Estimating Statistical Significance for Reverse-sequence Null Models

Kevin Karplus

University of California, Santa Cruz

Supported in part by NSF grant DBI-9808007, DOE grant DE-FG03-99ER62849, and NSF grant EIA-9905322
Outline of Talk

- What is a null model?
- Why use the reverse-sequence null?
- Two approaches to statistical significance.
- What distribution do we expect for scores?
- Fitting the distribution.
- Does calibrating the E-values help?
Scoring hmms and Bayes Rule

- The model $M$ is a computable function that assigns a probability $\text{Prob}(A \mid M)$ to each string $A$.

- When given a string $A$, we want to know how likely the model is. That is, we want to compute something like $\text{Prob}(M \mid A)$.

- Bayes Rule:
  \[
  \text{Prob}(M \mid A) = \text{Prob}(A \mid M) \frac{\text{Prob}(M)}{\text{Prob}(A)}.
  \]

- Problem: $\text{Prob}(A)$ and $\text{Prob}(M)$ are inherently unknowable.
Null models

- Standard solution: ask how much more likely $M$ is than some null hypothesis (represented by a null model).

\[
\frac{\text{Prob} \left( M \mid A \right)}{\text{Prob} \left( N \mid A \right)} = \frac{\text{Prob} \left( A \mid M \right) \text{Prob}(M)}{\text{Prob} \left( A \mid N \right) \text{Prob}(N)}.
\]

- $\frac{\text{Prob}(M)}{\text{Prob}(N)}$ is the prior odds ratio, and represents our belief in the likelihood of the model before seeing any data.

- $\frac{\text{Prob}(M \mid A)}{\text{Prob}(N \mid A)}$ is the posterior odds ratio, and represents our belief in the likelihood of the model after seeing the data.

- We can generalize to a forced choice among many models ($M_1, \ldots, M_n$)

\[
\frac{\text{Prob} \left( M_i \mid A \right)}{\sum_j \text{Prob} \left( M_j \mid A \right)} = \frac{\text{Prob} \left( A \mid M_i \right) \text{Prob}(M_i)}{\sum_j \text{Prob} \left( A \mid M_j \right) \text{Prob}(M_j)}.
\]

The Prob($M_j$) values can be scaled arbitrarily without affecting the ratio.
Standard Null Model

- Null model is an i.i.d (independent, identically distributed) model, that is, each letter is treated as being independently drawn from the background distribution.

\[ \text{Prob}(A \mid N, \text{len}(A)) = \prod_{i=1}^{\text{len}(A)} \text{Prob}(A_i). \]

\[ \text{Prob}(A \mid N) = \text{Prob}(\text{string of length len}(A)) \prod_{i=1}^{\text{len}(A)} \text{Prob}(A_i). \]

- The length modeling is often omitted, but one must be careful then to normalize the probabilities correctly.
When using the standard null model, certain sequences and HMMs have anomalous behavior. Many of the problems are due to unusual composition—a large number of some usually rare amino acid.

For example, metallothionein, with 24 cysteines in only 61 total amino acids, scores well on any model with multiple highly conserved cysteines.

We avoid this (and several other problems) by using a reversed model \( M^r \) as the null model.

The probability of a sequence in \( M^r \) is exactly the same as the probability of the reversal of the sequence given \( M \).

If we assume that \( M \) and \( M^r \) are equally likely, then

\[
\frac{\text{Prob} (M \mid S)}{\text{Prob} (M^r \mid S)} = \frac{\text{Prob} (S \mid M)}{\text{Prob} (S \mid M^r)}.
\]

This method corrects for composition biases, length biases, and several subtler biases.
A cysteine-rich protein, such as metallothionein, can match any HMM that has several highly-conserved cysteines, even if they have quite different structures:

<table>
<thead>
<tr>
<th>HMM</th>
<th>sequence</th>
<th>cost in nats</th>
<th>model — standard null</th>
<th>model — reversed-model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1kst</td>
<td>4mt2</td>
<td>-21.15</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>1kst</td>
<td>1tabl</td>
<td>-15.04</td>
<td>-0.93</td>
<td></td>
</tr>
<tr>
<td>4mt2</td>
<td>1kst</td>
<td>-15.14</td>
<td>-0.10</td>
<td></td>
</tr>
<tr>
<td>4mt2</td>
<td>1tabl</td>
<td>-21.44</td>
<td>-1.44</td>
<td></td>
</tr>
<tr>
<td>1tabl</td>
<td>1kst</td>
<td>-17.79</td>
<td>-7.72</td>
<td></td>
</tr>
<tr>
<td>1tabl</td>
<td>4mt2</td>
<td>-19.63</td>
<td>-1.79</td>
<td></td>
</tr>
</tbody>
</table>
Composition examples

Metallothionein Isoform II (4mt2)

Kistrin (1kst)
Composition examples

Kistrin (1kst)

Trypsin-binding domain of Bowman-Birk Inhibitor (1tabl)
Long helices as source of error

Long helices can provide strong similarity signals from the periodic hydrophobicity, even when the overall folds are quite different:

<table>
<thead>
<tr>
<th>HMM</th>
<th>sequence</th>
<th>cost in nats, normalized using Null model</th>
<th>reversed-model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1av1A</td>
<td>2tmaA</td>
<td>-22.06</td>
<td>2.13</td>
</tr>
<tr>
<td>1av1A</td>
<td>laep</td>
<td>-21.25</td>
<td>1.03</td>
</tr>
<tr>
<td>1av1A</td>
<td>1cii</td>
<td>-13.67</td>
<td>-1.75</td>
</tr>
<tr>
<td>1av1A</td>
<td>lvsgA</td>
<td>-7.89</td>
<td>-0.51</td>
</tr>
<tr>
<td>2tmaA</td>
<td>1cii</td>
<td>-20.62</td>
<td>0.46</td>
</tr>
<tr>
<td>2tmaA</td>
<td>1av1A</td>
<td>-17.96</td>
<td>1.01</td>
</tr>
<tr>
<td>2tmaA</td>
<td>laep</td>
<td>-12.01</td>
<td>0.78</td>
</tr>
<tr>
<td>2tmaA</td>
<td>lvsgA</td>
<td>-8.25</td>
<td>0.08</td>
</tr>
<tr>
<td>1vsgA</td>
<td>2tmaA</td>
<td>-14.82</td>
<td>-1.20</td>
</tr>
<tr>
<td>1vsgA</td>
<td>1av1A</td>
<td>-13.04</td>
<td>-2.68</td>
</tr>
<tr>
<td>1vsgA</td>
<td>laep</td>
<td>-13.02</td>
<td>-3.52</td>
</tr>
<tr>
<td>1vsgA</td>
<td>1cii</td>
<td>-11.12</td>
<td>0.28</td>
</tr>
<tr>
<td>1aep</td>
<td>1av1A</td>
<td>-11.30</td>
<td>1.79</td>
</tr>
<tr>
<td>1aep</td>
<td>2tmaA</td>
<td>-10.73</td>
<td>1.06</td>
</tr>
<tr>
<td>1aep</td>
<td>1cii</td>
<td>-8.35</td>
<td>1.38</td>
</tr>
<tr>
<td>1aep</td>
<td>lvsgA</td>
<td>-6.87</td>
<td>0.53</td>
</tr>
<tr>
<td>1cii</td>
<td>2tmaA</td>
<td>-23.24</td>
<td>-1.48</td>
</tr>
<tr>
<td>1cii</td>
<td>1av1A</td>
<td>-19.49</td>
<td>-5.62</td>
</tr>
<tr>
<td>1cii</td>
<td>laep</td>
<td>-12.85</td>
<td>-1.77</td>
</tr>
<tr>
<td>1cii</td>
<td>lvsgA</td>
<td>-10.20</td>
<td>-1.57</td>
</tr>
</tbody>
</table>
Helix examples

Tropomyosin (2tmaA)

Colicin Ia (1ciι)

Flavodoxin mutant (1vsgA)
Helix examples

Apolipophorin III (1aep)

Apolipoprotein A-I (1av1A)
SCOP whole chains

without reversed-model scoring
with reversed-model scoring

False Positives vs. True Positives graph.
What is Statistical Significance?

- The statistical significance of a hit, \( P_1 \), is the probability of getting a score as good as the hit “by chance,” when scoring a single “random” sequence.
- When searching a database of \( N \) sequences, the significance is best reported as an E-value—the expected number of sequences that would score that well by chance: \( E = P_1 N \).
- Some people prefer the p-value: \( P_N = 1 - (1 - P_1)^N \). For large \( N \), \( P_N \approx 1 - e^{-E} \), so \( P_N \) is essentially the same as \( E \) for small E-values.
- I prefer to use E-values, because our best scores are often not significant, and it is easier to distinguish between E-values of 10, 100, and 1000 than between p-values of 0.999955, 1 - 4E-44, and 1 - 5E-435.
• (Markov’s inequality) For any scoring scheme that uses
\[
\ln \frac{\text{Prob (seq } \mid M_1)}{\text{Prob (seq } \mid M_2)}
\]
the probability of a score better than $T$ is less than $e^{-T}$ for sequences distributed according to $M_2$. This method is independent of the actual probability distributions. We have had good results with this method.

• (Classical parameter fitting) If the “random” sequences are not drawn from the distribution $M_2$, but from some other distribution, then we can try to fit some parameterized family of distributions to scores from a random sample, and use the parameters to compute $P_1$ and $E$ values for scores of real sequences.
What family should we use for reverse-sequence null?

**Bad assumption 1:** The scores with a standard null model are distributed according to an extreme-value distribution:

\[ P(\ln \text{Prob}(\text{seq} \mid M) > T) \approx G_{k,\lambda}(T) = 1 - \exp(-ke^{\lambda T}) . \]

**Bad assumption 2:** The scores with the model and the reverse-model are independent of each other.

**Result:** The scores using a reverse-sequence null model are distributed according to a sigmoidal function:

\[ P(\text{score} > T) = (1 - e^{\lambda T})^{-1} . \]
(Derivation for costs, not scores, so more negative is better.)

\[
P(\text{cost} < T) = \int_{-\infty}^{\infty} P(c_M = x) \int_{x-T}^{\infty} P(c_{M'} = y) dy \, dx \\
= \int_{-\infty}^{\infty} P(c_M = x) P(c_{M'} > x - T) \, dx \\
= \int_{-\infty}^{\infty} k \lambda \exp(-ke^{\lambda x})e^{\lambda x} \exp(-k e^{\lambda(x-T)}) \, dx \\
= \int_{-\infty}^{\infty} k \lambda e^{\lambda x} \exp(-k(1 + e^{-\lambda T})e^{\lambda x}) \, dx
\]

If we introduce a temporary variable to simplify the formulas:
\[
K_T = k(1 + \exp(-\lambda T)),
\]
then

\[
P(\text{cost} < T) = \int_{-\infty}^{\infty} (1 + e^{-\lambda T})^{-1} K_T \lambda e^{\lambda x} \exp(-K_T e^{\lambda x}) \, dx \\
= (1 + e^{-\lambda T})^{-1} \int_{-\infty}^{\infty} K_T \lambda e^{\lambda x} \exp(-K_T e^{\lambda x}) \, dx \\
= (1 + e^{-\lambda T})^{-1} \int_{-\infty}^{\infty} g_{K_T, \lambda}(x) \, dx \\
= (1 + e^{-\lambda T})^{-1}
\]
Fitting $\lambda$

- The $\lambda$ parameter simply scales the scores (or costs) before the sigmoidal distribution, so $\lambda$ can be set by matching the observed variance to the theoretically expected variance.
- The mean is theoretically (and experimentally) zero.
- The variance is easily computed, though derivation is messy:

\[
E(c^2) = (\pi^2/3)\lambda^{-2}.
\]

- $\lambda$ is easily fit by matching the variance:

\[
\lambda \approx \pi \sqrt{N/(3\sum_{i=0}^{N-1} c_i^2)}.
\]
• We made two dangerous assumptions: extreme-value and independence.

• To give ourselves some room to compensate for deviations from these assumptions, we can add another parameter to the family.

• We can replace $-\lambda T$ with any strictly decreasing odd function.

• Somewhat arbitrarily, we chose

$$-\text{sign}(T)|\lambda T|^\tau$$

so that we could match a “stretched exponential” tail.
Fitting a two-parameter family

- For two-parameter symmetric distribution, we can fit using 2nd and 4th moments:

\[
E(c^2) = \lambda^{-2/\tau} K_{2/\tau}, \\
E(c^4) = \lambda^{-4/\tau} K_{4/\tau},
\]

where \( K_x \) is a constant:

\[
K_x = \int_{-\infty}^{\infty} y^x (1 + e^y)^{-1} (1 + e^{-y})^{-1} dy \\
= -\Gamma(x + 1) \sum_{k=1}^{\infty} (-1)^k / k^x.
\]

- The ratio \( E(c^4)/(E(c^2))^2 \) is independent of \( \lambda \) and monotonic in \( \tau \), so we can fit \( \tau \) by binary search.

- Once \( \tau \) is chosen we can fit \( \lambda \) using \( E(c^2) \).
Calibration for 3chy.t2k-w0.5 HMM

Computed E-value vs. Rank of observation for different parameter values:
- Desired fit
- \( \tau = 1, \lambda = 1 \)
- \( \tau = 1, \lambda = 1.7628 \)
- \( \tau = 0.6757, \lambda = 3.0065 \)
Example for two-track HMM

Calibration for 3chy 2-track HMM

Computed E-value vs. Rank of observation

- desired fit
- \( \tau=1, \lambda=1 \)
- \( \tau=1, \lambda=22.1692 \)
- \( \tau=0.6132, \lambda=51.9222 \)
Fold recognition results

Fold recognition test (same superfamily=+, different fold=-)

- AA uncalib
- AA calib
- 2-track uncalib
- 2-track calib

False positives vs. True positives
Wave hands and say “but we can fix that”

- Why did calibrated fold recognition fail for 2-track HMMs?
- “Random” secondary structure sequences (i.i.d. model) are not representative of real sequences.
- Fixes:
  - Better secondary structure decoy generator.
  - Use real database, but avoid problems with contamination by true positives by taking only costs $> 0$ to get estimate of $E(\text{cost}^2)$ and $E(\text{cost}^4)$. 