BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 7 2005, pages 951–960
doi:10.1093/bioinformatics/bti125

Sequence analysis

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

Received on July 7, 2004; revised on October 18, 2004; accepted on November 2, 2004
Advance Access publication November 5, 2004

ABSTRACT

Motivation: Protein homology detection and sequence alignment are 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 the 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, superfamily and fold level, respectively.

Speed: HHsearch scans a query of 200 residues against 3691 domains in 33 *s* on an AMD64 2GHz PC. This is 10 times faster than PROF_SIM and 17 times faster than COMPASS.

Availability: HHsearch can be downloaded from http://www.protevo.eb.tuebingen.mpg.de/download/ together with up-to-date versions of SCOP and PFAM. A web server is available at http://www.protevo.eb.tuebingen.mpg.de/toolkit/index.php?view=hhpred

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 led 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 non-conserved 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 for identifying 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; Ginalski *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 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 inserts and deletions along the alignment (Fig. 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*. (1999) developed an algorithm for the alignment of two HMMs based on the exact maximization of the co-emission probability. 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 alignment that is based on this theory and that makes some simplifications for increased efficiency. We show that by aligning profile HMMs instead of simple sequence profiles we are able to improve both sensitivity and alignment quality significantly.

Even sequences that are only distantly homologous will have secondary structures more similar to each other than what is to be expected by chance. For this reason the predicted secondary structure can help to distinguish real homologs from chance hits. Several methods score secondary structure to improve homology recognition. Kelley *et al*. (2000) score secondary structure with a simple $+1/-1$ scoring function. Hargbo and Elofsson (1999) include predicted probabilities to emit one of three secondary structure states in their profile HMMs.[1] Kawabata and Nishigawa (2000) developed a statistical approach using a 3×3 substitution matrix for secondary structure states in their structure comparison program Matras. For HHsearch we developed a statistical method which aims at exploiting all available information, including the confidence values and the full seven-state secondary structure determined by DSSP[2] (Kabsch and Sander, 1983). Like Kawabata and Nishigawa, we score pairs of aligned secondary structure states with substitution matrices, but we use ten 3×7 matrices, one for each confidence value, that we derive from a statistical analysis of PSIPRED predictions.

Our motivation in developing HHsearch was to provide the scientific community with a powerful tool for remote homology detection which maximizes sensitivity while ensuring reliability, speed and ease of use. To achieve maximum sensitivity we include as much information about query and database sequences as possible. We use 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 1.4×10^7 pairwise comparisons of our benchmark.

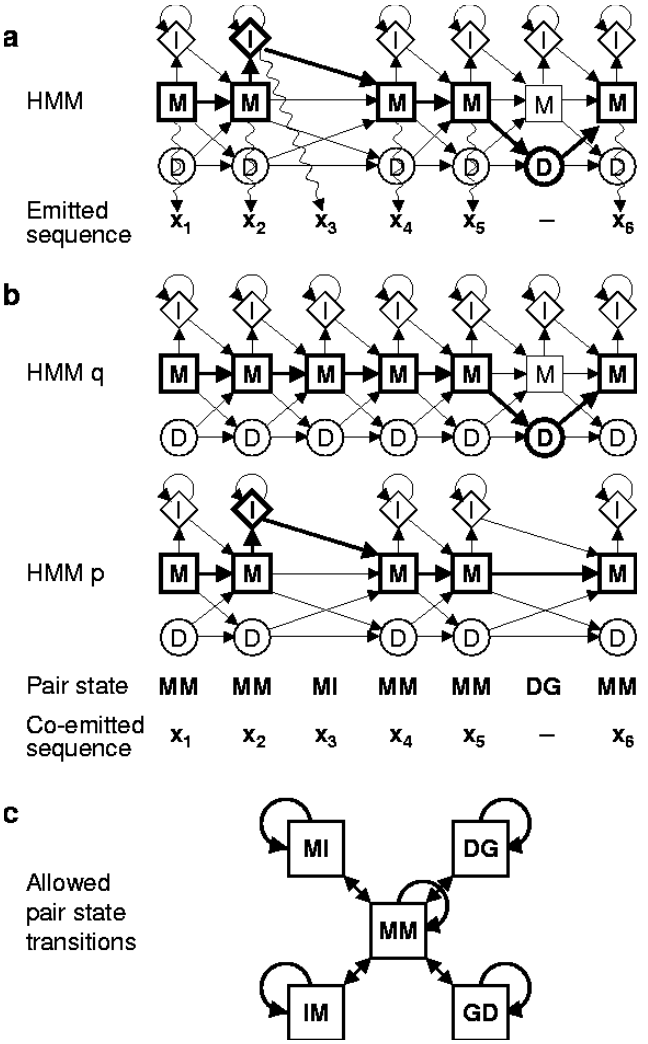

Fig. 1. (**a**) The alignment of a sequence to a profile HMM can be represented by a path through the HMM (bold arrows). (**b**) Alignment of two HMMs by maximization of log-sum-of-odds score. The path through the two HMMs corresponds to a sequence that is co-emitted by both HMMs. With dynamical programming one finds the path which maximizes the log-sum-of-odds score [Equation (2)]. (**c**) Allowed transitions between pair states. Other transitions are possible but can be neglected.

THEORY

In the first subsection we show how to generalize the log-odds score to the case of pairwise comparison of profile HMMs. In the next subsection we then derive an efficient method to find the alignment

[1] A problem with this approach is that the probabilities given for example by PSIPRED (Jones, 1999) do not represent a probability distribution since they do not sum to one.

[2] The π-helix is very rare and is mapped to the coil state.

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)

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

The denominator is the probability of the standardly used null model, $P(x_1, \ldots, x_L|\text{Null}) = \prod_{l=1}^{L} f(x_l)$, 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 (Fig. 1b). This alignment corresponds to a certain path through the two HMMs along which the HMMs emit amino acid residues. A natural generalization of Equation (1) to the case of HMM–HMM comparison is the log-sum-of-odds (LSO) score

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

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 Fig. 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 Equation (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 Equation (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 $Y \in \{M, I, D\}$ in column i or j to a 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 (Fig. 1b). We define K as the number of columns of the alignment of q with p (e.g. $K = 7$ in Fig. 1b). Let the $X_k, Y_k \in \{M, I, D\}$ be the states in q and p in the kth 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 residues $x\_1 (1 = 1, \ldots, L)$ emitted along the path, we define $q^{\text{P}}_{k(l)}(a)$ and $p^{\text{P}}_{k(l)}(a)$ as the emission probabilities from q and p. More explicitely, $q^{\text{P}}_k(a) = q_{i(k)}(a)$ for $X_k = M$ and $q^{\text{P}}_k(a) = f(a)$ for $X_k = I$. Finally, we define $\mathcal{P}_{\text{tr}}$ 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

$$\begin{aligned} S_{\text{LSO}} &= \log \sum_{x_1,\ldots,x_L} \frac{\prod_{l=1}^{L} q^{\text{P}}_{k(l)}(x_l)\, p^{\text{P}}_{k(l)}(x_l) \times \mathcal{P}_{tr}}{\prod_{l=1}^{L} f(x_l)} \\ &= \log \sum_{x_1=1}^{20} \cdots \sum_{x_L=1}^{20} \prod_{l=1}^{L} \frac{q^{\text{P}}_{k(l)}(x_l)\, p^{\text{P}}_{k(l)}(x_l)}{f(x_l)} \times \mathcal{P}_{\text{tr}} \\ &= \log \prod_{l=1}^{L} \left(\sum_{a=1}^{20} \frac{q^{\text{P}}_{k(l)}(a)\, p^{\text{P}}_{k(l)}(a)}{f(a)} \right) + \log \mathcal{P}_{\text{tr}} \\ &= \sum_{k:X_k Y_k = MM} S_{\text{aa}}(q_{i(k)}, p_{j(k)}) + \log \mathcal{P}_{\text{tr}}. \end{aligned} \qquad (3)$$

In the last line we have introduced the column score,

$$S_{\text{aa}}(q_i, p_j) = \log \sum_{a=1}^{20} \frac{q_i(a)\, p_j(a)}{f(a)}, \qquad (4)$$

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 $1/f(a)$ ensure that we retrieve the log-odds score $S_{\text{aa}}(q_i, p_j) = \log(p_j(x_i)/f(x_i))$. Furthermore, when one of the columns is completely non-conserved, $p_j(a) = f(a)$, 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 (Fig. 1a). Match states and insert states emit amino acids whereas delete states do not. 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 (Fig. 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 HMM p emits a residue from the I state. The pair state for this

[3]Residues or columns from multiple alignments are homologous if they evolve from the same residue in an ancestral sequence.

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 [Equation (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 (Fig. 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,

$$S_{MM}(i,j) = S_{\text{aa}}(q_i, p_j) + \max \begin{cases} S_{MM}(i-1,j-1) + \log[q_{i-1}(M,M)\, p_{j-1}(M,M)] \\ S_{MI}(i-1,j-1) + \log[q_{i-1}(M,M)\, p_{j-1}(I,M)] \\ S_{IM}(i-1,j-1) + \log[q_{i-1}(I,M)\, p_{j-1}(M,M)] \\ S_{DG}(i-1,j-1) + \log[q_{i-1}(D,M)\, p_{j-1}(M,M)] \\ S_{GD}(i-1,j-1) + \log[q_{i-1}(M,M)\, p_{j-1}(D,M)] \end{cases} \tag{5}$$

$$S_{MI}(i,j) = \max \begin{cases} S_{MM}(i-1,j) + \log[q_{i-1}(M,M)\, p_j(M,I)] \\ S_{MI}(i-1,j) + \log[q_{i-1}(M,M)\, p_j(I,I)] \end{cases} \tag{6}$$

$$S_{DG}(i,j) = \max \begin{cases} S_{MM}(i-1,j) + \log[q_{i-1}(M,D)] \\ S_{DG}(i-1,j) + \log[q_{i-1}(D,D)] \end{cases} \tag{7}$$

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 (Fig. 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 Equation (4) by the average amino acid frequencies in the query or target protein, $\bar{q}(a)$ or $\bar{p}(a)$, 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 (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} [Equation (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 lth 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 lth pair state, i.e. $S_l = S_{\text{aa}}(q_{i(l)}, p_{j(l)})$ if the lth pair state is an MM state and zero otherwise. The autocorrelation function

$$g(d) = \sum_{l=1}^{L-d} S_l\, S_{l+d} \tag{8}$$

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

$$S_{\text{corr}} = w_{\text{corr}} \sum_{d=1}^{4} g(d) \tag{9}$$

to the total score, *after* the best alignment is found.[4] The weight $w_{\text{corr}} = 0.1$ was determined empirically on a small test set of 317 × 317 pairwise alignments.

Scoring secondary structure

HHsearch allows to score a predicted secondary structure either against a predicted secondary structure or against a 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 for each residue 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 σ, ρ and c to occur together, as well as the probability $P(\sigma)$ for σ to occur and the probability $P(\rho, c)$ for the pair (ρ, c) to occur. In this way we derived ten 3×7 substitution matrices, one for each value of c:

$$M_{\text{SS}}(\sigma; \rho, c) = \log \frac{P(\sigma; \rho, c)}{P(\sigma) P(\rho, c)}. \tag{10}$$

Now suppose column i of HMM q has predicted secondary structure ρ_i^q and confidence value c_i^q and column j of HMM p has known secondary structure[5] σ_j^p. The secondary structure score for q_i and p_j is obtained by multiplying the log-odds in M_{SS} with a weight w_{SS},

$$S_{\text{SS}}(q_i, p_j) = w_{\text{SS}} M_{\text{SS}}(\sigma_j^p; \rho_i^q, c_i^q). \tag{11}$$

This score is added to the amino acid column score $S_{\text{aa}}(q_i, p_j)$ 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 stretches of identical states of predicted secondary structure is ∼7 we expect an optimum weight $w_{\text{SS}} \approx 1/7$. Empirically we indeed find a broad optimum around $w_{\text{SS}} = 0.15$, 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 7×7 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.[6]

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 states twice, once for q and once for p. Again we can omit the mapping of the DSSP seven-state secondary structure from one profile HMM to a homologous HMM. The substitution matrix for the alignment of states (ρ_i^q, c_i^q) and (ρ_j^p, c_j^p) is

$$\begin{aligned} M_{\text{SS}}(\rho_i^q, c_i^q; \rho_j^p, c_j^p,) &= \log \frac{P(\rho_i^q, c_i^q; \rho_j^p, c_j^p)}{P(\rho_i^q, c_i^q)\, P(\rho_j^p, c_j^p)} \\ &= \log \sum_\sigma \frac{P(\rho_i^q, c_i^q | \sigma)}{P(\rho_i^q, c_i^q)} \frac{P(\rho_j^p, c_j^p | \sigma)}{P(\rho_j^p, c_j^p)} P(\sigma), \end{aligned} \tag{12}$$

where the sum runs over all seven DSSP states. This matrix tells us how much more probable it is to obtain predictions (ρ_i^q, c_i^q) and (ρ_j^p, c_j^p) for a pair of aligned homologous residues than to obtain them independently of each other, whatever the actual secondary structure state σ may be. The secondary structure score calculated from this matrix by $S_{\text{SS}}(q_i, p_j) = w_{\text{SS}} M_{\text{SS}}(\rho_i^q, c_i^q; \rho_j^p, c_j^p)$ is added to the column score with the same weight $w_{\text{SS}} = 0.15$ 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.[7]

In order to pinpoint the source of improvments, 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 Equations (5)–(7). HHsearch 1 is the basic HMM–HMM version, HHsearch 2 includes the correlation score [Equation (9)]; in addition to this HHsearch 3 compares predicted with predicted secondary structure [Equation (12)] and HHsearch 4 uses predicted versus known secondary structure [Equation (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

[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.

[6] One explanation is that the PSIPRED prediction is calculated from alignments that may be quite diverse. Therefore the matrix M_{SS} already contains some contribution of evolution in time.

[7] We obtained significantly better results by changing the default setting '`-g 0.5`' to '`-g 1.0`' and building the profiles from those columns of the multiple alignment that have a residue in the seed sequence instead of using the 50% gaps rule.

only homologous sequences enter the alignments. All methods in the benchmark (except BLAST) were tested with this same set of alignments.

Detection of homologs

Each domain in SCOP is classified into a hierarchy of family, superfamily, 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 cannot 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') $FP/(TP+FP) = 10\%$ (dashed line) it finds only 908 homologous pairs, or 2.2% out of a total of 41 505. 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 [Equation (9)] improves this value to 46.7% (HHsearch 2). When in addition, the predicted secondary structure is used for both HMMs, a value of 48.8% is achieved (HHsearch 3). And finally, HHsearch 4 uses actual secondary structure from DSSP in one of the two HMMs and finds 50.0% of the 41 505 homologs. This is a factor 23 more than BLAST, 2.8 and 2.7 times more than PSI-BLAST and HMMER, 2.0 times more than PROF_SIM and 1.47 times more than COMPASS.

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 owing 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

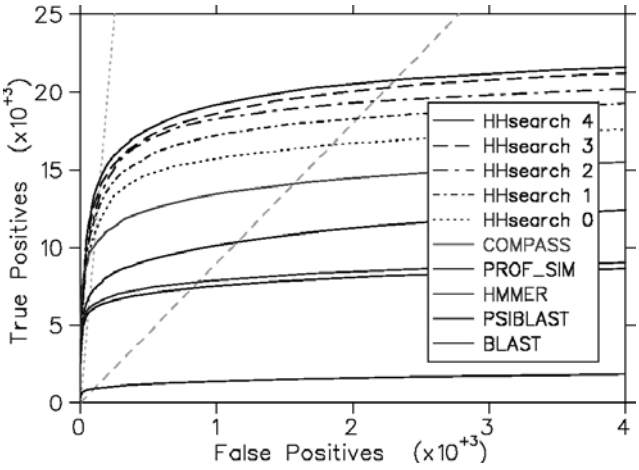

Fig. 2. Sensitivity of various homology detection tools, measured by how many true positives are detected at varying numbers of false positives in an all-against-all benchmark on SCOP-20 . True positives are pairs from the same superfamily, false positives are pairs from different classes. Dashed straight line: error rate 10%. There are 41 505 true positives and 1.08×10^7 false positives in total. For definitions of HHsearch 0–4, please refer to the main text.

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.[8]

Figure 3 plots true versus false positives for the new definition. The overall picture is similar to the previous figure with a few noteworthy differences. First, all tools except BLAST find more true positives at a fixed error rate. Second, the more sensitive tools improve more than the less sensitive ones, even in relative terms. This indicates that the new definition of true and false positives comes closer to defining homology than the previous, more rigid definition by SCOP superfamilies. The improvement is particularly conspicuous for HHsearch 3 and 4 that use secondary structure. The reason is that the 'new' true positives which come from different superfamilies are on average harder to detect than the 'old' true positives used in the previous figure. Since the less sensitive tools are not likely to detect them as homologs it is mainly the most sensitive tools which profit from their reclassification as true positives. Third, a notable exception to the

[8]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 ≥ 0.125 the MaxSub score is set to 0.

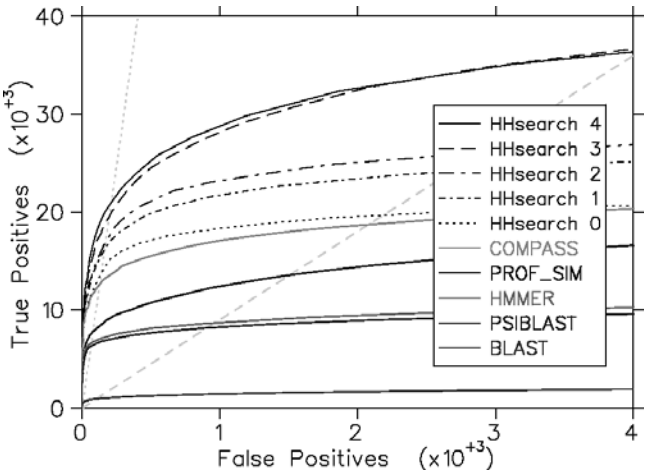

Fig. 3. Same as previous figure, but for a broader definition of true positives: True positives are pairs from the same superfamily *or* with MaxSub score of at least 0.1; false positives are pairs from different classes *and* zero MaxSub score.

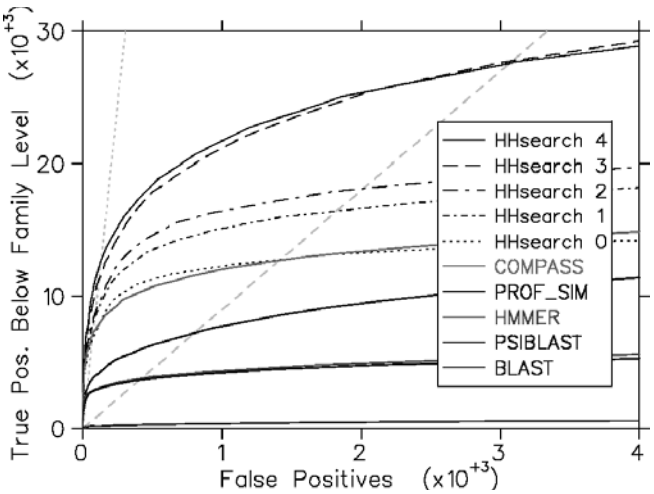

Fig. 4. Same as in previous figure, but all pairs at family level are ignored. This leaves as closest homologs only the pairs related at the superfamily level.

above remark is the improvement of PROF_SIM and COMPASS in relation to the basic profile–profile version of HHsearch (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 (Fig. 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 Figures 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 [Equation (12)] embodies twice as much uncertainty as the scoring matrix for known versus predicted secondary structure [Equation (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, $S_{\mathrm{Dev}} = N_{\mathrm{correct}}/\min(L_q, L_p)$, where N_{correct} is the number of residue pairs that are 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 $S_{\mathrm{Mod}} = N_{\mathrm{correct}}/L_{\mathrm{ali}}$, 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:

$$S_{\mathrm{balanced}} = (S_{\mathrm{Dev}} + S_{\mathrm{Mod}})/2\,. \qquad (13)$$

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).

Figure 5a–c plots the binned distribution of MaxSub scores for all pairs related at the family level (10 223 pairs), superfamily level

[9]The 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.

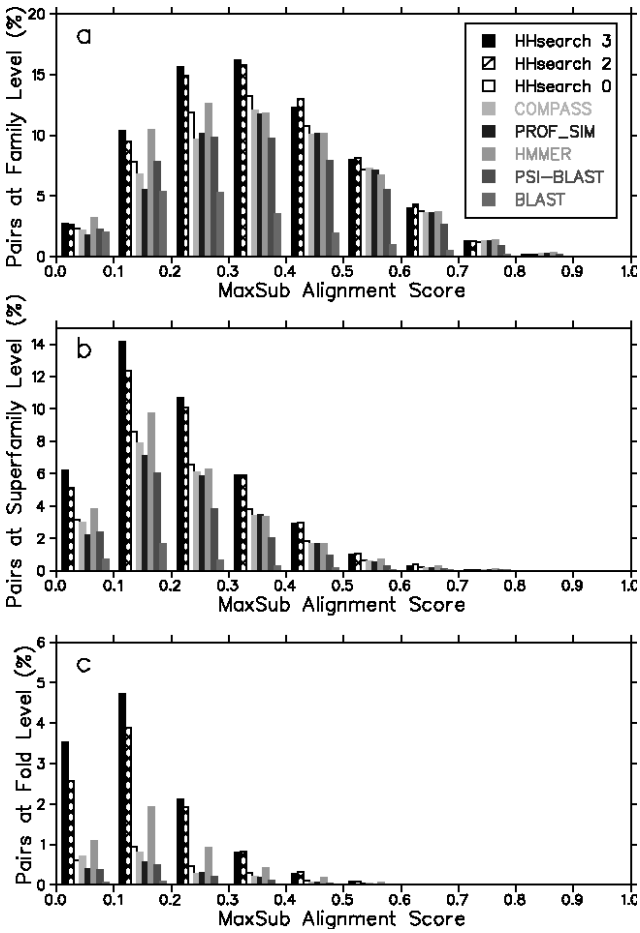

Fig. 5. Distribution of MaxSub scores for alignments of domain pairs related at the family, superfamily and fold level in percent of the total number of homologous sequence pairs at that level of relationship. Counts with MaxSub score of exactly zero are not shown.

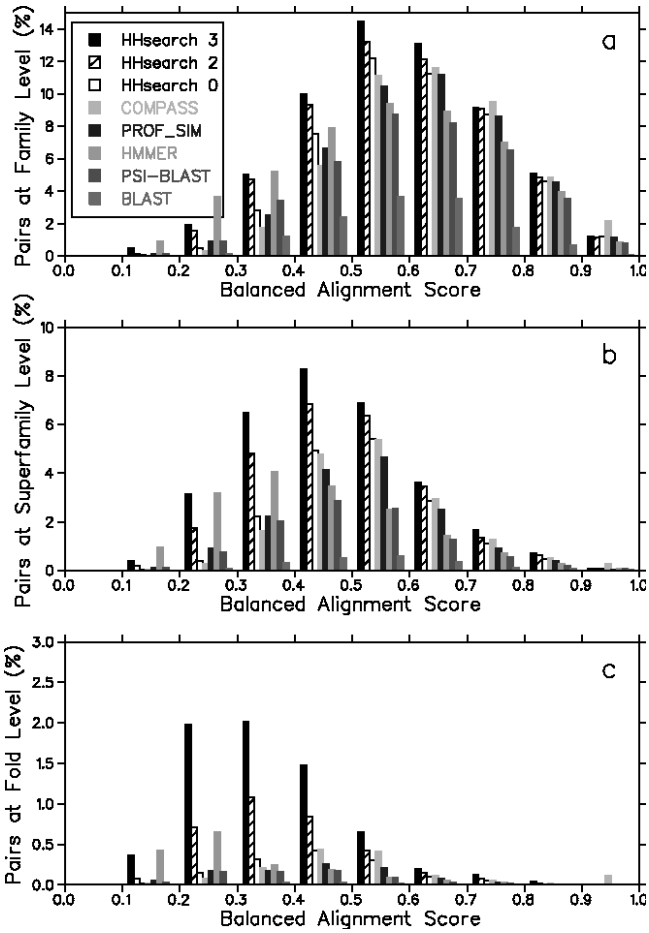

Fig. 6. Distribution of balanced scores [Equation (13)] for alignments of domain pairs related at the family, superfamily and fold level. Counts with zero score are not shown.

(31 282 pairs) and fold level (66 813 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 2–4. 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 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 factors are 3.3 (COMPASS), 6.0 (PROF_SIM), 7.3 (HMMER), 9.4 (PSI-BLAST) and 63 (BLAST).[10]

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; Marti-Renom *et al*., 2004; Edgar and Sjölander, 2004). In these studies differences in performance between the tested

[10]Note that 4.5% of all alignments at the fold level is quite a lot. Domain pairs related at the fold level are deemed non-homologous by SCOP and we might not expect any reasonably good alignments at all. This relatively high number suggests that many sequences classified into different superfamilies by SCOP are in fact homologous.

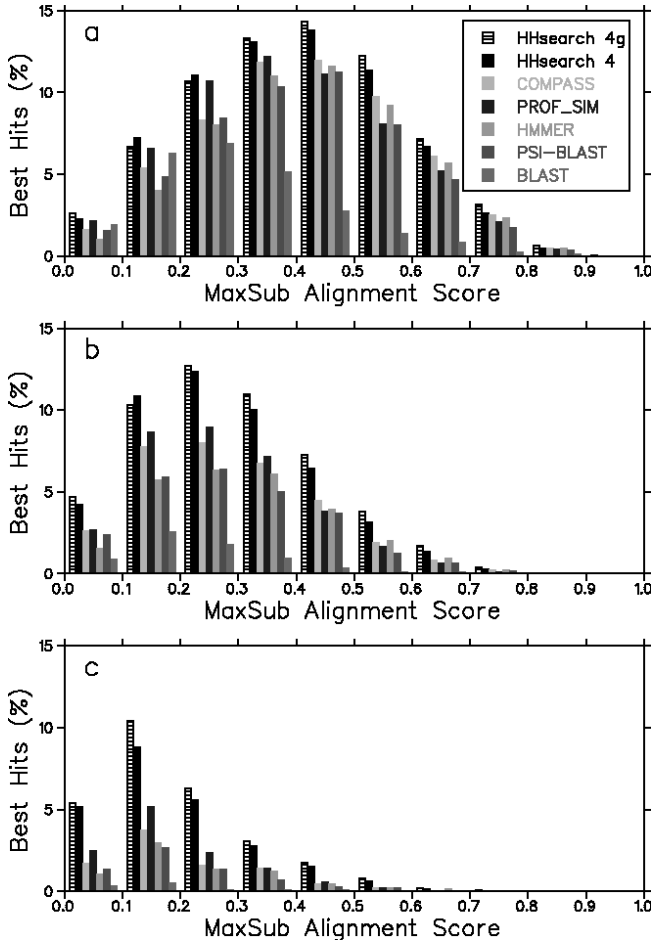

Fig. 7. Distribution of MaxSub scores for the best match in each of 3691 database scans that are related at maximum (**a**) at the family level, (**b**) at the superfamily, or (**c**) at the fold level. For HHsearch 4g all HHsearch 4 alignments are realigned with the semi-global algorithm.

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 templates 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.

As expected, the performance in Figure 7 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. 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 2–4. 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 $\geq$0.1 in 66% of all cases, and 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 generalization of the well-known log-odds score. A novel correlation score was derived which increases the 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 dataset 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 seperate 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

Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Altschul,S.F., Madden,T.L., Schäffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acid Res.*, **25**, 3389–3402.

Barrett,C., Hughey,R. and Karplus,K. (1997) Scoring hidden markov models. *Comput. Appl. Biosci.*, **13**, 191–199.

Bork,P. and Koonin,E.V. (1998) Predicting functions from protein sequences – where are the bottlenecks. *Nat. Genet.*, **18**, 313–318.

Chandonia,J.M., Hon,G., Walker,N.S., Lo Conte,L., Koehl,P., Levitt,M. and Brenner,S. (2004) The ASTRAL compendium in 2004. *Nucleic Acids Res.*, **32**, D189–D192.

Cline,M., Hughey,R. and Karplus,K. (2002) Predicting reliable regions in protein sequence alignments. *Bioinformatics*, **18**, 306–314.

Doolittle,R.F. (1981) Similar amino acid sequences: chance or common ancestry. *Science*, **214**, 149–159.

Durbin,R., Eddy,S., Krogh,A. and Mitchison,G. (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge.

Eddy,S.R. (1998) Profile hidden markov models. *Bioinformatics*, **14**, 755–763.

Edgar,R.C. and Sjölander,K. (2003) SATCHMO: sequence alignment and tree construction using hidden markov models. *Bioinformatics*, **19**, 1404–1411.

Edgar,R.C. and Sjölander,K. (2004) A comparison of scoring functions for protein sequence profile alignment. *Bioinformatics*, **20**, 1301–1308.

Fischer,D., Rychlewski,L., Dunbrack,R.L.J., Ortiz,A.R. and Elofsson,A. (2003) Cafasp3: the third critical assessment of fully automated structure prediction methods. *Proteins*, **53**, 503–516.

Ginalski,K., Pas,J., Wyrwicz,L.S., von Grotthus,M., Bujnicki,J.M. and Rychlewski,L. (2003) ORFeus: detection of distant homology using sequence profiles and predicted secondary structure. *Nucleic Acid Res.*, **31**, 3804–3807.

Gonnet,G.H., Cohen,M.A. and Brenner,S.A. (1992) Exhaustive matching of the entire protein sequence database. *Science*, **256**, 1443–1445.

Hargbo,J. and Elofsson,A. (1999) Hidden markov models that use predicted secondary structures for fold recognition. *Proteins*, **36**, 68–76.

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

Henn-Sax,H.B., Wilmanns,M. and Sterner,R. (2001) Divergent evolution of $(\beta\alpha)_8$–barrel enzymes. *Biol. Chem.*, **382**, 1315–1320.

Jones,D.T. (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.*, **292**, 195–202.

Kabsch,W. and Sander,C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577–2637.

Karplus,K., Karchin,R., Barrett,C., Tu,S., Cline,M., Diekhans,M., Grate,L., Casper,J. and Hughey,R. (2001) What is the value added by human intervention in protein structure prediction. *Proteins*, **45** Suppl. 5, 86–91.

Kawabata,T. and Nishikawa,K. (2000) Protein structure comparison using the markov transition model of evolution. *Proteins*, **41**, 108–122.

Kelley,L.A., MacCallum,R.M. and Sternberg,M.J. (2000) Enhanced genome annotation using structural profiles in the program 3D-PSSM. *J. Mol. Biol.*, **299**, 499–520.

Kinch,L. and Grishin,N. (2002) Evolution of protein structures and functions. *Curr. Opin. Struct. Biol.*, **12**, 400–408.

Kinch,L.N., Wrabl,J.O., Krishna,S.S., Majumdar,I., Sadreyev,R.I., Qi,Y., Pei,C.H.J. and Grishin,N.V. (2003) CASP5 assessment of fold recognition target predictions. *Proteins*, **53**, 395–409.

Koh,I., Eyrich,V.A., Marti-Renom,M.A., Przybylski,D., Madhusudhan,M.S., Eswar,N., Grana,O., Pazos,F., Valencia,A., Sali,A. and Rost,B. (2003) EVA: evaluation of protein structure prediction servers. *Nucleic Acids Res.*, **31**, 3311–3315.

Krogh,A., Brown,M., Mian,I.S., Sjölander,K. and Haussler,D. (1994) Hidden markov models in computational biology. Applications to protein modeling. *J. Mol. Biol.*, **235**, 1501–1531.

Kunin,V., Chan,B., Sitbon,E., Lithwick,G. and Pietrokovski,S. (2001) Consistency analysis of similarity between multiple alignments: prediction of protein function and fold structure from analysis of local sequence motifs. *J. Mol. Biol.*, **307**, 939–949.

Lyngsø,R.B., Pedersen,C.N.S. and Nielsen,H. (1999) Metrics and similarity measures for hidden markov models. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, pp. 178–186.

Marti-Renom,M.A., Madhusudhan,M.S. and Sali,A. (2004) Alignment of protein sequences by their profiles. *Protein Sci.*, **13**, 1071–1087.

Mittelman,D., Sadreyev,R. and Grishin,N.V. (2003) Probabilistic scoring measures for profile–profile comparison yields more accurate short seed alignments. *Bioinformatics*, **19**, 1531–1539.

Murzin,A.G., Brenner,S.E., Hubbard,T. and Chothia,C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.

O'Sullivan,O., Zehnder,M., Higgins,D., Bucher,P., Grosdidier,A. and Notredame,C. (2003) APDB: a novel measure for benchmarking sequence alignment methods without reference alignments. *Bioinformatics*, **19**, i215–i221.

Panchenko,A.R. (2003) Finding weak similarities between proteins by sequence profile comparison. *Nucleic Acids Res.*, **31**, 683–689.

Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.

Pei,J. and Grishin,N.V. (2001) AL2CO: calculation of positional conservation in a protein sequence alignment. *Bioinformatics*, **17**, 700–712.

Pei,J., Sadreyev,R. and Grishin,N.V. (2003) PCMA: fast and accurate multiple sequence alignment based on profile consistency. *Bioinformatics*, **19**, 427–428.

Pietrokovski,S. (1996) Searching databases of conserved sequence regions by aligning protein multiple-alignments. *Nucleic Acids Res.*, **24**, 3836–3845.

Rychlewski,L., Fischer,D. and Elofsson,A. (2003) LiveBench–6: large-scale automated evaluation of protein structure prediction servers. *Proteins*, **53**, 542–547.

Rychlewski,L., Jaroszewski,L., Li,W. and Godzik,A. (2000) Comparison of sequence-profiles. strategies for structural predictions using sequence information. *Protein Sci.*, **9**, 232–241.

Sadreyev,R.I., Baker,D. and Grishin,N.V. (2003) Profile–profile comparisons by COMPASS predict intricate homologies between protein families. *Protein Sci.*, **12**, 2262–2272.

Sadreyev,R.I. and Grishin,N.V. (2003) COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance. *J. Mol. Biol.*, **326**, 317–336.

Sauder,J.M., Arthur,J.W. and Dunbrack,R.L.J. (2000) Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins*, **40**, 6–22.

Siew,N., Elofsson,A., Rychlewski,L. and Fischer,D. (2000) MaxSub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics*, **16**, 776–785.

Tang,C.L., Xie,L., Koh,I.Y., Posy,S., Alexov,E. and Honig,B. (2003) On the role of structural information in remote homology detection and sequence alignment: new methods using hybrid sequence profiles. *J. Mol. Biol.*, **334**, 1043–1062.

Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.

Tomii,K. and Akiyama,Y. (2004) FORTE: a profile–profile comparison tool for protein fold recognition. *Bioinformatics*, **20**, 594–595.

Venclovas,C. (2003) Comparative modeling in CASP5: progress is evident, but alignment errors remain a significant hindrance. *Proteins*, **53**, 380–388.

von Öhsen,N., Sommer,I. and Zimmer,R. (2003) Profile–profile alignment: a powerful tool for protein structure prediction. *Pac. Symp. Biocomput.*, pp. 252–263.

Wang,G. and Dunbrack,R.L.J. (2004) Scoring profile–profile sequence alignments. *Protein Sci.*, **13**, 1612–1626.

Yona,G. and Levitt,M. (2002) Within the twilight zone: a sensitive profile–profile comparison tool based on information theory. *J. Mol. Biol.*, **315**, 1257–1275.