

Creation of HMMER-based fold recognition methods for the Paracel GeneMacher

Julian Macoveanu

Department of Biochemistry and biophysics, Stockholm University

Supervisor: Arne Elofsson, Stockholm Bioinformatics Center,
Stockholm University

Contents

Contents	2
Abstract.....	3
Introduction	3
Methods	4
Sequence alignments	4
Statistical Profiles	5
Theory behind profile HMMs	7
Scoring a Sequence with an HMM	10
What the Score Means	11
Building an HMM	12
Web based search tools made available using the Paracel GeneMacher..	13
GeneMacher	16
Search Tools used.....	16
SCOP - Structural Classification of Proteins	18
Comparison of the web based search tools using SCOP	20
Methods.....	20
Results.....	21
Discussion	25
The speed performance.....	28
Size of database	30
Conclusion	31
References.....	32

Abstract

It has been shown that by using multiple aligned sequence information three times as many remote homologue proteins could be detected than by employing standard techniques that use a pairwise sequence aligning method. Multiple aligned sequences can be used to generate Hidden Markov Models (HMMs) profiles. Unlike traditional pairwise alignment algorithms such as BLAST [1] or FASTA [10], HMMs search methods use position-specific scoring. This allows HMMs to distinguish entire families of sequences by modelling the extent to which the regions should be conserved in a multiple alignment. This results in a better capability in finding related proteins even with low sequence identity.

Two different web based fold recognition methods were created by using algorithms available on the Paracel GeneMacher supercomputer. The two methods have been benchmarked together with the public available PSI-BLAST [1] using the manual classification of proteins offered by SCOP [7]

Introduction

The relationships between proteins span a broad range, from the case of almost identical sequences to apparently unrelated sequences sharing only rough 3D structure. Proteins might have considerable structural similarities even when no evolutionary relationship of their sequences can be detected. This property is often referred to as the proteins sharing only a "fold". There are also sequences of common origin in each fold, called a "superfamily", and in them groups of sequences with clear similarities, designated "family". Developing algorithms to reliably identify proteins related at any level is one of the most important challenges in the fast growing field of bioinformatics today. However, it is not at all certain that a method proficient at finding sequence similarities performs well at the other levels, or vice versa.

By determining how sequences are related to known proteins one can make predictions of their structural, functional and evolutionary features. During the last few years several excellent studies have changed the view of the best methods to detect relationship between proteins. These studies differ in detail but have one common nominator, they use the SCOP classifications [7], to create a benchmark used in evaluating the performance of different recognition methods. SCOP is a hierarchical scheme where each protein domain is classified into a family which in turn belongs to a superfamily that is a subclassification of the fold category. The SCOP database is to a large extent hand tuned by Alexei Murzin, giving the following explanations to the levels: proteins sharing family have a clear evolutionary relationship; those within a superfamily are of probable common evolutionary origin; while the fold level is characterized by a major structural similarity.

Here, I have compared the performance of three search methods on these different levels of similarity. As expected, it becomes much harder to detect proteins as their sequences diverge. For family related sequences the best method gets 97 % of the top hits correct. When the sequences differ but the proteins belong to the same superfamily this drops to 47 %, and in the case of proteins with only fold similarity it is as low as 10 %.

Two of the three search methods compared here were made available for public use using the GeneMacher hardware provided by Paracel. The third method PSI-BLAST [1], is also available online on different sites on the internet.

Methods

Sequence alignments

There are two main reasons to align proteins sequences. To study the relationship between two proteins and to scan a database with newly determined protein sequence and identify possible functions for the protein

by analogy with characterized proteins. Protein sequences can be aligned in a “pairwise” manner, or they can build up a “multiple alignment”. Pairwise alignments can be made using the global alignment algorithm or the local alignment algorithm.

Global alignments aim at optimally aligning two or more sequences. The most used methods for global alignments are based on algorithms originally developed by Needleman and Wunch and modified by Sellers. This procedure (the NWS method) is using a dynamic programming algorithm that simplifies the enormous task of calculating a score for all possible alignments of two sequences with gaps of any lengths. The sequences to be aligned are arranged as rows and columns of a rectangular matrix. A score is calculated for each position of the matrix according to three possible events: replacement (or conservation) of a residue, insertion in sequence A or insertion in sequence B.

The dynamic programming algorithm can also be used for finding local sequence similarities. The Smith and Waterman algorithm is very similar to the NWS method except that instead of calculating the score given by the best alignment over the whole length of the sequence the alignments are made to produce a local sequence identity with as high score as possible [2].

Traditional search programs such as BLAST [3] or FASTA [10] all use algorithms that employ pairwise alignments, especially local alignments. It have been shown [9], that by using multiple aligned sequence information three times as many remote homologue proteins could be detected than by employing standard techniques that use a pairwise sequence aligning method. Profile search algorithms are all using multiple alignments in order to build the statistical profile.

Statistical Profiles

The sequence of proteins which share a common ancestor are not exactly alike. However, they inherit many similarities in primary structure from their

ancestor. This is known as conservation of primary structure in a protein family.

By multiple aligning several protein sequences the detection of distantly related homologous proteins is improved. New sequences can also be aligned more accurately when the alignment is based on the pattern of conservation from already aligned sequences. The alignment of many homologous sequences may also provide more information about the relationship between and functions of these proteins.

An example is seen in *Figure 1*, where most of the positions (columns) in the multiple alignment contain only a few distinct amino acids.

These sequence similarities make it possible to create a statistical model of a protein family. The model shown in *Figure 1* is a simplified statistical *profile*, a model which shows the amino acid probability distribution for each position in the family [6]. According to this profile, the probability of C in position 1 is 0.8, the probability of G in position 2 is 0.4, and so forth. The probabilities are calculated from the observed frequencies of amino acids in the family.

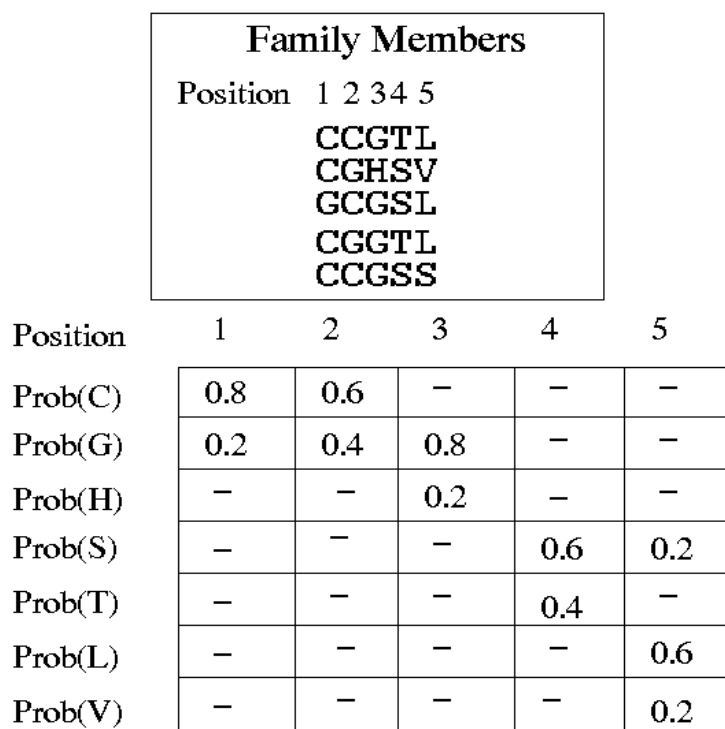


Figure 1. A statistical model of ten related proteins

Given a profile, the probability of a sequence is the product of the amino acid probabilities given by the profile. For example, the probability of CCGSV, given the profile in *Figure1*, is:

$$0.8 * 0.4 * 0.8 * 0.6 * 0.2 = .031$$

Given a statistical model, the probability of a sequence is used to calculate a *score* for the sequence. Because multiplication of fractions is computationally expensive and prone to floating point errors such as underflow, a convenient transformation into the logarithmic world is used. The score of a sequence is calculated by taking the logs of all amino acid probabilities and adding them up. Using this method with base *e* logarithms, the score of CCGSV is:

$$\log_e(0.8)+\log_e(0.4)+\log_e(0.8)+\log_e(0.6)+\log_e(0.2) = -3.48$$

In practice, profile models take other factors into account. For example, members of a protein family have varying lengths, so a score *penalty* is charged for insertions and deletions. The scores of individual amino acids in a profile are also *position specific*. In other words, more weight must be given to an unlikely amino acid which appears in a structurally important position in the protein than to one which appears in a structurally unimportant position [4].

Although these refinements are necessary to create good profile models, they introduce many additional free parameters which must be calculated when building a profile, and unfortunately, the calculations must be done by trial and error. These limitations set the stage for a new kind of profile, based on the Hidden Markov model.

Theory behind profile HMMs

In a standard substitution matrix, such as BLOSUM62, the substitution of one amino acid with another is associated with a single score. Profile models take other factors into account. For example, members of a protein family have varying lengths, so a score penalty is charged for insertions and deletions.

Database searches can be customized to find specific protein families or domains using substitution scores that reflect the substitution frequencies of each individual amino acid position in a domain. Position-specific scoring matrix or HMMs profiles, in its simplest form, may consist of a set of 20 substitution scores at each position along the motif – one for each of the amino acids. The position specificity implies that more weight must be given to an unlikely amino acid which appears in a structurally important position in the protein than to one which appears in a structurally unimportant position.

The use of position-specific scores for the matching or substitution of a residue and for the opening or extension of a gap separates HMMs from the traditional pair-wise alignment algorithms such as BLAST [1] or FASTA [10]. Position-specific scoring allows HMMs to distinguish entire families of sequences by modelling the extent to which the regions should be conserved in a multiple alignment.

As a result, distantly related proteins can be found even with low sequence identity, if the similarities and differences are common to the family members. This type of analysis is quite powerful because the function of divergent proteins is conserved through evolution even though sequence elements are free to change in some areas [7].

The HMM is used to statistically describe a protein family's consensus sequence. This statistical description can be used for sensitive and selective database searching.

The model consists of a linear sequence of nodes with a begin state (START) and an end state (END) as shown in the *Figure 2*. Each node in an HMM has a match state, insert state and delete state with position-specific probabilities for transitioning into each of these states from the previous node.

In addition to a transition probability, the match state also has position-specific probabilities for seeing a particular residue. Similarly, the insert state

has probabilities for inserting a residue at the position given by the node. There is also a chance that no residue is associated with a state. That probability is indicated by the probability of transitioning to the delete state. Both transition and emission probabilities can be generated from a multiple alignment of a family of sequences.

An HMM can be visualized as a finite state machine as shown in the figure below:

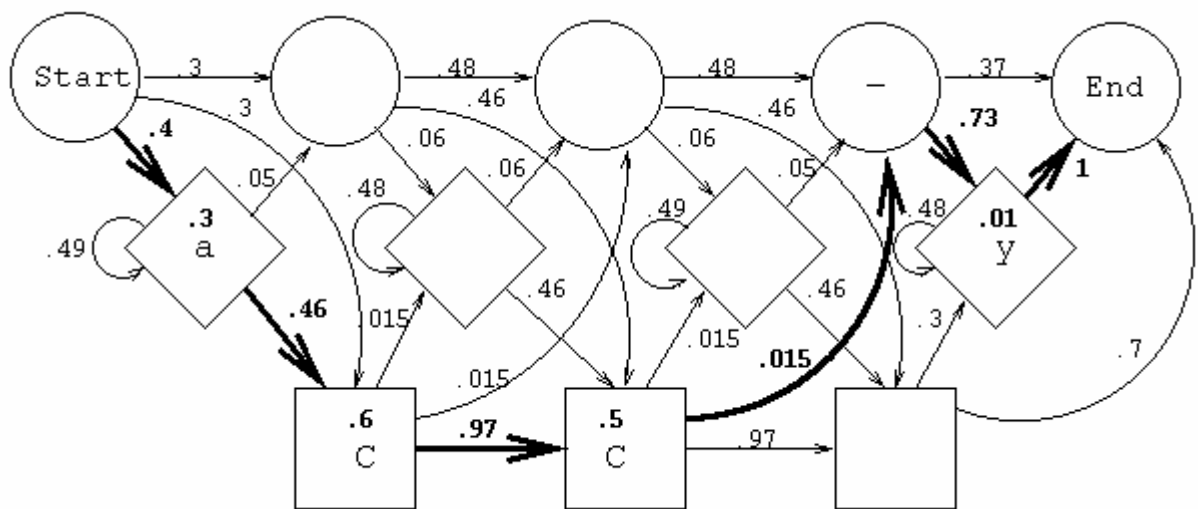


Figure 2. A possible hidden Markov model for the protein ACCY. The protein is represented as a sequence of probabilities. The numbers in the boxes show the probability that an amino acid occurs in a particular state, and the numbers next to the directed arcs show probabilities which connect the states. The probability of ACCY is shown as a highlighted path through the model.

Finite state machines typically move through a series of states and produce some kind of output either when the machine has reached a particular state or when it is moving from state to state. The HMM generates a protein sequence by emitting amino acids as it progresses through a series of states. Each state has a table of amino acid emission probabilities similar to those described in a profile model. There are also transition probabilities for moving from state to state.

Figure 2 shows one topology for a hidden Markov model. There are three kinds of states represented by three different shapes. The squares are called match states, and the amino acids emitted from them form the conserved primary structure of a protein. These amino acids are the same as those in the common ancestor or, if not, are the result of substitutions. The diamond shapes are insert states and emit amino acids which result from insertions. The circles are special, silent states known as delete states and model deletions.

Transitions from state to state progress from left to right through the model, with the exception of the self-loops on the diamond insertion states. The self-loops allow deletions of any length to fit the model, regardless of the length of other sequences in the family.

Scoring a Sequence with an HMM

Any sequence can be represented by a path through the model. The probability of any sequence, given the model, is computed by multiplying the emission and transition probabilities along the path.

In *Figure 2*, a path through the model represented by ACCY is highlighted. In the interest of saving space, the full tables of emission probabilities are not shown. Only the probability of the emitted amino acid is given. For example, the probability of A being emitted in position 1 is 0.3, and the probability of C being emitted in position 2 is 0.6. The probability of ACCY along this path is:

$$.4 * .3 * .46 * .6 * .97 * .5 * .015 * .73 * .01 * 1 = 1.76 \times 10^{-6}.$$

As in the profile case described above, the calculation is simplified by transforming probabilities to logs so that addition can replace multiplication. The resulting number is the raw score of a sequence, given the HMM.

For example, the score of ACCY along the path shown in *Figure 2* is:

$$\log_e(.4) + \log_e(.3) + \log_e(.46) + \log_e(.6) + \log_e(.97) + \log_e(.5) + \log_e(.015) + \log_e(.73) + \log_e(.01) + \log_e(1) = -13.25$$

The calculation is easy if the exact state path is known, as in the toy example of *Figure 2*. In a real model, many different state paths through a model can generate the same sequence. Therefore, the correct probability of a sequence is the sum of probabilities over all of the possible state paths. Unfortunately, a brute force calculation of this problem is computationally impractical, except in the case of very short sequences. Two good alternatives are to calculate the sum over all paths inductively using the forward algorithm, or to calculate the most probable path through the model using the Viterbi algorithm.

An HMM can be compared (that is, aligned) with a new sequence to determine the probability that the sequence belongs to the modelled family. The most probable path through the HMM (i.e., which transitions were taken and which residues were released at match and insert states) taken to generate a sequence similar to the new sequence determines the similarity score.

What the Score Means

Once the probability of a sequence has been determined, its score can be computed. Because the model is a generalization of how amino acids are distributed in a related group (or class) of sequences, a score measures the probability that a sequence belongs to the class. A high score implies that the sequence of interest is probably a member of the class, and a low score implies it is probably not a member.

Building an HMM

Parameters can be set for an HMM in two ways. An HMM can be trained from initially unaligned sequences or it can be built from pre-aligned sequences (i.e. where the state path are assumed to be known). In the second case, the parameters are estimated by converting observed counts of symbols emissions and state transitions into probabilities. To build the HMM profile an existing multiple aligned program is given as input. If the state paths for all the training sequences are known, the emission and transition probabilities in the model can be calculated by computing their expected value: observing the number of times each transmission or emission occurs in the training set and dividing by the sum of all the transmission probabilities or all the emission probabilities.

Training algorithms are of interest because the probable alignment for the sequence in question may not be known. The most widely training algorithm used is Baum-Welch expectation maximization or gradient descent algorithms [11].

Web based search tools made available using the Paracel GeneMacher

1. Search a protein database through a HMM model:

<http://www.sbc.su.se/~julian/hmm/protdb.html>

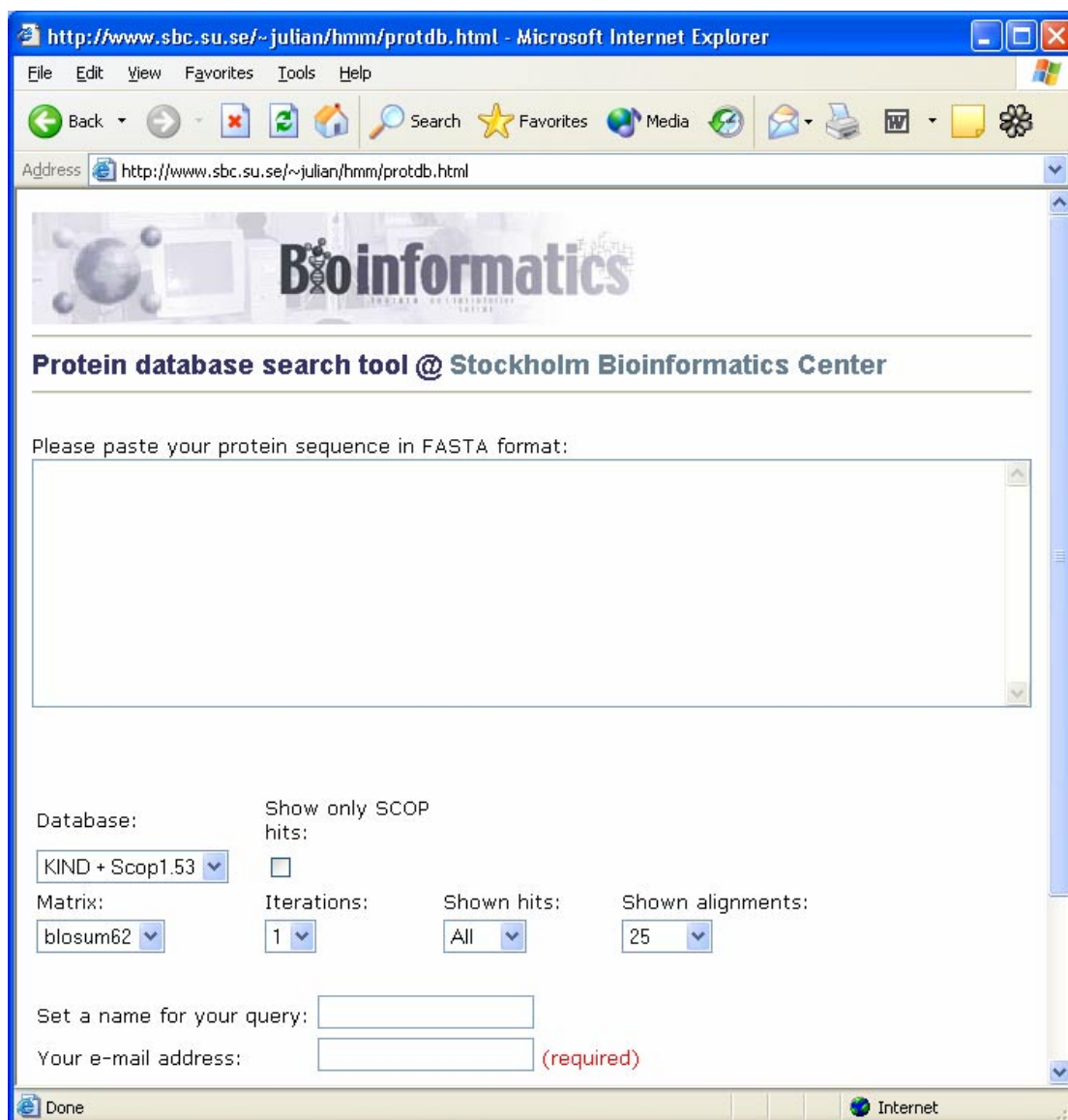


Figure3. Example of the web based user interface for the ClustalW search method

The search uses a protein query in FASTA format and runs it against the KIND (Karolinska Institutet Non-redundant Database), Scop [7] and PDB [3] databases using the position independent Smith-Waterman algorithm (swp).

The search can be done using several standard substitution matrixes like BLOSUM62 or PAM250. The matching protein sequences are multiple aligned by a hierarchical method using the ClustalW software [17]. A position-specific scoring matrix or HMM is then calculated and calibrated with the HMMER software package. The HMM is searched against the databases. By multiple aligning the output and building a new HMM the user can choose to iterate several times in order to find new sequences. The different steps are depicted in *figure 4*.

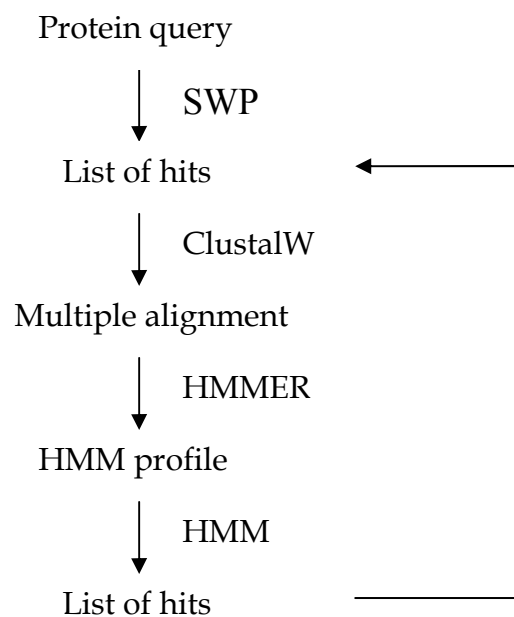


Figure 4. Visualization of the different steps for the ClustalW-HMM search method

2. Search a HMM database (Scop, Pfam)

<http://www.sbc.su.se/~julian/hmm/hmmdb.html>

The search compares a protein sequence against a database of HMM models like Pfam [12] and SCOP [7]. HMM-Superfamily searches perform position independent gap scoring using the Viterbi algorithm. The user interface is analogue to the one for the ClustalW-HMM method.

Both search methods are run on the GeneMacher hardware and the results are sent to the user by e-mail.

A search example:

Protein query from the SCOP database:

```
>d1alo_3 4.36.1.1.1 (194-310) Aldehyde oxidoreductase, domain 3 {Desulfovibrio gigas}
dygadlglkmpagtlhlamvqakvshanikgidtsealtmpgvhsvithkdvkgnritg
litfptnkgdgdwdrpilcdekfvfygdialvcadseanaraaekvkvdleelpay
```

Result sent by e-mail to the user:

HMM BTK 4.0.12-77/77 2001-05-30 (Fdf Client 1.442)

Copyright 2000 Paracel, Inc

```
-----
HMM file:      swp [swp]
Sequence file: scop_1.prot
-----
```

Query HMM: swp | [swp]
[HMM has been calibrated; E-values are empirical estimates]

Scores for complete sequences (score includes all domains):
Sequence Description Score E-value N

```
-----
d1qj2b1 4.36.1.1.3 (10-146) Carbon monoxide (CO) deh 355.3 2.8e-103 1
d1qj2h1 4.36.1.1.3 (10-146) Carbon monoxide (CO) deh 354.8 4.1e-103 1
d1alo_3 4.36.1.1.1 (194-310) Aldehyde oxidoreductase 282.0 3.2e-81 1
d1dgja3 4.36.1.1.2 (194-310) Aldehyde oxidoreductase 273.0 1.7e-78 1
[no more scores below E threshold]
```

Parsed for domains:

```
Sequence Domain seq-f seq-t  hmm-f hmm-t  score E-value
-----
d1qj2b1 1/1 1 137 [] 1 143 [] 355.3 2.8e-103
d1qj2h1 1/1 1 137 [] 1 143 [] 354.8 4.1e-103
d1alo_3 1/1 1 117 [] 27 143 .] 282.0 3.2e-81
d1dgja3 1/1 1 117 [] 27 143 .] 273.0 1.7e-78
```

Alignments of top-scoring domains:

```
d1qj2b1: domain 1 of 1, from 1 to 137: score 355.3, EVD = 1.1e-107
      *->tsaeRaekLqGmGcKrkRveDirFteGkGadvdlkkpegtLhlalvq
      tsaeRaekLqGmGcKrkRveDirFteGkG+vd++k++g+L++++v+
d1qj2b1 1 TSAERAELQGMGCKRKRVEDIRFTQGKGNVDDVKLPGMLFGDFVR 47
```

```
      akvsHArIKgIDTSeAkalPGVfaVLThkDvKgnRITGLitfPTnkGDG
      ++++HArIK+IDTS+AkalPGVfaVLT++D+K++N L+++PT++GD
d1qj2b1 48 SSHAHARIKSIDTSKAKALPGVFAVLTAADLKPLN----LHYMPTLAGD- 92
```

```
      WerpiLaDeKvlqygdevAIVvAdseanAraAaEkVkvDIEeLPvy<-*
      ++++LaDeKv1++++evA+VvA+++++A++A+E+V+vD+E+LPv+
d1qj2b1 92 -VQAVLADEKVLQNFQNEVAFVVAKDRYVAADAIELVEVDYEPLPVL 137
```

```
d1qj2h1: domain 1 of 1, from 1 to 137: score 354.8, EVD = 1.6e-107
      *->tsaeRaekLqGmGcKrkRveDirFteGkGadvdlkkpegtLhlalvq
      tsaeRaekLqGmGcKrkRveDirFteGkG+vd++k++g+L++++v+
d1qj2h1 1 TSAERAELQGMGCKRKRVEDIRFTEGKGNVDDVKLPGMLFGDFVR 47
```

```
      akvsHArIKgIDTSeAkalPGVfaVLThkDvKgnRITGLitfPTnkGDG
      ++++HArIK+IDTS+AkalPGVfaVLT++D+K++N L+++PT++GD
d1qj2h1 48 SSHAHARIKSIDTSKAKALPGVFAVLTAADLKPLN----LHYMPTLAGD- 92
```

```
      WerpiLaDeKvlqygdevAIVvAdseanAraAaEkVkvDIEeLPvy<-*
      ++++LaDeKv1++++evA+VvA+++++A++A+E+V+vD+E+LPv+
d1qj2h1 92 -VQAVLADEKVLQNFQNEVAFVVAKDRYVAADAIELVEVDYEPLPVL 137
```

```
d1alo_3: domain 1 of 1, from 1 to 117: score 282.0, EVD = 1.2e-85
      *->gkGadvdlkkpegtLhlalvqakvsHArIKgIDTSeAkalPGVfaVL
      ++Gad++k+p+gtLhla+vqakvsHA+IKgIDTSeA+++PGV++V+
d1alo_3 1 DYGADLGLKMPAGTLHLAMVQAKVSHANIKGIDTSEALTMPGVHSVI 47
```

```

ThkDvKgkNRITGLItfPTnkGDGWerpiLaDeKvlqygdevAIVvAdse
ThkDvKgkNRITGLItfPTnkGDGW+rpiL+DeKv+qygd++AIV+Adse
d1alo_3 48 THKDVKGKNRITGLITFPTNKGDGWDRPILCDEKVFQYGDICIALVCADSE 97

anAraAaEkVkvDIEeLPvy<-*
anAraAaEkVkvDIEeLP+y
d1alo_3 98 ANARAAAEEKVKVDLEELPAY 117

d1dgja3: domain 1 of 1, from 1 to 117: score 273.0, EVD = 6.4e-83
*->gkGadvdlkkpegtLhlalvqakvsHArIKgIDTSeAkalPGVfaVL
++Gad++l++pe+tLhlal+qakvsHA+IKgIDTSeA+++PGV++VL
d1dgja3 1 EFGADAAALRMPENTLHLALAQAQVSHALIKGIDTSEAEKMPGVYKVL 47

ThkDvKgkNRITGLItfPTnkGDGWerpiLaDeKvlqygdevAIVvAdse
ThkDvKgkNRITGLItfPTnkGDGWerpiL+D+K++qygd++A+V+Adse
d1dgja3 48 THKDVKGKNRITGLITFPTNKGDGWERPILNDSKIFQYGDALAIVCADSE 97

anAraAaEkVkvDIEeLPvy<-*
anAraAaEkVkv+DIE+LP+y
d1dgja3 98 ANARAAAEEKVKFDLELLPEY 117

[no more alignments below domE threshold]

```

GeneMacher

The GeneMatcher is a high-throughput supercomputer manufactured by Paracel. It offers superior sensitivity and performance in sequence similarity analysis by accelerating a large number of heuristic and dynamic programming algorithms. The version used here utilizes 6400 processors which gives the computer a high throughput. This means that sequences made of a total number of 6400 amino acids can be analyzed at the same time. A post processor is used to store query sequences and databases before they are submitted to GeneMacher. The post processor even provides both a browser-based GUI and a command-line interface. Through the interfaces users are able to perform several nucleotide and protein searches. The alignments generated can be local, global or mixed local-global with affine and double affine scoring. Results can be output in several standard formats [8].

Search Tools used

HMMER – A software package for database searching using profile HMMs. It includes several programs:

hmmalign: aligns a sequence to a given HMM profile.

hmmbuild: build a profile HMM from a multiple sequence alignment.

hmmcalibrate: Takes an HMM and empirically determines parameters that are used to make searches more sensitive, by calculating more accurate expectation value scores (E-values).

hmmpfam: search a single sequence against an HMM database.

hmmsearch: search a sequence database with a profile HMM.

SWP [8] – is a position dependent search program available on the Paracel GeneMacher. It is based on the Smith-Waterman algorithm and is used to find sequences similar to a query sequence. The algorithm uses dynamic programming to find out the optimal of the query to the database sequences. SWP uses protein queries and databases and it is thought to be more sensitive than the analogue heuristic method used in BLAST.

ClustalW [13] – is a program for constructing multiple alignments using a hierarchical method. All pairs of sequences to be aligned are compared first by a pairwise method of sequence comparison. A three is then calculated to place more similar pairs of sequences closer together than sequences that are less similar. The multiple alignment is built by starting with the pair of sequences that is most similar and then aligning the pair that are next most similar and so on.

PSI-BLAST [1] – is a profile search program that uses a multiple alignment for more sensitive searches of protein databases. It employs some principles of full probabilistic modeling to build HMM-like models. PSI-BLAST searches the protein query using the BLAST method and then as part of the search the program generates a multiple alignment from the query sequence. The alignment is however different from the alignment made by ClustalW. In a conventional multiple alignment all sequences in the set have equal weight. As a consequence the multiple alignments will normally be longer than any one of the individual sequences because gaps will be inserted to optimize the alignment. In contrast, a PSI-BLAST multiple alignment has always the length

of the query sequence used for the search. If alignment of the query to a database requires an insertion in the query sequence then the inserted sequence from the database sequence is discarded. A profile is then built using position specific scores for each amino acid residues. The profile is then run against a database so new similar sequences can be detected. A new multiple alignment can be made with the new sequences, a new profile constructed and searched again against the database, so the search procedure can be iterated until no new sequences are found (convergence).

SCOP - Structural Classification of Proteins

The SCOP database [7] is created by manual inspection and assisted by a series of automated methods. It aims to provide a detailed and comprehensive description of the structural and evolutionary relationships between all proteins whose structure is known. SCOP provides a thorough examination of all known protein folds and detailed information about the close relatives of any particular protein.

The classification reflects both structural and evolutionary relatedness. There are several levels in the hierarchy, but the principal levels are family, superfamily and fold.

Family: “Clear evolutionary relationship”

Proteins clustered together into families are clearly evolutionarily related. Generally, this means that pairwise residue identities between the proteins are 30% and greater. However, in some cases similar functions and structures provide definitive evidence of common descent in the absence of high sequence identity; for example, many globins form a family though some members have sequence identities of only 15%.

Superfamily: “Probable common evolutionary origin”

Proteins that have low sequence identities, but whose structural and functional features suggest that a common evolutionary origin is probable are placed together in superfamilies.

Fold: “Major structural similarity”

Proteins are defined as having a common fold if they have the same major secondary structures in the same arrangement and with the same topological connections. Different proteins with the same fold often have peripheral elements of secondary structure and turn regions that differ in size and conformation. In some cases, these differing peripheral regions may include half the structure. Proteins placed together in the same fold category may not have a common evolutionary origin: the structural similarities could arise just from the physics and chemistry of proteins favouring certain packing arrangements and chain topologies.

Thanks to the manual classification of protein relationships in SCOP it is possible to create a benchmark in order to evaluate the performance of different search programs. The benchmark was created from the release 1.53 of SCOP containing the PDB set of proteins.

“Superfamily” is a database consisting of all superfamilies from SCOP each of which is represented by a group of hidden Markov models. Each model is created from a seed sequence which is aligned to many superfamily homologues.

Comparison of the web based search tools using SCOP

Methods

The search tools are available at <http://www.sbc.su.se/~julian/hmm> . The comparison was done between three search methods: “searching a protein database through a HMM model” (ClustalW-HMM), “searching a HMM database with a protein query” (HMM-Superfamily) and PSI-BLAST as explained previously.

The ClustalW-HMM method was made by searching the proteins using the *swp* algorithm against a concatenated database made from KIND February 2000 release and SCOP 1.53. A multiple alignment was constructed using ClustalW with default settings. An HMM model was then calculated and calibrated using “hmmbuild” and “hmmcalibrate” in the HAMMER package with default settings. The HMM was then run against the same database (KIND + SCOP) using GeneMacher HMM algorithm. For subsequent iterations the resulting protein hits were multiple aligned again using ClustalW then a new HMM model was build and run against the database until no new hits were found.

The HMM-Superfamily search was done using the GeneMacher HMM-invert algorithm based on the Virtebri algorithm. All settings ware default. In this case the database was Superfamily which is based on the HMM models made from SCOP.

The PSI-BLAST search was done using the same database (KIND + SCOP). The searches were iterated until convergence.

All the results from the searches of the KIND + SCOP database were filtered from the KIND hits hence only SCOP hits were interesting for the analyze.

Results

The comparison was made by running the same 30 proteins from SCOP using all three search methods. The query proteins have been chosen from all different classes and the ones from the same class belonged to different folds. This assures the sequence identity of the proteins was as low as possible. In order to distinct the found proteins in the three levels, family, superfamily and fold the hits that belonged to the same family were discarded when looking for proteins at superfamily level. When looking for proteins sharing the same fold as the query protein both the hits found at family and superfamily level were discarded.

Protein queries	30
Possible correct hits on family level	5892
Possible correct hits on superfamily family level	207
Possible correct hits on fold level	111

Table 1. Size of the test set

Two criteria were used when analyzing the different methods. The first one is a quite intuitive method concerning only the true hits found at the top of the rank. The second criterion discloses the reliability of the hits by measuring the sensitivity and specificity of the results [5]. True hits on the top rank won't get a high score if there E-values are low.

The sensitivity and specificity were defined as:

$$sensitivity = \frac{TP_{E-value}}{TP_{E-value} + FN_{E-value}}$$

$$specificity = \frac{TP_{E-value}}{TP_{E-value} + FP_{E-value}}$$

Where TP is the number of correct hits above a certain E-value threshold, FN is the number of correct hits with an E-value less than the threshold and FP is the number of false hits with an E-value greater than the E-value threshold.

When reducing the specificity the sensitivity gets higher, that is, by decreasing the E-value threshold fewer true hits will be taken into account at the same time as fewer false hits will have a value over the respective E-value. For the hypothetical perfect method both the specificity and the sensitivity should be as close to 1 as possible at the same time. A true hit is a protein found in the result that belongs to the same level of classification according to SCOP as the query protein.

The results from the top ranks analyze are summarized in the tables below:

Search method	Rank 1	Rank 5
ClustalW-HMM	21 (70%)	23 (76%)
HMM-Superfamily	29 (97%)	29 (97%)
PSI-BLAST	27 (90%)	29 (97%)

Table 2. Recognized proteins on family level

Search method	Rank 1	Rank 5
ClustalW-HMM	3 (10%)	4 (13%)
HMM-Superfamily	14 (47%)	20 (67%)
PSI-BLAST	3 (10%)	3 (10%)

Table 3. Recognized proteins on superfamily level

Search method	Rank 1	Rank 5
ClustalW-HMM	0 (0%)	0 (0%)
HMM-Superfamily	1 (3%)	2 (7%)
PSI-BLAST	3 (10%)	3 (10%)

Table 4. Recognized proteins on fold level

Rank 1 represents the fraction of the query proteins that gave a true hit as the first protein found (the one with the lowest E-value). Rank 5 shows the fraction of the query proteins that gave at least one true hit among the top five hits. The values in parentheses are given as the percentage of the possible correct hits on respective level.

The specificity and sensitivity results of the three methods are summarized in the three figures below:

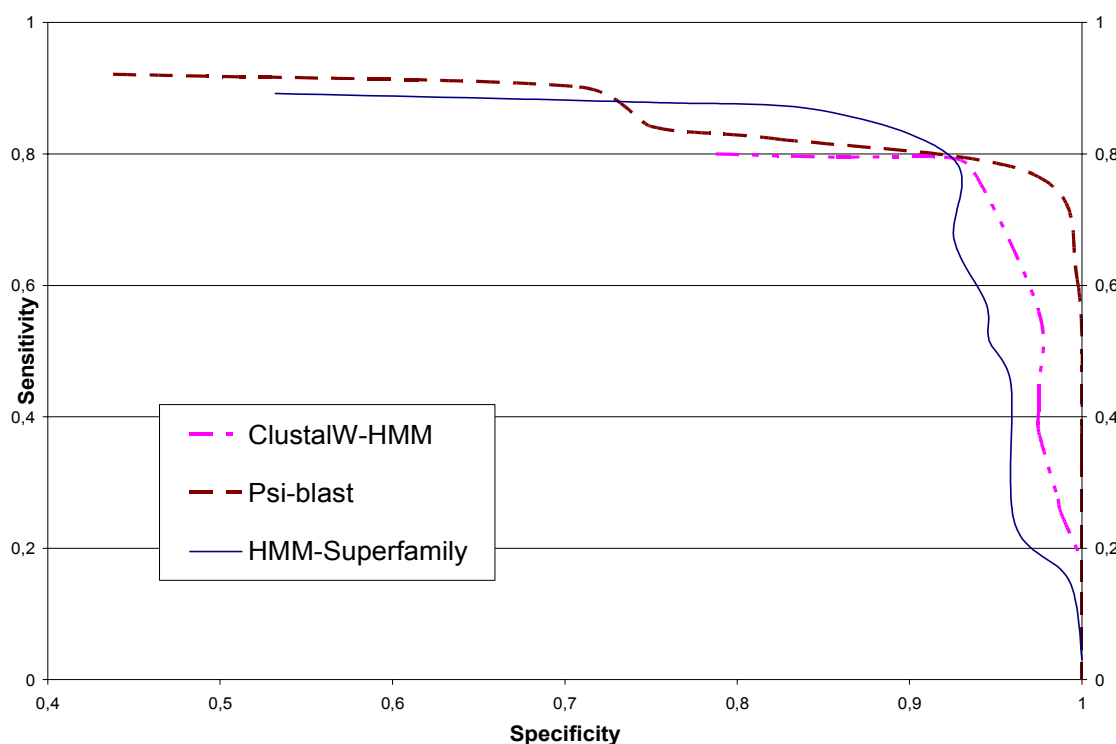


Figure 5. The specificity-sensitivity curve at family level for the three search methods.

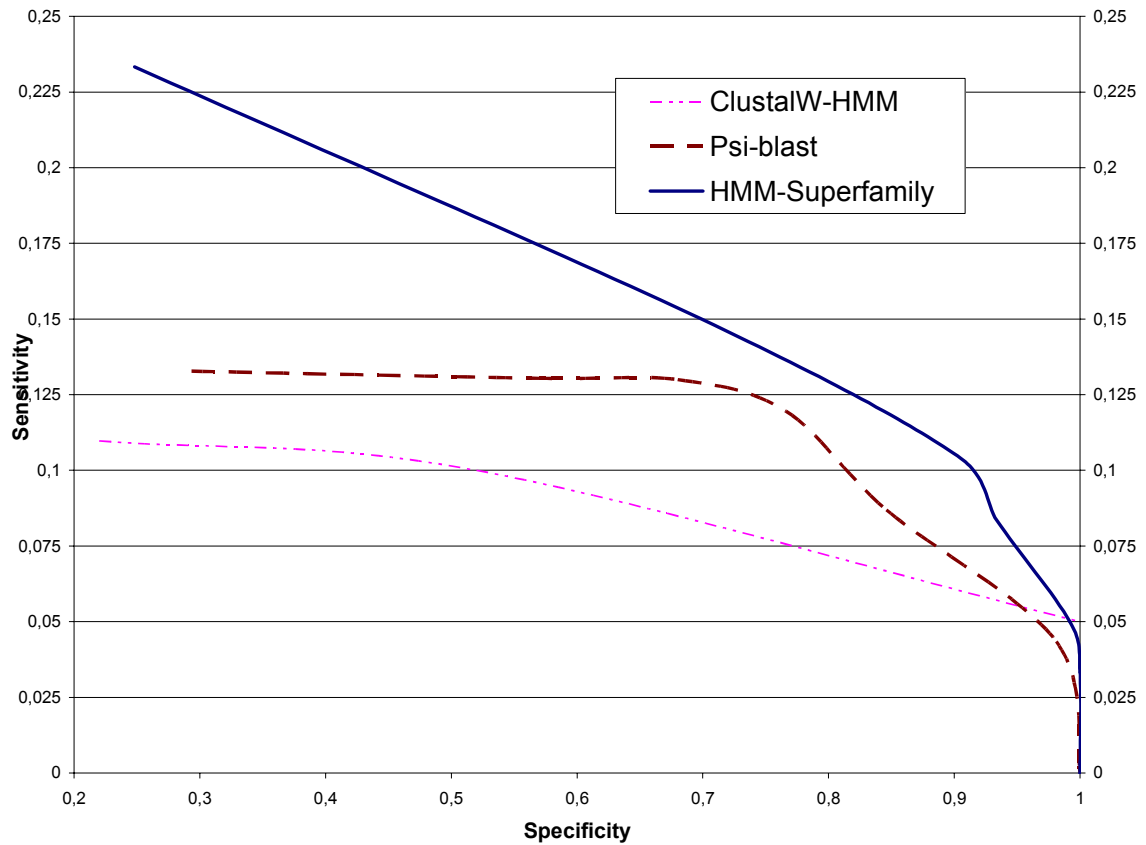


Figure 6. The specificity-sensitivity curve at superfamily level for the three search methods.

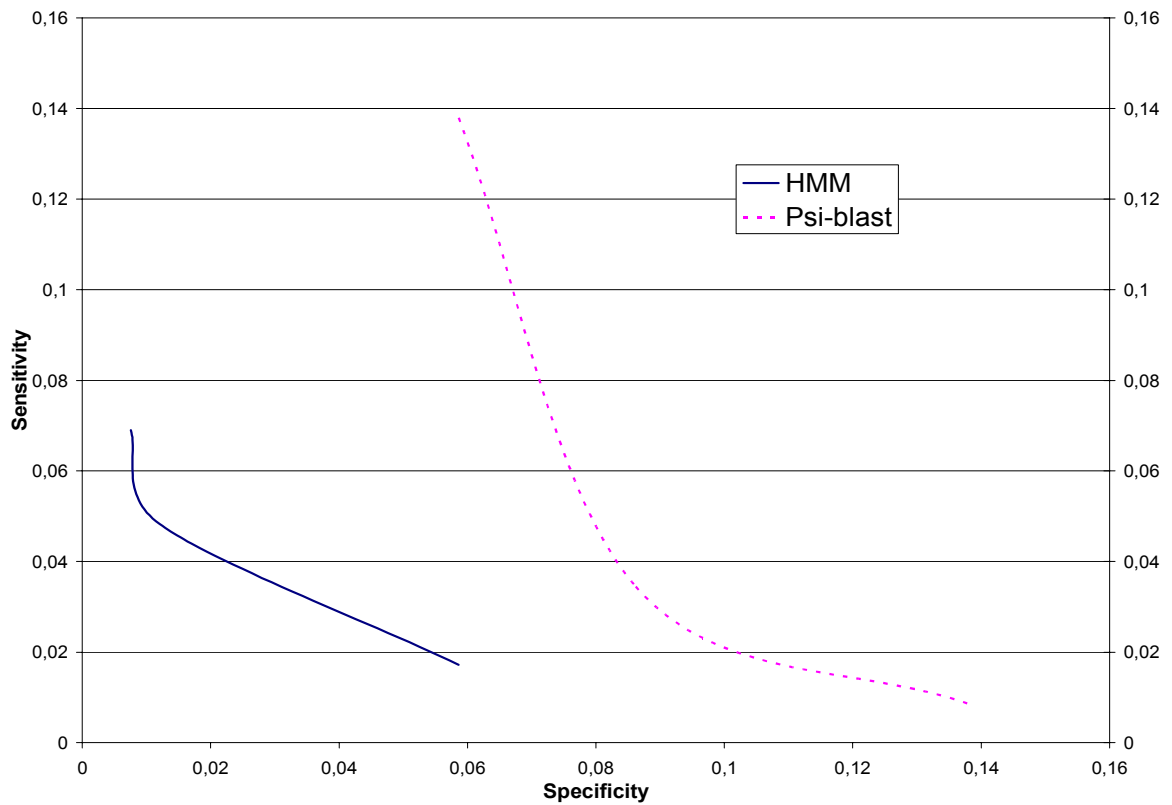


Figure 7. The specificity-sensitivity curve at fold level for two of the search methods.

The plots were constructed by plotting the specificity against the sensitivity, each point corresponding to a certain E-value.

Discussion

Family level

The analyze of the results from the top ranks on the family level shows that the HMM-Superfamily method found most homolog proteins to the query protein that belong to the same family if the E-value threshold for the shown hits was the default value '10'. 97 % of the queries had a true hit as the first hit. The value persisted when considering the rank 5. The results were slightly lower for the PSI-BLAST search method, where 90% of the queries had a true hit for the rank 1 and 97% at rank 5.

ClustalW-HMM performed worse than PSI-BLAST with only 70% for the rank 1 and 76% for the rank 5.

The specificity-sensitivity analyze can be observed in *Figure 5*. PSI-BLAST performed very well as expected, especially for low E-values. The specificity starts to decrease though sooner than for the other methods but it's still very high until the specificity gets very close to one. HMM-Superfamily's performance is very close to the one of PSI-BLAST particularly for high E-values and for a shorter interval even higher than PSI-BLAST. The curve starts though to decline at lower specificity values than PSI-BLAST. ClustalW-HMM performs well to a certain extent, especially for higher E-values where it outperforms HMM-Superfamily with somewhat higher sensitivity values. E-values can be chosen quite high without risking to lose true hits.

Superfamily level

The hits on superfamily level were counted after exclusion of all the hits belonging to the family level. This was in order to separate SCOP's different

levels of classification and obtain a better understanding of the search capabilities of the three different search methods.

Table 3 shows the results from the top ranks. Considering ClustalW-HMM, 10 respectively 13 percent off the query proteins made the rank 1 and 5.

HMM-Superfamily performed a lot better here finding more of the superfamily proteins. The ranks 1 and 5 scores are much better, 47 and 67 % respectively.

On this level PSI-BLAST performed almost as good as ClustalW-HMM but much worse than HMM-Superfamily. This search method found fewest proteins belonging to the same superfamily as the query protein when compared to the results of the other search methods. Further, the scores from the rank 1 and 5 were also as low as the ones for ClustalW-HMM, (10 %).

The results from the specificity-sensitivity curves in *figure 6* confirms the results from the top ranks on Superfamily level. ClustalW-HMM method performs worse compared with the other methods having the lowest sensitivity values. HMM-Superfamily seems to have a lot higher sensitivity values for high E-values, but the performance of all three methods appears to be very close to each other as the specificity approaches 1.

All three methods had a lot poorer sensitivity vs. specificity scores on the superfamily level then on the family level and the methods lose even more of their competence as we look at the scores from the fold level.

Fold level

Both hits on the family level and superfamily level were discarded when counting the hits on the fold level, that is, proteins having major structural similarity.

ClustalW-HMM did not have any hit on the fold level from the 30 proteins that were tested. Even the other two methods had very few hits. HMM-

Superfamily was able to find a few proteins belonging to the same fold as the query proteins. Even the rank 1 and 5 scores were very low, 3 and 7 percent respectively.

PSI-BLAST's results were a little better compared to the HMM-Superfamily's. It found a lot more of the proteins of the same fold with 10% for the rank 1 and 5.

The analyze of specificity-sensitivity curves shows that PSI-BLAST have somewhat higher specificity but the results are inconclusive. Both methods show very low specificities and sensitivities and they found only a few hits.

The results could have been a little more favourable for the ClustalW-HMM method if one looks at the raw data of the search results. They had many hits, comparable to PSI-BLAST and with quite high E-value as well but it missed a few proteins by not being able to find any hits at all. A closer look showed this happened because the ClustalW algorithm used in multiple-aligning protein sequences failed in some case to build the multiple-alignment probable due to the poor similarity of the found proteins.

Another aspect was the difficultness the search methods had in finding the protein that were used as the query sequence. In less than 50% of all the cases the query protein came as the hit with the highest score. In some cases the query protein was not found at all. The reliability of the search methods is still subject of improvement.

Some protein classes seemed to be favoured among others by the search methods. No clear result can be established however and the plausible reason is that the classes had great variety in their amount of proteins on fold, superfamily and family level.

Another observation can be made when trying to analyze the results according to the length of the sequence query. There was clear evidence that

shorter query sequences had a lot worse specificity due to the low E-values of the top rank hits.

The speed performance

There were quite many limitations when trying to make a fair comparison between the three search methods so the results obtained here should only be regarded as a evaluation of the speed of the web services made available rather than the speed of the different search algorithms. This is because different methods needed different amount of iteration until the results converged and even more the hardware used was not the same in all cases. Comparison tables for the speed performance are shown below.

Method	Average time for 1 iteration	Average time 1 query (convergence)	Average time for 20 queries (~6000 AA)
PSI-BLAST	39 sec	5 min 51 sec	1h 57 min
ClustalW-HMM	6min 55 sec	27 min 13 sec	9h 4 min

Table 5. Performance for the KIND + SCOP database (206 MB)

Method	Average time for 1 iteration	Average time 1 query (convergence)	Average time for 20 queries (~6000 AA)
PSI-BLAST	1 sec	3 sec	1 min
ClustalW-HMM	5 min 46 sec	23 min 16sec	7h 45 min

Table 6. Performance for the SCOP database (6,52 MB)

Database	Average time for 1 search	Average time for 20 queries (~6000 AA)
Superfamily(306MB)	6 min 45 sec	7min 10sec
Pfam(252MB)	6min 20 sec	6min 50sec

Table 7. Performance for the HMM - Superfamily algorithm

The *tables 5, 6 and 7* show the average time the search methods took to run but the actual time that passes from time the request is made on the web server to the time the e-mail with the results is received can be longer. This is due to the method used to run the search. The requests are accumulated for 10 minutes for the ClustalW-HMM and for 30 minutes for HMM-Superfamily, then they are all run at the same time.

Table 5 shows the average time it takes for PSI-BLAST and ClustalW-HMM to run one iteration of a query, several iterations until no new proteins are detected in the result, and 20 protein requests. The database used is KIND + SCOP. PSI-BLAST is faster than ClustalW-HMM in all three cases.

The same search methods are tested even in table 6, this time using only the SCOP database. When the search of the database decreases PSI-BLAST search time seems to decrease proportionally. This is not true for ClustalW-HMM which only showed a small decrease in the search time.

Table 7 shows the HMM-superfamily performance for both Pfam and Superfamily profile databases. It is interesting to make notice of the modest difference in time when running one protein query or 20 protein queries. The search time is almost identical. This will be true even if the number of queries will increase until a certain threshold value. After that value the search time will double.

A few comments are necessary here in addition to the results from the tables. The algorithms used for the ClustalW-HMM search method run on different hardware. ClustalW was not run entirely on the GeneMacher because the computer does not support the algorithm. SWP and HMM run on the supercomputer, ClustalW, hmmbuild, hmmcalibrate were run in a multiprocessor Intel platform. This doesn't affect the speed of the query so much it only decreases the throughput of the system, that is, the amount of queries that can be run at the same time.

PSI-BLAST was entirely run on an Intel platform and HMM-Superfamily was entirely run on GeneMacher. This makes HMM-Superfamily the method with the highest throughput, GeneMacher is specialized in running many sequences at the same time (a maximal amount of 6400 amino acids).

Size of database

By using a larger database the theoretical specificity of the obtained hits should increase. The multiple alignment constructed by the search algorithms gets better if the number of protein sequences is increased. For this reason the searches were run against the concatenated database KIND + SCOP instead of only SCOP. *Figure 8* shows a comparison of the results obtained with PSI-BLAST when run against both SCOP and KIND + SCOP.

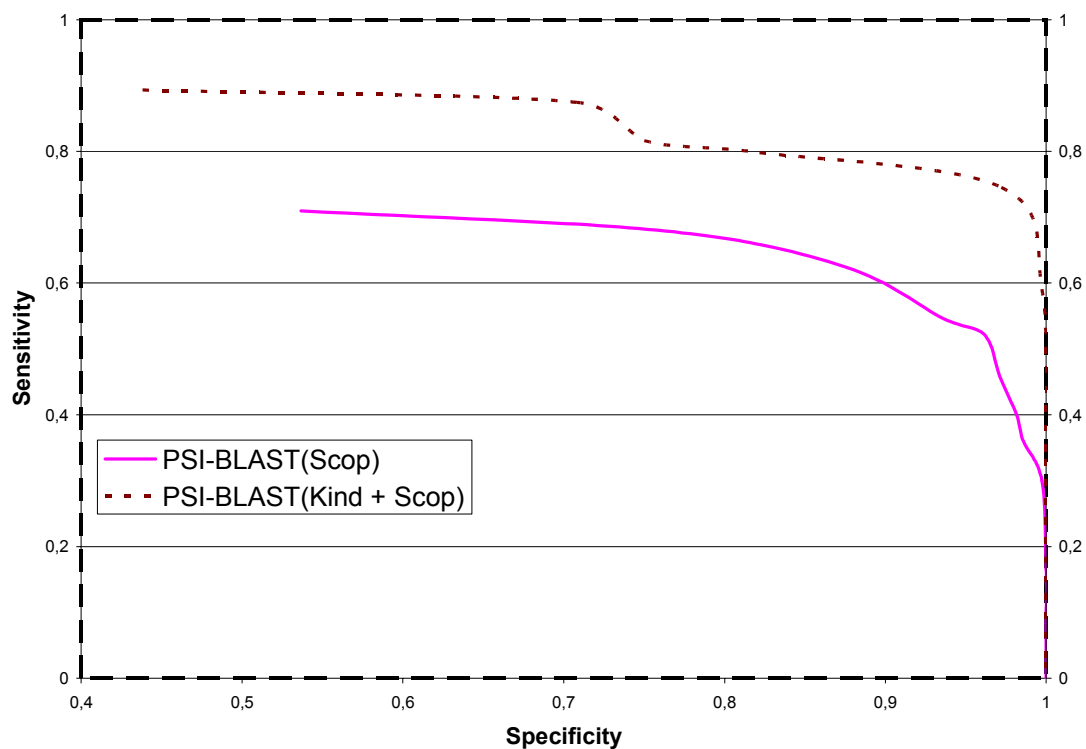


Figure 8. Comparison of PSI-BLAST performance for SCOP hits when run on SCOP and KIND+SCOP databases. The curves show the specificity-sensitivity relationship for the SCOP hits on family level.

Conclusion

None of the three search methods tested here were able to be superior to the other ones at all three classification methods. The sensitivity of all methods decreases radically when looking for proteins with less homology as in the case of superfamily or fold proteins. The reliability of the results for the fold level is though very low. If we consider the family level which is the most common one for protein searches the winner method was not so obvious. The PSI-BLAST method which is even available on the World Wide Web was the most sensitive one but HMM-Superfamily was able to find most true hits than the rest of the methods.

Coming to the superfamily level HMM-Superfamily was the most sensitive method, especially for low E-values. It could also find most of the possible hits when compared to the other methods. ClustalW-HMM does not seem to be suited for searches of proteins with less sequence similarities than proteins belonging to the same family.

Searches for proteins sharing the same fold appear to be made best using PSI-BLAST. The method is a little more sensitive than HMM-Superfamily and is also able to find a few more of the true hits than the other methods. The results are however quite unreliable.

When the speed is concerned PSI-BLAST was the fastest method by far. Even if the ClustalW-HMM has a little lower average search time than HMM-Superfamily, when the server gets more concurrent queries the later method will not show any decrease in response time until a certain load threshold. The response times of the ClustalW-HMM are proportional to the amount of concurrent queries run by the system.

The relatively bad performance of ClustalW-HMM was quite unexpected. The method is analogue to PSI-BLAST but uses more sensitive algorithms. The reason for this can be the global alignment method that ClustalW uses when constructing the multiple alignments.

References

1. Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D.J. (1997) "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." *Nucleic Acids Res.* 25:3389-3402
2. Andreas D. Baxevanis, B. F. Francis Ouellette. "Bioinformatics – A practical Guide to the Analysis of Genes and Proteins, Wiley-Interscience
3. Bernstein F C, Koetzle T F, Williams G J B, Meyer E F Jr, Brice M D, Rogers J R, Kennard O, Shimanouchi T & Tasumi M (1977). The Protein Data Bank: a computer-based archival file for macromolecular structures. *J. Mol. Biol.*, **112**, 535-542.
4. J.D. Thompson, D.G. Higgins, and T.J. Gibson. Improved sensitivity of profile searches through the use of sequence weights and gap excision. *Computer Applications in the Biosciences*, 10:19-29. 1994b.
5. Lindahl, E. Eloffson, A. Identification of related proteins on family, superfamily and fold level, *J. Mol. Biol.* (2000) 295, 613-625
6. M. Gribskov, A.D. McLachlan, and D. Eisenberg. Profile analysis: detection of distantly related proteins. *Proceedings of the National Academy of Sciences of the USA*, 84:4355-4358, 1987.
7. Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. (1995). Scop: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* 247, 536-540.
8. Paracel BioView Toolkit Software documentation, Version 4.0, April 2001

9. Park, J., Karplus, K., Barrett *et al.* (1998). Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J Mol. Bio* 284, 1201-1210.
10. Pearson, W. R. & Lipman, D. J. (1988). Improved tools for biological sequence analysis. *Proc. Natl Acad. Sci. USA*, 85, 2444-2448.
11. Sean R. Eddy, "Profile hidden Markov models", *Bioinformatics* 14(9):755-63, 1998
12. Sonnhammer, E. L., Eddy, S. R. & Durbin, R. (1997). Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins, Struct. Funct. Genet.* 28, 405-420.
13. Thompson J.D., Higgins D.G., Gibson T.J.; "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice."; *Nucleic Acids Res.* 22:4673-4680(1994).