MARSHALL PLAN SCHOLARSHIP PROGRAM REPORT

Improvement of the scoring schemes in HMM protein anaylsis: e-values, p-values and alternatives to Viterby-scoring

prepared for the
Marshall Plan Scholarship Program
in cooperation with Salzburg University of Applied Sciences
Undergraduate Program
Information Technology & Systems Management

submitted by:
Werner Umshaus

Head of Faculty:	FH-Prof. DI Dr. Gerhard Jöchtl
Supervisor:	FH-Prof. Univ.-Doz. Dr. Stefan Wegenekittl

Salzburg, March 2010

Details

First Name, Surname:	Werner Umshaus
University:	Salzburg University of Applied Sciences
Undergraduate:	Information Technology & Systems Management
Title of Thesis:	Improvement of the scoring schemes in HMM protein anaylsis: e-values, p-values and alternatives to Viterby-scoring
Academic Supervisor:	FH-Prof. Univ.-Doz. Dr. Stefan Wegenkittl

Keywords

1^{st} Keyword:	Forward Algorithm
2^{nd} Keyword:	Profile- HMM
3^{rd} Keyword:	Bioinformaticcs
4^{th} Keyword:	Protein Families
5^{th} Keyword:	e-values and p-values

Abstract

Profile Hidden Markov Models are prominent tools in protein analysis. Proteins can be represented by a chain of characters, each one out of a set of twenty different amino acids. Hidden Markov Model (HMM)s on the other hand rate the similarity of proteins using only their respective character representation. The degree of similarity is expressed by a score that is calculated by the so-called viterbi or forward algorithms. This paper is a report on the implementation of the forward algorithm for the HMModeler proteins scanner software developed at Salzburg University of Applied Sciences and Salzburg University.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Preview

HMMs are a very prominent tool in protein analysis. Proteins can be represented by a chain of characters, each one out of a set of twenty different acids. HMMs are used to generate a probability distribution on the space of such strings. Classical algorithms in the realm of machine learning and numerical statistics are applied in order to train and evaluate the HMMs for various protein families. The ultimate ambition is to correctly classify unknown proteins with respect to a genealogy of well analysed ones. This paper primarily aims to show the implementation of the forward algorithm. All important facts about biology that represent the basis for the algorithms are summarized in this chapter.

1.2 Algorithms

The most common dynamic programming used for decoding sequences, is the so-called viterbi algorithm [6, Kap. 3]. This algorithm finds the most probable path of hidden states in an HMM given the observation of output symbols. Other algorithms are:

- Forward algorithm (This algorithm is used to define the probability of generating the sequence through any paths from a HMM)
- Backward algorithm (A variant of the forward algorithm that processes in the opposite direction.)

- Forward- Backward algorithm (Combination from forward and backward. Is used in order to find the probability in any state at any time,[4, Kap. 7.3.2].)

The main focus of this paper is on the forward algorithm. The goal is, to enrich the arsenal of scoring variants by the use of the forward algorithm. For optimal results when using the above algorithm, it is useful to use a null model, because the highly variable length of the sequences in a database would otherwise yield scores that are difficult to compare and rank. See [1] for a more elaborate discussion of these problems. When using such a null-model, scores are easier to compare. Another important issue is that one might run into numerical problems (underflow) caused by numerous multiplications of probabilities. This problem can be solved using a log-space representation, see chapter 2.4.

1.3 Profile-HMM

First of all, HMMs will be described. Simply put, this is a statistical model which describes the emission probabilities for all characters for each position, [8, Kap. 2.6.8]. The goal is to find a similar or related protein sequence and family in a protein database. So-called profile HMMs mainly consist of three types of states (match, delete, insert) which are connected with arrows representing possible transitions between the states. Figure 1.1 shows these three states, match states $M_1, \ldots, M_n$, insert states $I_1, \ldots, I_n$ and delete states $D_1, \ldots, D_n$. In this example there is a known model length $n = 4$. Through transitioning, it is possible to go into another state and thereby generating the next symbol in the output sequence. If the next state is a match state or an insert state, it is possible to emit one of the symbols of the query sequence (in the delete state, no character is emitted by the HMM). This happens with different probabilities whose distribution depends on the state, the HMM is currently in.

Figure 1.1: Profile HMM,[8]

1.4 Coding

The existing software (HMModeler protein scanner software developed at Salzburg University of Applied Sciences and Salzburg University, [1] and [7]) is written in JAVA. Therefore, the source code for the algorithm had to be written in JAVA, too. Code fragments can be found in Chapter 3, which are important for understanding the algorithm. Other programs like for example Matlab[1] were used to investigate the new part of the program and to show results in diagrams and other illustrations.

---

[1] http://www.mathworks.com

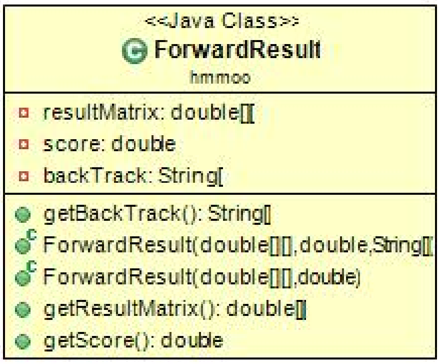

Figure 3.5: ForwardResult class diagram

the values from these variables, e.g. score, the function getScore():double returns the value etc.

3.3 Results

In this chapter we compare the scores calculated by forward and viterbi algorithms. This calculation was used on the same system as described in chapter 3.2.
The output will produce 6 different scores, they are:

1. Plain score
2. Simple score
3. Reversed score
4. Simple corrected score
5. Reversed corrected score
6. HMMer corrected score

The *Plain score* is the plain output from the HMM. The second score, the *Simple score*, is the output score from the simple null models. The idea behind the usage of a null model is to produce a target sequence (from the observed HMM) with a extremly simplified HMM consisting of only one insert state. This will be done in a loop until the target sequence is reached. But when this length-dependent score, which does not regard additional information in the HMM, is subtracted from the score, the result will be a good normalised score. The *Reversed score* is the forward score with the HMM from back to front which is an alternative to the simple null model for doing length normalization. The fourth to sixth score are corrected scores that depend on scores 1-3: The idea behind these scores is the inclusion of null models in the forward score. In other words, the log score from the forward algorithm will be reduced by a null model log score. With that it is possible to reduce the length dependence of the scores. Put differently, we calculate the amount by which the *HMM-score* is higher than the null model score. This idea will be used in the *Simple corrected score* through the simple score. Using the *Reversed corrected score* is following the same idea as the simple corrected score, but there the reversed score is used. Finally, the *HMMer corrected score* will produce the score with two null models, details for this procedure can be found in [1].

In Listing 3.1 an output with all scores with the forward and viterbi algorithm is depicted. When the score has a high value, it can be clearly seen that the protein is related to the particular protein family. Another indicator for a relation with the protein family is a high count of match states in the path. In comparison with both scores, it is important that the simple scores are the same. This is because it is the null model which is the same in both calculations. The other scores are usually higher than the viterbi scores because the forward calculation is the sum of the paths. Some background about the mathematics for the forward algorithm and the differences between these two algorithm can be found in chapter 2.2 and 2.3.

```
Sequenz: stlyeklggttavdlavdkfyervlqddrikhffadvdmakqrahqkafltya
fggtdkydgrymreahkelvenhglngehfdavaedllatlkemgvpedliaevaavagapa
hkrdvlnq
Plain Score: −439.2257937682402
Plain Score (Forward): −398.22664464796867
Simple Score: −441.1580822956077
```

```
 Simple Score (Forward): -441.1580822956077
 Reversed Score: -435.26756686881646
 Reversed Score (Forward): -395.0170941859909
 Simple Corrected Score: 1.9322885273675183
 Simple Corrected Score (Forward): 42.93143764763903
 Reverse Corrected Score: -3.958226899423721
 Reverse Corrected Score (Forward): -3.2095504619777557
HMMer corrected Score:  1.9315035588126432
HMMer corrected Score (Forward):  42.931437647638745

 Sequenz: gllsrlrkrepisiydkiggheaievvvedfyvrvladdqlsaffsgtnmsr
 lkgkqveffaaalggpepytgapmkqvhqgrgitmhhfslvaghladaltaagvpsetit
 eilgviaplavdvts
 Plain Score: -453.0267523965976
 Plain Score (Forward): -410.0064834039041
 Simple Score: -459.31210239376827
 Simple Score (Forward): -459.31210239376827
 Reversed Score: -447.7493415735511
 Reversed Score (Forward): -410.12682999410566
 Simple Corrected Score: 6.28534999717067
 Simple Corrected Score (Forward): 49.30561898986417
 Reverse Corrected Score: -5.277410823046523
 Reverse Corrected Score (Forward): 0.12034659020156369
HMMer corrected Score:  6.267806473656339
HMMer corrected Score (Forward):  49.30561898986417
```

Listing 3.1: Scores of the hemoglobin Cyanobacteria (Synechocystis sp.) and hemoglobin Mycobacterium tuberculosis HbNt with an HMM

4

Evaluation

To determine the statistical significance of a score from the forward or viterbi algorithms, a set of twenty trained protein family models has been scored against a database of known proteins. Astral, the database with version 1.73 from the Structural Classification of Proteins (SCOP) classification[1] will be used. The database contains different families which are labeled. They start with a letter followed by three numbers and separated by dots, e.g. a.1.1.2.

From these twenty proteins we build a HMM for every family and calculate a score from all sequences in the database. The scores which are calculated are the same as described in chapter 3.3. Families, which are used can be found in Appendix A.

4.1 Density functions

For the distribution, meaning how the scores are distributed in the database, it is necessary to use a density function. This density function is implemented in Matlab as function *ksdensity* and will be used for this evaluation. A density function will be created from these twelve scores. More information about this function can be found on the Matlab website[2].

Figure 4.1 shows the density function for every single HMM in the database which looks like a "hat". It can be seen that in most scores the density is independent from the HMM (in all twenty test families) and looks similar. An exception is the last one where

---

[1] http://astral.berkeley.edu/scopseq-os-1.73.html

[2] http://www.mathworks.com/access/helpdesk/help/toolbox/stats/ksdensity.html

the densities differ from each other. The reason for that can be the length correction for the null-model which does not work very well.

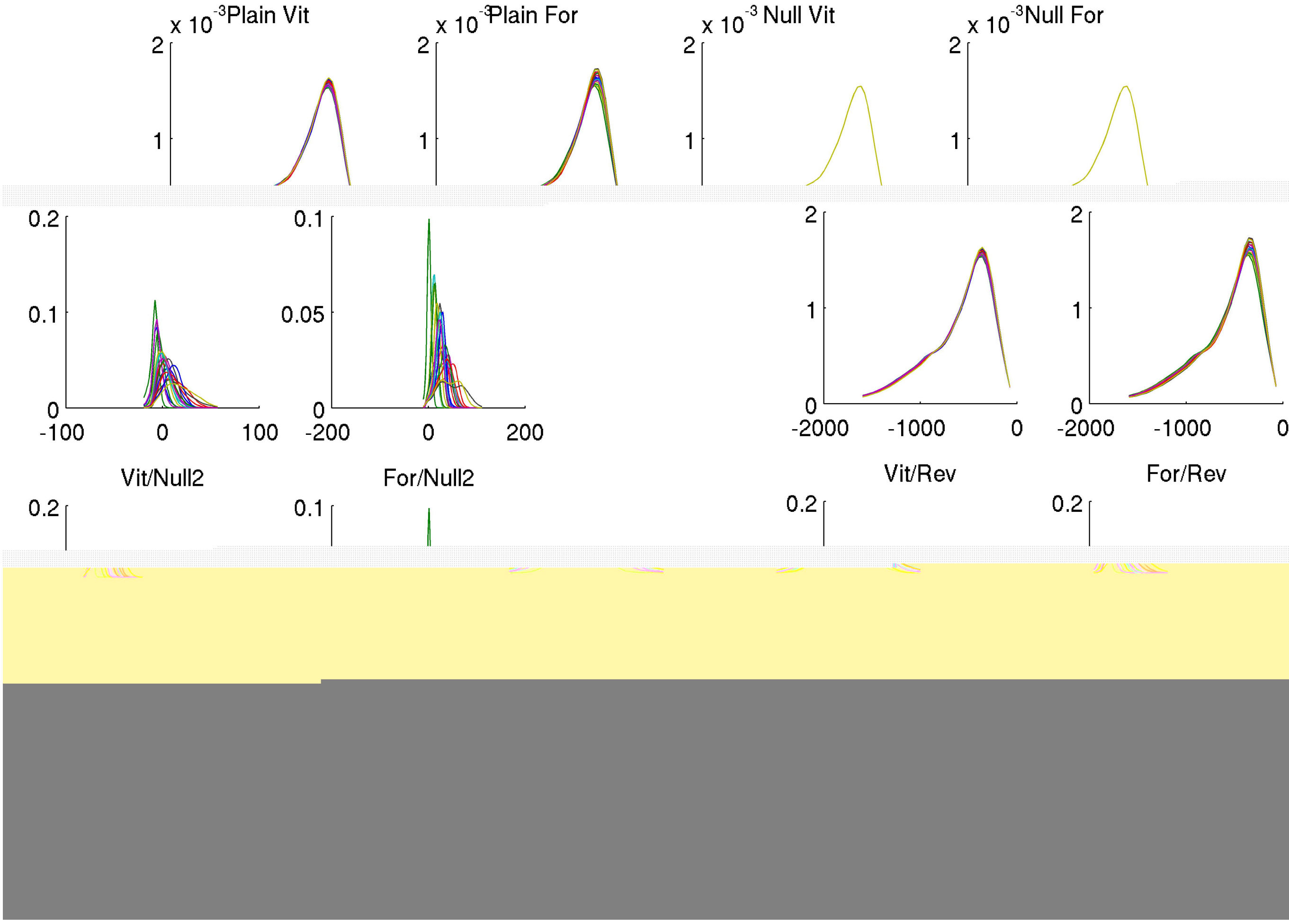

Figure 4.1: Density function for 20 families [3]

4.2 Outcome

Based on the previous results with this density approximation, in the next steps, probability values (p-value) and extreme values distribution (e-value) will be worked on. This is a statistically meaningful analysis of the raw-scores (the output of viterbi and forward algorithms). An estimator for e-values gives the probability of a protein

---

[3] graphics compiled by Stefan Wegenkittl

belonging to a family by relating the raw-scores S to the probability of finding an even better score x at random in the search database. A p-value is the probability of an alignment where there is at least one score S equal or better to some score x in the search database, [2]. Following [2], we thus have:

$$
\begin{aligned}
P(x, n) &= 1 - e^{-nP(S \geq x)} \\
\epsilon(x, n) &= nP(S \geq x) \qquad (4.1)
\end{aligned}
$$

The idea behind p-values is to integrate the distribution of scores from the score reached by certain sequence x until infinity. The results are an approximation for the probability which is to find a random high score in the database. e-values can be derived from p-values.

5

Conclusion and Outlook

Summing up, the implementation of the forward algorithm was successful. Now the software has two algorithms with the possibility to compare the corresponding scores. Sometimes the forward algorithm reaches better results than the viterbi. Furthermore, this is a basis for e- and p-values. With these raw-scores a statistical meaningful analysis can be created. In particular e- and p-values are the prefered variant of scores for many biologists when applying bioinformatic tools.
The next step is to write a bachelor thesis about this subject using this paper as a basis. Also, the source code will be published then.

Bibliography

[1] Auer, F.: *Scoring schemes and parameter prediction for profile hmms.* Diploma thesis, Salzburg University of Applied Sciences, 2009.

[2] Eddy, Sean R.: *Maximum likelihood fitting of extreme value distributions.* Technical report, Washington University School of Medicine- Dept. of Genetics, 1997.

[3] Hamzic, M.: *Proteinfamilienerkennung mit dem viterbi- algorithmus.* Diploma thesis, Salzburg University of Applied Sciences, 2008.

[4] P. Baldi, S. Brunak: *Bioinformatics: The Machine Learing Approach.* Massachusetts Institute of Technology, Cambridge, Massachusetts, 2001.

[5] Papula, L.: *Mathematische Formelsammlung fuer Ingenieure und Naturwissenschaftler.* Friedr. Vieweg & Sohn Verlag / GWV Fachverlag GmbH, Wiesbaden, Germany, 2006.

[6] R. Durbin, S. Eddy, A. Krogh G. Mitchison: *Biological sequence analysis.* Cambride University Press, Cambridge, UK, 1998.

[7] Radlingmaier, M.: *Optimierte Hidden Markov Modelle fuer das Sequenz-Scoring auf Basis von Mutliplen Alignments.* Diploma thesis, Salzburg University of Applied Sciences, 2007.

[8] Sperschneider, Volker: *Bioinformatics- Problem Solving Paradigms.* Springer, New York, USA, 2008.

[9] University of Leeds UK: *Hidden Markov Models- University of Leeds Homepage.* `http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html` (28.01.2010).

List of Abbreviations

HMM Hidden Markov Model

B Begin State

Mi Match State

Ii Insert State

Di Delete State

IB Insert Begin State

IE Insert End State

FB Flanking Begin

FE Flanking End

E End State

SCOP Structural Classification of Proteins

e-value extreme values distribution

p-value probability values

Appendix

A

Test families for evaluation

- a.1.1.2
- a.138.1.1
- a.25.1.2
- a.3.1.4
- b.121.4.1
- b.122.1.1
- b.45.1.1
- b.47.1.2
- b.6.1.1
- c.31.1.3
- c.36.1.9
- c.45.1.2
- c.46.1.2
- c.72.1.1
- d.131.1.2
- d.14.1.5
- d.153.1.4
- d.32.1.2
- d.38.1.5
- e.3.1.1