**ABSTRACT**

Protein Database is growing rapidly but it is difficult to get all the information from the sequences directly. Therefore many methods have come into existence to analyse the protein sequences, the existing methods have their limitations to numerically characterize the protein sequences exactly. Hence here we are using mechanisms with the main aim of reducing the risk of inherited diseases in the future generations by accurately predicting the secondary structure from the primary protein sequence. Hence we are predicting the secondary structure using Chou-fasman method and Hidden Markov Model there by increasing the accuracy of the protein secondary structure. Predicting the secondary structure of proteins is one of the best studied problems in bioinformatics and consequently presents a rigorous test of our techniques. The problem tackled is to provide a label for each residue in a protein sequence depending on its secondary structure. That is, whether the protein residue is part of an alpha-helix, a beta-sheet or some other structure. This is a first step towards predicting the structure and function of a protein from its sequence. Our approach is to this problem is to evolve an HMM using a Genetic Algorithm. A Hidden Markov Model (HMM) is a probabilistic finite state machine used to model stochastic sequences. An HMM is defined by the set of states, emission probabilities associated with each state, and transitions that connect states. Hidden Markov models are widely used in science, engineering and many other areas (speech recognition, optical character recognition, machine translation, bioinformatics, computer vision, finance and economics, and in social science).  We here using the algorithms under Hidden Markov they are Forward algorithm and Viterbi algorithm. To achieve higher accuracy in predicting the protein secondary structure classification plays a major role and KNN here plays major role in classification of proteins. The algorithm we shall investigate is the k-nearest neighbour algorithm, which is most often used for classiﬁcation, although it can also be used for estimation and prediction. K-Nearest neighbour 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 Hence We used Kth nearest neighbour in order to classify the protein sequences until the level 4 there by increasing the accuracy of the protein secondary structure.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  |  |  |
|  | **ABSTRACT** | **i** |
|  | **LIST OF FIGURES** | **iv** |
|  |  |  |
| **1.** | **PROJECT DESCRIPTION** | **1** |
|  | **1.1. Aim** | **1** |
|  | **1.2. Objective** | **1** |
|  | **1.3. Applicability** | **1** |
| **2.** | **INTRODUCTION**  **2.1 Purpose**  **2.2 Scope** | **2**  **2**  **2** |
|  |  |  |
| **3.** | **PROBLEM STATEMENT** | **3** |
|  |  |  |
| **4.**  **5.**  **6.**  **7.**  **8.**  **9.** | **LITERATURE REVIEW**  **STUDY OF TECHNIQUES**  **5.1 Hidden Markov Model**  **5.2 Forward Algorithm**  **5.3 Viterbi Algorithm**  **5.4 Kth Nearest Neighbor**  **PROPOSED METHODLOGY**  **6.1 Hidden Markov Model**  **6.1.1 Transition Matrix**  **6.1.2 Construction of Transition Matrix**  **6.1.3 Construction of Emissive Matrix**  **6.2 Forward Algorithm**  **6.3 Viterbi Algorithm**  **6.4 Kth Nearest Neighbor**  **CONTEXT MODEL**  **PROCESS MODEL**  **PROJECT PLANNING**  **9.1 Work Breakdown Structure**  **9.1.1 Hidden Markov Model**  **9.1.2 Chou-fasman Algorithm**  **9.1.3 Kth Nearest Neighbor**  **9.2 Pert Chart** | **4**  **6**  **6**  **7**  **8**  **10**  **11**  **11**  **11**  **11**  **12**  **13**  **15**  **17**  **17**  **17**  **18**  **18**  **18**  **19**  **20**  **20** |
|  |  |  |
| **10.** | **RISK MANAGEMENT** | **21** |
|  | **10.1. Software Requirement Risks** | **21** |
|  | **10.2. Software Quality Risks** | **21** |
|  | **10.3. Software Cost Risk** | **21** |
|  |  |  |
| **11.** | **STAKE HOLDERS** | **21** |
|  |  |  |
| **12.**  **13.**  **14.**  **15.**  **16.**  **17.**  **18.**  **19.**  **20.**  **21.**  **22.**  **23.**  **24.**  **25.**  **26.** | **REQUIREMENTS**  **FUNCTIONALITY**  **13.1 Functional Requirement**  **13.2.NonFunctional requirements**  **13.2.1.Stability**  **13.2.2.Reliability**  **13.2.3. Availability**  **13.2.4.Performance**  **USER CHARCTERISTICS**  **CONSTRAINTS**  **ASSUMPTIONS AND DEPENDENCIES**  **SYSTEM REQUIREMENTS**  **17.1. User Interface**  **17.2. Hardware Requirements**  **17.3. Software Requirements**  **ARCHITECTURE DIAGRAM**    **DATA FLOW DIAGRAM**  **19.1.level 19.2.level**  **SEQUENCE DIAGRAM**  **20.1. Accuracy Sequence**  **20.2. Prediction Sequence**  **ER DIAGRAM**  **RESULT ANALYSIS**  **SCREEN SHOTS**  **VERIFICATIONA AND VALIDATION**  **21.1 Test Cases**  **CONCLUSION**  **REFERENCE** | **22**  **21**  **21**  **23**  **23**  **23**  **23**  **23**  **23**  **24**  **24**  **24**  **24**  **24**  **24**  **25**  **26**  **26**  **27**  **28**  **28**  **28**  **29**  **30**  **32**  **33**  **33**  **33**  **34** |

**LIST OF FIGURES**

Figure 1 : General Diagram………………………………………………3

Figure 2 : Flow Chart of forward Algorithm……………….…………….15

Figure 3 : Flow Chart of forward Algorithm……………….…………….16

Figure 4 : Context Model .…………………………………………….....17

Figure 5 : Work Breakdown Structure…… ………………………..........18

Figure 6 : Pert Chart……….…………………………………………......20

Figure 7 : Architecture Diagram.……………………………………...... 25

Figure 8 : DFD 0………….…………………………………………….. 26

Figure 9 : DFD 1………….…………………………………………….. 27

Figure 10 : Accuracy Sequence …………………………....................... 28

Figure 11 : Prediction Sequence ……………………………………….. 28

Figure 12:ER Diagram…………………………………………...………29