BIOINFORMATICS ORIGINAL PAPER

Vol. 28 no. 24 2012, pages 3240–3247
doi:10.1093/bioinformatics/bts622

Sequence analysis

Advance Access publication October 17, 2012

Discriminative modelling of context-specific amino acid substitution probabilities

Christof Angermüller[1], Andreas Biegert[2] and Johannes Söding[1,*]

[1]Gene Center Munich and Department of Biochemistry, Ludwig-Maximilians-Universtät München, 81377 Munich, Germany and [2]Genedata, Lena-Christ-Strasse 50, 82152 Martinsried, Germany

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Protein sequence searching and alignment are fundamental tools of modern biology. Alignments are assessed using their similarity scores, essentially the sum of substitution matrix scores over all pairs of aligned amino acids. We previously proposed a generative probabilistic method that yields scores that take the sequence context around each aligned residue into account. This method showed drastically improved sensitivity and alignment quality compared with standard substitution matrix-based alignment.

Results: Here, we develop an alternative discriminative approach to predict sequence context-specific substitution scores. We applied our approach to compute context-specific sequence profiles for Basic Local Alignment Search Tool (BLAST) and compared the new tool (CS-BLASTdis) to BLAST and the previous context-specific version (CS-BLASTgen). On a dataset filtered to 20% maximum sequence identity, CS-BLASTdisis was 51% more sensitive than BLAST and 17% more sensitive than CS-BLASTgenin, detecting remote homologues at 10% false discovery rate. At 30% maximum sequence identity, its alignments contain 21 and 12% more correct residue pairs than those of BLAST and CS-BLASTgen, respectively. Clear improvements are also seen when the approach is combined with PSI-BLAST and HHblits. We believe the context-specific approach should replace substitution matrices wherever sensitivity and alignment quality are critical.

Availability: Source code (GNU General Public License, version 3) and benchmark data are available at ftp://toolkit.genzentrum.lmu.de/pub/csblast/.

Contact: soeding@genzentrum.lmu.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on August 3, 2012; revised on September 28, 2012; accepted on October 14, 2012

1 INTRODUCTION

Inferring the functions and the structures of proteins from those of homologous proteins has proven to be an extremely powerful approach in biology. To predict a homologous relationship between two proteins, their sequences are aligned such as to maximize the sum of scores over all aligned pairs of amino acid residues minus penalties for gaps. A sufficiently high score indicates a homologous relationship. The standard method for calculating scores for pairs of amino acids is the substitution matrix method (Dayhoff and Schwartz, 1972; Henikoff and Henikoff, 1992). The substitution score for amino acids a and b can be written as $S(a,b) = \text{const} \times \log(P(a|b)/P(a))$, where $P(a|b)$ is the probability of amino acid b mutating into a, and $P(a)$ is the background probability of a. The probabilities $P(a)$ and $P(a|b)$ are derived by counting the numbers of amino acids a and of aligned pairs (a, b) in a large set of trusted sequence alignments.

As protein sequences of folded domains are constrained by the necessity to maintain a stable structure, the substitution probabilities for a given residue are largely determined by the structural context within which it resides. Substitution matrices have, therefore, been trained for particular structural contexts, for example, depending on the residue's secondary structure, solvent accessibility or polarity (Overington *et al.*, 1992; Rice and Eisenberg, 1997; Shi *et al.*, 2001; Goonesekere and Lee, 2008). Methods that infer substitution probabilities of amino acids solely from their local sequence context have the advantage that they do not require the structure of the query protein to be known (Jones *et al.*, 1994; Baussand *et al.*, 2007; Huang and Bystroff, 2006). In Biegert and Söding (2009), we formulated a general approach to predict substitution probabilities from sequence context; for each residue in the query sequence, we compared the 13-residue window centred around it with a precomputed library of 13-column sequence profiles that represent all known sequence contexts. The substitution probabilities are computed as the weighted mixture of the central columns in these context profiles, with weights proportional to the similarity between the context profile and the 13-residue sequence context.

The approach was based on a generative model for learning context-specific substitution probabilities. The goal of this work is to further improve the prediction accuracy by developing a discriminative machine learning method for the prediction of substitution probabilities. As in our previous work, we apply the method to enhance Basic Local Alignment Search Tool (BLAST) by storing the predicted substitution probabilities for the query sequence in a sequence profile and jump-starting PSI-BLAST with it. We also apply the new method to generate context-specific pseudocounts for PSI-BLAST (Altschul *et al.*, 1997) and HHblits (Remmert *et al.*, 2011), our highly sensitive iterative sequence search software based on the pairwise comparison of hidden Markov models (HMMs). In all cases, we observe significant improvements over the generative version.

*To whom correspondence should be addressed.

2 METHODS

2.1 Generative versus discriminative models

A generative model explicitly describes the joint distribution $P(\mathbf{x},\mathbf{y}) = P(\mathbf{x}|\mathbf{y})P(\mathbf{y})$ over the observed variable $\mathbf{x}$ and the target variable $\mathbf{y}$. A generative model allows one to generate new data points $(\mathbf{x},\mathbf{y})$. Usually, it models the probabilities $P(\mathbf{x}|\mathbf{y})$ and $P(\mathbf{y})$ separately (Sutton and McCallum, 2006). To predict the unobserved target variable $\mathbf{y}$ given the observed data $\mathbf{x}$, the generative model uses Bayes' theorem, $P(\mathbf{y}|\mathbf{x}) = P(\mathbf{x}|\mathbf{y})P(\mathbf{y})/[\sum_{\mathbf{y}} P(\mathbf{x}|\mathbf{y})P(\mathbf{y})]$. A discriminative model directly models the probability $P(\mathbf{y}|\mathbf{x})$ of the target variable conditioned on the observed variable (Rubinstein and Hastie, 1997; Sutton and McCallum, 2006). It does not model the distribution of the input variable $\mathbf{x}$, which is not needed to predict $\mathbf{y}$ given $\mathbf{x}$. Generative models are commonly trained by maximizing the joint probability $\prod_n P(\mathbf{x}_n,\mathbf{y}_n)$ over the training data $(\mathbf{x}_n,\mathbf{y}_n)$, whereas discriminative models are usually trained by maximizing the conditional probability $\prod_n P(\mathbf{y}_n|\mathbf{x}_n)$. Therefore, if the goal is to predict $\mathbf{y}$ given $\mathbf{x}$, discriminative models seem more appropriate (Ng and Jordan, 2001; Caruana and Mizil, 2006).

2.2 Discriminative model for context-specific substitution probabilities

Given a query sequence $x_1,\ldots,x_L$, we want to predict context-specific substitution probabilities $P(a|\mathbf{C}_i)$. $P(a|\mathbf{C}_i)$ is the probability to observe amino acid a given sequence context $\mathbf{C}_i$. The context $\mathbf{C}_i$ describes the sequence of $l=2d+1$ amino acids around position i of the input sequence. More precisely, $\mathbf{C}_i$ is a binary profile, $\mathbf{C}_i(j,a) = I(x_{i+j} = a)$ for $j \in \{-d,\ldots,d\}$, whose entries are 1 if $x_{i+j} = a$ and zero otherwise.

Like the generative approach in Biegert and Söding (2009) (summarized in the Supplementary Material), the discriminative approach for modelling the substitution probabilities $P(a|\mathbf{C}_i)$ is again based on *K context states*, indexed by $k \in \{1,\ldots,K\}$. Each context state k is characterized by the following real-valued parameters: emission weights $\nu_k(a)$, bias weights π_k and context weights $\lambda_k(j,a)$. The emission probabilities $P(a|k)$ from context state k are given by the emission weights as follows:

$$P(a|k) = \frac{\exp(\nu_k(a))}{\sum_{a'=1}^{20}\exp(\nu_k(a'))} \tag{1}$$

In the generative model, the probability for context state k given context $\mathbf{C}_i$ was obtained with Bayes' theorem as $P(k|\mathbf{C}_i) = P(\mathbf{C}_i|k)P(k)/[\sum_{k'} P(\mathbf{C}_i|k)P(k)]$, where $P(\mathbf{C}_i|k)P(k)$ was modelled with a multinomial distribution, and the previous cluster probabilities $P(k)$ were model parameters. In the discriminative approach, we model $P(k|\mathbf{C}_i)$ directly by the exponential of an affine function of the context count profile $\mathbf{C}_i(j,a)$,

$$P(k|\mathbf{C}_i) = \frac{1}{Z(\mathbf{C}_i)}\exp\left(\pi_k + \sum_{j=-d}^{d}\sum_{a=1}^{20}\lambda_k(j,a)\mathbf{C}_i(j,a)\right) \tag{2}$$

with a normalization constant $Z(\mathbf{C}_i)$ that normalizes $P(k|\mathbf{C}_i)$ to 1.

The bias weights π_k quantify how much cluster k is preferred over the other clusters and roughly correspond to the $P(k)$ of the generative model. The context information is encoded by the context weights; $\lambda_k(j,a)$ is positive if amino acid a is preferred in column j and negative otherwise. The $\lambda_k(j,a)$ corresponds to the probabilities $p_k(j,a)$ of the context profiles of the generative model [see Equations (5–7) in Supplementary Material for details].

As in the generative model, we assume that the emitted amino acid a only depends on the context through the context states k:

$$P(a|\mathbf{C}_i) = \sum_{k=1}^{K} P(a,k|\mathbf{C}_i) = \sum_{k=1}^{K} P(a|k)P(k|\mathbf{C}_i) \tag{3}$$

In other words, the target distribution $P(a|\mathbf{C}_i)$ is obtained by mixing the emission probabilities $P(a|k)$ of each context state k weighted by the similarity $P(k|\mathbf{C}_i)$ of $\mathbf{C}_i$ to k. In essence, our discriminative model is a logistic regression maximum entropy classifier (Ng and Jordan, 2001; Rubinstein and Hastie, 1997) for discriminating between context states k given $\mathbf{C}_i$. Figure 1 illustrates the computation of the context-specific substitution probabilities $P(a|\mathbf{C}_i)$.

To train the model parameters, abbreviated as (π,λ,ν), we constructed a training set consisting of $N = 6\times10^6$ training pairs $(\mathbf{C}_1,\tilde{\mathbf{c}}_1),\ldots,(\mathbf{C}_N,\tilde{\mathbf{c}}_N)$, where $(\mathbf{C}_n,\tilde{\mathbf{c}}_n)$ was sampled from a multiple sequence alignment (MSA) with query sequence x and sequence profile q at position $i(n)$. $\mathbf{C}_n$ describes the sequence context of x at position $i(n)$, $\mathbf{C}_n(j,a) = I(x_{i(n)+j} = a)$. The vector $\tilde{\mathbf{c}}_n$ stores how often each amino acid a occurs in alignment column $i(n)$, $\tilde{\mathbf{c}}_n(a) = N_q(i(n))\,q(i(n),a)$. Here, $N_q(i)$ is the effective number of sequences in column i of the MSA from which q was built. It measures the diversity of the profile and is the

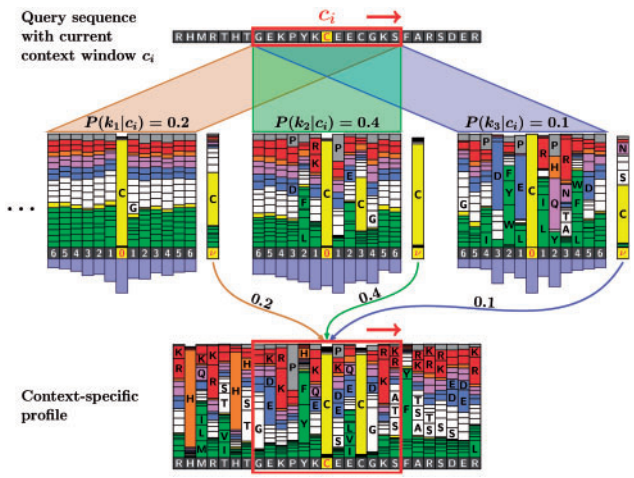

Fig. 1. Discriminative model for computing context-specific substitution probabilities. For each position i in the query, the context $\mathbf{C}_i$ (red window) is compared with all context states k. The 13-column coloured histogram blocks show the distribution of amino acids in the context states [box height $\propto \exp(\lambda_k(j,a))$]. Blue bars indicate the 'weight' of each column [mean absolute deviation of the $\lambda_k(j,a)$ from their median]. The emission probabilities $P(a|k)$ of the context states are represented by separate histogram columns. These emission probabilities are weighted with $P(k|\mathbf{C}_i)$ and summed up to produce the column of substitution probabilities in the context-specific profile. CS-BLAST then jump-starts PSI-BLAST with this profile

exponential of the mean entropy of the amino acid probabilities over all profile columns (Supplementary Material).

The generative model was trained by maximizing the product over all probabilities $P(\mathbf{C}_n)$ that context $\mathbf{C}_n$ can be generated by a mixture of K context profiles p_k. However, the actual goal should rather be to predict $P(a|\mathbf{C}_n)$, the frequency of a given context $\mathbf{C}_n$, not to learn the distribution of the contexts $\mathbf{C}_n$, which are observed anyway. We, therefore, trained our model parameters π, λ, ν by maximizing the logarithm of the conditional probability of target amino acid distributions $\widetilde{\mathbf{c}}_n$ given the observed sequence contexts $\mathbf{C}_n$ in the training set as follows:

$$f(\pi, \lambda, \nu) = \log\left(P(\pi, \lambda, \nu)\prod_{n=1}^{N} P(\widetilde{\mathbf{c}}_n|\mathbf{C}_n, \pi, \lambda, \nu)\right) \overset{\pi,\lambda,\nu}{\longrightarrow} \max \quad (4)$$

Here, $P(\widetilde{\mathbf{c}}_n|\mathbf{C}_n, \pi, \lambda, \nu)$ follows a multinomial distribution,

$$\log P(\widetilde{\mathbf{c}}_n|\mathbf{C}_n) = \sum_{a=1}^{20} P(a|\mathbf{C}_n, \pi, \lambda, \nu)^{\widetilde{\mathbf{c}}_n(a)} + \text{const} \quad (5)$$

whose parameters $P(a|\mathbf{C}_i, \pi, \lambda, \nu)$ are calculated according to Equations (1–3). The previous probability $P(\pi, \lambda, \nu)$ is modelled as product of Gaussian distributions with zero means and with SDs $\sigma_\pi, \sigma_j = \sigma_{\text{center}}\gamma^{|j|}$ and σ_ν as follows:

$$P(\pi, \lambda, \nu) \propto \exp\sum_{k=1}^{K}\left(-\frac{\pi_k^2}{2\sigma_\pi^2} - \sum_{j=-d}^{d}\sum_{a=1}^{20}\frac{\lambda_k(j,a)^2}{2\sigma_j^2} - \sum_{a=1}^{20}\frac{\nu_k(a)^2}{2\sigma_\nu^2}\right)$$

We used stochastic gradient descent (Bottou, 2004) with initial learning rate η_0 for optimizing the log conditional probability f in Equation (4) (see Supplementary Material).

The approach described so far can easily be generalized to sequence profiles in a way that all equations (1–5) remain valid without change. A sequence profile q is built from a MSA of sequences that are homologous to the query sequence. The profile probabilities $q(i, a)$ correspond to the relative frequencies of residues $a \in \{1, \ldots, 20\}$ in alignment column i. The context $\mathbf{C}_i$ describes the number of effective residue counts $\mathbf{C}_i(j, a)$ at positions $(i-d, \ldots, i+d)$ of profile q, $\mathbf{C}_i(j, a) = N_q(i+j)\,q(i+j, a)$, where $N_q(i)$ is the effective number of sequences in column i of the query MSA.

2.3 CS-BLAST and CSI-BLAST

Our homology search tool CS-BLAST extends BLAST by context-specific (CS) substitution probabilities that are either derived with the generative model (CS-BLAST*gen*) or with the discriminative model (CS-BLAST*dis*). Analogously, CSI-BLAST is our context-specific iterative (CSI) version of position-specific iterative BLAST (PSI-BLAST); given a query profile q, CSI-BLAST first computes the pseudocount profile with substitution probabilities $P(a|\mathbf{C}_i)$, and then mixes it with the query profile $q(i, a)$ in a way similar to PSI-BLAST (Altschul *et al.*, 1997) as follows:

$$p_{\text{cs}}(i, a) = (1 - \tau_i)q(i, a) + \tau_i P(a|\mathbf{C}_i). \quad (6)$$

The pseudocount admixture coefficient τ_i,

$$\tau_i = \varphi\frac{\psi + 1}{\psi + N_q(i)} \quad (7)$$

attains its maximum of φ when the query profile consists of a single sequence ($N_q(i) = 1$). When the effective number of sequences $N_q(i)$ is large, the relative contribution of the pseudocount profile is reduced.

We then jump-start PSI-BLAST with a checkpoint file containing the profile matrix p_{cs} multiplied by a constant 2^δ. The profile-to-sequence bit score between profile column i and residue a that PSI-BLAST calculates from this checkpoint file is

$$S(p_{\text{cs}}(i, \cdot), a) = \log_2\left(\frac{p_{\text{cs}}(i, a)2^\delta}{P(a)}\right) \quad (8)$$

where $P(a)$ is the background probability of a. The factor 2^δ translates into a constant score offset of δ bits. This offset controls the trade-off between the alignment sensitivity and the alignment precision (see Section 3.3).

3 RESULTS

3.1 Datasets and parameter optimization

The structural classification of proteins (SCOP) database (Murzin *et al.*, 1995) provides a hierarchical clustering of protein domains with known structures and is the *de facto* standard for evaluating sequence search tools. We filtered the SCOP database with a maximum pairwise sequence similarity of 20% (SCOP20) and also 30% (SCOP30), 40% (SCOP40), 60% (SCOP60) and 80% (SCOP80). We randomly assigned every fifth fold to the optimization set (1329 sequences, 215 folds in SCOP20) and all remaining folds to the test set (5287 sequences, 862 folds in SCOP20). This ensures that the optimization set does not share homologous sequences with the test set. We performed an all-against-all comparison and defined members belonging to the same fold as *true positives* (TPs) and those of different folds as *false positives* (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 Rossman-like folds (*c*.2–*c*.5, *c*.30, *c*.66, *c*.78, *c*.79, *c*.111) and α-helical and 4Fe-4S ferredoxins (*a*.1.2, *d*.58.1).

The discriminative model has several adjustable parameters. As shown in (Biegert and Söding, 2009), not only the sensitivity but also the computation time increase with the number of context states K and the window length l. We chose $K = 4000$ and $l = 13$ as a trade-off between sensitivity and run time. Further parameters are the pseudocount admixture parameters φ and ψ, the score offset δ, the previous parameters $\sigma_\pi, \sigma_{\text{center}}, \gamma$ and σ_ν, and the initial learning rate η_0. The generative model had positional weight factors w_{center} and β instead of the previous parameters.

The optimum setting of the parameters for the generative and discriminative models was determined by maximizing the mean receiver operating characteristic five (ROC5) score on the optimization set. The mean ROC5 score is the same as the area under the ROC5 curve, which is explained in Section 3.2. The mean ROC5 score is a single numerical value that measures the mean sensitivity on all query sequences and is robust with respect to overtraining.

We iteratively optimized each parameter in turn using line search, several times for each parameter. We found $(\varphi = 0.88, \psi = 14, w_{\text{center}} = 1.6, \beta = 0.85)$ as the optimum parameter setting for the generative model and $(\varphi = 1.0, \psi = 15, \sigma_\pi = 1.0, \sigma_{\text{center}} = 1.6, \gamma = 0.85, \sigma_\nu = 1.0,$

$\eta_0 = 0.13$) for the discriminative model. The score offset δ was manually set. Choosing a negative score offset increases the alignment precision (Fig. 3) and the reliability of the reported *E*-values (Supplementary Fig. S4), but it simultaneously decreases the alignment sensitivity. As a compromise, we chose $\delta = -0.005$ bits for both models.

3.2 Sensitivity

We analysed the sensitivity of NCBI BLAST (blastpgp, version 2.2.26), CS-BLAST*gen* and CS-BLAST*dis* by using two complementary methods, the receiver operating characteristic (ROC) analysis and the ROC5 analysis.

The ROC plot (Fig. 2A) shows the number of TPs versus FPs up to a certain *E*-value threshold for the SCOP20 test set. It measures how well the matches are ranked by the *E*-value *across all database searches*. In Söding and Remmert (2011), it was argued that it is important to weight down the contribution of FP and TP pairs from large superfamilies to avoid large superfamilies from dominating the ROC plot. Indeed, in Biegert and Söding (2009), we reported that CS-BLAST is 139% more sensitive than BLAST at a false discovery rate (FDR) of 20% if FPs and TPs are weighted by 1/(size of the query's family). This number drops to 40% if the size of the *superfamily* is used instead. To be even more conservative, we, therefore, used *fold*-weighted FPs/TPs.

CS-BLAST*dis* detects 20 and 17% more homologues than CS-BLAST*gen* at a FDR of 1 and 10%, respectively. Compared with BLAST, CS-BLAST*dis* finds 43 and 51% more homologues, respectively. The improvement of CS-BLAST*gen* over BLAST could be nearly doubled by CS-BLAST*dis*. If FPs and TPs are weighted by the reciprocal size of the query's family instead of its fold, the improvements are even stronger (Supplementary Fig. S1).

We tested the sensitivity of the iterative search tool CSI-BLAST by performing all but the last search iteration against NCBI's non-redundant database with 16 million sequences to create a MSA (*E*-value inclusion threshold 10^{-3}), which was then used for searching the SCOP20 database. With two iterations, CSI-BLAST*dis* is 4.7 and 4.5% more sensitive than CSI-BLAST*gen* at an FDR of 1 and 10%, respectively, and 26.9 and 21.5% more sensitive than BLAST, respectively. Note that two iterations CSI-BLAST*dis* yield better results than five iterations of PSI-BLAST.

In contrast to the ROC plot, the ROC5 plot (Fig. 2B) reveals how reliably hits are ranked *within each database search*. It is defined as the normalized area under the ROC curve until the fifth FP. An ROC5 plot shows the fraction of query sequences whose ROC5 score is above the value on the *x*-axis. It is more robust than the ROC plot analysis, which is prone to overtraining, as a few families of high-scoring FPs can greatly influence the ROC plot (Söding and Remmert, 2011).

CS-BLAST*dis* improves the mean ROC5 score by 9.5% over CS-BLAST*gen* and by 39.46% over BLAST. The sensitivity increases most for cases with a mean ROC5 score between 0.1 and 0.5. After two search iterations, the ROC5 score is still 4.2 and 19% higher compared with CSI-BLAST*gen* and PSI-BLAST, respectively. Figure 2C and Supplementary Figure S1 show that the improvements are still appreciable in databases

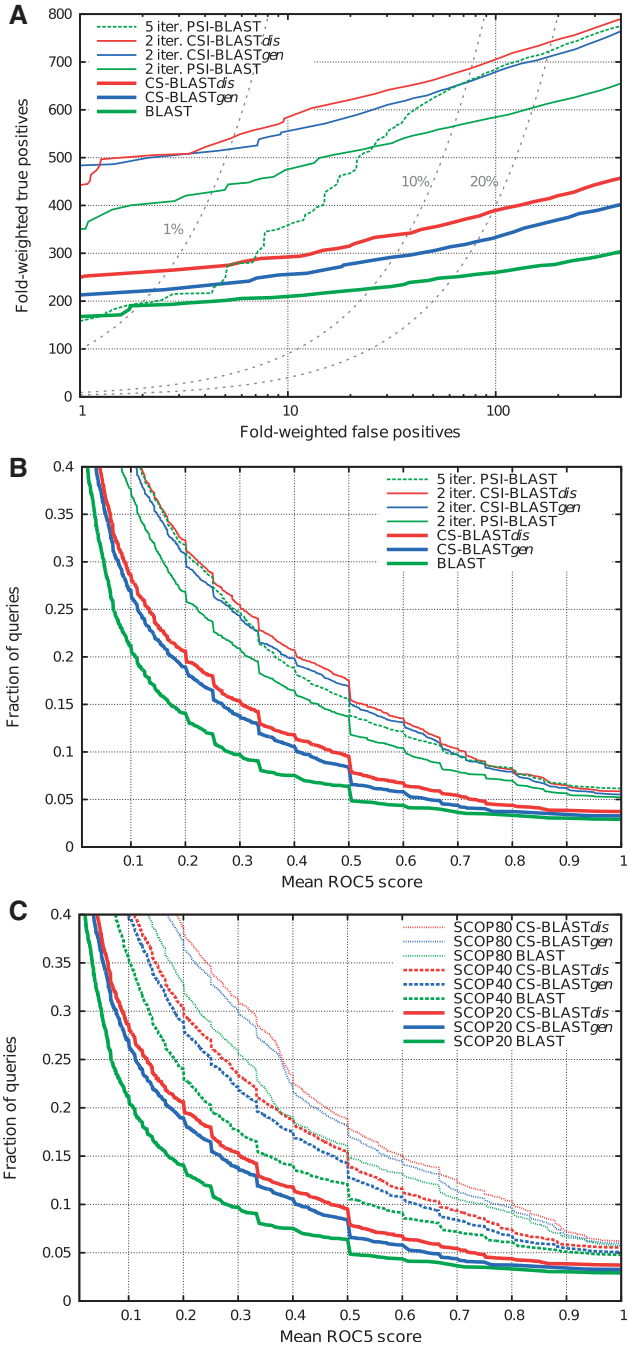

Fig. 2. Sensitivity to detect homologous sequences for BLAST, PSI-BLAST, the generative and discriminative versions of CS-BLAST (CS-BLAST*gen*, CS-BLAST*dis*), and CSI-BLAST. (**A**) ROC plot showing the number of true-positive results found (same fold) versus false-positive results (different fold), weighted by 1/(size of the query's fold), on the SCOP20 test set. Dashed lines indicate FDR of 1, 10 and 20%. (**B**) ROC5 plot showing the fraction of queries whose ROC5 score is above the value on the *x*-axis (on SCOP20 test set). (**C**) As in B, but comparing the performance on test sets SCOP20, SCOP40 and SCOP80

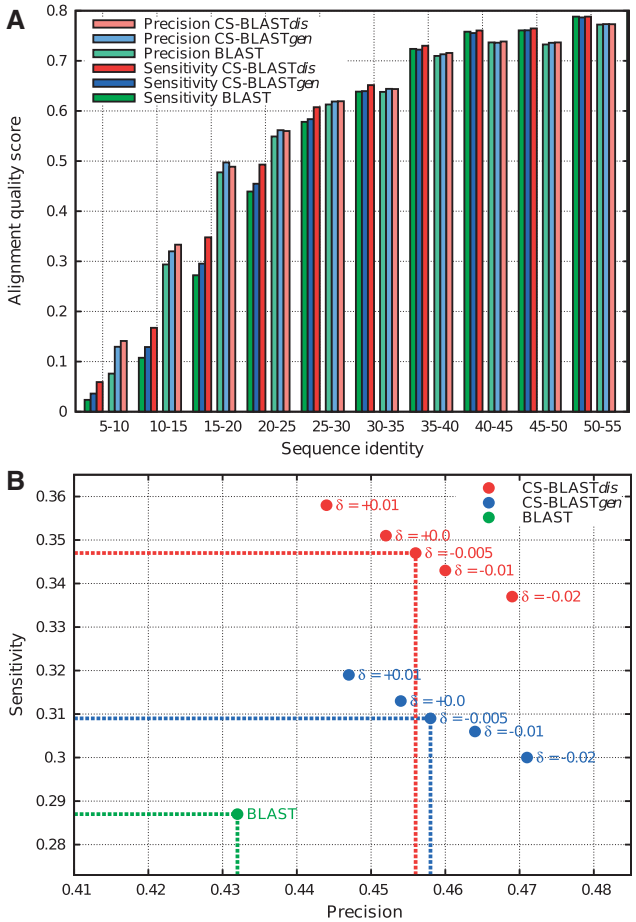

Fig. 3. Alignment quality assessment. Structural alignments by TM-align with TM-score ≥ 0.6 were used as gold standard. (**A**) Alignment precision and sensitivity binned by sequence identity for 7663 sequence pairs sampled from SCOP80. (**B**) Average alignment sensitivity versus precision depending on the score offset δ for 3700 sequence pairs sampled from SCOP30 (pairwise sequence identity $<30\%$). The default score offset is -0.005 bits (dashed lines). CS-BLAST*dis* alignments have a similar precision to those of CS-BLAST*gen* but contain, on average, 12% more correctly aligned residue pairs

SCOP40, and SCOP80 containing sequence pairs with up to 40 and 80% sequence identity, respectively.

These results demonstrate that the discriminative approach for predicting context-specific substitution probabilities improves the homology detection performance both of sequence and profile searches.

3.3 Alignment quality

A high quality of pairwise and multiple sequence alignments is essential for many downstream applications. For example, secondary structure prediction could be significantly improved by simply generating MSAs using HHblits instead of PSI-BLAST (Remmert *et al.*, 2011). In homology modelling, target-template alignment quality is still the bottleneck for more accurate models. Here, we assessed the quality of CS-BLAST*dis* alignments by comparing them to reference alignments obtained from pairwise structural alignments.

We sampled up to 10 sequence pairs from each family in SCOP80 (for Fig. 3A) and SCOP30 (for Fig. 3B), yielding 37 663 and 3700 pairs, respectively. We built reference alignments for them using the structural aligner TM-align (Zhang and Skolnick, 2005). These alignments were compared with sequence alignments built by BLAST (blastpgp, version 2.2.26), CS-BLAST*gen* and CS-BLAST*dis*. To make sure that BLAST produced an alignment for each pair of sequences in the test set, we decreased the threshold for extending hits (option -f) from 11 to 8. We used blastpgp's -s option to build alignments with the Smith–Waterman algorithm. We then evaluated the precision and sensitivity of each alignment. The *alignment precision* is the number of correctly aligned pairs divided by the number of aligned pairs. The *alignment sensitivity* is the number of correctly aligned pairs divided by the number of aligned pairs in the reference alignment.

Up to a sequence identity of 50%, context-specific substitution scores improve the alignment sensitivity compared with block substitution matrix (BLOSUM62) scores used in BLAST (Fig. 3A). For pairs with sequence identities $<30\%$, the alignment precision improves on average by 5.6% compared with BLAST (Fig. 3B). The sensitivity of CS-BLAST*dis* alignments is 12.3% higher than for CS-BLAST*gen* and 21% higher than for BLAST. All in all, the improvement in alignment quality from CS-BLAST*gen* to CS-BLAST*dis* is of similar magnitude as the improvement from BLAST to CS-BLAST*gen*. The most striking improvements are seen in the difficult alignments <25% pairwise sequence identity (Fig. 3A).

The score offset δ in Equation (8) allows CS-BLAST users to control the trade-off between the rate of alignment errors and the length of the alignment or, in other words, between alignment precision and sensitivity. A higher score offset allows CS-BLAST to extend alignments for longer while still accumulating positive score contributions. This leads to longer, less precise but more sensitive alignments (Fig. 3B).

3.4 CS-BLAST *E*-values

E-values are used to assess the significance of sequence search results. Their correctness is essential for iterative search tools, which automatically add all hits with *E*-values below a specified threshold to the MSA for the next search iteration. To estimate the reliability of *E*-values, we plotted the *reported E-values* of all hits from an all-against-all comparison on the SCOP test set (see Section 3.2) against their *actual E-value* (Supplementary Fig. S3). The actual *E*-value was estimated as the number of observed FPs with an *E*-value below the reported *E*-value on the *x*-axis, averaged over all searches.

We used the score offset δ for adjusting the reliability of the reported *E*-values. High-scoring alignments between non-homologous sequences mainly occur between compositionally biased or repetitive regions. These alignments tend to have weak similarities over relatively long stretches. Therefore, shifting the substitution scores towards the negative range can shrink these alignments and reduce their score, effectively suppressing high-scoring false-positive matches (Supplementary Fig. S2). As a trade-off between *E*-value reliability and alignment

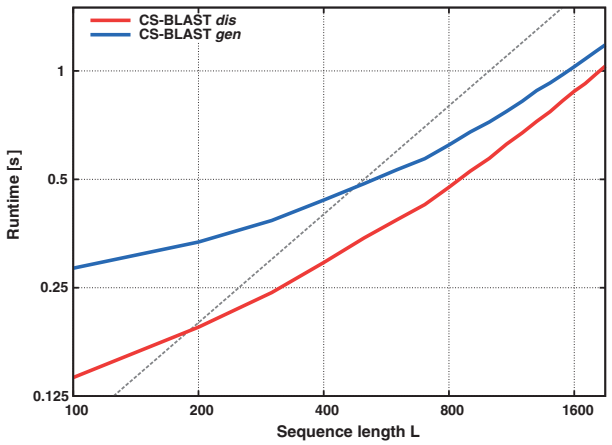

Fig. 4. Run time for generating the context-specific profile p_{cs} of length L using $K = 4000$ context-states of length $l = 13$, measured on an Intel 1.8 GHz quad-core processor. CS-BLAST*dis* is 1.6 times faster than CS-BLAST*gen* for a sequence length of $L = 350$ residues

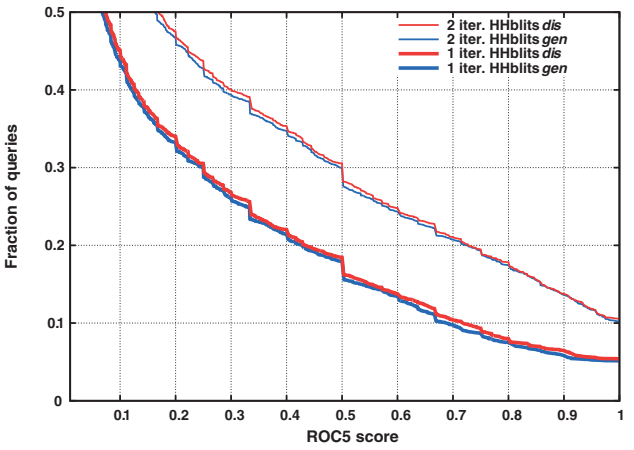

Fig. 5. ROC5 plot showing the fraction of queries whose ROC5 score is below the value on the x-axis. Using HHblits along with context-specific pseudocounts predicted by the discriminative model (HHblits*dis*) instead of the generative model (HHblits*gen*) increases the mean ROC5 score by 3.0% (one iteration) and 1.5% (two iterations)

sensitivity (Fig. 3B), we choose $\delta = -0.005$ bits. This choice made CS-BLAST's E-values as reliable as BLAST's (Supplementary Figs S3 and S4).

3.5 Run time

We compared the run time of CS-BLAST*gen* and CS-BLAST*dis* for generating the context-specific profile p_{cs} for sequences of length L, which were sampled from the SCOP database (Fig. 4). The time complexity for both models is $\mathcal{O}(LKl \times 20)$. We used $K = 4000$ context states of length $l = 13$. CS-BLAST*dis* requires ~0.3 s for computing p_{cs} on an Intel quad-core processor with 1.8 GHz given an average sequence length of $L = 350$ residues. CS-BLAST*dis* is ~1.6 times faster than CS-BLAST because Equation (2) can be computed more efficiently. The context specific substitution probabilities $P(a|c_i)$ are computed in parallel for each position i, and the run time scales inversely with the number of processor cores.

3.6 HHblits sensitivity

HHblits (Remmert *et al.*, 2011) is an iterative homology search tool based on HMM-to-HMM (HH) comparison that is not only faster than PSI-BLAST but also much more sensitive and accurate. This is achieved by representing both the query and the database sequences by HMMs, which are derived from MSAs of related sequences and, therefore, contain much valuable evolutionary information. HHblits is an extension of HHsearch (Söding, 2005), and both are used in top ranking structure prediction servers like HHpred (Mariani *et al.*, 2011). HHblits can be used for fast, iterative searches through clustered versions of large sequence databases, such as universal protein resource (UniProt) or NCBI's non-redundant (NR) database. At the beginning of each search iteration, HHblits uses context-specific substitution probabilities to enrich the query HMM, which is then compared with the database HMMs. Hits with E-values below a predefined threshold are added to the query MSA, from which the HMM for the next iteration is built.

We wanted to test whether the discriminative model for context-specific substitution probabilities can further improve the sensitivity of HHblits searches. The present version of HHblits already uses context-specific substitution probabilities computed with the generative model. The same optimization and test set were used as described in Section 3.1. Models with $K = 4000$ context states of length $l = 13$ were compared for one and two iterations of HHblits through the uniprot20_02Sep11 database offered together with HHblits. We optimized the admixture parameters φ and ψ as described in Section 3.1. For one iteration, we found ($\varphi = 0.95$, $\psi = 24$) and ($\varphi = 1.0$, $\psi = 14$) for the generative and discriminative model, respectively, and for two iterations, ($\varphi = 0.58$, $\psi = 10$) and ($\varphi = 0.58$, $\psi = 18$) were optimal. The positional weight factors of the generative model and previous parameters were left as in Section 3.1.

The mean ROC5 score, after one iteration HHblits on our SCOP20 test set, could be improved by 3.0% through the discriminative model (Fig. 5). At two iterations, we could gain 1.5% in sensitivity. In relative terms, this is much less than the improvements on single sequences because the HMMs after the first iteration of HHblits already contain detailed position-specific substitution probabilities from homologous sequences. However, the absolute improvement in ROC5 score is retained after the second iteration.

4 DISCUSSION

4.1 Information in sequence context

What amino acid substitutions are likely to be observed in homologous proteins is largely determined by the fixation probability of mutations, which depends mostly on whether the protein can still fold into its stable structure. The folding requirement exerts specific constraints on different positions depending on their local structural context. The substitution matrix method merely

assesses the similarity of physicochemical properties between the original and the mutated amino acid. In contrast, the sequence context-specific method indirectly makes a fine-grained prediction about the probabilities of various structural contexts and, hence, exquisitely captures the selective constraints at each position. For example, a leucine within a membrane helix will have different substitution probabilities from a leucine on the hydrophobic side of an amphipathic β-strand or from a leucine within a natively unfolded region. The dramatic improvements in sensitivity and alignment quality bear testimony to the value of sequence contexts for predicting substitution probabilities.

4.2 Generative versus discriminative model

Both the discriminative approach and the previously proposed generative approach for computing context-specific substitution probabilities are based on a library of context states that each represents a specific typical sequence context. Why does the discriminative model perform considerably better than the generative model?

First, in contrast to the generative model that describes the joint distribution $P(\mathbf{C}_i, k)$ and then indirectly infers $P(k|\mathbf{C}_i)$ through Bayes' theorem, in the discriminative approach we directly model $P(k|\mathbf{C}_i)$. We do not model the probability distribution of the input contexts $\mathbf{C}_i$, which is not needed, as the contexts are observed anyway. Second, the discriminative model learns the importance of the various positions individually for each sequence context, whereas in the generative model, we had to set the same weights for all sequence contexts explicitly (Supplementary Material). Third, the discriminative model has a set of parameters $\nu_k(a)$ to model the emission probabilities independent of the context information, whereas in the generative model, the emission distribution is identical to the central context profile column. This gives the discriminative model more flexibility to optimize its objective function. As a consequence of these three points, the generative model with any parameters can be accurately reproduced by the discriminative model with appropriately chosen parameters [Supplementary Material, Equations (5–7)], showing that the discriminative model has a higher descriptive potential. Fourth, the discriminative model is trained by optimizing the conditional probability of the target variable (the substitution probabilities) given the observed variable (the sequence context). Hence, in contrast to the generative model, it is trained to maximize some measure of prediction quality.

4.3 Model training

Although discriminative models typically excel for unlimited amounts of training data, generative models often perform better in practice when training data are scarce or training time is limiting (Ng and Jordan, 2001). The comparison of our discriminative and generative approach is a case in point, as training the generative model worked well by simple expectation maximization, whereas training the discriminative model turned out to be challenging. First, we had to increase the number of training samples from 1.0 to 6.0 million, in line with the findings by Ng and Jordan (2001). Second, the insight that each generative model can be written exactly as a discriminative model allowed us to initialize the discriminative model with the equivalent generative model parameters. Third, we had to try out several optimization techniques to obtain satisfactory results; straightforward stochastic gradient descent worked better in the end than the much more sophisticated Hybrid Monte Carlo algorithm combined with replica exchange Monte Carlo sampling (Neal, 1993). Also, using a simple hyperbola function for controlling the learning rate of stochastic gradient descent and optimizing the initial learning rate η_0 (Supplementary Material) proved better than various schemes for dynamic learning rate adaptation proposed by Almeida *et al*. (1998).

5 CONCLUSION

The present work shows that a discriminative approach to context-specific sequence comparison can further improve the sensitivity of sequence searches and, in particular, the alignment quality over a previous generative approach. This was demonstrated by comparing BLAST and PSI-BLAST with its context-specific versions (CS-BLAST*gen* and CS-BLAST*dis*). The new discriminative model also increased the sensitivity of our iterative sequence search method HHblits and, hence, will become the default pseudocount model in CS-BLAST and HHblits. We are convinced that the context-specific approach for substitution probabilities presented here is clearly superior to context-ignorant approaches, such as substitution matrices and Dirichlet mixtures pseudocounts (Sjölander *et al*., 1996), and should supersede those methods wherever maximum performance is critical.

ACKNOWLEDGEMENT

The authors thank Stefan Seemayer, Armin Meier and Markus Meier for discussions.

Funding: Deutsche Forschungsgemeinschaft (SFB646 and GRK1721); Bavarian Systems Biology Network (BaySysNet) financed by the Bavarian Ministry for Science, Research and Arts; LMU through the Excellence Initiative of the BMBF.

Conflict of Interest: none declared.

REFERENCES

Almeida,L. *et al*. (1998) Parameter adaptation in stochastic optimization. In: *Online Learning in Neural Networks*. Cambridge University Press, New York, pp. 111–134.

Altschul,S.F. *et al*. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*., **25**, 3389–3402.

Baussand,J. *et al*. (2007) Periodic distributions of hydrophobic amino acids allows the definition of fundamental building blocks to align distantly related proteins. *Proteins*, **67**, 695–708.

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

Bottou,L. (2004) Stochastic learning. *Lect. Notes Comput. Sci*., **3176**, 146–168.

Caruana,R. and Mizil,A.N. (2006) An empirical comparison of supervised learning algorithms. In: *Proceedings of 23rd International Conference Machine Learning, (ICML 06)*, ACM, New York, NY. pp. 161–168.

Dayhoff,M.O. *et al*. (1972) A model of evolutionary change in proteins. In: Dayhoff,M. (ed.), *Atlas of Protein Sequence and Structure*, vol. 5. National Biomedical Research Foundation, Washington, DC, pp. 345–352.

Goonesekere,N.C. and Lee,B. (2008) Context-specific amino acid substitution matrices and their use in the detection of protein homologs. *Proteins*, **71**, 910–919.

Henikoff,S. and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.

Huang,Y.-M. and Bystroff,C. (2006) Improved pairwise alignments of proteins in the twilight zone using local structure predictions. *Bioinformatics*, **22**, 413–422.
Jones,D.T. *et al.* (1994) A mutation data matrix for transmembrane proteins. *FEBS Lett.*, **339**, 269–275.
Mariani,V. *et al.* (2011) Assessment of template based protein structure predictions in CASP9. *Proteins*, **79**, 37–58.
Murzin,A.G. *et al.* (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.
Neal,R. (1993) Probabilistic inference using markov chain monte carlo methods. In: *Technical report CRG-TR-93-1*. Department of Computer Science, University of Toronto.
Ng,A.Y. and Jordan,M.I. (2001) On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. *Adv. Neural Inf. Process Syst.*, **14**, 841–848.
Overington,J. *et al.* (1992) Environment-specific amino acid substitution tables: tertiary templates and prediction of protein folds. *Protein Sci.*, **1**, 216–226.
Remmert,M. *et al.* (2011) HHblits: lightning-fast iterative protein sequence searching by HMM–HMM alignment. *Nat. Methods*, **9**, 173–175.
Rice,D.W. and Eisenberg,D. (1997) A 3D-1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence. *J. Mol. Biol.*, **267**, 1026–1038.
Rubinstein,Y.D. and Hastie,T. (1997) Discriminative versus informative learning. In: *Proceedings of Third International Conference on Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence, Newport Beach, CA, USA, pp. 49–53.
Shi,J. *et al.* (2001) FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J. Mol. Biol.*, **310**, 243–257.
Sjölander,K. *et al.* (1996) Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Comput. Appl. Biosci.*, **12**, 327–345.
Söding,J. (2005) Protein homology detection by HMM–HMM comparison. *Bioinformatics*, **21**, 951–960.
Söding,J. and Remmert,M. (2011) Protein sequence comparison and fold recognition: progress and good-practice benchmarking. *Curr. Opin. Struct. Biol.*, **21**, 404–411.
Sutton,C. and McCallum,A. (2006) Introduction to conditional random fields for relational learning. In: Getoor,L. and Taskar,B. (eds.), *Introduction to Statistical Relational Learning*. MIT Press, Cambridge, USA, pp. 93–128.
Zhang,Y. and Skolnick,J. (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.*, **33**, 2302–2309.