Protein homology detection by HMM–HMM comparison

Johannes Söding

Department of Protein Evolution, Max-Planck-Institute for Developmental Biology, Spemannstrasse 35, D-72076 Tübingen, Germany

ABSTRACT

Motivation: Protein homology detection and sequence alignment are strategies at the basis of protein structure prediction, function prediction, and evolution.

Results: We have generalized the alignment of protein sequences with a profile hidden Markov model (HMM) to the case of pairwise alignment of profile HMMs. We present a method for detecting distant homologous relationships between proteins based on this approach. The method (HHsearch) is benchmarked together with BLAST, PSI-BLAST, HMMER, and the profile-profile comparison tools PROF_SIM and COMPASS, in an all-against-all comparison of a database of 3691 protein domains from SCOP 1.63 with pairwise sequence identities below 20%.

Sensitivity: When predicted secondary structure is included in the HMMs, HHsearch is able to detect between 2.7 and 4.2 times more homologs than PSI-BLAST or HMMER and between 1.44 and 1.9 times more than COMPASS or PROF_SIM for a rate of false positives of 10%. Approximately half of the improvement over the profile–profile comparison methods is attributable to the use of profile HMMs in place of simple profiles.

Alignment quality: Higher sensitivity is mirrored by an increased alignment quality. HHsearch produced 1.2, 1.7, and 3.3 times more good alignments (“balanced” score > 0.3) than the next best method (COMPASS), and 1.6, 2.9, and 9.4 times more than PSI-BLAST, at the family, superfamility, and fold level.

Speed: HHsearch scans a query of 200 residues against 3691 domains in 33s on an AMD64 3GHz PC. This is 10 times faster than PROF_SIM and 17 times faster than COMPASS.


Contact: johannes.soeding@tuebingen.mpg.de

INTRODUCTION

Homology detection and sequence alignment are central themes in bioinformatics because of their manifold applications in areas such as protein function prediction, 3D protein structure prediction, and protein evolution (Bork and Koonin, 1998; Kinch et al., 2003; Henn-Sax et al., 2001). But often no close homolog with known function or structure can be found that would allow to make inferences about the protein of interest. In many of these cases, new and highly sensitive methods could detect and align remotely homologous sequences that provide information about the protein’s function, structure, or evolution. Extending the limits of sensitivity is therefore of great practical importance.

The development of profile–sequence comparison methods such as PSI-BLAST (Altschul et al., 1997) has lead to a great improvement in sensitivity over sequence–sequence comparison methods such as FASTA or BLAST (Pearson and Lipman, 1988; Altschul et al., 1990). This is because a sequence profile, which is built from a multiple alignment of homologous sequences, contains more information about the sequence family than a single sequence. The profile allows one to distinguish between conserved positions that are important for defining members of the family and nonconserved positions that are variable among the members of the family. More than that, it describes exactly what variation in amino acids is possible at each position by recording the probability for the occurrence of each amino acid along the multiple alignment.

A significant improvement over profile–sequence based methods was made possible by comparing profiles to profiles. Several programs for homology recognition have recently been developed that are based on profile–profile comparison: LAMA by Pietrokovski (1996), PROF_SIM by Yona and Levitt (2002), and COMPASS by Sadreyev and Grishin (2003). These programs were shown to be significantly more sensitive than PSI-BLAST and have been applied to identify evolutionary links between protein families previously thought to be unrelated (Pietrokovski, 1996; Kunin et al., 2001; Sadreyev et al., 2003). LAMA is part of the BLOCKS database software suite and was developed to compare a sequence alignment with a database of conserved, ungapped alignments (blocks) that characterize protein families. PROF_SIM and COMPASS allow for gaps and use the Smith-Waterman local alignment algorithm. PROF_SIM employs a column score based on Jensen-Shannon entropy. Statistical significance is reported as a P-value that is calculated directly from the raw score. COMPASS uses a column score based on the relative entropy between the two amino acid distributions. It estimates E-values analytically by generalizing the approach of PSI-BLAST to the profile–profile case. Before profile–profile comparison was applied to homology detection it was standardly employed in multiple sequence alignment methods, for example in the popular tool
CLUSTAL by Thompson et al. (1994). Programs for multiple sequence alignment that incorporate recent advances in profile–profile comparison are PCMA by Pei et al. (2003) and SATCHMO by Edgar and Sjölander (2003). A number of structure prediction servers exist that rely on profile–profile comparison (Rychlewski et al., 2000; Ginalska et al., 2003; Tang et al., 2003; von Öhsen et al., 2003; Tomii and Akiyama, 2004). They build a profile from the query sequence and search for homologous templates of known structure. In general, these templates are similar in structure because structures diverge much more slowly than sequences, and proteins may remain structurally very similar long after their sequence similarity has disappeared (Kinch and Grishin, 2002). These servers are among the best-performing present-day methods for fold recognition, as can be seen from the results of the blind, automated structure prediction contests CAFASP, LIVEBENCH, and EVA (Fischer et al., 2003; Rychlewski et al., 2003; Koh et al., 2003).

Profile Hidden Markov Models (HMMs) are similar to simple sequence profiles, but in addition to the amino acid frequencies in the columns of a multiple sequence alignment they contain the position-specific probabilities for insertions and deletions along the alignment (Figure 1a). The logarithms of these probabilities are in fact equivalent to position-specific gap penalties (Durbin et al., 1998). Profile HMMs perform better than sequence profiles in the detection of homologs and in the quality of alignments (Krogh et al., 1994; Eddy, 1998; Karplus et al., 2001), albeit at the price of a decrease in computational speed. The higher sensitivity is due to the fact that the position-specific gap penalties penalize chance hits much more than true positives which tend to have insertions or deletions at the same positions as the sequences from which the HMM was built. Lyngsø et al. developed an algorithm for the alignment of two HMMs based on the exact maximization of the co-emission probability (Lyngsø et al., 1999). They compared several score variants with each other but did not benchmark their method against others. Here, we generalize the log-odds score maximized in HMM–sequence alignment to the case of HMM–HMM alignment. We present a novel algorithm for HMM–HMM comparison instead of profile–profile comparison and we score predicted secondary structure. Reliability is crucial for a tool that is to be applied to detect evolutionary links. It was found that E-values reported by most tools, including ours, can be very unreliable. HHsearch therefore reports, in addition to E-values, the probability for each match to be a true positive, based on the $1.4 \times 10^7$ pairwise comparisons of our benchmark.

THEORY

In the first subsection we show how to generalize the log-odds score to the case of profile–profile comparison of profile HMMs. In the next subsection we then derive an efficient method to find the alignment between two profile HMMs that maximizes the log-sum-of-odds score.

Log-sum-of-odds score

The log-odds score for sequence–profile or sequence–HMM comparison has proven to be highly successful in homology recognition. This is underscored by the fact that virtually all sequence–profile and sequence-HMM comparison methods are based on it (Barrett et al., 1997). The log-odds score is a measure for how much more probable it is that a sequence is emitted by an HMM rather than by a random null model. More specifically, we write the probability for emitting the sequence $x_1, \ldots, x_L$ along the path through an HMM (Fig. 1a) by $P(x_1, \ldots, x_L | \text{emission on path})$. This probability is a product of the amino acid emission probabilities for each state on the path and the transition probabilities between states. The log-odds score for the sequence to be emitted

...
along the path by the HMM is (Durbin et al., 1998)

\[ S_{LO} = \log \frac{P(x_1, \ldots, x_L | \text{emission on path})}{P(x_1, \ldots, x_L | \text{Null})} . \] (1)

The denominator is the probability of the standardly used null model, \( P(x_1, \ldots, x_L | \text{Null}) = \prod_{i=1}^{L} f(x_i) \), where \( f(a) \) are the fixed amino acid background frequencies. We would like to generalize the log-odds score for sequence–HMM comparison to the case of HMM-HMM comparison. Suppose we are given an alignment of two profile HMMs, \( q \) and \( p \) (Figure 1b). This alignment corresponds to a certain path through the two HMMs along which the HMMs emit amino acid residues. A natural generalization of eq. (1) to the case of HMM-HMM comparison is the log-sum-of-odds (LSO) score

\[ S_{LSO} = \log \sum_{x_1, \ldots, x_L} \frac{P(x_1, \ldots, x_L | \text{co-emission on path})}{P(x_1, \ldots, x_L | \text{Null})} . \] (2)

The sum over \( x_1, \ldots, x_L \) runs over all sequences of \( L \) residues that can be emitted along the alignment path through the HMMs (e.g. \( L = 6 \) in Figure 1b). The numerator is the probability that \( x_1, \ldots, x_L \) is co-emitted by both HMMs along the alignment path and the denominator is the same null model probability as before. The log-sum-of-odds score generalizes the log-odds score: when we use one of the HMMs to represent a single sequence, i.e. the HMM can emit only this single sequence, only one term can contribute in the sum and we get the same result as with eq. (1). Note that the omission of the null model probability in the denominator would yield the logarithm of the co-emission probability.

In order to apply the Viterbi algorithm (i.e. dynamical programming) to find the path through the two HMMs with the maximum log-sum-of-odds score, we need to be more explicit about what eq. (2) means in terms of HMM probabilities. Let the two HMMs \( q \) and \( p \) have probabilities \( q_i(a) \) and \( p_j(a) \) to emit amino acid \( a \) in match state \( i \) or \( j \) and transition probabilities \( q_i(X, X') \) and \( p_j(Y, Y') \) to go from state \( X \) or \( Y \) to state \( X' \) or \( Y' \) in \( \{M, I, D\} \). Insert states emit amino acids according to the fixed amino acid background frequencies \( f(a) \).

Suppose we are given an alignment of \( q \) with \( p \), or rather the path \( P \) through the two HMMs (Figure 1b). We define \( K \) as the number of columns of the alignment of \( q \) with \( p \) (e.g. \( K = 7 \) in Figure 1b). Let the \( X_k, Y_k \in \{M, I, D\} \) be the states in \( q \) and \( p \) in the \( k \)th column of the pairwise alignment of \( q \) and \( p \) and let \( i(k) \) and \( j(k) \) be the respective columns from \( q \) and \( p \). For the \( L \) residues emitted along the path, we define \( q_{i(k)}^{P}(a) \) and \( p_{j(k)}^{P}(a) \) as the emission probabilities from \( q \) and \( p \). More explicitly, \( q_{i(k)}^{P}(a) = q_i(a) \) for \( X_k = M \) and \( q_{i(k)}^{P}(a) = f(a) \) for \( X_k = I \). Finally, we define \( P_{tr} \) as the product of all transition probabilities for the path through \( p \) and \( q \). With these definitions, we can rewrite the log-sum-of-odds score as

\[ S_{LSO} = \log \sum_{x_1, \ldots, x_L} \frac{\prod_{k=1}^{K} q_{i(k)}^{P}(x_k)p_{j(k)}^{P}(x_k)}{\prod_{i=1}^{L} f(x_i)} \times P_{tr} \]

\[ = \log \sum_{x_1, \ldots, x_L} \prod_{k=1}^{K} q_{i(k)}^{P}(x_k)p_{j(k)}^{P}(x_k) f(x_i) \times P_{tr} \]

\[ = \log \sum_{x_1, \ldots, x_L} \prod_{k=1}^{K} q_{i(k)}^{P}(x_k) p_{j(k)}^{P}(x_k) + \log P_{tr} . \] (3)

In the last line we have introduced the column score,

\[ S_{aa}(q_i, p_j) = \log \sum_{a=1}^{20} q_i(a)p_j(a) f(a) . \] (4)

by which we compare the amino acid distributions from the two HMMs. If we omitted the factor \( 1/f(a) \), we would obtain the logarithm of the co-emission probability as the total score. In this respect, the \( 1/f(a) \) can be interpreted as weight factors to the co-emission probability. They increase the weight of the rare amino acids with respect to the more common ones. This makes sense since co-emission of rare amino acids is harder to produce by chance. When one profile column contains only amino acid \( x_i \), i.e. \( q_i(x_i) = 1 \), the \( 1/f(a) \) ensures that we retrieve the log-odds score \( S_{aa}(q_i, p_j) = \log(p_j(x_i)/f(x_i)) \). Furthermore, when one of the columns is completely nonconserved, \( p_j(a) = f(a) \), the column score vanishes. For the same reason, insert states have vanishing column scores. The column score is positive when the two distributions are similar and negative otherwise, a property that makes local alignment possible. The column score is symmetric and furthermore fast to evaluate since it contains only one logarithm.

**Pairwise alignment of HMMs**

A profile HMM contains in each column a match state \( M \), a delete state \( D \) and an insert state \( I \) (Figure 1a). Match states and insert states emit amino acids whereas delete states don’t. Therefore a match or insert state in one HMM can only be aligned with a match or insert state in the other HMM. Conversely, a delete state can only be aligned with a delete state or with a Gap \( G \) (Figure 1b). A gap in a pairwise alignment of HMMs is completely analogous to a gap in a pairwise sequence alignment. It signifies that the column of the other HMM that is aligned with the gap does not have a homologous partner.\(^3\) We denote the alignment pair states as \( MM, MI, IM, II, DD, DG, \) and \( GD \). Figure 1b shows an example of two aligned profile HMMs. In the third column HMM \( q \) emits a residue from its \( M \) state and

\(^3\)Residues or columns from multiple alignments are homologous if they evolved from the same residue in an ancestral sequence.
HMM $p$ emits a residue from the $I$ state. The pair state for this alignment column is $MI$. In column six of the alignment HMM $q$ does not emit anything since it passes through the $D$ state. HMM $p$ does not emit anything either since it has a gap in the alignment. The corresponding pair state is $DG$. In principle, pair states $MI$ and $DG$ can be interchanged without changing the alignment of the two HMMs. The reason why we distinguish between them is that changing the path through the two HMMs changes the transition probabilities that contribute to the total score (eq. 2).

At this point, we make two simplifications that speed up the algorithm and that can be argued to have a negligible or even positive effect on its performance: First, we exclude pair states $II$ and $DD$, and second, we only allow transitions between a pair state and itself and between pair state $MM$ and pair states $MI$, $IM$, $DG$, or $GD$ (Figure 1c). The reasoning is very similar to the case of neglecting the $I \rightarrow D$ and $D \rightarrow I$ transitions in profile HMMs (Durbin et al., 1998).

To calculate the log-sum-of-odds score according to equation (3), we need five dynamical programming matrices $S_{XY}$, one for each pair state $XY \in \{MM, MI, IM, DG, GD\}$. They contain the score of the best partial alignment which ends in column $i$ of $q$ and column $j$ of $p$ in pair state $XY$. These matrices are calculated recursively,

$$S_{MM}(i,j) = S_{aa}(q_i, p_j) +$$

$$\max \begin{cases} S_{MM}(i-1,j-1) + \log(q_{i-1}(M,M)p_{j-1}(M,M)) \\ S_{MI}(i-1,j-1) + \log(q_{i-1}(M,M)p_{j-1}(I,M)) \\ S_{IM}(i-1,j-1) + \log(q_{i-1}(I,M)p_{j-1}(M,M)) \\ S_{DG}(i-1,j-1) + \log(q_{i-1}(D,M)p_{j-1}(M,M)) \\ S_{GD}(i-1,j-1) + \log(q_{i-1}(M,M)p_{j-1}(D,M)) \end{cases}$$

and similarly for $S_{IM}(i,j)$ and $S_{GD}(i,j)$. Note that in the last equation no transition probabilities for HMM $p$ appear. The pair state $DG$ that is joined to the best partial alignment by this equation has a gap in HMM $p$ and therefore no new transition is added to the path through $p$ (Figure 1b).

We have implemented both a semi-global and a local alignment version in HHsearch. For semi-global alignment the terminal gaps are not scored, so we set $S_{MM}(i,0) = S_{MM}(0,j) = 0$. The other four matrices are initialized to $-\infty$ to forbid any pair state except $MM$ as the first state. The total score $S_{LSO}$ is the maximum over the last column and last row of $S_{MM}$. For local alignment a zero is added as a sixth case to the maximization in equation (5) to permit the HMM–HMM alignment to start at any $MM$ pair state without penalty. The total score $S_{LSO}$ is found as the maximum over the whole matrix $S_{MM}$. The optimal alignment is constructed as usual by backtracing from the cell with maximum score.

**Score offset**

Most profile–profile methods add a score offset to the column score $S_{aa}$ in order to adjust how greedily the alignments will be constructed (Wang and Dunbrack, 2004), negative offsets producing shorter alignments. We found that adding a small offset of $-0.1$ bits indeed improves the performance of HHsearch and we use it as a default parameter. We think that this suppresses false matches caused by compositional bias, i.e. by a global similarity in amino acid composition. This compositional bias can lead to per-column
scores slightly above zero (but in general below 0.1) which can add up to appreciable total scores over long proteins. We have also experimented with more refined methods for compositional bias correction. We replaced the background frequencies \( f(a) \) in eq. (4) by the average amino acid frequencies in the query or target protein, \( \hat{q}(a) \) or \( \hat{p}(a) \), for example, but found the simple offset method to work best.

**Sequence weighting and pseudocounts**

For sequence weighting, we use the scheme of PSI-BLAST (Altschul et al., 1997) which is a modified version of Henikoff and Henikoff’s scheme (Henikoff and Henikoff, 1994). We add amino acid pseudocounts to both HMMs with a substitution matrix method similar to PSI-BLAST (Altschul et al., 1997), employing the Gonnet matrix (Gonnet et al., 1992) in place of the BLOSUM62 matrix as default. In contrast to the scheme of PSI-BLAST, the pseudocount admixture depends on the position in the multiple alignment. The modification ensures that, as in the sequence weighting scheme, alignments that are composed of several (sub)domains and which contain many sequences that cover only parts of the alignment get transformed to profiles in the same way as if the alignment was first cut into (sub)domains and the profiles calculated separately. Transition pseudocounts are added in a way analogous to amino acid pseudocounts.

**Scoring correlations**

It was shown by Pei et al. that in alignments of homologous sequences conserved columns tend to occur in clusters along the sequence (Pei and Grishin, 2001). When applied to the alignment of homologous HMMs, conserved columns of the underlying super-alignment should also occur in clusters. The conservation score of the super-alignment constructed from the two alignments will be higher wherever the distributions in the two aligned columns are similar, or, in other words, wherever the column score \( S_{aa} \) (eq. 4) is high. To sum up, in an alignment of two homologous HMMs we expect high column scores to occur in clusters along the sequence whereas in an alignment of non-homologous HMMs we do not expect any clustering.

This observation can help to distinguish homologous from non-homologous alignments. Suppose the \( l \)th pair state of the optimum path aligns columns \( i(l) \) from \( q \) and \( j(l) \) from \( p \). We write \( S_l \) for the column score of the \( l \)th pair state, i.e. \( S_l = S_{aa}(q_{i(l)}, p_{j(l)}) \) if the \( l \)th pair state is an MM state and zero otherwise. The autocorrelation function

\[
g(d) = \sum_{l=1}^{L-d} S_l S_{l+d}
\]

(8)
describes the correlation of \( S_l \) at fixed sequence separation \( d \). When the two HMMs are homologous we expect \( g(d) \) to be positive for small \( d \). We therefore add

\[
S_{corr} = w_{corr} \sum_{d=1}^{4} g(d)
\]

(9)
to the total score, after the best alignment is found\(^4\). The weight \( w_{corr} = 0.1 \) was determined empirically on a small test set of \( 317 \times 317 \) pairwise alignments.

**Scoring secondary structure**

HHsearch allows to score predicted secondary structure either against predicted secondary structure or against known secondary structure. We first treat the latter case which is applicable to 3D structure prediction. The goal is a statistical score for aligning a pair of secondary structure states that takes the confidence values of the secondary structure prediction into account. Intuitively, the confidence values contain very valuable information since, for example, an \( H \) aligned to a predicted \( E \) should be penalized much more when the confidence value is 9 instead of 0.

We use DSSP (Kabsch and Sander, 1983) to assign one of seven states of observed secondary structure. PSIPRED is employed to predict secondary structure states \( H, E, \) and \( C \) (Jones, 1999). We predicted the secondary structure for all domains in SCOP (version 1.63, filtered to a maximum sequence identity of 20%) and compared the PSIPRED predictions with the DSSP assignments.

We counted how often each combination \( (\sigma, \rho, c) \) occurred in which a DSSP state \( \sigma \in \{ H, E, C, G, B, S, T \} \) was predicted by PSIPRED as state \( \rho \in \{ H, E, C \} \) with confidence value \( c \in \{ 0, 1, \ldots, 9 \} \). From this we calculated the probability \( P(\sigma; \rho, c) \) for \( \sigma \), \( \rho \) and \( c \) to occur together, as well as the probability \( P(\sigma) \) for \( \sigma \) to occur and the probability \( P(\rho, c) \) for the pair \( (\rho, c) \) to occur. In this way we derived ten \( 3 \times 7 \) substitution matrices, one for each value of \( c \):

\[
M_{SS}(\sigma; \rho, c) = \log \frac{P(\sigma; \rho, c)}{P(\sigma) P(\rho, c)}.
\]

(10)

Now suppose column \( i \) of HMM \( q \) has predicted secondary structure \( \rho_i \) and confidence value \( c_i \) and column \( j \) of HMM \( p \) has known secondary structure\(^5\) \( \sigma_j \). The secondary structure score for \( q_i \) and \( p_j \) is obtained by multiplying the logarithms in \( M_{SS} \) with a weight \( w_{SS} \),

\[
S_{SS}(q_i, p_j) = w_{SS} M_{SS}(\sigma_j^p; \rho_i^q, c_i)
\]

(11)

This score is added to the amino acid column score \( S_{aa}(q_i, p_j) \) in equation (5). The weight coefficient \( w_{SS} \) accounts for the fact that the secondary structure states are not independent of their neighbors. Since the average length of

\( ^{4} \) We devised an alignment algorithm that maximizes the total score including the correlation score but the performance was only marginally better at approximately twice the computation time.

\( ^{5} \) The known secondary structure of \( p \) is the secondary structure of the seed sequence of the alignment.
Protein homology detection by HMM–HMM comparison

stretches of identical states of predicted secondary structure is \( \sim 7 \) we expect an optimum weight \( w_{SS} \approx 1/7 \). Empirically we indeed find a broad optimum around \( w_{SS} = 0.15 \), the value we use in this benchmark.

Note that matrix \( M_{SS} \) only quantifies how actual secondary structure is correlated with predicted secondary structure for one profile HMM. What is missing is a matrix that quantifies the mapping of the actual secondary structure from one HMM to another distantly related HMM. We derived such a \( 7 \times 7 \) matrix and provided it with a variable exponent, but we find that an exponent of zero represents the optimum case, which is why this matrix was omitted in the following.

We now come to the case of scoring predicted against predicted secondary structure. This time we need to account for the mapping of DSSP secondary structure states to predicted secondary structure. We now come to the case of scoring predicted against predicted predictions.

What is missing is a matrix that quantifies the value we use in this benchmark.

\[ w_{c} \text{ substitution matrix for the alignment of states } (\rho^q, c^q) \text{ and } (\rho^p, c^p) \]

\[ M_{SS}(\rho^q_i, c^q_i, \rho^p_j, c^p_j) = \log \frac{P(\rho^q_i, c^q_i; \rho^p_j, c^p_j)}{P(\rho^q_i, c^q_i) P(\rho^p_j, c^p_j)} \]

where the sum runs over all seven DSSP states. This matrix tells us how much more probable it is to obtain predictions \((\rho^q_i, c^q_i)\) and \((\rho^p_j, c^p_j)\) for a pair of aligned homologous residues than to obtain them independently of each other, whatever the actual secondary structure state \( \sigma \) may be. The secondary structure score calculated from this matrix by \( S_{SS}(q, p) = w_{SS} M_{SS}(\rho^q_i, c^q_i, \rho^p_j, c^p_j) \) is added to the column score with the same weight \( w_{SS} = 0.15 \) as before.

RESULTS AND DISCUSSION

We have performed an all-against-all comparison with various similarity search tools to test their ability to detect remote homologs and to produce high-quality alignments below the twilight zone (Doolittle, 1981) of sequence similarity. We compared BLAST and PSI-BLAST (version 2.2.9) as popular representatives of sequence–sequence and profile–sequence methods, the HMM–sequence comparison package HMMER (2.2g), the profile–profile alignment tools PROF_SIM (obtained 04/02/2004) and COMPASS (1.24), and our method HHsearch (1.0). All tools except COMPASS were run with default parameters.

Detection of homologs

Each domain in SCOP is classified into a hierarchy of family, superfamly, fold, and class (Murzin et al., 1995). Domains within one family are clearly homologous, based either on a sequence identity \( > 30\% \) or on a very similar structure and function at lower sequence similarity. Domains from the same superfamily but different family are likely to be homologous based on an expert analysis of structural and sequence similarity, location of binding sites, functional groups etc. Domains in the same fold but different superfamilies share the same spatial arrangement and connectivity of secondary structure elements. They may be similar either by common descent or by convergence. Following SCOP, we classify each pair of domains as homologous if they are members of the same superfamily. Domains from different classes are classified as non-homologous. All other pairs are considered as “unknown” in the benchmark since their evolutionary relationship can not be ascertained.

Figure 2 is a classical chart with the number of true positives (TP) versus the number of false positives (FP). True positives are homologous pairs and false positives are non-homologous pairs with a score above a certain threshold. By varying the threshold score the curve of TP versus FP is traced out. The ideal method would detect all homologs before the first non-homologous pair is reported. The curve would rise up vertically from zero until it reached the total number of homologous pairs.

Starting from the bottom, we see that BLAST is obviously inadequate to search for homologies in such a difficult dataset. At a rate of false positives (“error rate”)

In order to pinpoint the source of improvements, we benchmarked four versions of HHsearch. HHsearch 0 uses simple profile–profile comparison by setting all gap opening penalties to -3.5 bits and all gap extension penalties to -0.2 bits and using these instead of the logarithms of the transition probabilities in eqs. (5-7). HHsearch 1 is the basic HMM–HMM version, HHsearch 2 includes the correlation score (eq. 9), in addition to this HHsearch 3 compares predicted with predicted secondary structure (eq. 12), and HHsearch 4 uses predicted versus known secondary structure (eq. 10).

The 3691 sequences of the SCOP database (Murzin et al., 1995) (version 1.63) filtered to a maximum sequence identity of 20% (“SCOP-20”) were obtained from the ASTRAL server (Chandonia et al., 2004). Each sequence corresponds to a single structural domain, except for 73 sequences from the SCOP class of multi-domain proteins. An alignment was built from each seed sequence by PSI-BLAST with up to eight iterations. An inclusion threshold of \( 10^{-4} \) in the last iteration and \( 10^{-5} \) in previous iterations was used. We used several filters in order to make sure that only homologous sequences enter the alignments. All methods in the benchmark (except BLAST) were tested with this same set of alignments.
Figure 2: Sensitivity of various homology detection tools, measured by how many true positives are detected at varying numbers of false positives in an all-against-all benchmark on SCOP-20. True positives are pairs from the same superfamily, false positives are pairs from different classes. Dashed straight line: error rate 10%. There are 41505 true positives and $1.08 \times 10^7$ false positives in total. For definitions of HHsearch 0–4 please refer to the main text.

\[
FP/(TP + FP) = 10\% \text{ (dashed line) it finds only 908 homologous pairs, or 2.2\% out of a total of 41505. PSI-BLAST detects 17.7\% and HMMER finds 18.7\%. PROF\_SIM and COMPASS find 24.9\% and 34.0\%, respectively. Next in performance is HHsearch 0 with 40.0\% and the basic HMM–HMM version HHsearch 1 with 44.2\%. Inclusion of the correlation score (eq. 9) improves this value to 46.7\% (HHsearch 2). When in addition predicted secondary structure is used for both HMMs, a value of 48.8\% is achieved (HHsearch 3).}

All of the HHsearch versions in Figure 2 use local alignment. We found that the semi-global version did not perform nearly as well (data not shown). We believe that this is owed to the fact that distant homologs are often not alignable over their entire length but only over a core that defines their superfamily. The semi-global algorithm aligns these non-homologous regions by force which leads to random noise added to the score of the aligned homologous regions.

In an analysis of the complete data we found many pairs of sequences from different superfamilies and sometimes even different folds that HHsearch predicts as homologs with high confidence. In most cases their structures are also very similar, either in parts or globally. This convinced us that many superfamilies that are classified by SCOP into different folds are in fact homologous. We name just two examples, the TIM barrels (Henn-Sax et al., 2001) (SCOP superfamilies c.1.1 – c.1.25) and the beta propellers (SCOP folds b.66 – b.70). To test how well the various methods detect these cases of structural similarity and putative homology, we analyze the data with a second, alternative definition of true and false positives. A pair is now defined as true positive if the domains belong to the same SCOP superfamily or if the sequence-based alignment yields a structural alignment with a MaxSub score (Siew et al., 2000) of at least 0.1. Pairs of sequences from different classes and with zero MaxSub score are classified as non-homologous. All other relationships are classified as unknown. Roughly speaking, the MaxSub score tells us what fraction of the query residues can be structurally superposed with the aligned residues from the other structure. It is defined such that a score $>0$ occurs rarely by chance. More specifically, MaxSub equals the weighted number of aligned pairs that can be superimposed with a maximum distance per pair of 3.5 Å, divided by the number of residues in the query sequence. Pairs with 0 Å deviation carry weight 1 and pairs with 3.5 Å deviation have weight 0.5. If no subset with 40 or more aligned residue pairs can be found that are within 3.5 Å and if no more than 25 such pairs can be found with score $\geq 0.125$ the MaxSub score is set to 0.

Figure 3: Same as previous figure, but for a broader definition of true positives: True positives are pairs from the same superfamily or with MaxSub score of at least 0.1, false positives are pairs from different classes and zero MaxSub score.
more: The scoring matrix for this case (eq. 12) embodies mologies for which the secondary structures have diverged secondary structure makes it better optimized for remote that the way in which we score predicted versus predicted secondary structure is used in one HMM. The likely reason is 4 decreases slightly when known instead of predicted sec- Interestingly, the sensitivity for HHsearch in Figure 3 and quickly with increasing evolutionary divergence. in sensitivity due to inclusion of secondary structure grows 2.2 times more than COMPASS. Note that the improvement 4 which use secondary structure information are now HHsearch 0. Whereas HHsearch 0 was 18% more sensitive than COMPASS in figure 2 it is only 6% more sensitive now (at 10% error rate). This could mean that they reach their peak performance at more remote relationships than HHsearch. To test this hypothesis we plot the number of true versus false positives again (Figure 4), but this time we keep only the true positive pairs from different families. Indeed PROF_SIM further improves with respect to HHsearch 0 and COMPASS even draws equal. Remarkably, HHsearch 3 and 4 which use secondary structure information are now much more sensitive (~50%) than HHsearch 2. At an error rate of 10%, HHsearch 3 detects a factor of 190 more true positives than BLAST, 7.5 and 7.2 times more than PSI- BLAST and HMMER, 4.0 times more than PROF_SIM, and 2.2 times more than COMPASS. Note that the improvement in sensitivity due to inclusion of secondary structure grows quickly with increasing evolutionary divergence. Interestingly, the sensitivity for HHsearch in Figure 3 and 4 decreases slightly when known instead of predicted secondary structure is used in one HMM. The likely reason is that the way in which we score predicted versus predicted secondary structure makes it better optimized for remote homologies for which the secondary structures have diverged more: The scoring matrix for this case (eq. 12) embodies twice as much uncertainty as the scoring matrix for known versus predicted secondary structure (eq. 10).

Alignment quality

The quality of an alignment between a query protein and a distant homolog is critical to its usefulness for structure prediction, evolutionary studies, and functional analysis. In comparative modeling, for example, the alignment between query and template is the key determinant of model quality (Venclovas, 2003). The quality of sequence alignments can be assessed by comparing them with reference alignments generated by structural alignment algorithms (‘the gold standard’). Here we employ a more direct approach, developed for the automatic assessment of structure prediction servers (Siew et al., 2000), where the generation of a structure-based sequence alignment is omitted as an intermediate step. One thus avoids the arbitrariness involved in transforming a structural superposition into a sequence alignment (see also O’Sullivan et al., 2003). Instead, the sequence alignment is assessed directly by looking at the spatial distances between aligned pairs of residues upon superposition of their 3D structures.

We use two scores for alignment quality. The first is the plain MaxSub score. A drawback of this score is that it does not penalize overprediction: pairs of residues that are wrongly predicted to be superposable are not penalized at all. A method optimized for this score will generate alignments of maximal length even when only a few residues can be reliably aligned. Similar to the MaxSub score is the developer’s score, $S_{Dev} = N_{correct} / \min(L_q, L_p)$, where $N_{correct}$ is the number of residue pairs that is present in the maximum subset identified by MaxSub, and $L_q$ and $L_p$ denote the number of residues in the two sequences to be aligned.

At the other extreme, the so-called modeler’s score does not penalize underprediction of residues. It is defined as $S_{Mod} = N_{correct} / L_{ali}$, where $L_{ali}$ is the number of aligned residue pairs in the sequence alignment. A method optimized for this score alone would always predict just one pair of aligned residues.9

As the golden mean between these two extremes we define a “balanced” score which penalizes both overprediction and underprediction:

$$S_{balanced} = \left( S_{Dev} + S_{Mod} \right)/2.$$  

We set $S_{balanced}$ to zero when the maximum subset contains less than 40 residue pairs. As for the MaxSub score, this ensures that a score larger than zero is unlikely to occur by chance. Other balanced scores have been proposed by Cline et al. (2002) and Yona and Levitt (2002).

9The developer’s score and the modeler’s score were first defined by (Sauder et al., 2000) in a slightly different way using the structure-based alignments as gold standard. In their definition, $\min(L_q, L_p)$ is replaced by the number of aligned residue pairs in the structural alignment and $N_{correct}$ refers to the number of residues which are aligned in the same way as in the structure-based sequence alignment.
Figure 5: Distribution of MaxSub scores for alignments of domain pairs related at the family, superfamily, and fold level in percent of the total number of homologous sequence pairs at that level of relationship. Counts with MaxSub score of exactly zero are not shown.

Figure 5a–c plots the binned distribution of MaxSub scores for all pairs related at the family level (10223 pairs), superfamily level (31282 pairs), and fold level (66813 pairs). First of all, note that PSI-BLAST produces much better alignments than BLAST. Second, the group of profile–profile methods PROF_SIM, COMPASS, and HHsearch 0 perform clearly better than PSI-BLAST, especially at the superfamily level. Third, within this group COMPASS is a little better than PROF_SIM and HHsearch 0 is a little better than COMPASS at all levels of relationship. Fourth, aligning profile HMMs (HHsearch 2) instead of simple profiles (HHsearch 0) improves the alignment quality significantly, especially for the difficult alignments on the superfamily and fold levels. Fifth, adding predicted secondary structure greatly improves alignment quality on the superfamily and fold levels. Sixth, as a general trend the good methods get even better relative to the others with increasing difficulty of the alignments, the same as was observed for the sensitivity in Figures 2 to 4. Last, HMMER alignments have better MaxSub scores than the simple profile–profile methods because HMMER is run in its default global mode and MaxSub does not penalize overpredicted residues.

Figure 6: Distribution of balanced scores (eq. 13) for alignments of domain pairs related at the family, superfamily, and fold level. Counts with zero score are not shown.

Figure 6a–c plots the binned score distribution for the balanced score defined in equation (13). The points discussed with the previous figure are borne out here, with the exception that HMMER now comes out as inferior to the profile–profile methods, as it should. HHsearch 3 is the clear winner. At the family level, it aligns 58% of all pairs with a balanced score of 0.3 or larger. This is 1.23 times more than COMPASS, 1.28 times more than PROF_SIM, 1.34 times more than HMMER, 1.57 times more than PSI-BLAST, and 4.4 times more than BLAST. At the superfamily level, where 27% of HHsearch 3 alignments have a score of 0.3 or above, the improvement over the other tools is by a factor 1.7 (COMPASS), 1.9 (PROF_SIM), 2.2 (HMMER), 2.9 (PSI-BLAST), and 14 (BLAST). At the fold level, where 4.5% of HHsearch 3 alignments have a score of 0.3 or above, the fac-
tors are 3.3 (COMPASS), 6.0 (PROF_SIM), 7.3 (HMMER), 9.4 (PSI-BLAST), and 63 (BLAST). In several recent benchmark studies, column scores for profile–profile alignment were compared for their ability to produce alignments similar to structure-based alignments (Mittelman et al., 2003; Panchenko, 2003; Martí-Renom et al., 2004; Edgar and Sjölander, 2004). In these studies differences in performance between the tested column scores are generally small and no clear winner has emerged. Indeed, the quality of alignments produced by HHsearch 0, COMPASS, and PROF_SIM is rather similar in our benchmark. In this light the improvements by HMM–HMM alignment (HHsearch 2) and secondary structure scoring (HHsearch 3 or 4) is all the more remarkable and shows that they matter much more than the choice of column score.

Structure prediction

When predicting structure we are allowed to use the best match with a sequence in the database, whereas Figures 5 and 6 show the score distribution for all pairs at a given level. Figure 7a shows the score distribution of the best match in each of the 3691 database scans. Figure 7b plots the alignment score distribution of the best matches at or below the superfamily level, i.e. where members from the same family have been excluded as templates. Similarly, Figure 7c shows the score distribution for pairs at or below the fold level. For structure prediction the true secondary structure of the template is available and HHsearch 4 can be used. We also show the results for HHsearch 4g, which is the same as HHsearch 4 except that the alignments have all been realigned with the semi-global algorithm. The success in this figure depends on a combination of alignment quality and sensitivity per database scan because the more sensitive a method is the better it will be able to rank the best 3D template at the top.

Indeed, the performance in Figure 7 is explicable largely by a combination of sensitivity and alignment quality. HHsearch 4 is again much better than COMPASS and PROF_SIM. COMPASS and PROF_SIM are much better than PSI-BLAST and HMMER due to their much higher sensitivities. A bit surprisingly, PROF_SIM is better than COMPASS on the superfamily level and particularly so at the fold level. We think that the method of calculating its P-values is the cause for PROF_SIM’s rather sub-optimal sensitivity in Figures 2 to 4. On a per-scan basis it seems to be even better than COMPASS in ranking the best structural templates at the top, at least below the family level. Finally, HHsearch 4g with its global alignments fares a bit better than HHsearch 4.

What chances does one have to get a structural template with a usable alignment? If a template from the same family as the query is available in the database HHsearch 4 will produce a usable alignment with MaxSub score ≥ 0.1 in 66% of all cases, COMPASS and PROF_SIM in 56% of all cases. When the closest relative in the structure database is from the same superfamily, a usable alignment is produced in 44% (HHsearch 4), 29% (COMPASS) and 31% (PROF_SIM) of all cases. When the most closely related structure has the same fold, HHsearch 4 can still come up with an alignment with a score of at least 0.1 in 19% of all cases, COMPASS in 7.3% and PROF_SIM in 9.7%.

CONCLUSION

We have generalized HMM–sequence alignment to the pairwise alignment of profile HMMs and presented a fast algorithm that maximizes the log sum-of-odds score, the gener-
alization of the well-known log-odds score. A novel correlation score was derived which increases sensitivity by 5-10% at no cost and which can easily be applied to other similarity search methods. Moreover, we have proposed a statistical method to score predicted versus known secondary structure as well as predicted versus predicted secondary structure that exploits the confidence values of the secondary structure prediction.

Based on these methods, we have developed the homology detection tool HHsearch which we benchmarked together with five other homology detection tools on a hard data set below the twilight zone of sequence similarity (20% sequence identity). HHsearch represents a significant improvement over existing methods, both in terms of sensitivity and alignment quality, and the contributions to this improvement were analyzed.

Two servers (HHpred.2/3) that use HHsearch have been registered for the blind structure prediction contests CAFASP4 (Fischer et al., 2003) and LiveBench (Rychlewski et al., 2003). Preliminary results are below our expectations and indicate that the multiple alignment construction method rather than HHsearch limited the performance, since it was geared too much to high selectivity at the cost of sensitivity. We plan to improve this by using separate alignment databases for structure prediction and homology detection. We hope that HHsearch will be a useful tool for functional annotation, structure prediction, and protein evolution. We have set up a web server for homology detection and structure prediction that we plan to extend into a structure and function prediction pipeline with maximum flexibility for manual use. But the speed of HHsearch should also allow an application to large-scale automatic annotation projects and any requests in this direction are welcome.

ACKNOWLEDGEMENTS

I am grateful to Andrei Lupas for many fruitful discussions, mentoring and encouragement. Many thanks go to Daniel Huson for critically reading the manuscript, to Ruslan Sadreyev and Golan Yona for making their tools COMPASS and PROF_SIM available, and to an anonymous referee for his helpful comments.

References