Discriminative modeling of context-specific amino acid substitution probabilities

C. Angermüller\textsuperscript{1}, A. Biegert\textsuperscript{2} and J. Söding\textsuperscript{1*}

\textsuperscript{1}Gene Center Munich and Center for Integrated Protein Science (CIPSM), Ludwig-Maximilians-Universität München, Feodor-Lynen-Str. 25, 81377 Munich, Germany and \textsuperscript{2}Genedata, Lena-Christ-Strasse 50, 82152 Martinsried, Germany

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

1 THE GENERATIVE MODEL

In this model, each context state $k$ is described by a context profile. These are sequence profiles of length $l = 2d + 1$, where $p_k(j, a)$ is the frequency of amino acid $a$ in column $j \in \{-d, \ldots, d\}$ of the context state. The emission probabilities $P(a|k)$ are chosen to be identical to the distribution in the central column of $p_k$:

$$P(a|k) = p_k(0, a) \quad (1)$$

The model parameters are the prior probabilities of the context states, $P(k) = \alpha_k$, and the set of context profiles, $p_k(j, a)$ $\forall k \in \{1, \ldots, K\}, \forall j \in \{-d, \ldots, d\}, \forall a \in \{1, \ldots, 20\}$, which we conveniently summarize as $(\alpha, p)$. The model is generative since it models the joint distribution $P(C_i, k)$, which implies a model for the input context profile $C_i$:

$$P(C_i, k) = P(C_i|k)P(k)$$

$$= \prod_{j=-d}^{d} \left(M_j(C_i) \sum_{a=1}^{20} p_k(j, a)^{C_i(j, a)}\right)^{\alpha_k} \quad (2)$$

Here, $M_j(C_i)$ normalizes the multinomial distribution, which gives the probability of observing counts $C_i(j, \cdot)$ of column $j$ given the corresponding profile column $p_k(j, \cdot)$. The positional weights $w_j = w_{\text{center}} \beta^j (\beta \in [0, 1])$ determine the contribution of each column and decrease exponentially with the distance from the central column. $P(k|C_i)$ is inferred from Equation (2) via Bayes' theorem:

$$P(k|C_i) = \frac{P(C_i|k)P(k)}{\sum_{k'} P(C_i|k')P(k')}$$

$$= \frac{\prod_{j=-d}^{d} \left(M_j(C_i) \sum_{a=1}^{20} p_k(j, a)^{C_i(j, a)}\right)^{\alpha_k}}{\sum_{k'=-d}^{d} \prod_{j=-d}^{d} \left(M_j(C_i) \sum_{a=1}^{20} p_{k'}(j, a)^{C_i(j, a)}\right)^{\alpha_{k'}}} \quad (3)$$

For training the generative model, $N = 10^6$ context windows $C_n$ were sampled from a set of 50,000 MSAs generated from randomly sampled sequences in NCBI’s non-redundant database. We then maximized the likelihood that these context windows were generated by the context profiles

$$\mathcal{L}(\alpha, p) = \prod_{i=1}^{N} P(C_n; \alpha, p)$$

$$= \prod_{n=1}^{N} \sum_{k=1}^{K} P(C_n|k; \alpha, p)P(k; \alpha, p)^{\alpha_k} \max \quad (4)$$

which was carried out by using the Expectation Maximization algorithm (Biegert and Söding, 2009). Note that maximizing the likelihood $\mathcal{L}$ is equivalent to maximizing the joint probability $P(C_n, \hat{c}_n)$, where $\hat{c}_n(a) = C_n(0, a)$ is the target variable.

2 THE DISCRIMINATIVE MODEL SPACE CONTAINS THE GENERATIVE MODEL SPACE

In the following we will show that the generative model with any set of parameters is equivalent to the discriminative model with an appropriately chosen set of parameters. In other words, the discriminative model with these particular parameters predicts the same context-specific substitution probabilities $P(a|C_i)$ as the generative model. This is very useful for initializing the parameters of the discriminative model for training, because the discriminative model tends to get trapped in local optimima with very unsatisfactory performance. Initializing the training with the parameters of the generative model was an important measure to improve the training of the discriminative model.

Suppose the generative model has $K$ context states of length $l = 2d + 1$ and parameters $\alpha_k$, $p_k(j, a)$, and $w_j = w_{\text{center}} \beta^j$. We set the parameters of the discriminative model to the following values:

$$\pi_k = \log \alpha_k \quad (5)$$

$$\lambda_k(j, a) = w_j \log p_k(j, a) \quad (6)$$

$$\nu_k(a) = \log p_k(0, a) \quad (7)$$

We will show that this produces identical predictions $P(a|C_i)$ by proving that the emission probabilities $P(a|k)$ and the context state probabilities $P(k|C_i)$ agree between both models. The emission
probabilities are
\[
P(a|k) = \frac{\exp(v_k(a))}{\sum_{a' = 1}^{20} \exp(v_k(a'))} = \frac{\exp(\log p_k(0, a))}{\sum_{a' = 1}^{20} \exp(\log p_k(0, a'))} = p_k(0, a)
\]
which is identical to the emission probability of the generative model [Main Text Eq. (2)]. Furthermore, \(P(k|\mathcal{C})\) for the discriminative model is
\[
P(k|\mathcal{C}) = \frac{1}{Z(\mathcal{C})} \exp \left( \sum_{j, a} \lambda_k(j, a) C_i(j, a) \right)
\]
\[
= \frac{1}{Z(\mathcal{C})} \exp \left( \sum_{j, a} \lambda_k(j, a) \log p_k(j, a) C_i(j, a) \right)
\]
\[
= \frac{1}{Z(\mathcal{C})} \alpha_k \prod_{j, a} p_k(j, a) C_i(j, a)^{w_j}
\]
\[
= \frac{1}{Z(\mathcal{C})} \alpha_k \prod_{j, a} \left( \prod_{i = 1}^d p_k(j, a) C_i(j, a)^{w_j} \right) \alpha_k
\]
which is the same as \(P(k|\mathcal{C})\) of the generative model (Eq. 1). Since Equation (8) and (9) are true for any state \(k\), the \(P(a|k)\) will also be the same [Main Text Eq. (4)]. Consequently, Equations (7–9) map each generative model onto an equivalent discriminative model.

3 TRAINING

We chose \(K = 4000\) context states of length \(l = 13\) as a compromise between speed and performance (Biegert and Söding, 2009). For building the training set, we clustered the UniProt database into groups with a maximum intergroup sequence identity of 20% using our in-house method kClust (M. Hauser, C.E. Mayer, J. Söding, submitted). We used the complete UniProt database to ensure that the training set covered sequence contexts from all classes, e.g. membrane helices or disordered regions, both from sequences with known and unknown structure. The resulting clusters were enriched by homologous sequences found by up to four iterations HHblits, and the search was terminated when the number of effective sequences in the cluster surpassed 4.0. Next, cluster MSAs with an effective number of sequences smaller than 4.0 or larger than 10.0 were discarded in order to derive the substitution probabilities over an evolutionary distance similar to the BLOSUM 62 matrix. The remaining MSAs were used to build \(N = 6.0 \times 10^5\) training pairs (\(C_1, \epsilon_1\), ..., \(C_N, \epsilon_N\)), where (\(C_n, \epsilon_n\)) was sampled from an MSA with query sequence \(x\) and sequence profile \(q\) at position \(i(n)\). \(\epsilon_i\) describes the context of \(x\) at position \(i(n)\), i.e. \(C_i(j, a) = I(\epsilon_i(j), a + j = a)\). The vector \(\epsilon_n\) stores how often each residue \(a\) occurs in alignment column \(i(n)\), i.e. \(\epsilon_n(a) = q(i(n), a) N_n(i(n))\).

We employed mini-batch gradient descent (Bottou, 2004) with deterministic learning rate adaptation for finding the parameters \(\theta = (\pi, \lambda, \nu)\) to maximize the objective function [Main Text Eq. (6)].

Let \(f\) be the objective function and \(\theta^{(t)}\) the parameters in iteration \(t\). Then the parameters \(\theta^{(t+1)}\) for the next iteration are computed as follows:
\[
\theta^{(t+1)} = \theta^{(t)} + \eta_t \sum_{n = \text{ran}(1), \ldots, \text{ran}(b)} \frac{\partial f(\theta^{(t)}, C_n, \epsilon_n)}{\partial \theta}
\]
Unlike computing the gradient over all training points (batch gradient descent) or a single training point (stochastic gradient descent), the gradient is computed over \(b\) randomly chosen training points \(\text{ran}(1), \ldots, \text{ran}(b) \in \{1, \ldots, N\}\) (Bottou and Bousquet, 2008). The batch size \(b\) allows us to control the trade-off between low noise (large \(b\)) and fast convergence (small \(b\)). As such, the procedure is a compromise between gradient descent (for which \(b = N\)) and stochastic gradient descent (for which \(b = 1\)). To initialize the discriminative model, the parameters of the generative model described in Biegert and Söding (2009) were mapped onto the initial parameters \(\theta^{(0)}\) according to Equations (1–3).

The learning rate \(\eta_t\) was determined by
\[
\eta_t = \eta_0 \frac{t(d-1)}{10} + 1
\]
where \(\eta_0\) is the initial learning rate and \(d\) is the factor by which the learning rate is reduced after the gradient has been computed over \(10^d\) training samples. Equation (11) satisfies the conditions for almost sure convergence of mini-batch gradient descent to the global optimum (Bottou, 2004). This simple scheme for adapting the learning rate proved to be more robust for optimizing the objective function than the stochastic learning rate adaptation proposed by Almeida et al. (1998). We found \(b = 3000\) and \(d = 2.5\) to be near-optimal values at \(K = 200\) and kept these values constant for all further runs. We optimized \(\eta_0\) together with the other parameters (\(\sigma_\pi\), \(\sigma_{\text{center}}\), \(\gamma\), and \(\sigma_x\)), and obtained \(\eta_0 = 0.13\) (see Main Text Section 3.1).

4 EFFECTIVE NUMBER OF SEQUENCES

The effective number of sequences is a measure of the local coverage or “richness”, or “thickness” of a multiple sequence alignment at a certain position (see the HH-suite user guide). It is calculated on the subalignment \(M_i\), formed by all sequences that have no gap in column \(i\) and by all columns that have at most 10% terminal gaps in these sequences. A terminal gap is a gap that lies either to the left or to the right of the entire sequence. For each column \(j\) of \(M_i\), we calculate amino acid frequencies \(p(i, a)\), using the Hennikoff sequence weighing scheme (Henikoff and Henikoff, 1994). Then the number of effective sequences is the exponential of the average entropy over the frequency distributions \(p(i, a)\) over all \(L_i\) columns of the subalignment \(M_i\):
\[
N_{e}(i) = \exp \left( -\frac{1}{L_i} \sum_{j \in M_i} \sum_a p(i, a) \log p(i, a) \right).
\]
5 SUPPLEMENTAL FIGURES

Supplemental Figure S1: ROC plot showing the number of true positives (same fold) versus false positives (different fold) weighted by one divided by the size of the query’s family. Dashed lines indicate the false discovery rate (FDR) at 1%, 10%, and 20%. The differences between the three methods are smaller when TPs and FPs are weighted by the reciprocal size of the superfamily or the reciprocal size of the fold (see Main Text Section 3.2).

Supplemental Figure S2: Mean ROC5 values measuring the homology detection sensitivity on four different test sets obtained by filtering the SCOP database to get a maximum pairwise sequence identity of 20%, 40%, 60%, and 80%. These results show that the improvements stay similar in absolute terms even for test sets containing many nearly trivial sequence pairs.

Supplemental Figure S3: E-value reliability estimated by comparing the reported E-value to the actual E-value. The actual E-value is derived from the number of FP with an E-value below the value on the x-axis. Using $\delta = -0.005$ bits makes CS-BLASTdis E-values as reliable as BLAST E-values.

Supplemental Figure S4: Reliability of reported E-values. The relationship between reported E-value and actual E-value is shown. The actual E-value is estimated from the number of FPs with an E-value below the value on the x-axis. Lowering the score offset renders the reported E-value more conservative. $\delta = -0.005$ bits is the default parameter in CS-BLAST.

REFERENCES