Abstract

Motivation: High-quality protein sequence alignments are essential for a number of downstream applications such as template-based protein structure prediction. In addition to the similarity score between sequence profile columns, many current profile–profile alignment tools use extra terms that compare 1D-structural properties such as secondary structure and solvent accessibility, which are predicted from short profile windows around each sequence position. Such scores add non-redundant information by evaluating the conservation of local patterns of hydrophobicity and other amino acid properties and thus exploiting correlations between profile columns.

Results: Here, instead of predicting and comparing known 1D properties, we follow an agnostic approach. We learn in an unsupervised fashion a set of maximally conserved patterns represented by 13-residue sequence profiles, without the need to know the cause of the conservation of these patterns. We use a maximum likelihood approach to train a set of 32 such profiles that can best represent patterns conserved within pairs of remotely homologs, structurally aligned training profiles. We include the new context score into our HMM-HMM alignment tool hhsearch and improve especially the quality of difficult alignments significantly.

Conclusion: The context similarity score improves the quality of homology models and other methods that depend on accurate pairwise alignments.

Contact: soeding@mpibpc.mpg.de

Supplementary information: Supplementary data are available at Bioinformatics online.
2 Materials and Methods

2.1 General approach and notation

We built a large training set of N aligned profile window pairs of D = 2d + 1 = 13 columns, by cutting out windows from the structural alignments of the full-length protein domains from the SCOP database (see Section 3.1). We seek to identify the maximally conserved patterns to properties is ‘many to few’, for example, many quite different patterns are all characteristic of alpha helix states, more information might be extracted by learning and comparing conserved patterns directly, independent of what actual structural properties they are associated with.

Two studies learned local sequence context patterns to improve alignments. Ohlson et al. (2006) train a self organizing map (SOM) to cluster local profile windows. They then trained a neural network to compute an optimum similarity score for aligned pairs of SOM states. Improvements were small, however. Ma et al. (2012) and (2013) reported substantial improvements in alignment quality using a non-linear extension of conditional random fields which include as features the local sequence profile neighborhood.

Here, we devised a method to explicitly learn strongly conserved local patterns. For training, we cut out pairs of sequence profile windows from structurally aligned, homologous proteins and learn the set of the 32 best-conserved patterns using the expectation maximization (EM) algorithm. With these patterns, which we call ‘context states’, we define a score that helps to discriminate homologous from non-homologous positions by analyzing the conservation of patterns between the aligned positions. We show that the new context similarity score improves the quality of global and local alignments of our pairwise alignment tools hhsearch and hha1ign (Söding, 2005) and that this in turn results in better 3D homology models.

2.2 Generative model

We want to find parameters Θ that maximize the likelihood function

\[ L(\Theta) = P((X_1, Y_1), \ldots, (X_N, Y_N)|\Theta) = \prod_{n=1}^{N} P(X_n, Y_n|\Theta) \]

All training samples are supposed to be independent of each other so that the likelihood can be decomposed into a product. We use a mixture model for \( P(X_n, Y_n|\Theta) \) as shown in Figure 1. The hidden variable \( z_n \in \{1, \ldots, K\} \) indicates the index of the context state that gave rise to \( (X_n, Y_n) \):

\[ \prod_{n=1}^{N} P(X_n, Y_n|\Theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} P(X_n, Y_n, z_n = k|\theta_k) \]

Because our model assumes conditional independence of \( X_n \) and \( Y_n \) given the hidden context state \( z_n \), it follows that

\[ \prod_{n=1}^{N} P(X_n|\Theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} P(X_n|\theta_k)P(z_n = k|\theta_k) \]

The context state prior probabilities \( p(z_n|\theta_k) \) are simply the mixture weights \( \theta_k \). We model \( P(X_n|\theta_k) \), the probability to observe counts \( X_n(i,a) \) of amino acid \( a = 1, \ldots, 20 \) in column \( i = -d, \ldots, d \), using a multinomial distribution for each column \( i \):

\[ P(X_n|\theta_k) = \prod_{j=d}^{d} \left( \sum_{a=1}^{20} \frac{\Gamma(N_{a}^{(j)}(i)+1)}{\Gamma(N_{a}^{(j)}(i)+1)+1} \prod_{a=1}^{20} p_{k}(j,a)^{X_{n}(i,a)} \right)^{w_{i}} \]

and analogously for \( P(Y_n|p_{j}) \). Because the effective counts \( X_n(i,a) \) can assume values outside the natural numbers, we replaced factorials \( ! \) with Gamma functions \( \Gamma(x+1) \). We assign a weight \( w_{i} \) to each column in Equation (4). The weights are parameterized as \( w_{i} = w_{\text{const}} \beta^{i} \), so that central columns contribute more than flanking columns when \( \beta < 1 \). \( p_{j} \) and \( x \) are discrete probability distributions which need to satisfy

\[ \sum_{k=1}^{K} \theta_k = 1, \sum_{a=1}^{20} p_{k}(j,a) = 1, \text{for } k = 1, \ldots, K \text{ and } j = -d, \ldots, d. \]
2.3 EM algorithm

We use the EM algorithm (Dempster et al., 1977) to maximize the likelihood in eq (1) of generating the pairs of aligned training profiles. Lagrange multipliers allowed us to analytically perform the optimization in the M-step under the constraints in Equation (5) (see Supplementary Material).

2.4 Scoring functions

2.4.1 Context states score

We define the context score for position i in profile X and position j in profile Y as a log-odds score,

\[ S_{\text{ctx}}(X_i, Y_j) = \log \left( \frac{P(X_i, Y_j | \Theta)}{P(X_i | \Theta) P(Y_j | \Theta)} \right) \]  

that is, the logarithm of the ratio of probability for \( X_i \) and \( Y_j \) to have been generated together from the same context state (see Equation 3), divided by the probability for \( X_i \) and \( Y_j \) to have been generated independently of each other. By applying Bayes’ Theorem twice,

\[ P(X_i | z_k, \Theta) = \frac{P(z_k | X_i, \Theta) P(X_i | \Theta)}{P(z_k | \Theta)} \]

this expression can be transformed into the following form,

\[ S_{\text{ctx}}(X_i, Y_j) = \log \sum_{z_k} \frac{P(z_k | X_i, \Theta) P(z_k | Y_j, \Theta)}{P(z_k | \Theta)} \]

Note the analogy to the log-sum-of-odds scoring function for profile-profile alignment that was derived in (Soding, 2005),

\[ S_a(p_X(i), p_Y(j)) = \log \frac{\sum_d p_X(i, d) p_Y(j, d)}{f(a)} \]

Here, the amino acid \( a \) is analogous to our context state \( k \). The numerators describe the probability to co-emit the same amino acid \( a \) or the same context state \( z_k \), respectively, \( f(a) \), the background frequency of amino acid \( a \), is analogous to \( z_k \). Multiplying by \( 1/|a| \) (or \( 1/|z_k| \)) corrects for the fact that frequent amino acids (context states) match up more frequently by chance than rare ones.

Finally, the total score is a linear combination of profile column score, context states score, and HHBLAST’s standard three-state secondary structure score (Soding, 2003):

\[ S_{\text{total}}(i,j) = (1 - w_a) S_a(p_X(i), p_Y(j)) + w_{\text{ctx}} S_{\text{ctx}}(X_i, Y_j) + w_a S_a(i,j) \]

As the context states score comprises of \( D = 13 \) columns, we correct the weight \( w_{\text{ctx}} \) for the redundancy caused by the overlap of \( D - 1 \) positions between two consecutive windows (Supplementary Material).

2.4.2 atx alphabet score

Apart from secondary structure, we tested a fine-grained structural alphabet atx, which was developed to improve the alignment quality and which performed well in several CASP competitions (Karchin et al., 2003). It is an enhanced version of the DSSP alphabet (Kabsch and Sander, 1983), which subdivides the E state (β-strand) into six states. We applied the improved four-layer neural networks of (Katzman et al., 2008) and determined for all query and template residues the probabilities for each of the 13 letters in the atx alphabet.

We denote \( p_{\text{atx}}^q(i,s) \) as the probability for letter \( s \in \{1, \ldots, 13\} \) at position i of profile X and similarly \( p_{\text{atx}}^t(j,s) \) for Y. The atx structural score \( S_{\text{atx}} \) is defined as a log-sum-of-odds score in analogy to Equations (8) and (9):

\[ S_{\text{atx}}(i,j) = \sum_{s=1}^{13} \frac{p_{\text{atx}}^q(i,s) p_{\text{atx}}^t(j,s)}{p_{\text{bg}}^{\text{atx}}(s)} \]

where \( p_{\text{bg}}^{\text{atx}} \) is the background probability for atx state \( s \) in a large set of proteins. This atx score was added to the total score with its own optimized weight (Section 2.6).

2.5 Data sets

First, we filtered the SCOP (V1.75, Lo Conte et al., 2000) to obtain a set with a maximum pairwise sequence identity of 20% and enriched each SCOP20 sequence by generating a multiple sequence alignment with our iterative hmm-hmm searching tool hhblits (Remmert et al., 2012) (two iterations against uniprot20 with standard parameters). Then each msa was converted into an hhm via hhmake (Remmert et al., 2015), ftp://toolkit.genzentrum.lmu.de/pub/HH/hhblurapperguide.pdf) with standard parameters. Finally, the dataset was divided into two sets by assigning the members of every fifth fold into a smaller set \( S_{\text{train}} \) (1492 domains) and the rest into a set \( S_{\text{test}} \) (5426 domains). Query and templates for the training and optimization set were then sampled from \( S_{\text{train}} \), whereas test alignments were sampled from \( S_{\text{test}} \). This procedure is important to ensure that none of the sequences in the test sets are homologous to any of the sequences in the training and optimization sets.

2.6 Parameter optimization

Because the time to compute the context score is proportional to \( K \), we need to keep \( K \) low in order not to significantly slow down HHSEARCH. The improvements between \( K = 128 \) and \( K = 32 \) were moderate, so we chose \( K = 32 \). As \( D = 13 \) for the window width was found to perform well in various related applications (e.g. Biegert and Söding, 2009) we chose the same value without further optimization.

We needed to optimize the parameters \( w_{\text{center}} \) and \( \beta \) describing the weights \( w_X \) in Section 2.2 and the weight \( w_{\text{ctx}} \) of the context score in Equation (10). We could get better results by using separate parameters for training the context states library \( \left( w_{\text{center}}^{\text{ctx}}, \beta^{\text{ctx}} \right) \) for which no weight \( w_{\text{ctx}} \) is needed and for the alignment stage \( \left( w_{\text{center}}^{\text{atx}}, \beta^{\text{atx}}, w_{\text{ctx}} \right) \). Because systematic testing of \( w_{\text{center}}^{\text{atx}} \) and \( \beta^{\text{atx}} \) requires to generate a context library for each setting and furthermore the performance then depends on the other parameters, these were also adapted from (Biegert and Söding, 2009) \( \left( w_{\text{center}}^{\text{atx}} = 1.3 \right. \) and \( \beta^{\text{atx}} = 0.85 \), so that the left and rightmost columns in a context profile get a weight \( w_{\text{center}}^{\text{atx}} = w_{\text{ctx}} = 0.49 \). We checked libraries with lower \( w_{\text{center}}^{\text{atx}} = 0.2 \) and 0.5, but this led to flatter context states and a drop in performance.

To optimize the alignment algorithm parameters, we performed a grid search for \( w_{\text{center}}^{\text{ctx}} \in \{0.2, 0.25, 0.5, 1\} \) and \( w_{\text{ctx}} \in \{0.8, 0.9, 1.1, 1.2\} \) and measured the average of alignment sensitivity and precision on 1000 pairwise alignments where query and template were sampled from \( S_{\text{train}} \). We obtained best results for \( w_{\text{center}}^{\text{ctx}} = 0.2 \) and \( w_{\text{ctx}} = 1.0 \). Surprisingly, \( w_{\text{center}}^{\text{ctx}} \) turned out to be clearly smaller than 1 so that the context states become flatter during scoring.

We optimized the parameters \( \left( w_{\text{ctx}}, w_{\text{center}}^{\text{ctx}}, w_{\text{corr}} \right) \) specifically for the ROC5 homology detection benchmark by maximizing the area under the ROC5 curve, using the same context state library as in the alignment quality benchmarks. We used all sequences in \( S_{\text{train}} \). In
addition to $w_{\text{ctx}}$ and $w_{\text{center}}^l$, the parameter $\text{corr}$ from hhsearch was reoptimized. Differing from the setting for alignment quality, we arrived at $w_{\text{ctx}} = 0.6$, $w_{\text{center}}^l = 0.4$ and $\text{corr} = 0.2$. The optimization of the secondary structure score weight and the $\ast \tau$ alphabet weight were done on the same set and yielded $w_{\text{aa}} = 0.25$, $w_{\text{str}} = 0.12$.

3 Results

3.1 Training

We sampled up to 10 pairs of proteins per superfamliy in $S_{\text{train}}$ and accepted if their structural alignment score using tmalign (Zhang and Skolnick, 2005) was between 0.5 and 0.85. If a window of width $D = 13$ centered at a structurally aligned residue pair had at least nine pairs within a distance of 5 Å in the structural alignment, the window was selected as a training sample and the $D$ columns in the two corresponding count profiles were cut out. This procedure returned 141 508 training profile window pairs from 2987 pairwise alignments. Subsequently, these training pairs were filtered by calculating the mean column score $S_{\text{aa}}$ (Equation 9) over all $D = 13$ columns and we rejected the trivial cases with $S_{\text{aa}} > 1.5$ (46 839 samples). At the beginning of training, we initialized the context states library randomly and ran 25 EM iterations. Different initializations and more iteration led to quite similar log-likelihood values and libraries, indicating a robust training (see Supplementary Section 9 for a plot of the library).

3.2 Alignment quality

We first assess the effect of context similarity scoring for global alignments (Tables 1–4) since global alignments (or quasiglobal local alignments) are used as input to homology modeling, the most important application of our method. We then proceed to analyze the quality of local alignments (Fig. 2).

We created two sets of pairwise alignments, a ‘hard set’ and an ‘easier set’. For the hard set, we sampled 6000 query-template pairs from $S_{\text{test}}$ by randomly selecting pairs from the same SCOP superfamliy but from a different family, with a tmalign score between 0.5 and 0.9, up to a maximum number of 25 pairs. For the easier set, we sampled 3000 query-template pairs from $S_{\text{test}}$ by randomly selecting up to 25 pairs from the same SCOP family with a tmalign score between 0.6 and 0.95. These resulted in a mean tmalign score of 0.61 for the hard and 0.72 for the easier set and in a mean sequence identity of 14.3% for the hard and 16.4% for the easier set (see Supplementary Table S2 and Supplementary Data File).

The difficulty for an HMM-HMM alignment algorithm also depends strongly on the amount of evolutionary information available in the two profile HMMs. Even structurally very similar pairs can be difficult to align when their profile HMMs were only trained on thin MSAs with few homologous proteins. Vice versa, even very remote homologs can often be reliably aligned when their profile HMMs were trained on thick, diverse MSAs.

To test the influence of the context similarity score on the amount of evolutionary information available in the profile HMMs, we created variant test sets of HMMs trained on MSAs with low diversity. These reflect better the diversity of MSAs encountered in practice than the typically rich and diverse MSAs from sequences in the SCOP, which mostly belong to large, very well studied protein families. To this end, we reduced the number of effective sequences (Neff) of the MSAs to a maximum value of 3 by using hhmfilter (Remmert et al., 2013) with the -neff 3 option. Neff quantifies the diversity in an MSA (Supplementary Material). It lies between 1 for a single sequence and 20. In summary, we have created four different test sets: hardNeff_def, hardNeff_low, easiNeff_def and easiNeff_low.

We measured the alignment accuracy in terms of residue-based sensitivity and precision, where sensitivity =TP/(TP+FN) and precision =TP/(TP+FP). A true positive (TP) is a pair of residues that is aligned correctly, that is, occurs in the reference alignment by tmalign (Zhang and Skolnick, 2005). A false positive (FP) occurs in the test alignment but not in the reference alignment. A false negative (FN) occurs in the reference alignment but not in the test alignment. All alignments were generated in global alignment mode using hhalign with option -mact 0.

We evaluated six different score combinations on each of the four benchmark sets (Tables 1–3): (i) the baseline version (‘profile’) that uses only the column score $S_{\text{aa}}$ (Equation 9) and no secondary structure score, (ii) the secondary structure score based on PSIPRED predictions (Jones, 1999) for the first and 3D structure-based DSSP assignments for the second sequence of each pair (‘ss’), (iii) the secondary structure score based only on PSIPRED predictions for both sequences (‘SSpred’), (iv) the sum of the score in (3) and the score based on the predictions of the 13-state $\ast \tau$ alphabet (Equation 11) (‘ss+str’) that was optimized for its positive impact on alignment quality (Karchin et al., 2003), (v) the context similarity score (Equation 8) (‘ctx’ and (vi) the sum of the score in (3) and the context similarity score (‘ss+ctx’).

Tables 1 and 2 show the results of the alignment benchmark for the hard test set with default diversity and with low diversity MSAs, respectively. The score ‘SSpred’ that makes use of only predicted secondary structure performs almost as well as the score ‘ss’ that requires the actual secondary structure of one of the aligned proteins, for high and low diversity MSAs. When combined with the secondary structure score based on DSSP, both the ‘str’ alphabet-based score (‘ss+str’) and the context similarity score (‘ss+ctx’) lead to additional improvements, but these are clearly more pronounced for the ‘ss+ctx’ score, which achieves the highest sensitivity and precision on high and low diversity MSAs. All three secondary structure classes profited to a similar degree from the additional scoring terms.

Interestingly, the improvements owing to the secondary structure scoring and to the context score are much stronger for low-diversity MSAs than for high-diversity ones (improvement of ‘ss+ctx’ over ‘ss’ is 3.6/2.7% (sens/prec) for high-diversity MSAs and of 11.5/9.2% for low-diversity MSAs). Although the purely sequence-based score SSpred performs similarly to the context similarity score ‘ctx’ for high-diversity MSAs, the new context score is clearly superior for low-diversity MSAs.

On the easier dataset (Table 3), secondary structure was still beneficial but the relative improvements in sensitivity/precision declined from +3.6/+2.7% for more distantly related pairs to +1.3/+1.1% for the easier cases. In contrast to the hard cases, the ‘str’-based score and the context scoring led to only minor gains. However, as for the hard set of protein pairs, when MSA diversity was low, str and in particular the context score again yielded significant improvements (Table 3) over the secondary structure score alone (sens/prec gain: +1 and +0.9% for str and +4.8 and +3.9% for ctx, respectively).

So far we have assessed global alignments. To render the comparison of the quality of local alignments meaningful, we have to measure residue-wise precision and sensitivity for different settings of sensitivity versus precision tradeoff. The alignment tools in HH-suite allow the user to control this tradeoff with the -mact option. Figure 2 shows the resulting receiver operator characteristic (ROC) plot. Similarly to the global alignment case (-mact 0), context similarity scoring improves the alignment quality and is most beneficial when the number of effective sequences is low.
The paired t-test P-values refer to ‘ss+ctx’ versus ‘ss’.

The upper part summarizes which information is used by each versions (PSIPRED predictions, 13-state ss prediction (Katzman et al., 2008), the new context score, and the 8-state DSSP secondary structure assignments from the known 3D structure. The lower part gives the overall sensitivity and precision, below subdivided into helix (h) extended beta strand (e) and coil (c) residues, as assigned by DSSP. The differences between ‘ss+ctx’ and ‘ss’ are significant according to the paired t-test P-value (<2.2e−16).

In summary, our new context similarity score consistently improved the alignment quality when combined with the standard secondary structure scoring in hhsearch. In the cases in which no secondary structure is available, the context score (‘ctx’) also consistently improved the alignment quality in comparison with the other purely sequence-based score (‘SSpred’). The extent of improvements is larger the more difficult the alignment is, that is, the more diverged the two proteins are and the lower the diversity of the MSAs that their profile HMMs were trained on (see Supplementary Figure S2 for the dependence on alignment diversity). Because the calculation of the context states score increases the runtime by a factor of about 100, one profit mostly when realigning only a set of preselected templates for homology modeling (Section 3.4).

Table 2. Residue-based alignment sensitivity and precision as shown in Table 1) but on the hard set with low diversity MSAs (averaged Neff 2.85)

<table>
<thead>
<tr>
<th></th>
<th>profile</th>
<th>ss_pred</th>
<th>ss</th>
<th>ss+str</th>
<th>ctx</th>
<th>ss+ctx</th>
</tr>
</thead>
<tbody>
<tr>
<td>sens</td>
<td>0.305</td>
<td>0.350</td>
<td>0.364</td>
<td>0.372</td>
<td>0.393</td>
<td>0.406</td>
</tr>
<tr>
<td>prec</td>
<td>0.288</td>
<td>0.331</td>
<td>0.346</td>
<td>0.354</td>
<td>0.363</td>
<td>0.378</td>
</tr>
<tr>
<td>sens_h</td>
<td>0.318</td>
<td>0.367</td>
<td>0.377</td>
<td>0.384</td>
<td>0.412</td>
<td>0.420</td>
</tr>
<tr>
<td>prec_h</td>
<td>0.303</td>
<td>0.350</td>
<td>0.364</td>
<td>0.371</td>
<td>0.386</td>
<td>0.399</td>
</tr>
<tr>
<td>sens_e</td>
<td>0.324</td>
<td>0.371</td>
<td>0.394</td>
<td>0.402</td>
<td>0.412</td>
<td>0.434</td>
</tr>
<tr>
<td>prec_e</td>
<td>0.338</td>
<td>0.387</td>
<td>0.409</td>
<td>0.418</td>
<td>0.420</td>
<td>0.443</td>
</tr>
<tr>
<td>sens_c</td>
<td>0.267</td>
<td>0.303</td>
<td>0.315</td>
<td>0.322</td>
<td>0.340</td>
<td>0.350</td>
</tr>
<tr>
<td>prec_c</td>
<td>0.234</td>
<td>0.266</td>
<td>0.278</td>
<td>0.283</td>
<td>0.291</td>
<td>0.302</td>
</tr>
</tbody>
</table>

The differences between ‘ss+ctx’ and ‘ss’ are significant according to the paired t-test P-value (<2.2e−16).

3.3 Comparison with the profile alignment tool PPAS

Next, we wanted to compare hhsearch/hhalign with other profile–profile alignment tools. Although many profile–profile alignment methods have been developed by the protein structure prediction community, most modern methods were not designed to run independently of their protein structure prediction pipeline. Because our goal here is to compare alignment methods and not methods to generate sequence profiles, we could not benchmark these tools. However, PPAS, a profile–profile alignment tool developed in the lab of Yang Zhang, could be modified to run on user defined database profiles. It had been reported to yield equal or better results than hhsearch on a benchmark with hard and medium targets (Yan et al., 2013) and performed only slightly worse than their flagship aligner MUSTER (Wu and Zhang, 2008) that includes several 1D structure-based scores. We were unable to benchmark MUSTER as tools for pre-computing template profiles containing the 1D structure information are not available.

Because PPAS requires profiles in PSI-BLAST format, we converted our template MSAs into PSI-BLAST format by calling blastpgp with the -c option and a dummy database containing a single sequence. Yet for the query we had to keep the dependency on the PSI-BLAST output, because PPAS needs to parse it directly. For the default Neff benchmarks (easierNeff_def), we ran PPAS with three PSI-BLAST iterations, and used the default MSAs from $S_{out}$ for hhalign. For the low Neff benchmarks (easierNeff_low, hardNeff_low), we reduced the number of iterations to two, the minimum valid value for PPAS to run. This resulted in an average diversity of Neff ∼5.9 for the easy set and 5.67 for the hard set which is clearly above our filtered low Neff MSAs (Neff ∼2.84).

Table 3. Residue-based alignment sensitivity and precision based on 6000 pairwise alignments in the harder benchmark set for both default Neff and low Neff alignments

<table>
<thead>
<tr>
<th></th>
<th>profile</th>
<th>ss_pred</th>
<th>ss</th>
<th>ss+str</th>
<th>ctx</th>
<th>ss+ctx</th>
</tr>
</thead>
<tbody>
<tr>
<td>sens</td>
<td>0.639</td>
<td>0.653</td>
<td>0.658</td>
<td>0.661</td>
<td>0.638</td>
<td>0.667</td>
</tr>
<tr>
<td>prec</td>
<td>0.611</td>
<td>0.625</td>
<td>0.632</td>
<td>0.635</td>
<td>0.627</td>
<td>0.639</td>
</tr>
</tbody>
</table>

It is compared with hhalign with secondary structure and context score. For a comparison of hhalign with COMA (Margakevich and Venclovas, 2010), another profile–profile alignment tool (see Supplementary Table S1).
Thus, we converted these query PSI-BLAST alignments into a format readable by hhalign, ensuring that PPAS and hhalign received the same input.

We compared PPAS with two versions of hhalign: version (3) with secondary structure scoring based on PSIPRED and DSSP and version (6) that additionally included the context score. Both versions exceeded PPAS’s sensitivity and precision by 14 and 17% on the hard set and by 7 and 10% on the easier set, respectively. Surprisingly, hhsearch outperformed PPAS even without secondary structure information.

### 3.4 Application to homology modeling

A quality bottleneck in homology modeling is the generation of accurate alignments between the query and template sequences. We therefore tested the impact of our context similarity scoring on homology modeling by comparing the quality of 3D models generated from alignments of different methods. To build the 3D models we used MODELLER (Sali and Blundell, 1993), the most widely used tool for homology modeling. The results are shown in Table 5.

As expected, the better alignments led to better homology models: The context score in hhalign improved models on the hard set on high diversity MSAs by 10.2% (default Neff) and 9.9% (low Neff) over PPAS models and on the easier set by 6.4% (default Neff) and 5.9% (low Neff). Again, the more difficult the query-template alignments, the larger were the improvements due to the context similarity score.

### 3.5 Remote homology detection

When searching large databases like the PDB, out of the numerous matches detected, often only a few are of interest. For homology modeling, for instance, it suffices to identify 1–5 suitable homologous templates. Consequently, it is important to rank homologous proteins on top. We therefore analyze the sensitivity for remote homology detection using a ROC5 plot. For each query protein, one computes the ROC5 value, which is the area under the ROC curve up to the fifth FP. The ROC5 plot shows the fraction of queries for which the ROC5 value is above the threshold on the x-axis. A measure that summarizes the performance on the ROC5 benchmark is the area under the ROC5 curve (AUC).

We performed an all-against-all search with hhsearch in local alignment mode (the standard setting for template searches in our HHpred structure prediction server) on the proteins in S_{test}. We defined members belonging to the same superfamily as TPs and those of different folds as FPs. Pairs with both proteins within the four- to eight-bladed beta-propellers (SCOP fold IDs b.66–b.70) were treated as unknown, and the same for Rossmann-like folds (c.2 – c.5, c.30, c.66, c.78, c.79, c.111). The ROC5 analysis in Figure 3 shows that adding secondary structure (‘ss’) increases the AUC from 0.583 to 0.609 (4.4%). str and ctx scoring give moderate improvements to 0.625 and 0.641 (2.6 and 5.2% compared with ‘ss’), respectively.

### 4 Discussion

#### 4.1.1 Context-specific pseudocounts

Each residue in a protein is subject to very specific selection constraints that mostly are caused by the requirement of folding into a stable 3D structure. The constraints depend on the local structural context, which can be predicted to some extent directly from each residues’ sequence context. In Angermüller et al. (2012) and Biegert and Söding (2009), we exploited this concept to learn a set of 4000 patterns best describing a representative set of 106 training sequence profiles. Using these, we could enrich sequences and sequence profiles with context-specific pseudocounts. This approach is implemented in all hh-suite programs since version 2.0.15. Hence, the improvements observed here come on top of those already reported for context-specific pseudocounts.

The difference between the previous approach and the one taken here is the degree to which we demand conservation of patterns. In Angermüller et al. (2012) and Biegert and Söding (2009), conservation needed to be just good enough to leave a clear pattern in the training profiles built from relatively closely related sequences. Here, in contrast, we use pairs of remotely homologous proteins and structural alignments, which are more reliable than HMM-HMM alignments at low sequence similarities, to find patterns that are highly conserved across large evolutionary distances, roughly corresponding to the SCOP superfamily level.

#### 4.1.2 3D structural properties

In contrast to secondary structure similarity scores and similar scores based on the conservation of 1D structural properties, we take an unsupervised approach of learning the conserved patterns. Hence, we do not need to know what particular property led to the conservation of the patterns we learn. Therefore, while we have not succeeded in capturing all possible conserved patterns in our 32-state library (shown in Supplementary Figure S3), we have manifestly learned conserved patterns whose information cannot be
reduced to a 3-state or even 13-state alphabet of local backbone geometries. Because protein structure is known to be well conserved, we expect many recurring local structural features such as those described by structural alphabets to be overlapping to some degree with our context states.

4.1.3 Failed approach 1: discriminative learning

Instead of maximizing the likelihood in Equation (1) we tried hard to maximize the sum of similarity scores of positive training samples minus the sum of scores for negative training samples. Yet, this objective function is no longer likelihood, precluding use of the EM algorithm. Moreover, it proved to be prone to degenerate solutions and required careful enforcement of the restraint that the probability in the denominator in Equation (8) be equal to the average probability of that context state over all training states.

4.1.4 Failed approach 2: transitions between context states

We tried out a more general model that allows transitions between context states \( k \) and \( k' \). We learned the matrix of transition probabilities \( P(k'|k) \) by maximum likelihood. The score between local profiles \( X \) and \( Y \) was

\[
\log \sum_{k=1}^{K} P(z = k | X) \sum_{k'=1}^{K} P(z = k' | Y) P(k' | k).
\]

The alignment quality and sensitivity did not improve, however, probably because \( K = 32 \) states are not yet fine-grained enough to necessitate substitutions between these states.

Conclusion

The new context score helps most in the difficult cases: (i) when little evolutionary information is contained in the HMMs to be aligned, and (ii) when proteins are remotely related. In the first case, integrating the sparse evolutionary information vertically within an MSA leads to only little noise suppression (i.e. the distinction of correct from incorrect alignments). Therefore, we profit most from pulling together information horizontally along the MSAs. In the second case, it makes sense to focus on the features that are best conserved among remote homologs, which is what our context score was trained to do. The new score slows down hhsearch by a factor of 100. This precludes its use in hhblits, whereas it will be unproblematic for homology modeling and other applications, where a relatively small set of proteins needs to be aligned with the best possible quality.

Acknowledgement

We would like to thank Stefan Seemayer, Jessica Andreani, and Martin Steinegger for comments and discussions.

Funding

This work was funded by the Bavarian Center for Molecular Biosystems (BioSysNet grant) and the Deutsche Forschungsgemeinschaft (grants GRK1721, SFB464).

Conflict of Interest

none declared.

References


Yang, Y. et al. (2011) Improving protein fold recognition and template-based modeling by employing probablistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates. Bioinformatics, 27, 2076–2082.