# Project Description

Data Mining Techniques to find the secondary structure of the protein. We have proteins made up of amino acids, using the data mining techniques we are trying to find out the secondary structure through which we can predict the diseases which a person may be getting in the future through inheritance by looking at the secondary structure and is very much useful for drug manufacturing. This is going to be one of the revolution in the future era.

## Aim

The main aim of this current project aims at reducing the risk of inherited diseases in the future generations by predicting the secondary structure from the primary protein sequence. We found using Hidden Markov Model to find the secondary structure of the protein more accurately and we found the secondary structure using the following algorithms they are Forward algorithm and Viterbi algorithm. We used Kth Nearest Neighbor for classification of protein sequences. Now the main aim of this is project is combining the Hidden Markov Model with Kth Nearest Neighbor and to more accurately predict the secondary structure.

## Objective

The main objective is to predict secondary structure with more accuracy by implementing the Hidden Markov Model combined with KNN.

## Applicability

Before any X-ray or NMR structure was known for the family, the prediction of protein secondary structure from an aligned family of proteins has been highlighted by several accurate predictions. Successful secondary structure prediction provides a starting point for direct tertiary structure modeling and provides necessary information for protein folding resides completely within the primary structure. Although the development of advanced molecular biology laboratory techniques such as X-ray crystallography and NMR in silicon prediction methods will narrow the gap between available sequences and structures.

# Introduction

Data Mining Techniques to find the secondary structure of the protein. We have proteins made up of amino acids using the data mining techniques we are trying to find out the secondary structure by converting the 20 different amino acids present in the sequence into three letter secondary structures as alpha, beta sheet, coil and others which means that amino acid cannot be predicted.

## Purpose

The main purpose of this project is to find the secondary structure of the protein with better accuracy for better prediction. This helps in determining several inherited diseases through genes using the mining there by taking proper precautions may help them lead a healthy life. In this project we are trying to improve the accuracy by using different data mining algorithms and then trying to combine them with other algorithms for further accuracy. Here we predicted the secondary structure using Forward and Viterbi algorithms. We classified the protein sequence using Kth Nearest Neighbor there by accurately predicting the Protein secondary structure.

## Scope

Bio molecular structure prediction is the prediction of the three-dimensional structure of a protein from its [amino acid](http://en.wikipedia.org/wiki/Amino_acid) sequence, or of a [nucleic acid](http://en.wikipedia.org/wiki/Nucleic_acid) from its [base](http://en.wikipedia.org/wiki/Nucleobase) sequence. In other words, it is the prediction of secondary and tertiary structure from its primary structure. Structure prediction is the inverse of bio molecular design. Protein structure prediction is one of the most important goals pursued by [bioinformatics](http://en.wikipedia.org/wiki/Bioinformatics) and theoretical chemistry. Protein structure prediction is of high importance in [medicine](http://en.wikipedia.org/wiki/Medicine) (for example, in [drug design](http://en.wikipedia.org/wiki/Drug_design)) and [biotechnology](http://en.wikipedia.org/wiki/Biotechnology) (for example, in the design of novel [enzymes](http://en.wikipedia.org/wiki/Enzymes)). Every two years, the performance of current methods is assessed in the [CASP](http://en.wikipedia.org/wiki/CASP) experiment.

Secondary structure of small nucleic acid molecules is determined largely by strong, local interactions such as [hydrogen bonds](http://en.wikipedia.org/wiki/Hydrogen_bond) and [base stacking](http://en.wikipedia.org/wiki/Base_stacking). Summing the free energy for such interactions, usually using a [nearest-neighbor model](http://en.wikipedia.org/wiki/Nearest-neighbor_thermodynamic_parameters), provides an approximation for the stability of given structure. The most straightforward way to find the lowest free energy structure would be to generate all possible structures and calculate the free energy for it, but the number of possible structures for a sequence increases exponentially with the length of the nucleic acid. For longer molecules, the number of possible secondary structures is enormous.

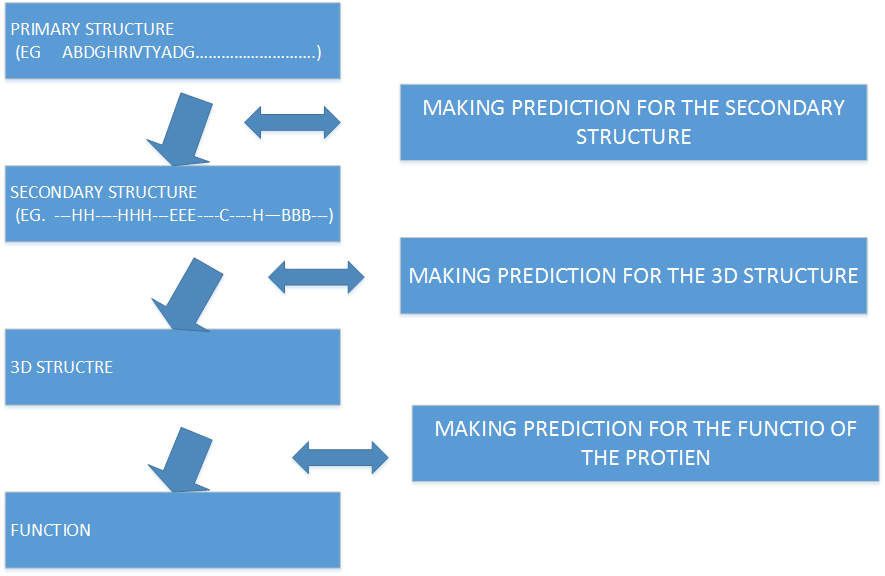


Figure 1: General Diagram

# Problem Statement

Predicting protein structure from amino acid sequence is one of the most important unsolved problems of molecular biology and biophysics. Not only would a successful prediction algorithm be a tremendous advance in the understanding of the biochemical mechanisms of proteins , but , since such an algorithm could conceivably be used to design proteins to carry out specific functions. Prediction of the secondary structure of a protein (alpha-helix, beta-sheet, coil) is an important step towards elucidating its three dimensional structure as well as its function. In this project, we use different Hidden Markov models for predicting protein secondary structure prediction. We have also used Kth nearest neighbor for classifying the proteins and there by accurately predicting the secondary structure of the protein. We have used Hidden Markov model with forward algorithm and Viterbi algorithms.

# Literature Review

A lot of work has been done on predicting secondary structures, and over the last 10 to 20 years the methods have gradually improved in accuracy. Some of the first work on the secondary structure prediction was based on statistical methods in which the likelihood of each amino acid being in one of the three types of secondary structures was estimated from known protein structures. These probabilistic were then averaged in some way over a small window to obtain the prediction. Around 1988 the first attempts were made to use neural networks to predict protein secondary structures. The accuracy of the predictions made by Qian Sjnowski seemed is better than those obtained by previous methods and was reported to be in the range of **62.7-64.4%.** Rost and Sander have developed the prediction mail server called PHD with a prediction accuracy of **71.6%** was reported.

In proposed algorithm instead of constructing One Hidden Markov model for three states of secondary structure elements. We propose three separate Hidden Markov Models i.e. for every secondary structural element we construct One Hidden Markov Model. Every Hidden Markov Model will give a probability. Out of these three probabilities which Hidden Markov Model is giving maximum probability that becomes the secondary structure of that particular amino acid [1].

The simplest model for three-class prediction is a HMM with three hidden states, each state accounting for a secondary structure class. Parameter estimation of such a model is straight forward because the segmentation is fully determined. But the performance of this model is limited**: the Q3 score (proportion of residues with correct prediction) is 58.3%**. **A random prediction gives a Q3 score equals to 34.5%** [2] .

In terms of prediction accuracy, neural networks are among the most popular methods in use today, delivering a point wise prediction accuracy (Q3) of about 77% and a segment overlap measure (SOV) of about 74%. However, to improve the long-term performance of secondary structure pre- diction, it likely will be necessary to develop a cost model that mirrors the underlying biological constraints. While neural networks oﬀer good performance today, their operation is largely opaque. However, the leading HMM methods to date have not exceeded a Q3 value of 75%, and SOV scores are often unreported [4].

Bystroﬀ, Thorsson, and Baker design an HMM to recognize speciﬁc structural motifs and assemble them into protein secondary structure predictions [3]. Using alignment proﬁles, they report an overall Q3 value of 74.3%. Our approach may use fewer parameters, as they manually encode each target motif into a separate set of states. Martin, Gibrat, and Rodolphe develop a 21-state HMM model with 471 parameters that achieves an overall Q3 value of 65.3% (without alignment proﬁles) and 72% (with alignment proﬁles) [21]. Alpha helices are identiﬁed based on an amphiphilic: a succession of two polar residues and two non-polar residues. Won, Hamelryck, Pru¨gel-Bennet and Krogh give a genetic algorithm that automatically evolves an HMM for secondary structure prediction [40, 41]. Using alignment proﬁles, they report an overall Q3 value of 75% (only 69.4% for helices). They claim that the resulting 41-state HMM is better than any previous hand-designed HMM. While they restrict their HMM building blocks to “biologically meaningful primitives”, it is unclear if there is a natural energetic interpretation of the ﬁnal HMM. Schmidler, Liu, and Brutlag develop a segmental semi-Markov Model (a generalization of the HMM), allowing each hidden state to produce a variable-length sequence of the observations [35, 36]. They report a Q3 value of 68.8% without using alignment proﬁles [5].

The ﬁrst algorithm we shall investigate is the k-nearest neighbor algorithm, which is most often used for classiﬁcation, although it can also be used for estimation and prediction. K-Nearest neighbor is an example of instance-based learning, in which the training data set is stored, so that a classiﬁcation for a new unclassiﬁed record may be found simply by comparing it to the most similar records in the training set [6].

# Study of Techniques

## Hidden Markov Model

Hidden Markov models (HMMs) offer a more systematic approach to estimating model parameters. The HMM is a dynamic kind of statistical profile. Like an ordinary profile, it is built by analyzing the distribution of amino acids in a training set of related proteins. However, an HMM has a more complex topology than a profile. It can be visualized as a finite state machine.

***Definition:*** The Hidden Markov Model (HMM) is a variant of a finite state machine having a set of hidden [states](http://www.nist.gov/dads/HTML/state.html), ***Q***, an output [alphabet](http://www.nist.gov/dads/HTML/alphabet.html) (observations), ***O***, transition probabilities, ***A***, output (emission) probabilities, ***B***, and initial state probabilities, **Π**. The current state is not observable. Instead, each state produces an output with a certain probability (***B***). Usually the states, ***Q***, and outputs, ***O***, are understood, so an HMM is said to be a triple, ( ***A***, ***B***, ***Π*** ).

***Formal Definition:***

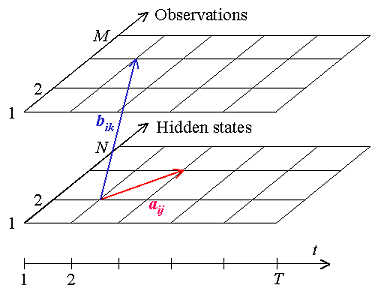
Hidden states ***Q*** = { *qi* }, *i* = 1, . . . , *N*.

Transition probabilities ***A*** = {*aij* = *P*(*qj* at *t* +1 | *qi* at *t*)}, where *P*(*a* | *b*) is the conditional probability of *a* given *b*, *t* = 1, . . . , *T* is time, and *qi* in ***Q***. Informally, ***A*** is the probability that the next state is *qj* given that the current state is *qi*.

Observations (symbols) ***O*** = { *ok* }, *k* = 1, . . . , *M* .

Emission probabilities ***B*** = { *bik* = *bi*(*ok*) = *P*(*ok* | *qi*) }, where *ok* in ***O***. Informally, ***B*** is the probability that the output is *ok* given that the current state is *qi*.

Initial state probabilities ***Π*** = {*pi* = *P*(*qi* at *t* = 1)}.



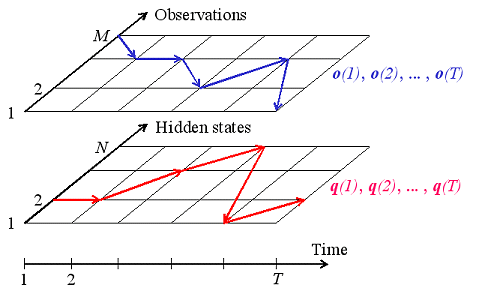
The model is characterized by the complete set of parameters: ***Λ*** = **{*A***, ***B***, ***Π* }**.

## Forward Algorithm

Let *αt*(*i*) be the probability of the partial observation sequence *Ot* = {*o*(1), *o*(2), ... , *o*(*t*)} to be produced by all possible state sequences that end at the *i*-th state.

*αt*(*i*) = *P*(*o*(1), *o*(2), ... , *o*(*t*) | *q*(*t*) = *qi* ).

Then the unconditional probability of the partial observation sequence is the sum of *αt*(*i*) over all *N* states.



Observed and hidden sequences

The Forward Algorithm is a recursive algorithm for calculating *αt*(*i*) for the observation sequence of increasing length *t* . First, the probabilities for the single-symbol sequence are calculated as a product of initial *i*-th state probability and emission probability of the given symbol *o*(1) in the *i*-th state. Then the recursive formula is applied. Assume we have calculated *αt*(*i*) for some *t*. To calculate *αt*+1(*j*), we multiply every *αt*(*i*) by the corresponding transition probability from the *i*-th state to the *j*-th state, sum the products over all states, and then multiply the result by the emission probability of the symbol *o*(*t*+1). Iterating the process, we can eventually calculate *αT*(*i*), and then summing them over all states, we can obtain the required probability.

***Formal Definition***

***Initialization:***

*α*1(*i*) = *pi bi*(*o*(1)) , *i* =1, ... , *N*

***Recursion:***

http://www.shokhirev.com/nikolai/abc/alg/hmm/images/frecur1.gif

here  *i* =1, ... , *N* , *t* =1, ... , *T* – 1

***Termination:***

http://www.shokhirev.com/nikolai/abc/alg/hmm/images/fterm.gif

## Viterbi Algorithm

The Viterbi algorithm chooses the best state sequence that maximizes the likelihood of the state sequence for the given observation sequence.

Let *δ t*(*i*) be the maximal probability of state sequences of the length *t* that end in state *i* and produce the *t* first observations for the given model.

*δ t*(*i*) = max{*P*(*q*(1), *q*(2), ..., *q*(*t*-1) ; *o*(1), *o*(2), ... , *o*(*t*) | *q*(*t*) = *qi* ).}

The Viterbi algorithm is a dynamic programming algorithm that uses the same schema as the Forward algorithm except for two differences:

1. It uses maximization in place of summation at the recursion and termination steps.
2. It keeps track of the arguments that maximize *δ t*(*i*) for each *t* and *i*, storing them in the *N* by *T* matrix ***ψ***. This matrix is used to retrieve the optimal state sequence at the backtracking step.

***Initialization:***

|  |
| --- |
| *δ*1(*i*) = *pi bi*(*o*(1)) |
| *ψ*1(*i*) = 0 , *i* =1, ... , *N* |

***Recursion:***

|  |
| --- |
| *δt* ( *j*) = max *i* [*δt* - 1(*i*) *aij*] *b j* (*o*(*t*)) |
| *ψt*( *j*) = arg max *i* [*δt* - 1(*i*) *aij*] |

***Termination:***

|  |
| --- |
| *p\** = max *i* [*δT*( *i* )] |
| *q*\**T* = arg max *i* [*δT*( *i* )] |

***Path (state sequence) backtracking:***

*q*\**t* = *ψt*+1(q\**t*+1), *t* = *T* - 1, T - 2, . . . , 1

## Kth Nearest Neighbor

* ***If it walks like a duck, quacks like a duck, and looks like a duck, then it is probably a duck.***



**Training Records**

**Test Record**

**Compute Distance**

**Choose k of the “nearest” records**

**Requires three things**

* + **The set of stored records**
  + **Distance Metric to compute distance between records**
  + **The value of *k*, the number of nearest neighbors to retrieve**

**To classify an unknown record:**

* + **Compute distance to other training records**
  + **Identify *k* nearest neighbors**
  + **Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)**

Euclidean distance between amino acids is calculated using this formulae



# Proposed Methodology:

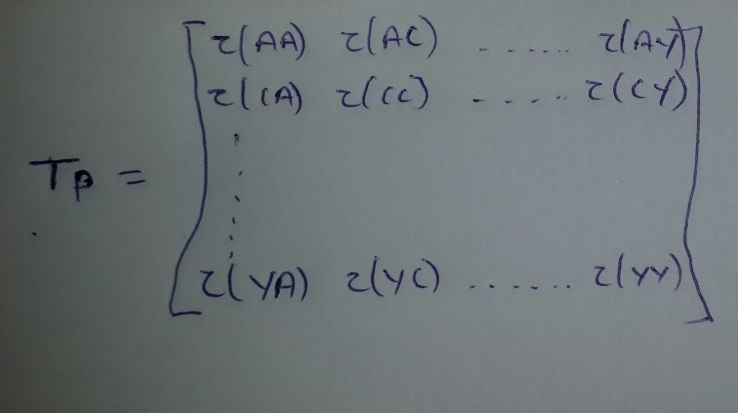
## Hidden Markov Model

### Transition Matrix

If {xk} is a stochastic process with state space S = {1,2,3. . .}, k = {1,2, . . .,}. If we assume that the current state is only related to the state before it, the biological sequence can be regarded as a stochastic process.

### Construction of transition matrices.

If a = a1a2. . .an is a given protein sequence, we can regard it as a discrete-time Markov chain. Its state space is S ({A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}), and the movements among the state space S are AA, AR, AN, AD, and so on. If r and t denote a pair of random neighboring states of the given sequence, the transition matrix (TP) can be defined as follows:

****

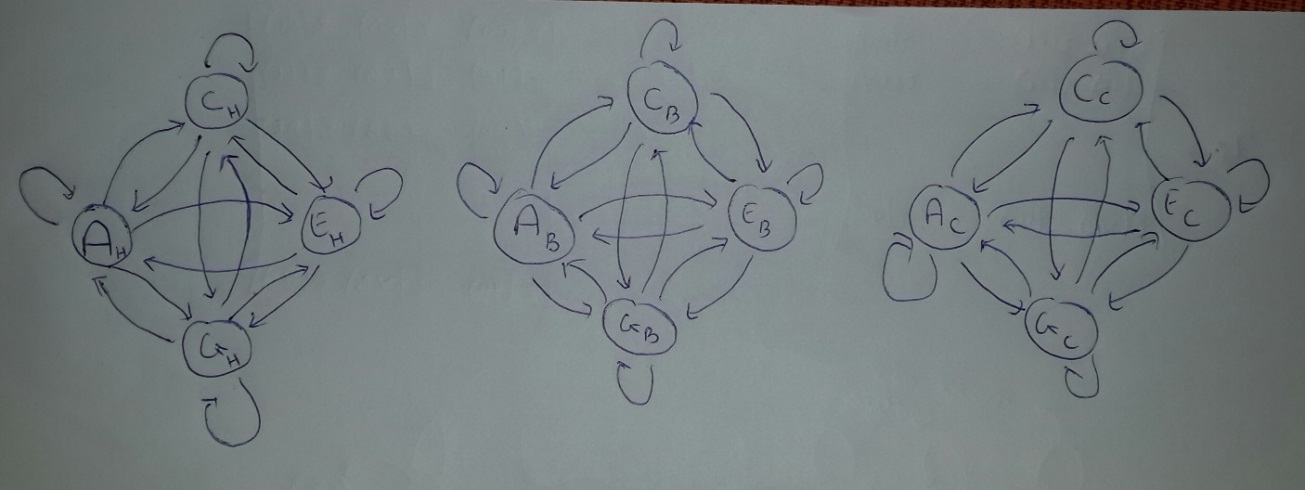
This transition matrix is calculated by finding such pairs in the database. The final condition is that the summation of the whole row should be one and in order to do that we have calculated the total number of pairs for each and then we have divided the each column with the summation of that particular row. The following algorithm is used to find the transition matrix.

### Construction of emissive matrices

If a = a1a2. . .an is a given protein sequence, we can regard it as a discrete-time Markov chain. Its state space is S ({A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}), and the corresponding protein secondary structure P (H,B,C). We are calculating the tendency for an amino acid to be in helix, beta and coil. Comparing the amino acid and its corresponding structure, we will get the number of times helix, beta, coil was present when that particular amino acid was presented. The emissive matrix (EP) can be defined as follows:

In our proposed algorithm instead of constructing One Hidden Markov model for three states of secondary structural elements. We propose three separate Hidden Markov Models i.e. for every secondary structural element we construct One Hidden Markov Model. Every Hidden Markov Model will give a probability. Out of these three probabilities which Hidden Markov Model is giving maximum probability that becomes the secondary structure of that particular amino acid.

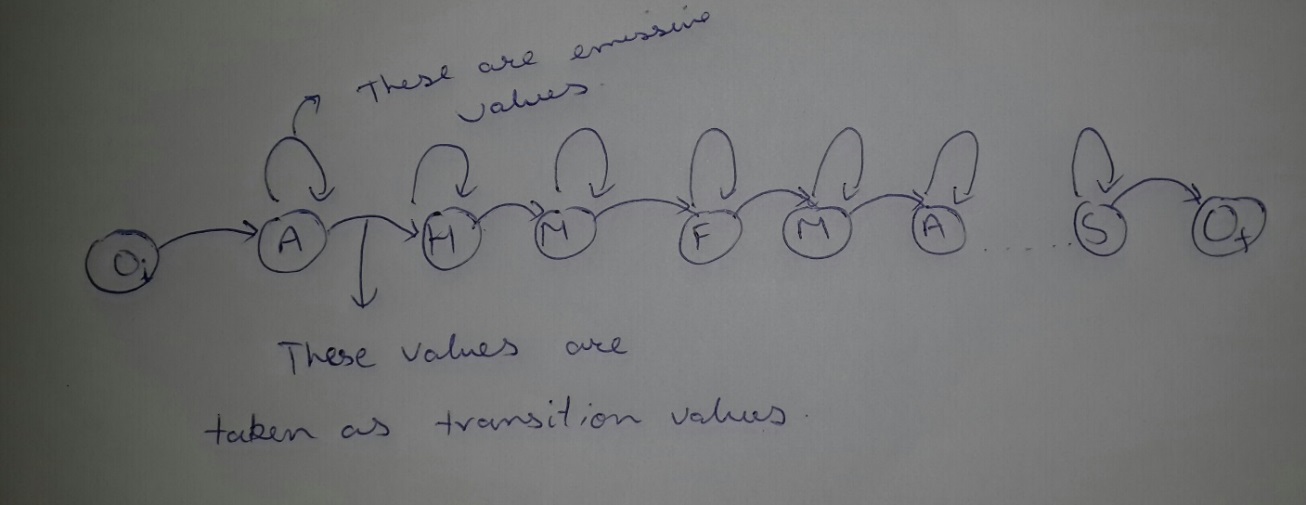
This is how the transition and the emissive values will be assigned for the amino acid ACEG separately for alpha, beta and coil respectively for this amino acid. This is the only for 4 amino, but in the project this will be for all the 20 amino acid transition for one to another.



The number of states that we have assumed in the hidden state is 20 whereas when we are calculating the probabilities, for a given primary structure we have used N+2 states where N is the length of the sequence. N states represents each amino acid in the protein sequence and the first and the last state are assumed as ‘0’ state.

So the procedure is as follows:

If we have the sequence AHMFMAENRLQLQKGS…….then the state diagram for this sequence will be



Where Oi and Of are the initial and final states.

## Forward Algorithm

The goal of the forward algorithm is to compute the joint probabilityp(x_t,y_{1:t}), where for notational convenience we have abbreviated x(t)as x_tand (y(1), y(2), ..., y(t))as y_{1:t}. Computing p(x_t,y_{1:t})directly would require marginalizing over all possible state sequences\{x_{1:t-1}\}, the number of which grows exponentially with t. Instead, the forward algorithm takes advantage of the conditional independence rules of the hidden Markov model (HMM) to perform the calculation recursively.

We calculate partial probabilities as:

at ( j )= Pr( observation | hidden state is j ) x Pr(all paths to state j at time t)

In the special case where t = 1, there are no paths to the state. The probability of being in a state at t = 1 is therefore the initial probability, i.e. Pr( state | t = 1) = P(state), and we therefore calculate partial probabilities at t = 1 as this probability multiplied by the associated observation probability;

formula

Thus the probability of being in state j at initialization is dependent on that state's probability together with the probability of observing what we see at that time.

Now this at ( j ) is used to calculate the at ( j +1) along with the transition value from j to j+1.

***Initialization:***

**i=0; f0(0)=1, fk(0)=0 for K>0**

***Recursion***

**fl(i)=el(xi)(∑ fk(i-1) \* akl )**

***Termination***

**P(x)= ∑**fk(L)ak0

Where

Transition probability akl = P(πi=l | πi-1 = k) is the probability ability of transitioning from state K to state L for K, l ≤≥ Q.

Emission probability, Ek(b)= P(xi=b| πi = k), for each state , K, and each symbol b, where Ek(b) is the probability of seeing symbol b in state k.

**Flowchart of the Algorithm**

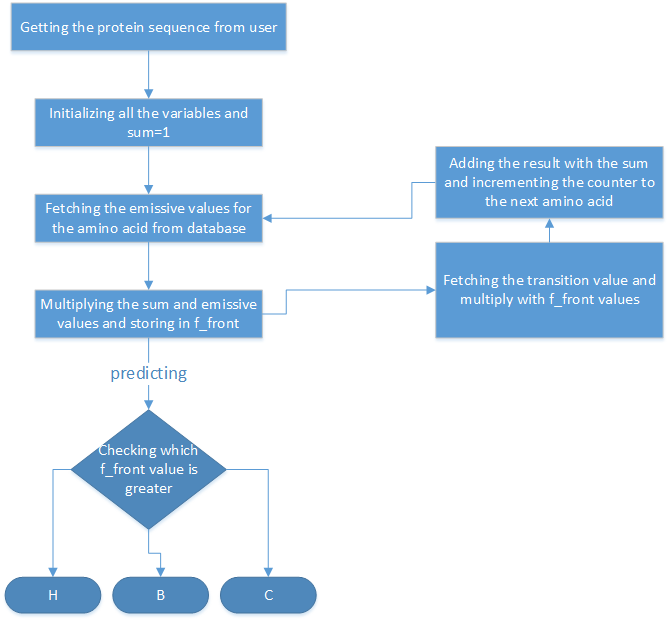


Figure 2: Flowchart for Forward Algorithm

## Viterbi Algorithm

***Input:*** x = x1……xN

***Initialization:***

V0 (0) = 1 (0 is the imaginary first position)

Vk (0) = 0, for all k > 0

***Iteration:***

Vj (i)= ej (xi) × maxk akj Vk(i – 1)

Ptrj(i) = argmaxk akj Vk(i – 1)

***Termination:***

P(x, π\*) = maxk Vk(N)

***Trackback:***

πN\* = argmaxk Vk(N)

πi-1\* = Ptrπi (i)

***Time:***

O(K2N)

***Space:***

O(KN)

Underflows are a significant problem

**P[ x1,…., xi, π1, …, πi ] = a0π1 aπ1π2……aπi eπ1(x1)……eπi(xi)**

These numbers become extremely small – underflow

**Solution:** Take the logs of all values

**Vl(i) = logek(xi) + maxk [ Vk(i-1) + log akl ]**

***Flowchart of the algorithm:***

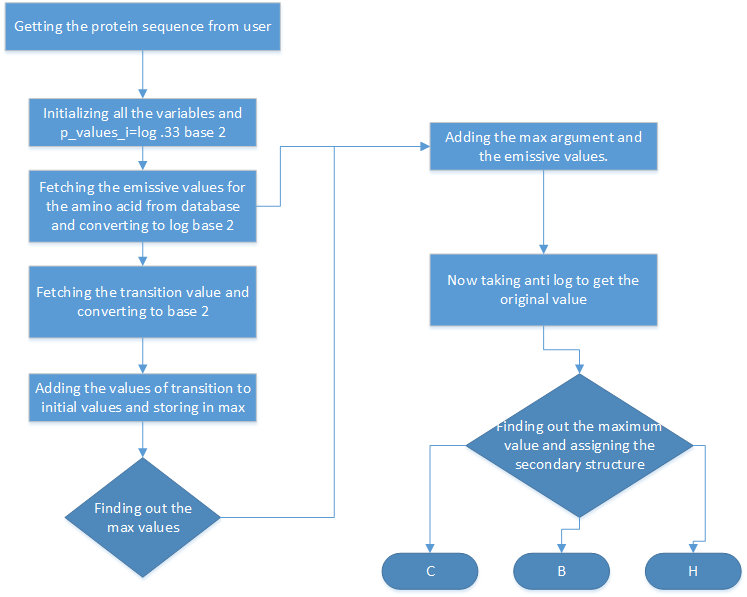
****

Figure 3: Flowchart for Viterbi algorithm

## Kth Nearest Neighbor

We implemented Kth Nearest Neighbor to classify the proteins and to find the secondary structure and this we have done in three different levels as Level 2, Level 3 and Level 4. In Level 2 we have formed pairs of 2, two adjacent amino acids without repetition, in this level similarly in level 3 formed pairs of 3 and in level 4 pairs of 4 and then these structures obtained by the three different levels are found and the accuracy is found separately for these three different levels.

# Context Model

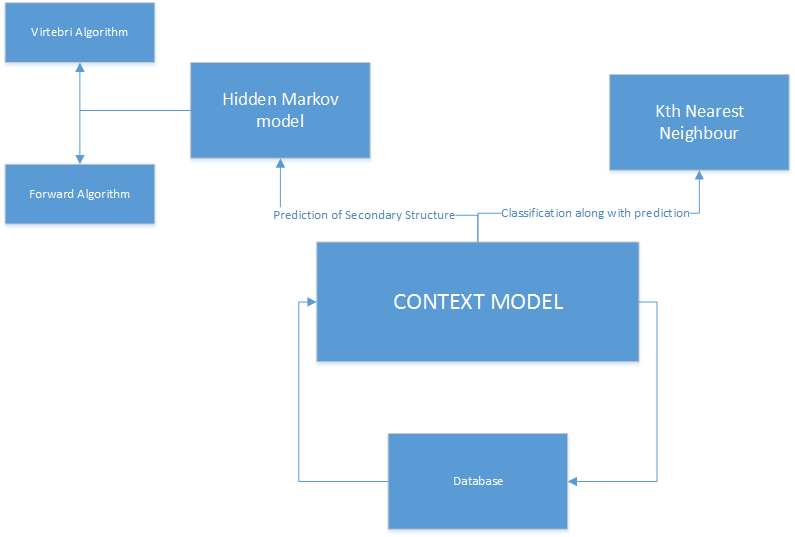
****

Figure 4: Context Model

# Project Model

The process model which will be used is Incremental model as the system needs to be updated very often and features are getting added day by day and as it is a research project it will need a lot to add and a lot of changes may be done Hence we are using incremental model and though, protein database is very vast we cannot use Waterfall Model as we are not yet clear of the complete protein behaviour and research is still going on and when compared to spiral and revolutionary as features are getting added Incremental suits the best.

# Project Planning

## Work Breakdown Structure

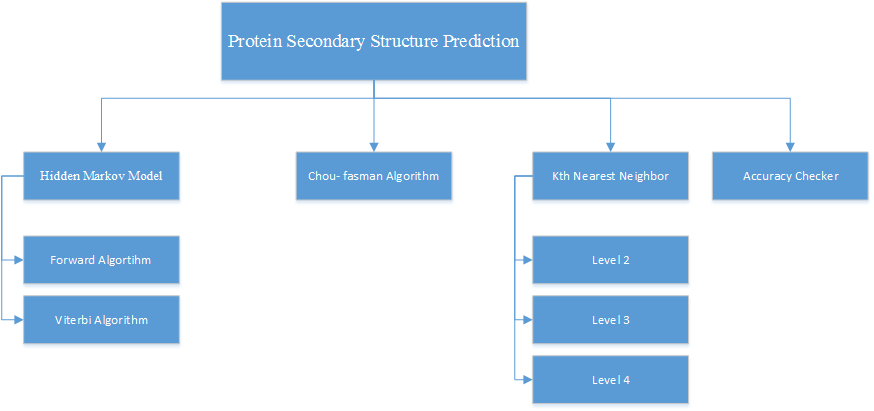


Figure 5: Work Breakdown Structure

This section attempts to describe each module of the project in brief and the detailed description of each of these modules. The Protein Secondary Structure in this project we have done using three different algorithms.

* Hidden Markov Model

1. Forward Algorithm
2. Viterbi Algorithm

* Chou- Fasman Algorithm
* Kth Nearest Neighbor

1. Level 2
2. Level 3
3. Level 4

* Accuracy checker

### Hidden Markov Model

This module is divided into two sub modules. They are:

* Forward Algorithm
* Viterbi Algorithm

#### Forward Algorithm

The forward algorithm sums probability values, so it is not a viable solution to log the values in order to avoid underﬂow. The most common solution to this problem is to use scaling coeﬃcients that keep the probability values in the dynamic range of the machine, and that are dependent only on t.

#### Viterbi Algorithm

As the Viterbi algorithm only multiplies probabilities, a simple solution to underﬂow is to log all the probability values and then add values instead of multiply. In fact if all the values in the model matrices (A,B) are stored logged, then at runtime only addition operations are needed.

### Chou- Fasman Algorithm

The Chou-Fasman method predicts helices and strands in a similar fashion, first searching linearly through the sequence for a “nucleation” region of high helix or strand probability and then extending the region until a subsequent four-residue window carries a probability of less than 1.

1. Assign all of the residues in the peptide the appropriate set of parameters.
2. Scan through the peptide and identify regions where 4 out of 6 contiguous residues have P(a-helix) > 100. That region is declared an alpha-helix. Extend the helix in both directions until a set of four contiguous residues that have an average P(a-helix) < 100 is reached. That is declared the end of the helix. If the segment defined by this procedure is longer than 5 residues and the average P(a- helix) > P(b-sheet) for that segment, the segment can be assigned as a helix.
3. Repeat this procedure to locate all of the helical regions in the sequence.
4. Scan through the peptide and identify a region where 3 out of 5 of the residues have a value of P(b- sheet) > 100. That region is declared as a beta-sheet. Extend the sheet in both directions until a set of four contiguous residues that have an average P(b-sheet) < 100 is reached. That is declared the end of the beta-sheet. Any segment of the region located by this procedure is assigned as a beta- sheet if the average P(b-sheet) > 105 and the average P(b-sheet) > P(a-helix) for that region.

### Kth Nearest Neighbor

This is implemented in three different levels as Level 2, 3, 4.And accuracy is found separately at that particular levels.

## Pert Chart

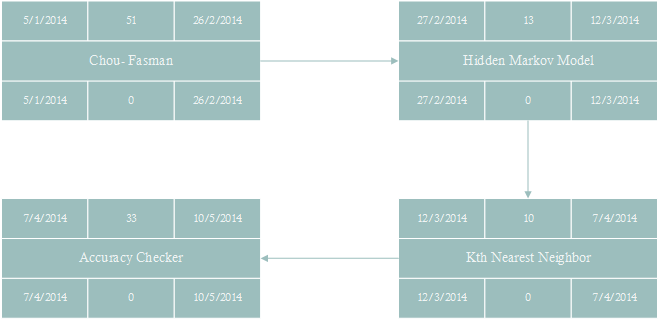


Figure 6: Pert Chart

# Risk Management

* **Software Requirement Risks**
* Lack of analysis for change of requirements.
* Change extension of requirements.
* Lack of report for requirements.
* **Software Quality Risks**
* Lack of enough skill.
* Lack of testing and good estimation in projects.
* Inadequate knowledge about techniques, programming language, tools, and so on.
* **Software Cost Risks**
* Complexity of architecture.
* Large size of architecture.
* Lack of monitoring.

# Stake Holders

* Admin/user

# Requirements Stake Holder

* The system shall ask the user/Admin to enter a valid protein sequence.
* The system shall ask the fasta format for knn.
* The system shall require User/Admin to enter equal length of secondary structure for accuracy prediction.
* The system shall accept N length protein sequence for prediction.

# Functionality

## Functional Requirements

* Enter the protein sequence and predict the secondary structure.
* Check for accuracy of secondary structure of proteins with respect to the standard protein secondary structure obtained from standard PDB database.
* Calculating the emission and transition probabilities.
* Classification of proteins into different levels using Kth  Nearest Neighbor.

**Main Functionalities:**

***Functionality***: Manage Database

***Input****:* We saved the protein sequences in the database and calculated emission and transition probabilities and saved them.

***Output:*** Protein secondary structure is obtained based on the algorithm which we use.

***Functional Description:*** Protein sequence is entered and the secondary structureis predicted and then using the secondary structure which we obtain by using any of the algorithms.

***Action:***

User: Enters the protein sequence

System: Calculate the Emission and transition probabilities and then predict the secondary structure using any of the algorithms.

***Precondition*: No conditions**

***Post condition*: No conditions**

## Non Functional Requirements:

This section describes, in details, the functional requirements of the Disease Diagnostic System.

### ****Stability:****

The system shall provide the user with database and all the stored data to predict the protein secondary structure.

### Reliability:

The system has to be very reliable due to the importance of the data. The system shall provide 100% access reliability.

### ****Availability:****

**The system is available 100% for the user and is used 24hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.**

### ****Performance:****

* **Throughput: The no of transactions is directly dependent on the no of users using the system; the users may be the doctors or the administrator.**
* **Response: The system shall take as less time as possible to provide the desired output to the user or the admin.**

# ****User Characteristics****

The product to be developed interacts only with a set of authorized users

**Users/ Administrator** - The User/Administrator user will be computer literate and technically competent in performing administration on computer systems. They will have the access to the database and will be able to find the secondary structure and will be able to determine the accuracy.

# ****Constraints:****

* The users must have the basic knowledge about working with the software.
* GUI is only in English
* **The users/Admin must enter valid input to the system.**

# Assumptions and Dependencies:

* **The users have sufficient knowledge of computers.**
* **The users know English language as the interfacing language would be English.**
* **The inputs entered into the system are valid.**

# ****System Requirements:****

## User Interface:

Keyboard, mouse and touch screen.

## Hardware Requirements:

* PC or laptop.
* Processor Pentium or AMD.
* Hard disk (min 40 GB).
* RAM 1GB.
* Printer.
* Modem.

## Software Requirements:

* Microsoft Visio 2010
* Rational rose enterprise edition
* Net Beans
* XAMPP-Web server
* MySQL-Database
* Web browser (Supporting JavaScript)
* Supporting operating system
* Database Protein sequences from PDB database.

# Architecture Diagram

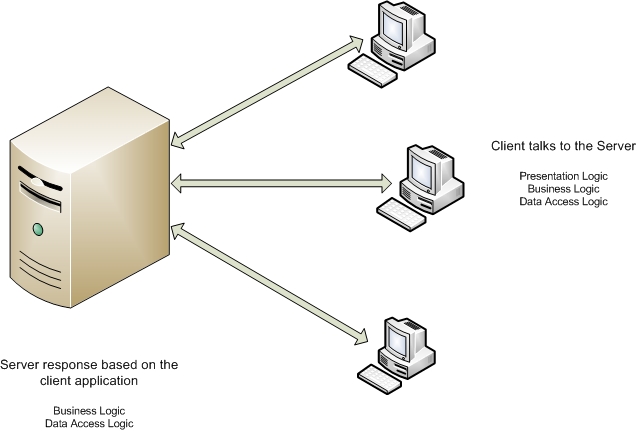
****

Figure 7: Architecture Diagram

The Two -tier application programming model was developed to enhance the file server application programming model. As compared with the file server application programming model, the two-tier application programming model provides you with improved usability, scalability, and flexibility of applications. In Two-Tier model, you will have two separate layers namely Client & Server.

The applications developed using the two-tier application programming model have a **user-friendly interface**. These applications **can support only a few users** and allow data to be shared within a homogeneous environment. The various tiers in a two-tier architecture are separated from each other by physical boundaries. These physical boundaries can be machine boundaries, process boundaries, or corporate boundaries. Two-tier application programming model is a combination of a client application and a server application.

In this application programming model, the client application directly interacts with the server application **without the presence of any intermediate application**. The Client application communicates with the data layer through a database bridge Application Programming Interface (API).The Database driver is installed in each computer that runs the client application.

Disadvantage:

If the database changes, we need to reinstall Database driver in all the computer. Database Connection is retained even when the client is not accessing the database. This makes the database connection unavailable to other users. Therefore only a limited number of clients can access the database at a time. High network traffic because of an increase in the number of trips of data transfer across the physical boundaries of the network. Each time a database operation is performed, the data is transmitted across the physical boundary that separates the business logic and data layers.

# Data Flow Diagram:

## Level 0:

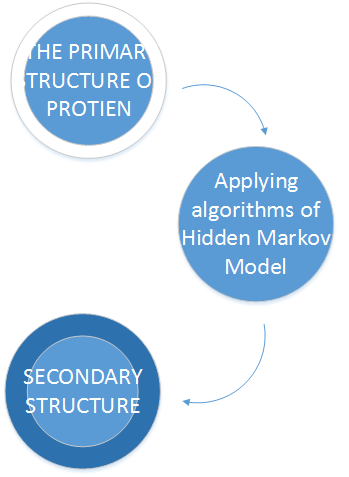


Figure 8: Data Flow Diagram: Level 0

## Level 1:

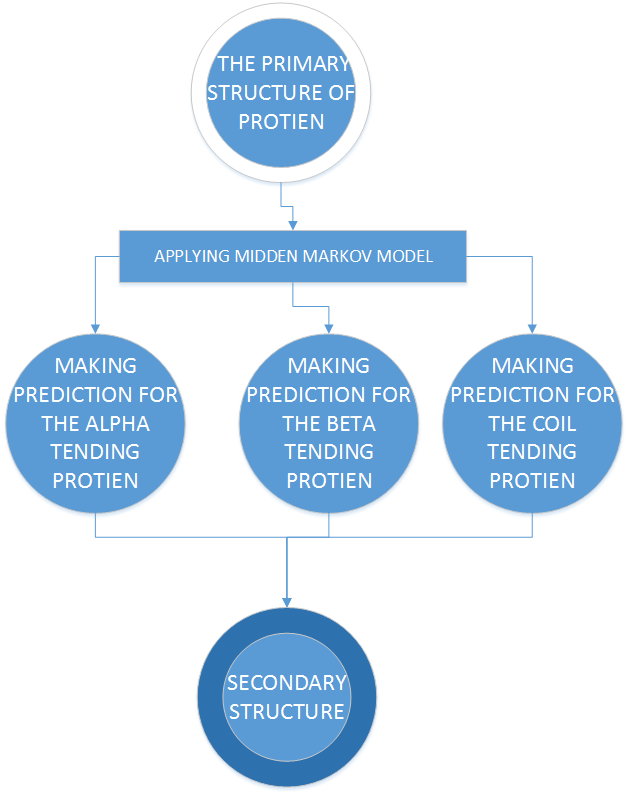
****

Figure 9: Data Flow Diagram: Level 1

# SEQUENCE DIAGRAM:

## Accuracy Sequence:

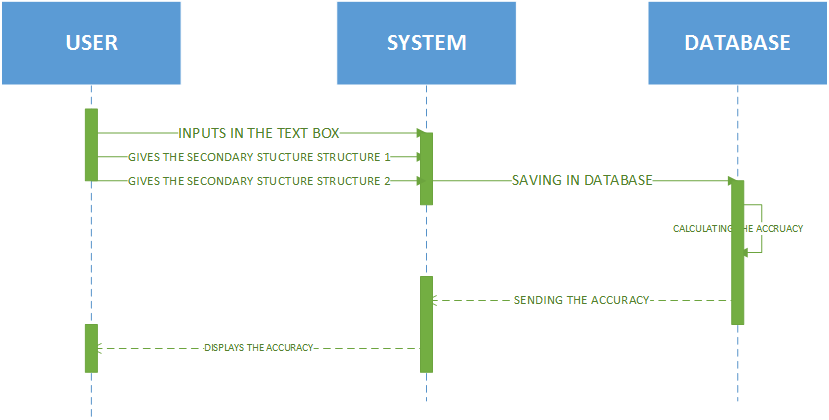
****

Figure 10: Sequence Diagram: Accuracy Sequence

## Prediction Sequence

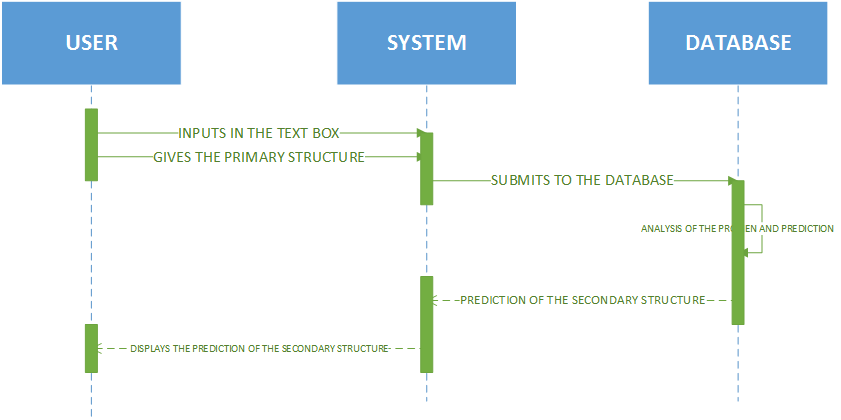
****

Figure 11: Sequence Diagram: Prediction Sequence

# ER DIAGRAM:

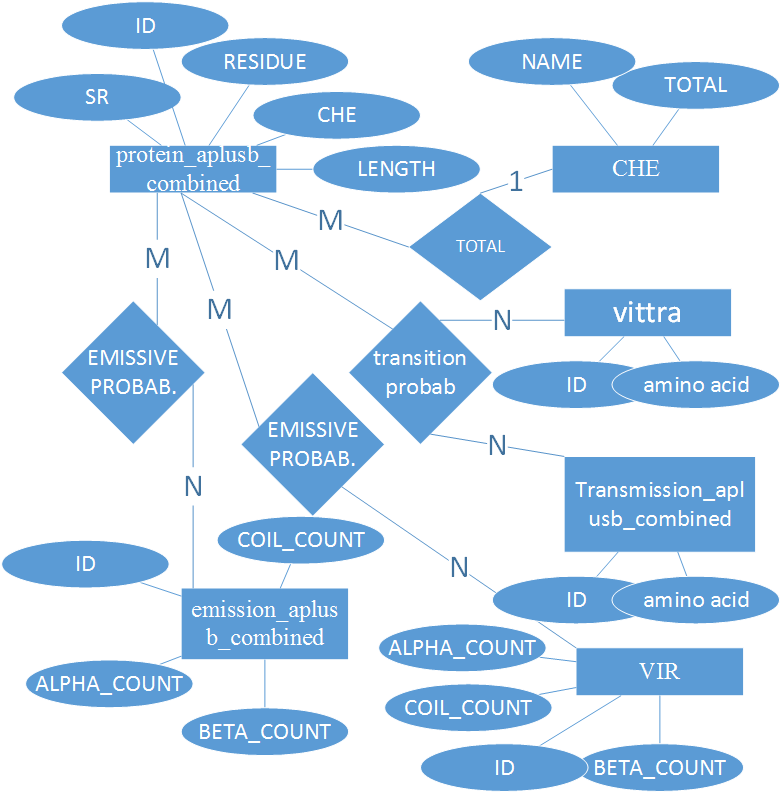
****

Figure 12: ER Diagram

# Result analysis:

In this project we found the secondary structure of the protein by using Forward Algorithm, Viterbi Algorithm and classified the proteins using Kth Nearest Neighbor. Now we are trying to analyze the results of the existing algorithms with that of the algorithms which we implemented by slight modification. . The accuracy of the predictions made by Qian Sjnowski seemed is better than those obtained by previous methods and was reported to be in the range of 62.7-64.4%. Rost and Sander have developed the prediction mail server called PHD with a prediction accuracy of 71.6% was reported. The simplest model for three-class prediction is a HMM with three hidden states, each state accounting for a secondary structure class. Parameter estimation of such a model is straight forward because the segmentation is fully determined. But the performance of this model is limited: the Q3 score (proportion of residues with correct prediction) is 58.3%. A random prediction gives a Q3 score equals to 34.5% [2] .

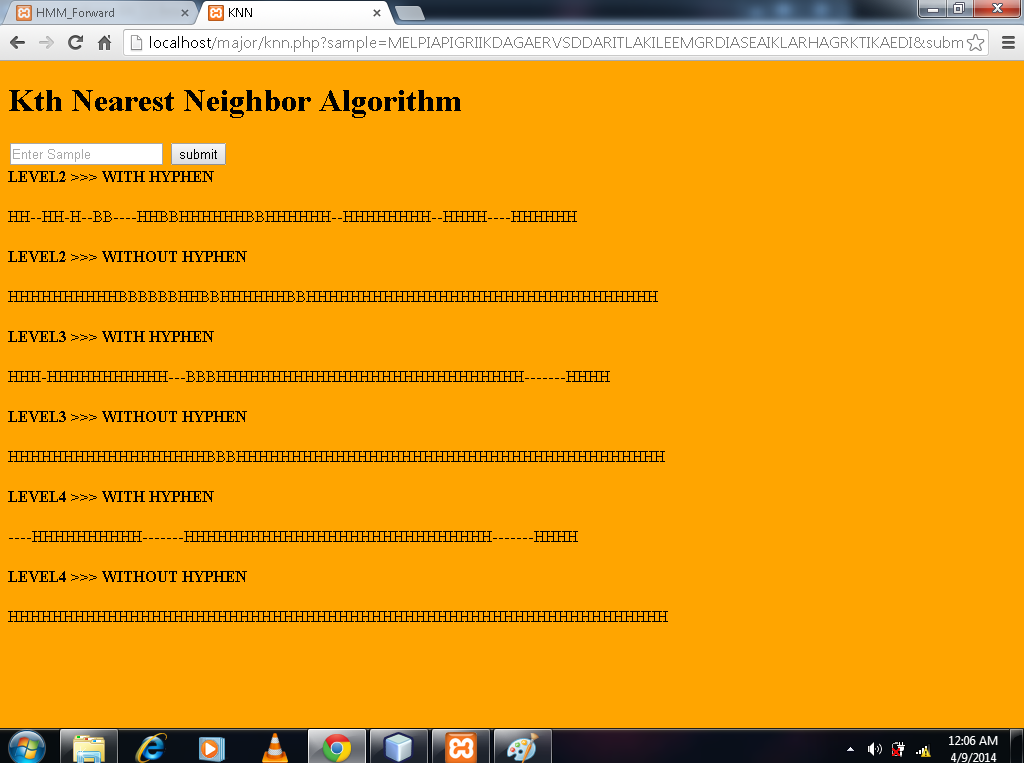
In terms of prediction accuracy, neural networks are among the most popular methods in use today, delivering a point wise prediction accuracy (Q3) of about 77% and a segment overlap measure (SOV) of about 74%. However, to improve the long-term performance of secondary structure pre- diction, it likely will be necessary to develop a cost model that mirrors the underlying biological constraints. While neural networks oﬀer good performance today, their operation is largely opaque. Often containing up to 10,000 parameters and relying on complex layers of non-linear perceptron, neural networks oﬀer little insight into the patterns learned. Moreover, they mask the short- comings of the underlying models, rendering it a tedious and ad-hoc process to improve them. In fact, over the past 15 years, the largest improvements in neural network prediction accuracy have been due to the integration of homologous sequence alignments rather than speciﬁc changes to the underlying cost model. Of the approaches developed to date, Hidden Markov Models (HMMs) offer perhaps the most natural representation of protein secondary structure. An HMM consists of a ﬁnite set of states with learned transition probabilities be- tween states. In biological terms, each transition corresponds to a local folding event, with the most likely sequence of states corresponding to the lowest-energy protein structure. HMMs generally contain hundreds of parameters, 1-2 orders of magnitude less than that of neural networks. In addition to providing a tractable model that can be reasoned about, the reduction in parameters lessens the risk of overlearning. However, the leading HMM methods to date have not exceeded a Q3 value of 75%, and SOV scores are often unreported [4].

Bystroﬀ, Thorsson, and Baker design an HMM to recognize speciﬁc structural motifs and assemble them into protein secondary structure prediction [3]. Using alignment proﬁles, they report an overall Q3 value of 74.3%. Our approach may use fewer parameters, as they manually encode each target motif into a separate set of states. Martin, Gibrat, and Rodolphe develop a 21-state HMM model with 471 parameters that achieves an overall Q3 value of 65.3% (without alignment proﬁles) and 72% (with alignment proﬁles) [21]. Alpha helices are identiﬁed based on an amphiphilic: a succession of two polar residues and two non-polar residues. Won, Hamelryck, Pru¨gel-Bennet and Krogh give a genetic algorithm that automatically evolves an HMM for secondary structure prediction [40, 41]. Using alignment proﬁles, they report an overall Q3 value of 75% (only 69.4% for helices). They claim that the resulting 41-state HMM is better than any previous hand-designed HMM. While they restrict their HMM building blocks to “biologically meaningful primitives”, it is unclear if there is a natural energetic interpretation of the ﬁnal HMM. Schmidler, Liu, and Brutlag develop a segmental semi-Markov Model (a generalization of the HMM), allowing each hidden state to produce a variable-length sequence of the observations [35, 36]. They report a Q3 value of 68.8% without using alignment proﬁles [5].

We used Kth Nearest Neighbor to classify proteins and to find the protein secondary structure.the accuracy of this Kth nearest neighbor achieves a of now is 89%.

# Screen Shoots:





# VERIFICATION AND VALIDATION:

## Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Test Case | Expected Result | Actual Result | Test Outcome |
| 1 | Inputs must be entered | It should show an alert box stating that inputs not entered correctly | Yes, it’s showing | Passes |
| 2 | Both the proteins entered for accuracy check must be of same length | A prompt message to show length must be same | It is displaying | Passed |
| 3 | If only one input is given it should show need t input both the sequences to check | A prompt message to enter both the sequences | A prompt message to enter both sequences | Passed |
| 6 | Login facility | There should be a username and password for anyone to enter the system | No login available. | Failed |

# Conclusion:

Protein secondary structure plays an important role in the rapidly developing branches of engineering specializing in the study of objects in 3D structure prediction. But many of the researches they have applied probabilistic approach. Some companies have already used various architectures but three layer architectures gives good performance with minimum number of nodes. Association classification breaks the limitations of algorithms built on homological analysis and classic artificial intelligence, such as decision tree and the SVM and by pursuing a different way; it develops, out of rules of association classification. Up to now, for this problem maximum 90% achieved with less similarity. In our opinion, it is time for extensive search for the ways of practical use PSS accuracy and speed consideration are likely to remain important as genomic, proteomic and protein engineering projects continue to generate great challenges and opportunities in this area. The protein 2D structure prediction is one of the most hopeful works in the future.

Now below is the table with the accuracy achieved by us

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fasta Format | ForwardAlgorithm | Viterbi Algorithm | Level 2 KNN | Level 3 KNN | Level 4 KNN |
| 3GRS | 54 | 55 | 66 | 95 | 53.2 |
| 1ZTE | 62.2 | 64.8 | 70 | 79.7 | 56.08 |
| 1A7W | 72.3 | 72.3 | 70.2 | 85.16 | 100 |
| 1ABS | 68.75 | 71 | 73 | 91.4 | 91.4 |
| 2FP1 | 81.74 | 84.1 | 80.9 | 90.47 | 98.4 |

# Reference:

1. Jieyue He, Hae-Jin Hu, Robert Harrison, Phang C. Tai, Yi Pan (2006) ‘Rule Generation for Protein Secondary Structure Prediction With Support Vector Machines and Decision Tree’, IEEE Transactions on Nanobioscience, Vol. 5, No. 1, pp. 46-53.
2. K. Karplus, C. Barrett, M. Cline, M. Diekhans, L. Grate, R. Hughey (1999) ‘Predicting Protein Structure Using Only Sequence Information’, Proteins, Vol. 37, pp. 121-125.
3. Lin, K. Simossis, V.A. Taylor, W.R. Heringa J. (2005) ‘A Simple and Fast Secondary Structure Prediction Method using Hidden Neural Networks’, Bioinformatics, Vol. 21 No. 2, pp. 152-159.
4. Reet Kamal Kaur, Manjot Kaur, Amanjot Kaur. (2010) ‘Using Cluster Analysis for Protein Secondary Structure Prediction’, International Journal of Computer Applications, Vol.4 No.12, pp.20-22.
5. Pan X. M. (2001) ‘Multiple Linear Regression for Protein Secondary Structure Prediction’, Proteins, Vol. 43 No. 3, pp. 256-259.
6. Qin, S. He, Y. Pan X.M. (2005) ‘Predicting Protein Secondary Structure and Solvent Accessibility with an Improved Multiple Linear Regression Method’, Proteins, Vol. 61 No. 3, pp. 473-480.
7. Salamov A. A. and Solovyev V.V. (1995) ‘Prediction of Protein Secondary Structure by Combining Nearest-Neighbor Algorithms and Multiple Sequence Alignments’, Journal of Molecular Biology, Vol. 247 No. 1, pp. 11-15.
8. Zhun Zhou a,Bingru Yang b and Wei Hou b (2010) ‘Association classification algorithm based on structure sequence in protein secondary structure prediction’, Expert Systems with Applications, Vol.37 No.9, pp. 6381–6389.
9. Chou PY, Fasman GD (1974) ‘Prediction of protein conformation’, Biochemistry, Vol.13 No.2, pp. 211–215.
10. Garnier J, Osguthorpe DJ, Robson B (1978) ‘Analysis and implications of simple methods for predicting the secondary structure of globular proteins’, Journal of Molecular Biology, Vol.120 No.1, pp. 97–120.
11. Jones, D.T. (1999) ‘Protein Secondary Structure Prediction Based on Position-Specific Scoring Matrices’, Journal of Molecular Biology, Vol. 292 No. 2, pp. 195-202.
12. Holley HL, Karplus M. (1989) ‘Protein secondary structure prediction with a neural network’ Proceeding of the National Academy of Science of the United states of America,Vol.86, pp. 152-156.
13. Baldi, P. Brunak, S. Frasconi, P. Soda, G. Pollastri. G. (1999) ‘Exploiting the Past and the Future in Protein Secondary Structure Prediction’, Bioinformatics, Vol. 15 No. 11, pp. 937-946.
14. Pollastri, G. and McLysaght, A. (2005) ‘Porter: A New, Accurate Server for Protein Secondary Structure Prediction’, Bioinformatics, Vol. 21 No. 8, pp. 1719-1720.