Sequence analysis

Calibrating E-values for hidden Markov models using reverse-sequence null models

Kevin Karplus1,*, Rachel Karchin2, George Shackelford1 and Richard Hughey1

1Department of Biomolecular Engineering, University of California, Santa Cruz, CA 95064, USA and
2Department of Biopharmaceutical Sciences, University of California, San Francisco, CA, USA

Received on February 11, 2004; revised on August 10, 2005; accepted on August 12, 2005
Advance Access publication August 25, 2005

ABSTRACT

Motivation: Hidden Markov models (HMMs) calculate the probability that a sequence was generated by a given model. Log-odds scoring provides a context for evaluating this probability, by considering it in relation to a null hypothesis. We have found that using a reverse-sequence null model effectively removes biases owing to sequence length and composition and reduces the number of false positives in a database search.

Any scoring system is an arbitrary measure of the quality of database matches. Significance estimates of scores are essential, because they eliminate model- and method-dependent scaling factors, and because they quantify the importance of each match. Accurate computation of the significance of reverse-sequence null model scores presents a problem, because the scores do not fit the extreme-value (Gumbel) distribution commonly used to estimate HMM scores’ significance.

Results: To get a better estimate of the significance of reverse-sequence null model scores, we derive a theoretical distribution based on the assumption of a Gumbel distribution for raw HMM scores and compare estimates based on this and other distribution families. We derive estimation methods for the parameters of the distributions based on maximum likelihood and on moment matching (least-squares fit for Student’s t-distribution).

We evaluate the modeled distributions of scores, based on how well they fit the tail of the observed distribution for data not used in the fitting and on the effects of the improved E-values on our HMM-based fold-recognition methods.

The theoretical distribution provides some improvement in fitting the tail and in providing fewer false positives in the fold-recognition test. An ad hoc distribution based on assuming a stretched exponential tail does an even better job. The use of Student’s t to model the distribution fits well in the middle of the distribution, but provides too heavy a tail. The moment-matching methods fit the tails better than maximum-likelihood methods.

Availability: Information on obtaining the SAM program suite (free for academic use), as well as a server interface, is available at http://www.soe.ucsc.edu/research/compbio/sam.html and the open-source random sequence generator with varying compositional biases is available at http://www.soe.ucsc.edu/research/compbio/gen_sequence

Contact: karplus@soe.ucsc.edu

*To whom correspondence should be addressed.

1 BACKGROUND

1.1 Profile hidden Markov models

Profile hidden Markov models (HMMs) (Haussler et al., 1993; Krogh et al., 1994) or generalized profiles (Bucher and Bairoch, 1994) have been demonstrated to be very effective in detecting conserved patterns in multiple sequences (Baldi et al., 1994; Eddy et al., 1995; Eddy, 1995; Bucher et al., 1996; Hughey and Krogh, 1996; McClure et al., 1996; Karplus et al., 1997, 1998, 1999, 2001; Grundy et al., 1997; Karchin and Hughey, 1998). The typical profile HMM is a chain of match (square), insert (diamond) and delete (circle) nodes, with all transitions between nodes and all character costs in the insert and match nodes trained to specific probabilities.

In 2000, we extended our SAM software (Hughey and Krogh, 1996 (See Fig. 1), Hughey et al., 1999, http://www.soe.ucsc.edu/research/compbio/sam.html) to handle multi-track profile HMMs, in which each match node contains emission probabilities of predicted secondary structure states, in addition to amino acid emission probabilities. The states can be defined simply as helix, sheet and coil, or by a more detailed system, such as DSSP (Kabsch and Sander, 1983), STRIDE (Frishman and Argos, 1995) or PROTEIN BLOCKS (de Brevem et al., 2000).

When an HMM is trained on sequences that are members of a protein family, the resulting HMM can identify the positions of amino acids that describe conserved primary structure of the family. The two-track HMM also identifies the family’s conserved secondary structure. This HMM can then be used to discriminate between family and non-family members in a search of a sequence database, by calculating the probability that each sequence was generated by the HMM. A multiple alignment of sequences to the HMM will reveal the regions in the primary and the secondary structure that are conserved and that are characteristic of the family. All experiments described in this paper use the SAM-T2K iterated seed alignment algorithm (Karplus et al., 2001) to train the amino acid emission probabilities and transition probabilities of the HMMs. The secondary structure probabilities are estimated by using neural networks (Karplus et al., 2001).

Although multi-track HMMs can substantially improve fold-recognition, this improvement is not the focus of this paper. A detailed description of multi-track HMMs, fold-recognition, and the effects of choosing a particular description of secondary structure states can be found in our recent papers (Karchin et al., 2003, 2004).
1.2 Reverse-sequence null models

Log-odds scoring is a means of evaluating the probability that a sequence was generated by a particular HMM by comparing it with a null hypothesis, usually a simpler statistical model intended to represent the universe of sequences as a whole, rather than the group of interest (Altschul et al., 1991; Barrett et al., 2001). The only change from the profile HMMs used previously with SAM is the addition of predicted emission probabilities for secondary structure to the match states—the insertion states get background probabilities for secondary structure and the transition probabilities are identical to the amino-acid-only HMMs. The HMMs used in the paper have as many match states as there are alignment columns in the multiple alignment used as a seed.

Fig. 1. This picture shows how the two-track HMM is organized. The ‘AA’ and ‘2ry’ labels in the boxes refer to emission-probability tables for amino acids and secondary structure labels, respectively (Karplus et al., 2001). The only change from the profile HMMs used previously with SAM is the addition of predicted emission probabilities for secondary structure to the match states—the insertion states get background probabilities for secondary structure and the transition probabilities are identical to the amino-acid-only HMMs. The HMMs used in the paper have as many match states as there are alignment columns in the multiple alignment used as a seed.

1.2 Reverse-sequence null models

Log-odds scoring is a means of evaluating the probability that a sequence was generated by a particular HMM by comparing it with a null hypothesis, usually a simpler statistical model intended to represent the universe of sequences as a whole, rather than the group of interest (Altschul et al., 1991; Barrett et al., 2001). The score is computed as follows:

\[
S = \log \frac{P(X|M)}{P(X|N)},
\]

where \(X\) denotes the sequence being scored, \(M\) is the HMM and \(N\) is a null model. One key to improved performance of the SAM HMM method is a new score-normalization technique that compares the raw score with the score with a reversed model rather than with a simple null model with a single amino acid distribution (Karplus et al., 1998, 1999).

Before we found the reverse-sequence null model, our best choice for remote homolog detection was a geometric null model (Barrett et al., 1997). (The emission tables in the match states of the geometric null model are the geometric mean of the emission tables in the match states of the HMM, renormalized to sum to one—this can be thought of as the average of log-probability scores.) When developing our fold-recognition library, we found that many protein families’ multiple alignments show columns of strict conservation of what are usually the more rarely seen residues (e.g. cysteine). When scoring databases with an HMM built for these families, using the geometric null model or other simple null models, sequences that are compositionally biased toward these residues tend to receive inflated scores and become false positives (Karplus et al., 1998, 1999).

To address this problem, we decided to look at the difference of the log-probability of the sequence and the log-probability of the sequence with a reversed HMM (equivalently, the score of the reversed sequence with the HMM). Since the reversed sequence has the same length and composition as the sequence, these two sources of error are effectively eliminated (Karplus et al., 1998, 1999).

This idea was introduced in the mid-1980s by Taylor (1986), who developed a consensus template representation of conserved patterns in a protein multiple alignment. He found that randomly generated sequences of a desired amino acid composition lacked important features common in proteins, such as the hydrophobicity patterns associated with secondary structure, and made poor decoys for testing whether the templates could discriminate between significant and spurious matches to the conserved pattern. Reversed sequences were effective decoys in these tests, because they retained such features, specifically those that lacked directionality and, thus, remained intact upon sequence reversal (Taylor, 1986).

We also have found that the reversed sequence can be a much more realistic decoy for HMM scoring than our earlier null models. Not only does the reverse null model scoring method correct for length and composition biases, but also it cancels some other subtler effects—for example the periodic hydrophobicity patterns of amphipathic helices or beta strands also appear in the reversed sequence, as does the lower frequency surface-core hydrophobicity pattern.

Reverse-sequence null models are not restricted to use on HMMs, but can be used with any scoring systems for sequences, including BLAST (Altschul et al., 1990), Smith–Waterman alignment (Smith and Waterman, 1981) and profile alignment. Altschul’s group tested reverse-sequence null models with PSI-BLAST (Altschul et al., 1997) and found that the null models were effective for removing false positives (Söhnke et al., 2001). They found their own compositional correction to be more effective for PSI-BLAST than reverse-sequence null models.

2 TESTING REVERSE-SEQUENCE NULL MODELS

Before making a lot of effort trying to calibrate HMMs using reverse-sequence null models, we checked to see whether the reverse-sequence log-odds scores were better at doing fold-recognition than log-odds scores from simpler null models.

Our initial experiments with reverse-sequence null model scoring used SAM-T2K amino-acid-only HMMs.

We compared reverse-sequence null model scoring and geometric null model scoring with these HMMs, on a difficult fold-recognition benchmark set of 1298 chains, sharing very low sequence identity (cut-off of 20%). We created this benchmark set by modifying one of Dunbrack’s culled PDB sets (resolution cut-off of 3.0 \(\AA\) and \(R\)-factor cut-off of 1.0) (Dunbrack, 2001, http://dunbrack.fccc.edu/PISCES.php), using only chains containing SCOP version 1.55 domains (Murzin et al., 1995, http://scop.mrc-lmb.cam.ac.uk/scop) and eliminating fragments of <20 residues. We excluded from our benchmark classes e (multi-domain), i (low resolution), j (peptide) and k (designed proteins). We also excluded some fold classes that are labeled as not representative true folds: a.137 (non-globular all-alpha subunits of globular proteins), d.184 (non-globular alpha–beta subunits of globular proteins) and f.2.1 (membrane all-alpha). Unlike some other fold-recognition tests, we did not collapse the multi-bladed beta propellers into a single fold, nor did we collapse the various
Rossman folds into a single fold. The appropriate rules for collapsing SCOP folds vary from release to release, so we chose the simpler approach of considering all SCOP folds distinct.

All our fold-recognition experiments used the following protocol. A target HMM was built for each sequence in the benchmark set and used to score all the sequences in the set. We then computed the number of true positives and false positives at each score threshold, using the SCOP definition of fold to identify correct pairings.

Figure 2 shows results when one- and two-track SAM-T2K HMMs are tested, with just amino acids and with STRIDE or PROTEIN BLOCKS secondary structure alphabets on the second track (DSSP on the second track behaves almost identically to STRIDE). We see that reverse-sequence null model scoring improves the specificity of all HMMs tested, except for the two-track PROTEIN BLOCKS HMMs, whose log-odds scores are artificially inflated by frequently occurring directed motifs in the PROTEIN BLOCKS alphabet.

An extreme example of this problem occurred when we built a two-track amino acid/PROTEIN BLOCKS HMM for the PDB chain 1ezIA, a three-layer a/b/a structure of eight helices and a mixed beta sheet of seven strands, with an antiparallel seventh strand (Murzin et al., 1995). The reversed version of this PROTEIN BLOCKS sequence was so unlikely, given the log-probability computations made by the HMM that only 76 costs (out of a database of 5271) were >0, instead of approximately half of the database costs. The false positive rate for this HMM was very large: 1201 chains sharing no common domains with 1ezIA (92.5% of the test set) received $E$-values < 0.01 (a common threshold for statistical significance).

For the alphabets other than PROTEIN BLOCKS, the reverse-sequence null model does appear to do what we want, removing misleading signals from the scoring. Therefore, calibrating statistical significance for reverse-sequence null models is a useful exercise.

### 3 Estimating $E$-Values

The statistical significance of a particular score can be summarized by an $E$-value, an estimate of the number of sequences that would get this score (or better) by chance. To accurately match the number of expected random hits, we need a statistical model that fits the distribution of random scores (scores that would be given to a large, random sample of protein sequences).

We can estimate $E$-values for reverse-sequence null model scores, if we make some simplifying assumptions:

**Reversibility assumption.** We assume that reversed sequences look like protein sequences, i.e. that they can be thought of as coming from the same underlying distribution. This assumption is clearly violated if we have directed motifs, and it turns out to be the most problematic of our assumptions (see Section 10).

**Extreme value assumption.** Unnormalized costs $-\log(P(M|X))$ are distributed with an extreme-value (Gumbel) distribution. The probability that cost $c$ is better than some threshold $a$ is approximately$^7$

$$P(c < a) \approx \frac{1}{\beta} \exp \left( -\frac{c - \mu}{\beta} \right).$$  

The density corresponding to this distribution is

$$g_{\mu, \lambda}(c) = \frac{1}{\beta} \exp \left( -\frac{c - \mu}{\beta} \right).$$  

Although several scoring systems have been shown to be well approximated by extreme-value distributions, there is still a question about how well we can fit the sum-of-all-paths local alignment scores used in SAM HMMs with such a distribution. We will return to this question in Section 5.

**Independence assumption.** The costs for sequences with the model and reverse model are independent. This assumption is clearly wrong, since the whole point of the reverse-model is to cancel some of the misleading signals. The ways in which the assumption is violated (length, composition, helixity signal, spacing of repeated residues, etc.) all tend to make the normal and reverse costs more correlated and the difference signal closer to zero. Since we are mainly interested in the extreme negative tail of the costs, the violations should make the significance estimates become rather conservative.

With these assumptions, we are ready to derive the distribution of the raw costs normalized by the reverse-sequence null model. Let $c_M$ be the cost of a sequence given model $M$, $c_M'$ be the cost of the reversed sequence given $M$, and $r_M$ be the normalized cost $c_M - c'_M$. Then the distribution of the normalized scores is

$$P(r_M < a) = \int_{-\infty}^{\infty} P(c_M = x) P(c_M' > x - a) \, dx$$

$$= \int_{-\infty}^{\infty} k \lambda \exp \left( -\frac{a}{\lambda} \right) \frac{1}{\beta} \exp \left( -\frac{c - \mu}{\beta} \right) \, dx$$

$$= \int_{-\infty}^{\infty} k \lambda \exp \left( -\frac{a}{\lambda} \right) \frac{1}{\beta} \exp \left( -\frac{a}{\beta} \right) \, dx.$$  

If we introduce a temporary variable to simplify the formulas, $K_a = k(1 + \exp(-\lambda a))$, then

$$P(r_M < a) = \int_{-\infty}^{\infty} (1 + e^{-\lambda a})^{-1} K_a \lambda e^{\lambda x} \exp \left( -\frac{a}{\beta} \right) \, dx$$

$$= (1 + e^{-\lambda a})^{-1} \int_{-\infty}^{\infty} g_{\mu, \lambda}(c) \, dx$$

$$= (1 + e^{-\lambda a})^{-1}.$$  

and we have a simple sigmoidal function with only one parameter as the distribution function.

Note that for $a \ll 0$ (that is on the tail for good matches),

$$P(c_M < a) = 1 - e^{-\lambda a} \approx \mu e^{\lambda a},$$

and

$$P(r_M < a) = (1 + e^{-\lambda a})^{-1} \approx e^{\lambda a}.$$  

That means that the tail of the normalized costs is approximately of the same shape as on the original distribution—all that happens is that the costs are centered on 0, eliminating the offset by $\ln k$.

### 4 Calibrating by Setting $\lambda$

Our initial use of the $E$-value computations made the assumption that the scale factor $\lambda$ is 1, since the log-probability computations in the SAM package are all performed using natural logarithms. This worked reasonably well in our fold-recognition tests (Section 7), but...
we thought that we could do better if we calibrated our HMMs by optimizing \( \lambda \).

### 4.1 Generating random sequences

When fitting a distribution by optimizing a parameter, one needs a large sample of scores to work from. There have been two popular approaches for generating this sample: using a simple null model to generate random sequences and using a database of decoy sequences that should not match the model. The decoy database is often created by shuffling the residues of real sequences, or even by reversing sequences.

We initially decided to use random sequences from a null model for calibration, since

- no database needs to be distributed with the program, installed or accessed;
- large numbers of sequences can be generated to explore the shape of the tails; and
- unusual sequence distributions are easily generated, so that the effects of composition bias, length or other properties can be explored.

We created a sequence generator (gen_sequence) that for each sequence generates a length \( L \) from a log–normal distribution, and a probability distribution \( P \) over the alphabet from a Dirichlet mixture prior, then makes \( L \) independent draws from \( P \). This sequence generator is available as open-source code (Karplus, 2000, http://www.soe.ucsc.edu/research/compbio/gen_sequence/).

### 4.2 Fitting \( \lambda \) by moment matching

We consider two ways by which we can try to fit the distribution of a set of \( N \) reverse-sequence-null costs: least-squares fit and maximum-likelihood estimation.

For least-squares fit, the most direct approach is to sort the costs \( \{ c_0 \leq \cdots \leq c_{N-1} \} \) and to do a least-squares fit of the costs to the inverse of the distribution function. The least-square fitting can be simplified to matching the second moment of the distribution:

\[
\lambda \approx \pi \left( \frac{N}{3} \sum_{i=0}^{N-1} c_i^2 \right) \quad \text{(8)}
\]

(see Appendix section for derivation). This is the method used in the SAM package to calculate the \( E \)-value used in the one-parameter sigmoidal distribution.

### 4.3 Fitting \( \lambda \) by maximum-likelihood estimation

In maximum-likelihood estimation, \( \lambda \) is fitted by maximizing the likelihood function

\[
L_\theta = \prod_{i=1}^{n} p(x_i|\theta), \quad \text{(9)}
\]

which is equivalent to maximizing the log-likelihood function, which is easier to compute (see Equation 15 in the Appendix section):

\[
\ln \left( L_\theta \right) = \sum_{i=1}^{n} \ln p(x_i|\theta) \quad \text{(10)}
\]

\[
= \sum_{i=1}^{n} \ln \left( \lambda \left(1 + e^{-\lambda x_i}\right)^{-1} \right) \left(1 + e^{\lambda x_i}\right)^{-1} \right) \\
= n \ln \lambda + \lambda \sum_{i=1}^{n} x_i - 2 \sum_{i=1}^{n} \ln \left(e^{\lambda x_i} + 1\right). \quad \text{(11)}
\]

We find the optimal \( \lambda \) by using the golden section method (Vetterling, 1988). A search is performed with increasing intervals until a local maximum is bracketed, then with decreasing intervals to pinpoint the maximum.

### 5 ALTERNATIVES TO THE THEORETICAL DISTRIBUTION

Yu and Hwa (2001) have raised the concern that, in their experience, local alignment of HMMs does not result in a Gumbel distribution, but in a stretched exponential tail (Yu and Hwa, 2001). The observed distributions of reverse null model costs have fatter tails than what is expected from our theoretically derived sigmoidal distributions, confirming Yu and Hwa’s observation. For this reason, we became interested in fitting the parameters of other distribution families.

We tried two families with fatter tails: an ad hoc sigmoidal family that added one parameter to our theoretical distribution and Student’s \( t \)-distribution.

#### 5.1 Ad hoc two-parameter sigmoidal distribution

We generalized the theoretically derived sigmoidal distribution to get a family of symmetric distributions that could have stretched exponential tails:

\[
P(x_M < a) = \left( 1 + e^{-(\lambda a)^2} \right)^{-1}, \quad \text{(12)}
\]

where \( x^2 \) is interpreted as \(-(-x)^2\) for \( x < 0 \).

We can do moment matching on this two-parameter family using both the second and fourth moments, as explained in Section 12.1.

#### 5.2 Student’s \( t \)-distribution

The Student’s \( t \)-distribution family \( \text{t}(\mu, \sigma^2) \) is a generalization of the standard normal distribution and is often a good model for heavy-tailed data. Tail-weight is controlled by the degrees-of-freedom parameter \( \nu \), which is small for fat tails. The Student’s \( t \) density function is

\[
p(x) = \frac{\Gamma((\nu+1)/2)}{\sqrt{\nu\pi} \Gamma(\nu/2)} \left( 1 + \frac{(x-\mu)^2}{\nu \sigma^2} \right)^{-(\nu+1)/2}. \quad \text{(13)}
\]

Other than knowing that the mean \( \mu = 0 \) and \( \nu \) is small, we do not have much prior information about what values of \( \nu \) and \( \sigma \) would best fit the distribution of costs. Following the suggestion of the statistician David Draper (personal communication), we decided to pick these two parameters by maximum-likelihood estimation, rather than choosing a more expensive Bayesian maximum a posteriori approach (details in Section 12.2).

We also tried using a least-squares fit to the observed cumulative distribution, but we cannot use moment matching for Student’s \( t \), as the values of \( \nu \) that are appropriate result in distributions with infinite fourth moments.

After fitting \( \nu \) and \( \sigma^2 \) with a training set, we need to compute \( E \)-values for the reverse null model costs. Student’s \( t \) does not have a closed-form distribution, so the cumulative distribution function

6 PROBLEMS WITH CALIBRATION AND SECONDARY-STRUCTURE SEQUENCES

When we attempted to use random sequences to calibrate our HMMs, we got good results with single track (amino-acid-only) HMMs, but truly terrible results when calibrating two-track HMMs—worse than using uncalibrated HMMs (i.e. assuming \( \tau = 1 \) and \( \lambda = 1 \)) (data not shown). The problem is that the secondary-structure sequences are not well modeled as independent draws from identical distributions—there is a very high correlation from one position to the next. In our test set, helices (runs of the letter H) average 12.5 residues and beta strands (runs of the letter E) average 5.4 residues. Many real secondary structure sequences fit our HMMs much better than random sequences do, making significance estimation from random sequences useless.

Luckily, we have a simple workaround—we can estimate the second and fourth moments from the database that we are scoring (or any other database we care to use). Normally, such estimates are very difficult to make, as the true positives produce a ‘lump’ on one tail that throws the estimates way off. The symmetry about 0 of our distributions lets us ignore the tail that is contaminated by true positives, and estimate the moments from just the other half of the scores.

Another approach is to try to estimate where the lump of true positives ends, and fit a truncated distribution to the data past that point. This approach has been advocated by Bailey and Gribskov (2002), but we did not try to use it, as the reversibility assumption allows the luxury of a symmetric distribution. By throwing out about half the data, we are almost certain not to be contaminated by true positives, even for models that provide relatively poor separation between the true positives and the true negatives. In those cases where the reversibility assumption is not reasonable, Bailey and Gribskov’s approach may be more useful.

When we calibrate our HMMs (single-track or 2-track) using the database scores, we get very good behavior, as shown in Section 7.

7 CONFIRMING CALIBRATION

The first test is to evaluate the quality of E-values computed with the distributions discussed in Sections 3, 4 and 5. In the optimal fit, we should see sorted E-values equal to their ranks.

For this and later evaluations we use a set of 6545 proteins of known structure—a snapshot of the template library used in the SAM-T2K protein-structure-prediction web server, from August 2003. This template library provides thorough coverage of the available protein structures, but has over-representation of many families. We used this set for calibration, despite the over-representation, since it is the set that we usually score with our HMMs, and so calibration on this set is of particular value to us. We have not yet investigated whether a representative subset would provide better calibration of the HMMs.

From the template library we selected a subset of 1000 proteins at random, and used the corresponding HMMs built by the SAM-T02 method for our calibration tests (Karpplus et al., 2003). Each HMM was calibrated by scoring 10,000 randomly generated sequences and fitting the parameters of the one- and two-parameter sigmoidal, using either moment-matching or maximum-likelihood, and the two-parameter Student’s t-distribution families, using either least-squares fit or maximum-likelihood. E-values were then computed for an independent test set of 10,000 random sequences, using the six fitted distributions and an unfitted sigmoidal (\( \lambda = 1 \)). The average E-value for each rank is plotted in Figure 3. To make the graph more readable, the average E-value is divided by the rank, so that an ideal fit would be a constant 1, and we use a log scale for the rank to emphasize the behavior on the tail that we are interested in. The two-parameter sigmoidal model fit with moment matching has the best fit.

A database calibration of the same 1000 HMMs was performed. Because we need many samples to estimate the parameters of the distributions, we used two-thirds of the sequences (4363) to set the parameters and the remaining one-third (2182) for testing. To avoid contamination by the ‘lumpy’ tail of true positives described in Section 6, all our calibration on testing and real data use only the positive costs. This means that the E-values are not for the best fits to the HMMs, but for the worst fits. The results (not shown) are very similar to those for calibration with random sequences.

For moment-matching, the moments are easily estimated from only the positive costs, but for the Student’s t parameter estimation, we had to symmetrize the data by creating artificial observations that were the negatives of the positive tail. We show only this tail of the distribution, estimating the E-value using twice the number of positive costs.

Of the six distributions examined, the two-parameter sigmoidal family fitted by using moment-matching produces E-values closest to the desired fit followed by the two-parameter sigmoidal family fitted by using maximum-likelihood estimate and Student’s t fitted by using least-squares fit. Of the two methods used to fit the two parameters, moment-matching does better than the maximum-likelihood estimate.

The ‘fat tail’ of Student’s t causes us to underestimate significance at the extreme values, where we are most interested in getting an accurate estimate. Nearer the mean, where small inaccuracies in significance are irrelevant, it is consistently better than the two-parameter sigmoidal.

We further examine the three best distributions by plotting the standard deviation of the log of the ratio of E-value and rank (Fig. 4). The two-parameter sigmoidal family fitted by moment-matching is distinguished by its tighter standard deviation, as well as the closer fit to the desired ratio of 1. This distribution and fitting method are currently implemented in SAM version 3.3 and the later versions.

8 FOLD-RECOGNITION EXPERIMENTS

In a test of fold-recognition performance of reverse-sequence null model scores and E-values, we used both single-track (amino-acid-only) HMMs and two-track HMMs (amino acid and predicted STRIDE class), uncalibrated, calibrated with random sequences, and calibrated with a database of sequences from the PDB. The two-parameter sigmoidal family was used for calibration.

We expected that fold-recognition should remain unchanged or improve slightly with calibration of the E-values, as calibration makes scores from the different HMMs have similar interpretations.
Curves in descending order for false positives per query worse for two-track HMMs (Fig. 5), most probably because the calibration with random sequences actually made performance shown, but fold-recognition performance was almost unchanged.

The tests were performed using amino-acid-only SAM-T2K HMMs and two-track SAM-T2K HMMs with STRIDE or PROTEIN BLOCKS secondary-structure alphabets on the second track. Reverse-sequence null model scoring improves the performance of all HMMs tested, except for the two-track PROTEIN BLOCKS HMMs, whose reverse-sequence null model scores are artificially inflated by frequently occurring directed motifs in the PROTEIN BLOCKS alphabet.

These tests use just the log-odds scores, not the E-values. The key lists the curves in descending order for false positives per query ~0.1.

Fig. 2. Comparison of reverse-sequence null model scoring and geometric null model scoring on a difficult fold-recognition benchmark set of 1298 non-homologous chains. True positives are those pairs with log-odds scores better than some threshold that contain domains in the same fold family in SCOP version 1.55. False positives are those pairs that are in different fold families.

The fitting of the parameters was done using a separate set of 10,000 random sequences. The distributions shown are an unfitted sigmoidal distribution, one-parameter sigmoidal distributions fitted by moment-matching and maximum-likelihood, two-parameter sigmoidal distributions fitted by moment-matching and maximum-likelihood, and two-parameter Student’s t-distributions fitted by least-squares fit and maximum-likelihood estimation. The fitting was performed on 4363 sequences that were not part of the test set.

Fig. 3. Averaged and normalized E-value versus rank of reverse-sequence null model costs for 1000 amino-acid-only HMMs on a test set of 10,000 random sequences. The distributions shown are an unfitted sigmoidal distribution, one-parameter sigmoidal distributions fitted by moment-matching and maximum-likelihood, two-parameter sigmoidal distributions fitted by moment-matching and maximum-likelihood, and two-parameter Student’s t-distributions fitted by least-squares fit and maximum-likelihood estimation. The fitting of the parameters was done using a separate set of 10,000 random sequences. At rank 30 the curves are in the order shown in the legend. The constant 1 line represents the desired behavior from perfectly calibrated E-values.

On calibration, the reported E-values were more accurate (data not shown), but fold-recognition performance was almost unchanged. Calibration with random sequences actually made performance worse for two-track HMMs (Fig. 5), most probably because the secondary-structure sequences are not well modeled as independent draws from a composition.

Fig. 4. Averaged and normalized E-value versus rank, showing plus and minus the standard deviation for the log of the E-value/rank of the set of 1000 proteins. E-values shown are the average and standard deviation of log of E-values/rank computed using the two-parameter sigmoidal distribution fitted by moment-matching and by maximum-likelihood estimation, and of Student’s t fitted with least-squares fit to the cumulative distribution function. The moment-matching fits are consistently tighter than the maximum-likelihood fits and tighter than Student’s t. The fitting was performed on 4363 sequences that were not part of the test set.

Fig. 5. Fold-recognition results on 1298 chains. True positives are those pairs with a lower E-value than some threshold that contain domains in the same fold family in SCOP version 1.55. False positives are those pairs with lower E-values than the threshold that are in different fold families. The two-track HMMs used predictions of the STRIDE alphabet as the second track.

9 DISTRIBUTION OF λ AND τ

The question arose whether the lack of improvement in fold-recognition came from having similar λ and τ values for all the HMMs (so that calibration provided a uniform transform of all results) or whether calibration was improving on some HMMs and doing worse on others. The answer appears to be a little of both.

We calibrated the approximately 8900 amino-acid-only HMMs of the SAM-T04 template library using moment-matching and found that τ had a relatively small range, while λ varied considerably. The
distributions seen for $\lambda$ and $\tau$ are dependent on the way in which the HMMs were built, in particular, on the sequence weighting used, as it determines how general or specific the match states of the HMM are, but all the SAM-T04 HMMs were built with the same parameters and the same iterative search procedure to build the multiple alignments, just different seed sequences were used.

The $\lambda$ values have a very skewed distribution, with a peak around 1.65, mean of 2.84 and standard deviation of 2.61. The $\tau$ values have a roughly normal distribution, with a mean of 0.74 and standard deviation of 0.069. The two values are coupled, as shown in Figure 6, with $\tau \approx 0.86 - 0.14 \ln \lambda$. Since many of the HMMs had $\lambda \approx 1.65$ and $\tau \approx 0.74$, the calibration of these HMMs was nearly a uniform transform. The outliers did not show a consistent improvement as a result of calibration.

The question was also raised whether the $E$-values had any dependence on the length of the sequence or HMM. We have not observed any dependence of $E$-value on the length of the sequence being scored, nor do we expect any strong effect. The length dependence one usually sees for Gumbel distributions of sequence match scores is in the $\lambda$ parameter, which is eliminated by reverse-sequence nulls.

There is a dependence on model length for $\lambda$ and $\tau$ with the larger $\lambda$ values coming from shorter HMMs. We can get a reasonable estimate of $\lambda$ with

$$\ln \lambda \approx 7.35 - 2.11 \ln n + 0.16(\ln n)^2, \quad (14)$$

where $n$ is the number of match states in the HMM (data not shown).

10 ANALYSIS OF THE REVERSIBILITY ASSUMPTION

Our most serious problem with the reverse-sequence null model concerns the assumption that the reverse sequence is representative of the universe of sequences as a whole. More specifically, we are assuming that sequences and reversed sequences come from the same underlying distribution—that they are equally protein-like. We want the differences between the sequence and the reversed sequence to be characteristic of the particular protein family we are modeling, and not the difference between a protein and a non-protein.

This assumption is wrong when the sequence distribution contains directed motifs, frequently occurring combinations of letters that rarely appear in reversed form. Amino-acid directed motifs are fairly rare, but we have encountered problems when applying the reverse-sequence null model to two-track HMMs, because some secondary-structure alphabets contain many directed motifs. In this situation, the reverse sequence is highly uncharacteristic and tends to have low probability. Rather than correcting for bias, the normalization with the null then artificially inflates the log-odds scores [Equation (1)], increasing the number of false positives.

Secondary structure alphabets such as STRIDE and DSSP contain few directed motifs, and we observe a reduction in false positives when reverse-sequence null model scoring is used, both with SAM-T2K amino-acid-only HMMs and with two-track SAM-T2K HMMs having a STRIDE or DSSP secondary track (Fig. 2—DSSP not shown, but almost identical to STRIDE).

In contrast, the PROTEIN BLOCKS alphabet of secondary structure (de Brevn et al., 2000) is highly directed. This alphabet provides a fine-grained description of the protein backbone, consisting of 16 canonical conformations, formed by backbone segments of 5 consecutive residues. Ten out of the sixteen PROTEIN BLOCKS describe motifs that specifically occur at the N-terminal or C-terminal ends of helices and strands. When PROTEIN BLOCKS is used as a secondary HMM track, reverse-sequence null model scoring results in significantly more false positives than geometric null scoring (Fig. 2).

We can do a crude quantification of the reversibility of an alphabet by building a first-order Markov chain to model a set of sequences, then looking at the difference in encoding cost for forward sequences and reverse sequences using that first-order chain. If the difference is large, then the alphabet is not reversible and our reverse-sequence null models will not do what we want.

In Table 1 we can see that amino acid sequences are fairly well modeled as independent draws from a background distribution—first-order models do not reduce the encoding cost much, and the cost of encoding reversed sequences is only slightly more than the cost of encoding sequences in the forward direction. The secondary structure alphabets DSSP and STRIDE are not well modeled as strings of independent draws from a background (the first-order model cuts the encoding cost in half), but the sequences are still fairly reversible. The PROTEIN BLOCKS alphabet is highly

<table>
<thead>
<tr>
<th>Alphabet</th>
<th>0-order</th>
<th>1st-order</th>
<th>rev-for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amino acid</td>
<td>4.1896</td>
<td>4.1759</td>
<td>0.0153</td>
</tr>
<tr>
<td>STRIDE</td>
<td>2.3330</td>
<td>1.0455</td>
<td>0.0042</td>
</tr>
<tr>
<td>DSSP</td>
<td>2.5494</td>
<td>1.3387</td>
<td>0.0590</td>
</tr>
<tr>
<td>PROTEIN BLOCKS</td>
<td>3.3935</td>
<td>1.4876</td>
<td>3.0551</td>
</tr>
</tbody>
</table>

The final column gives the difference between encoding sequences in the reverse and forward direction, using a first-order Markov model trained on forward sequences. To avoid problems with zero encoding costs, the first-order models were built using the zero-order probabilities as pseudocounts.

In Table 1 we can see that amino acid sequences are fairly well modeled as independent draws from a background distribution—first-order models do not reduce the encoding cost much, and the cost of encoding reversed sequences is only slightly more than the cost of encoding sequences in the forward direction. The secondary structure alphabets DSSP and STRIDE are not well modeled as strings of independent draws from a background (the first-order model cuts the encoding cost in half), but the sequences are still fairly reversible. The PROTEIN BLOCKS alphabet is highly

![Fig. 6. Results of fitting the two-parameter sigmoidal distributions to 8900 amino-acid-only HMMs using moment matching. The calibrations were performed on a database of real sequences, computing the moments using only those sequences whose reversals scored better than the sequences themselves, to avoid distortions owing to true positive matches.](https://academic.oup.com/bioinformatics/article-abstract/21/22/4107/194239)

Calibrations for 8900 HMMs

### Table 1. For the alphabets in the first column, the next two columns give the cost (in bits/character) of encoding a training set of sequences with a zero-order and first-order Markov chain

<table>
<thead>
<tr>
<th>Alphabet</th>
<th>0-order</th>
<th>1st-order</th>
<th>rev-for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amino acid</td>
<td>4.1896</td>
<td>4.1759</td>
<td>0.0153</td>
</tr>
<tr>
<td>STRIDE</td>
<td>2.3330</td>
<td>1.0455</td>
<td>0.0042</td>
</tr>
<tr>
<td>DSSP</td>
<td>2.5494</td>
<td>1.3387</td>
<td>0.0590</td>
</tr>
<tr>
<td>PROTEIN BLOCKS</td>
<td>3.3935</td>
<td>1.4876</td>
<td>3.0551</td>
</tr>
</tbody>
</table>
irreversible, as well as being poorly modeled as independent draws from a background.

The directionality of protein amino acid sequences is quite weak. First-order Markov models do no better than zero-order ones for modeling protein sequences, other than for the high probability of MET as the starting residue. There have been protein structures crystallized for the reversal of a natural protein sequence (only for coiled-coils so far). There may be some long-range asymmetries in protein sequences, but we have not found a model that captures them well. Our best model for amino acid sequences in protein databases is a hierarchical model that models composition by a mixture of Dirichlet distributions and the sequence as independent identically distributed draws from that composition.

11 CONCLUSIONS AND FUTURE WORK

We have found a theoretical justification for using a simple family of distributions for estimating statistical significance of the reverse-sequence null models, but the assumptions of the theory are often violated in practice, requiring ad hoc extensions to the theory. The ad hoc two-parameter family of distributions presented here seems to fit the data reasonably well.

Moment-matching and least-squares fitting of the cumulative distribution did a better job of estimating the parameters than maximum-likelihood, probably because we are primarily interested in matching the tails, while maximum-likelihood matching concentrates on the main mass of the distribution, which is of little interest in this application.

Precisely determining the statistical significance of complex stochastic models such as our multi-track HMMs is difficult. For amino acid HMMs and for HMMs involving reversible alphabets, reverse-sequence null models are effective and the resulting scores can be calibrated fairly accurately using database sequences.

There are several directions for further research.

One is to investigate better null models for secondary structure alphabets. For example, first- and second-order Markov chains clearly model these sequences better than simple draws from a composition. Using such a higher-order model for generating random sequences may allow us to do calibration of our two-track HMMs using random sequences, avoiding problems inherent in calibration from a database of known sequences.

The reverse-sequence null model is unlikely to be useful for HMMs that involve non-reversible alphabets, such as PROTEIN BLOCKS. We will have to find a better approach for the correction of composition, length and helicity anomalies when working with such HMMs.

Our theoretical derivation of sigmoidal distributions for scoring with reverse-sequence null models turns out not to provide a very good fit, probably because our underlying scoring system is not well modeled as a Gumbel distribution. By including some ad hoc parameters to allow fitting a stretched-exponential tail provides much better fitting to the data, but the Student’s t family (also with two parameters) did not fit the data consistently well. It would be useful to have a more principled derivation of distributions that fit the data, since we are using the distributions to extrapolate the tails well past the point where we have useful calibration data.

We plan to continue our search for good HMM calibration and will experiment with other heavy-tailed distributions.

ACKNOWLEDGEMENTS

The authors are grateful to Mark Diekhans for contributing to the software used in our experiments and to David Draper for suggesting Student’s t-distribution. The authors specially thank the anonymous referees who suggested comparison with maximum-likelihood fits and who raised questions about length dependencies. Fixing these omissions delayed publication by several months, but greatly improved the paper. This work was supported in part by the DOE grant DE-FGO395-99ER62849, NSF grants DBI-9808007 and EIA-9905322, NIH grant 1 R01 GM068570-01 and a National Physical Sciences Consortium graduate fellowship.

Conflicts of interest: none declared.

REFERENCES


Dunbrack,R. (2001) Calling the PDB by resolution and sequence identity.


