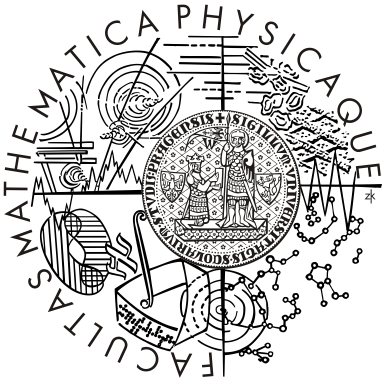

Karlova Univerzita v Praze
Matematicko-fyzikální fakulta

Vrije Universiteit Amsterdam
Faculteit der Exacte Wetenschappen

Diplomová práce - Master's thesis

Improving and extending the multiple sequence alignment suite PRALINE

Jan Hudeček

Kabinet software a výuky informatiky
Vedoucí diplomové práce: RNDr. František Mráz, CSc.
Studijní program: Informatika, softwarové systémy

Computer Science Department
student number: 1825658
coordinator: Jaap Heringa
second reader: Bernd Brandt

2010

I would like to thank my coordinators – prof. Heringa and Dr. Mráz – for helpful guidance. Their ideas and support had a major influence on this thesis. I also want to thank the whole Bioinformatics Section for keeping the doors opened so that students can come and ask for help. Especially Walter Pirovano, Bernd Brandt and Anton Feenstra.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 10.2.2010 Jan Hudeček

Abstrakt

Název práce: Improving and extending the multiple sequence alignment suite PRALINE
Autor: Jan Hudeček
Katedra (ústav): Kabinet software a výuky informatiky
Vedoucí diplomové práce: RNDr. František Mráz, CSc.
E-mail vedoucího: Frantisek.Mraz@mff.cuni.cz
Abstrakt: Cílem této práce bylo zkoumání možností vylepšení souboru programů pro zarovnávání mnoha sekvencí PRALINE. Nejprve je prezentován přehled současných metod pro řešení problému zarovnávání více sekvencí (multiple sequence alignment) s přihlédnutím k variantám reprezentace jádra zarovnání – mezivýsledku používaných algoritmů. PRALINE byla rozšířena o zarovnávání uživatelsky zadaných profilů. Uživatelem zadaný profil je v tomto rozšíření použit v pokročilejši fázi progresivního zarovnávání, jako by se jednalo o mezivýsledek předchozích kroků. Toto rozšíření bylo otestováno v typickém případu užití.

Přidali jsme 2 nové protokoly pro zarovnávání založené na skrytých Markovových řetězcích (HMM) a otestovali kvalitu jejich výsledků. Protokol HMMGUIDE vytvoří pro každou sekvenci preprofil skládající se ze segmentů ostatních sekvencí s vysokou lokální podobností. Z preprofilu HMMER vygeneruje pro každou sekvenci HMM a PRC zjistí stupeň podobnosti mezi dvojicemi HMM. Protokol pak progresivně zarovnává sekvence, jejichž HMM byly nejpodobnější. Protokol PRCALIGN postupuje obdobně, ale pro zarovnání použije výstup z PRC, sekvence tedy zarovná podle nejlepšího zarovnání HMM. Přestože protokoly nedokončily všechny testy úspěšně, výsledky ukazují významné zlepšení oproti původní metodě.
Klíčová slova: analýza sekvencí, Markovovy řetězce, PRALINE

Abstract

Title: Improving and extending the multiple sequence alignment suite PRALINE
Author: Jan Hudeček
Department: Department of Software and Computer Science Education
Supervisor: RNDr. František Mráz, CSc.
Supervisor's e-mail address: Frantisek.Mraz@mff.cuni.cz
Abstract: The aim of this work is to study potential improvements in the core routines of multiple sequence alignment suite PRALINE. A general overview of multiple sequence alignment methods used with emphasis on representation of the alignment core is given. A new option for aligning sequence profiles was implemented and its usefulness assessed. This option allows a user to input a profile which is used in an advanced phase of the progressive protocol as if it was a result of the previous steps.

Two new protocols using profile Hidden Markov models (HMM) and their alignment were implemented and tested. The HMMGUIDE protocol creates for each sequence a preprofile consisting of segments of other sequences with high local similarity. HMM is generated from each preprofile by HMMER, and alignment of every pair is scored by PRC. The protocol then progressively aligns the sequence whose HMMs achieved the best score. The PRCALIGN protocol works similarly but aligns the sequences according to the best alignment of the HMMs. While not all test alignments were finished successfully for both protocols, the results constitute a statistically significant improvement over the original PRALINE protocol.
Keywords: sequence alignment, Hidden Markov Model, PRALINE

Table of contents

1. Introduction

This project was done over the course of 8 months as a part of Software Systems master degree programme at Charles University in Prague and High-Performance Distributed Computing at Vrije Universiteit in Amsterdam as a part of the Short track Master's programme.

The thesis focused on improving the quality of the multiple sequence alignment suite PRALINE [Heringa, 1999; Heringa, 2002; Simossis & Heringa, 2005; Pirovano et al., 2008]. Multiple sequence alignment is an important problem in biology, it is used for constructing phylogenetical trees, it helps to derive structure and function for proteins and more. There are several tools available for computing multiple sequence alignment: PROMALS [Pei et al., 2008], CLUSTALW [Thompson et al., 1994] and many others. One of them is PRALINE, hosted [Simossis & Heringa, 2003] at Vrije Universiteit in Amsterdam, accessible at http://www.ibi.vu.nl/programs/pralinewww/. It uses sophisticated methods for improving the quality of the alignment like transmembrane segments prediction, PSI-BLAST [Altschul et al., 1997] and secondary structure prediction. However, we believe that the quality of alignments can be improved as other alignment methods sometimes provide more accurate results.

Chapter 2 of the thesis describes pairwise sequence alignment in detail including an important concept of homology. For definition and discussion of the multiple sequence alignment consult Chapter 3. This chapter also includes a look at the current methods used to perform the multiple sequence alignment including the ones used in PRALINE (Chapter 3. 3. 7). Chapter 4 introduces Hidden Markov models that were used in some of the new protocols for multiple sequence alignment that were implemented in PRALINE. These new protocols are described in more detail in Chapter 5. Chapter 6 presents the results of these protocols and compares them directly with the current version of PRALINE. Chapter 7 summarizes the conclusions and discusses possible further work.

The results of this work are accessible at http://www.ibi.vu.nl/programs/pralineprofwww/.

Results of the experiments and the modified source code can be found at http://dl.dropbox.com/u/98435/thesis.zip.

2. Sequence alignment

In this chapter I describe the problem of pairwise sequence alignment and algorithms used to solve it.

2. 1. Description of the problem

Sequence alignment means arranging strings of characters (which may stand for amino acids in protein sequences or nucleotides in DNA or RNA sequences) so that related characters end up aligned in one column. In this work, we will only deal with protein sequence alignment.

Pairwise sequence alignment is used to obtain a measure of similarity (from the percentage of identical residues or by using scoring matrices) of two proteins. It helps to derive 3D structures of proteins from known similar structures, and it generally allows inferring properties of a sequence that can be sufficiently well aligned with another sequence sharing a common ancestor sequence; sequence alignment also helps to find such ancestor sequences.

A formal description of the protein sequence alignment follows.

Input: sequences $S_a = (a_1, .., a_n)$ and $S_b = (b_1, .., b_m)$ where $a_1, .., a_n, b_1, .., b_m \in C$={ ‘A’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’, ‘H’, ‘I’, ‘K’, ‘L’, ‘M’, ‘N’, ‘P’, ‘Q’, ‘R’, ‘S’, ‘T’, ‘V’, ‘W’, ‘Y’}, $n,m \geq 1$.

These letters are one-letter codes for amino acids (for actual meaning of each letter consult for example [Bourne & Weissig, 2003]. The proteins are linear chains of these amino acids.

Output: $\alpha_1, .., \alpha_p$ and $\beta_1, .., \beta_p$ where α_i is either a gap (noted as ‘.’ or ‘–‘) or $a_{k(i)}$ where $k(0) = 0$, $k(i) = k(i–1)$ for gaps and $k(i) = k(i–1)+1$ otherwise. Similarly for β_i. At the same time the arrangement $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is such that the sequence scoring function

$$F(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i=1}^{\mathrm{p}} f(\alpha_i, \beta_i)$$

is maximal over all alignments. Function f is called a scoring function. It is defined on $C \cup \{\text{'-'}\} \times C \cup \{\text{'-'}\}$. The value of f can be obtained by using a substitution matrix S: $S[a_i,b_j]$ specifies the score (bonus or penalty) for aligning amino acid a_i with b_j. It can be simply 1 when amino acids match and –1 otherwise or a log-odds score for each combination of amino acids (as in PAM [Dayhoff et al., 1978] or BLOSUM [Henikoff and Henikoff, 1992]). Later we discuss other possible ways how to calculate value of the scoring function.

The number of possible sequence alignments grows exponentially with the lengths m,n of sequences according to the formula [Waterman, 1995]:

$$A(m,n) = \sum_{k=0}^{\min(m,n)} \frac{(m+n-k)!}{k!\,(m-k)!\,(n-k)!}$$

Thus evaluating it for all sequences would be impossible even for very short protein sequences.

To overcome this, multiple methods have been designed. Initially simple ones such as the dot matrix method [Maizel and Lenk, 1981] were used. This method could be easily performed by hand without the need for a computer as it does not involve any complex calculations. It creates an $n \times m$ matrix with columns corresponding to residues of the first sequence and rows to the residues of the second sequence, with a dot in a cell $M_{i,j}$ if a_i is equal to a_j. Diagonal stretches of dots indicate similarity in the corresponding parts of sequences. As Figure 1 shows, the plot contains a lot of "noise". This can be alleviated by using a method called windowing. Windowing is used mainly in

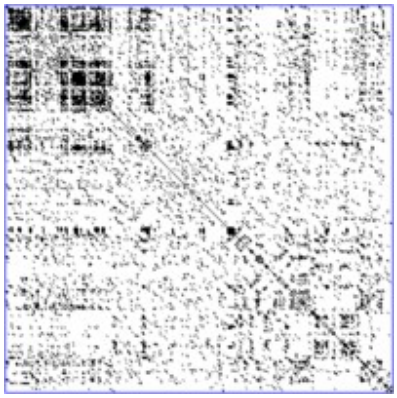

Figure 1 Dot matrix method

DNA/RNA sequence alignment where instead of comparing single nucleotides, we compare segments or windows of length larger than 1 [Wilbur and Lipman, 1983]. However, finding the best alignment is impossible with this method, the sequences have to be very similar to find a good alignment [Xia, 2001].

[Needleman and Wunsch, 1970] designed a dynamic programming algorithm to solve the problem in time $O(mn)$. This algorithm in the simplest form also creates an $n \times m$ matrix, but the cell $M_{i,j}$ corresponds to the best value of F for alignments of $a_1 \ldots a_i$ and $b_1 \ldots b_j$.

Initialization: All cells are initialized to 0 except cells in the first column and first row – these are initialized to i (resp. j) times a constant $c_1 > 0$ called "gap penalty". The gap penalty ($c_1 = f(\text{'}-\text{'}, a) = f(a, \text{'}-\text{'})$ for all $a \in C$) is a constant which controls how likely gaps are to occur in the alignment.

$$M_{i,j} = 0, M_{i,0} = -c_1 i, M_{0,j} = -c_1 j (\text{for } i, j > 1) \text{and } M_{0,0} = 0.$$

Matrix computation: The whole matrix is then filled in according to the recursive formula

$$M_{i,j} = \max\left(M_{i-1,j-1} + f(a_i, b_j), M_{i,j-1} - c_1, M_{i-1,j} - c_1\right) \text{for } i, j > 0.$$

Backtracking: After filling the whole matrix the algorithm performs backtracking – starting from the final position $i = n$, $j = m$ it builds the alignment:

- If $M_{i,j} = M_{i-1,j-1} + S[a_i, b_j]$ then prepend a_i, b_j to the alignment and set $i = i - 1$ and $j = j - 1$.
- If $M_{i,j} = M_{i,j-1} - c_1$ then prepend '–', b_j to the alignment and set $j = j - 1$.
- If $M_{i,j} = M_{i-1,j} - c_1$ then prepend a_i,'–' to the alignment and set $i = i - 1$.

This is repeated until $i = 0$ and $j = 0$ (omitting impossible options when reaching the edge).

Example:

	–	A	M	I	R	D
–	**0**	–1	–2	–3	–4	–4
A	–1	**1**	**0**	–1	–2	–3
I	–2	0	–1	**1**	0	–1
R	–3	–1	–1	0	**2**	**1**

```
AMIRD
A-IR-
```

Figure 2 Backtracking and alignment

The matrix was filled by following the rules (using -1 as both a gap and mismatch penalty and +1 for a match, see Matrix Computation above) for sequences AMIRD and AIR. Bold numbers indicate a match between amino acids. The arrows show a possible backtracking route resulting in the alignment presented below the matrix.

This variant of the algorithm is called 'global alignment with linear gap penalties'. 'Global alignment' means that the two sequences are aligned from the first to the last residue. Another variant is 'semi-global alignment' where the leading and trailing gaps do not add penalties to the score. This can be useful when aligning a multi-domain protein against one of its domains. 'Local alignment' is an often-used variant of the problem, its aim is to find highest scoring segment of alignment i.e. subsequences which are most similar. It is solved using Smith-Waterman algorithm [Smith and Waterman, 1981]. It differs by disallowing negative cell values (all cells that would be negative are set to zero) and starting the backtracking in the cell with the highest value. One of the reasons why this approach works is that the scoring function is positive for a pair of similar amino acids and negative otherwise.

The method for scoring gaps we used in the example is called linear gap penalty: the penalty for a gap of length k is $c_1 k$. Thus the penalty increases linearly

with the length of the gap. Another method for calculating the penalty for long gaps is an affine gap penalty function and it models the situation in nature where an insertion is an unlikely event, but when it happens it can easily be more than one amino acid long. The cost of a gap 10 amino acids long incurs smaller penalty than ten isolated gaps of a single amino acid. The penalty is calculated as $c_2+c_1(k-1)$ by using one constant for starting a gap (c_2) and another smaller constant for extending it (c_1) [Smith and Waterman, 1981].

2. 2. Homology

Sequence alignment is an important tool for biologists because when two sequences have a high similarity (>30% of residues are identical), it is likely that they originated from the same ancestor organism possibly far in the past; these sequences are called homologues. Homology is an important concept because it can be used to infer unknown features (function, structure, etc) of a sequence from a homologue for which these features are known. With declining sequence similarity the probability that the alignment happens due to pure chance and not because the sequences are biologically related increases, and at the same time these sequences are increasingly difficult to align in a biologically meaningful way. Sequence pairs from a so-called 'twilight zone' (20%–30% of sequence identity) are particularly difficult to align correctly, while it is crucial to get the highest possible score [Sander and Schneider, 1991].

Most proteins fold into a complex 3D structure. To understand the structure better it is beneficial to consider it at several levels. The sequence of amino acids in the protein is often referred to as the primary structure. This primary structure can form secondary structure elements – a helix, a sheet or a coil. When the chain of amino acids folds into a spiral we call it a helix. The most common type of helix is left-handed α-helix with 3.6 residues per turn. A sheet is formed by two or more parallel chains in the same plane, for historical reasons it is sometimes called β-sheet. Parts of the chain which are neither helices nor sheets are said to have secondary structure 'coil'. Biochemists also talk about tertiary structure at the level

of the whole protein molecule and quaternary structure when multiple protein subunits bind together to create protein complexes.

The secondary structure in which a gap occurs can often be more significant than its length. For example a single additional amino acid in a helix region causes the rest of the helix to be rotated by approximately 100 degrees (including the structure after the helix), which can significantly alter the fold of the protein, its functional sites and other properties. Similarly when an odd number of residues is inserted into a beta-strand it changes the direction which the residues are facing. A gap in a coil region does not have to have so pronounced effects.

Depending on the actual position of the strand or helix, such an insertion can have profound effects on the geometry of the protein and on the way it folds (see Figure 3). However, without knowing the complete fold of the protein it is very hard to decide whether placing a gap at a particular position will have such a profound effect.

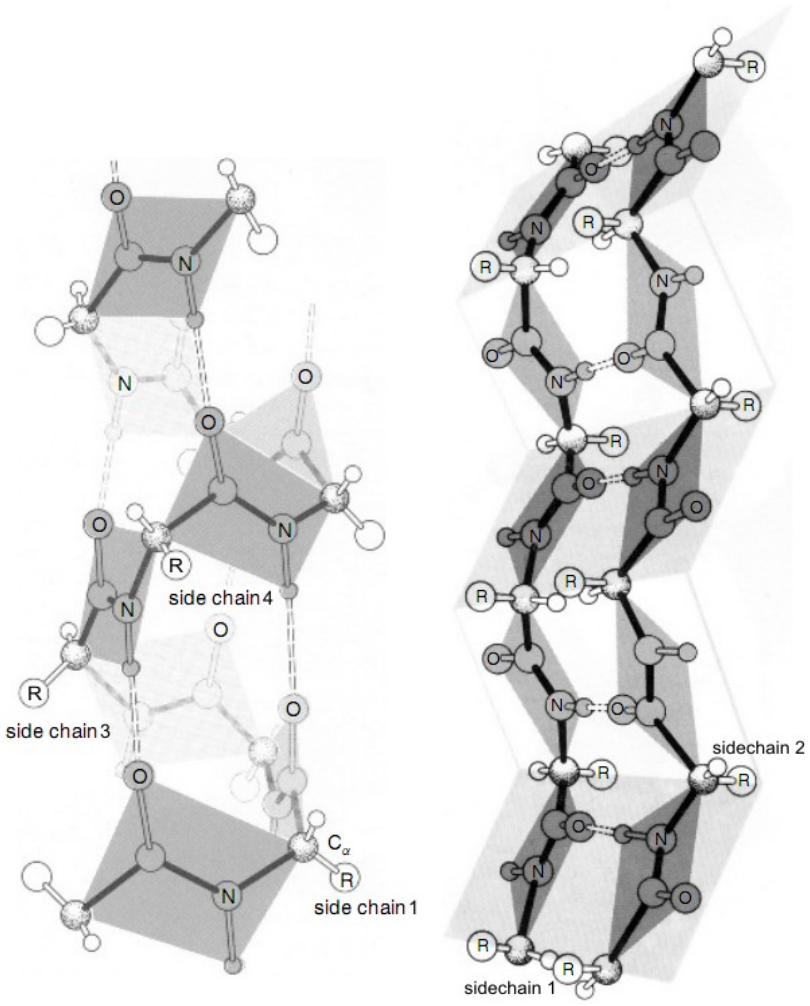

Figure 3 Protein secondary structure

Diagram of an α-helix using ball-and-stick model (left) and an antiparallel β-sheet (right) (adapted from [Bourne and Weissig, 2003]).

An alignment placing a gap at sidechain 3 of the helix would effectively remove this residue from the protein shifting residue 4 and all following by one place. This means that from this residue all sidechains extend into a completely different direction than before.

However, the situation in β-sheets can be even worse – removing residue 1 in a similar fashion swaps the sidechains of both sides of the β-sheet.

2. 3. Scoring

Sequence alignment should (but often does not) align similar amino acids and sequence features from the sequences. Secondary and tertiary structure elements, functional sites, locations of post-translational modifications and other important locations should be in the same columns in the alignment. Unfortunately, this does not always happen – even related sequences sometimes differ in function or 3D structure of a particular segment.

The scoring function should ideally incorporate information about all sequence features as soft constraints. We have already introduced simplest forms of scoring – a penalty for a mismatch or a gap and a bonus for matching residues. However, often a substitution matrix is used. A substitution matrix specifies for each combination of amino acids the bonus or penalty if these two residues are aligned together that is in line with the probability that in the past there was a mutation event from one to the other. These matrices are generated from sets of known alignments by counting the occurrence of mutations between particular pairs of amino acids. Thus, they should intrinsically represent penalties for changing properties of the side chains such as hydrophobicity, size, charge and so on, based only on the statistics of mutations. Example substitution matrices include BLOSUM [Henikoff and Henikoff, 1992] or PAM [Dayhoff, 1978].

PRALINE uses these matrices only in the simplest case without secondary structure or trans-membrane prediction. When secondary structure is predicted for the sequences, Lüthy's [Lüthy et al., 1991] set of matrices is used. This set contains different matrices for scoring residues known to be in a helix, strand or coil region. When the assigned secondary structure does not match, PRALINE uses the matrix for coil. When trans-membrane prediction is requested it uses PHAT [Ng et al., 2000] for the membrane regions of proteins. By using these matrices, PRALINE's scoring function reflects more properties of the sequences than just the string of amino acids, while not enforcing it strictly. This allows aligning regions even when the secondary structures do not match.

3. Multiple sequence alignment

Multiple protein sequence alignment is an essential tool for biologists and its exact solution is an NP-hard problem [Wang and Jiang, 1994]. This chapter defines the problem, presents general techniques of solving it and discusses contemporary methods used and their advantages and drawbacks. Finally, it contains information about the current state of the PRALINE suite.

3. 1. Description of the problem

Multiple sequence alignment (MSA) – aligning more than 2 sequences so that corresponding letters in the sequences are in the same column – is a natural extension of the pairwise alignment described in Chapter 2. It reveals much more information about the sequences allowing observation of features conserved across whole families or identification of features whose absence or presence determines important functionality of given sequences.

Unfortunately, it is inherently more difficult than pairwise sequence alignment [Murata et al., 1985]; in fact it has been proven to be NP-complete. Moreover, when certain conditions are met it is proven that it does not have a polynomial-time approximation scheme [Wang and Jiang 1994]. Even when using some clever heuristics [Carillo and Lipman, 1988], MSA starts to be prohibitively computationally expensive for more than approximately 10 typical protein sequences [Helal 2008].

Biologists frequently need to perform MSAs of more than 20 sequences, thus performing an optimal alignment becomes infeasible even with the algorithms from the previous paragraph. They use MSA for a wide range of applications – phylogenetic mapping of sequences and generation of whole phylogenetic trees of organisms (although this is somewhat controversial [Brocchieri, 2001]), assessing similarity among a family of proteins, detection of mutation events, and identification of motifs and functional sites [Xiong 2006]. MSA is also used in

several secondary structure prediction tools [Rost et al. 2004; Cole et al. 2008], and for prediction of protein features [Bendtsen et al. 2004].

The performance features of the optimal algorithm make it unusable for MSAs of larger groups of sequences. Heuristic algorithms have been designed to overcome these problems. Most of these follow the "progressive algorithm" [Hogeweg and Hesper, 1984; Feng and Doolittle, 1987]. They first construct a "guide tree" – a rooted binary tree – which should be close to phylogenetic tree of the sequences, in which closely related sequences should end up near each other in a branch, whereas more divergent sequences should be further away in the tree. During the construction of the tree an average over all alignments between sequences in the branches [Corpet, 1987] is used as a score. Alternatively, the most similar sequence pair can be joined first and other sequences added to it one at a time [Barton and Strenberg, 1987]. The whole branch can be represented by some data structure instead: a simple consensus sequence [Hogeweg and Hesper, 1984], a profile [Taylor 1987; Gribskov et al. 1987] or a more complex data structure, like a hidden Markov model (HMM) [Churchill 1989] or finite state automaton (FSA) [Searls and Murphy 1995]. As these data structures are the main focus of this work we will describe them later in greater detail.

The next step is to align the sequences by traversing the tree performing a pairwise alignment starting with the two closest sequences. While scoring a pairwise sequence alignment is relatively straightforward, the direct equivalent in MSA – the so-called sum-of-pairs (i.e. sum of pairwise alignment scores) often scores the biologically meaningful and "correct" alignment with lower sum-of-pairs score than other alignments [Waterman and Byers, 1985]. This can be improved by using a scoring matrix that does take into account consistency, secondary structure and other possible sequence features in matrix-based methods, or a complex scoring function in consistency-based methods [Notredame et al., 2000].

All heuristic MSA methods make the (reasonable) assumption that closely related sequences should be "easy" to align, thus there will be fewer errors when aligning them than when aligning two distantly related ones. If any errors are made when creating the core of the alignment, it is very hard to correct them later ("Once a gap, always a gap" [Feng and Doolittle 1987]). The progressive alignment methods focus on building the most correct guide tree. However, they can be further extended by trying to find and correct the errors made in previous steps in an iterative fashion. Multiple algorithms employ iterations to improve the quality of the alignment: Probcons, MUSCLE, PRALINE and others.

3. 2. Representation of the alignment core

During the course of progressive alignment it is necessary to score two groups of (already aligned) sequences. This can be done by storing the "alignment core" of these sequences – i.e. what they have in common. The first attempt to store such information was with consensus sequences. Consensus sequence contains a particular amino acid at a position *i* if and only if all the sequences from the group also contain this particular amino acid at this position. This approach works only for the simplest cases of a few sequences with few or no mutations. Indeed even a common mutation would cause the position in a consensus sequence to be empty and thus carry no information. However, working with consensus sequences is easy and they require minimal changes to the alignment algorithm itself.

A better alternative [Notredame, 2002] is to use profiles. A profile is a matrix $(p_{i,j})$ storing for *j*-th position in the alignment counts or frequencies of amino acid (or gap) *i*. It typically has 20 rows (one for each amino acid) and as many columns as the alignment. It can thus record which amino acids are expected at a particular position. Typically gap counts are also stored so that the profile alignment score receives smaller penalty for a gap if the alignment at this position already contains some gaps. The scoring gets a bit more complicated though, for profiles with amino acid frequencies $p_{i,k}$ and $p'_{i,k}$ the score for position *k* can be calculated as

$$\text{score}_\text{k} = \sum_{i=1}^{20}\sum_{j=1}^{20} p_{i,k}p'_{j,k}S[a_i, a_j].$$

The substitution matrix can be written as

$$S[a_i, a_j] = \log\frac{P(a_i, a_j)}{Q(a_i)Q(a_j)},$$

where $P(a_i, a_j)$ is the probability of aligning amino acids a_i and a_j and $Q(a_i)$ is the relative frequency of amino acid a_i or background gap frequency. [Von Öhsen and Zimmer, 2001] proposed a modification to this scoring mechanism – the log-average score:

$$\text{score}_\text{k} = \log\sum_{i=1}^{20}\sum_{j=1}^{20} p_{i,k}p'_{j,k}\frac{P(a_i, a_j)}{Q(a_i)Q(a_j)},$$

which performs better than the previous model. MUSCLE [Edgar 2004] further improves the scoring using the log-expectation score:

$$\text{score}_\text{k} = (1 - p_{G,k})(1 - p'_{G,k})\log\sum_{i=1}^{20}\sum_{j=1}^{20} p_{i,k}p'_{j,k}\frac{P(a_i, a_j)}{Q(a_i)Q(a_j)},$$

where $p_{G,k}$ is the number of gaps in the column.

HMMs (Hidden Markov Models) are even more sophisticated; we describe them in greater detail in Chapter 4.

However, even HMMs do not carry all the information contained in a sequence alignment. All presented data structures model the alignment as if residues in the column *k* were independent from residues observed in some other column *l*, while this is seldom the case. Often a particular residue at position *k* appears always with a different residue at position *l* and the alignment clearly shows it (for example when the protein fold brings these residues close together). Such information can

be kept in partial order graphs; this approach is used in POA [Grasso and Lee, 2004].

Examples of representations of the alignment core

For the following sequence alignment:

```
KDKWVDDVRNER-G
K-NWIKELKTML-G
K-SWIKELQRQA-P
A-KWIQEIERYA-R
K-QWLSEIDRYA-S
M-KWVSDVDEYA-P
K-QWMQEIQRYA-C
K-MWLQEIDRYAYT
```

A consensus sequence would look like this:

```
...W.........
```

A profile (depending on sequence weighting scheme) would look like this:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	**0.125**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.750**	0.000	0.000
C	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.125**
D	0.000	**0.125**	0.000	0.000	0.000	**0.125**	**0.250**	0.000	**0.375**	0.000	0.000	0.000	0.000	0.000
E	0.000	0.000	0.000	0.000	0.000	0.000	**0.750**	0.000	**0.125**	**0.125**	**0.125**	0.000	0.000	0.000
F	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
G	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.250**
H	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
I	0.000	0.000	0.000	0.000	**0.375**	0.000	0.000	**0.500**	0.000	0.000	0.000	0.000	0.000	0.000
K	**0.750**	0.000	**0.375**	0.000	0.000	**0.250**	0.000	0.000	**0.125**	0.000	0.000	0.000	0.000	0.000
L	0.000	0.000	0.000	0.000	**0.250**	0.000	0.000	**0.250**	0.000	0.000	0.000	**0.125**	0.000	0.000
M	**0.125**	0.000	**0.125**	0.000	**0.125**	0.000	0.000	0.000	0.000	0.000	**0.125**	0.000	0.000	0.000
N	0.000	0.000	**0.125**	0.000	0.000	0.000	0.000	0.000	0.000	**0.125**	0.000	0.000	0.000	0.000
P	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.250**
Q	0.000	0.000	**0.250**	0.000	0.000	**0.375**	0.000	0.000	**0.250**	0.000	**0.125**	0.000	0.000	0.000
R	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.125**	**0.625**	0.000	**0.125**	0.000	**0.125**
S	0.000	0.000	**0.125**	0.000	0.000	**0.250**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.125**
T	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.125**	0.000	0.000	0.000	**0.125**
V	0.000	0.000	0.000	0.000	**0.250**	0.000	0.000	**0.250**	0.000	0.000	0.000	0.000	0.000	0.000
W	0.000	0.000	0.000	**1.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.625**	0.000	**0.125**	0.000

It is clear that this structure contains much more information than the consensus sequence.

3. 3. Approaches to MSA

This subchapter describes currently used methods for MSA and discusses their advantages and innovative features. We also describe PRALINE, its evolution and areas of possible improvement.

3. 3. 1. CLUSTAL W

CLUSTAL W [Thompson et al., 1994] (http://www.ebi.ac.uk/Tools/clustalw2/) is the most frequently used package for performing multiple sequence alignment. It is a classical progressive alignment approach – first it performs the pairwise alignment of sequences. Using these pairwise scores as inverse distances, CLUSTAL W then uses the neighbour-joining algorithm [Saitou and Nei, 1987] to create an unrooted binary guide tree. This tree is then rooted by mid-point method so that the mean lengths of the branches on either side are equal. This tree is later used for sequence weighting. The closest two sequences on this tree are then aligned and replaced by a profile representing their alignment. CLUSTAL W uses position-specific gap penalties – lower penalties for gaps in hydrophilic regions, for more distant sequences and for extending the gaps made in previous steps and higher penalties for gaps near existing gaps. The scoring matrices used for particular alignment depend on their similarity – an appropriate matrix is chosen from either the PAM or BLOSUM series of matrices. This alignment package is widely used (the original paper has almost 30 000 citations) even though there are better and faster methods.

3. 3. 2. T-COFFEE

T-COFFEE [Notredame et al., 2000] (http://www.ebi.ac.uk/Tools/t-coffee/) uses a library of pairwise sequence alignments to arrive at the final multiple sequence alignment. This library is generated from CLUSTAL W global alignment and the top ten local non-overlapping alignments from LAlign (a very fast local alignment program [Huang and Miller, 1991]). These pairwise alignments are then refined by checking consistency when a third sequence is added. T-COFFEE does not weigh sequences, but rather pairs of aligned residues – aligned residues could

be regarded as constraints in this scheme. The distance matrix and a guide tree are built from these refined alignments. Creating an optimal alignment satisfying these weighted constraints is an NP-complete problem. Consequently the authors had to use a heuristic approach that uses at least some of these constraints.

3. 3. 3. MUSCLE

MUSCLE [Edgar, 2004] (http://www.drive5.com/muscle/) is a fast and fairly accurate method. It first estimates a distance matrix, which it uses to produce a quick alignment. This alignment is then used to improve the original distance matrix and construct a better alignment. MUSCLE then divides the sequences into two groups, realigns them and if the resulting alignment is better, the original one is replaced – see Figure 4 from [Edgar, 2004].

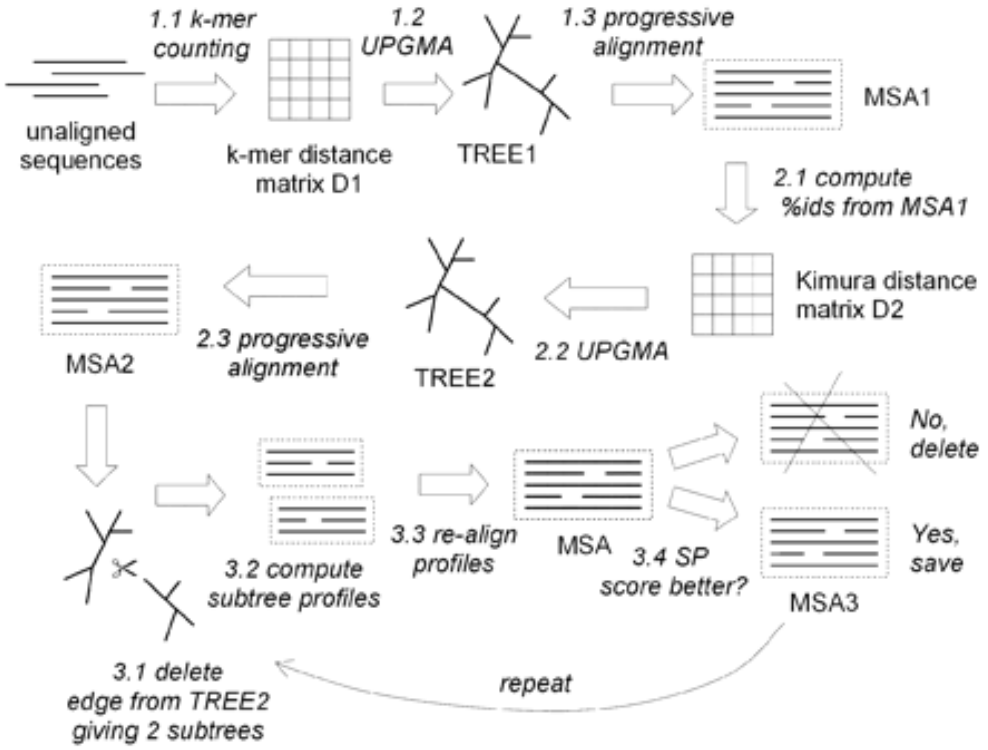

Figure 4 MUSCLE algorithm

Sequence of steps taken by the MUSCLE algorithm: using guide tree to calculate rough MSA, identity ratios between sequences are used to build a refined distance matrix which in turn is used to build another tree. This tree is used to actually guide the progressive alignment step and the algorithm iteratively modifies it to obtain better results.

3. 3. 4. Probcons

Probcons [Do et al., 2005] (http://probcons.stanford.edu/) is a probabilistic consistency-based method. First it calculates probabilities of aligning a particular pair of residues for all the pairs of sequences and for all the residues. Then it re-estimates this probability by aligning the residues from this pair of residues with residues from a third sequence – similarly to the refinement in T-COFFEE. This re-estimation should improve the precision of the probability distribution and it can

be repeated. The guide tree is then constructed with expected values from these distributions used as distances between sequences, and with weighted averages used as distances between aligned blocks of sequences. A scoring function for the alignment is then derived from the probabilities.

Probcons achieves very good results and is generally very fast.

3. 3. 5. PROMALS

PROMALS3D [Pei et al., 2008] (http://prodata.swmed.edu/promals3d/promals3d.php) is claimed by its authors to be the most accurate algorithm for multiple sequence alignment. It searches for homologues with PSI-BLAST [Altschul et al., 1997] and uses the results to predict secondary structure with PSIPRED as well. The alignment algorithm has two stages. In the first stage, sequences are aligned in a pairwise fashion and only those sequences that have a similarity above a certain threshold are aligned together. This is used to process the "easy" sequences quickly and yields a set of relatively divergent sequence groups. From each such group, a representative is selected which is used as a query to PSI-BLAST. Checkpoint files from PSI-BLAST are used for secondary structure prediction with PSIPRED [Jones, 1999], and to generate a “profile-profile HMM”. This HMM is similar to HMMs for pairwise alignment of sequences [Durbin et al., 1996] with a match state emitting two positions from the profiles (instead of a single residue as in the profile HMM or two residues in the HMM for pairwise alignment). This produces posterior probabilities of residue matches (similar to those from Probcons). These are again used to generate a scoring function for aligning.

PROMALS uses special representation of the core of the alignment consisting of two profile components (frequency vectors: effective and target frequencies) based on position specific independent counts [Sunyaev et al., 1999; Pei and Grishin, 2001] weighting of sequences.

3. 3. 6. Dialign

Local alignment program DiAlign2 [Morgenstern, 1999] does not use gap penalties. It breaks up sequences into smaller blocks, aligns all of them and uses the highest scoring ones to compile a multiple sequence alignment while leaving the residues between the blocks unaligned.

3. 3. 7. PRALINE

In this section we will describe PRALINE. As the focus of the thesis is on extending it, this section will be more detailed than the previous ones.

PRALINE [Pirovano et al., 2008] (http://ibi.vu.nl/programs/pralinewww) is a multiple sequence alignment suite first described in [Heringa, 1999]. The multiple sequence alignment progressive protocol is enhanced in PRALINE by several additions aimed at improving consistency and avoiding errors in the alignment, especially in the initial stage. We will describe the following additions in some detail: pre-profiles, no guide tree, position dependent scoring matrices and gap penalties, sequence weighting, secondary structure and transmembrane prediction integration and iterating with consistency scores.

PRALINE performs a pre-processing step before starting the alignment routine – it aligns each sequence (the query sequence) to all the other sequences and if the resulting score is higher than a threshold it adds the sequence to a "stack" from which a pre-profile is created for the query sequence. Columns where the query sequence would have a gap are not considered for the pre-profile. Pre-profile for the sequence is then generated according to the relative frequencies of the amino acids. Gaps in the alignment (except for the query sequence) then contribute to the calculation of the position-dependent gap penalty, which is a part of the pre-profile.

The main difference in regard to other progressive alignment routines is that PRALINE does not use an explicit guide tree. It computes a distance matrix of all the pre-profiles in the beginning and then after each step when two pre-profiles are

joined, it realigns this new pre-profile with the rest. This leads to a rather long running time, a common complaint about PRALINE [Xiong, 2006].

The long running time is also caused by secondary structure prediction as some of the predictors run a PSI-BLAST search. These searches were later [Simossis & Heringa, 2005] adopted into the protocol itself – in PRALINE-PSI, PSI-BLAST results are used to build the pre-profiles. Predicted secondary structure information is used in PRALINE in a conservative way – when aligning two sequences, the score for a match of a_i and b_j can be taken from a substitution matrix for helix (or sheet or coil; matrices from [Lüthy et al., 1991]) if both positions i and j in the respective sequences are predicted as helix (resp. sheet or coil). This contrasts with the method SPEM [Zhou and Zhou, 2005] which disallows gaps in helix and coil regions completely. The transmembrane prediction, which was added later [Pirovano et al., 2008], is implemented in a similar way. This prevents overreliance on prediction results; as accuracies of even the best predictors are around 80% it is a necessary measure. PRALINE can use PSIPRED [Jones, 1999], PORTER [Pollastri and McLysaght, 2005], YASPIN [Lin et al., 2005] or JPRED[Cole et al., 2008] as secondary structure predictor.

The use of pre-profiles allows scoring the consistency of a given residue from any sequence based on how often this particular residue appears in pre-profiles of the other sequences. This further allows iterating the whole algorithm using the weights obtained when scoring the pre-profiles to ensure the core regions of the alignment are aligned correctly. These consistency weights then give a general idea about the quality of the alignment at a given column [Notredame, 2002]. The fact that PRALINE iterates to improve the alignment leads some authors to include it in iterative alignment methods [Pevsner, 2009], others classify it as an iterative progressive algorithm [Caroll, 2008].

In [Heringa, 2002] pre-profile construction was further improved by offering the choice to use local or local-global pre-processing. The original pre-processing performed a global alignment of the sequences and these two newer strategies

include only better scoring fragments (local) or best scoring fragments, which improve the global alignment of the sequences (local-global double dynamic programming). These protocols (as well as the new protocols we implemented and a host of other options and tweaks) can be used through specific command-line options.

4. HMMs

This chapter contains formal definition of Hidden Markov models (HMMs) and gives a general overview of some important algorithms that are used when working with them as well as specific features of profile HMMs used for representation of the alignment core.

HMMs are used in sequence analysis [Churchill, 1989; Krogh, 1994; Durbin et al., 1996; Eddy, 1998; Finn et al., 2009], in other areas of bioinformatics (transmembrane topology prediction [Krogh et al.,2001; Sonnhammer et al., 1998], gene prediction [Munch and Krogh, 2006], modeling of DNA sequencing errors [Lottaz et al., 2003], protein secondary structure prediction [Asai et al., 1993], ncRNA identification [Zhang et al., 2006], RNA structural alignment [Yoon et al., 2008], RNA structure prediction and alignment [Harmanci et al., 2007]), speech and image recognition [Xie et al., 2004; Juang and Rabiner 1991], signal processing [Crouse et al., 1998] and other disciplines.

4. 1. Definition and basic description

Hidden Markov models are statistical models used for modeling sequences of data or states which have the Markov property:

Definition A Markov chain is a sequence of random variables $X_1, X_2, \ldots$ where the value of X_i depends only on X_{i-1} for $i>1$ (the Markov property).

A hidden Markov model (HMM) is (adapted from [Rabiner and Juang, 1986]) a quintuple M = (Q, V, A, B, π), where

Q is a finite set of states $\{q_1, .., q_N\}$ for N>1,

V is a finite set of observations $\{v_1, .., v_M\}$ for M>1,

A is a state transition probability distribution $\{a_{i,j}\}$: $a_{i,j}$ = P(HMM will be in state q_j in the next step | HMM is in state q_i) for $i,j \in \{1, .., N\}$,

B is a set of independent emission probability distributions $\{b_j(k)\}$:$b_j(k)$ = P(HMM emits symbol v_k|HMM is in state q_j)} and π is the initial state probability distribution $\{\pi_j\}$: π_j = P(HMM starts in state q_j) for $j \in \{1,..,N\}$. $b_j(k)$ can be 0 for all $k \in \{1,..,M\}$, thus creating a non-emitting state referred to as *silent* or *mute*. An HMM is valid only if it does not contain a directed cycle with non-zero probability consisting of silent states.

HMM starts in time $t = 1$ in one of the states q_{i_1} with $\pi_{i_1}>0$ (for $i_1 \in \{1,..,N\}$), and in each step s randomly transitions to one of the states according to its probability $a_{i_s,i_{s+1}}$, where it emits a symbol $O_t = v_k \in V$ according to the probability $b_{i_s}(k)$ creating an observation sequence $\{O_t\}_{t=1}^{m}$.

It can be seen that the sequence of states q_{i_s} (where $1 \leq s \leq n$) is a Markov chain.

Important problems are:

1. Given an HMM M and observed sequence $\{O_t\}$, find the most likely state trajectory $\{q_{i_s}\}_{s=1}^{n}$ emitting $\{O_t\}_{t=1}^{m}$.

2. Given $\{O_t\}_{t=1}^{m}$, Q and V, find A, B and π, so that $\{O_t\}_{t=1}^{m}$ is the most likely observation sequence emitted by $M = (Q, V, A, B, \pi)$.

3. Given 2 HMMs M and M', find the most likely trajectories in both of them that produce the same most likely common output sequence $\{O_t\}_{t=1}^{m}$.

4. 2. Profile HMM

Profile HMM is an HMM used in sequence analysis to store information about multiple related sequences (typically a family). In profile HMMs, set of observations V is a set of all amino acids and the sequence $\{O_t\}_{t=1}^{m}$ corresponds to a protein sequence that would belong to the family represented by this profile HMM with probability P($\{O_t\}_{t=1}^{m}$). The set of states Q of profile HMMs consists of a start state (which is the only one with non-zero start probability), an end state (which does not allow transitions to other states), and a certain number of groups of states

called nodes. Each node corresponds to a column in the alignment from which this profile HMM was built. They consist of three states: match, insert and delete. Match state outputs one amino acid letter and roughly corresponds to a column in a profile – the emission probabilities are related to probabilities in the profile, an insert state allows inserting arbitrary number of amino acids between this position and the next one and a delete state is used to skip the next position (it does not emit any amino acids). Profile HMM can transition from the start state to one of the states in the first group, match (*M*) and delete (*D*) states transition to one of the states in the next group, an insert state (*I*) can transition to itself or to the match and delete state in the current group. For an example of a profile HMM, see Figure 5.

4. 3. Algorithms used with HMM

Here I present some problems illustrating use of HMMs.

Problem 1, the most likely state trajectory for a given observation sequence, can be solved by the Viterbi algorithm (see for example [Rabiner, 1989])

Input: HMM $M = (Q, V, A, B, \pi)$, $O = \{O_t\}_{t=1}^{m}$ where $O_t \in V$ for $t = 1,..,m$

Output: $T = \{q_{i_s}\}_{s=1}^{n}$ where $q_{i_s} \in Q$ for $s = 1,..,n$ and T is the most likely state trajectory emitting O.

The Viterbi algorithm exploits the independence of states in a Markov Chain. It recursively calculates $v_{i_s}(j)$ – the probability that the character $O_t = v_j \in V$ was emitted by state q_{i_s} in step *s,* from the probabilities that O_{t-1} was emitted by state $q_{i_{s-1}}$ (which is already known from the previous recursive step for all possible states). This can be used with only small modifications even if *M* contains silent states.

Problem 2 – building of the model – is solved by the "maximum a posteriori" (MAP) model construction algorithm (see [Durbin et al., 1996])

Input: Q, V, multiple sequence alignment of K sequences $\boldsymbol{O_k} = \{O_{k,t}\}_{t=1}^{m}$ where $O_{k,t} \in V \cup \{\text{'\_'}\}$ for $t = 1,..,m$, $k = 1,...,K$

Output: A, B, π so that for HMM $M = (Q, V, A, B, \pi)$, the set of sequences $\{\boldsymbol{O_k}\}_{k=1}^{K}$ are the most likely sequences to output.

The MAP model construction algorithm is a type of expectation-maximization (EM) algorithm. This algorithm is guaranteed to find only a local optimum, which should then be tested against real data.

Problem 3 – the most likely common sequence and its sequence of states – corresponds to finding the most probable HMM-HMM 'alignment'. This algorithm was designed by [Lyngsø et al., 1999] and later used in [Söding et al., 2005], and eventually implemented in PRC [Madera, 2008], which we use in the implementation.

Input: HMMs $M = (Q, V, A, B, \pi)$ and $M' = (Q', V', A', B', \pi')$.

Output: 2 aligned sequences of states through which M and M' passed when generating the most likely common sequence of observations.

The output of the PRC program (see [Madera, 2008]) can be formatted in various ways according to the compilation options. Here we use the most restrictive option "**SPACE5**, **VIA_MM"** (described on page 52 in [Madera, 2008].). This variant does not allow alignment of two states unless at least one of them is a match state which makes handling of the output easier.

In the context of profile HMMs Problem 2 corresponds to building and training a profile HMM model, Problem 1 to aligning a query protein sequence to the profile HMM and Problem 3 represents aligning two profile HMMs to each other. The sequence of states in Problem 2 can be thought of as a representation of an

alignment – match states represent a match, delete states represent a gap in the query sequence, and insert states a gap in the profile.

4. 4. PLAN7

HMMER 2.0 introduced an arrangement of states and transitions that allows using the same model for local and global alignments. It is called **PLAN7** as it allows 7 transitions $M \rightarrow M$, $M \rightarrow I$, $M \rightarrow D$, $I \rightarrow I$, $I \rightarrow M$, $D \rightarrow M$, $D \rightarrow D$. Transitions $I \rightarrow D$ and $D \rightarrow I$ are not allowed to simplify the model and disallow alignments where a gap in the first sequence is immediately followed by a gap in the other.

PLAN7 then contains additional states (*B*, *N, E, C, J*) that can be used to enable local alignment and to control algorithm dependent features of the model.

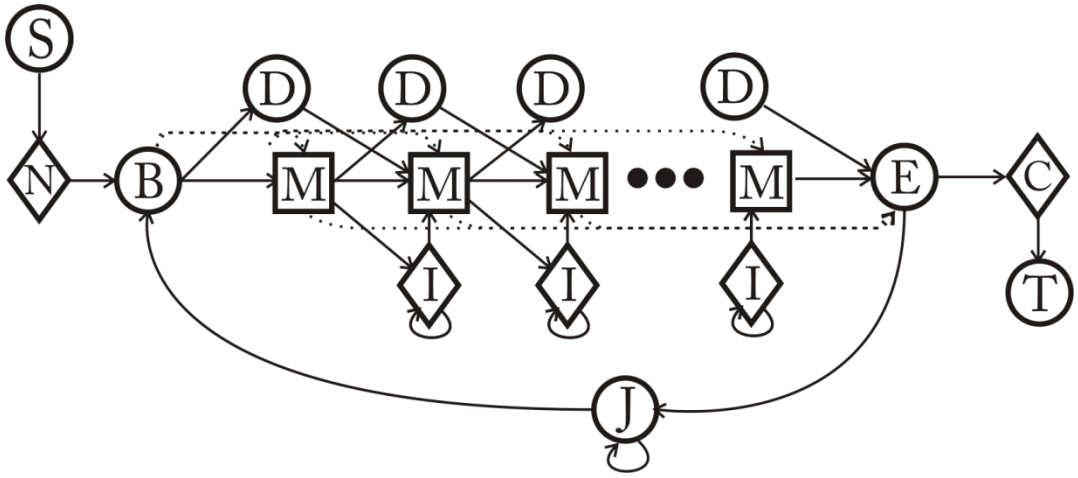

Figure 5 PLAN7

A schema of PLAN7 profile HMM. The state *S* is the only one with non-zero initial probability, *T* terminates the generation of observed sequence. The *N* and *C* states are N resp. C terminal unaligned sequence states and can be used to allow local alignment. Begin (*B*) and end (*E*) states are for entering and exiting the main model which consists of match (*M*), insert (*I*) and delete (*D*) states. If the transition $E \rightarrow J$ has a non-zero probability multihit alignments are allowed, i.e. all local alignments above certain score threshold.

4. 5. Other variants of HMMs

Among new variants of Hidden Markov Models that were recently proposed, the fuzzy HMM [Tran and Wagner, 1999] caught our attention. These models remove the assumption of independence between states (which is unrealistic in protein sequences). They have been successfully employed in sequence alignment [Bidargaddi et al., 2005; Collyda et al., 2006] and have shown improved performance compared to traditional HMMs, especially in the training of the model.

5. Algorithm development

This Chapter provides a description of the work done – profile alignment and new protocols using HMMs that were added into PRALINE.

5. 1. Overview

In this thesis we focused on improvements of the quality of the alignments the PRALINE suite produces. We explored 2 ways to this improvement: allowing to align two or more profiles in addition to sequences (Chapter 5. 2) and replacing the profile representation of the core of the alignment with one based on profile Hidden Markov Models (Chapter 5. 4).

Aligning two profiles instead of a list of sequences can be beneficial to scientists who already have an alignment, which they perhaps hand-tuned with great effort and want to join it with another one or another set of sequences.

The quality of the alignment itself depends heavily on the representation of the alignment. We felt that PRALINE already has a set of great protocols to align sequences however, with a careful look on modern alternatives to sequence profiles its accuracy could be improved.

5. 2. Profile alignment

As a first task and an introduction to the problem domain, alignment of profiles was added to PRALINE as an option PROFPROFALI. With this option PRALINE reads multiple alignment files in MSF or FASTA format, runs a secondary structure prediction for the sequences, and aligns the profiles according to their relative distance – closest first. The rationale for adding it was that researchers often already have a manually tuned alignment and want to "add" other multiple alignments to it, and not many alignment packages allow to do that (the only one that could do it reliably was M-COFFEE [Wallace et al., 2006], although others, like MUSCLE [Edgar, 2004] provide profile-profile alignment option for aligning two profiles).

As some secondary structure prediction methods (PORTER and SSPRO) in PRALINE can accept an alignment instead of a sequence as an input, we considered whether to predict secondary structure for one sequence at a time or predict it from the alignment for the whole profile at once. The former method where the prediction of secondary structure would be based on PSI-BLAST results is computationally expensive and time consuming but should produce superior results as the input alignment is not guaranteed to contain enough information, especially in comparison with PSI-BLAST results for all the sequences. Furthermore, using the alignment as an input would depend too strongly on the quality of the prediction, as there would be only one prediction which itself would be based on the user-supplied alignment. As the general philosophy in the use of secondary structure prediction in PRALINE is to use it only when we can be certain about it, running PSI-BLAST for each sequence in the alignment was chosen. However, secondary structure is assigned to the profile position only when all the sequences have the same prediction at that position. The prediction can be done with any of the predictors supported by PRALINE (see Chapter 3. 3. 7)

5. 3. Use of HMMs in PRALINE

To improve the sensitivity of PRALINE alignment and scoring of profiles, it was decided to use profile HMMs [Durbin et al., 1996] instead of Gribskov's amino acid frequency profiles [Gribskov et al., 1987] for representation of the core of the alignment. As state-of-the-art utilities that could be used for building and comparing HMMs are available we chose to modify hmmbuild from HMMER in version 2.0 [Eddy, 2008] (http://hmmer.janelia.org/) and alignment of two HMMs by PRC [Madera, 2008] (http://supfam.mrc-lmb.cam.ac.uk/PRC/) to hold secondary structure information in the model. The SAM package [Karplus et al., 1998], which provides means of building HMMs like HMMER, was reported to produce better results [Wistrand and Sonnhammer, 2005] but its sources are not available for download, therefore we chose to use HMMER. PRC was chosen as

the source code is freely available and well documented and it has been shown to perform very well [Reid et al., 2007].

PRC and HMMER were then used in PRALINE to perform the actual alignment. As the source code of both of these programs is distributed under GNU General Public License their source codes were easily available. However, the modifications of these programs would have to be made available if PRALINE were to be distributed.

As an intermediate step before the actual implementation of HMM profiles, it was decided to try guiding the alignment by profile similarity computed by scoring HMMs constructed from the pre-profiles (see Chapter 3. 3. 7) against each other, instead of aligning the profiles by the traditional dynamic programming method. We will refer to this method as 'hmmguide'. First, we perform pre-processing (either global, local, local-global or PSI-BLAST; using normal dynamic programming) to obtain pre-profiles of the sequences, use these pre-profiles to build profile-HMMs by hmmbuild (with parameter: '-g'), score similarity of these HMMs by PRC (with parameters '-hits 1 -stop -500 -align prc -mode global-global'), and align (again using normal dynamic programming method) the two pre-profiles which are found to be the closest. These HMMs did not include any information on the secondary structure of the profiles to enable running the experiment quickly without extensive changes to the code of both HMMER and PRC. Using only such simple technique we were able to achieve improvement over standard PRALINE using the local pre-processing protocol, see Chapter 6. 2.

5. 4. Aligning with PRC

These results promised good performance and thus we decided to use HMMs fully in the protocol. The difference between the 'HMMGUIDE' method and this one ('PRCALIGN') is that HMMs are built using not only sequence information, but also the predicted secondary structure and the alignment produced by PRC is actually used to align the two profiles. A number of modifications to both

HMMER and PRC had to be made to accommodate the secondary structure in the HMM model.

The protocol is as follows:

1. Pre-process the sequences (either using PSI-BLAST results or PRALINE pre-processing) to obtain prealignments.
2. Feed these prealignments to HMMER to build HMMs.
3. Following the progressive alignment protocol align these HMMs with PRC and insert gaps into the sequences according to this alignment to obtain the final alignment of the input sequences.

These steps are illustrated in Figure 6.

The step 3 requires a bit more explanation: for 2 HMMs PRC can return a match (a significant alignment between the HMMs). If it does not return a match it is assumed that these two sequences are unalignable. This is not as serious as it looks because PRALINE can just assign a very low score to this combination of HMMs effectively setting their distance in the distance matrix to infinity. Hopefully, there will be pairs which can be aligned normally and from those the one with the highest score will be used. The row representing this joint alignment will then be recalculated in the distance matrix.

If PRC returns a match, PRALINE parses the output for the resulting score (and stores it in the distance matrix) and for the alignment strings. PRC allows multiple output formats for the description of the alignment, we chose 'prc'. This consists of two state trajectories (as described in the problem 3). PRALINE joins the alignments according to these rules:

- Combination of *M* and *M or M* and *D*: a match between alignments.
- Combination of *M* and *I* or *D* and '~': a gap in the second alignment (i.e. the final alignment will include the full column from the first alignment and a column full of gaps from the second).

Thanks to the SPACE5, VIA_MM parameters these were the only combination of states PRALINE has to parse (except for the symmetrical case).

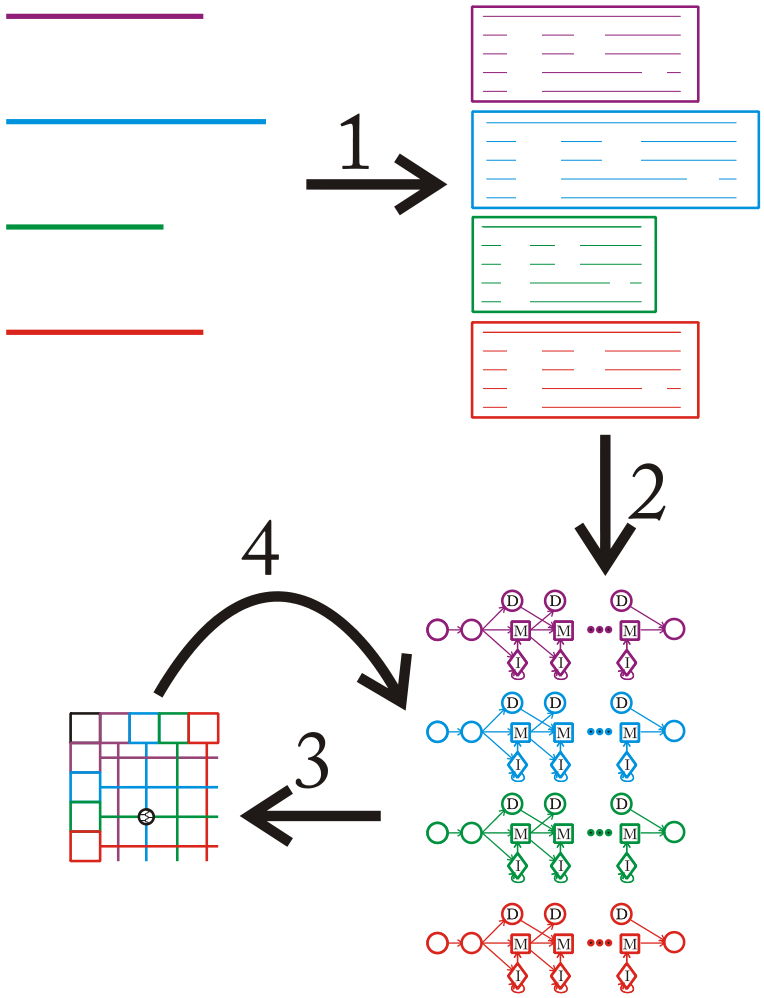

Figure 6 The PRCALIGN PRALINE protocol

In step 1 we perform a local preprocessing – gathering the segments of the sequences that have a high similarity to a query sequence. For each of the query sequences we thus obtain the pre-profile, which we treat as an alignment and feed it in step 2 to hmmbuild. It creates HMMs that are in pairs aligned with PRC in step 3. The best alignment is used to join the corresponding pre-profiles and a new HMM is built from this joint alignment in step 4. The steps 3 and 4 repeat until there is only one block: the result.

5. 5. HMMER modifications

The modification of HMM building in HMMER consisted mainly of extending the alphabet used from 20 amino acids to 60 symbols: amino acid x secondary structure (helix, strand or coil) combinations. This presented some challenges as even though the HMMER User's Guide states that changing alphabet is easy and it has been done, we have found only one published instance of such modifications [Bernsel et al. 2008] and the authors have used only two symbols in the alphabet – extending it beyond 20 caused some difficulties.

Our extended alphabet was represented by 60 consecutive characters from the ASCII table starting with the exclamation mark character '!' (ASCII code 33) and ending with ']' (ASCII code 93). '!' was chosen to fit the whole alphabet between a space and an underscore. The underscore ('_', ASCII code 95) was used for gaps to avoid problems with the MSF format (that was used as an input for hmmbuild), which would arise if a space was used.

When HMMER encounters a column consisting of mostly gaps it does not create a match state but an insert state. This could derail PRALINE when it is parsing the state trajectories from PRC, thus it reads the Stockholm alignment (SXL) output of HMMER, which contains annotated alignment including states assigned to particular columns, and stores indices of those that were assigned to insert states (thus effectively removing them from the HMM). When PRALINE is combining the alignments and it reaches such a column it automatically treats it as a gap in the other alignment. When it encounters situation where both columns were assigned insert states it considers it a match of these columns.

5. 6. Priors

The HMMER does not use scoring matrices [Henikoff and Henikoff, 1992; Dayhoff, 1978; Lüthy, 1991] for assessing similarity between sequences and building a model, instead it uses a single probability distribution called Dirichlet mixtures (DM) [Brown et al. 1993] [Sjölander et al., 1996] – a mixture of Dirichlet distributions. DMs have been proven to achieve the best performance in a related

field of statistical text modelling [Yamamoto and Sadamitsu, 2005]. This distribution is dependent on multiple parameters (referred to as priors): weight coefficients of the mixtures, the number of mixtures, and the parameters of the underlying Dirichlet distribution (one for each mixture and alphabet symbol) [Sjölander et al., 1996; Becker et al., 2001].

As the modified hmmbuild creates HMMs over a different alphabet, it needed a different prior. The quality of HMMs built from the input multiple sequence alignment strongly depends on the quality of the prior used [Wistrand and Sonnhammer, 2004]. We obtained the prior for the extended alphabet by training DM over vector counts (see [Sjölander et al., 1996]) calculated from the HOMSTRAD database [Mizuguchi et al., 1998] of multiple sequence alignments with structural information.

Every column in each alignment of the latest release of the HOMSTRAD database was first assigned a secondary structure (a coil, a strand or a helix). Then they were divided into 'match states' and 'insert states' datasets according to a simple heuristic: a column with a gap was assigned to the 'insert states' so that both datasets contained a significant number of vector counts to estimate DM parameters properly. We are aware that hmmbuild can assign an insert state to a column even if it does not contain any gaps at all and the presence of a gap does not automatically mean that it will be assigned an insert state. However, we believe that this approach presents a reasonable tradeoff between accuracy and speed of counting. For each such column we created a vector of length 60 with the counts of occurrences of each symbol of the extended alphabet in this particular column. The sequences were weighted according to [Henikoff and Henikoff, 1994], thus the vector counts were not necessarily integer. The actual training was performed on the vector counts by the 'dm' utility [Mochihashi, 2006].

The secondary structure was assigned by running DSSP and applying a standard mapping ('H' from DSSP as helix, 'E' as strand and everything else as coil [Pollastri et al., 2002]). HOMSTRAD database from 9th November 2009 was

used. Only the alignments where the sequence from PDB file could be globally aligned with the sequence from the alignment (with average score higher than 3 per residue) were used, yielding 3 427 sequences in 1 029 alignments with 254 738 columns. 49 221 of these columns were regarded as insert and 205 517 as match states.

An alternative way of producing vector counts was to use hmmbuild itself to read the alignments and assign insert and match states with the algorithm that is actually used when building the HMMs. However, for this a prior for the extended alphabet is needed, thus making it unusable for the first prior estimation. Nevertheless, we tried to use this method for improving the quality of the prior after adding the prior we obtained in the previous steps into hmmbuild. Unfortunately, in this way only 478 columns were assigned to the insert dataset. We believe it is not possible to properly train the model with at least 60 parameters on such a small sample, thus we kept the first prior.

The vector counts served as input for the 'dm' utility for estimating Dirichlet mixture parameters. The 'dm' utility requires the number of mixtures as a parameter, thus multiple training runs with different values to achieve a reasonable tradeoff between model precision and complexity were performed. Perplexity

$$PPL = 2^{\frac{\sum \log(q(x_i))}{N}},$$

where $q(x_i)$ is the probability of vector count x_i in the model and N is the number of vector counts, was reported by 'dm' and was thus used as a measure to decide on the number of mixtures. It was recorded for models using 9, 18, 27 and 35 mixtures for the match emission prior and 3, 6 and 9 mixtures for the insert emission prior, see Figure 7 and Figure 8. These numbers were chosen arbitrarily. As estimating the priors took about a day on our machine and the difference in perplexity between the worst one and the best one was 0.2%, we did not continue looking for the best prior.

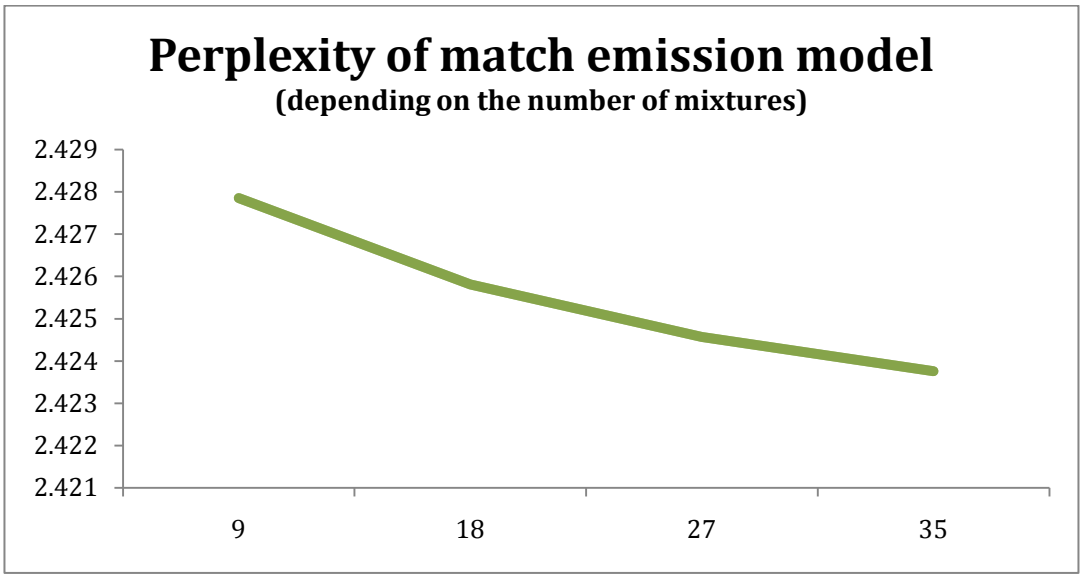

Figure 7 Match emission model perplexity

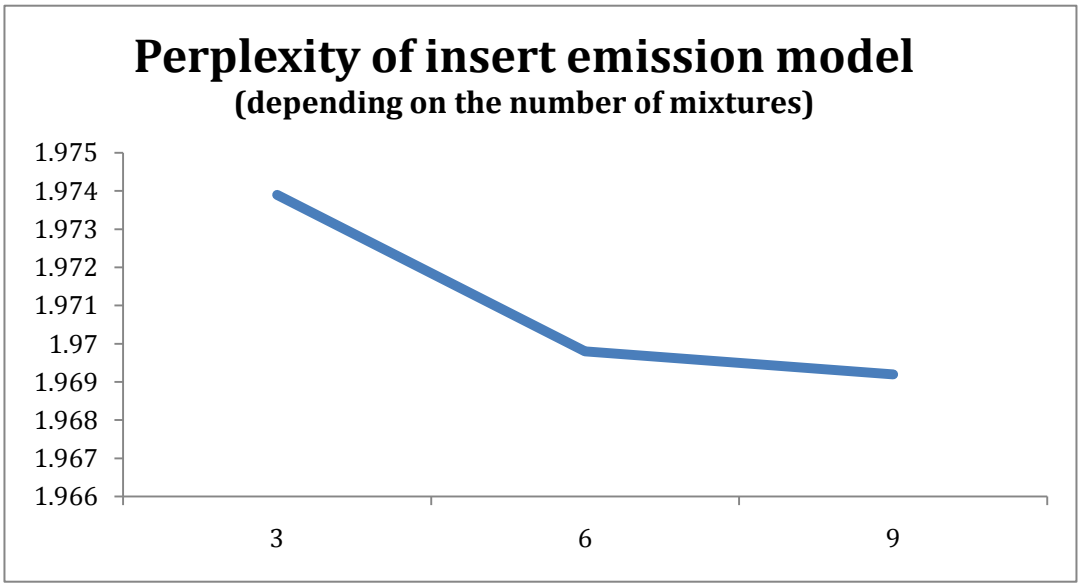

Figure 8 Insert emission model perplexity

These figures present results of prior estimation in terms of the model perplexity depending on the number of mixtures.

A model with 27 mixtures was chosen for the match emission prior and with 6 mixtures for the insert emission priors. 27 was chosen for the match emission prior because the original prior used in hmmbuild used 9 mixtures and the new model has 3 times more states, thus a 3 times increase in the number of mixtures seems

like a logical choice and the PPL does not suggest it is a wrong choice. 6 mixtures for the insert emission were chosen as the increase from 3 to 6 seems to improve the model much more than a step from 6 to 9. Further research into prior estimation is needed, possibly based on benchmarking the priors on an alignment database as this method does not test the priors in the way they are used in hmmbuild. These priors were then hard-coded into HMMER source files along the original ones. The transition probabilities were kept unchanged. However, we believe that using a different set of transition probabilities for different secondary structure regions can be strongly beneficial. It was not attempted as it would require additional changes in HMMER source code and a rigorous approach to estimating them would be required.

Another set of required parameters was the background frequencies of the symbols of the extended alphabet. This was also obtained from the 3 427 sequences from HOMSTRAD database. The frequencies and ratios of secondary structure regions can be seen on Figure 9 and Figure 10, respectively. The amino acid frequencies show a good agreement with the current knowledge about the biases of certain amino acids towards secondary structure – for example that amino acids M, A, L, E and K are expected in helix regions.

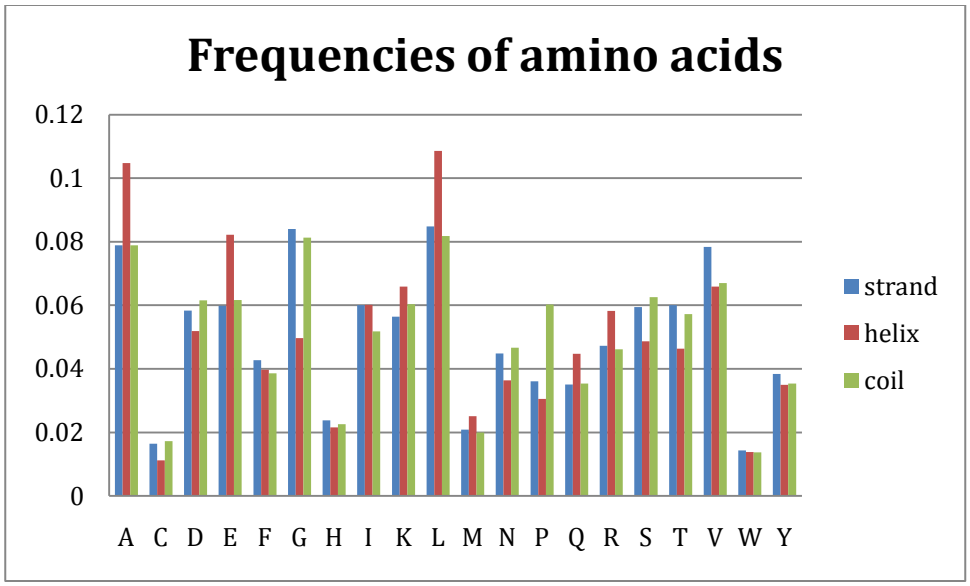

Figure 9 Amino acid frequencies according to secondary structure

The frequencies were normalized (for each secondary structure they were divided by the ratio of this secondary structure) so that bias of some amino acids towards particular secondary structure can be observed.

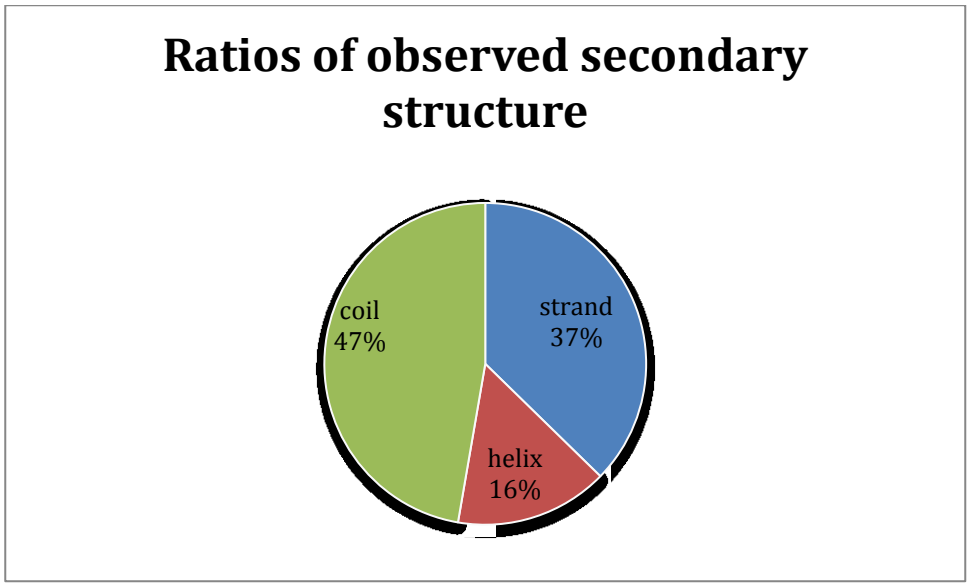

Figure 10 Secondary structure ratios

6. Results

This chapter presents the results of the experiments described earlier. First the results of the profile alignment are presented and then behavior of the two new protocols is discussed.

Chapter 6. 1 discusses the results of the PROFPROFALI option for aligning profiles. Chapter 6. 2 summarizes the results of the HMMGUIDE and PRCALIGN protocols on the BAliBASE 3.0 benchmark.

6. 1. Results of PROFPROFALI

The usefulness of the option PROFPROFALI was tested using sample PFAM [Eddy and Bateman, 2009] seed alignments of superfamily Ras. PFAM is a database of alignments for biologically related sequences. The Ras family was used as it is well studied in literature and the seed alignment contains 4 subfamilies suitable for use as profiles for alignment. The test was performed by comparing the original PFAM alignment with three alignments produced by PRALINE with default options using PSIPRED secondary structure prediction:

1) Aligning all sequences at the same time.

2) Alignment produced by taking blocks from PFAM alignment corresponding to different families and aligning them using the new PROFPROFALI option.

3) Aligning first the sequences from the families and then aligning the resulting MSAs together using PROFPROFALI.

The first method represents the "old way", the second one is a reference how close can the results from PRALINE get to the original alignment and the last method was used to see whether PRALINE would perform better just with information on family membership. Furthermore, Probcons [Do et al. 2005] and Muscle [Edgar 2004] were used as they were readily available and produce

alignments of a quality comparable to PRALINE. However, only the simple test 1) was performed with them.

No large scale testing was done as dividing sequences into subfamilies and locating subfamily specific residues and functional sites have to be done manually. It is in fact quite difficult to find superfamilies described well enough to perform such studies.

The sample PFAM alignment of the Ras superfamily was divided to families Ran, Ras, Rho and Rab. The quality of the alignment of functional and family specific position was assessed according to [Wennerberg et al. 2005; Bourne et al. 1991] and [Pacold et el. 2000]. The position numbers mentioned below are from the PFAM alignment.

[Bourne et al. 1991] mentions three positions which distinguish families of Ras from each other: position 33, 41 and 68. The PFAM alignment does not align the residues mentioned in this paper in one column; Rho family should have a phenylalanin at position 33 but has a gap there instead. All alignments except 1) exhibit the pattern from the paper however. In other columns all methods resulted in the same alignment as PFAM, with the exception of method 1) at position 41. Method 1) placed a gap there shifting it by 1 in families Ras and Rho. Position 68 did not result in any significant differences. [Pacold et al. 2000] mentions position 85 (labeled there as *arg73*) as important for the function of Ras superfamily as it is in a loop region close to the functional site. All programs manage to align it correctly, but methods 1) and 3) place gaps directly next to it, which was judged to be wrong as they are so close to a superfamily-wide functional site.

[Stenmark and Olkonen 2001] lists functional sites in the whole superfamily and some specific to Rab family. Sites G1, G2, G3, PM1 and PM3 are aligned in the same correct way in the results of all the methods. The dissimilarity of the sequence RHO3_YEAST confused methods 1) and 3) at PM2 as they shifted the functional site by a few residues.

We present also the sum-of-pairs score (SP) and column conservation score (TC) when compared with the PFAM alignment:

	SP	TC
Muscle	0.957	0.78
Probcons	0.959	0.75
PRALINE 1)	0.946	0.64
PRALINE 2)	0.969	0.84
PRALINE 3)	0.92	0.63

Table 1 Sum-of-pairs score (SP) and column conservation (TC)

Unsurprisingly method 2) (aligning parts of the PFAM alignment) was the best method both in SP and TC scores. However, method 1) (aligning all the sequences at the same time) outperformed 3) (aligning subfamilies first). PRALINE apparently aligned the sequences in a better order in method 1) than when using the order suggested by the subfamilies.

Overall all methods produce alignment that can be used to distinguish the families. Moreover, the results show slightly better results from 1) compared to 3) caused probably by the ability of PRALINE to perform the alignment without actually generating a guide tree and always aligning closest sequences/profiles first instead. When PRALINE is supplied with the information about alignment in the families (method 2) it is vastly superior to MUSCLE and Probcons.

6. 2. Results of HMMGUIDE and PRCALIGN

Both PRCALIGN and HMMGUIDE were tested using BAliBASE 3.0 [Thompson et al., 2005] reference sets 1–5 (denoted RV11, RV12, RV20, RV30, RV40 and RV50). This test suite contains 217 alignments. Median of number of sequences in one alignment is 21 and maximum 142. Altogether 7 methods were compared:

1. MUSCLE v 3.7 with default settings,
2. PRALINE without modifications,
3. PRALINE with HMMGUIDE parameter,
4. PRALINE with PRCALIGN parameter,
5. PRALINE without modifications with secondary structure prediction,
6. PRALINE with HMMGUIDE parameter with secondary structure prediction, and
7. PRALINE with PRCALIGN parameter with secondary structure prediction.

PRALINE was always run with local pre-processing (with score threshold 80), default scoring matrices and gap penalties. The secondary structure was predicted using PSIPRED [Jones, 1999] running over 'nr' BLAST database. These alignments were scored using the *bali_score_zeus* comparison tool from BAliBASE benchmark which produces both sum-of-pairs (SP) and column conservation (TC) scores. Means were calculated for every method (see Figure 13).

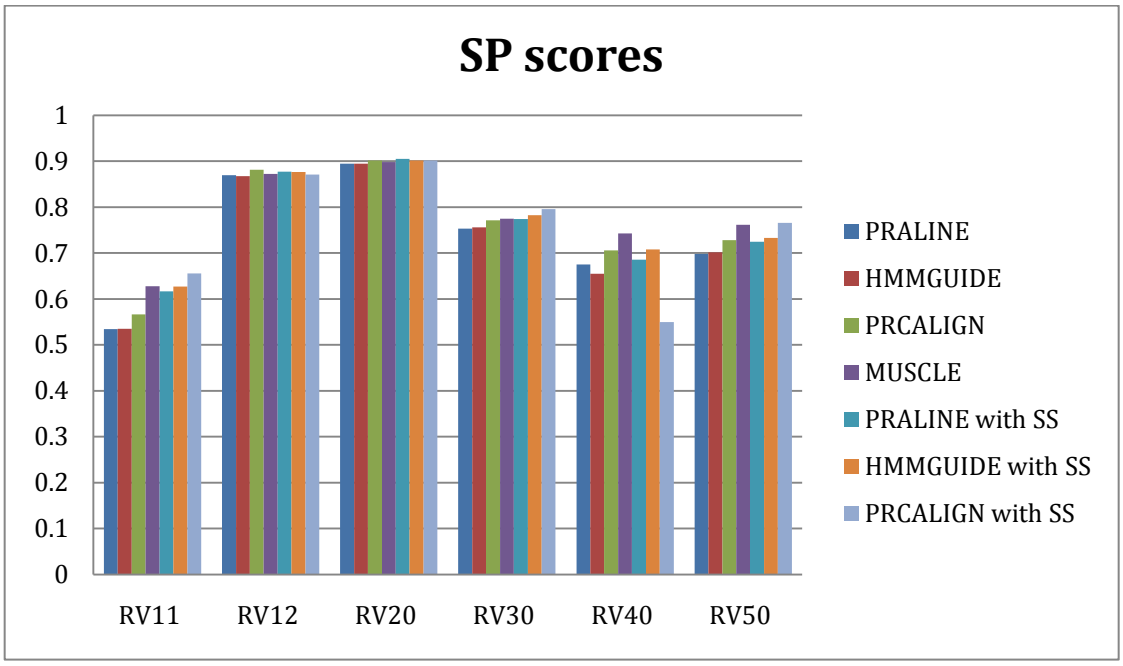

Figure 11 SP scores

Figure presents average sum-of-pairs scores over different reference sets of the BAliBASE benchmark. The subsets RV11 and RV12 together form the reference set 1, but they are separated in the graph as they clearly represent significantly different tasks. The difference between methods was perhaps the most pronounced in RV11, where PRCALIGN with secondary structure was 20% better than classical PRALINE and cca 5% better than classical PRALINE with secondary structure. The reference set 4 (RV40) caused some problems (see Chapter 6. 3, if a method did not finish the run the alignment was ignored when computing the averages for this method) that affected the results. However, the overall conclusion that new methods produce better results than classical PRALINE is clearly visible.

The coloring in this graph is used in all other graphs presenting the results of different methods as well.

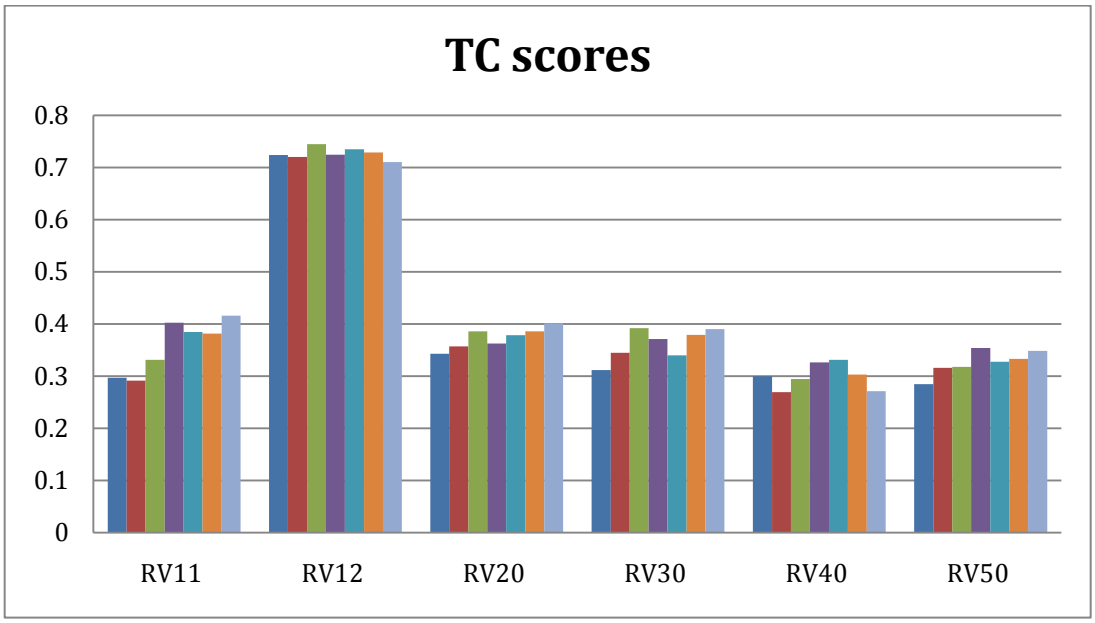

Figure 12 TC scores

This plot presents the average total column conservation scores for reference sets for the BAliBASE benchmark. As on previous figure, the results for set RV40 show that the PRCALIGN method encountered severe problems. Colors and order of different methods is the same as in the previous graph.

The protocol PRCALIGN has not produced a useful alignment due to problems with the protocol itself (see Chapter 6. 3 for discussion of the problems). The alignments with incomplete results were removed from the dataset resulting in 198 alignments out of 217. All other methods correctly aligned all 217 alignments.

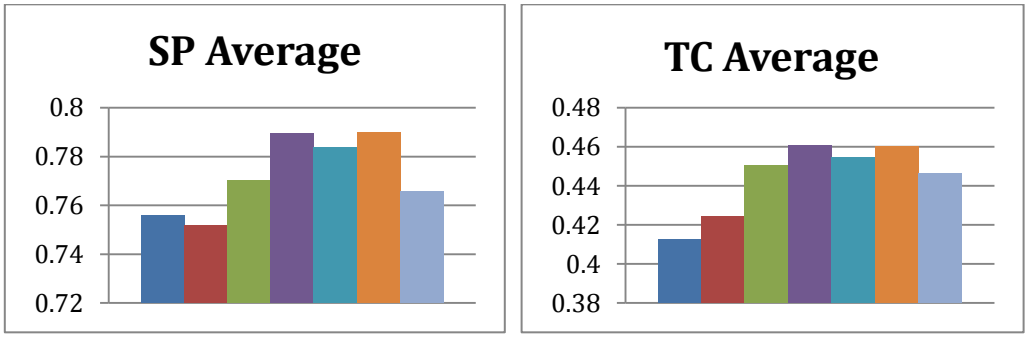

Figure 13 SP and TC averages

These plots show average sum-of-pairs and column conservation scores for the whole BAliBASE benchmark. They were cropped (the *y* axis does not start at 0) so that the differences can be seen easily. The PRCALIGN without secondary structure represents the biggest improvement in quality over the classical PRALINE (8%), but problems in alignment prevented the PRCALIGN method from reaching the full potential when using the secondary structure prediction. The HMMGUIDE probably does not perform much better than the original PRALINE.

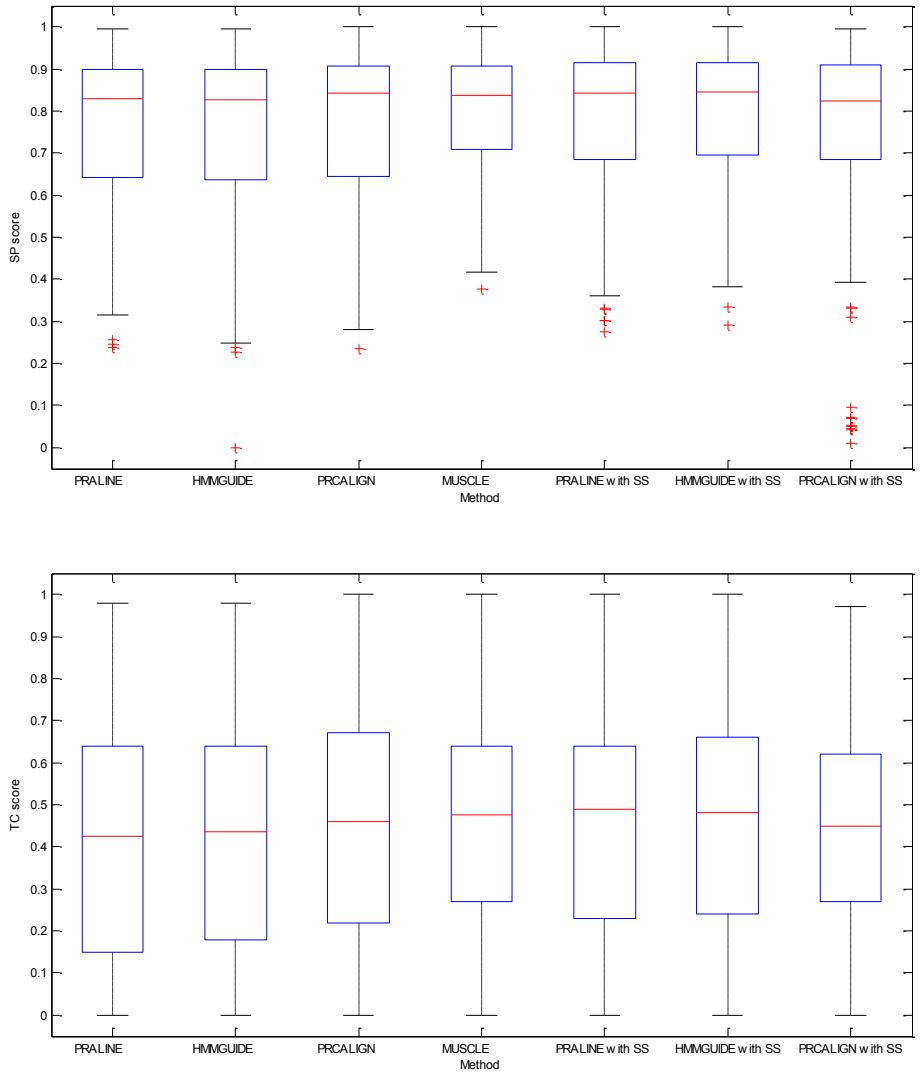

Figure 14 Boxplots of SP and TC scores

Boxplots of combined results for the alignment scores where all methods finished. The central mark is the median, the blue box extends from 25th to 75th percentile, the black line ends at the most extreme data point or at 1.5 multiple of 25th to 75th percentile distance from the blue box and the red crosses represent outliers. Notable features include numerous outliers close to zero for the method PRCALIGN with secondary structure in TC scores and much smaller variance of results in TC scores for MUSCLE compared to any PRALINE method. The improvement in quality is obvious for the new protocols without secondary structure, but questionable when using secondary structure.

As the absolute differences between the methods over the whole BAliBASE dataset were rather small (about 5% in both SP and TC scores, see and Figure 14), we decided to use ranking to visualize and test the differences. To simplify the data handling we used only TC scores for additional visualizations as they are considered more relevant [Waterman and Byers, 1985]. The difference between SP and TC scores can be probably explained by different alignment of columns with low conservation.

When testing whether the new protocols were better than the classical PRALINE, we considered which method (of the two) produced better results. We thus obtained a set binary (is/is not better) values. These were tested using one sided binomial sign test [Sheskin, 2003] to assess the statistical significance of the differences between different methods. This method was used because it is nonparametric, robust and takes advantage of the fact that the data are paired. However, it does not take into account the size of the difference in score between the methods, only whether it produced better results. The p-values of the test (that is the probability that the proportion of observations of the method being better than the classical PRALINE happening just by chance) are tabulated in Table 2.

	without SS	with SS
HMMGUIDE	**0,017**	**5,00E–04**
PRCALIGN	**1,65E–13**	0,5

Table 2 P-values

The table presents p-values of the test that the method (in the first column) is produces alignment which is closer to the one from BAliBASE benchmark than the classical PRALINE.

Thus on 95% significance level we can conclude that HMMGUIDE is better than PRALINE with or without secondary structure and PRCALIGN is conclusively better than PRALINE only when the secondary structure prediction is not used.

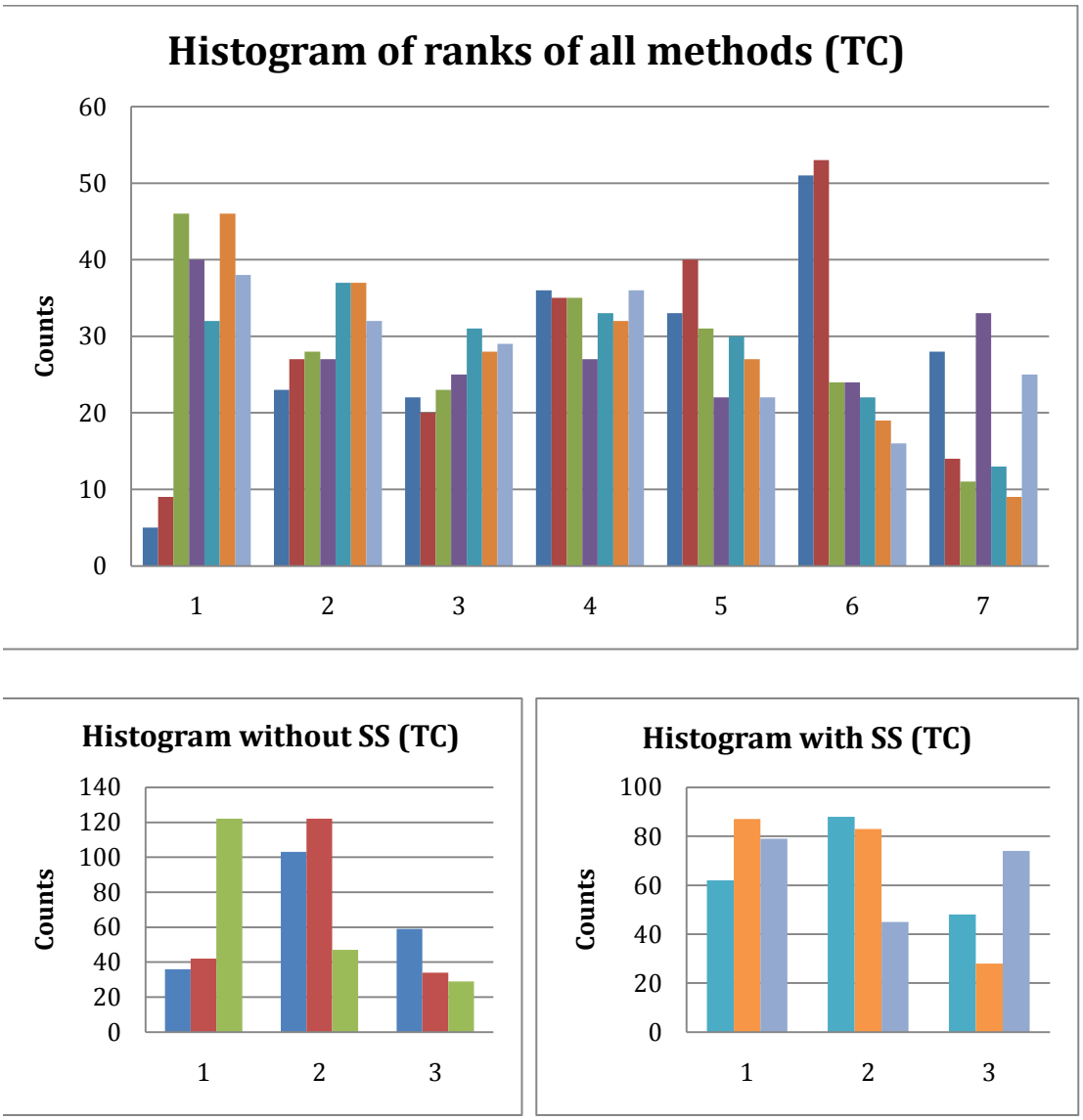

Figure 15 Histograms

This plot shows how often was each method ranked by column conservation score at that particular position. I.e. the first bars in the graph labeled 1 show how often each method was the best for an alignment. For example method PRCALIGN without secondary structure was the best method 46 times (out of 198). The upper graph shows overall ranks and graphs below show results when only PRALINE with or without secondary structure were considered. These graphs show very good results for PRCALIGN without secondary structure (it was the best method without secondary structure for 122 times). This graph uses the same colors and order of the methods as in Figure 11 SP scores

To get a detailed view of the differences between the methods we divided the methods to groups with and without the use of predicted secondary structure and for each BAliBASE alignment ranked the scores the methods achieved. Histograms of the ranks can be seen on Figure 15. Without secondary structure PRCALIGN (the green, 3rd column) is clearly the best method however problems in alignment sent it to second place when using secondary structure (the blue, last column).

6. 3. Problems in aligning

In order to describe the problems we faced we will first review the PRCALIGN with secondary structure prediction protocol where most of the errors occurred. The protocol started with a local pre-processing step which created the pre-profiles from which HMMs were built. This step was needed as HMMER would not produce relevant HMMs from a single sequence itself. However, sequences that were added to the pre-profile were carried along to later stages which resulted in huge pre-profile alignment files (1000 sequences with 2000 alignment columns). These alignments were mostly filled with gaps thus producing very low quality HMMs which then scored badly against other HMMs in the group. PRC would often even refuse to align them altogether. We managed to force it to align all HMMs by setting the STOP parameter to –500 (the default is –2) while setting HITS to 1 so that it does not enter a "near-infinite" loop as recommended by the README file of PRC.

However, we soon encountered another problem: HMMER was not able to construct an HMM from the alignment. It either assigned an insert state to all the columns (which is relatively reasonable as they were at least 50% gaps) or did not find any other match states than the first and the last – thus producing a B→E transition which is forbidden in PLAN7. We were not able to overcome this problem as it is inherent to the complex alignments that we obtained through the

use of local pre-processing. Identifiers of the alignments which failed to produce results for PRCALIGN with secondary structure prediction can be seen in Table 3.

RV40.BB40002
RV40.BB40004
RV40.BB40011
RV40.BB40013
RV40.BB40015
RV40.BB40016
RV40.BB40017
RV40.BB40022
RV40.BB40023
RV40.BB40027
RV40.BB40037
RV40.BB40038
RV40.BB40039
RV40.BB40040
RV40.BB40041
RV40.BB40047
RV40.BB40049
RV50.BBS50003
RV50.BBS50011

Table 3 Failed alignments

List of identifiers of alignments for which PRCALIGN with secondary structure failed to produce results. The prefix RV40 indicates it comes from reference set 4 and RV50 reference set 5 of BAliBASE benchmark.

Reasons why the local pre-processing resulted in such complex alignments might include: bias towards secondary structure conservation in the priors, extended alphabet, "once a gap, always a gap" and the complexity of set 4 of BAliBASE. We describe these causes and their possible corrections in the following paragraphs.

When we prepared the data for training the priors we assigned a single secondary structure to the whole column of an alignment from HOMSTRAD. Thus, we trained the prior only on columns with total secondary structure conservation. Total conservation of secondary structure in a column is however very unlikely in real life alignments. This could be alleviated by randomly changing the secondary structure for single residues in a column of HOMSTRAD alignment when preparing the vector counts. Yet, it would require significant additional effort and further testing, consequently it was not attempted.

The extension of the alphabet from 20 to 60 symbols significantly decreased chances of randomly aligning two matching residues and getting a completely conserved column from the alignment. With the accuracy of secondary structure prediction around 80% we can expect to have a wrong secondary structure of 1 residue in a column of 5 sequences. Lower conservation means HMMER assigns an insert state to this column rather than a match state. The more columns are marked as insert states, the less information the HMM contains.

The number of gaps in the alignment grows also due to the "once a gap, always a gap" principle [Feng and Doolittle, 1987]. The local pre-processing cannot insert gaps into the main sequence (to protect against this), however it can and does insert gaps in the other sequence segments. Such a profile can then easily have more than 50% gaps in columns in parts of the sequence for which no analogous segments in other sequences were found. In fact columns with a single non-gap symbol can occur; if the pre-profile then contains 10 sequences we get 90% ratio of gaps. These gaps get carried through the whole protocol always increasing the ratio of gaps in a column.

Finally, the set 4 of BAliBASE benchmark is notoriously difficult to align correctly – the classical PRALINE also struggles with it. It is probably caused by the vast differences between sequences (length ratios can be up to 1:4). This further worsens all previously mentioned problems. The scores of PRCALIGN show that without secondary structure prediction on set 4 and on other sets show

how bad the performance relatively is. The method performs quite well and finishes almost all of the alignments (with the exception of 2 alignments from set 5) however the combination of extended alphabet and set 4 is hard. 17 out of 51 alignments in set 4 did not produce results. At the same time those that did finish produced alignment of unsatisfactory quality. If PRCALIGN with secondary structure prediction performed at least as well as PRCALIGN without secondary structure prediction it would be definitely the best performer (see Figure 14). HMMGUIDE was not hit by these problems as severely, it managed to pass all tests. It is probably caused by the fact that the alignment was actually constructed by PRALINE without the problems that plague the alignment of extended alphabet HMMs.

6. 4. Performance

Performance improvements were not really the main aim of this work; we present here just a general overview of performance considerations.

As the methods used in these protocols are significantly more complex than the original PRALINE the processing time needed is multiple times more. Simple PRALINE without secondary structure prediction took about 6 seconds on a 3.2GHz CPU while the PRCALIGN protocol needed 9 seconds to finish alignment of 15 sequences. However, when considering that the prediction of secondary structure alone takes approximately 3 full minutes for each sequence the difference between the protocols becomes insignificant.

No formal analysis was done but we estimate that the time complexity of the new methods is similar to that of the old method – $O(n^4)$ where n is the size of the input – as the most demanding part of the algorithm is computing the initial distance matrix (i.e. $n(n-1)/2$ comparisons) over sequences and each such comparison has a quadratic time complexity. We assume here that the time to build and compare two HMMs is quadratic which seems reasonable.

7. Conclusion

We implemented a profile alignment method which improved the accuracy of the alignment when user supplied additional information and 2 new protocols which produced significantly (with 95% confidence level) better results on the BAliBASE dataset. Particularly the protocol PRCALIGN without secondary structure represents a major improvement over PRALINE. The goal of improving PRALINE was thus reached. However, the new method has not produced results for some particularly complicated inputs.

We believe that the results show superiority of profile HMMs when compared to ordinary profiles for representation of the alignment core at the cost of increased computational overhead. The main advantage HMMs have over the profiles is a more precise statistical model of the alignment which can capture more nuanced relations between sequences.

8. Further work

We believe even better results could be achieved with further effort put into these protocols. We present some further research that would improve the results obtained here.

8. 1. Finishing BAliBASE

Producing an alignment for all sequences in BAliBASE should have the highest priority. Careful consideration should be given to the methodology of training the priors. Saving vector counts from processing parts of BAliBASE benchmark which produce reasonable alignments and combining them with the ones from HOMSTRAD might produce better results.

Large scale testing and tuning of the parameters of the alignment [Joachims et al., 2006] such as local pre-processing thresholds could be used to simplify using these tools, as the sequences themselves frequently contain all the information necessary for deciding on the value of all parameters.

8. 2. Integration of other predictions

When using tools such as PSI-BLAST the MSA program has a vast amount of information available – the alignments with all homologues sequences from the database can be used to obtain further information about the sequence. Predictions of the secondary structure [Sander and Schneider, 1991; Heringa, 1999; Pei and Grishin, 2008] are known to improve the quality of the sequence alignment. Transmembrane topologies can also be used [Pirovano et al., 2008]. Current state-of-the-art allows several predictions to be made solely from the sequence data. Information on contacts [Kleinjung et al., 2004], solvent accessibility [Chang et al., 2008], motifs describing families [de Castro et al., 2006], post-translational modifications – signal peptides [Dyrløv Bendtsen et al., 2004] and phosphorylation sites [Blom et al., 2004; Blom et al., 1999; Wan et al., 2008] etc. could all be incorporated into calculations for more details about the sequences and about

particular positions. Function tends to be conserved more than structure and structure more than sequence [Illergård et al., 2009], thus insights into the function of a particular segment in aligned sequences can help us align it in an evolutionary meaningful way.

8. 3. Performance

Finally most of the effort is spent on improving the quality of the alignments. However, the speed might be considerably improved as well. Using vector instructions (SIMD Streaming Extensions – SSE on x86), parallelizing main parts of the algorithm for use on a single multi-core CPU or parallelization on a cluster would allow scientists to perform more alignments in a shorter time.

9. List of figures

10. Bibliography

Altschul, S. F.; Gish, W. & Miller, W. (1990), 'Basic local alignment search tool', *J. Mol. Biol* **215**(3), 403-410.

Asai, K.; Hayamizu, S. & Handa, K. (1993), 'Prediction of protein secondary structure by the hidden Markov model', *Bioinformatics* **9**(2), 141.

Barton, G. J. & Sternberg, M. J. E. (1987), 'A strategy for the rapid multiple alignment of protein sequences: Confidence levels from tertiary structure comparisons', *J. Mol. Biol.* **198**(2), 327-337.

Becker, O. M.; Jr, A. D. M.; Roux, B. & Watanabe, M. (2001), *Computational Biochemistry and Biophysics*, CRC.

Bendtsen, J. D.; Nielsen, H.; von Heijne, G. & Brunak, S. (2004), 'Improved Prediction of Signal Peptides: SignalP 3.0', *J. Mol. Biol.* **340**(4), 783-795.

Bernsel, A.; Viklund, H. & Elofsson, A. (2008), 'Remote homology detection of integral membrane proteins using conserved sequence features', *Proteins: Struct., Funct., Bioinf.* **71**(3), 1387-1399.

Bidargaddi, N.; Chetty, M. & Kamruzzaman, J. (2005), A Fuzzy Viterbi Algorithm for Improved Sequence Alignment and Searching of Proteins'Applications on Evolutionary Computing', pp. 11-21.

Blackshields, G.; Wallace, I. M.; Larkin, M. & Higgins, D. G. (2006), 'Analysis and Comparison of Benchmarks for Multiple Sequence Alignment', *In Silico Biology* **6**(4), 321-339.

Blom, N.; Gammeltoft, S. & Brunak, S. (1999), 'Sequence and structure-based prediction of eukaryotic protein phosphorylation sites', *J. Mol. Biol.* **294**(5), 1351-1362.

Blom, N.; Sicheritz-Pontén, T.; Gupta, R.; Gammeltoft, S. & Brunak, S. (2004), 'Prediction of post-translational glycosylation and phosphorylation of proteins from the amino acid sequence', *Proteomics* **4**(6), 1633-1649.

Bourne, H. R.; Sanders, D. A. & McCormick, F. (1991), 'The GTPase superfamily: conserved structure and molecular mechanism', *Nature* **349**(6305), 117-127.

Bourne, P. E. & Weissig, H. (2003), *Structural Bioinformatics*, Wiley-Liss.

Brocchieri, L. (2001), 'Phylogenetic Inferences from Molecular Sequences:

Review and Critique', *Theoretical Population Biology* **59**(1), 27-40.

Brown, M.; Hughey, R.; Krogh, A.; Mian, I. S.; Sjölander, K. & Haussler, D. (1993), Using Dirichlet mixture priors to derive hidden Markov models for protein families, *in* 'Proceedings of the 3rd International Conference on Intelligent Systems for Molecular Biology'.

Carrillo, H. & Lipman, D. (1988), 'The multiple sequence alignment problem in biology', *SIAM Journal on Applied Mathematics* **48**(5), 1073-1082.

Carroll, H. (2008), *Biologically Relevant Multiple Sequence Alignment*, Vol. PhD. dissertation, Department of Computer Science, Brigham Young University.

de Castro, E.; Sigrist, C. J. A.; Gattiker, A.; Bulliard, V.; Langendijk-Genevaux, P. S.; Gasteiger, E.; Bairoch, A. & Hulo, N. (2006), 'ScanProsite: detection of PROSITE signature matches and ProRule-associated functional and structural residues in proteins', *Nucl. Acids Res.* **34**(suppl-2), W362-365.

Chang, D.; Huang, H.-Y.; Syu, Y.-T. & Wu, C.-P. (2008), 'Real value prediction of protein solvent accessibility using enhanced PSSM features', *BMC Bioinformatics* **9**(Suppl 12), S12.

Churchill, G. (1989), 'Stochastic models for heterogeneous DNA sequences', *Bulletin of Mathematical Biology* **51**(1), 79-94.

Cole, C.; Barber, J. D. & Barton, G. J. (2008), 'The Jpred 3 secondary structure prediction server', *Nucl. Acids Res.* **36**(suppl-2), W197-201.

Collyda, C.; Diplaris, S.; Mitkas, P. A.; Maglaveras, N. & Pappas, C. (2006), 'Fuzzy Hidden Markov Models: a new approach in multiple sequence alignment', *Studies in Health Technology and Informatics* **124**, 99-104.

Corpet, F. (1988), 'Multiple sequence alignment with hierarchical clustering', *Nucl. Acids Res.* **16**(22), 10881.

Crooks, G. E. & Brenner, S. E. (2004), 'Protein secondary structure: entropy, correlations and prediction', *Bioinformatics* **20**(10), 1603-1611.

Crouse, M. S.; Nowak, R. D. & Baraniuk, R. G. (1998), 'Wavelet-based statistical signal processing using hidden Markov models', *IEEE Transactions on Signal Processing* **46**(4), 886-902.

Dayhoff, M. O.; Schwartz, R. M. & Orcutt, B. C. (1978), 'A model of evolutionary change in proteins', *Atlas of protein sequence and structure* **5**(Suppl 3), 345-352.

Do, C. B.; Mahabhashyam, M. S.; Brudno, M. & Batzoglou, S. (2005), 'ProbCons: Probabilistic consistency-based multiple sequence alignment', *Genome Res.* **15**(2), 330-340.

Durbin, R.; Eddy, S. R.; Krogh, A. & Mitchison, G. (1998), *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*, Cambridge University Press.

Eddy, S. (2003), 'HMMER user's guide', *Department of Genetics, Washington University School of Medicine*(1).

Eddy, S. (1998), 'Profile hidden Markov models', *Bioinformatics* **14**(9), 755-763.

Edgar, R. C. (2004), 'MUSCLE: multiple sequence alignment with high accuracy and high throughput', *Nucl. Acids Res.* **32**(5), 1792-1797.

Feng, D. F. & Doolittle, R. F. (1987), 'Progressive sequence alignment as a prerequisite to correct phylogenetic trees', *J. Mol. Evol.* **25**(4), 351-360.

Finn, R. D.; Mistry, J.; Tate, J.; Coggill, P.; Heger, A.; Pollington, J. E.; Gavin, O. L.; Gunasekaran, P.; Ceric, G.; Forslund, K.; Holm, L.; Sonnhammer, E. L. L.; Eddy, S. R. & Bateman, A. (2009), 'The Pfam protein families database', *Nucl. Acids Res.*.

Grasso, C. & Lee, C. (2004), 'Combining partial order alignment and progressive multiple sequence alignment increases alignment speed and scalability to very large alignment problems', *Bioinformatics* **20**(10), 1546-1556.

Gribskov, M.; McLachlan, A. D. & Eisenberg, D. (1987), 'Profile analysis: detection of distantly related proteins', *Proc. Natl. Acad. Sci. U. S. A.* **84**(13), 4355-4358.

Harmanci, A. O.; Sharma, G. & Mathews, D. H. (2007), 'Efficient pairwise RNA structure prediction using probabilistic alignment constraints in Dynalign', *BMC Bioinformatics* **8**(1), 130.

Helal, M.; El-Gindy, H.; Mullin, L. & Gaeta, B. (2008), Parallelizing Optimal Multiple Sequence Alignment by Dynamic Programming, *in* '2008 IEEE International Symposium on Parallel and Distributed Processing with Applications', pp. 669-674.

Henikoff, S. & Henikoff, J. G. (1994), 'Position-based sequence weights', *J. Mol. Biol.* **243**(4), 574-578.

Henikoff, S. & Henikoff, J. G. (1992), 'Amino acid substitution matrices from

protein blocks', *Proc. Natl. Acad. Sci. U. S. A.* **89**(22), 10915-10919.

Heringa, J. (2002), 'Local weighting schemes for protein multiple sequence alignment', *Computers & Chemistry* **26**(5), 459-477.

Heringa, J. (1999), 'Two strategies for sequence comparison: profile-preprocessed and secondary structure-induced multiple alignment', *Computers & Chemistry* **23**(3-4), 341-364.

Hogeweg, P. & Hesper, B. (1984), 'The alignment of sets of sequences and the construction of phyletic trees: an integrated method', *J. Mol. Evol.* **20**(2), 175-186.

Illergård, K.; Ardell, D. & Elofsson, A. (2009), 'Structure is three to ten times more conserved than sequence - A study of structural response in protein cores', *Proteins: Structure, Function, and Bioinformatics* **77**(3), 508, 499.

Joachims, T.; Galor, T. & Elber, R. (2006), 'Learning to Align Sequences: A Maximum-Margin Approach', *New Algorithms for Macromolecular Simulation*, 57-69.

Juang, B. H. & Rabiner, L. R. (1991), 'Hidden Markov models for speech recognition', *Technometrics* **33**(3), 251-272.

Karplus, K.; Barrett, C. & Hughey, R. (1998), 'Hidden Markov models for detecting remote protein homologies', *Bioinformatics* **14**(10), 846-856.

Katoh, K.; ichi Kuma, K.; Toh, H. & Miyata, T. (2005), 'MAFFT version 5: improvement in accuracy of multiple sequence alignment', *Nucl. Acids Res.* **33**(2), 511-518.

Kleinjung, J.; Douglas, N. & Heringa, J. (2002), 'Parallelized multiple alignment', *Bioinformatics* **18**(9), 1270-1271.

Kleinjung, J.; Romein, J.; Lin, K. & Heringa, J. (2004), 'Contact-based sequence alignment', *Nucl. Acids Res.* **32**(8), 2464-2473.

Krogh, A.; Brown, M.; Mian, I. S.; Sjölander, K. & Haussler, D. (1994), 'Hidden Markov Models in Computational Biology : Applications to Protein Modeling', *J. Mol. Biol.* **235**(5), 1501-1531.

Krogh, A.; Larsson, B.; von Heijne, G. & Sonnhammer, E. L. L. (2001), 'Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes', *J. Mol. Biol.* **305**(3), 567-580.

Lin, K.; Simossis, V. A.; Taylor, W. R. & Heringa, J. (2005), 'A simple and fast secondary structure prediction method using hidden neural networks',

Bioinformatics **21**(2), 152.

Lottaz, C.; Iseli, C.; Jongeneel, C. V. & Bucher, P. (2003), 'Modeling sequencing errors by combining Hidden Markov models', *Bioinformatics* **19**(2), 103-112.

Lyngsø, R. B.; Pedersen, C. N. & Nielsen, H. (1999), Metrics and similarity measures for hidden Markov models, *in* 'Proc. Int. Conf. Intell. Syst. Mol. Biol', pp. 178-186.

Lüthy, R.; McLachlan, A. D. & Eisenberg, D. (1991), 'Secondary structure-based profiles: Use of structure-conserving scoring tables in searching protein sequence databases for structural similarities', *Proteins: Structure, Function, and Genetics* **10**(3), 229-239.

Madera, M. (2008), 'Profile Comparer: a program for scoring and aligning profile hidden Markov models', *Bioinformatics* **24**(22), 2630-2631.

Maizel, J. V. & Lenk, R. P. (1981), 'Enhanced graphic matrix analysis of nucleic acid and protein sequences', *Proc. Natl. Acad. Sci. U. S. A.* **78**(12), 7665.

Minka, T. (2003), *Estimating a Dirichlet distribution*, Microsoft Technical Report.

Mizuguchi, K.; Deane, C. M.; Blundell, T. L. & Overington, J. P. (1998), 'HOMSTRAD: A database of protein structure alignments for homologous families', *Protein Sci.* **7**(11), 2469-2471.

Mochihashi, D. (), *dm*, Available at http://chasen.org/~daiti-m/dist/dm/.

Morgenstern, B. (1999), 'DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment', *Bioinformatics* **15**(3), 211-218.

Munch, K. & Krogh, A. (2006), 'Automatic generation of gene finders for eukaryotic species', *BMC Bioinformatics* **7**(1), 263.

Murata, M.; Richardson, J. S. & Sussman, J. L. (1985), 'Simultaneous comparison of three protein sequences', *Proc. Natl. Acad. Sci. U. S. A.* **82**(10), 3073-3077.

Needleman, S. B. & Wunsch, C. D. (1970), 'A general method applicable to the search for similarities in the amino acid sequence of two proteins', *J. Mol. Biol.* **48**(3), 443-453.

Ng, P. C.; Henikoff, J. G. & Henikoff, S. (2000), 'PHAT: a transmembrane-specific substitution matrix', *Bioinformatics* **16**(9), 760-766.

Notredame, C. (2002), 'Recent progresses in multiple sequence alignment: a survey', *Pharmacogenomics* **3**(1).

Notredame, C.; Higgins, D. G. & Heringa, J. (2000), 'T-coffee: a novel method for fast and accurate multiple sequence alignment', *J. Mol. Biol.* **302**(1), 205-217.

von Öhsen, N. & Zimmer, R. (2001), Improving profile-profile alignments via log average scoring, *in* 'Algorithms in Bioinformatics: First International Workshop', Springer Verlag, 11.

Pacold, M. E.; Suire, S.; Perisic, O.; Lara-Gonzalez, S.; Davis, C. T.; Walker, E. H.; Hawkins, P. T.; Stephens, L.; Eccleston, J. F. & Williams, R. L. (2000), 'Crystal Structure and Functional Analysis of Ras Binding to Its Effector Phosphoinositide 3-Kinase γ', *Cell* **103**(6), 931-944.

Pedersen, C. (2001), 'Algorithms in Computational Biology', PhD thesis, University of Aarhus, Denmark.

Pei, J.; Kim, B.-H. & Grishin, N. V. (2008), 'PROMALS3D: a tool for multiple protein sequence and structure alignments', *Nucl. Acids Res.* **36**(7), 2295-2300.

Pevsner, J. (2009), *Bioinformatics and Functional Genomics*, WileyBlackwell.

Pirovano, W.; Feenstra, K. A. & Heringa, J. (2008), 'The meaning of alignment: lessons from structural diversity', *BMC Bioinformatics* **9**(1), 556.

Pirovano, W.; Feenstra, K. A. & Heringa, J. (2008), 'PRALINETM: a strategy for improved multiple alignment of transmembrane proteins', *Bioinformatics* **24**(4), 492-497.

Pollastri, G. & McLysaght, A. (2005), 'Porter: a new, accurate server for protein secondary structure prediction', *Bioinformatics* **21**(8), 1719.

Pollastri, G.; Przybylski, D.; Rost, B. & Baldi, P. (2002), 'Improving the Prediction of Protein Secondary Structure in Three and Eight Classes Using Recurrent Neural Networks and Profiles', *Proteins: Structure, Function, and Genetics* **47**, 228-235.

Rabiner, L. R. & Juang, B. H. (1986), 'Introduction to Hidden Markov Models.', *IEEE ASSP Mag.* **3**(1), 4-16.

Rabiner, L. R. (1989), 'A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition', *Proceedings of the IEEE* **77**(2), 257.

Reid, A. J.; Yeats, C.; Orengo, A. C. (2007), 'Methods of remote homology

detection can be combined to increase coverage by 10% in the midnight zone', *Bioinformatics* **23**(18):2353-2360;

Rost, B.; Yachdav, G. & Liu, J. (2004), 'The PredictProtein server', *Nucl. Acids Res.* **32**(suppl-2), W321-326.

Sadreyev, R. I. & Grishin, N. V. (2008), 'Accurate statistical model of comparison between multiple sequence alignments', *Nucl. Acids Res.* **36**(7), 2240-2248.

Sander, C. & Schneider, R. (1991), 'Database of homology-derived protein structures and the structural meaning of sequence alignment', *Proteins: Structure, Function, and Genetics* **9**(1), 56-68.

Searls, D. B. & Murphy, K. P. (1995), 'Automata-theoretic models of mutation and alignment', *Proceedings International Conference on Intelligent Systems for Molecular Biology* **3**, 341-349.

Sheskin, D. J. (2003), *Handbook of Parametric and Nonparametric Statistical Procedures: Third Edition*, Chapman & Hall.

Simossis, V. A. & Heringa, J. (2005), 'PRALINE: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information', *Nucl. Acids Res.* **33**(suppl-2), W289-294.

Simossis, V. A. & Heringa, J. (2003), 'The PRALINE online server: optimising progressive multiple alignment on the web', *Comput. Biol. Chem.* **27**(4-5), 511-519.

Sjölander, K.; Karplus, K.; Brown, M.; Hughey, R.; Krogh, A.; Mian, I. & Haussler, D. (1996), 'Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology', *Comput. Appl. Biosci.* **12**(4), 327-345.

Smith, T. F. & Waterman, M. S. (1981), 'Identification of common molecular subsequences', *J. Mol. Biol.* **147**(1), 195-197.

Sonnhammer, E. L. L.; von Heijne, G. & Krogh, A. (1998), A hidden Markov model for predicting transmembrane helices in protein sequences, *in* 'Proc Int Conf Intell Syst Mol Biol', pp. 175-82.

Stenmark, H. & Olkkonen, V. M. (2001), 'The Rab GTPase family', *Genome Biol* **2**(5), 3007.

Sunyaev, S. R.; Eisenhaber, F.; Rodchenkov, I. V.; Eisenhaber, B.; Tumanyan, V. G. & Kuznetsov, E. N. (1999), 'PSIC: profile extraction from sequence alignments with position-specific counts of independent observations', *Protein*

Eng. **12**(5), 387-394.

Söding, J. (2005), 'Protein homology detection by HMM-HMM comparison', *Bioinformatics* **21**(7), 951-960.

Taylor, W. R. (1987), 'Multiple sequence alignment by a pairwise algorithm', *Comput. Appl. Biosci.* **3**(2), 81-87.

Thompson, J. D.; Higgins, D. G. & Gibson, T. J. (1994), 'CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice', *Nucl. Acids Res.* **22**(22), 4673-4680.

Thompson, J. D.; Koehl, P.; Ripp, R. & Poch, O. (2005), 'BAliBASE 3.0: Latest developments of the multiple sequence alignment benchmark', *Proteins: Struct., Funct., Bioinf.* **61**(1), 127-136.

Tran, D. & Wagner, M. (1999), Fuzzy hidden Markov models for speech and speaker recognition, *in* '18th International Conference of the North American Fuzzy Information Processing Society', pp. 426-430.

Wallace, I. M.; O'Sullivan, O.; Higgins, D. G. & Notredame, C. (2006), 'M-Coffee: combining multiple sequence alignment methods with T-Coffee', *Nucl. Acids Res.* **34**(6), 1692-1699.

Wan, J.; Kang, S.; Tang, C.; Yan, J.; Ren, Y.; Liu, J.; Gao, X.; Banerjee, A.; Ellis, L. B. M. & Li, T. (2008), 'Meta-prediction of phosphorylation sites with weighted voting and restricted grid search parameter selection', *Nucl. Acids Res.*.

Wang, L. & Jiang, T. (1994), 'On the complexity of multiple sequence alignment', *J. Comput. Biol.* **1**(4), 337-348.

Waterman, M. S. (1995), *Introduction to computational biology: maps, sequences and genomes*, Chapman & Hall/CRC.

Waterman, M. & Byers, T. (1985), 'A dynamic programming algorithm to find all solutions in a neighborhood of the optimum', *Math. Biosci.* **77**(1-2), 179-188.

Wennerberg, K.; Rossman, K. L. & Der, C. J. (2005), 'The Ras superfamily at a glance', *J. Cell Sci.* **118**(5), 843-846.

Wilbur, W. J. & Lipman, D. J. (1983), 'Rapid similarity searches of nucleic acid and protein data banks', *Proc. Natl. Acad. Sci. U. S. A.* **80**(3), 726.

Wistrand, M. & Sonnhammer, E. L. L. (2005), 'Improved profile HMM performance by assessment of critical algorithmic features in SAM and HMMER',

BMC Bioinformatics **6**(1), 99.

Wistrand, M. & Sonnhammer, E. L. L. (2004), 'Transition priors for protein hidden Markov models: an empirical study towards maximum discrimination', *J. Comput. Biol.* **11**(1), 181-193.

Xia, X. (2001), *Data Analysis in Molecular Biology and Evolution*, Springer Verlag.

Xie, L.; Xu, P.; Chang, S. F.; Divakaran, A. & Sun, H. (2004), 'Structure analysis of soccer video with domain knowledge and hidden Markov models', *Pattern Recognition Letters* **25**(7), 767-775.

Xiong, J. (2006), *Essential bioinformatics*, Cambridge University Press.

Yamamoto, M. & Sadamitsu, K. (2005), 'Dirichlet mixtures in text modeling'(CS-TR-05-1), Technical report, University of Tsukuba.

Yoon, B. J. & Vaidyanathan, P. P. (2008), 'Structural alignment of RNAs using profile-csHMMs and its application to RNA homology search: overview and new results', *IEEE Transactions on Automatic Control* **53**(Special Issue), 10-25.

Zhang, S.; Borovok, I.; Aharonowitz, Y.; Sharan, R. & Bafna, V. (2006), 'A sequence-based filtering method for ncRNA identification and its application to searching for riboswitch elements', *Bioinformatics* **22**(14), e557.

Zhou, H. & Zhou, Y. (2005), 'SPEM: improving multiple sequence alignment with sequence profiles and predicted secondary structures', *Bioinformatics* **21**(18), 3615-3621.