Method for recognizing local descriptors of protein structures using Hidden Markov Models

Patrik Björkholm

Abstract

Being able to predict the sequence-structure relationship in proteins will extend the scope of many bioinformatics tools relying on structure information. Here we use Hidden Markov models (HMM) to recognize and pinpoint the location in target sequences of local structural motifs (local descriptors of protein structure, LDPS) These substructures are composed of three or more segments of amino acid backbone structures that are in proximity with each other in space but not necessarily along the amino acid sequence. We were able to align descriptors to their proper locations in 41.1% of the cases when using models solely built from amino acid information. Using models that also incorporated secondary structure information, we were able to assign 57.8% of the local descriptors to their proper location. Further enhancements in performance was yielded when threading a profile through the Hidden Markov models together with the secondary structure, with this material we were able assign 58,5% of the descriptors to their proper locations. Hidden Markov models were shown to be able to locate LDPS in target sequences, the performance accuracy increases when secondary structure and the profile for the target sequence were used in the models.

Sammanfattning

Förmågan att kunna förutsäga sekvens-struktur-förhållandet i proteiner skulle vidga möjligheterna för många bioinformatikverktyg som är beroende av strukturell information. Har använder vi Hidden Markov models för att känna igen och lokalisera positionen av lokala strukturmotiv (local descriptors of protein structure, LDPS) i en målsekvens. Dessa substrukturer består av tre eller fler segment som av aminosyror som ligger nära i rymden men inte nödvändigtvis i aminosyrasekvensen. Vi lyckades arrangera lokala descriptorer till deras rätta position till 41.1% av fallen när vi använde modeller som enbart var byggda från aminosyra information. När modeller som även använde sig av sekundär strukturer så lyckades vi arrangera descriptorerna till sin rätta position till 57.8%. Modellernas prestationer förbättrades ytterligare då en profil for mål sekvensen trådades genom modellen istället för mål sekvensen, då arrangerades 58.5% av descriptorerna till sina rätt positioner. Hidden Markov-modeller har visat sig vara kapabla att lokalisera LDPS i mål sekvenser och dessa modeller presterar som bäst då sekundärstrukturer och profiler för målsekvenser används.

Preface

A master Thesis concludes a Master of Science Degree. It is a compulsory work which is written in the ninth term. This work concludes a Master of Science in Engineering Biology at Linköping University, Institute of Technology. This Master Thesis was performed at The Linnaeus Centre for Bioinformatics in Uppsala.

Table of contents

Chapter 1

Introduction

1.1 Background

1.1.1 Protein structure prediction

Predicting the three dimensional structure of a protein from its amino acid sequence is a very important unsolved problem in bioinformatics [4]. Enabling science to predict proteins three-dimensional structures accurately would help scientists understand a variety of different hereditary diseases and would also cheapen and shorten the time required to develop new drugs and other treatments [9]. This knowledge would also allow scientists to better understand a variety of different biological processes [8]. The main methods for obtaining the three dimensional structures of proteins experimentally are X-ray crystallography or NMR, which both are expensive methods both in terms of time and money. The consequence of this is that the gap between collected structural data compared to collected sequence data is ever increasing [10]. Currently the amino acid sequences of a lot of proteins are known, due to many different sequencing projects ongoing or finished in the world [9-10]. If a reliable computational method can be found to accurately predict the three-dimensional structure of a protein, the growing gap between the sequence data and structure data can in time lessen and perhaps be bridged. The problem with predicting the three dimensional structures from the amino acid sequence is that the number of possible three-dimensional structures of a protein is exponential. In order to be able to build a model of a protein with a computer and make it computable the number of possible structures must be limited, and several approaches has been made that constraints the number of possible structures. Three main approaches have evolved during the last thirty years to predict the three-dimensional structures of amino acid sequences. These are Homology modeling, Protein threading and Ab initio modeling.

1.1.1.1 Homology modeling

In Homology modeling, sometimes referred to as comparative modeling, a homologous protein or proteins with a known three dimensional structure are used as a mold for predicting the three dimensional structure of the target sequence [1]. One drawback of this method is that it requires a homologous protein with known three dimensional structures in order to work. A proper homologous protein should have at least 25% sequence identity [20]. Another downside with this method is that the reliability of the predicted model decreases with lower sequence identity between template and target [5].

1.1.1.2 Protein Threading

In Protein Threading, also known as fold recognition, the target proteins sequence is threaded through backbone structures of collections of template proteins from a fold library. Then a score is obtained for each structure-sequence alignment. In the early methods the score in a Protein Threading was based on an empirical energy function, while more modern methods mostly or exclusively rely on sequence comparisons [10]. The underlying thought with this method is that there are a limited number of folds in nature and this will in time lead to all folds eventually

being known. This method is used when distantly related homologous that has similar fold can be detected [2][10].

1.1.1.3 Ab initio protein modeling

In Ab initio protein modeling, also called the de novo protein modeling the initial idea was that protein structures could be predicted purely on physical principles. Ab initio modeling is done by using environments and circumstances that are similar to those that occur during the folding process or looking at global optimization energy in search of a favorable energy function. The downside of this method is that it does not restrict the number of possible structures as efficient as the other two methods. The consequence of this is that the method cannot be used to model anything but tiny proteins, and even these predictions require vast amounts of computational resources [3]. This type of modeling is also considered to be the most difficult type of modeling compared with the other two methods [6]. This method has seen considerable progress through the observation that parts of the amino acid sequence can be accurately predicted through comparisons with segments from other proteins. This is done by using recombination on contiguous backbone fragments to fit the target sequence [10].

1.1.2 Local Descriptors of protein structures

Local Descriptors of protein structures is a recently developed method for looking at reoccurring three-dimensional structure elements in proteins. This method is a non rigid approach that shows similarities between proteins on a local level. These substructures can be composed of one or more segments of amino acid backbone structures that are in proximity of each other in space but not necessarily along the amino acid sequence as seen in *figure 1* [11]. The local descriptors that are of interest to this thesis are those that are composed of three or more structural fragments. These local substructures are then organized into a library consisting of groups that are similar in structures. This provides a set of building blocks for protein prediction that are common for proteins, even proteins that lack a common global structure but shows similar long range interactions [11]. The proteins that the descriptors were extracted from have less than 40% sequence identity to one another (ASTRAL) [7].

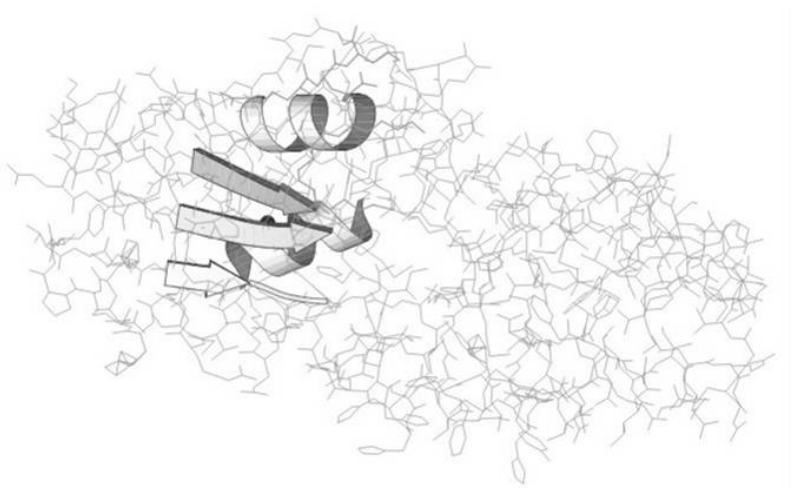


Figure 1. This is a figure showing the local descriptor denoted as 1qgoa. It consists of six fragments that are in proximity of one another in space but not necessarily along the amino acid sequence.

```
GROUP: 1io7a_#13 : 5
1cpt__#39  36-42    DEQPLAM  54-61    ATKHADVM  326-332  EVRGQNI  a.104 Pseudomonas sp.
1io7a_#13  10-16    KKDPVYY  23-30    VFSYRYTK  266-272  KLGDQTI  a.104 Archaeon Sulfolobus solfataricus
1jfba_#30  27-33    ATNPVSQ  45-52    VTKHKDVC  299-305  MIGDKLV  a.104 Fungus
1jipa_#28  25-31    ETAPVTP  42-49    VTGYDEAK  300-306  EIGGVAI  a.104 Saccaropolyspora erythraea
1n40a_#30  27-33    TREPIRK  45-52    VSSYALCT  293-299  QVGDVLV  a.104 Mycobacterium tuberculosis
```

Figure 2. This is an example of a descriptor group. It consists of three different fragments and has been found in five different proteins. The sequences for these reoccurring fragments can be seen aligned against one another with their positions in the protein sequence written in front of the segment Descriptors are named by using the syntax "domain name"# central amino acid position. The information seen in the back of each row is first the name of the fold this structure can be found and last is the name of the organism that the protein comes from.

1.1.3 Amino acid sequence analysis

There are several methods for analyzing amino acid sequences, the most commonly used methods are pattern recognition, profiles and hidden Markov models (HMM). These methods are most commonly geared towards recognizing known domains within an unknown protein sequence by showing what parts of a sequence are more conserved. A very simple method for studying relationships between proteins is to build a multi sequence alignment (MSA) [18]. Most patterns, profiles and HMM are built using multiple sequence alignments as training data.

1.1.3.1 Patterns

Patterns are the most simple and basic method for recognizing specific descriptive motifs in amino acid sequences. Patterns build on knowledge that certain short consecutive amino acid sequences sometimes can be associated to certain biological functions, such as binding sites or specific enzymatic activities [16]. These are patterns are then compared to the conserved parts of the amino acid sequence. These fingerprints for specific functions are generally 10-20 amino acids long [16]. Patterns have weaknesses due to their rigid nature, for example patterns are sometimes unable to recognize homologous that are not closely related because patterns can only handle a few number of mismatches between the patterns and the target sequences [16]. A good example of were this method is used is PROSITE, one of the more known bioinformatics sites in the world.

1.1.3.2 Profiles

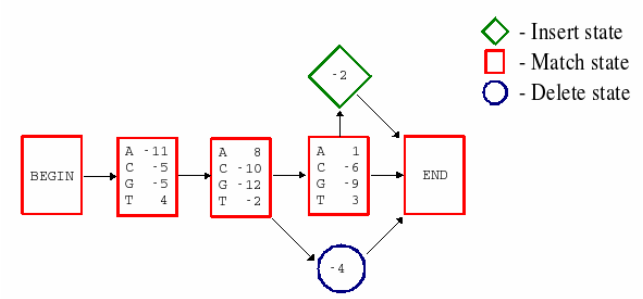


Figure 3.This is an example of a simple profile created to look for the sequence TAT. The numbers inside the red boxes are the scores received for the observed matches while the gap penalty scores can be seen inside the green box and the circle.

Profiles are sometimes referred to as position specific scoring matrices (PSSM). A profile is quantitative motif descriptor providing a numerical score for every possible match or mismatch between a sequence residue and a profile position, as can be seen in *Figure 3* [16]. One advantage of profiles compared to patterns is that they can use the whole protein sequence in its scoring and not only short and conserved parts of it. A profile can also take into account deletions and insertions when looking for certain motifs in a sequence, it does this by using gap and insertion penalties into the score that

allows the profile to look for better matches[22]. Profiles are used in a wide array of applications, they are used by PROSITE and Psi-Blast to look for homologous or characterizing families of proteins [16][21][22].

1.1.4 Hidden Markov Models

Hidden Markov Models are probabilistic models that are used on linear sequences in Bioinformatics [15]. Their uses in bioinformatics are wide, for example prediction of secondary structures from amino acid sequences, looking at possible protein-coding regions in genome sequences and other similar applications.

1.1.4.1 Definition for Markov Models

In order to understand and define HMM it is easier to begin with defining the more simple Markov models (MM) or Markov chains as they are also called. Consider a system which may described at any time as being in one of a set N distinct states, $\mathbf{S = S_1, S_2, \ldots S_N}$. At discrete intervals the system undergoes a change of state according to a set of probabilities connected with the states. These intervals are denoted as **t = 1, 2, ..., T** and the occupied state at time **t** is called $\mathbf{q_t}$. A full probabilistic description of this system at **t** would require specification of the state at **t** as well as all predecessor states. For first order discrete Markov chains the probabilistic description can be truncated to current and the previous state [14].

$$P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \cdots] = P[q_t = S_j | q_{t-1} = S_i]. \quad \textbf{(0-1)}$$

If only processes that are independent of **t** are considered, a set of transition probabilities can be formulated as:

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], \quad 1 \leq i, j \leq N$$

Where $\mathbf{a_{ij}}$, due to the fact that the state transition coefficient obey the standard stochastic constraints is:

$$a_{ij} \geq 0$$

$$\sum_{j=1}^{N} a_{ij} = 1$$

Since the signal of the process is a set of states at each **t** that each corresponds to a observation, this can be considered a Markov Model. If the number of possible states is limited to N. Then it can be postulated that at interval **t, t** can be characterized by one of the N number of limited states and that the matrix $\mathbf{a_{ij}}$ consisting of the states different transition probabilities can be defined as:

$$a_{ij} = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & & & \\ \cdots & & \cdots & \\ a_{N1} & & & a_{NN} \end{vmatrix}$$

If the state at **t = 1** is known, the probability of a sequence of observations **O = { S_1, S_2, …S_T}** that corresponds to **t = 1, 2, ..., T** can be answered. This probability can be formulated as:

$$\mathbf{P(O \mid Model) = P[S_1, S_2, \ldots S_T \mid Model]}$$

Where the initial state is denoted as π_i and has the following definition:

$$\pi_i = P[q_1 = S_i], \quad 1 \le i \le N$$

As can be seen, in Markov Models each state corresponds to an observable event.

1.142 Definition for Hidden Markov Models

In HMMs the observation is a probabilistic function of the state, this creates a doubly embedded stochastic signal with an underlying signal that cannot be observed directly, but that can be observed indirectly by another stochastic signal that produce a set of visible observations. To clarify the difference between MMs and HMMs, here comes an example in more general view. Given that you know the weather for a week, and the observation is that it rained seven days in a row. With Markov Models you could calculate the probability that it would rain seven days in a row if it was summer. With hidden Markov Models given the observation of seven days of rain, it would be possible to calculate which season of the year it is most likely to be. To formally define a hidden Markov Model, its basic elements must be properly defined.

A HMM is characterized by following parameters:

- The number of possible states in the model **N**. The individual states can be denoted as **S = S_1, S_2, …S_N.** The state at time **t** is denoted as q_t.
- The number of possible observations **M** denoted to **V = {V_1, V_2, …, V_M}**
- The state transition probability distribution **A (a_{ij}):**

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], \quad 1 \le i, j \le N$$

- The observation symbol probability in state **j, B ($b_i(k)$):**

$$b_j(k) = P[v_k \text{ at } t | q_t = S_j], \quad 1 \le j \le N$$
$$1 \le k \le M.$$

- The initial state distribution π_i:

$$\pi_i = P[q_1 = S_i], \quad 1 \le i \le N.$$

For a complete specification of a HMM as can be seen from the points above, two model parameters must be specified **N** and **M**. The probability measures must also be specified, meaning the measures **A**, **B** and $\boldsymbol{\pi}$. A convenient way to show this is to use the notation

$$\boldsymbol{\lambda = (A, B, \pi)}$$

So in short a hidden Markov model $\boldsymbol{\lambda}$ can be said to consists of a finite number of states $\mathbf{Q = (q_1, q_2, \ldots q_N)}$ that are tied together by multidimensional probability distributions **B**. These states are associated to one another through a set of transition probabilities in **A**, these transition probabilities depend only on the previous and current state $\mathbf{a_{ij}}$ **(0-1)**. Each state also gives off an emission probability $\mathbf{b_i(k)}$ depending on an observable event $\mathbf{O_t}$ (although there are exceptions due to different types of states) that is one of **M** possible different observations of **V**. So the hidden Markov Model can be defined as **(A, B, $\boldsymbol{\pi}$)**. As can be seen in *Figure 3*, the arrows are transition probabilities connecting the states shown as boxes or circles through **A.** The emission probabilities are seen inside the states are and are emitted from **B** through observations $\mathbf{O_t}$ that is one of **M = 4** possible different observations **V = {A, C, G, T}.**

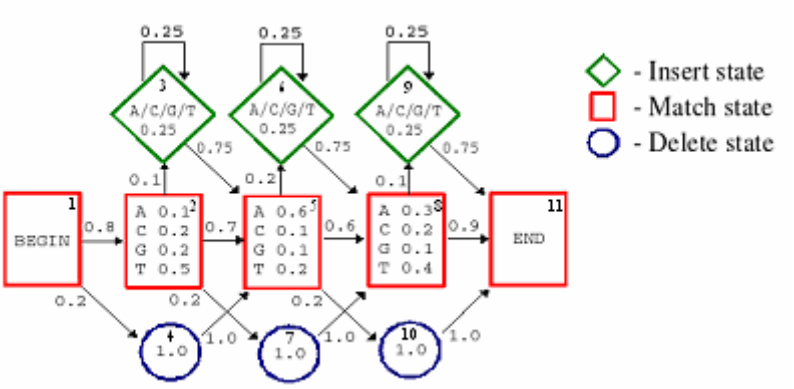


Figure 4. This is an example of a simple HMM for a short genomic sequence. In this example the HMM is trained for locating the sequence TAT. The arrows between the states are the possible transitions between the different states and the number close to the arrow is the transition probability for that transition. The numbers inside of the boxes (insert- and match- states) represent the emission probabilities for the observed events (A, C, G and T). The circles represent delete-states; in theory when using HMM delete-states should be non-emitting. In this model, delete states emit the probability 1.0 independent of the observed event which yields the same result as if the delete states were non-emitting.

In most cases in bioinformatics the series of ordered observable events $\mathbf{(O_1, O_2, \ldots O_T)}$ will be an amino acid, genomic- or secondary structure sequence. A formal technique for finding the single best state path sequence through a hidden Markov Model, i.e. to maximize $\mathbf{P(Q|O, \lambda)}$ which is the same as to maximize $\mathbf{P(Q, O \mid \lambda)}$ **[**17**]** is used. This

technique is based on dynamic programming methods and is referred to as the Viterbi Algorithm [12][13]. In order to find the optimal path $\mathbf{Q = \{q_1, q_2, \ldots q_T\}}$ through the observed sequence $\mathbf{O = \{O_1, O_2, \ldots O_T\}}$ there is need to define the best path at position **t** in the sequence which is denoted $\boldsymbol{\delta_t(i)}$ and is the highest probability at interval t**.**

$$\boldsymbol{\delta_t(i) = \max(q_1, q_2 \ldots q_{t-1})\ P[q_1\ q_2 \ldots q_t = i,\ O_1\ O_2\ \ldots\ O_t \mid \lambda]} \quad \textbf{(1-1)}$$

By use of induction on $\boldsymbol{\delta_t(i)}$, $\boldsymbol{\delta_{t+1}(j)}$ can be solved [17].

$$\boldsymbol{\delta_{t+1}(j) = [\max \delta_t(i)\ a_{ij}]\ b_j(O_{t+1})} \quad \textbf{(1-2)}$$

In order to be able to backtrack later there is a need to hold onto the arguments that maximized **(1-2)** for every **t** and **j**. These values will be stored in the matrix $\boldsymbol{\psi(t, j)}$**.** The most probable path through the HMM can now found by following these three statements.

Initialization:

$$\boldsymbol{\delta_1(i) = \pi_i b_i(O_i)\ \text{ for every } 1 \le i \le N} \quad \textbf{(1-3)}$$

$$\boldsymbol{\psi(1, N) = 0\ \text{ for every } 1 \le i \le N} \quad \textbf{(1-4)}$$

Intermediate:

$$\boldsymbol{\delta_t(j) = (\max(1 \le i \le N)\ [\delta_{t-1}(i)\ a_{ij}])\ b_i(O_j) \text{ for } 2 \le t \le T \text{ and } 1 \le j \le N} \quad \textbf{(1-5)}$$

$$\boldsymbol{\psi(t, j) = \operatorname{argmax}(1 \le i \le N)\ [\delta_{t-1}(i)\ a_{ij}] \text{ for } 2 \le t \le T \text{ and } 1 \le j \le N} \quad \textbf{(1-6)}$$

Termination:

$$\boldsymbol{P^* = \max(1 \le i \le N)\ [\delta_t(i)]} \quad \textbf{(1-7)}$$

$$\boldsymbol{q_t^* = \operatorname{argmax}(1 \le i \le N)\ [\delta_t(i)]} \quad \textbf{(1-8)}$$

Backtracking to find optimal path:

$$\boldsymbol{q_t^* = \psi_{t+1}(q_{t+1}^*),\ t = T\text{-}1,\ T\text{-}2, \ldots, 1} \quad \textbf{(1-9)}$$

This solution, the Viterbi algorithm is very powerful solution for finding the optimal path through a HMM. The great advantage of HMM compared with profiles and patterns is that they are built upon probability theory and can be said to be more mathematically robust then profiles and patterns [17]. Another advantage with the HMMs is that they incorporate insertions and deletions in a far more natural manner then profiles.

1.2 Aims of the Master thesis

The main purpose of this Master Thesis is to develop an application for recognizing and locating local descriptors of protein structures in target sequences. The main focus of the project has been to develop computer application for building and using profile HMMs. Secondary objectives for the thesis have been to evaluate and optimize the application.

Chapter 2

Results

2.1 General information

The HMMs in this thesis are built to find short segments of amino acid sequences that are spread out in different domains of the protein. The idea in this thesis is to model the descriptor fragments as matches and the rest of the sequence as inserts. Deletions do occur but only at the beginning and the end parts of the segments. In order to ensure that these deletions do not interconnect, and delete whole fragments the model uses two different types of match states denoted $\mathbf{D_B}$ and $\mathbf{D_E}$ for delete-begin delete-end respectively. No transitions are allowed between the state types $\mathbf{D_B}$ and $\mathbf{D_E}$. To measure the accuracy and performance of the models Cross validation is used. This is a statistical method where the initial training data is partitioned into smaller subsets of data called test sets and these test sets are used to confirm and validate the performance of the method. The type used in this thesis is Leave one out cross-validation. Here you use a single observation from the training set as test-set when you perform your validation and the remaining observations are used as training data, this is then repeated so that all observations in the training data are used one time in the validation of the method [19]. One thing that is different in the cross validation compared to the normal definition we perform is that observations from the same protein domains are also removed from the training data in order to ensure that the models do no receive an unfair advantage when being validated. Each descriptor in every descriptor group will be used as an observation that a validation can be performed on. The parameter that is being measured is the ability of each group to find the proper positions of the descriptor from its corresponding amino acid sequence when this descriptor or descriptors have been removed from the training data of that specific group. In order to get a proper evaluation of the accuracy of our models we specifically look and see if match states and delete states have been properly aligned between the models answer and the known correct alignment. The answer received is a percentage of how many of the match and delete states have been properly aligned between the known correct answer and the answer from the generated hidden Markov model. The total number of local descriptor groups available numbered about seven thousand groups. Cross validating the full data set if everything went smoothly would take about three days, so in order to be able to faster create and validate improvements a test set was created. This **test-set** was created by randomly picking groups from the full data set trying to get a similar distribution of descriptor groups concerning the internal size of the groups meaning the number of descriptors that compromising the groups as this seemed to be the parameter that visible parameter that effected group performance the most. In total 176 groups were picked out, ranging in descriptor size from 5 descriptors per group to 52 descriptors per group.

2.2 Different Models

2.2.1 Simple Amino Acid Model (Model One)

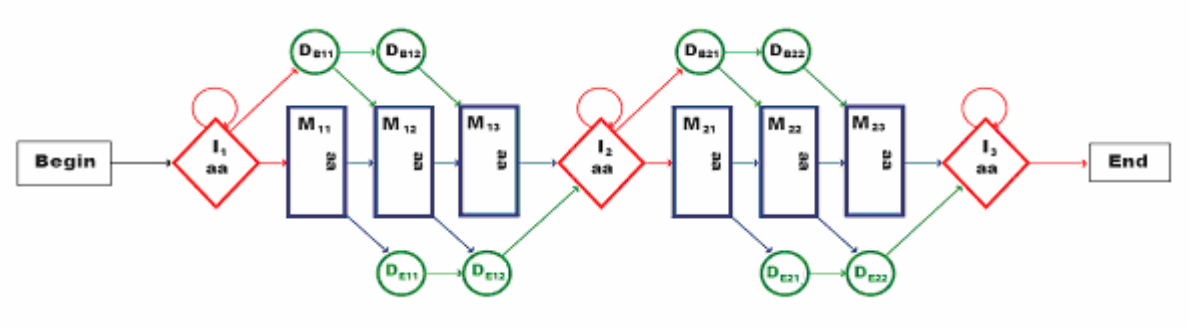
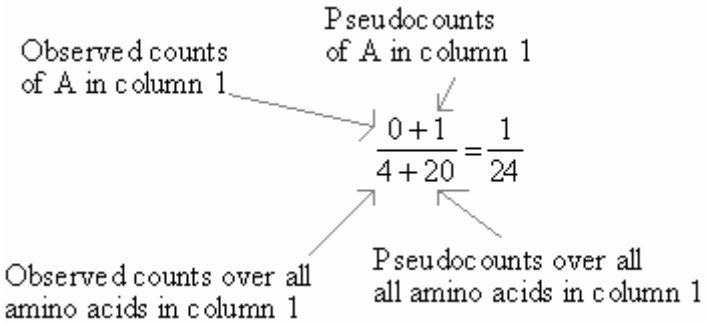


Figure 5 This figure shows the basic form that was used to build the simple amino acid model types of HMM. It consists of match-, insert- and two types of delete-states that are separated from one another called $\boldsymbol{D_B}$ *and* $\boldsymbol{D_E}$*. The boxes that contain* ***aa*** *are boxes that send out emission probabilities dependent on observed amino acids in that state.*

This model was the simplest model built, it built on the principle that all fragments in the descriptors were separate in the amino acid sequences and that these fragments were neither in the beginning nor at the end of the amino acid sequences as can be seen in *Figure 5*. This is quite a simplification and does not fully take into account and use the information that can be found in the training data. The main state types are match- (**M**), insert- (**I**), delete-begin- ($\mathbf{D_B}$) and delete-end- ($\mathbf{D_E}$) states. The idea is to that the segments that make up the LDPS are modeled as match-states and the sequences in between the segments are modeled as insert-states. Delete states are tied to segments as seen in the model, delete states only exist if deletes are seen in the training data tied to the positions that they have been observed in. The emissions in this model were calculated in a simplified manner, they were calculated using a method called pseudo count that can be seen in *example 1*.



Example 1 This is an example of a pseudo count of an amino acid. In this pseudo count the alignment used to build emission matrix consists of four aligned sequences. In this example and for the specified residue (A) there are no observations of this residue seen in the position of the alignment (column).

The downside of pseudo count is that it does not take account discriminate common substation patterns to uncommon ones due to similar physical and chemical properties between certain known amino acids like charge and polarity of the residues. The transition probabilities were modeled solely on observed transitions in the training data. In this model transition from the begin state always goes to the insert. The transition possibilities for inserts were calculated from the mean number of inserts seen in the training data, meaning mean number of amino acids between descriptor segments, combined with possible delete-begin states due to the frequency of deletes in the first position in the segment. Delete-begin states transition probabilities are calculated from the frequency of interconnected delete-begin states in the next position, when the frequency reaches zero the delete-begin state is forced to enter a match state. Match states are calculated through looking at the frequency of amino acid matches in the next position versus interconnected delete-end-states as long as the segment end is not reached as because when this happen the match state is forced into the next insert state. This model was a good starting point for the applications as it is easier to focus on building a solid base for the application not having to take every possible scenario into account. The result of applying this model to the test set can be seen in *Table 1*.

Table 1. This is the result from the Leave one out cross validation performed on the test-set using Model one. The first row shows how many descriptor groups has been validated, the second row shows how many zero descriptors (descriptors with 0% as a result). The third row contains the number of descriptors that were perfectly aligned with their proper position. The fourth row contains the number of inserts in all cross validations and the row below that one contains the number of non inserts in the cross validation (M match-state, C delete-begin-state, D delete-end-state), the reason for this is that it is the non inserts that we wish to align correctly and by gathering this information we know the fraction of our search area compared to the full area.

Results from Simple Amino Acid Model	
No of descriptor groups Crossvalidated:	176
No of descriptors with 0%:	392
No of descriptors with 100%:	137
Number of I:	379718
Number of M, C, D :	45059
Total group %:	38,9%
No of descriptors analyzed:	1553

2.2.2 Extended Amino Acid Model (Model Two)

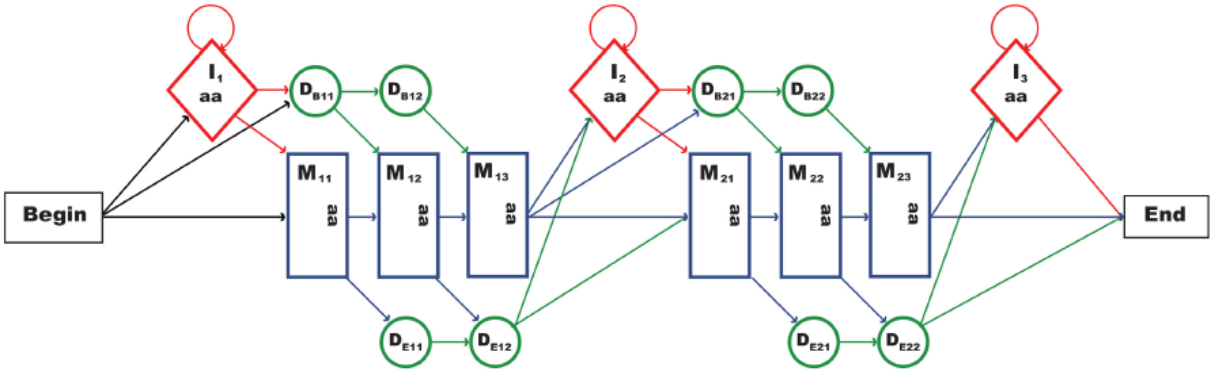


Figure 6 This is a more sophisticated model then the one shown in Figure 5 and this model allows for different fragments to be connected directly to one another in the amino acid sequence. This model also allows for the possibility that fragments may be located in the beginning and the end of the target sequence.

This model is an improvement from Model 1 in several ways as it takes into account the possibility that a fragment can be located at both the end and the beginning of an amino acid sequence. This model also allows for the possibility that two separate fragments can be located next to one another in the amino acid target sequence and does not have to be separated by insert states as can be seen in *Figure 6*, but these transitions are only allowed if this is observed in the training data. The emission matrix has also seen improvements as the emission state probabilities are not calculated from pseudo counts but by using the substitution matrix BLOSUM62, the explicit matrix used in this thesis can be seen in **Appendix A.** The BLOSUM62 is a two dimensional matrix (20x20) were every row (and column) represents the substitution for a specific amino acid to the other amino acids. The BLOSUM matrixes were built by looking at blocks of aligned sequence segments looking for all possible exchanges between different amino acids [27]. This matrix is recalculated into an approximate probability matrix by looking at lowest score (score minimum) in a row then add absolute value of (score minimum + 1) to all positions in the row, this ensures that all values are positive integers > 0. All positions in the row are then divided by the sum of all positions in the row. This procedure is then repeated for all rows in the matrix. The substation matrix is then used to build the emission matrix by adding the “probabilities” for a specific residue received from the observed amino acid at that state, and when all observations have been added the probability is divided by the number of observations, yielding the mean value for the probabilities received from the substitution matrix. This repeated for all different residues The results from the cross validation using Model two on the test set can be seen in *Table 2* below.

Table 2 The table shows the result yielded from the Leave one out cross validation performed on the test-set using HMM built from the Model Two system. This Table contains more information than previous table as result that more parameters were

needed to properly be able to compare the different models. The new parameters added were descriptor groups that consisted solely of zero descriptors (row 3) and groups that were perfectly aligned (row 5). The last parameter added was the descriptor average to see the performance of the models if the descriptors were looked at independent of their group.

Results from Extended Amino Acid Model	
No of descriptor groups Crossvalidated:	176
No of descriptors with 0%:	379
No of descriptor groups with 0%:	2
No of descriptors with 100%:	132
No of descriptor groups with 100%:	1
Number of I:	379718
Number of M, C, D :	45059
Total group %:	39,5%
Total desc %:	41,1%
No of descriptors analyzed:	1553

2.2.3 Combined Dual Hidden Markov Model (Model Three)

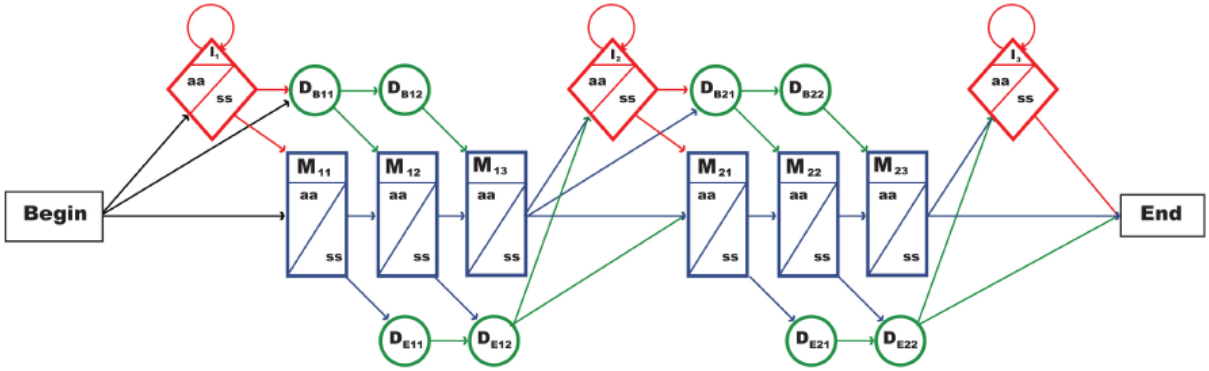


Figure 6 The Combined Dual Hidden Markov Models were built from Extended Amino Acid Model but to include known information about secondary structure ***ss****. This model does this by combing the amino acid* ***aa*** *observation with the secondary structure observation* ***ss*** *and emitting them together.*

This model was the first model built that utilized both the secondary structure information as well as amino acid information. In this model the secondary structures and the amino acid observations are combined and emitted together as seen in *Figure* 6. The emission probabilities are calculated through pseudo count combinations of amino acids and secondary structures **(b_{cd})**. In order to find the best path, Viterbi is used but with a few adjustments to be able to accommodate the second data set used in the hidden Markov Model. The second data set containing the secondary structure observations is defines as $\mathbf{O_{ss} = \{O_{ss,1}, O_{ss,2}, \ldots O_{ss,T}\}}$ which is the observed sequence of the secondary structure parallel to the observed structure of the amino acid sequence $\mathbf{O = \{O_1, O_2, \ldots O_T\}}$. The best path through the Combined Dual Hidden Markov Model can be found by following the steps below (**2-1** to **2.9**).

As before it is necessary to define the best possible path at time t, $\boldsymbol{\delta_t(i)}$.

$$\boldsymbol{\delta_t(i) = \max(q_1, q_{2,\ldots}q_{t-1})\ P[q_1\ q_2 \ldots q_t = i,\ O_1\ O_2\ \ldots O_t\ ,\ O_{ss,1}\ O_{ss,2}\ \ldots\ O_{ss,t}\ |\ \lambda]} \qquad \textbf{(2-1)}$$

As previously by use of induction on $\boldsymbol{\delta_t(i)}$, $\boldsymbol{\delta_{t+1}(j)}$ can be solved.

$$\boldsymbol{\delta_{t+1}(j) = [\max\ \delta_t(i)\ a_{ij}]\ b_{cd,j}(O_{t+1}, O_{ss,t+1})} \qquad \textbf{(2-2)}$$

What happens here in $\mathbf{b_{cd,j}(O_{t+1}, O_{ss,t+1})}$ is that instead of emitting a character representing an amino acid, which is then compared against twenty characters (one per amino acid) to send back the proper emission probability. What happens is that two chars are combined into a string compromised of two chars (one chat for amino acids and one for secondary structures) that is emitted and compared against sixty other strings compromised of two chars (all possible combinations of secondary structures and amino acids) sending back the proper emission probability.

Initialization:

$$\delta_1(i) = \pi_i b_{cd,i}(O_i, O_{ss,i}) \text{ for } 1 \leq i \leq N \quad (2\text{-}3)$$

$$\psi(1, N) = 0 \text{ for } 1 \leq i \leq N \quad (2\text{-}4)$$

Intermediate:

$$\delta_t(j) = (\max(1 \leq i \leq N) [\delta_{t-1}(i)\, a_{ij}])\, b_{cd,i}(O_j, O_{ss,j}) \text{ for } 2 \leq t \leq T \text{ and } 1 \leq j \leq N \quad (2\text{-}5)$$

$$\psi(t, j) = \text{argmax}\ (1 \leq i \leq N) [\delta_{t-1}(i)\, a_{ij}] \text{ every } 2 \leq t \leq T \text{ and } 1 \leq j \leq N \quad (2\text{-}6)$$

Termination:

$$P^* = \max(1 \leq i \leq N) [\delta_t(i)] \quad (2\text{-}7)$$

$$q_t^* = \text{argmax}(1 \leq i \leq N) [\delta_t(i)] \quad (2\text{-}8)$$

Backtracking to find optimal path:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), t = T\text{-}1, T\text{-}2, \ldots, 1 \quad (2\text{-}9)$$

The result given using the extended weighted Viterbi Algorithm with Combined Dual Hidden Markov Models yielded the results seen in *Table 3*.

Table 3 This table shows the results from using HMM built from Model Three and using both amino acid- and secondary structure sequence information.

Results from Combined Dual Hidden Markov Model	
No of descriptor groups Crossvalidated:	176
No of descriptors with 0%:	247
No of descriptor groups with 0%:	14
No of descriptors with 100%:	241
No of descriptor groups with 100%:	1
Number of I:	339390
Number of M, C, D :	40696
Total group %:	51,3%
Total desc %:	52,8%
No of descriptors analyzed:	1482

2.2.4 Dual Emitting Hidden Markov Model (Model Four)

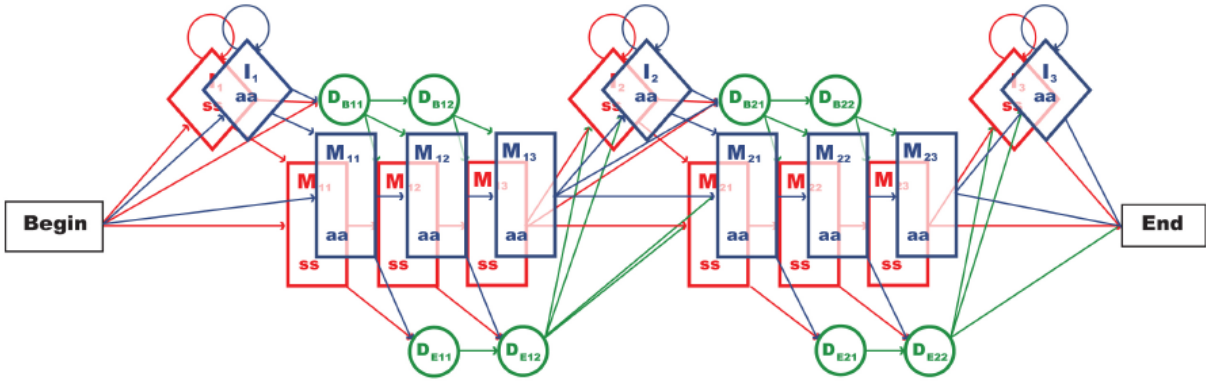


Figure 6 Dual Emitting Hidden Markov Model was built from Extended Amino Acid Models to utilize both amino acid and secondary structure information. The difference between model Three and this model is that this model emits two separate emission probabilities in every state, one amino acid emission probability and one secondary structure probability.

The difference between this model and the previous model is that in this model the secondary structure and the amino acid sequences have separate emission matrixes. This allows them to be built separately. In this model the amino acid sequence emission matrix was built using a substitution matrix while the secondary structure matrix was built using pseudo counts ($\mathbf{b_{ss}}$) as seen in *Figure 6*. In order to use two separate emission matrixes that each corresponds to two linear series of linked parallel observations $\mathbf{O} = \{O_1, O_2, \ldots O_T\}$ and $\mathbf{O_{ss}} = \{O_{ss,1}, O_{ss,2}, \ldots O_{ss,T}\}$ some modifications must be made to the Viterbi algorithm. The steps (**3-1** to **3-9**) needed to find the most probable path through Dual Emitting Hidden Markov Model can be seen below.

As before it is necessary to define the best possible path at time t, $\boldsymbol{\delta_t(i)}$.

$$\boldsymbol{\delta_t(i) = \max(q_1, q_{2,\ldots}q_{t-1})\ P[q_1\ q_2 \ldots q_t = i,\ O_1\ O_2\ \ldots O_t\ ,\ O_{ss,1}\ O_{ss,2}\ \ldots\ O_{ss,t}\ |\ \lambda]} \qquad \textbf{(3-1)}$$

As previously by use of induction on $\boldsymbol{\delta_t(i)}$, $\boldsymbol{\delta_{t+1}(j)}$ can be solved.

$$\boldsymbol{\delta_{t+1}(j) = [\max\ \delta_t(i)\ a_{ij}]\ b_j(O_{t+1}) b_{ss,j}(O_{sst+1})} \qquad \textbf{(3-2)}$$

What happens here is that two characters are emitted separately. In $\mathbf{b_j(O_{t+1})}$ one char representing an amino acid is emitted compared against twenty other characters and the associated probability is sent back. What happens in $\mathbf{b_{ss,j}(O_{sst+1})}$ is that a character representing a secondary structure is emitted and compared against three other character, one per secondary structure type and the corresponding probability is sent back were the two probabilities are multiplied together.

Initialization:

$$\boldsymbol{\delta_1(i) = \pi_i b_i(O_i) b_{ss,i}(O_{ss,i})\ \ \text{for } 1 \le i \le N} \qquad \textbf{(3-3)}$$

$$\psi(1, N) = 0 \text{ for } 1 \leq i \leq N \quad (3\text{-}4)$$

Intermediate:

$$\delta_t(j) = (\max(1 \leq i \leq N) [\delta_{t-1}(i)\, a_{ij}])\, b_i(O_j) b_{ss,i}(O_{ss,j}) \text{ for } 2 \leq t \leq T \text{ and } 1 \leq j \leq N \quad (3\text{-}5)$$

$$\psi(t, j) = \text{argmax}\ (1 \leq i \leq N) [\delta_{t-1}(i)\, a_{ij}] \text{ every } 2 \leq t \leq T \text{ and } 1 \leq j \leq N \quad (3\text{-}6)$$

Termination:

$$P^* = \max(1 \leq i \leq N) [\delta_t(i)] \quad (3\text{-}7)$$

$$q_t^* = \text{argmax}(1 \leq i \leq N) [\delta_t(i)] \quad (3\text{-}8)$$

Backtracking to find optimal path:

$$q_t^* = \psi_{t+1}(q_{t+1}^*),\ t = T\text{-}1, T\text{-}2, \ldots, 1 \quad (3\text{-}9)$$

In *Table 4* results from using the Dual Emitting Hidden Markov Model with the non-weighted Viterbi Algorithm on the test set can be seen.

Table 4 Here the results from Dual Emitting Hidden Markov Model used on the test set can be seen.

Results from Dual Emitting Hidden Markov Model	
No of descriptor groups Crossvalidated:	176
No of descriptors with 0%:	247
No of descriptor groups with 0%:	13
No of descriptors with 100%:	241
No of descriptor groups with 100%:	1
Number of I:	339390
Number of M, C, D :	40696
Total group %:	54,9%
Total desc %:	57,8%
No of descriptors analyzed:	1482

2.2.5 Hybrid Hidden Markov Inspired Model (Model Five)

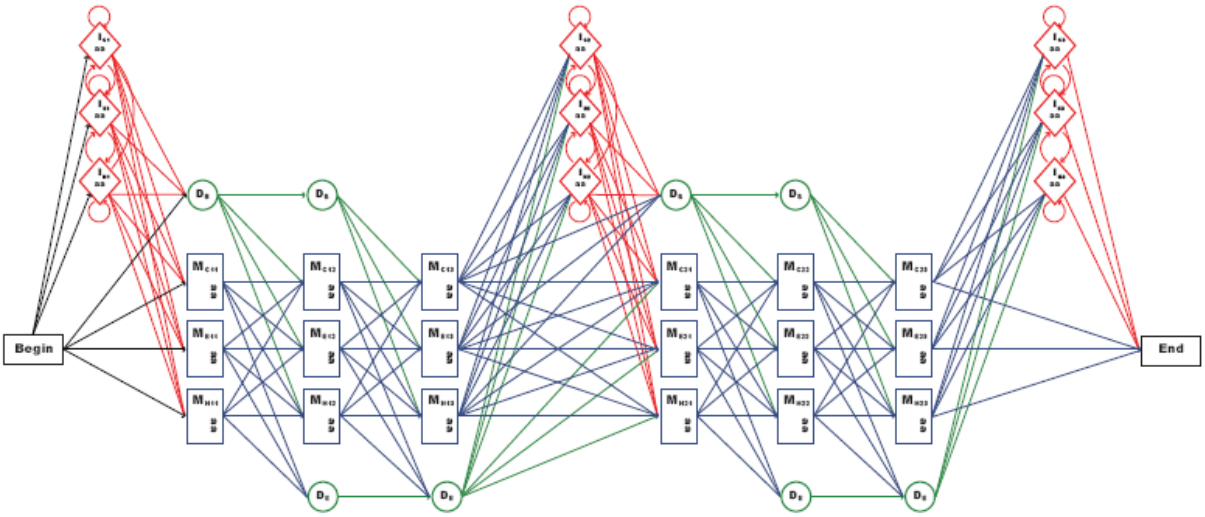


Figure 7 This model was built using Extended Amino Acid Models to use known transitions between certain secondary structures by creating three separate parallel states that each corresponds to a specific secondary structure (C, E, H). This is done for all match- and insert- states.

In this model the idea was to use the possibility of transitions between certain secondary structures in the transition matrix by creating three separate parallel matrix states for every previous insert and match state each corresponding to a certain a secondary structure (C, E, H). This model only utilizes one emission matrix, for amino acid observations. The difference in this model compared with previous models is that instead of looking at the possibility of a certain secondary structure emission at a given state you look at the probability of a certain transition between two secondary structures from one state to another. This is the reason for calling this model Hidden Markov inspired rather than calling it a Hidden Markov Model as the transitions are not only dependent of the current state but also indirectly of the previous state due to the fact the state in itself is state dependent. In order to find the most probable path through this model a Viterbi inspired algorithm is used. This can be seen below (**4-1** to **4-9**).

$$\boldsymbol{\delta_t(i) = \max(q_1, q_2 \ldots q_{t-1})\ P[q_1\, q_2 \ldots q_t = i,\ O_1\, O_2 \ldots O_t\ ,\ O_{ss,1}\, O_{ss,2} \ldots O_{ss,t} \mid \lambda]} \qquad \textbf{(4-1)}$$

By use of induction on $\boldsymbol{\delta_t(i)}$, $\boldsymbol{\delta_{t+1}(j)}$ can be solved.

$$\boldsymbol{\delta_{t+1}(j) = [\max \delta_t(i)\ a_{ij}\ (O_{ss,j})]\ b_j(O_{t+1})} \qquad \textbf{(4-2)}$$

Initialization:

$$\boldsymbol{\delta_1(i) = \pi_i b_i(O_i) \quad \text{for every } 1 \le i \le N} \qquad \textbf{(4-3)}$$

$$\boldsymbol{\psi(1, N) = 0 \quad \text{for every } 1 \le i \le N} \qquad \textbf{(4-4)}$$

Intermediate:

$$\delta_t(j) = (\max(1 \leq i \leq N) [\delta_{t-1}(i)\, a_{ij}\, (O_{ss,j})])\, b_i(O_j) \text{ for } 2 \leq t \leq T \text{ and } 1 \leq j \leq N \quad \textbf{(4-5)}$$

$$\psi(t, j) = \operatorname{argmax}(1 \leq i \leq N) [\delta_{t-1}(i)\, a_{ij}\, (O_{ss,j})] \text{ for } 2 \leq t \leq T \text{ and } 1 \leq j \leq N \quad \textbf{(4-6)}$$

Termination:

$$P^* = \max(1 \leq i \leq N) [\delta_t(i)] \quad \textbf{(4-7)}$$

$$q_t^* = \operatorname{argmax}(1 \leq i \leq N) [\delta_t(i)] \quad \textbf{(4-8)}$$

Backtracking to find optimal path:

$$q_t^* = \psi_{t+1}(q_{t+1}^*),\ t = T-1, T-2, \ldots, 1 \quad \textbf{(4-9)}$$

The result from using this model on the test set can be seen down below in *Table 5*.

*Table 5. Showing the result using the above model finding the best possible path by following the steps above **(4-1** to **4-9)** the following results were given when used on the test set.*

Results from Hybrid Hidden Markov Inspired Model	
No of descriptor groups Crossvalidated:	176
No of descriptors with 0%:	292
No of descriptor groups with 0%:	14
No of descriptors with 100%:	177
No of descriptor groups with 100%:	2
Number of I:	339390
Number of M, C, D :	40696
Total group %:	48,7%
Total desc %:	50,0%
No of descriptors analyzed:	1482

2.3 Different Target Data

Using the model that yielded the best result for this application type, which was the Dual Emitting Hidden Markov Model further attempts to improve accuracy have been made by building alignments from the target amino acid sequences and then using these multi sequence alignments to thread through the hidden Markov Models. The idea was to try and improve our performance by utilizing the evolutionary information available in alignments made up from homolog protein sequences. Attempts using MSA for threading built from BLAST was tried with two different alignments. In order to be able to execute Viterbi on this, a couple of modulations must be done to the algorithm and a few new definitions must be declared.

$$\mathbf{O_{\gamma,\,j} = (\gamma_{j1},\, \gamma_{j2} \ldots\, \gamma_{jm})\ for\ 1 \le m \le 20}$$

Were γ is the fraction of a specific amino acid type in column j in the alignment being threaded. So **O** is no longer a sequence of observed linear sequences but rather an array consisting of probabilities for a limited number of possible observations **T** for that position:

$$\mathbf{O = \{ O_{\gamma,1}, O_{\gamma,2} \ldots O_{\gamma,T}\}}$$

The change from a single observed event to an array of probabilities for a limited number of possible observations puts new demands on the emission function $\mathbf{b_j(O_{t+1})}$, in order to cope with these demands the following definition has been created:

$$\mathbf{B_i(O_{\gamma,\,j}) = b_i(\gamma_{j1})\ b_i(\gamma_{j2})\ \ldots\ b_i(\gamma_{jm})}$$

Using these definitions the best possible path through the Dual Emitting Hidden Markov Model can be found by following these steps.

As before it is necessary to define the best possible path at the time t, $\mathbf{\delta_t(i)}$.

$$\mathbf{\delta_t(i) = \max(q_1, q_{2,\ldots}q_{t-1})\ P[q_1\, q_2 \ldots q_t = i,\, O_{\gamma,1}\, O_{\gamma,2}\, \ldots O_{\gamma,t}\, ,\, O_{ss,1}\, O_{ss,2}\, \ldots\, O_{ss,t} \mid \lambda]} \qquad \textbf{(5-1)}$$

As previously by use of induction on $\mathbf{\delta_t(i)}$, $\mathbf{\delta_{t+1}(j)}$ can be found:

$$\mathbf{\delta_{t+1}(j) = [\max\, \delta_t(i)\, a_{ij}]\ B_j(O_{\gamma,t+1}) b_{ss,j}(O_{sst+1})} \qquad \textbf{(5-2)}$$

Initialization:

$$\mathbf{\delta_1(i) = \pi_i B_i(O_{\gamma,i}) b_{ss,i}(O_{ss,i})\ \ for\ 1 \le i \le N} \qquad \textbf{(5-3)}$$

$$\mathbf{\psi(1, N) = 0\ \ for\ 1 \le i \le N} \qquad \textbf{(5-4)}$$

Intermediate:

$$\mathbf{\delta_t(j) = (\max(1 \le i \le N)\ [\delta_{t-1}(i)\, a_{ij}])\ B_i(O_{\gamma,j}) b_{ss,i}(O_{ss,j})\ for\ 2 \le t \le T\ and\ 1 \le j \le N} \qquad \textbf{(5-5)}$$

$$\mathbf{\psi(t, j) = \operatorname{argmax}\ (1 \le\ i \le N)\ [\delta_{t-1}(i)\, a_{ij}]\ every\ 2 \le t \le T\ and\ 1 \le j \le N} \qquad \textbf{(5-6)}$$

Termination:

$$P^{*} = \max(1 \le i \le N)\ [\delta_t(i)] \quad \textbf{(5-7)}$$

$$q_t^{*} = \operatorname{argmax}(1 \le i \le N)\ [\delta_t(i)] \quad \textbf{(5-8)}$$

Backtracking to find optimal path:

$$q_t^{*} = \psi_{t+1}(q_{t+1}^{*}),\ t = T-1, T-2,\ldots,1 \quad \textbf{(5-9)}$$

2.3.1 Ungapped Blast Alignment (Alignment One)

The First Alignment was built from blasting all the target sequences against Swissprot using the cutoff value 0,05, using ungapped Blast. The alignments were built from a maximum of 20 sequences, in this case the alignments with the lowest E-values were picked first. The result from using Dual Emitting Hidden Markov Model with the weighted fraction Viterbi algorithm (**5-1** to **5-9**) can be seen in *Table 6*.

Table 6 This table shows the results from running an amino acid alignment and a secondary structure sequence through Dual Emitting Hidden Markov Models with the test set using the expanded Viterbi algorithm to find the most probable path.

Results from Ungapped Blast Aligment	
No of descriptor groups Crossvalidated:	176
No of descriptors with 0%:	441
No of descriptor groups with 0%:	29
No of descriptors with 100%:	130
No of descriptor groups with 100%:	0
Number of I:	339390
Number of M, C, D :	40696
Total group %:	40,5%
Total desc %:	45,0%
No of descriptors analyzed:	1482

2.3.2 Gapped Blast Alignment (Alignment Two)

This alignment was built using gapped BLAST with a cutoff value E-value of less than 0,0001. In this alignment there were no restrictions set to how many sequences the alignments were allowed to be made up from. The results from the test set using the model seen in *Figure 6* with the fraction weighted Viterbi (**5-1** to **5-9**) can be seen in *Table 7*.

Table 7 This table shows the results from running an amino acid alignment and a secondary structure sequence through Dual Emitting Hidden Markov Models with the test set using the expanded Viterbi algorithm to find the most probable path.

Results from Gapped Blast Aligment	
No of descriptor groups Crossvalidated:	176
No of descriptors with 0%:	435
No of descriptor groups with 0%:	29
No of descriptors with 100%:	134
No of descriptor groups with 100%:	0
Number of I:	339390
Number of M, C, D :	40696
Total group %:	40,8%
Total desc %:	45,6%
No of descriptors analyzed:	1482

2.3.2 Profile PSSM (Alignment Three)

In order to try and fully use the evolutionary information in the target data we built profiles for all target sequences using psi-blast set to iterate three times. These profiles were then extracted of amino acid scores that were rearranged to be positive integers but keeping the distance between consistent. These scores were then divided with the sum of all other scores yielding an array consisting of probabilities all amino acids for all positions in the array. This array was then threaded through the Dual Emitting Hidden Markov Models yielding the result seen below in *Table 8* when testing against the test-set.

*Table 8. This table shows the results received when running Profiles built from Psi-Blast with three iterations. This array was threaded through the Dual Emitting Hidden Markov Models with the test set using the rules (**5-1** to **5-9**) giving the results seen below.*

Results from Profile	
No of descriptor groups Crossvalidated:	176
No of descriptors with 0%:	263
No of descriptor groups with 0%:	15
No of descriptors with 100%:	281
No of descriptor groups with 100%:	3
Number of I:	339390
Number of M, C, D :	40696
Total group %:	55,6%
Total desc %:	58,5%
No of descriptors analyzed:	1482

Chapter 3

Discussion

Table 8 This table summarizes the results presented in chapter 2.

Summary	Model 1.	Model 2.	Model 3.	Model 4.	Model 5.	Blast 1.	Blast 2.	Profile
No of descriptor groups Crossvalidated:	176	3	176	176	176	176	176	176
No of descriptors with 0%:	362	379	247	247	292	441	435	263
No of descriptor groups with 0%:	----------	2	14	13	14	29	29	15
No of descriptors with 100%:	137	132	241	241	177	130	134	281
No of descriptor groups with 100%:	----------	1	1	1	2	0	0	3
Antal I:	379718	379718	339390	339390	339390	339390	339390	339390
Antal M, C, D :	45059	45059	40696	40696	40696	40696	40696	40696
Total group %:	38,9%	39,5%	51,3%	54,9%	48,7%	40,5%	40,8%	55,6%
Total desc %:	----------	41,1%	52,8%	57,8%	50,0%	45,0%	45,6%	58,5%
Number of analisys:	1553	1553	1482	1482	1482	1482	1482	1482

3.1 Discussion - Amino Acid models

As can be seen in *Table 8* the extended amino acid model performed better then the simple model, locating groups of descriptors with an increase of 0.6% (38.9% for the simple model versus 39.5% for the extended model) per group. That the extended amino acid model performed better then the simple amino acid model is perhaps not surprising as it fits the data better and because it allows fragments to exist both at the end and the beginning of the target sequences, it is perhaps more surprising that the model does not improve the accuracy more. This could in part be explained by pointing out that the extended model only allows fragments to be located for better fitting if the data allows it, meaning that fragments may only be located for example in the beginning of the sequence for a descriptor group if there is data supporting that position within that group. For example if you have a descriptor group consisting of five different local descriptors from five different proteins where only one of the descriptors is known to be found for example in the beginning of its training sequence, when doing the leave one out cross validation on the descriptor found in the beginning, the data allowing fragments to be found in the beginning is removed from the data set used to build the HMM. The consequence of this is that HMM built from this smaller dataset cannot find 100% of the first fragment as is unable to look for the fragment where it is supposed to be located. The same thing happens if you have groups consisting of five descriptors were four of the fragments were located at the beginning of the training data sequences, when doing the cross validation on the fifth descriptor the model would be wrong when looking for the first fragment by default as it would be forced to look in the wrong place by default. This is a price paid for when using hidden Markov Models which has its roots in the methods solid roots in mathematics when you have insufficient or to limited amount of data. When looking at the result from using the extended amino acid model (model 2.) it might be observed that there is a difference when comparing the accuracy between descriptor groups and individual descriptor average.

As can be noted when looking at the table the descriptor accuracy 39.5% is higher then the descriptor group accuracy 41.1%. This is something of a surprise as bigger groups should be harder to locate. Beacuse the more general a structure is, i.e. can be found several different folds, the more general this structure is in proteins, making it less conserved sequentially. So a descriptor group consisting of more descriptors is less specific to different folds and so more general by nature, making these groups more difficult to locate accurately in the sequence. This idea is not supported by the results seen in *Table 8* when looking at model 2, where the descriptor accuracy higher then the group accuracy, something that would indicate that the bigger the group, the higher the accuracy. Unfortunately this is probable not the case, for one not all of the descriptors in every group are used in the cross validation because some of the descriptors have been added from another data set of descriptors but unfortunately during the thesis the sequences for these descriptors were not available.

3.2 Discussion – Dual Data Set Models

Looking at the results yielded from the three attempts made when combined the secondary structure information with amino acid information it becomes clear that combining the secondary structure information with amino acid information enhances the performance of hidden Markov Models. From *Table 8* it is also quite easy to see that Model 5 or Hybrid Hidden Markov Inspired Model performs worse then the other two secondary structure elements. The reason for this is most likely that it is more difficult to appropriately utilize the information of a second data set in the transition probabilities compared to the using them in the emission probabilities. The difference between the Combined Dual Hidden Markov Model (model 3) and the Dual Emitting Hidden Markov Model (model 4) is very interesting as it yields some information in regard to the nature of the descriptors themselves. When the data sets are emitted separately as in model 4, a data set capable of emitting fewer different observations will per definition weight more because these emissions will not be so thinned out across the different kinds of observations, while the emission of probabilities are evened out when emitted together like in model 3. The result seen in *Table 8* shows that model 4 performed better then model 3 showing that the HMMs performs better when the secondary structure “hit” weighs more then a amino acid hit. What this tells us of the descriptors is that the secondary structure is more conserved in the descriptor elements then the amino acids. Looking at the result from the dual set models it can be seen that the performance come at a cost. Looking at the result from Model 4 versus Model 2 a significant reduction in descriptors with result 0 % can be seen in *Table 8*. The reduction for zero descriptors is reduced from 24.4% to 16.3% of the total amount of descriptors from Model 2 to Model 4. What is important to notice though is that the number of groups where no descriptor is found is significantly higher in Model 4 (13) then in Model 2 (2). This could mean that most likely the zero descriptor amount is not only reduced in Model 4 versus Model 2 but also redistributed along the test set. In the result seen from Extended Amino Acid Model 24.4% of our descriptors are zero descriptors

while only two groups are totally “blacked out”. This implies that the zero descriptors are quite evenly distributed among the groups. The Dual Emitting Hidden Markov Model results contain fewer zero descriptors, about 8% less but these zero descriptors seem be concentrated when using these models leading to a significant increase in “blacked out” groups. This is most likely due to the fact the some would appear no to have conserved secondary structures. The reasons for this could many, one thing could be that there is not sufficient data to adequately represent the groups secondary structure. One other possible explanation could be that some of the groups, grouped together because of their structural similarities might be actually be several groups that look similar but are actually different groups undistinguishable due to the set threshold when grouped together making the group hard to locate. It could also be so that the some structures lack a strong sequence – structure bond, it might be that some groups are formed because other close by structures exist this puts a certain strain on the sequence that might be relieved in some “standardized” manner that is both proximity structure and sequence dependent.

3.3 Discussion – Different Target Data

Looking at the result from the different experiments trying to use different the first two attempts were quite the failures, most likely due to the fact that the threshold values for finding related structures were set way to lax. The threshold values should have been set a lot more strictly on both attempts. So whatever evolutionary information received from these alignments were most likely drowned out and distorted by the noise created by non related sequence information received from the alignments. In order to be able to use the known evolutionary knowledge of our structures PSSMs were created from our sequences by use of Psi-Blast set on three iterations and then remade into position specific probability matrixes that had been created by extracting amino acid information from the PSSM. These matrixes were then threaded through the hidden Markov Models (Model 4) together with the sequences secondary structure yielded the results seen in *Table 8*. The use of PSSM amino acid data combined with secondary structure information on Model 4 was the combination that yielded the best performance. It also surprisingly enhanced the polarizing effects seen when using secondary structures with fewer zero descriptors even more concentrated on specific groups. It could be that the combination of using secondary structures and amino acids simultaneously is a manner of creating a more evolutionary like signal and by using PSSM data we do not counteract that signal but actually purifying it further making it harder to find less evolutionary conserved groups to be found.

Chapter 4

Conclusion and Outlook

4.1 Related Research

The main objective in this thesis is to use hidden Markov Models to locate substructures in proteins. In this section of the thesis we look at research that could be considered to be in the proximity of the research done in this thesis. This research can broadly be separated into two sub groups, hidden Markov Models used for structure prediction (4.11) and Hidden Markov model concept improvements.

4.1.1 Hidden Markov Models used for structure predicting

Hidden Markov Models has been used both to predict and classify protein structure using structure representations but HMMs have also been used to create these kinds of structural representations. Protein substructures have been grouped to build libraries; these libraries have then been assigned an alphabet, one letter per library forming a local structure alphabet (LAS) [24]. This is a clever way to turn a three dimensional structure to a one dimensional string of letters representing amino acid fragments. These fragments have then been used to capture characteristics from protein folds by use of HMMs [24]. HMMs have also been used to create classifications from the structure directly by using hidden Markov models that uses discretized states four residues long composed of protein backbone conformations seen as series of overlapping fragments [23].

4.1.2 Hidden Markov Model concept improvements

The efficiency of HMMs has been proven more effective by fully utilizing the tools available in Bioinformatics. By using an alignment built from Homologous it is not only the sequence being threaded through the hidden Markov Model but also the sequences evolutionary information. Doing this makes sense as sequence features likely would be shared by closely related sequences. Threading an alignment built from known homologous of a target sequence using algorithms optimized for this improves the performance of a substantially HMM [25]. Most homologous are found using Blast or Psi-Blast and these use Profiles to search for closely related sequences, how closely related can be adjusted by changing the threshold. As profiles are used to find homologous they should contain information about a sequences evolutionary history as it as tool for finding them. Therefore threading a profile through a HMM should improve the performance of the model. By pairwise comparisons between profiles and hidden Markov Models have proved a powerful method for applications that rely on diverse multiple sequence alignments as targets like function and structure prediction [26]. HMMs using both amino acid information and secondary structure have been used for protein class prediction and fold recognition.

4.2 Conclusion

The results received from our Models are promising; the best performance was yielded when using the Dual Emitting Hidden Markov Model with secondary structures and PSSM information. Using this information we are able to accurately locate the correct position of the descriptors groups to a level 55.6 % and descriptors to a level of 58.5% per descriptor. Not bad considering that the sought after fragments only make up about ~11% of the sequences used as training data and this ~11% is

spread out into at least three separate parts dispersed in the sequence. By viewing the results it becomes clear that it is possible to build Hidden Markov Models that can locate local descriptors of protein structures. As can be seen from the above section, most of the improvements that has been devised in this thesis, for example using alignments and profiles instead of a target sequences has been done before and most likely with more sophisticated algorithms then those devised in this thesis. Utilizing secondary structure as well as amino acid information in HMMs is not something new either although no publications have been found that have used the information in the same manner as in this thesis but this could be a misconception. Using hidden Markov models to align local descriptors of protein structures to target sequences is something novel that has not been done before.

4.3 Outlook

Using Hidden Markov Models to find local descriptors through amino acid sequence information is likely going to be a ready to use application in not a too far off future. Still in order for this happen a few things have to be done and some things have to be looked over. There are some things that need to be done:

- The Hidden Markov Models must be cross validated against all descriptor groups, and not only the test- set that consists of a fraction of all of the groups. If secondary structure is to be used the cross validation must be done with predicted secondary structures for the amino acid to reflect the reality of how the models perform when used on a target sequence with unknown structure.

- Threshold values must be found for all groups to ensure that the models are capable of discriminating poor matches.

- The consequences for using allowing transitions back and forth between insert-states and delete-states when building the hidden Markov Models as these have a proven weakness by favoring delete states over match states in some cases. Something even more important when using models with two different types of delete states.

- To validate this approach to proteins it must be tested and used in a wide array of approaches, the first on most likely to try and predict long range interactions in proteins with unknown structure.

Further possible improvements to the application could be:

- Using PSSM for the match-state emission matrixes instead of using only the alignments for the segments.

- To find or create better algorithms to utilize the information received from PSSMs and the substitution Matrix BLOSUM62.

- To tailor-make a system that uses the model and information type that favors a group the most, for example why use secondary structure HMMs on groups that are “blacked out” when the can be found using amino acid models for example. Basically tailor making the search for each group according to the search parameters and models that favors that group the most.

Chapter 5

Acknowledgements

I wish to thank Torgeir Hvidsten (PhD) for giving me the opportunity to work with this interesting research. I would also like to thank my girlfriend Hanna Eriksson and my family (Anders, Ann-Christin and Jenny) for their constant support and patience. Daniel Larsson should also be mentioned as he helped me with trying to create sensible illustrations of the different types of models developed.

References

[1] Marti-Renom MA, Stuart AC, Fiser A, Sanchez R, Melo F, Sali A. 2000 *Comparative protein structure modeling of genes and genomes*. Annu Rev Biophys Biomol Struct, 29:291-325.

[2] JU. Bowie, R. Lüthy, D. Eisenberg 1991 *A method to identify protein sequences that fold into a known three-dimensional structure*. Science, 253:164-170

[3] Zhang Y, Skolnick J 2004 *Tertiary structure predictions on a comprehensive benchmark of medium to large size proteins*. Biophysical journal, 87:2647-2655.

[4] Baker D, Sali A 2001 *Protein structure prediction and structural genomics*. Science, 294(5540):93-96

[5] Sali A, Blundell TL 1993 *Comparative protein modelling by satisfaction of spatial restraints*. J Mol Biol, 234(3):779-815

[6] Floudas CA, Fung HK, McAllister SR, Monnigmann M, Rajgaria R 2006 *Advances in Protein Structure Prediction and De Novo Protein Design: A Review*. Chemical Engineering Science, 61:966-988

[7] Brenner, S.E., Koehl, P. and Levitt, M. 2000 *The astral compendium for sequence and structure analysis*. Nucleic Acids Research. Vol. 28, No. 1 , 254–256.

[8] Rigden DJ 2006 *Review Understanding the cell in terms of structure and function: insights from structural genomics.* Curr Opin Biotechnol. Oct:17(5):457-64

[9] Collins FS, Green ED, Guttmacher AE, Guyer MS 2003 *A vision for the future of genomics research* Nature 422, 835-847

[10] Bujnicki JM 2006 *Protein-Structure Prediction by recombination of Fragments* ChemBioChem 7 19-27

[11] Hvidsten TR, Kryshtafovych A, Fidelis K. 2008 *Local Descriptors of protein Structure: A systematical analysis of the sequence-structure relationship in proteins using short- and long-range interactions.* Submitted

[12] Viterbi AJ 1967, *Error bounds for convolutional codes and an asymptotically optimal decoding algorithm* IEEE Trans. Informat. Theory IT-13 260-269

[13] Forney GD 1973, *The Viterbi Algorithm* Proc. IEEE 61 268-278

[14] Smith K 2002 *Hidden Markov Models in Bioinformatics with Application to Gene Finding in Human DNA*, Machine Learning Project Proposal. 308-761

[15] Eddy SR 1998 *Profile hidden Markov models* Bioinformatics Review Vol. 14, 755-763.

[16] Sigrist CJ, Cerutti L, Hulo N, Gattiker A, Falquet L, Pagni M, Bairoch A, Bucher P 2002 *PROSITE: a documented database using patterns and profiles as motif descriptors.* Brief Bioinform 3: 265-74.

[17] Rabiner LR. 1989 *A tutorial on hidden Markov models and selected applications in speech recognition* Proc. IEEE, Vol. 77, pp. 257–286.

[18] Wang L, Jiang T. 1994 *On the complexity of multiple sequence alignment* J Comput Biol 1:337-348

[19] Gavin C. Cawley, Nicola L.C. Talbot 2003 *Efficient leave-one-out cross-validation of kernel Fisher discriminant classiers*. Pattern Recognition 36 (2003) 2585 – 2592

[20] Tramantano A. 2003 *Of men and machines* Nat Struct Biol 10 87-90

[21] Altschul SF ,Koonin, EV 1998 *Iterated profile searches with PSI-BLAST - a tool for discovery in protein databases* TIBS 23(11):444-7

[22] Altschul SF, Madden TL, Schaffer AA., Zhang J, Anang Z, Miller W, Lipman DJ 1997 *Gapped BLAST and PSI-BLST: a new generation of protein database search programs* Nucl. Acids Res. 25:3389-3402

[23] Camproux AC, Gauter R, Tuffery P 2004 *A hidden Markov Model derived structural alphabet for proteins.* J .Mol .Biol 338:611-629

[24] Wang SL, Chen CM, Hwang MJ Classification of protein 3D folds by Hidden Markov Learning on sequences of structural alphabets

[25] Käll L, Krogh A, Sonnhammar E 2005 *An HMM posterior decoder for sequence feature prediction that includes homology information* Bioinformatics 21-i251-i257

[26] Söding J, Remmert M, Biegert A, Lupas AN 2006 *HHSenser: exhaustive transitive profile search using HMM-HMM comparison* Nucl. Acids Res.34 W374-W378

[27] Henikoff S, Henikoff JG1992 *Amino acid substitution matrices from protein blocks* Proc. Natl. Acad. Sci. USA 89 10915-10919

Appendix A

	A	R	N	D	C	Q	E	G	H	1	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
1	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4