Fold Recognition using Hidden Markov Models and Secondary Structure

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

- Fold-recognition
- Scoring (Bayesian statistical modeling)
- SAM-T2K for finding and aligning homologs
- Multi-track HMMs and secondary structure
- Reverse-sequence null model
- Results
The Folding Problem:
Given a protein expressed as a string $A$ over the alphabet of 20 amino acids
($A \in \{a, c, d, e, f, g, h, i, k, l, m, n, p, q, r, s, t, v, w, y\}$),
figure out how it folds up in 3-space.
The Fold-recognition Problem:
Given a protein expressed as a string $A$ over the alphabet of amino acids (the target sequence) and a library of proteins with known 3-D structures (the template library), figure out which template(s) $A$ matches best, and align the target to the template.

- The backbone for the target sequence is predicted to be very similar to the backbone of the chosen template.
- A quality measure is needed to decide when the best-matching template is still not a good match.
Remote-homology Problem

The Homology Problem:
Given a protein expressed as a string $A$ over the alphabet of amino acids (the target sequence), and a library of protein sequences, figure out which sequences $A$ is similar to and align them to $A$.

- This problem is fairly easy for recently diverged, very similar sequences, but difficult for more remote relationships.
- No structure information is used, just sequence information.
- Technically, “homology” means that the sequences evolved from the same ancestral sequence—but this is almost always inferred from similarity of sequence, structure, or function, and not directly known.
A *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 model

- Standard solution: ask how much more likely $M$ is than some null hypothesis (represented by a null model).
  \[
  \frac{\Pr(M \mid A)}{\Pr(N \mid A)} = \frac{\Pr(A \mid M)}{\Pr(A \mid N)} \frac{\Pr(M)}{\Pr(N)}.
  \]

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

- $\frac{\Pr(M \mid A)}{\Pr(N \mid A)}$ 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 $(M_1, \ldots, M_n)$
  \[
  \frac{\Pr(M_i \mid A)}{\sum_j \Pr(M_j \mid A)} = \frac{\Pr(A \mid M_i)}{\sum_j \Pr(A \mid M_j)} \frac{\Pr(M_i)}{\sum_j \Pr(M_j)}.
  \]

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

- Null model is a zero-order Markov model, that is, each letter is treated as being independently drawn from the same 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.
Target Model Method for the Fold-recognition Problem

- Find probable homologs of target sequence and make multiple alignment.
- Make secondary structure probability predictions based on multiple alignment.
- Build an HMM based on the multiple alignment and predicted 2ry structure (or just on multiple alignment).
- Score sequences and secondary structure sequences for all proteins that have known structure.
- Select the best-scoring sequence(s) to use as templates.
- If the modeling method is well-chosen, the alignment of the target and template is available as a by-product of the scoring.
**Template Library Method**

- Build a model for each protein in the template library, based on the template sequence (and any homologs you can find). The template library is selected as a subset of the PDB database of publicly released solved structures.

- For the fold-recognition problem, structure information can be used in building these models (though we currently don’t).

- Score target sequence with all models in the library.

- Select the best-scoring model(s) to use as templates.

- Again, the alignment of the target and template may be available as a by-product of the scoring.
Combined SAM-T2K method

- Choose (somehow) the alignment based on the target model or the alignment based on the template model.

- This method for fold-recognition is available (with only SAM-T99 amino-acid target HMMs, not SAM-T2K 2-track target HMMs) on http://www.cse.ucsc.edu/research/compbio/hmm-apps/.

- The library currently has over 5700 templates.
A hidden Markov Model (HMM) is a finite-state machine with a probability for emitting each letter in each state, and with probabilities for making each transition between states.

- Probabilities of letters sum to one for each state.
- Probabilities of transitions out of each state sum to one for that state.
- We also include null states that emit no letters, but have transition probabilities on their out-edges.
Profile Hidden Markov Model

- Circles are null states.
- Squares are *match states*, each of which is paired with a null *delete state*. We call the match-delete pair a *fat state*.
- Each fat state is visited exactly once on every path from Start to End.
- Diamonds are *insert states*, and are used to represent possible extra amino acids that are not found in most of the sequences in the family being modeled.
How is HMM built?

Overview of method for building a target HMM, given a single sequence (or a seed alignment):

1. Construct a profile HMM with one fat state for each letter of sequence (or column of multiple alignment).

2. Find sequences in a large database of protein sequences that score well with $M$. This is the *training set*.

3. Retrain $M$ (using forward-backward algorithm) to re-estimate all probabilities, based on the training set.

4. Make a multiple alignment (using Viterbi algorithm) of all sequences in the training set. The multiple alignment has one alignment column for each fat state of the HMM.

5. Repeat from step 1, with thresholds in step 2 loosened.
Some details of constructing HMMs from alignment

- Do weighting of sequences to reduce the effect of biased sampling in the database.
- Compute $\text{Prob}(a \mid s_i)$ for match states using a Dirichlet mixture regularizer and the weighted counts of the amino acids from the corresponding alignment column.
- Instead of background frequency, or normalizing the relatively few insertion counts, set insertion-state emission probabilities by normalizing the geometric mean of match state frequencies.
- Set transition probabilities based on weighted counts of insertions and deletions in the alignment, plus large pseudocounts based on transitions in many different alignments.
**Multi-track HMMs and secondary structure**

We can also use alignments built using a two-track target HMM:

- Amino-acid track (created with script w0.5 from the SAM-T2K multiple alignment).
- Secondary-structure track (probabilities of \{E, H, L\} or \{E, B, G, H, T, L\} from neural net). The correct letters are defined by STRIDE.
- Can align template (AA+2ry) to target model.
- Haven’t implemented good way to create 2-track template models, nor to align targets to template models.
Human input to alignments

- Alignments that scored well examined in 3D, marking aligned residues and identical residues.
- Look for compactness, clustering of identical residues, striping of identical residues across beta sheets, disulphide bridges, ...
- Tweak alignments to improve placement of gaps.
- 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

$$\frac{\text{Prob} \left( M \mid S \right)}{\text{Prob} \left( M^r \mid S \right)} = \frac{\text{Prob} \left( S \mid M \right)}{\text{Prob} \left( S \mid M^r \right)}.$$

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

Metallothionein Isoform II (4mt2)

Kistrin (1kst)
Composition examples

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:

<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>1aep</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>1vsgA</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>1aep</td>
<td>-12.01</td>
<td>0.78</td>
</tr>
<tr>
<td>2tmaA</td>
<td>1vsgA</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>1aep</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>1vsgA</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>1aep</td>
<td>-12.85</td>
<td>-1.77</td>
</tr>
<tr>
<td>1cii</td>
<td>1vsgA</td>
<td>-10.20</td>
<td>-1.57</td>
</tr>
</tbody>
</table>
Helix examples

Tropomyosin (2tmaA)

Colicin Ia (1cii)

Flavodoxin mutant (1vsgA)
Helix examples

Apolipophorin III (1aep)

Apolipoprotein A-I (1av1A)
Fold recognition results

Fraction of True Positives found vs. False Positives/Query for different methods:

- FSSP psiblast
- FSSP template
- FSSP t2k+0.3*EBGHTL target
- FSSP combined
- FSSP DALI z-score

The graph shows the performance of each method with respect to the number of false positives per query, with the fraction of true positives found increasing as the number of false positives decreases.
Web sites

UCSC bioinformatics (research and degree programs) info:
http://www.cse.ucsc.edu/research/compbio/

SAM tool suite info: http://www.cse.ucsc.edu/research/compbio/sam.html

HMM servers: http://www.cse.ucsc.edu/research/compbio/hmm-apps/


These slides: http://www.cse.ucsc.edu/~karplus/papers/genome9.pdf