MARSHALL PLAN SCHOLARSHIP PROGRAM REPORT

Expert Tuned Profile Hidden Markov Models for Primary and Secondary Structure Based Homology Prediction in Bioinformatics

prepared for the
Austrian Marshall Plan Fundation

submitted by

Christian Winkler, BSc

Salzburg University
of Applied Sciences

Salzburg, October 2017

Contents

List of Abbreviations

HMM Hidden Markov Model

pHMM profile Hidden Markov Model

MSA Multiple Sequence Alignment

SCOP Structural Classification of Proteins

SCOPe Structural Classification of Proteins-extended

PDB Protein Data Bank

NMR Nuclear Magnetic Resonance

GUI Graphical User Interface

PHD Profile Network from HeiDelberg

DSSP Dictionary of Secondary Structures in Proteins

AUC Area under the curve

ROC Receiver Operating Characteristic

FPR True Positive Rate

SVM Support Vector Machine

TPR False Positive Rate

List of Figures

List of Tables

1 Introduction

Proteins are the essential building blocks of all forms of life. They can be found in all living organism, and play a key role in many different functions of life. Understanding the function of proteins is a crucial task in bioinformatics. One approach to gain a better understanding of proteins is homology prediction, where sequence similarities of known proteins might give information about common ancestors they diverged from.

This report investigates the use of profile Hidden Markov Model (pHMM) in protein family classification. In detail, the common approach using only the primary structure to build a pHMM and score sequences against it, will be extended to make use of secondary structure information. In particular cases in which protein homology is not well represented by sequence similarity in certain parts of the protein, stability with respect to secondary structure is expected to increase the classification quality.

The following Chapter 2 gives an overview about relevant areas of bioinformatics, needed for this paper. These include topics, such as protein composition, secondary structure determination, and pHMM. In Chapter 3, the implementation of various methods in the software package *HMModeler* is described. Based on the theoretical background, the steps from a Multiple Sequence Alignment (MSA), determining secondary structure, building a pHMM and score sequences against it, are covered. Chapter 4 evaluates the implementation based on various datasets. Finally, Chapter 5 concludes this paper.

2 Selected Background Information

Bioinformatics covers a wide range of scientific fields, especially biology, chemistry and computer science. This chapter gives a short introduction in certain areas of bioinformatics, needed as basic understanding for the methods used in later chapters.

2.1 Proteins

Proteins play a key role in almost all biological activity. Proteins are large biological polymers composed of one or more chains of amino acids held together by spatial bonds called peptide bond. The material described in this section can be found in greater detail in [1, 4].

2.1.1 Amino Acids

Amino acids are the basic building blocks of proteins. Each amino acid consists of a central α carbon bound to a carboxyl group ($COOH$), an amino group (NH_2), a hydrogen atom, and a distinct side chain (or R group). Variation in the chain defines the 20 common amino acids found in protein molecules. For instance, if the side chain contains just one hydrogen atom, the amino acid is *Glycine*, while the side chain CO_2OH forms the amino acid *Serine*. Table 2.1 list the 20 common amino acids found in proteins with their assigned three-letter abbreviations and one-letter symbols, as defined in [5].

Amino acid	Abbreviation	Symbol		Amino acid	Abbreviation	Symbol
Alanine	Ala	A		Leucine	Leu	L
Arginine	Arg	R		Lysine	Lys	K
Asparagine	Asn	N		Methionine	Met	M
Aspartic acid	Asp	D		Phenylalanine	Phe	F
Cysteine	Cys	C		Proline	Pro	P
Glutamic acid	Glu	E		Serine	Ser	S
Glutamine	Gln	Q		Threonine	Thr	T
Glycine	Gly	G		Tryptophan	Trp	W
Histidine	His	H		Tyrosine	Tyr	Y
Isoleucine	Ile	I		Valine	Val	V

Table 2.1: The 20 common amino acids of proteins with their three-letter abbreviation and one letter symbols [5].

Two amino acids can form a dipeptide through a covalent chemical bond through cleavage of a water molecule (H_2O) from the carboxyl group of one amino acid and

the amino group of another. The resulting $CO - NH$ bond is called *peptide bond* (see Figure 2.1).

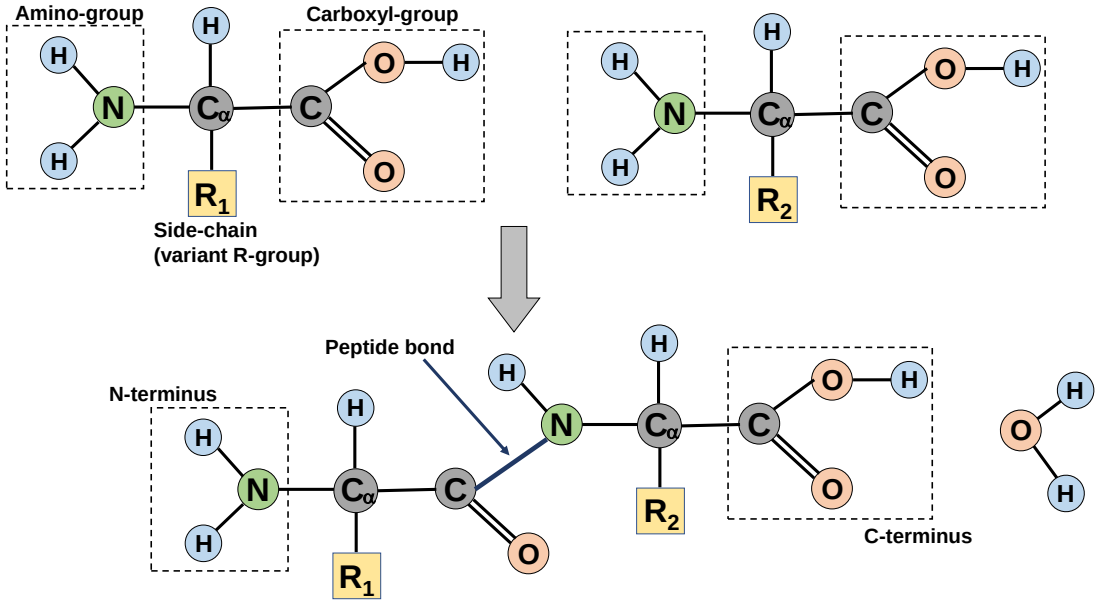

Figure 2.1: Dipeptide forming from the condensation of two amino acids.

These bonds can extend into peptide chains with a size up to several thousand amino acid residues. Sequences with up to 50 residues are generally referred to as peptides, for longer sequences, the term polypeptide or protein is used. By convention, the reading direction of a polypeptide chain is defined from the N- to C-terminus, where N-terminus refers to the free amino group at one end, and C-terminus to the free carboxyl group at the other end of the chain.

2.1.2 Protein Structure

The shape of a protein is critical to its function. As proteins typically has a stable three-dimensional structure, dictated by its amino acid sequence. Protein structure is divided into four hierarchical levels: primary, secondary, tertiary, and quaternary.

The **primary structure** is represented by the linear sequence of amino acids within a polypeptide connected by peptide bonds, where each element corresponds to one of the 20 amino acids described in section 2.1.1. The primary structure of a polypeptide can be determined from protein sequencing methods, such as spectrometry.

The **secondary structure** is defined by the local folding patterns of a region of the protein over a short range of amino acids in a polypeptide. The secondary structure

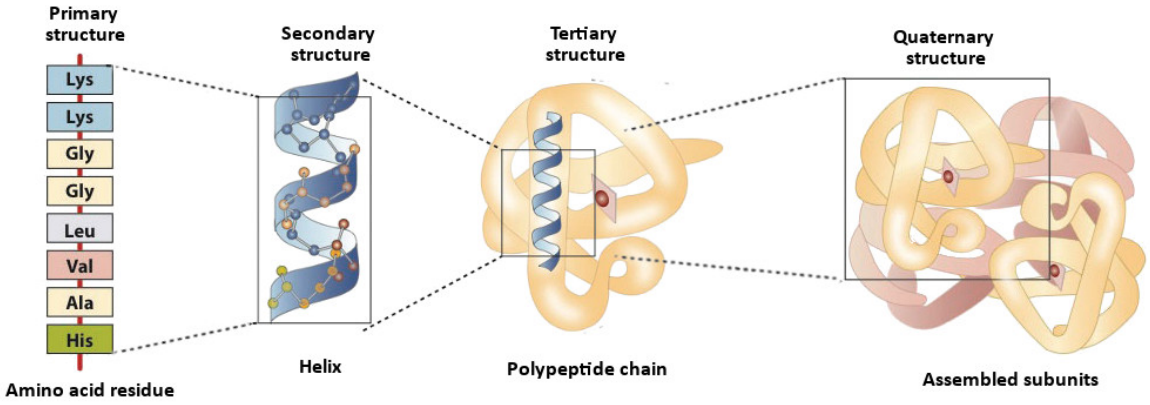

Figure 2.2: The four levels of structure in Proteins [1]

is mainly dependent by its primary structure, which is stabilized by hydrogen bonding due to interaction between the atoms of the backbone. The two most common spatial formations are the α-helix and the β-sheet, connected by irregular segments, referred to as loops or coils.

The α-helix folds by twisting the polypeptide chain into a right handed coiled structure, with the side chains positioned on the outside of the helix where they are free to interact with the surrounding.

The β-sheet is composed of two or more stretched polypeptide segments, also called β-strands, next to each other and held together by hydrogen bonds. The side chains in β-sheets alternate above and below the plane of the strands. Due to the direction of polypeptides, β-sheets can be differentiated into antiparallel β-sheets when the strands run in the opposite direction and parallel β-sheets when they run in the same direction [4].

If the tertiary structure is available, the secondary structure can (almost trivially) be computed from the former. Otherwise, it has to be predicted from the primary structure. Methods for secondary structure determination are explained in section 2.4.

The **tertiary structure** is defined by the coordinates of all atoms in the protein relative to each other and represents the complete three-dimensional structure of the entire folded polypeptide chain. The tertiary structure of a protein is determined by experimental methods such as X-ray crystallography and Nuclear Magnetic Resonance (NMR).

The **quaternary structure** represents complex protein structures made up of multiple polypeptide chains.

2.1.3 Protein Homology

Two proteins are homologous when they share common evolutionary ancestors. Often homologous proteins with similar biological functions have a similar sequence and structure, at least in some crucial regions. Differences occur due to mutations in the sequence, such as substitutions, insertions, and deletions of single amino acids. The degree of homology is determined by metrics, such as the similarity of two sequences. High sequence similarity between two sequences is an indication of them having the same ancestor, whereas the probability of them originating independently of each other increases with less similarity. As the structural fold of a protein is crucial to its function, regions that are critical to its function are better conserved than irregular loops. A way to model and describe protein homology is to use MSAs, see Section 2.3.1.

2.2 Protein Databases

Over time, more than 1,600 databases containing bioinformatic data have been created [6]. These databases can be categorized into primary and secondary databases. Primary databases are filled with sequence or structure data derived from experimental results by researchers, whereas secondary databases are composed of data derived from primary databases and organized with additional knowledge, such as family classification.

The following sections detail the protein structure database the Protein Data Bank (PDB), and the secondary database the Structural Classification of Proteins (SCOP), as they are used in the current study.

2.2.1 Protein Data Bank

In 1971, the Brookhaven National Laboratory established the PDB as primary database for the three-dimensional structures of proteins, nucleic acids, and complex structures. Since 2003, the PDB is managed by multiple organizations around the world by the worldwide PDB (wwPDB) organization[1].

The data in the PDB is mostly derived by methods like X-ray crystallography and Nuclear Magnetic Resonance spectroscopy. Protein structures are stored in the PDB-file format as three-dimensional position of each atom together with additional data, like temperature factor and amino acid residues, among others. Each entry published in the PDB has a unique, four-character identifier (PDB ID) [7].

[1]Worldwide PDB: `https://www.wwpdb.org/`

Currently, the PDB contain over 130,000 structures[2] and grows by ~10% in size annually [8]. However, compared to more than 90 million sequences in the TrEMBL Protein Database[3], three-dimensional structures are available for only a small number of proteins.

2.2.2 Structural Classification of Proteins

The SCOP classifies proteins with known structures from the PDB according to their evolutionary, functional, and structural relationships. Each protein in the SCOP is classified in a hierarchical system with the four main levels: *family*, *superfamily*, *fold*, and *class* [9].

- **Family** describes proteins with obvious evolutionary relationships due to high sequence similarity over the protein, or with low sequence similarity but high structural and functional similarity.
- **Superfamily** describes families with low sequence similarity, where, based on structural or functional features, a common evolutionary origin is probable.
- **Fold** describes superfamilies with high structural similarities, based on a similar arrangement of secondary structures and their topological connections.
- **Class** describes folds according to the appearance of their secondary structure, such as proteins with α-helices only.

The convention for describing protein classification is `Class.Fold.Superfamily.Family`. In this convention, the class is described as an alphanumerical letter, while the fold, superfamily, and family are described using numbers. For example, hemoglobin with the scop identifier (sid) *D1A3NA_* is classified as a.1.1.2, belonging to the family *globins* (2), the superfamily and fold *globin-like* (1), and under the class α-helices only (a).

Until version 1.73 of the SCOP, all protein structures were manually classified. With an increasing number of protein structure publications and the subsequent growth of the PDB, manual classification became too slow to classify all new proteins. Therefore, in later versions and with the introduction of Structural Classification of Proteins-extended (SCOPe), automated processes were introduced [10].

[2]PDB - Yearly Growth of Total Structures: `https://www.rcsb.org/pdb/statistics/contentGrowthChart.do?content=total`

[3]TrEMBL Statistics: `https://www.ebi.ac.uk/uniprot/TrEMBLstats`

2.3 Profile Hidden Markov Model

This section will focus on the main parts of pHMM. More background, especially regarding the fundamentals of Hidden Markov Models (HMMs) such as Markov chains can be found in [2].

2.3.1 Multiple Sequence Alignment (MSA)

Two sequences can be compared by a pairwise sequence alignment, where the residues of both sequences are directly compared to each other, allowing to use gap positions between the residues. Two protein-sequences are very similar when they have many match-states and only a few gaps. Sequences with high similarity are also very likely to be homologous and display a structural and evolutionary conservation.

A multiple sequence alignment compares three or more sequences to each other. Figure 2.3 shows a MSA built up from 15 sequences from the same superfamily, see Section 2.2.2.

```
# STOCKHOLM 1.0

d2dlqa4  ..........VECPT...CHKKFLSKYYLKVHNRKHTGEK...........
d2dlqa3  ......SEQVFTCSV...CQETFRRRMELRLHMVSHTGE............
d2dlqa2  .........PFECPK...CGKCYFRKENLLEHEARNCMNR...........
d1wjva1  GSSGSSGMVFFTCNA...CGES.VKKIQVEKHVS.NCRNC...........
d1x5wa1  ........HPEKCSE...CSYSCSSKAALRIHERIHCTD............
d1bboa2  .........PYHCTY...CNFSFKTKGNLTKHMKSKAHSKK..........
d2vy4a1  .......DEVVICPY...DSNHHMPKSSLAKHMASCRLRKMGYTK......
d1m36a_  ...GSRLPKLYLCEF...CLKYMKSRTILQQHMKKCGWF............
d2drpa2  .........VYPCPF...CFKEFTRKDNMTAHVKIIHK............
d1llmc2  .......EKPFACDI...CGRKFARSDERKRHRDIQHI............
d1tf3a3  ........KNFTCDSDG.CDLRFTTKANMKKHFNRFHNIK...........
d2ctda2  ........EMFTCHH...CGKQLRSLAGMKYHVMANHNSLP..........
d2glia1  .........ETDCRWDG.CSQEFDSQEQLVHHINSEHIHGER.........
d1wjpa3  .......YKKLTCLE...CMRTFKSSFSIWRHQVEVHNQNNMAPTSGPSSG
d2j7ja3  .........GYPCKKDDSCSFVGKTWTLYLKHVAECH.............
//
```

Figure 2.3: 15 sequences from the SCOP superfamily g.37.1 aligned to an MSA

Functionally important residues in a sequence are assumed to remain stable over generations, and are less likely to mutate. Therefore, highly conserved regions, with high similarities over the columns of the MSA, are indicated to be functionally important.

There are different approaches for generating an MSA, based on manual annotation by expert knowledge, or automatic methods based on structural information, see [2, Chapter 6].

2.3.2 Profile Hidden Markov Model

A pHMM is a stochastic model, based on Markov chains, especially used in bioinformatics to model a MSA and capture its degree of structural conservation. Figure 2.4 shows the structure of a pHMM, which uses three different types of hidden states for each column of the MSA. These are match state M_k, delete state D_k and insert state I_k.

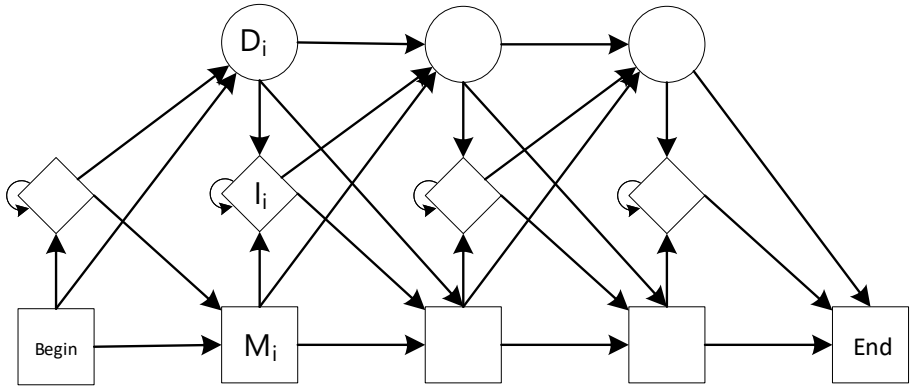

Figure 2.4: Structure of a pHMM with match, insert and delete states, adapted from [2].

Match state M_k is described by a distribution, trained with the symbol frequency in column k of the MSA. Each representative column in the MSA model matches one state. Gaps in the MSA influence the insertion and deletion states.

Insertion states model genetic insertions of additional symbols in a sequence that do not show in the majority of the sequences of a family of proteins in the MSA. They occur when there are many gaps in a column, typically above 50%. They allow for inserting the symbols in the MSA for those sequences that do not show gaps. Insertion states have a self-transition to cover repeated insertions over multiple columns. Instead of the actual symbol frequency of the MSA column, the emission probabilities can be set to a background probability, based on a general frequency of the symbols, as a few residues in the column might not be a representative.

Deletions allow a sequence to skip over a match state without emitting a symbol, and model situations of genetic deletions where a certain sequence has one or more positions fewer than the other proteins in a family modeled by the MSA. Deletions are silent states without emitting symbol probabilities. Self-transitions are not allowed for deletions, however gaps over multiple columns are modeled as a sequence of deletion-states, allowing different transition probabilities over multiple gap-positions.

2.3.3 Decoding and Scoring

In a fully defined pHMM any sequence can be represented by a matching score, by multiplying the transition and emission probabilities along the path. Due to the insert and delete states, many different paths can form the same sequence. To find the best suited path for the specific sequence, dynamic programming approaches are used, such as the *forward-algorithm* or the *viterbi algorithm*.

The *forward algorithm* calculates the overall probability as sum for each individual state of the pHMM. The *viterbi algorithm* calculates the probability for the most likely sequence of states by using the maximized path over the pHMM.

To prevent an underflow, caused by multiplication of many probabilities in the range of 0-1, a logarithmic scale is used to calculate the raw-score. However, those raw-scores are heavily dependent on the overall sequence length, as longer sequences generate smaller scores, and are thus difficult to compare. Therefore, the scores can be normalized using correction methods such as null models, either the simple null model or the reversed sequence null model. The simple null model uses a simplified one-state HMM with a specific residue composition, based on general occurrence frequencies of the amino acids, and scores the sequence against it. The reversed sequence null model uses the original pHMM and scoring algorithm, but inverses the direction of the used sequence. The final score without length dependency is calculated by subtracting the null model score from the raw score in the logarithmic scale.

2.4 Secondary Structure Estimation

The fundamentals of protein prediction, relevant for this work, are discussed by Mayer in [11]. First of all, secondary structure determination method Dictionary of Secondary Structures in Proteins (DSSP), then the two secondary structure prediction approaches Profile Network from HeiDelberg (PHD) and MetaSSPred will be discussed.

2.4.1 The Dictionary of Secondary Structures in Proteins

The DSSP is a method by Kabsch and Sander, described in [3], to determine the secondary structure of a protein based on the atomic coordinates of proteins. It uses pattern recognition in hydrogen bonding and specific geometric features to assign one of eight secondary structures (see table 2.2) to each residue in a protein. These eight types can be grouped into three major components helix: (H, G, I), strand (B, E) and loop (T, S, C).

Symbol	DSSP	secondary structure
α(H)	H	α-helix
	G	3_{10} helix
	I	π-helix
β (E)	B	residue in isolated β bridge
	E	extended strand
L(C)	T	hydrogen bounded turn
	S	bend
	C	loop or irregular

Table 2.2: Structure types determined by DSSP [3]

Helices are recognized as repetitive turn-sequences of hydrogen bonds. The α-helix is the most common helix-type with hydrogen bonds four residues apart. The 3_{10}-helix and π-helix are variations with hydrogen bond distances of three and five residues, respectively. The two types of beta strands are stretched polypeptide chains arranged for repeating hydrogen bonds with other strands. Short turns supported by hydrogen bounds reverses the overall direction of the polypeptide chain. Bends are regions, with an angle at least 70° from one residue to the residue after next. All other non-identifiable residues are marked as loop or irregular, whereas in generated DSSP-files *C* is a blank position.

2.4.2 PHD

The secondary structure prediction method PHD, introduced in [12] by Rost and Sander, is the first method which predicted secondary structure with an accuracy greater than 70% from the primary structure only.

The **P**rofile Network from **H**ei**D**elberg uses three steps to predict the secondary structure. In the first step, PHD examines evolutionary information by obtaining an MSA from homologous sequences. The probability for each amino acid in the MSA is fed into a two-level feedforward neural network system, previously trained through back-propagation with proteins of known structures. In the first level, for each position in the sequence, a sliding window of 13 consecutive amino acids is fed, and the likelihood of the central residue being a helix, sheet, or loop is returned. The independently trained second level receives the output from the first level and return new probabilities for the same three major secondary structure elements. The highest value determines the secondary structure. A reliability index is set with the difference of the two highest values. The final step is a filter to remove obvious errors, like helices shorter than 3 residues long.

Secondary structure prediction with PHD is available online with the PredictProtein web server[4] or locally through the Debian Linux package PROFPhd. The expected accuracy for the three major secondary structure types is, on average, 76% [13].

2.4.3 MetaSSPred

MetaSSPred is a secondary structure prediction method, developed in the Bioinformatics & Machine Learning Lab[5] at the University of New Orleans, see [14]. MetaSSPred is available as Linux based standalone software package[6].

MetaSSPred combines three binary Support Vector Machines (SVMs), one for each secondary structure element. The SVMs make use of 33 features, extracted from the primary structure. Each SVM predict for one secondary structure element, if a residue belongs or not belongs to the secondary structure, and returns a probability for it. The secondary structure class with the highest probability is assigned to the amino acid. In an additional step, the output from the SVMs is combined with the output of the secondary structure predictor *SPINE X* [15].

[4] https://www.predictprotein.org/
[5] http://biomall.cs.uno.edu/
[6] http://biomall.cs.uno.edu/software/

3 Implementation

This chapter covers the implementation based on HMModelers workflow. First, processing input data, including secondary structure determination, is described. This is followed by a description of adapting the pHMM, and finally of the scoring-algorithms. However, this section focuses on the modifications made in HMModeler, more information on HMModeler with its not mentioned implementations can be found in [16, 17, 11].

3.1 HMModeler

HMModeler is a software package for protein classification jointly developed by researchers at Salzburg University of Applied Sciences and the University of Salzburg. While HMModeler was designed as extension for UCSF Chimera[1], the current version runs independently with its own Graphical User Interface (GUI). The core of HMModeler is written in Python while time-consuming algorithms are also implemented with faster C++ libraries. The web based GUI communicates with the core system vie RESTful web services. HMModeler is platform independent and runs both on Windows and Linux operating systems. HMModeler uses the Smith-Watermann style variant of pHMM, shown in figure 3.1. This variant is used for local alignments, by using flanking states with transitions from the begin to each match state as start model, and from each match state to the end state as end-model.

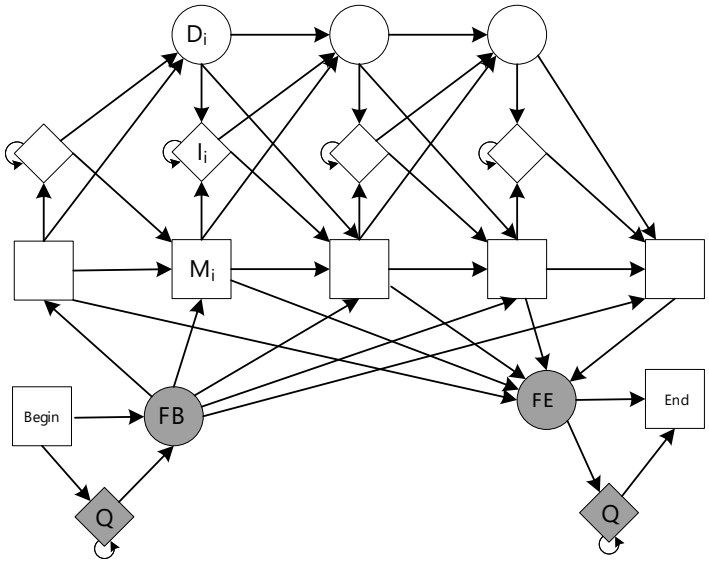

Figure 3.1: Smith-Waterman variant of a pHMM, adapted from [2].

[1]UCSF Chimera `https://www.cgl.ucsf.edu/chimera/`

The GUI allows skilled experts to introduce prior information about the protein family targeted at into the HMM. In particular, the user can interactively define parts of the protein with increased or decreased insertion and deletion probabilities. The user can also modify the extent to which the emission probabilities in the single model columns of the pHMM are extracted purely from given multiple sequence alignments, or are, alternatively, determined from a-priori distributions. Finally, the user can define so-called expert sets that override other estimation methods and set the possible emissions in certain model states to an explicit set of amino acids. Currently, the software uses only the primary structure for building the pHMM, and scores sequences against it with the viterbi and forward algorithm, see [11, 16, 17].

3.2 Input Data

First, HMModeler takes an MSA as input to train its pHMM. HMModeler can process files in the *Stockholm* file format, uploaded by the user or referenced by a project-ID from the Multiple Structure Alignment Server Pirates[2].

The Stockholm format is a markup-format for MSA, where each sequence can be annotated with additional features, such as the corresponding secondary structure for each amino acid residue. The definition of the Stockholm format can be found at [18].

Figure 3.2 shows the implemented workflow for determining the secondary structure. All sequences in the MSA are read. If there is no structure information provided for one or more sequences, the secondary structure is fetched from the PDB Database or predicted using the MetaSSPred method, as described in section 2.4.3.

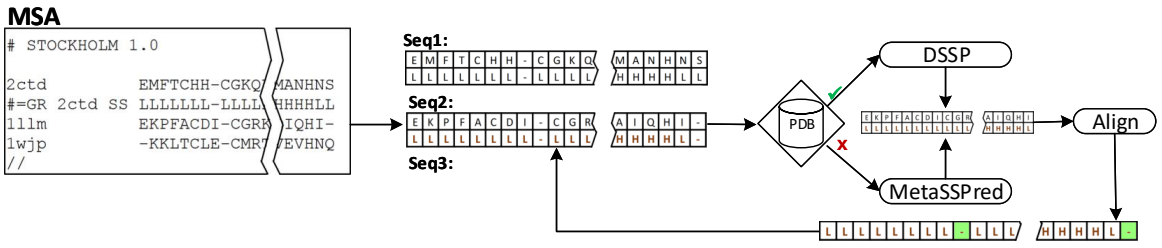

Figure 3.2: Workflow for processing input data and determination of secondary structure.

The secondary structure from sequences corresponding to 3D-data in PDB are calculated using the DSSP-program (see section 2.4.1). The extracted structural information must be aligned with the original sequence, considering the exact gap positions and possible differences in the overall sequence length. The Needleman-Wunsch algorithm,

[2]Multiple Structure Alignment Server Pirates: `https://biwww.che.sbg.ac.at/pirates`

described in [2, p. 19–21], is used for this task. Needleman-Wunsch is a dynamic programming approach for global sequence alignment that calculates the optimal path through a scoring matrix using match, mismatch, and gap penalties. The derived sequences, linked to their corresponding secondary structure, are aligned with the original sequence. Provided the original sequence from the MSA matches the aligned sequence from PDB, the structure is used.

If there is no match in PDB or the alignment fails, the secondary structure is predicted with *MetaSSPred*. For each sequence that has to be predicted, a Fasta-formatted file with gaps removed in the sequence is generated in the input directory of *MetaSSPred* and processed by the software. Finally, the predicted secondary structure is aligned to the sequence with the Needleman-Wunsch algorithm.

3.3 Training the HMM

Based on the MSA, which now includes both primary and secondary structures, the transitions and emission distribution are assigned to the pHMM. These are calculated by the frequencies across sequences for each column in the MSA, according to (3.1), where a_{kl} is the transition probability from state k to l, and (3.2) for the emission probability $e_k(a)$ of the symbol a at state k.

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \tag{3.1}$$

$$e_k(a) = \frac{E_k(a)}{\sum_{a'} E_k(a')} \tag{3.2}$$

To prevent overfitting by symbols with a probability of zero, it is important to add some background frequency like a small non-zero prior probability $pc(a)$ to each symbol, see (3.3). The simplest approach is the Laplace smoother, by adding a hypothetical observation in form of one pseudocount to each emission and transition.

$$e_k(a) = \frac{E_k(a) + pc(a)}{\sum_{a'} E_k(a') + \sum pc(a')} \tag{3.3}$$

The implementation for the emission probabilities were extended by calculating the primary structure with 20+1 symbols to generate two additional matrices, one with 3+1 symbols for the secondary structure and another with 84 symbols containing mixed

probabilities from the other two sets. The additional symbol for the primary and secondary structures were added to cover a wider set of sequences in the test-database (see Section 4.1), where single residues or structure elements in sequences can be unknown and therefore marked with the letter 'X'. As the symbol 'X' stands for all other symbols in the set, the probability will be set to 1, which equals to the sum of all other probabilities.

The transition probabilities only rely on the gap positions of the MSA, therefore no modifications in the current implementation of HMModeler are needed. Moreover, the functional emission probability implementation for the primary structure remains the same.

The emission probabilities for the secondary structure are calculated by using the three common types helix, strand and loop. To prevent zero probabilities, a prior-probability can be set by the user over HMModelers GUI.

Using both probability sets, the original primary structure implementation and the new secondary structure probabilities, a weighted emission frequency set that covers both primary and secondary probabilities is generated.

Mayer discussed the emission probability weighting for the use with pHMM in [11]. Following his research three different implementations are explained next.

3.3.1 Linear Weighting

Mayer discussed in [11] the problem with the different length of the two alphabets, as the primary structures involve 20 amino-acids symbols, whereas the secondary structure only involves three symbols. Therefore he recommended the approach in (3.4), where the mixed emission probabilities $e_k(p, s)$, are weighted by their symbol-length with 20 amino acids for the primary probability $e_k(p)$, and the three structure symbols in $e_k(s)$.

$$e_k(p, s) = \frac{20}{20 + 3} \cdot e_k(p) + \frac{3}{20 + 3} \cdot e_k(s) \tag{3.4}$$

However, the most convenient approach for mixing two probabilities by a simple multiplication as in (3.5) will also be implemented and tested.

$$e_k(p, s) = e_k(p) \cdot e_k(s) \cdot k \tag{3.5}$$

As mentioned in Section 3.4, where a threshold can be used to define if the scoring process uses the mixed or the primary probability, the additional factor k is introduced for scaling the resulting mixed probability. With this factor, the impact between a column in the pHMM where the mixed probability is used, and columns that prioritize the primary probability can be reduced. For example, if k is set to three, the size of the secondary structure alphabet, and the probability $e_k(s) = \frac{1}{3}$, the secondary structure would have no impact on the resulting probability.

3.3.2 Weighting by Shannon

In [11], Mayer introduces weighting the two emission-probabilities by the amount of information in each column using the Shannon theorem, defined in [19] as

$$H = -\sum_{i=1}^{N} p_i log_b p_i \tag{3.6}$$

where the entropy H is the negative sum over the probability of each possible outcome p_i multiplied by the logarithm of the same probability. If one single probabilistic result of an event has the probability of 1, the entropy is 0. In this case, there is no uncertainty. Conversely, if all probabilistic results have equal probability, the uncertainty is maximal with an entropy of $log_b(N)$, where N is the number of possible outcomes. The logarithmic base b is usually 2 as the common use is digital communication, or, in other cases, the natural logarithm e.

Figure 3.3 show the entropy as a function for a binary event with probabilities p and $q = 1 - p$ with a natural logarithm and logarithmic base of 2. The maximum entropy occurs when $p = q = 0.5$.

The lower the entropy, the higher the degree of information and thus the relevance for the mixed probability. The rate of information I will be determined by subtracting the entropy H from the maximal possible entropy $\log_b(N)$:

$$I = \log_b(N) - H \tag{3.7}$$

With the degree of information for both, the primary structure $I(p)$ and secondary structure $I(s)$ the probabilities will be weighted as follows:

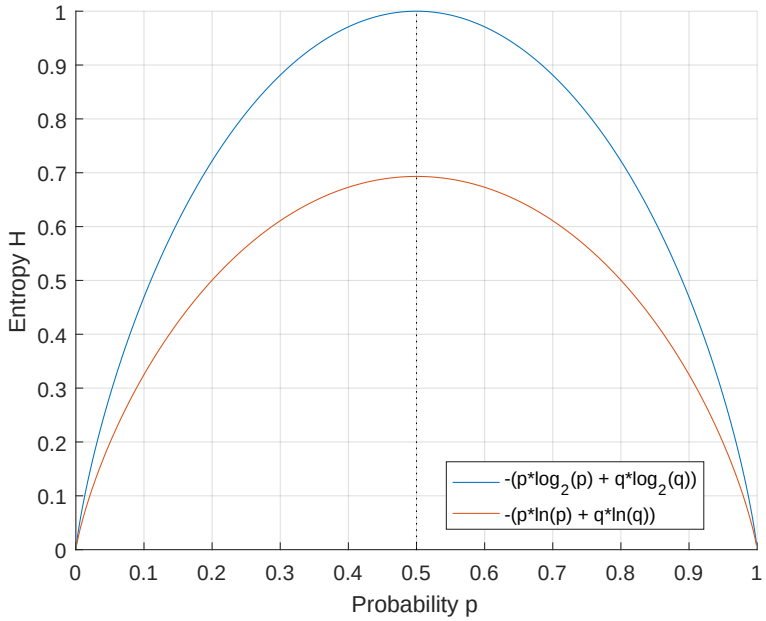

Figure 3.3: Entropy H for an event with two possible outcomes, with natural logarithm (red) and normalized logarithm with base 2 (blue) [19]

$$e_k(p,s) = \frac{I(p) \cdot e_k(p) + I(s) \cdot e_k(s)}{I(p) + I(s)} \tag{3.8}$$

In contrast to the natural logarithm for both the primary and secondary structure, as recommended by Mayer, the normalized logarithm with a base equal to the corresponding alphabet size will be used. Without the normalization both probabilities would not be equally weighted, as the maximum entropy for the primary structure would be $ln(20) = 2.9957$ and for the secondary structure $ln(3) = 1.0986$.

For programming languages, which does not support non standard logarithmic bases, any logarithmic equation with base a can be converted to any other base b, by dividing through the logarithm with the same base a, with the argument set to the new base b:

$$log_b(x) = \frac{log_a(x)}{log_a(b)} \tag{3.9}$$

3.4 Scoring Sequences

HMModeler calculates nine different types of scores, listed in Table 3.1, using the viterbi and forward algorithm, and variations using the simple null model and the reversed

sequence null model, see Section 2.3.3.

Score types	Description
Viterbi score	$V(s)$
Forward score	$F(s)$
Simple null model	$N(s)$
Reverse viterbi null model	$V(s^{-1})$
Reverse forward null model	$F(s^{-1})$
Simple corrected viterbi score	$V(s) - N(s)$
Simple corrected forward score	$F(s) - N(s)$
Reverse corrected viterbi score	$V(s) - V(s^{-1})$
Reverse corrected forward score	$F(s) - F(s^{-1})$

Table 3.1: Scores calculated by HMModeler

Only the algorithms for the viterbi score and the forward score must be adapted for the use of the secondary structure. The reversed null models use the same algorithm but with the sequence reversed, and the simple null model uses a one state HMM with general background distribution based on the primary structure that will continue to be used.

As described in Section 3.3, the mixed emission probabilities $e_k(p, s)$ are generated based on the primary structure probabilities $e_k(p)$ and secondary structure probabilities $e_k(s)$. However, with the aim of using only the primary structure instead of the mixed probabilities for regions in the MSA where the primary structure is highly conserved, a threshold can be set. If the highest emission probability in the primary structure is set below the threshold, the mixed probabilities are used, otherwise only the probabilities for the primary structure are used:

$$e_k(m) = \begin{cases} e_k(p) & if\ max(e_k(p')) > threshold \\ e_k(p, s) & \text{otherwise} \end{cases} \tag{3.10}$$

The resulting probability $e_k(m)$ will be used in the viterbi equation for the transition to the match-state V_j^M:

$$V_k^M(k) = \log e_k(m) + max \begin{cases} V_{k-1}^M(i-1) + \log a_{M_{k-1}M_k} \\ V_{k-1}^I(i-1) + \log a_{I_{k-1}M_k} \\ V_{k-1}^D(i-1) + \log a_{D_{k-1}M_k} \\ V_{k-1}^{FB}(i-1) + \log a_{FBM_k} \end{cases} \tag{3.11}$$

The other viterbi equations remain unchanged; as for insertion states in (3.12), a background probability $q(p)$ based on the primary structure alphabet will be emitted, and the silent deletion states in (3.13) do not emit any symbol.

$$V_k^I(k) = \log q(p) + max \begin{cases} V_k^M(i-1) + \log a_{M_k D_k} \\ V_k^I(i-1) + \log a_{I_k D_k} \\ V_k^D(i-1) + \log a_{D_k D_k} \end{cases} \tag{3.12}$$

$$V_k^D(k) = max \begin{cases} V_{k-1}^M(i-1) + \log a_{M_{k-1} D_k} \\ V_{k-1}^I(i-1) + \log a_{I_{k-1} D_k} \\ V_{k-1}^D(i-1) + \log a_{D_{k-1} D_k} \end{cases} \tag{3.13}$$

The same modification applies to the match state in the forward algorithm in (3.14), which is similar to the viterbi algorithm, except instead of using the max transition state, it sums up all transition states.

$$\begin{aligned} F_k^M(i) = \log e_k(m) + log[& a_{M_{k-1} M_k} \ \exp(F_{k-1}^M(i-1)) \\ & + a_{I_{k-1} M_k} \ \exp(F_{k-1}^I(i-1)) \\ & + a_{D_{k-1} M_k} \ \exp(F_{k-1}^D(i-1)) \\ & + a_{FBM_k} \ \exp(F_{k-1}^{FB}(i-1))] \end{aligned} \tag{3.14}$$

4 Tests and Result

In the previous chapter, the implementation in HMModeler was described. In this chapter these changes will be discussed by the testing the different methods that were used.

4.1 Dataset

The ASTRAL SCOP 1.73 sequence database[1] filtered to entries with less than 40% sequence similarity to each other was used. This specific version was used because it is the last complete manual curated version. The secondary structure for each sequence in the database was collected using DSSP. The database contains 9,540 sequences.

A set of 69 MSA based on the same Astral SCOP 1.73 database was provided by the Department of Molecular Biology at the University of Salzburg.[2] Each MSA is built up from 15 homologous sequences from the same superfamily. For the MSAs, the secondary structure from DSSP is also used.

4.2 Evaluation

The methods used in the previous chapter were tested with varying parameters on all 69 datasets. To keep this Section compact, only the relevant tests and results for specific methods will be presented.

For the first part of this section, mainly the MSA representing the SCOP superfamily `c.67.1` will be used. The superfamily c.67.1 from the class *alpha and beta proteins (a/b)* represents *pyridoxal phosphate-dependent transferase*, proteins involved in the biosynthesis of amino acids dependent on pyridoxal phosphate, the active form of vitamin B6. The MSA with 15 sequences is 500 columns long, while its pHMM has a length of 384 match-states, as the remaining 116 columns have more than 50% gaps and therefore fo not represent a match state. In the SCOP database used, 60 out of the 9,540 sequences are classified as belonging to the superfamily `c.67.1`. Unless otherwise stated, the method and parameters used for the scores, including the secondary structure is *M2_025_1_5*, the exact configuration will be described later in this section.

[1] Astral SCOP 1.73 `https://scop.berkeley.edu/astral/ver=1.73`
[2] Department of Molecular Biology: `https://biwww.che.sbg.ac.at/`

Figure 4.1 shows the change of the scores for the different scoring methods generated by HMModeler. A scatter-plot is used to visualize the variation for each sequence, with the old method, using only the primary structure, on the horizontal axis. On the vertical axis, the difference between the same score and the new score that includes secondary structure information is displayed. Scores from sequences with the same superfamily, as the MSA used to build the pHMM, are marked as red circles, while all other scores are represented by blue circles. The left figures show the scores related to the viterbi scoring method, the right figures show the forward method. The top figures, representing the plain scores, show the improvement, as the superfamily scores increase noticeably more than the other scores. However, as the plain scores are also influenced by the sequence length, the reverse scores used for compensation are shown in the second row. The simple null model is not included in the figures as it still only uses the primary structure, therefore there is no change in the scores. The four figures at the bottom show the final scores, corrected using the null models.

The average changes for the scores in figure 4.1, divided into the superfamily `c.67.1` and the other scores, are listed below. The improvements from the plain viterbi and forward score are strongly compensated in the reverse corrected scores, as the new reversed scores adjust similarly to the plain scores.

```
Viterbi Score                 Superfamily: +72.7664
                              Other:       +21.8893
Forward Score                 Superfamily: +71.0713
                              Other:       +25.1106
Reversed Viterbi              Superfamily: +53.9577
                              Other:       +21.3254
Reversed Forward              Superfamily: +64.7312
                              Other:       +24.6887
Simple Corrected Viterbi      Superfamily: +72.7664
                              Other:       +21.8893
Simple Corrected forward      Superfamily: +71.0713
                              Other:       +25.1106
Reverse Corrected Viterbi     Superfamily: +18.8087
                              Other:       +0.56383
Reverse Corrected Forward     Superfamily: +6.34041
                              Other:       +0.42186
```

Listing 4.1: Average change for MSA c.67.1 scored against the SCOP database using primary structure only to including secondary structure.

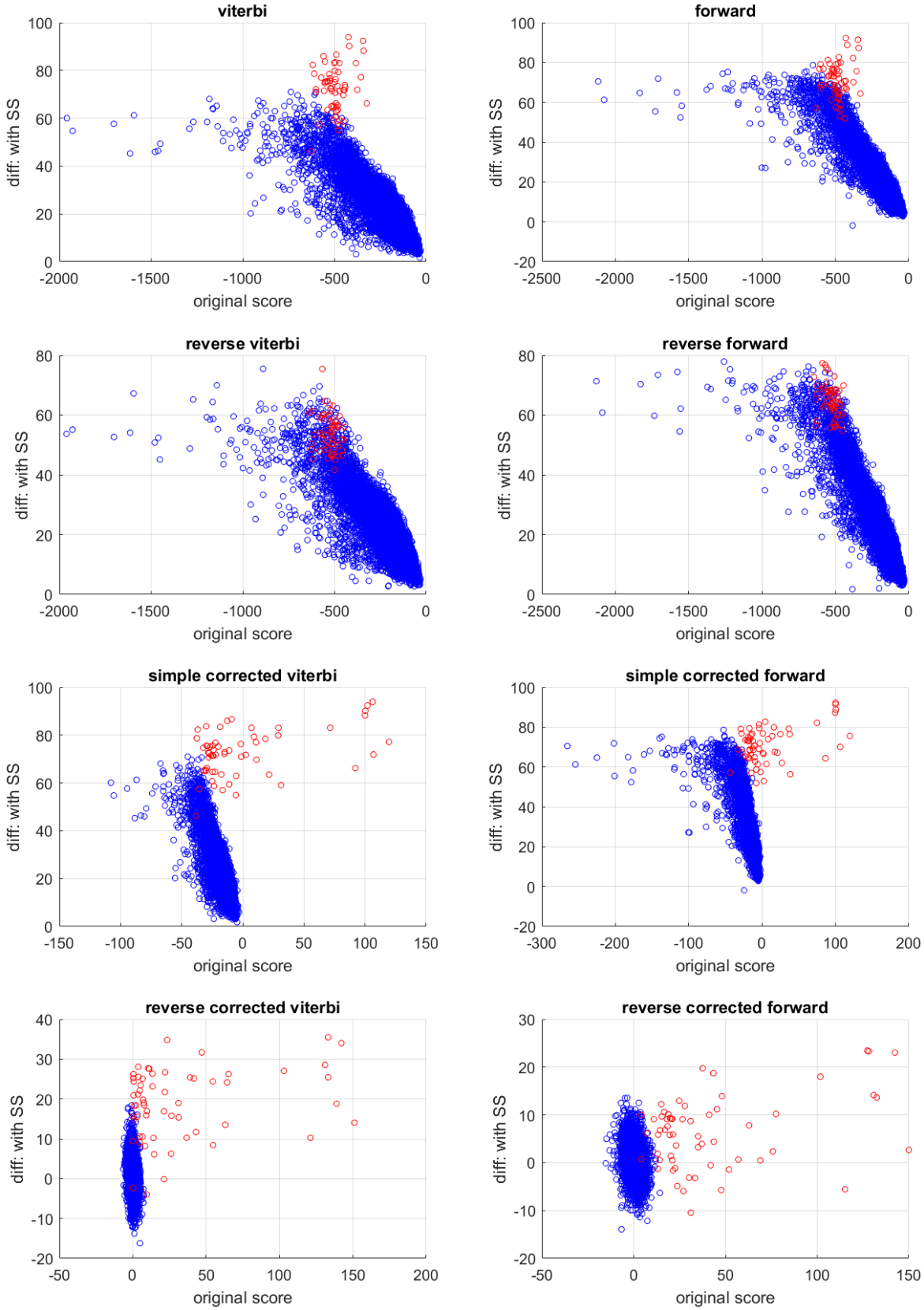

Figure 4.1: Comparison between scores with and without secondary structure information as scatter plot, with the score using only primary structure on the horizontal axis and on the vertical axis the difference from the primary structure score to the score including secondary structure information.

Figure 4.2 shows scatter plots with the simple corrected scores on the horizontal axis, and the reverse corrected scores on the vertical axis, generated from the original score with the primary structure only on top, and with secondary structure information below. For both approaches, viterbi and forward scoring, the improvements using the secondary structure are obvious, as the separation between the superfamily scores and the others increases.

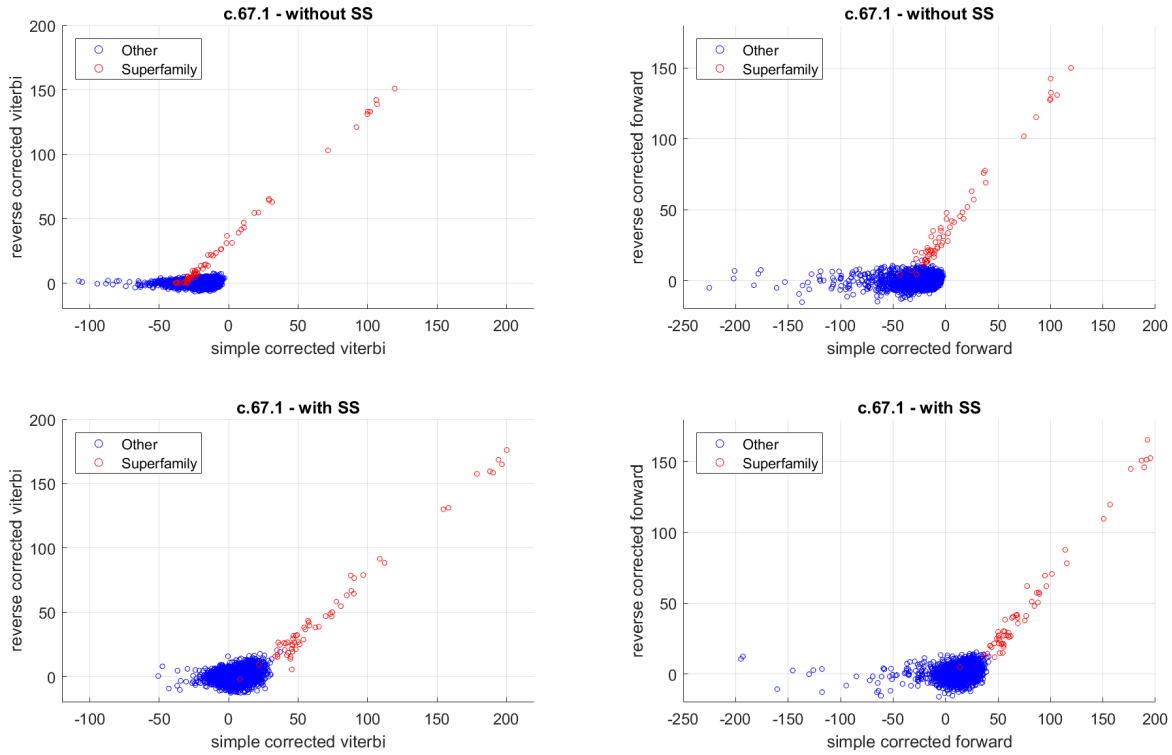

Figure 4.2: Scatter-plots of the corrected scores with and without secondary structure information.

This combination was used to evaluate the different weighting methods and their parameters described in Section 3.3, by scoring all 69 MSA against the SCOP database with different settings. The following five configurations will be explained in more detail.

- `M1_025_1` uses the weighting approach in (3.4), with a pseudo count of 1 for the secondary structure emission probabilities, and a threshold of 0.25 for the highest primary structure emission probability.
- `M1_025_3` uses the same approach as above, except with a pseudo count of 3.
- `M2_025_1_3` uses the weighting approach in (3.5), also with a threshold of 0.25 and a pseudo count of 1. The scale factor k is set to 3.

- `M2_025_1_5` uses the same parameters as M2_025_1_3 except with a scaling factor of 5.
- `M3_100_5` uses the Shannon approach in (3.8) with a pseudo count of 5 for generating the secondary structure emission probabilities, and a threshold of 100% resulting in always using the mixed probabilities for scoring.

For example, with the MSA `c.67.1`, the threshold of 0.25 for the first four approaches results in 272 columns where the highest emission probability is below the threshold; therefore the mixed probabilities are used, containing both the primary and the secondary structure. For the other 117 columns of the pHMM only the emission probabilities for the primary structure is used. For the last method, using the shannon entropy, it has been found that the best results are archived using the mixed probabilities only.

The different methods over the MSAs are compared in Matlab as listed in 4.2 by generating an Receiver Operating Characteristic (ROC) curve. A ROC curve compares the True Positive Rate (FPR) against the False Positive Rate (TPR) for varying thresholds of a classifier and is used to compare the quality of classifiers. The TPR measures the proportion of positives that are correctly classified as such, while the FPR specify negatives that are wrongly classified as positives, see [20, p. 34–35].

For each method and MSA the function *fitcsvm* trains a binary support vector machine classifier from *hmmScores*, containing the two dimensions for the reverse corrected and the simple corrected score and the *classes* indicating if the score relates to the tested superfamily or not. The function *fitPosterior* calculates the posterior probabilities for all scores, allowing *perfcurve* to generate the ROC curve.

```
1 mdlSVM = fitcsvm(hmmScores,classes,'Standardize',true);
2 mdlSVM = fitPosterior(mdlSVM);
3 [~,score_svm] = resubPredict(mdlSVM);
4 [X,Y,T,AUC] = perfcurve(hmmScores,score_svm(:,mdlSVM.
     ClassNames),'true');
5 plot(X,Y)
```

Listing 4.2: Matlab implementation for generating the ROC curve.

Figure 4.3 shows the generated ROC curve for the superfamily `c.67.1` using the methods described above, and the original score using only the primary structure listed as *AA*. As an optimal classifier would be a rectangular graph with 100% TPR at 0% FPR, the improvement of the new methods using secondary structure information is obvious.

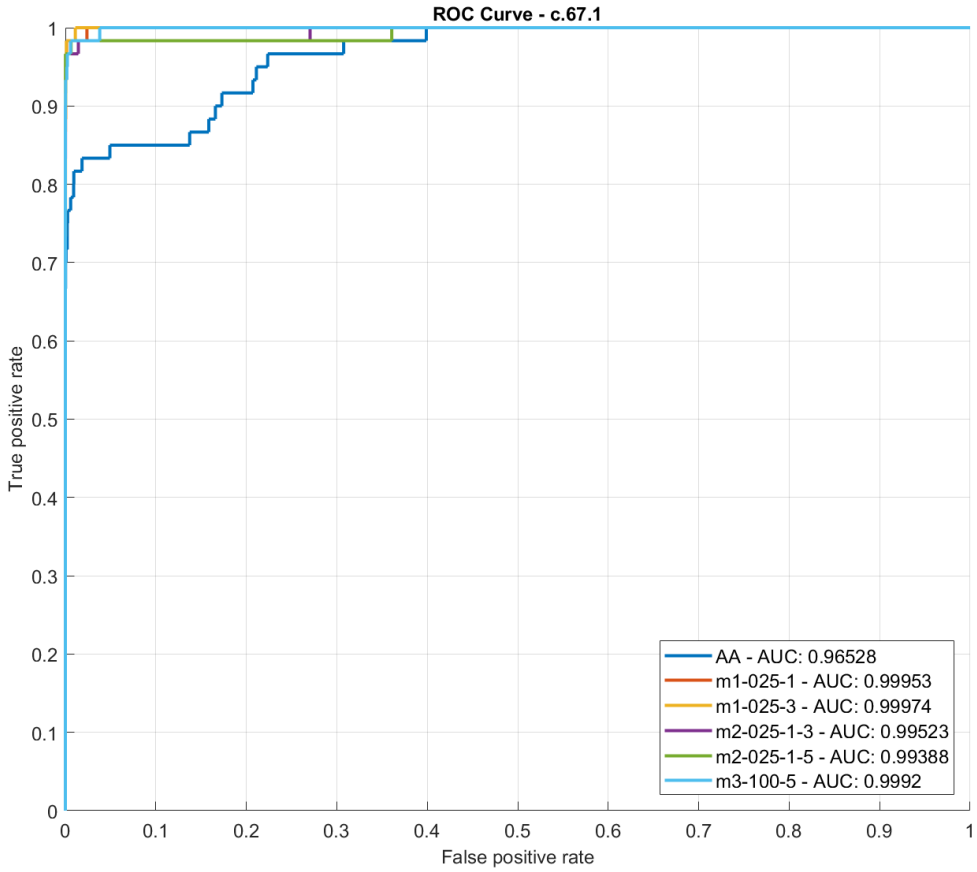

Figure 4.3: ROC curve for the MSA c.67.1.

For comparison of the different methods, the ROC curves can be ranked with the Area under the curve (AUC) score. The AUC score combines the whole ROC curve into one number in the range 0-1 and is calculated by the integral of the curve.

Table 4.1 provides an overview of the selected methods for the MSA. However, the table list only 54 MSA, as those with a variation of the AUC-score below 1% were filtered out. The Column *AA* lists the AUC score using only the primary structure. The highest AUC score for each MSA is highlighted in green, while all scores below the original method are highlighted in red. Summarizing the table, it can be said that except for a few scores, the secondary structure information improves the quality of the homology detection. For all MSAs the original score was improved by at least one method including the secondary structure. The best results were archived with the method `M2_025_1_5`, with only one score below the original score and 28 with the highest rank.

MSA	AA	M1_025_1	M1_025_3	M2_025_1_3	M2_025_1_5	M3_100_5
a.1.1	0.981660	0.999138	0.998953	0.992644	0.997789	0.997729
a.118.1	0.658467	0.753259	0.705426	0.746692	0.829864	0.644325
a.118.8	0.955319	0.984824	0.985946	0.986001	0.973027	0.977937
a.121.1	0.949624	0.995763	0.995616	0.996995	0.997438	0.983587
a.25.1	0.841495	0.937043	0.932680	0.941242	0.957709	0.937543
a.26.1	0.874773	0.983614	0.985158	0.969954	0.985943	0.964079
a.39.1	0.952474	0.981193	0.982275	0.959055	0.979121	0.948074
a.4.1	0.943772	0.955631	0.952671	0.969682	0.964489	0.918509
a.4.5	0.855347	0.926711	0.925392	0.908424	0.929695	0.916563
b.1.18	0.792153	0.817677	0.807238	0.815779	0.838481	0.802811
b.1.2	0.969961	0.995894	0.995995	0.981565	0.993793	0.997062
b.121.4	0.773631	0.967558	0.966336	0.866694	0.992417	0.953042
b.122.1	0.848138	0.845948	0.848089	0.845584	0.851564	0.779058
b.18.1	0.761240	0.930155	0.936215	0.817449	0.866927	0.929784
b.29.1	0.696135	0.960870	0.951270	0.849528	0.983448	0.940179
b.40.4	0.562231	0.645208	0.670961	0.429999	0.629892	0.567447
b.55.1	0.911226	0.979285	0.980650	0.928991	0.959427	0.957248
b.6.1	0.908563	0.960192	0.966695	0.937017	0.955909	0.957878
b.60.1	0.930032	0.984104	0.984360	0.970134	0.990348	0.978497
b.82.1	0.825039	0.938847	0.919801	0.866187	0.955782	0.912086
c.1.10	0.873501	0.874392	0.869802	0.872605	0.940730	0.872855
c.1.8	0.771208	0.928515	0.936675	0.773840	0.930788	0.945124
c.1.9	0.776061	0.984452	0.987866	0.849370	0.953804	0.968480
c.14.1	0.922587	0.973523	0.978553	0.895141	0.937373	0.943031
c.2.1	0.832922	0.877777	0.878921	0.830226	0.874660	0.849042
c.23.16	0.953418	0.984812	0.985529	0.946551	0.970777	0.979098
c.23.1	0.982697	0.997496	0.996486	0.996668	0.998236	0.994347
c.26.1	0.948562	0.976133	0.976961	0.960779	0.965373	0.976807
c.26.2	0.762557	0.851986	0.846126	0.772616	0.903719	0.878536
c.3.1	0.873175	0.889879	0.884945	0.874121	0.900278	0.855394
c.37.1	0.566633	0.770514	0.705588	0.633610	0.667131	0.721070
c.47.1	0.758197	0.851974	0.840596	0.832694	0.889318	0.819083
c.52.1	0.715274	0.687619	0.689950	0.728182	0.729150	0.689523
c.55.1	0.841824	0.890524	0.888007	0.872467	0.879920	0.896231
c.55.3	0.661511	0.724718	0.730029	0.744568	0.827794	0.709007
c.56.5	0.932965	0.975813	0.976139	0.931577	0.988535	0.970057
c.66.1	0.691936	0.792542	0.805687	0.763836	0.828851	0.826081
c.67.1	0.965279	0.999527	0.999741	0.995230	0.993875	0.999198
c.68.1	0.840345	0.954716	0.955924	0.896890	0.953844	0.942945
c.69.1	0.769510	0.951124	0.954956	0.886816	0.945654	0.938550
c.94.1	0.786186	0.885272	0.891178	0.822099	0.910171	0.838730
d.108.1	0.842174	0.985350	0.985038	0.890820	0.959551	0.956787
d.129.3	0.770835	0.906587	0.886342	0.779488	0.887589	0.929245
d.14.1	0.800006	0.820208	0.828362	0.813029	0.884111	0.847360
d.144.1	0.970321	0.980958	0.968966	0.968810	0.985371	0.979926
d.15.1	0.820850	0.841818	0.836339	0.938692	0.945802	0.819712
d.153.1	0.904484	0.963789	0.964523	0.911099	0.941617	0.949044
d.169.1	0.914845	0.984066	0.986238	0.949115	0.987850	0.936318
d.17.4	0.917448	0.997716	0.997654	0.949466	0.988231	0.996565
d.3.1	0.781971	0.922669	0.914309	0.795772	0.889263	0.920976
d.32.1	0.943212	0.988796	0.991518	0.974388	0.983423	0.983284
d.38.1	0.927538	0.975666	0.971828	0.955195	0.986411	0.971640
d.58.4	0.923971	0.939588	0.928825	0.936412	0.951180	0.935353
d.81.1	0.803026	0.821645	0.809739	0.798990	0.823598	0.824889
d.92.1	0.747920	0.791525	0.794027	0.803710	0.833569	0.819239
g.39.1	0.984596	0.982233	0.982742	0.988260	0.983897	0.966715

Table 4.1: AUC scores for different scoring methods compared to the original Method in column *AA* using primary structure, the highest score for each MSA is marked green, and red marked scores are below the original Method.

5 Conclusion

This paper shows methods for protein homology detection with pHMM using secondary structure information. The key tasks were to determine secondary structure information from a primary structure, build a pHMM using both primary and secondary structure information, and extend the scoring methods to make use of secondary structure information. Those steps have been implemented in the software package *HMModeler*.

The secondary structure information is determined by the PDB database, where available, or predicted using *MetaSSPred*. Building the model for the pHMM was extended by three different methods for generating the emission probabilities for the primary and secondary structure, and an additional set merging both probabilities. The viterbi and forward algorithm used for aligning sequences against the pHMM was extended by an additional threshold, which defines if the mixed probabilities are used or just the primary structure. The implementation was tested using the ASTRAL SCOP database and a set of 69 MSAs representing different superfamilies in the database. The tests show a general improvement when using secondary structure information.

This report will be used as base for the ongoing master's thesis with the title *"Expert Tuned Profile Hidden Markov Models for Primary and Secondary Structure Based Homology Prediction in Bioinformatics"* at the Salzburg University of Applied Sciences[1].

[1] https://www.fh-salzburg.ac.at/

Bibliography

[1] D. L. Nelson and M. M. Cox, *Lehninger Principles of Biochemistry*, 6th ed. New York: W.H. Freeman and Company, 2013.

[2] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[3] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, 1983.

[4] J. M. Berg, J. L. Tymoczko, L. Stryer, and N. D. Clarke, *Biochemistry*, 5th ed. New York: W.H. Freeman, 2002.

[5] IUPAC-IUB Joint Commission on Biochemical Nomenclature, "Nomenclature and symbolism for amino acids and peptides. recommendations 1983," *European Journal of Biochemistry*, vol. 138, no. 1, pp. 9–37, 1984.

[6] M. Y. Galperin, X. M. Fernández-Suárez, and D. J. Rigden, "The 24th annual nucleic acids research database issue: a look back and upcoming changes," *Nucleic acids research*, vol. 45, no. D1, pp. D1–D11, 2017.

[7] F. C. Bernstein *et al.*, "The protein data bank. a computer-based archival file for macromolecular structures," *European Journal of Biochemistry*, vol. 80, no. 2, pp. 319–324, 1977.

[8] P. W. Rose *et al.*, "The rcsb protein data bank: integrative view of protein, gene and 3d structural information," *Nucleic acids research*, vol. 45, no. D1, pp. D271–D281, 2017.

[9] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "Scop: a structural classification of proteins database for the investigation of sequences and structures," *Journal of molecular biology*, vol. 247, no. 4, pp. 536–540, 1995.

[10] N. K. Fox, S. E. Brenner, and J.-M. Chandonia, "Scope: Structural classification of proteins–extended, integrating scop and astral data and classification of new structures," *Nucleic acids research*, vol. 42, no. Database issue, pp. D304–9, 2014.

[11] U. Mayer, "Secondary structure profile hmms," Master's thesis, Salzburg University of Applied Sciences, Salzburg, 2014.

[12] B. Rost and C. Sander, "Prediction of protein secondary structure at better than 70% accuracy," *Journal of molecular biology*, vol. 232, no. 2, pp. 584–599, 1993.

[13] B. Rost, G. Yachdav, and J. Liu, "The predictprotein server," *Nucleic acids research*, vol. 32, no. Web Server issue, pp. W321–6, 2004.

[14] Md N. Islam, S. Iqbal, A. R. Katebi, and Md T. Hoque, "A balanced secondary structure predictor," *Journal of theoretical biology*, vol. 389, pp. 60–71, 2016.

[15] E. Faraggi, T. Zhang, Y. Yang, L. Kurgan, and Y. Zhou, "Spine x: Improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles," *Journal of computational chemistry*, vol. 33, no. 3, pp. 259–267, 2012.

[16] R. Graf, M. Aigner, M. Lechner, D. Schroffner, P. Lackner, and S. Wegenkittl, Eds., *HMModeler, a new approach for designing profile HMMs for protein families*. Proceedings of the 6th European Conference on Computer Systems, 2011.

[17] M. Oberkirchner, "Softwareentwicklung einer client-server-architektur für scientific computing am beispiel des bioinformatiktools hmmodeler," Master's thesis, Salzburg University of Applied Sciences, Salzburg, 2014.

[18] E. Sonnhammer, "Stockholm format." [Online]. Available: http://sonnhammer.sbc.su.se/Stockholm.html

[19] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[20] R. O. Duda, P. E. Hart, D. G. Stork, R. O. P. c. Duda, and scene analysis, *Pattern classification*, 2nd ed., ser. A Wiley-Interscience publication. New York and Chichester: Wiley, 2001.