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*).

$$\frac{\mathrm{Prob}\,(M \mid A)}{\mathrm{Prob}\,(N \mid A)} = \frac{\mathrm{Prob}\,(A \mid M)}{\mathrm{Prob}\,(A \mid N)} \; \frac{\mathrm{Prob}(M)}{\mathrm{Prob}(N)} \; .$$

- $\frac{\mathrm{Prob}(M)}{\mathrm{Prob}(N)}$ is the *prior odds ratio*, and represents our belief in the likelihood of the model before seeing any data.
- $\frac{\mathrm{Prob}(M|A)}{\mathrm{Prob}(N|A)}$ 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 $(M_1, \ldots, M_n)$

$$\frac{\mathrm{Prob}\,(M_i \mid A)}{\sum_j \mathrm{Prob}\,(M_j \mid A)} = \frac{\mathrm{Prob}\,(A \mid M_i)\,\mathrm{Prob}(M_i)}{\sum_j \mathrm{Prob}\,(A \mid M_j)\,\mathrm{Prob}(M_j)} \; .$$

The $\mathrm{Prob}(M_j)$ 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.

- $$\mathrm{Prob}\left(A \mid N, \mathrm{len}\left(A\right)\right) = \prod_{i=1}^{\mathrm{len}(A)} \mathrm{Prob}(A_i) \ .$$

- $$\mathrm{Prob}\left(A \mid N\right) = \mathrm{Prob}(\text{string of length } \mathrm{len}\left(A\right)) \prod_{i=1}^{\mathrm{len}(A)} \mathrm{Prob}(A_i) \ .$$

- The length modeling is often omitted, but one must be careful then to normalize the probabilities correctly.

Reversed model for null

- 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

$$\frac{\mathrm{Prob}\,(M \mid S)}{\mathrm{Prob}\,(M^r \mid S)} = \frac{\mathrm{Prob}\,(S \mid M)}{\mathrm{Prob}\,(S \mid M^r)} \ .$$

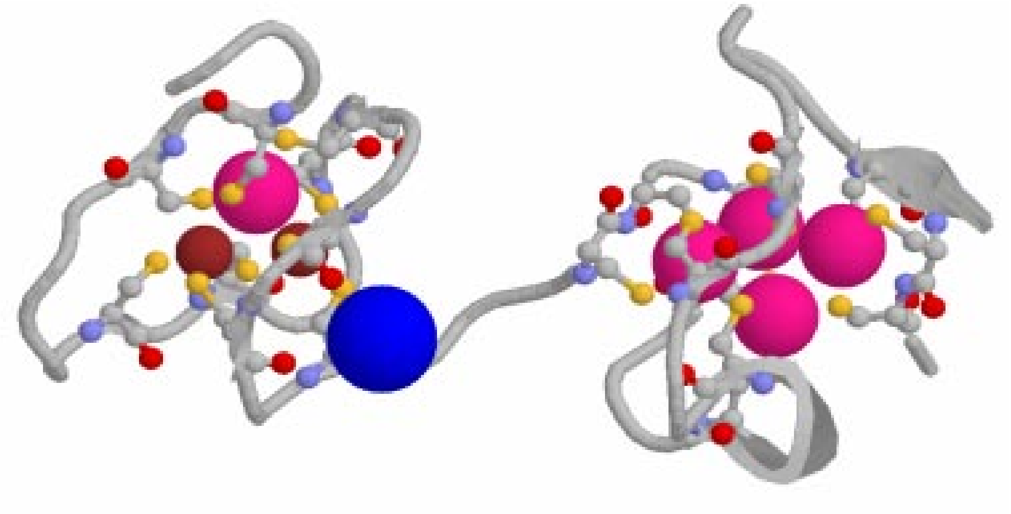
- This method corrects for composition biases, length biases, and several subtler biases.

Composition as source of error

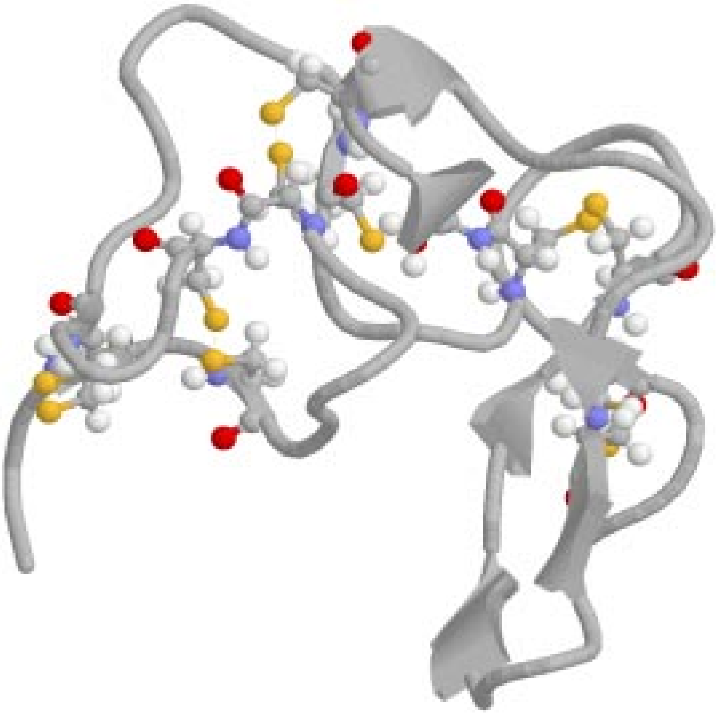
A cysteine-rich protein, such as metallothionein, can match any HMM that has several highly-conserved cysteines, even if they have quite different structures:

		cost in nats	
		model −	model −
HMM	sequence	standard null	reversed-model
1kst	4mt2	-21.15	0.01
1kst	1tabI	-15.04	-0.93
4mt2	1kst	-15.14	-0.10
4mt2	1tabI	-21.44	-1.44
1tabI	1kst	-17.79	-7.72
1tabI	4mt2	-19.63	-1.79

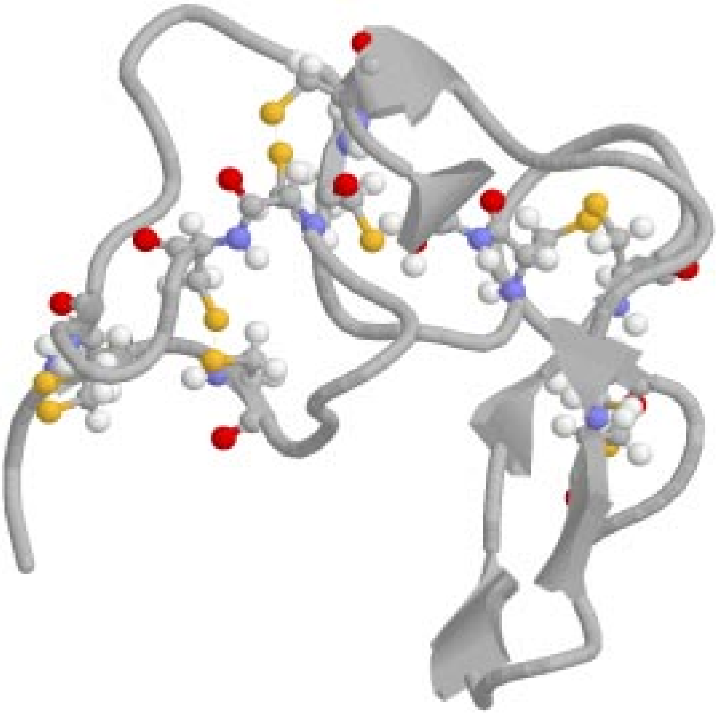
Metallothionein Isoform II (4mt2)



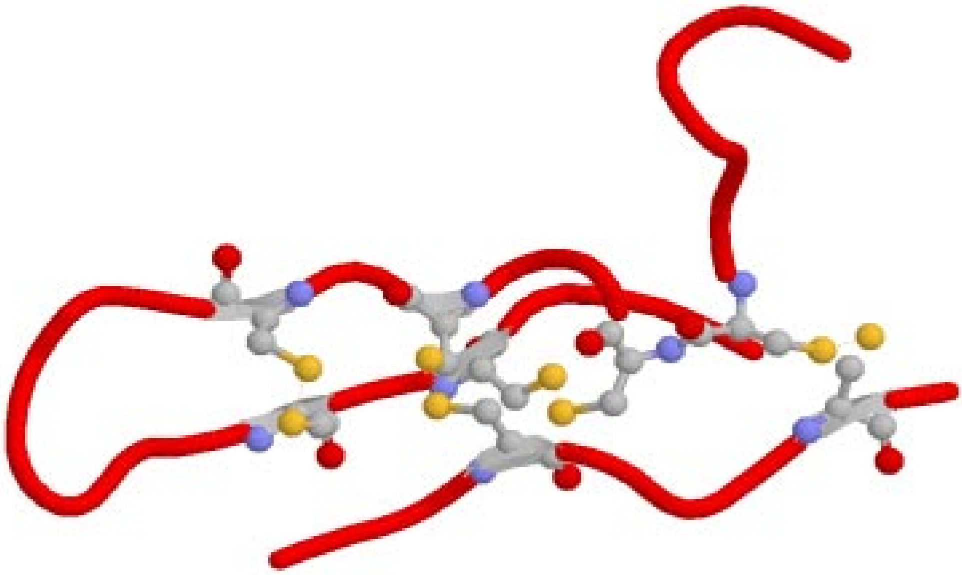
Kistrin (1kst)



Kistrin (1kst)



Trypsin-binding domain of Bowman-Birk Inhibitor (1tabI)



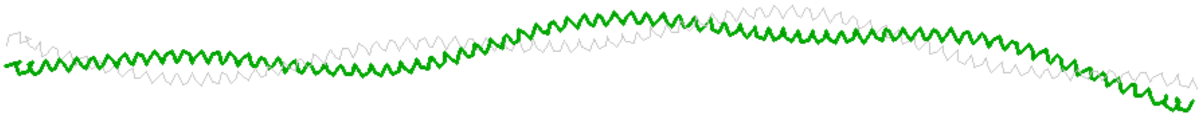
Long helices as source of error

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

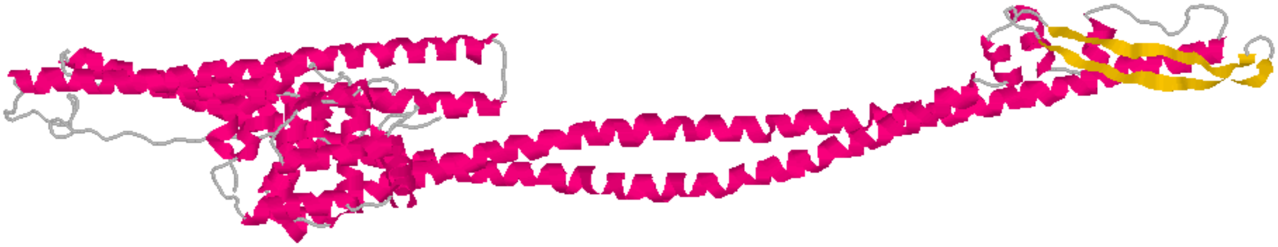
		cost in nats, normalized using	
HMM	sequence	Null model	reversed-model
1av1A	2tmaA	-22.06	2.13
1av1A	1aep	-21.25	1.03
1av1A	1cii	-13.67	-1.75
1av1A	1vsgA	-7.89	-0.51
2tmaA	1cii	-20.62	0.46
2tmaA	1av1A	-17.96	1.01
2tmaA	1aep	-12.01	0.78
2tmaA	1vsgA	-8.25	0.08
1vsgA	2tmaA	-14.82	-1.20
1vsgA	1av1A	-13.04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

Helix examples

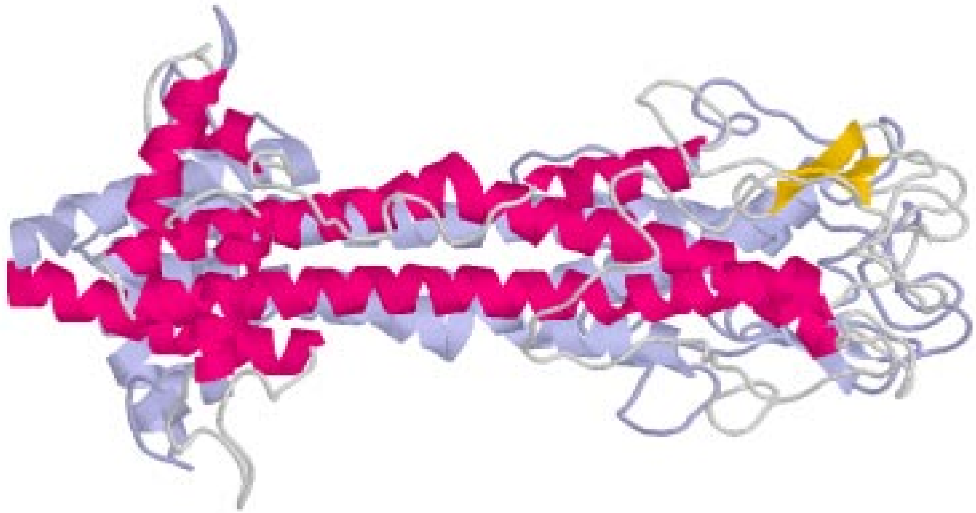
Tropomyosin (2tmaA)



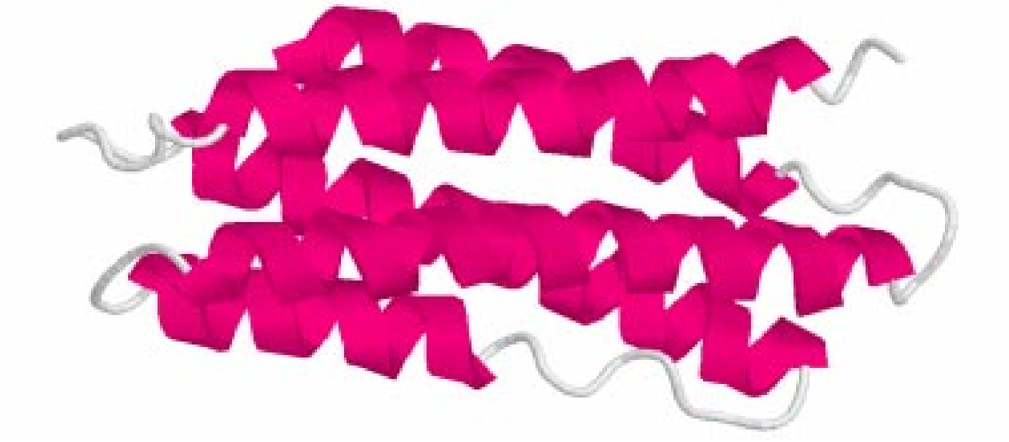
Colicin Ia (1cii)



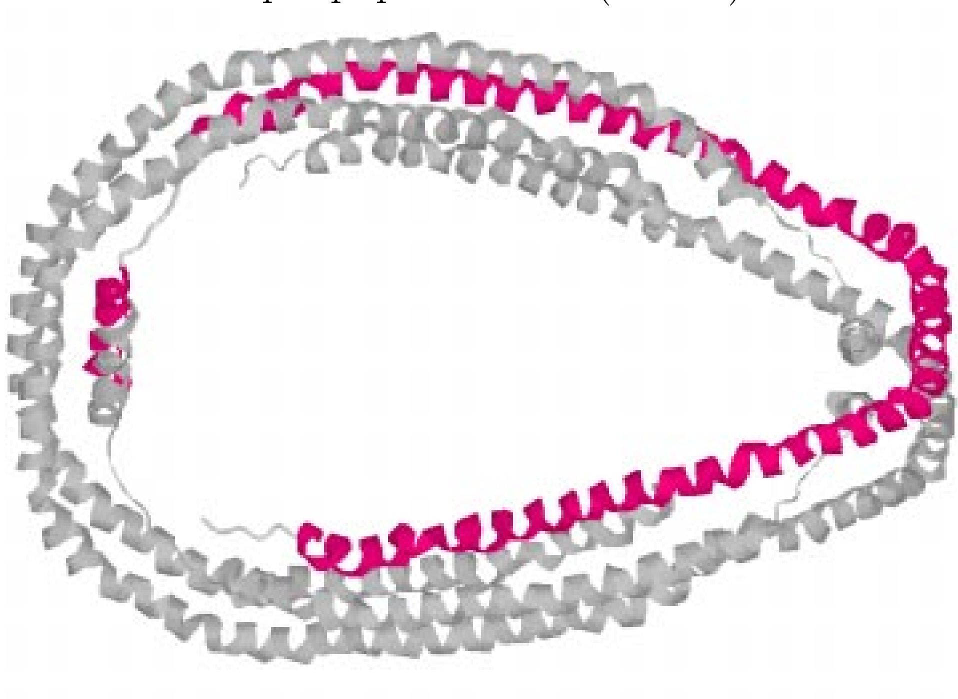
Flavodoxin mutant (1vsgA)

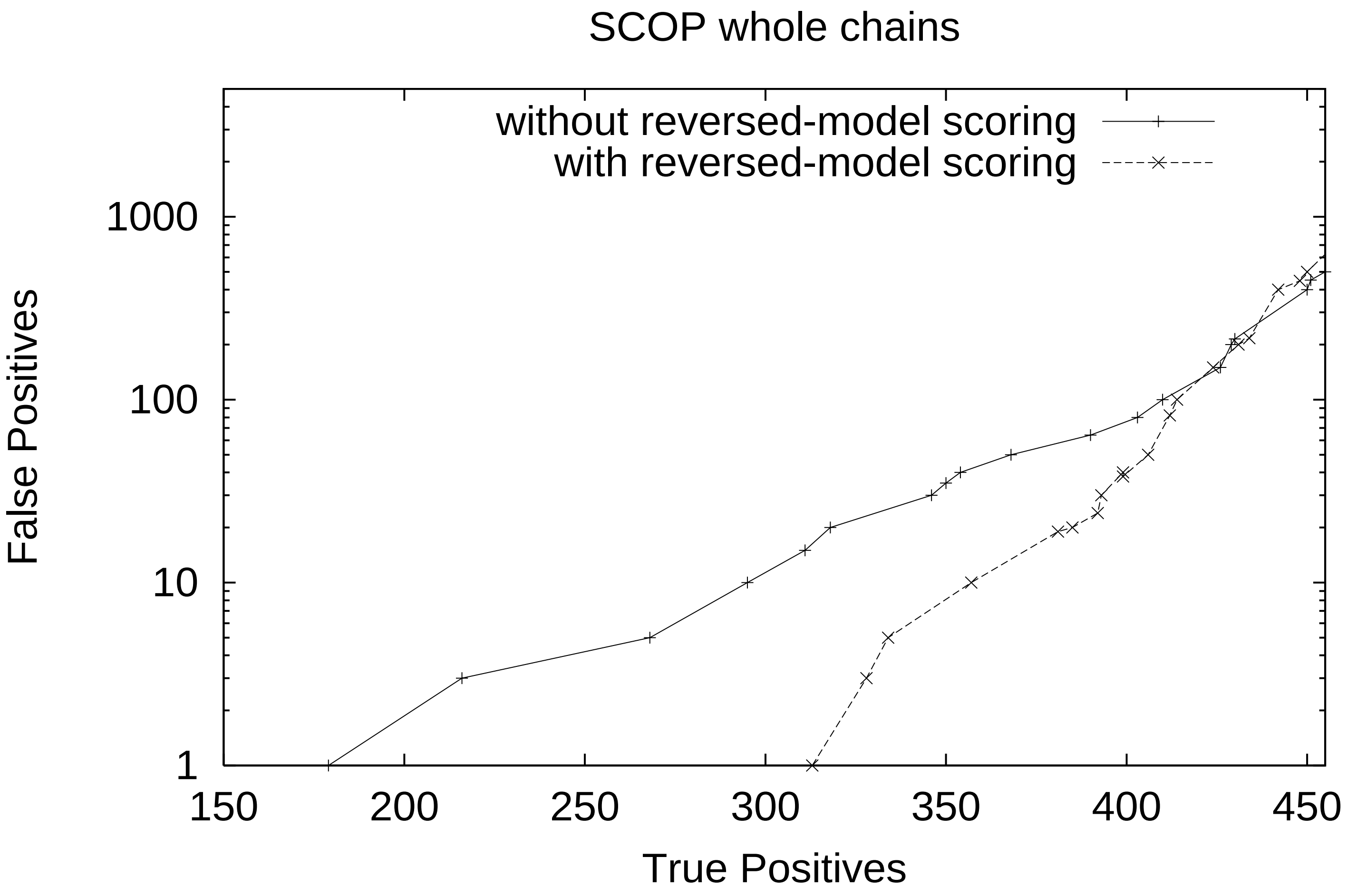


Apolipophorin III (1aep)



Apolipoprotein A-I (1av1A)




SCOP whole chains
without reversed-model scoring
with reversed-model scoring
False Positives
1000
100
10
1
150
200
250
300
350
400
450
True Positives

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, $1 - 4$E-44, and $1 - 5$E-435

Two Approaches to Statistical Significance

- (Markov's inequality) For any scoring scheme that uses

$$\ln \frac{\mathrm{Prob}\,(\mathrm{seq} \mid M_1)}{\mathrm{Prob}\,(\mathrm{seq} \mid M_2)}$$

  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:

$$P\left(\ln \mathrm{Prob}\left(\mathrm{seq} \mid M\right) > T\right) \approx G_{k,\lambda}(T) = 1 - \exp(-k e^{\lambda T}) \ .$$

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:

$$P(\mathrm{score} > T) = (1 - e^{\lambda T})^{-1} \ .$$

Derivation of sigmoidal distribution

(Derivation for *costs*, not *scores*, so more negative is better.)

$$\begin{aligned}
P(\text{cost} < T) &= \int_{-\infty}^{\infty} P(c_M = x) \int_{x-T}^{\infty} P(c_{M'} = y) dy dx \\
&= \int_{-\infty}^{\infty} P(c_M = x) P(c_{M'} > x - T) dx \\
&= \int_{-\infty}^{\infty} k\lambda \exp(-k e^{\lambda x}) e^{\lambda x} \exp(-k e^{\lambda(x-T)}) dx \\
&= \int_{-\infty}^{\infty} k\lambda e^{\lambda x} \exp(-k(1 + e^{-\lambda T}) e^{\lambda x}) dx
\end{aligned}$$

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

$$\begin{aligned}
P(\text{cost} < T) &= \int_{-\infty}^{\infty} (1 + e^{-\lambda T})^{-1} K_T \lambda e^{\lambda x} \exp(-K_T e^{\lambda x}) dx \\
&= (1 + e^{-\lambda T})^{-1} \int_{-\infty}^{\infty} K_T \lambda e^{\lambda x} \exp(-K_T e^{\lambda x}) dx \\
&= (1 + e^{-\lambda T})^{-1} \int_{-\infty}^{\infty} g_{K_T,\lambda}(x) dx \\
&= (1 + e^{-\lambda T})^{-1}
\end{aligned}$$

Fitting λ

- The λ parameter simply scales the scores (or costs) before the sigmoidal distribution, so λ 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:

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

- λ is easily fit by matching the variance:

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

Two-parameter family

- 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

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

  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:

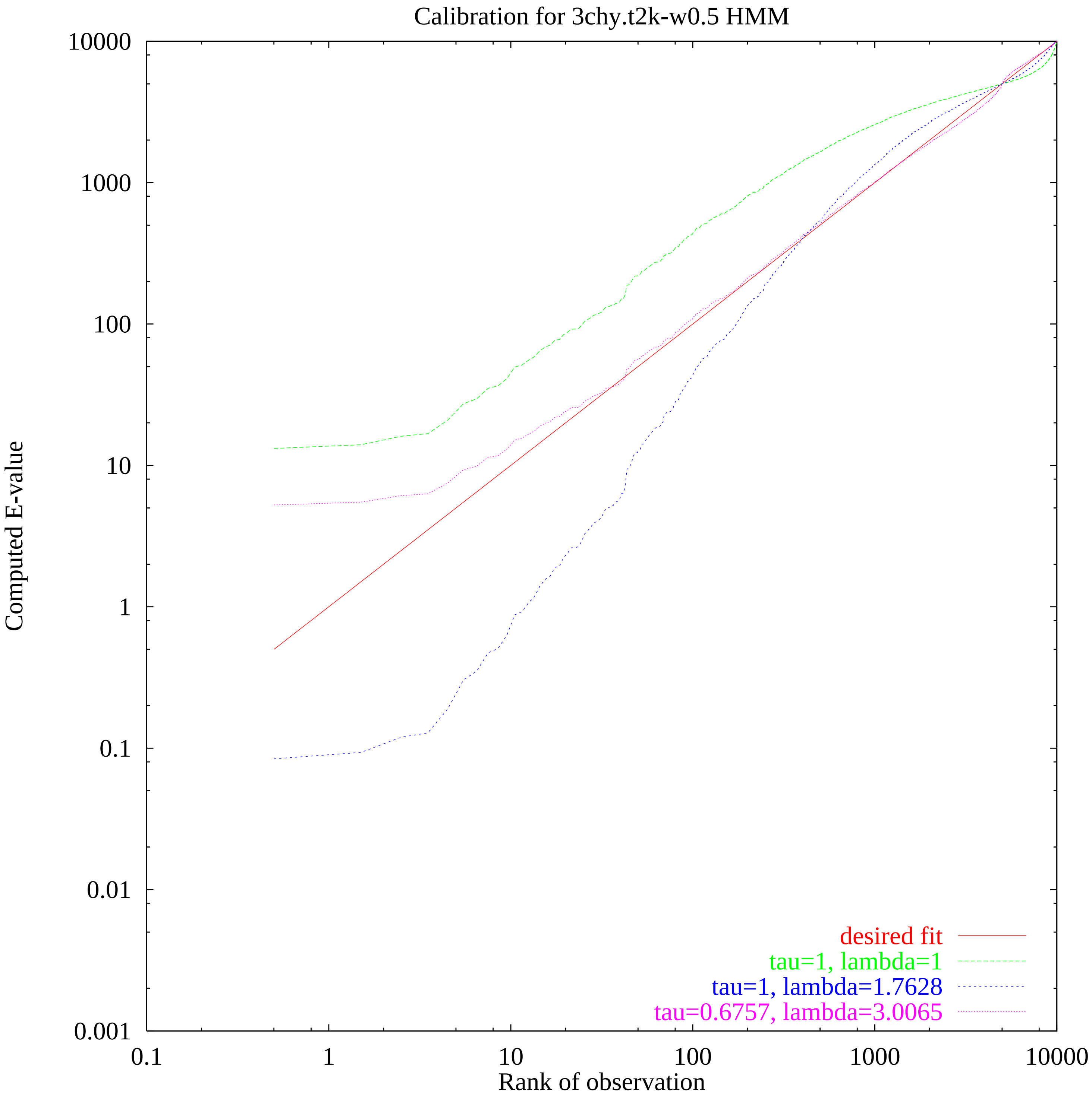
$$\begin{aligned} E(c^2) &= \lambda^{-2/\tau} K_{2/\tau} \\ E(c^4) &= \lambda^{-4/\tau} K_{4/\tau} \end{aligned}$$

where K_x is a constant:

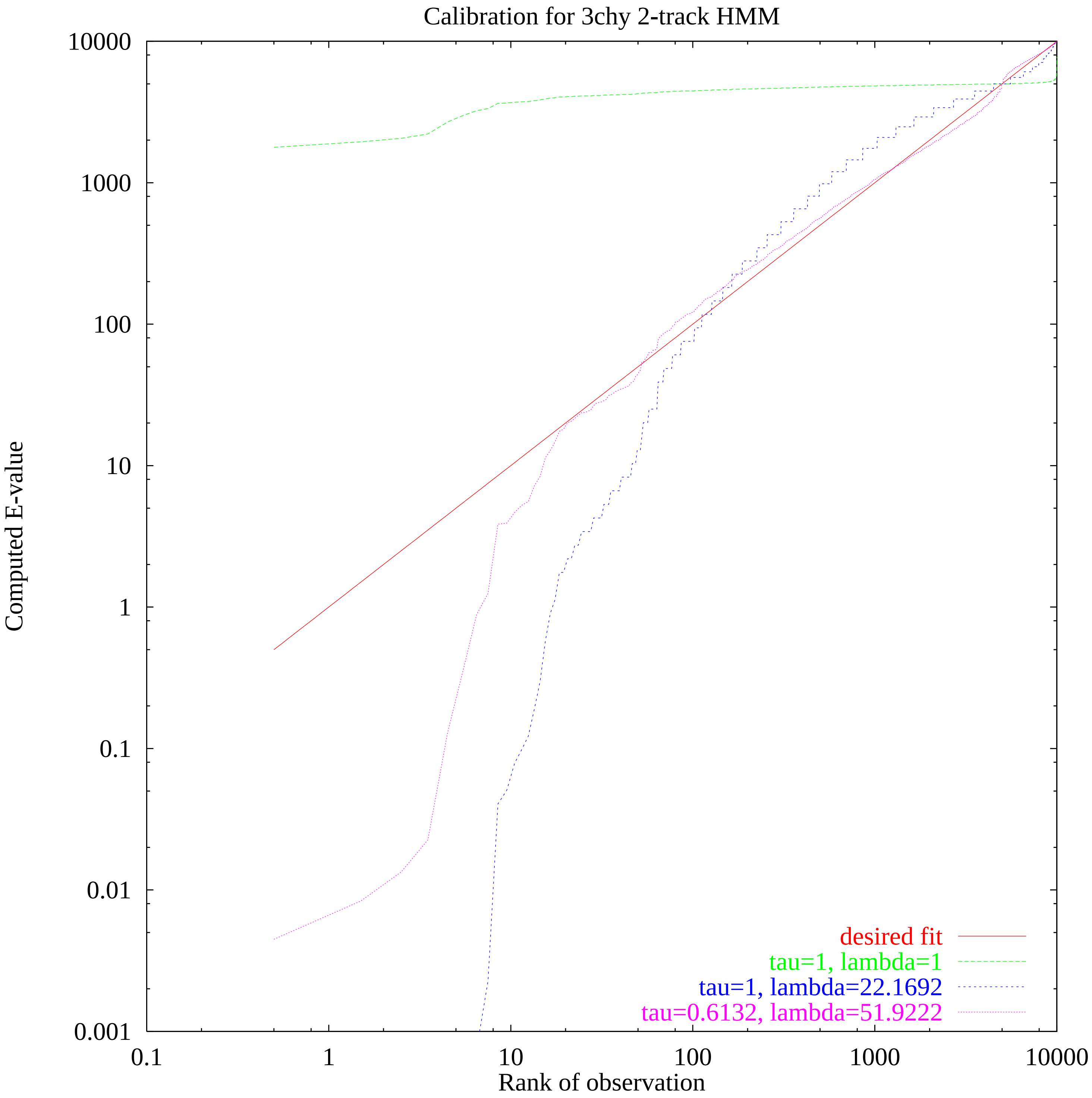
$$\begin{aligned} K_x &= \int_{-\infty}^{\infty} y^x (1+e^y)^{-1}(1+e^{-y})^{-1} dy \\ &= -\Gamma(x+1) \sum_{k=1}^{\infty} (-1)^k / k^x \ . \end{aligned}$$

- The ratio $E(c^4)/(E(c^2))^2$ is independent of λ and monotonic in τ, so we can fit τ by binary search.
- Once τ is chosen we can fit λ using $E(c^2)$.

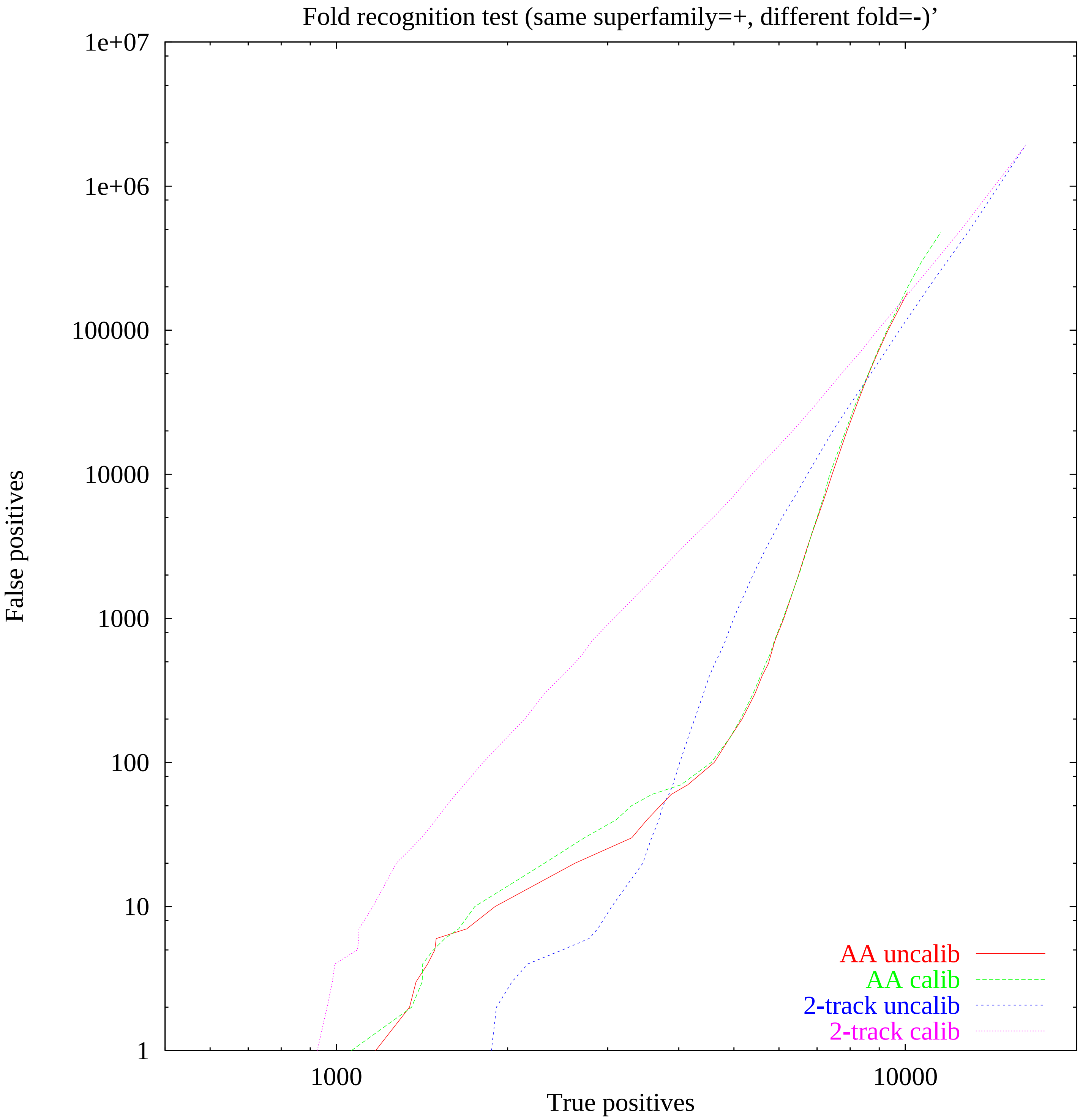
Example for single-track HMM



Example for two-track HMM



Fold recognition results



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 $E(\text{cost}^4)$.