Calibrating E-values for Hidden Markov Models with Reverse-sequence Null Models

Kevin Karplus[†] Rachel Karchin[‡] George Shackelford[†] Richard Hughey[†]

[†]Department of Biomolecular Engineering, University of California, Santa Cruz

[‡]Department of Biopharmaceutical Sciences, University of California, San Francisco

4 March 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 due 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. Using 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 from http://www.soe.ucsc.edu/research/compbio/sam.html and the open-source random sequence generator with varying compositional biases is available from http://www.soe.ucsc.edu/research/compbio/gen_sequence

Contact: Kevin Karplus
Biomolecular Engineering

University of California
Santa Cruz, CA 95064
karplus@soe.ucsc.edu
phone: 1-831-459-4250
fax: 1-831-459-4829

Keywords: hidden Markov models, HMMs, statistical significance, E-value, Gumbel distribution, reverse-sequence null, stretched exponential, Student's t distribution

1 Background

1.1 Profile Hidden Markov Models

Profile hidden Markov models [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, Grundy et al., 1997, Karplus et al., 1998, Karchin and Hughey, 1998, Karplus et al., 1999, Karplus et al., 2001]. 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.

We have recently extended our SAM software [Hughey and Krogh, 1996, Hughey et al., 1999] 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 Brevern 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 which 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 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 neural networks [Karplus et al., 2001].

Although multi-track HMMs can substantially improve fold-recognition, that 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, Karchin et al., 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 to 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, 1991, Barrett et al., 1997]. The score is computed as

$$S = \log \frac{\mathrm{P}(X|M)}{\mathrm{P}(X|N)} \ , \qquad (1)$$

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 to the score with a reversed model rather than to a simple null model with a single amino-acid distribution [Karplus et al., 1998, Karplus et al., 1999].

Before we found the reverse-sequence null model, our best choice for remote homolog detection was a *geometric null model* [Barrett et al.,

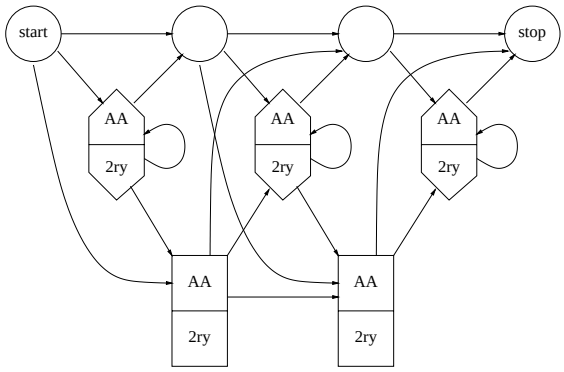


Figure 1: This picture shows how the two-track HMM is organized. The "AA" and "2ry" labels in the boxes refer to emission-probability tables for amino acids and secondary structure labels, respectively [Karplus et al., 2001]. The only change from the profile HMMs used previously with SAM is the addition 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.

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 (cysteine, for example). 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, Karplus et al., 1999].

To address the 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, Karplus et al., 1999].

This idea was introduced in the mid-eighties by Taylor [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 patterns. 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 some other, subtler effects are also canceled—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 hidden Markov models, but can be used with any scoring systems for sequences, including BLAST [Altschul et al., 1990], Smith-Waterman alignment [Smith and Waterman, 1981], profile alignment, and others. Altschul's group tested reverse-sequence null models with PsiBlast [Altschul et al., 1997], and found that the null models were effective for removing false positives [Schäffer et al., 2001]. They found their own compositional correction to be more effective for PsiBlast than reverse-

sequence null models.

2 Testing reverse-sequence null models

Before going to 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. Figure 2(a) compares 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 Å and R-factor cut-off of 1.0) [Dunbrack, 2001], using only chains containing SCOP version 1.55 domains [Murzin et al., 1995] and eliminating fragments of less than 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 representing 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(b), Figure 2(c), and Figure 2(d) show results when two-track SAM-T2K HMMs are tested, with STRIDE, DSSP, and PROTEIN BLOCKS secondary structure alphabets on the second track. 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.

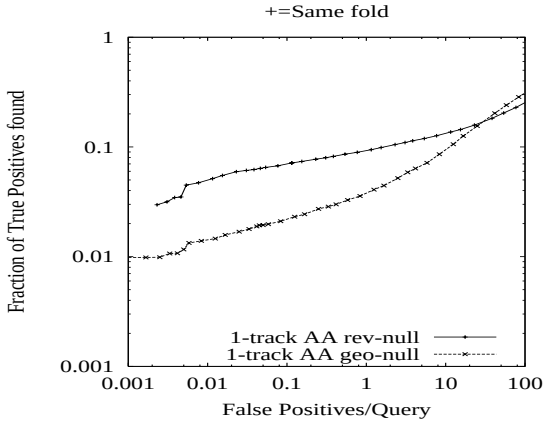
An extreme example of this problem occurred when we built a two-track amino-acid/PROTEIN BLOCKS HMM for PDB chain 1eziA, a 3-layer a/b/a structure of eight helices and a mixed beta sheet of seven strands, with an antiparallel seventh strand [Murzin et al., 1995]. The reversed version of this PROTEIN BLOCKS sequence was so unlikely, given the probability distributions learned by the HMM, that only 76 costs (out of a database of 5271) were greater than zero, 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 less than 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 model is a useful exercise.

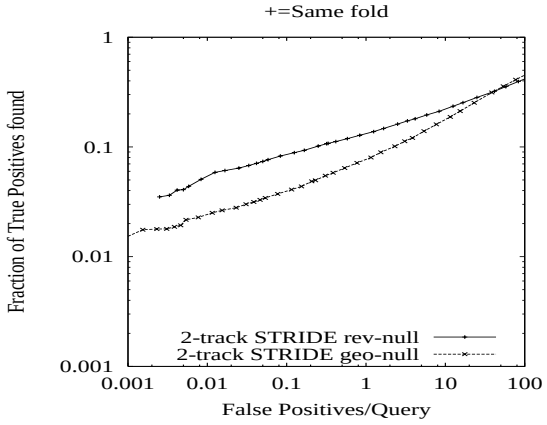
3 Estimating E-values

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).

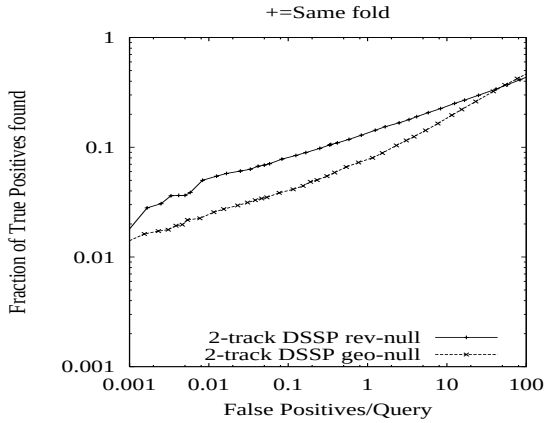
We can estimate E-values for reverse-sequence null model scores, if we make some simplifying



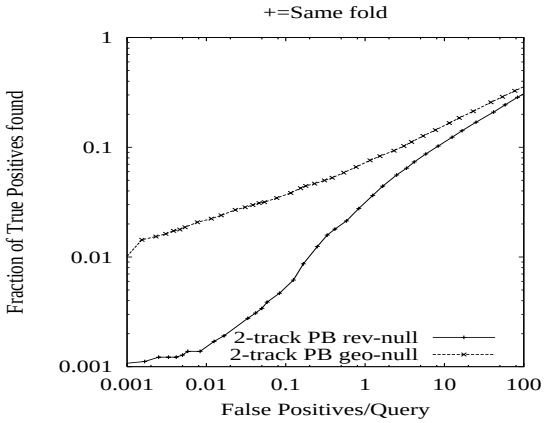
(a) One-track HMM: amino acid only.



(b) Two-track HMM: amino acid and STRIDE.



(c) Two-track HMM: amino acid and DSSP.



(d) Two-track HMM: amino acid and PROTEIN BLOCKS. Note: the reverse-sequence null makes fold-recognition worse!

Figure 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 tests were done using amino-acid-only SAM-T2K HMMs and two-track SAM-T2K HMMs with STRIDE, DSSP, 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 E-values.

assumptions:

Reversibility Assumption We assume that reversed sequences look like protein sequences, that is, 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 9).

Extreme Value Assumption Unnormalized costs ($-\log(\mathrm{P}(X|M))$) are distributed with an extreme-value (Gumbel) distribution. The probability that cost c is better than some threshold a is approximately[1]

$$\mathrm{P}(c < a) \approx G_{k,\lambda}(a) = 1 - \exp(-ke^{\lambda a}) \ . \quad (2)$$

The density corresponding to this distribution is

$$g_{k,\lambda}(a) = k\lambda e^{\lambda a} \exp(-ke^{\lambda a}) \ . \quad (3)$$

Although several scoring systems have been shown to be well approximated by extreme-value distributions, there is still some 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, helicity signal, spacing of repeated residues, ...) 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

$$\begin{aligned}
&\mathrm{P}(r_M < a) = \\
&\quad = \int_{-\infty}^{\infty} \mathrm{P}(c_M = x)\mathrm{P}(c_{M'} > x - a)dx \\
&\quad = \int_{-\infty}^{\infty} k\lambda \exp(-ke^{\lambda x})e^{\lambda x} \exp(-ke^{\lambda(x-a)})dx \\
&\quad = \int_{-\infty}^{\infty} k\lambda e^{\lambda x} \exp(-k(1 + e^{-\lambda a})e^{\lambda x})dx \quad (4)
\end{aligned}$$

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

$$\begin{aligned}
&\mathrm{P}(r_M < a) = \\
&\quad = \int_{-\infty}^{\infty} (1 + e^{-\lambda a})^{-1} K_a \lambda e^{\lambda x} \exp(-K_a e^{\lambda x})dx \\
&\quad = (1 + e^{-\lambda a})^{-1} \int_{-\infty}^{\infty} g_{K_a,\lambda}(x)dx \\
&\quad = (1 + e^{-\lambda a})^{-1} \quad (5)
\end{aligned}$$

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),

$$\mathrm{P}(c_M < a) = 1 - e^{-k\lambda a} \approx ke^{\lambda a} \ , \quad (6)$$

and

$$\mathrm{P}(r_M < a) = (1 + e^{-\lambda a})^{-1} \approx e^{\lambda a} \ . \quad (7)$$

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

[1] SAM reports costs, in which the most negative values are the good scores—negate appropriately for positive-good scoring systems.

4 Calibrating by setting λ

Our initial use of the E-value computations made the assumption that the scale factor λ is 1, since the log-probability computations in the SAM package are all done 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 λ.

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,
- 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].

4.2 Fitting λ by moment matching

We consider two ways 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 $\{c_0 \le \cdots \le c_{N-1}\}$ 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:

$$\lambda \approx \pi \sqrt{N / \left(3 \sum_{i=0}^{N-1} c_i^2 \right)} \tag{8}$$

(see Appendix 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 λ by maximum-likelihood estimation

In maximum-likelihood estimation, λ is fitted by maximizing the likelihood function

$$L_n(\theta) = \prod_{i=1}^{n} p(x_i \mid \theta) \tag{9}$$

which is equivalent to maximizing the log likelihood function, which is easier to compute (see Equation 35 in the Appendix):

$$\ln\left(L_n\left(\theta\right)\right) = \sum_{i=1}^{n} \ln p(x_i \mid \theta) \tag{10}$$

$$= \sum_{i=1}^{n} \ln(\lambda(1+e^{-\lambda x_i})^{-1}(1+e^{\lambda x_i})^{-1})$$

$$= n \ln \lambda + \lambda \sum_{i=1}^{n} x_i - 2 \sum_{i=1}^{n} \ln(e^{\lambda x_i}+1) \tag{11}$$

We find the optimal λ by using the *golden section* method [W.T.Vetterling, 1988]. A search is done with increasing intervals until a local maximum is bracketed, then with decreasing intervals to pinpoint the maximum.

5 Alternatives to the theoretical distribution

Yi-kuo Yu and Terence Hwa 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:

$$\mathrm{P}(r_M < a) = (1 + e^{-(\lambda a)^\tau})^{-1} \quad (12)$$

where x^τ is interpreted as $-(-x)^\tau$ for $x < 0$.

We can do moment-matching on this two-parameter family using both the second and fourth moments.

Using the change of variables $y = -(\lambda a)^\tau$ and defining a function

$$F(\beta) = \int_0^\infty y^\beta \frac{e^y}{(1+e^y)^2} dy \ , \quad (13)$$

we can solve as before:

$$E(c^2) = 2\lambda^{-2} F(2/\tau) \quad (14)$$
$$E(c^4) = 2\lambda^{-4} F(4/\tau) \ , \quad (15)$$

and

$$\frac{E(c^4)}{(E(c^2))^2} = \frac{F(4/\tau)}{2\,(F(2/\tau))^2} \ . \quad (16)$$

The right-hand side of Equation 16 is a strictly monotonic function of τ, and can be numerically evaluated using the following summation (see 3.424#2, p. 331 and 0.233#2, p. 7 of Gradshteyn and Ryzhik's table [Gradshteyn and Ryzhik, 1965]):

$$F(\beta) = \Gamma(\beta+1) \sum_{k=1}^{\infty} (-1)^{k+1}/k^\beta \quad (17)$$
$$= (1 - 2^{1-\beta})\zeta(\beta) \quad (18)$$

where $\zeta(\beta)$ is Riemann's zeta function.

Given estimates of the second and fourth moments, we can rapidly determine τ by doing binary search on the ratio in Equation 16.

Once τ has been chosen, we can approximate λ from the second moment:

$$\lambda \approx \sqrt{2F(2/\tau)/E(c^2)} \ . \quad (19)$$

Calibration using this two-parameter family is currently implemented in the SAM suite of HMM tools (starting with version 3.3) [Hughey et al., 1999].

There are other ways for fitting this two-parameter family besides moment-matching. Maximum likelihood estimation can again be done by maximizing the log of the likelihood function. First, we derive the density function from the cumulative distribution:

$$f(x) = \frac{d}{dx}(1 + e^{-(\lambda x)^\tau})^{-1}$$
$$= \lambda^\tau \tau x^{\tau-1} e^{-(\lambda x)^\tau} (1 + e^{-(\lambda x)^\tau})^{-2} \quad (20)$$

then we get the log likelihood:

$$\ln(L_n(\theta)) = \sum_{i=1}^{n} \ln p(x_i \mid \theta) \quad (21)$$
$$= \sum_{i=1}^{n} \ln(\lambda^\tau \tau x_i^{\tau-1} e^{-(\lambda x_i)^\tau} (1 + e^{-(\lambda x_i)^\tau})^{-2})$$
$$= n\tau \ln\lambda + n \ln\tau + (\tau - 1) \sum_{i=1}^{n} \ln x_i$$
$$-\lambda^\tau \sum_{i=1}^{n} x_i^\tau - 2 \sum_{i=1}^{n} \ln(e^{-(\lambda x_i)^\tau} + 1) \ . \quad (22)$$

To fit λ and τ we use a method where we first fix the value for the first variable, maximize the second using the *golden section*

method [W.T.Vetterling, 1988], then hold that value for the second while maximizing the first. We alternate until the change in the values for both variables falls below a specified tolerance. While the algorithm is not guaranteed to find a global maximum, it works well in practice when suitable initial values are chosen, in this case, 1.0 for both λ and τ.

5.2 Student's t distribution

The Student's t distribution family $t_\nu(\mu, \sigma^2)$ 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 ν, which is small for fat tails. The Student's t density function is

$$p(x) = \frac{\Gamma((\nu+1)/2)}{\Gamma(\nu/2)\sqrt{\nu\pi\sigma}} \left(1 + \frac{1}{\nu}\left(\frac{x-\mu}{\sigma}\right)^2\right)^{-(\nu+1)/2} . \tag{23}$$

Other than knowing that the mean μ is 0 and ν is small, we do not have much prior information about what values of ν and σ would best fit the distribution of costs. Following the suggestion of 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.

We wish to maximize the log-likelihood function

$$\begin{aligned} &\ln\left(L_n\left(\sigma,\nu\right)\right) = \\ &= \sum_{i=1}^{n} \ln p(x_i \mid \sigma, \nu) \qquad (24) \\ &= n\ln\left(\Gamma\left(\frac{\nu+1}{2}\right)\right) - n\ln\left(\Gamma\left(\frac{\nu}{2}\right)\right) \\ &\quad - \frac{n}{2}\ln(\pi) + \frac{n\nu}{2}\ln(\nu\sigma^2) \\ &\quad - \frac{(\nu+1)}{2}\sum_{i=1}^{n}\ln\left(\nu\sigma^2 + x_i^2\right) . \qquad (25) \end{aligned}$$

Because we cannot solve for ν and σ analytically, we initialize ν and σ^2 to approximate initial values (2.75 and 0.5 respectively) and use the numerical method we used in Section 5.1.

An alternative approach to maximum-likelihood is to use a least-squares fit of computed E-values to the ranks of the costs in the training set, as described in the Appendix for the derivation of the moment-matching techniques for the sigmoidal distribution. We can't use moment-matching for Student's t, as the values of ν that are appropriate for the distribution result in distributions with infinite 4th moments.

After fitting ν and σ^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 must be computed numerically—in our case, with the DCDFLIB library [Brown, 1997].

We also fit ν and σ by using the least-squares-fit method. The intent is to place more emphasis on the tails. Once again we use the alternating method to fit the variables for the least squares fit, using the same starting values as above. The DCDFLIB library [Brown, 1997] was used to provide an inverse to 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 (that is, assuming $\tau = 1$ and $\lambda = 1$). [Results 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 long, 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 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 [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 which 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, 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 [Karplus 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.

Figure 4 shows a database calibration of the same 1000 HMMs, using 4363 proteins as the calibration set and 2182 as the test set. Because we need many samples to estimate the parameters of the distributions for Figure 4, we used 2/3 of the sequences (4363) to set the parameters and the remaining 1/3 (2182) for the testing. To avoid contamination by the "lumpy" tail of true positives described in Section 6, all our calibration on testing on real data use only the positive costs. That means that the E-values reported here are not for the best fits to the HMMs, but for the worst fits.

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.

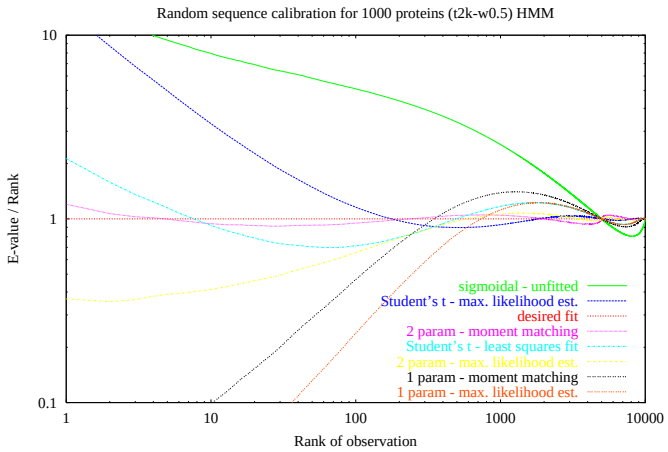


Figure 3: Averaged and normalized E-value vs. rank of reverse-sequence null model costs for 1000 HMMs on a test set of 10,000 random sequences.

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.

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.

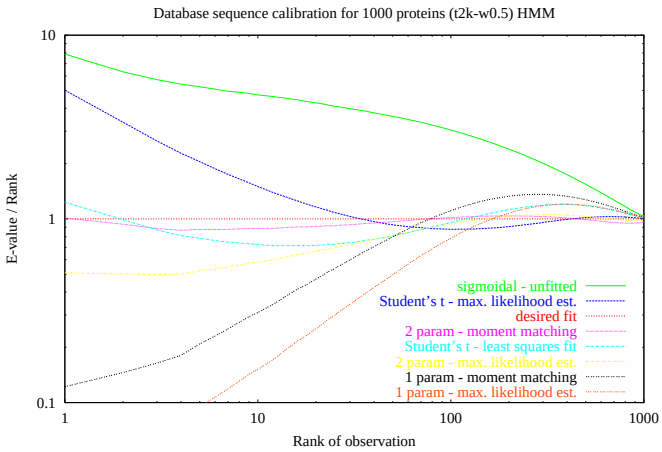


Figure 4: Averaged and normalized E-values vs. rank of reverse-sequence null model costs using single-track SAM-T2K HMMs for the same 1000 proteins for positive-cost sequences (out of the 2182 sequences scored).

E-values shown are the average of e-values computed using an unfitted sigmoidal distribution, one-parameter sigmoidal distribution fitted by either moment-matching or maximum likelihood, two-parameter sigmoidal distribution fitted by either moment-matching or maximum likelihood, and a two-parameter Student's t distribution fitted by either least-squares fit or maximum likelihood estimation. The fitting was done on the 4363 sequences that were not part of the test set.

At rank 10 the curves are in the order shown in the legend. The constant 1 is the desired behavior for a perfectly calibrated E-value.

We further examine the three best distributions by plotting the standard deviation of the log of the ratio of E-value and rank (Figures 5 and 6). 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.

8 Fold-recognition experiments

In a test of fold-recognition performance of reverse-sequence null model scores and E-values,

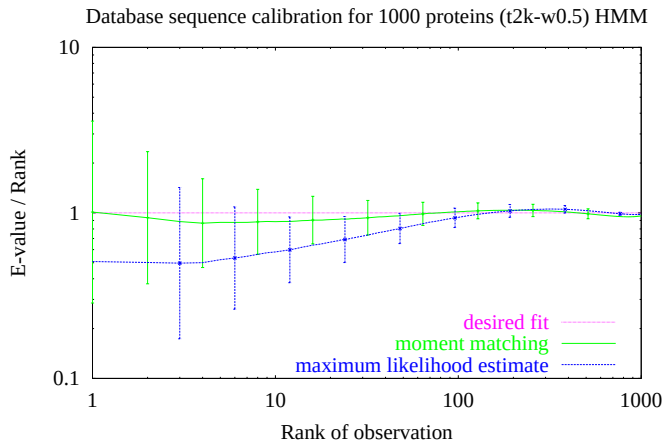


Figure 5: Averaged and normalized E-value vs. rank showing plus and minus the standard deviation for the log of the E-value/rank of the set of 1000 proteins.

E-values shown are the average and standard deviation of of log of E-values/rank computed using the two-parameter sigmoidal distribution fitted by moment-matching and maximum likelihood estimation. The moment-matching fits are consistently tighter than the maximum likelihood fits. The fitting was done on 4363 sequences that were not part of the test set.

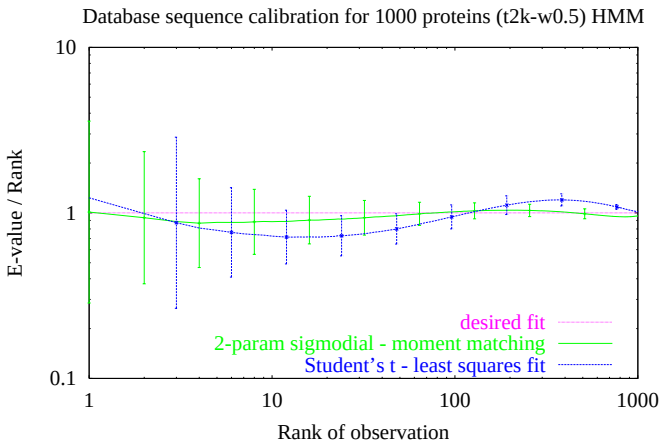


Figure 6: Averaged and normalized E-value vs. rank showing plus and minus the standard deviation for the log of the E-value/rank of the set of 1000 proteins.

E-values shown are the average and standard deviation of of log of E-values/rank computed using the two-parameter sigmoidal distribution fitted by moment-matching and Student's t distribution fitted by least-squares fit. The standard deviation of the values for the two-parameter sigmoidal fitted by is consistently tigher than Student's t fitted by least-squares fit. The fitting was done on 4363 sequences that were not part of the test set.

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 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. 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 (Figure 7), most likely because the secondary-structure sequences are not well modeled as independent draws from a composition.

9 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 mo-

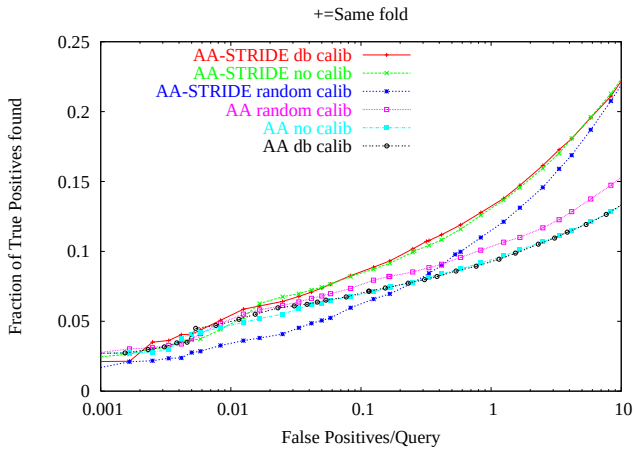


Figure 7: Fold-recognition results on 1298 chains. True positives are those pairs with a lower E-value than some threshold that contain domains in the same fold family in SCOP version 1.55. False positives are those pairs with lower E-values than the threshold that are in different fold families. The two-track HMMs used predictions of the STRIDE alphabet as the second track.

tifs 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 (Figure 2).

In contrast, the PROTEIN BLOCKS alphabet of secondary structure [de Brevern et al., 2000] is highly directed. This alphabet provides a fine-grained description of the protein backbone, consisting of sixteen canonical conformations, formed by backbone segments of five 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 (see Figure 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 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've not found a model that captures them well. Our best model for amino-acid sequences in protein databases remains a heirarchical model that models composition by a mixture

alphabet	0-order	1st-order	rev-for
amino acid	4.1896	4.1759	0.0153
STRIDE	2.3330	1.0455	0.0042
DSSP	2.5494	1.3387	0.0590
PROTEIN BLOCKS	3.3935	1.4876	3.0551

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. 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 counts, the first-order models were built using the zero-order probabilities as pseudocounts.

of Dirichlet distributions and the sequence as independent identically-distributed draws from that composition.

10 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.

Accurately 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 with 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 different approach for 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. Adding 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.

Acknowledgments

This work was supported in part by DOE grant DE-FG0395-99ER62849, NSF grants DBI-9808007 and EIA-9905322, NIH grant 1 R01 GM068570-01, and a National Physical Sciences Consortium graduate fellowship. We are grateful to Mark Diekhans for contributing to the software used in our experiments and to David Draper for suggesting Student's t distribution. Special thanks to the anonymous referees who suggested comparing to maximum-likelihood fits. Fixing this omission delayed publication by several months, but greatly improved the paper.

11 Appendix: Deriving least-squares fit and moment matching

There are two ways we can try to fit the distribution of a set of N reverse-sequence-null costs. 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. A somewhat simpler method is to match the moments of the distribution.

Let's examine the direct method first, since it can be used to fit just a portion of the data (for example, if we wanted to fit only the tails). First, invert the distribution function:

$$\begin{aligned} y &= (1 + e^{-\lambda a})^{-1} && (26) \\ a &= -\lambda^{-1} \ln(y^{-1} - 1) \ . && (27) \end{aligned}$$

We would like the estimated E-value of each score to be approximately the rank of the score—that is, we want

$$E(\text{number of } x < c_i) = N\text{P}(x < c_i) \approx i + 0.5 \ , \quad (28)$$

or

$$c_i \approx -\lambda^{-1} R_i \ , \quad (29)$$

where $R_i = \ln(N/(i + 0.5) - 1)$.

To get a least-squares fit, we want to minimize either

$$\sum_i (c_i + R_i/\lambda)^2 \quad (30)$$

or

$$\sum_i (\lambda c_i + R_i)^2 \ , \quad (31)$$

where the sum is taken over the subset of the costs we are interested in fitting (by default, all N of them). The two minimizations have somewhat different solutions.

Differentiating Equation 30 and setting the derivative to zero, we get

$$\lambda = \frac{-\sum_i R_i^2}{\sum_i R_i c_i} \ , \quad (32)$$

and differentiating Equation 31 we get

$$\lambda = \frac{-\sum_i R_i c_i}{\sum_i c_i^2} \ . \quad (33)$$

Combining these two estimates by taking the geometric mean, we get

$$\lambda = \sqrt{\sum_i R_i^2 / \sum_i c_i^2} \ , \quad (34)$$

which can be computed without needing to sort the costs (as long as we use the entire range of the data).

Of course, what we end up with by this direct method is just an estimate based on the moments of the distribution, since the first moment is always zero for this symmetric distribution, and the variance is $\sum_i c_i^2/N$. In fact, we can do better by eliminating the sum of R_i^2, since it is just an approximation to the expected value of $\sum_i c_i^2$, which we can derive analytically for the sigmoidal distribution.

First, we derive the density function from the cumulative distribution:

$$\begin{aligned} f(x) &= \frac{d}{dx}(1 + e^{-\lambda x})^{-1} \\ &= \lambda(1 + e^{-\lambda x})^{-1}(1 + e^{\lambda x})^{-1} && (35) \end{aligned}$$

and then compute the expected value of the square of the cost:

$$\begin{aligned} E(c^2) &= \int_{-\infty}^{\infty} f(x) x^2 dx \\ &= \lambda \int_{-\infty}^{\infty} \frac{x^2}{(1 + e^{-\lambda x})(1 + e^{\lambda x})} dx && (36) \end{aligned}$$

We can do a change of variable, $y = -\lambda x$, and look up the equation in Gradshteyn and Ryzhik's table of integrals (3.424#2, p. 331 for Equation 40 and 0.234#1, p. 7 for Equation 41) [Gradshteyn and Ryzhik, 1965]:

$$\begin{aligned} E(c^2) &= \\ &= \lambda^{-2} \int_{-\infty}^{\infty} \frac{y^2}{(1 + e^{y})(1 + e^{-y})} dy && (37) \\ &= 2\lambda^{-2} \int_{0}^{\infty} \frac{y^2}{(1 + e^{y})(1 + e^{-y})} dy && (38) \\ &= 2\lambda^{-2} \int_{0}^{\infty} y^2 \frac{e^y}{(1 + e^{y})^2} dy && (39) \end{aligned}$$

$$= 2\lambda^{-2}2!\sum_{k=1}^{\infty}(-1)^{k+1}/k^2 \quad (40)$$

$$= 2\lambda^{-2}2!\pi^2/12 \quad (41)$$

$$= (\pi^2/3)\lambda^{-2} \quad (42)$$

When estimating the variance for c, a small-sample correction is not needed, since we know that the mean should be 0—even a single sample gives us some information about the variance. So the computation formula for estimating λ is simply

$$\lambda \approx \pi\sqrt{N/\left(3\sum_{i=0}^{N-1} c_i^2\right)}\ . \quad (43)$$

Note that this computation, unlike the direct method in Equation 34 requires summing over the full set of costs. Alternatively, we can use the symmetry of the density function about 0 to estimate the second moment from just half the scores.

References

[Altschul et al., 1997] Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997). Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 25:3899–3402.

[Altschul, 1991] Altschul, S. F. (1991). Amino acid substitution matrices from an information theoretic perspective. *Journal of Molecular Biology*, 219:555–565.

[Altschul et al., 1990] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). A basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410.

[Bailey and Gribskov, 2002] Bailey, T. L. and Gribskov, M. (2002). Estimating and evaluating the statistics of gapped local-alignment scores. *Journal of Computational Biology*, 9:575–593.

[Baldi et al., 1994] Baldi, P., Chauvin, Y., Hunkapillar, T., and McClure, M. (1994). Hidden Markov models of biological primary sequence information. *Proceedings of the National Academy of Sciences, USA*, 91:1059–1063.

[Barrett et al., 1997] Barrett, C., Hughey, R., and Karplus, K. (1997). Scoring hidden Markov models. *Computer Applications in the Biosciences*, 13(2):191–199.

[Brown, 1997] Brown, B. W. (1997). DCDFLIB: Library of routines for cumulative distribution functions, inverses, and other parameters (C and Fortran). M.D. Anderson Cancer Center Biomathematics Archive. `http://odin.mdacc.tmc.edu/anonftp/page_2.html`.

[Bucher and Bairoch, 1994] Bucher, P. and Bairoch, A. (1994). A generalized profile syntax for biomolecular sequence motifs and its function in automatic sequence interpretation. In *Proceedings, 2nd International Conference on Intelligent Systems for Molecular Biology*, pages 53–61, Menlo Park, CA. AAAI/MIT Press.

[Bucher et al., 1996] Bucher, P., Karplus, K., Moeri, N., and Hoffman, K. (1996). A flexible motif search technique based on generalized profiles. *Computers and Chemistry*, 20(1):3–24.

[de Brevern et al., 2000] de Brevern, A., Etchebest, C., and Hazout, S. (2000). Bayesian probabilistic approach for predicting backbone structures in terms of protein blocks. *Proteins: Structure, Function and Genetics*, 41:271–287.

[Dunbrack, 2001] Dunbrack, R. (2001). Culling the PDB by resolution and sequence identity. `http://www.fccc.edu/research/labs/dunbrack/culledpdb.html`.

[Eddy, 1995] Eddy, S. (1995). Multiple alignment using hidden Markov models. In Rallings, C. et al., editors, *Proceedings, 3rd International Conference on Intelligent Systems for Molecular Biology*, pages 114–120, Menlo Park, CA. AAAI/MIT Press.

[Eddy et al., 1995] Eddy, S., Mitchison, G., and Durbin, R. (1995). Maximum discrimination hidden Markov models of sequence consensus. *J. Comput. Biol.*, 2:9–23.

[Frishman and Argos, 1995] Frishman, D. and Argos, P. (1995). Knowledge-based protein secondary structure assignment. *Proteins: Structure, Function, and Genetics*, 23:566–579.

[Gradshteyn and Ryzhik, 1965] Gradshteyn, I. S. and Ryzhik, I. M. (1965). *Table of Integrals, Series, and Products*. Academic Press, fourth edition.

[Grundy et al., 1997] Grundy, W. N., Bailey, W., Elkan, T., and Baker, C. (1997). Meta-MEME: Motif-based hidden Markov models of protein families. *Computer Applications in the Biosciences*, 13(4):397–406.

[Haussler et al., 1993] Haussler, D., Krogh, A., Mian, I. S., and Sjölander, K. (1993). Protein modeling using hidden Markov models: Analysis of globins. In *Proceedings of the Hawaii International Conference on System Sciences*, volume 1, pages 792–802, Los Alamitos, CA. IEEE Computer Society Press.

[Hughey et al., 1999] Hughey, R., Karplus, K., and Krogh, A. (1999). SAM: Sequence alignment and modeling software system, version 3. Technical Report UCSC-CRL-99-11, University of California, Santa Cruz, Computer Engineering, UC Santa Cruz, CA 95064. Available from `http://www.soe.ucsc.edu/research/compbio/sam.html`.

[Hughey and Krogh, 1996] Hughey, R. and Krogh, A. (1996). Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *Computer Applications in the Biosciences*, 12(2):95–107. Information on obtaining SAM is available at `http://www.soe.ucsc.edu/research/compbio/sam.html`.

[Kabsch and Sander, 1983] Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637.

[Karchin et al., 2004] Karchin, R., Cline, M., and Karplus, K. (2004). Evaluation of local structure alphabets based on residue burial. *Proteins: Structure, Function, and Genetics*, 55(3):508–518. Online: http://www3.interscience.wiley.com/cgi-bin/abstract/107632554/ABSTRACT.

[Karchin et al., 2003] Karchin, R., Cline, M., Mandel-Gutfreund, Y., and Karplus, K. (2003). Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. *Proteins: Structure, Function, and Genetics*, 51(4):504–514.

[Karchin and Hughey, 1998] Karchin, R. and Hughey, R. (1998). Weighting hidden Markov models for maximum discrimination. *Bioinformatics*, 14(9):772–782.

[Karplus, 2000] Karplus, K. (2000). gen_sequence: an open-source library. `http://www.soe.ucsc.edu/research/compbio/gen_sequence/`.

[Karplus et al., 1999] Karplus, K., Barrett, C., Cline, M., Diekhans, M., Grate, L., and Hughey, R. (1999). Predicting protein structure using only sequence information. *Proteins: Structure, Function, and Genetics*, Supplement 3(1):121–125.

[Karplus et al., 1998] Karplus, K., Barrett, C., and Hughey, R. (1998). Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856.

[Karplus et al., 2001] Karplus, K., Karchin, R., Barrett, C., Tu, S., Cline, M., Diekhans, M., Grate, L., Casper, J., and Hughey, R. (2001). What is the value added by human intervention in protein structure prediction? *Proteins: Structure, Function, and Genetics*, 45(S5):86–91.

[Karplus et al., 2003] Karplus, K., Karchin, R., Draper, J., Casper, J., Mandel-Gutfreund, Y., Diekhans, M., and Hughey, R. (2003). Combining local-structure, fold-recognition, and new-fold methods for protein structure prediction. *Proteins: Structure, Function, and Genetics*, 53(S6, pages=491-496).

[Karplus et al., 1997] Karplus, K., Sjölander, K., Barrett, C., Cline, M., Haussler, D., Hughey, R.,

Holm, L., and Sander, C. (1997). Predicting protein structure using hidden Markov models. *Proteins: Structure, Function, and Genetics*, Suppl. 1:134–139.

[Krogh et al., 1994] Krogh, A., Brown, M., Mian, I. S., Sjölander, K., and Haussler, D. (1994). Hidden Markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology*, 235:1501–1531.

[McClure et al., 1996] McClure, M., Smith, C., and Elton, P. (1996). Parameterization studies for the SAM and HMMER methods of hidden Markov model generation. In *Proceedings, 4rd International Conference on Intelligent Systems for Molecular Biology*, pages 155–164, St. Louis. AAAI Press.

[Murzin et al., 1995] Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C. (1995). SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540. Information on SCOP is available at `http://scop.mrc-lmb.cam.ac.uk/scop`.

[Schäffer et al., 2001] Schäffer, A. A., Aravind, L., Madden, T. L., Shavirin, S., Spouge, J. L., Wolf, Y. I., Koonin, E., and Altschul, S. F. (2001). Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Research*, 29(14):2994–3005.

[Smith and Waterman, 1981] Smith, T. F. and Waterman, M. S. (1981). Comparison of biosequences. *Advances in Applied Mathematics*, 2:482–489.

[Taylor, 1986] Taylor, W. (1986). Identification of protein sequence homology by consensus template alignment. *Journal of Molecular Biology*, 188:233–258.

[W.T.Vetterling, 1988] W.T.Vetterling, W. B. S. (1988). *Numerical Recipes in C*. Cambridge University Press.

[Yu and Hwa, 2001] Yu, Y. and Hwa, T. (2001). Statistical significance of probabilistic sequence alignment and related local hidden Markov models. *Journal of Computational Biology*, 8(3):249–282.