYHGHLGMQRSAGLYGSLIVDPPQGKKEP'
str{11}='UUEEEEEEEEEEEEEUTTSSUEEEEEETTBSSUUUEEEETTUEEEEEEEEUUSSUUBUEEEE
TUUUTTUGGGSUUBTTTBUUBUTTUEEEEEEEUUSUEEEEEEEUSTTGGGGTUEEEEEEEUSTTUUUS'

seq{12}='FHYDGEINLLLSDWWHQSIHKQEVGLSSKPIRWIGEPQTILLNGRGQFDCSIAAKYDSNLEP
CKLKGSESCAPYIFHVSPKKTYRIRIASTTALAALNFAIGNHQLLVVEADGNYVQPFYTSDIDIYSGESYS
VLITTDQNPSENYWVSVGTRARHPNTPPGLTLLNYLPNSVSKLPTSPPPQTPAWDDFDRSKNFTYRITAAM
GS'
str{12}='SUUSEEEEEEEEEEUSSUHHHHHHHHHSSSUUUUUSUSEEEETTBUUSSSBTTGGGUTTSUB
UUUUSUSTTSUUUEEEUTTUEEEEEEEEUUSSUEEEEEETTBUEEEEEETTEEEEEEEESUEEEUTTUEEE
EEEEUUSUTTSUEEEEEEEESSUUUSUUEEEEEEETTSUTTUUUSSUUUUUUUTTUHHHHHHHHHHUUBUT
TS'

seq{13}='PKPPVKFNRRIFLLNTQNVINGYVKWAINDVSLALPPTPYLGAMKYNLLHAFDQNPPPEVFP
EDYDIDTPPTNEKTRIGNGVYQFKIGEVVDVILQNANMMKENLSETHPWHLHGHDFWVLGYGDGKFSAEEE
SSLNLKNPPLRNTVVIFPYGWTAIRFVADNPGVWAFHCHIEPHLHMGMGVVFAEGVEKVGRIPTKALACGG
TAKSLINNPKNP'
str{13}='UUUUSSUSEEEEEEEEEEETTEEEEEETTEEEUUUSSUHHHHHHTTUTTSSUUSUUUUUUU
TTUUTTSUUUUTTUEEEUUUEEEUTTUEEEEEEEUUUUSSTTUUUUEEEEETTUUEEEEEEEESSUUGGGG
GGSUSSSUUEESEEEEUTTEEEEEEEEUUSUEEEEEEESSHHHHHTTUEEEEEEUGGGUUUUUHHHHUSHH
HHHHHSUUUSUU'

seq{14}='REPRGLGPLQIWQTDFTLEPRAPRSWLAVTVDTASSAIVVTQHGRVTSVAAQHHWATAIAVL
GRPKAIKTDNGSCFTSKSTREWLARWGIAHTTGIPGNSQGQAVERANRLLKDKIRVLAEGDGFKRIPTSKQ
GELLAKAYALNHFER'
```

```
str{14}='UUUSTTUUUUEEEEEEEEUUUUUSUEEEEEEETTTUUEEEEEESSUUHHHHHHHHHHHHHHH
UUUSEEEUUSUHHHHSHHHHHHHHHHTUEEEUUSSSUSUUUUUHHHHHHHHHHHHHHHHHTTUUUUUHHHH
HHHHHHUUTTTUUUU'

seq{15}='TTYADFIASGRTGRRNAIHD'
str{15}='UHHHHHHTSSUUSUUUUUUU'

seq{16}='RDPDAGIDEAQVEQDAQALFQAGELKWGTDEEKFITIFGTRSVSHLRKVFDKYMTISGFQIE
ETIDRETSGNLEQLLLAVVKSIRS'
str{16}='UUUUUUUUHHHHHHHHHHHHHHHHTTTUSUHHHHHHHHHHSUHHHHHHHHHHHHHHSSUGG
GGSUTTTUUHHHHHHHHHHHHHHU'

seq{17}='IPAYLAETLYYAMKGAGTDDHTLIRVMVSRSEIDLFNIRKEFRKNFATSLYSMIKGDTSGDY
KKALLLLCGEDD'
str{17}='HHHHHHHHHHHTSSSSSUUHHHHHHHHHHTTTTTHHHHHHHHHHHHSSUHHHHHHHHUUTHH
HHHHHHHHUSUU'

seq{18}='MRRWFHPNITGVEAENLLLTRGVDGSFLARPSKSNPGDFTLSVRRNGAVTHIKIQNTGDYYD
LYGGEKFATLAELVQYYMEHHGQLKEKNGDVIELKYPLN'
str{18}='UUTTBUSSUUHHHHHHHHHHTUUTTEEEEEEUSSSTTUEEEEEEETTEEEEEEEEEUSSUEE
SSSSSUBSSHHHHHHHHHHSTTUEEBTTUUEEUUUEEUU'

seq{19}='CSVDIQGNDQMQFNTNAITVDKSCKQFTVNLSHPGNLPKNVMGHNWVLSTAADMQGVVTDGM
ASGLDKDYLKPDDSRVIAHTKLIGSGEKDSVTFDVSKLKEGEQYMFFCTFPGHSALMKGTLTLK'
str{19}='UUEEEEUUSSSUUSUSSEEUUSSSSEEEEEEEUUSSUUTTTSUBUUEEEETTTTHHHHHHHH
HHUHHHHSSUSSUTTUSBUUUUBUTTUUEEEEEESSSUUSSUUEEEEUUSTTTTTTSEEEEEUU'

seq{20}='MEVEKEFITDEAKELLSKDKLIQQAYNEVKTSICSPIWPATSKTFTINNTEKNCNGVVPIKE
LCYTLLEDTYNWYREKPIDVYKEFIENSELKRVGMEFETGNISSAHRSMNKLLLGLKHGEIDLAIILMPIK
QLAYYLTDRVTNFEELEPYFELTEGQPFIFIGFNAEAYNSNVPLIPKGSDGMSKRSIKKWKDKVENK'
str{20}='UEEEEEEEUHHHHHHHTTUHHHHHHHHHHHHHHHHUUUBSTTUSSBEEEUSSTTUBUSHHHHH
HHHHHHHHHHHUUEEUUUUSEEEEEEETTEEEEEEEEEUUSUHHHHHHHHHHHHHHHHHTTSUSEEEEEEEUH
HHHTTBUSSUUUHHHHGGGGGGGTTSUEEEEEEEUSEEETTSUUUUUSTTSSUHHHHHHHHHHHHTU'
```