Bioinformatics, 31(5), 2015, 674–681
doi: 10.1093/bioinformatics/btu697
Advance Access Publication Date: 22 October 2014
Original Paper

Sequence analysis

Context similarity scoring improves protein sequence alignments in the midnight zone

Armin Meier[1] and Johannes Söding[1,2,*]

[1]Gene Center, LMU Munich, 81377 Munich and [2]Max Planck Institute for Biophysical Chemistry, 37077 Göttingen, Germany

*To whom correspondence should be addressed.

Associate Editor: Burkhard Rost

Received on 14 May 2014; revised on 30 September 2014; accepted on 20 October 2014

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.

1 Introduction

Most methods for fold recognition and protein structure prediction are based on the pairwise alignment of query and template sequence profiles (Elofsson, 2002; Yan *et al.*, 2013). Top-performing structure prediction tools add to the profile column similarity score a secondary structure score, which improves the sensitivity for detecting remote homologs and the quality of the resulting alignments (Karplus *et al.*, 2003; Xu and Xu, 2000). In order to maximize the information gain and therefore the improvements in alignment quality, various finer-grained alphabets of backbone structure states have been developed—together with tools to predict these states (Karchin *et al.*, 2003, 2004; Katzman *et al.*, 2008).

In addition to secondary structure, other 1D structural properties are employed to improve sequence alignments in the so-called twilight and midnight zone, such as solvent accessibility (Liu *et al.*, 2007), residue coordination numbers (Karchin *et al.*, 2004; Peng and Xu, 2009; Wu and Zhang, 2008), backbone dihedral torsion angles (Wu and Zhang, 2008), 1D environmental fitness scores (Peng and Xu, 2009; Teichert *et al.*, 2010), or a combination of these (Faraggi *et al.*, 2011; Ma *et al.*, 2012; Yang *et al.*, 2011). In all cases, the discretized 1D structural property is predicted from a local sequence profile window of 13 to 15 positions, and the similarity between predicted and actual 1D properties of the aligned query and template positions is scored.

One can get independent of structural information by comparing 1D predictions with 1D predictions. Surprisingly, this works almost as well (Przybylski and Rost, 2004; Söding, 2005). We believe the reason is that the conservation of a 1D structural property is tied to the conservation of characteristic local patterns of amino acid properties, and the conservation of these patterns is scored indirectly by comparing the predicted 1D property. Because the relationship of 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 `hhalign` (Söding, 2005) and that this in turn results in better 3D homology models.

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 ('context states') irrespective of any predefined structural or functional properties. The conserved patterns are represented by sequence profiles of length D.

We call the N aligned training profiles $\boldsymbol{X}_n = X_n(i-d,\cdot),\ldots, X_n(i+d,\cdot)$ and $\boldsymbol{Y}_n = Y_n(j-d,\cdot),\ldots, Y_n(j+d,\cdot)$, $n \in \{1,\ldots,N\}$. Here, $X_n(i,a)$ is the number of effective counts of amino acid a at position i in one training profile and position i is aligned to position j in the aligned profile. The effective counts are defined in the following way: let $p_{X_n}(i,a)$ be standard sequence profile built for sequence X_n, that is, the probability of amino acid a occurring at position i in the MSA for X_n. The effective counts are defined as $X_n(i,a) = p_{X_n}(i,a)$ $N^{\text{eff}}_{X_n}(i)$ and analogously $Y_n(j,a) := p_{Y_n}(j,a)\, N^{\text{eff}}_{Y_n}(j)$. Here, $N^{\text{eff}}_{X_n}(i)$ (abbreviated 'Neff') is the number of effective sequences at position i (Supplementary Material).

Each of the K conserved patterns (= context states) is parameterized by a sequence profile $\boldsymbol{p}_k$. $\boldsymbol{p}_k$ is a $D \times 20$ matrix with $p_k(j,a)$ being the occurrence probability of amino acid $a \in \{1,\ldots,20\}$ at profile column $j \in \{-d,\ldots,d\}$. Each context state has a mixture weight α_k. We abbreviate the model parameters by $\theta_k = (\boldsymbol{p}_k, \alpha_k)$ and $\Theta = (\theta_1,\ldots,\theta_K)$.

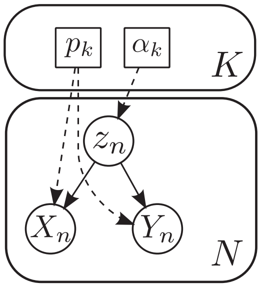

Fig. 1. Generative graphical model: each of the N training profile pairs (X_n, Y_n) is generated by a mixture distribution with K components, the 'context states'. The hidden variables z_n encode the context state that gave rise to the n'th training sample (X_n, Y_n). Each of the D columns in these count profiles is modeled by a multinomial distribution over the 20 amino acids with parameters $\boldsymbol{p}_k$. The context states have mixture weights $\alpha_k (k = 1,\ldots, K)$

2.2 Generative model

We want to find parameters Θ that maximize the likelihood function

$$L(\Theta) = P((\boldsymbol{X}_1, \boldsymbol{Y}_1),\ldots,(\boldsymbol{X}_N, \boldsymbol{Y}_N)|\Theta) = \prod_{n=1}^{N} P(\boldsymbol{X}_n, \boldsymbol{Y}_n|\Theta) \quad (1)$$

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 $P(\boldsymbol{X}_n, \boldsymbol{Y}_n|\Theta)$ 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 $(\boldsymbol{X}_n, \boldsymbol{Y}_n)$:

$$\prod_{n=1}^{N} P(\boldsymbol{X}_n, \boldsymbol{Y}_n|\Theta) = \prod_{n=1}^{N}\sum_{n=1}^{N} P(\boldsymbol{X}_n, \boldsymbol{Y}_n, z_n = k|\theta_k) \quad (2)$$

Because our model assumes conditional independence of $\boldsymbol{X}_n$ and $\boldsymbol{Y}_n$ given the hidden context state z_n, it follows that

$$\prod_{n=1}^{N} P(\boldsymbol{X}_n, \boldsymbol{Y}_n|\Theta) = \prod_{n=1}^{N}\sum_{n=1}^{N} P(\boldsymbol{X}_n|\boldsymbol{p}_k)P(\boldsymbol{Y}_n|\boldsymbol{p}_k)P(z_n = k|\alpha_k) \quad (3)$$

The context state prior probabilities $p(z_n|\alpha_k)$ are simply the mixture weights α_k. We model $P(\boldsymbol{X}_n|\boldsymbol{p}_k)$, the probability to observe counts $X_n(j,a)$ of amino acid $a = 1,\ldots,20$ in column $j = -d,\ldots,d$, using a multinomial distribution for each column j,

$$P(\boldsymbol{X}_n|\boldsymbol{p}_k) = \prod_{j=-d}^{d}\left(\frac{\Gamma(N^{\text{eff}}_{X_n}(j)+1)}{\prod_{a=1}^{20}\Gamma(X_n(j,a)+1)}\prod_{a=1}^{20} p_k(j,a)^{X_n(j,a)}\right)^{w_j}, \quad (4)$$

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

$$\sum_{k=1}^{K}\alpha_k = 1, \sum_{a=1}^{20} p_k(j,a) = 1, \text{for; } k = 1,\ldots,K \text{ and } j = -d,\ldots,d. \quad (5)$$

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,

$$S_{\mathrm{ctx}}(\mathbf{X}_i, \mathbf{Y}_j) = \log\left(\frac{P(\mathbf{X}_i, \mathbf{Y}_j|\Theta)}{P(\mathbf{X}_i|\Theta)\,P(\mathbf{Y}_j|\Theta)}\right) \tag{6}$$

that is, the logarithm of the ratio of probability for $\mathbf{X}_i$ and $\mathbf{Y}_j$ to have been generated together from the same context state (see Equation 3), divided by the probability for $\mathbf{X}_i$ and $\mathbf{Y}_j$ to have been generated independently of each other. By applying Bayes' Theorem twice,

$$P(\mathbf{X}_i|z_n, \Theta) = \frac{P(z_n|\mathbf{X}_i, \Theta)P(\mathbf{X}_i|\Theta)}{P(z_n|\Theta)}\,, \tag{7}$$

this expression can be transformed into the following form,

$$S_{\mathrm{ctx}}(\mathbf{X}_i, \mathbf{Y}_j) = \log\sum_{z_n}\frac{P(z_n|\mathbf{X}_i, \Theta)P(z_n|\mathbf{Y}_j, \Theta)}{P(z_n|\Theta)}. \tag{8}$$

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

$$S_{\mathrm{aa}}(p_X(i), p_Y(j)) = \log\sum_{a=1}^{20}\frac{p_X(i,a)\,p_Y(j,a)}{f(a)}. \tag{9}$$

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 $z_n = k$, respectively. $f(a)$, the background frequency of amino acid a, is analogous to α_k. Multiplying by $1/f(a)$ (or $1/\alpha_k$) 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 `hhsearch`'s standard three-state secondary structure score (Söding, 2005):

$$\begin{aligned} S_{\mathrm{total}}(i,j) = {} & (1 - w_{\mathrm{ctx}})\,S_{\mathrm{aa}}(p_X(i), p_Y(j)) + w_{\mathrm{ctx}}\,S_{\mathrm{ctx}}(\mathbf{X}_i, \mathbf{Y}_j) \\ & + w_{\mathrm{ss}}\,S_{\mathrm{ss}}(i,j) \end{aligned} \tag{10}$$

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

2.4.2 `str` alphabet score

Apart from secondary structure, we tested a fine-grained structural alphabet `str`, 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 `str` alphabet.

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

$$S_{\mathrm{str}}(i,j) = \sum_{s=1}^{13}\frac{p_X^{\mathrm{str}}(i,s)\,p_Y^{\mathrm{str}}(j,s)}{p_{\mathrm{bg}}^{\mathrm{str}}(s)}, \tag{11}$$

where $p_{\mathrm{bg}}^{\mathrm{str}}$ is the background probability for `str` state s in a large set of proteins. This `str` 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 HMM via `hhmake` (Remmert *et al.*, 2013), ftp://toolkit.genzentrum.lmu.de/pub/HH-suite/hhsuite-userguide.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_{train} (1492 domains) and the rest into a set S_{test} (5426 domains). Query and templates for the training and optimization set were then sampled from $S_{\mathrm{train},}$ whereas test alignments are sampled from S_{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_{center} and β describing the weights w_j in Section 2.2 and the weight w_{ctx} of the context score in Equation (10). We could get better results by using separate parameter sets for training the context states library ($w_{\mathrm{center}}^{\mathrm{tr}}$, β^{tr}) (for which no weight w_{ctx} is needed) and for the alignment stage ($w_{\mathrm{center}}^{\mathrm{al}}, \beta^{\mathrm{al}}, w_{\mathrm{ctx}}$). Because systematic testing of $w_{\mathrm{center}}^{\mathrm{tr}}$ and β^{tr} 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) ($w_{\mathrm{center}}^{\mathrm{tr}} = 1.3$ and $\beta^{\mathrm{tr}} = 0.85$), so that the left and rightmost columns in a context profile get a weight $w_{j=-6} = w_{j=6} = 0.49$. We checked libraries with lower $w_{\mathrm{center}}^{\mathrm{tr}} = 0.2$ 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 $w_{\mathrm{center}}^{\mathrm{al}} \in \{0.2, 0.25, 0.5, 1\}$ and $w_{\mathrm{ctx}} \in \{0.8, 0.9, 1, 1.1, 1.2\}$ and measured the average of alignment sensitivity and precision on 1000 pairwise alignments where query and template were sampled from $S_{\mathrm{train}.}$ We obtained best results for $w_{\mathrm{center}}^{\mathrm{al}} = 0.2$ and $w_{\mathrm{ctx}} = 1$. Surprisingly, $w_{\mathrm{center}}^{\mathrm{al}}$ turned out to be clearly smaller than 1 so that the context states become flatter during scoring.

We optimized the parameters ($w_{\mathrm{ctx}}, w_{\mathrm{center}}^{\mathrm{al}}$, corr) 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 $S_{\mathrm{train}.}$ In addition to w_{ctx} and $w^{\text{al}}_{\text{center}}$, the parameter corr from `hhsearch` was reoptimized. Differing from the setting for alignment quality, we arrived at $w_{\text{ctx}} = 0.6$, $w^{\text{al}}_{\text{center}} = 0.4$ and corr = 0.2. The optimization of the secondary structure score weight and the `str` alphabet weight were done on the same set and yielded $w_{\text{ss}} = 0.25$, $w_{\text{str}} = 0.12$.

3 Results

3.1 Training

We sampled up to 10 pairs of proteins per superfamily in S_{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_{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_{test} by randomly selecting pairs from the same SCOP superfamily 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_{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 `hhfilter` (Remmert *et al.*, 2013) with the `-neff 3` option. Neff quantifies the diversity in an MSA (Supplemental Material). It lies between 1 for a single sequence and 20. In summary, we have created four different test sets: $\text{hard}_{\text{Neff\_def}}$, $\text{hard}_{\text{Neff\_low}}$, $\text{easier}_{\text{Neff\_def}}$ and $\text{easier}_{\text{Neff\_low}}$.

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_{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 ('SS_{pred}'), (iv) the sum of the score in (3) and the score based on the predictions of the 13-state `str` 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 'SS_{pred}' 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' of 3.6/2.7% (sensitivity/precision) for high-diversity MSAs and of 11.5/9.2% for low-diversity MSAs). Although the purely sequence-based score SS_{pred} 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 0.0), context similarity scoring improves the alignment quality and is most beneficial when the number of effective sequences is low.

Table 1. Residue-bases alignment sensitivity and precision of six versions of `hhalign` on 6000 pairwise alignments in the hard set with default diversity MSAs (average Neff 6.55). Best version is in bold

hard set−default Neff: hard$_{\text{Neff\_def}}$						
	profile	ss$_{\text{pred}}$	ss	ss + str	ctx	ss + ctx
PSIPRED		●	●	●		●
Str				●		
Ctx					●	●
DSSP			●	●		●
Sens	0.435	0.464	0.471	0.478	0.464	**0.488**
Prec	0.400	0.430	0.439	0.445	0.423	**0.451**
sens_h	0.456	0.481	0.487	0.491	0.489	**0.503**
prec_h	0.424	0.452	0.461	0.467	0.449	**0.473**
sens_e	0.467	0.501	0.515	0.524	0.494	**0.532**
prec_e	0.471	0.505	0.521	0.530	0.495	**0.535**
sens_c	0.372	0.395	0.401	0.406	0.394	**0.414**
prec_c	0.321	0.342	0.348	0.352	0.337	**0.357**

The upper part summarizes which information is used by each versions (PSIPRED predictions, 13-state `str` 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 ($<2.2e{-}16$).

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

hard set−low Neff: hard$_{\text{Neff\_low}}$						
	profile	ss$_{\text{pred}}$	ss	ss+str	ctx	ss+ctx
sens	0.305	0.350	0.364	0.372	0.393	**0.406**
prec	0.288	0.331	0.346	0.354	0.363	**0.378**
sens_h	0.318	0.367	0.377	0.384	0.412	**0.420**
prec_h	0.303	0.350	0.364	0.371	0.386	**0.399**
sens_e	0.324	0.371	0.393	0.402	0.412	**0.434**
prec_e	0.338	0.387	0.409	0.418	0.420	**0.443**
sens_c	0.267	0.303	0.315	0.322	0.340	**0.350**
prec_c	0.234	0.266	0.278	0.283	0.291	**0.302**

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 ('SS$_{\text{pred}}$'). 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).

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 (easier$_{\text{Neff\_def}}$, hard$_{\text{Neff\_def}}$), we ran PPAS with three PSI-BLAST iterations, and used the default MSAs from S_{test} for `hhalign`. For the low Neff benchmarks (easier$_{\text{Neff\_low}}$, hard$_{\text{Neff\_low}}$), 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 3000 pairwise alignments in the easier benchmark set for both default Neff and low Neff alignments

easier set							
default Neff: easier$_{\text{Neff\_def}}$							
	profile	ss$_{\text{pred}}$	ss	ss+str	ctx	ss+ctx	*P*-value
sens	0.639	0.653	0.658	0.661	0.658	**0.667**	4.4e-9
prec	0.611	0.625	0.632	0.635	0.627	**0.639**	1.9e-5
low Neff: easier$_{\text{Neff\_def}}$							
sens	0.532	0.564	0.573	0.578	0.572	**0.601**	2.2e-16
prec	0.518	0.549	0.559	0.564	0.548	**0.581**	2.2e-16

The paired *t*-test *P*-values refer to 'ss+ctx' versus 'ss'.

Table 4. Residue-based sensitivity and precision of 6000 pairwise alignments in the hard and easier benchmark set. PPAS makes use of predicted and DSSP secondary structure

hard set						
	default Neff			low Neff		
	PPAS	`hhalign`		PPAS	`hhalign`	
		ss	ss+ctx		ss	ss+ctx
sens	0.415	0.471	**0.488**	0.392	0.425	**0.454**
prec	0.388	0.439	**0.451**	0.370	0.401	**0.423**
easier set						
sens	0.607	0.658	**0.667**	0.599	0.630	**0.647**
prec	0.585	0.632	**0.639**	0.581	0.610	**0.623**

It is compared with `hhalign` with secondary structure and context score. For a comparison of `hhalign` with COMA (Margelevicius and Venclovas, 2010), another profile–profile alignment tool (see Supplemetary Table S1).

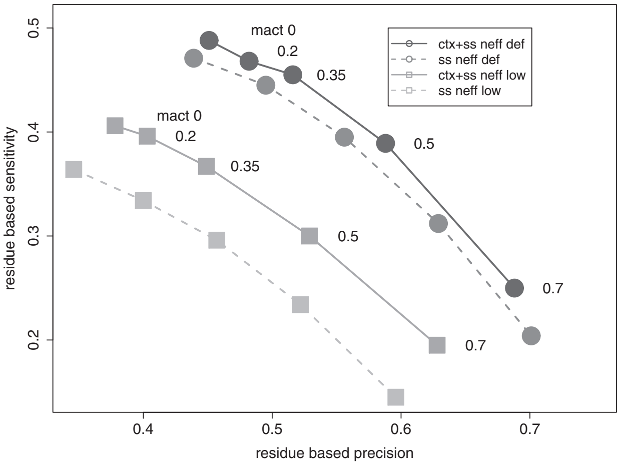

Fig. 2. Sensitivity and precision for local alignments on the hard set. The mact parameter controls the tradeoff of alignment sensitivity and precision ('greediness'). Global alignment corresponds to -mact 0.0

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 includes 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 β-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.

Table 5. Mean TMSCOREs of 3D homology models built from query-template alignments by three profile–profile alignment methods

set	Neff	PPAS	hhalign		*P*-value
			ss	ss+ctx	
hard	default	0.458	0.495	**0.505** (+2.0%)	2.2e-16
	medium	0.444	0.468	**0.488** (+4.2%)	2.2e-16
easier	default	0.610	0.644	**0.649** (+0.8%)	8.3e-7
	medium	0.604	0.628	**0.640** (+1.9%)	2.2e-16

The methods are tested on the same hard and easy set of query-template protein pairs as in the previous section. A high and a medium diversity set of MSAs was built from the query and template sequences using three and two iterations of PSI-BLAST, respectively. The paired t-test P-values demonstrate a statistically significant improvement of 'ss+ctx' over 'ss' quality scores.

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 10^6 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 D structural properties

In contrast to secondary structure similarity scores and similar scores based on the conservation of 1*D* 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

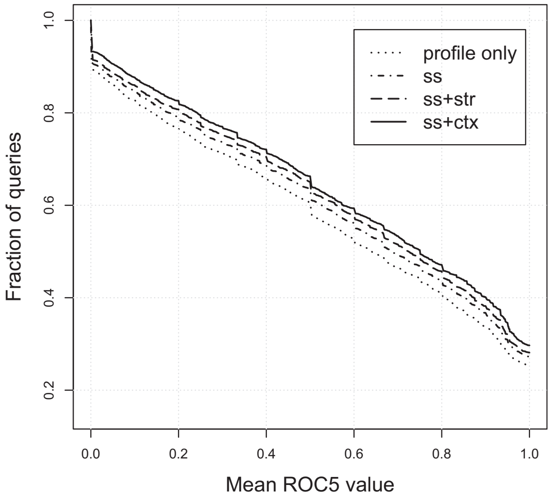

Fig. 3. ROC5 plot: fraction of query HMMs whose ROC5 value is above the value on the *x*-axis. The ROC5 value for a query is the average sensitivity in a query-specific ROC plot up to the fifth FP match

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 $\mathbf{X}$ and $\mathbf{Y}$ was $\log \sum_{k=1}^{K} \frac{P(z=k|\mathbf{X})}{P(z=k)} \sum_{k'=1}^{K} P(z=k'|\mathbf{Y})P(k'|k)$. 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, SFB646).

Conflict of Interest: none declared.

References

Angermüller,C. *et al.* (2012) Discriminative modelling of context-specific amino acid substitution probabilities. *Bioinformatics,* **28**, 3240–3247.

Biegert,A. and Söding,J. (2009) Sequence context-specific profiles for homology searching. *Proc. Natl Acad. Sci. USA,* **106**, 3770–3775.

Dempster,A. *et al.* (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. B.*, **39**, 1–38.

Elofsson,A. (2002) A study on protein sequence alignment quality. *Proteins,* **339**, 330–339.

Faraggi,E. *et al.* (2011) SPINE X: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *J. Comput. Chem.,* **33**, 259–267.

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.

Karchin,R. *et al.* (2003) Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. *Proteins,* **51**, 504–514.

Karchin,R. *et al.* (2004) Evaluation of local structure alphabets based on residue burial. *Proteins,* **55**, 508–518.

Karplus,K. *et al.* (2003) Combining local-structure, fold-recognition, and new fold methods for protein structure prediction. *Proteins,* **53**, 491–496.

Katzman,S. *et al.* (2008) PREDICT-2ND: a tool for generalized protein local structure prediction. *Bioinformatics,* **24**, 2453–2459.

Liu,S. *et al.* (2007) Fold recognition by concurrent use of solvent accessibility and residue depth. *Proteins,* **68**, 636–645.

Lo Conte,L. *et al.* (2000) SCOP: a structural classification of proteins database. *Nucleic Acids Res.,* **28**, 257–259.

Ma,J. *et al.* (2012) A conditional neural fields model for protein threading. *Bioinformatics,* **28**, i59–i66.

Ma,J. *et al.* (2013) Protein threading using context-specific alignment potential. *Bioinformatics,* **29**, i257–i265.

Margelevicius,M. and Venclovas,C. (2010) Detection of distant evolutionary relationships between protein families using theory of sequence profile-profile comparison. *BMC Bioinformatics,* **11**, 89.

Ohlson,T. *et al.* (2006) Improved alignment quality by combining evolutionary information, predicted secondary structure and self-organizing maps. *BMC Bioinformatics,* **7**, 357–366.

Peng,J. and Xu,J. (2009) Boosting protein threading accuracy. *Res. Comput. Mol. Biol.,* **5541**, 31–45.

Przybylski,D. and Rost,B. (2004) Improving fold recognition without folds. *J. Mol. Biol.,* **341**, 255–269.

Remmert,M. *et al.* (2012) HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat. Methods,* **9**, 173–175.

Remmert,M. *et al.* (2013) HH-suite for sensitive sequence searching based on HMM-HMM alignment, user-guide, HH-suite package.

Sali,A. and Blundell,T. (1993) Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.*, **234**, 779–815.

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

Teichert,F. *et al.* (2010) High quality protein sequence alignment by combining structural profile prediction and profile alignment using SABER-TOOTH. *BMC Bioinformatics,* **11**, 251.

Wu,S. and Zhang,Y. (2008) MUSTER: improving protein sequence profile-profile alignments by using multiple sources of structure information. *Proteins,* **72**, 547–556.

Xu,Y. and Xu,D. (2000) Protein threading using prospect: design and evaluation. *Proteins,* **40**, 343–354.

Yan,R. *et al.* (2013) A comparative assessment and analysis of 20 representative sequence alignment methods for protein structure prediction. *Sci. Rep.,* **3**, 2619.

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

Zhang,Y. and Skolnick,J. (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.,* **33**, 2302–2309.