HMM for sequence alignment:
profile HMM

Pair HMM
HMM for pairwise sequence alignment, which incorporates affine gap scores.

“Hidden” States
- Match (M)
- Insertion in x (X)
- Insertion in y (Y)

Observation Symbols
- Match (M): \(\{(a,b) \mid a,b \in \sum\}\).
- Insertion in x (X): \(\{(a,-) \mid a \in \sum\}\).
- Insertion in y (Y): \(\{(-,a) \mid a \in \sum\}\).

Pair HMMs

Alignment: a path \(\rightarrow\) a hidden state sequence

Multiple sequence alignment
(Globin family)

Profile model (PSSM)
- A natural probabilistic model for a conserved region would be to specify independent probabilities \(e_i(a)\) of observing nucleotide (amino acid) \(a\) in position \(i\).
- The probability of a new sequence \(x\) according to this model is
\[
P(x \mid M) = \prod_{i=1}^{L} e_i(x_i)
\]
**Profile / PSSM**

- DNA / proteins segments of the same length L;
- Often represented as positional frequency matrix:

```
LTMTRGDIGNYLGLTVETISRLLGRFQKSGML
LTMTRGDIGNYLGLTVETISRLLGRFQKSGML
LTMTRGDIGNYLGLTVETISRLLGRFQKSGML
LAMSRNEIGNYLGLAVETVSRVFSRFQQNELI
LAMSRNEIGNYLGLAVETV SRVFTRFQQNGLI
LPMSRNEIGNYLGLAVETVSRVFTRFQQNGLL
VRMSREEIGNYLGLTLETVSRLFSRFGREGLI
LRMSREEIGSYLGLKLETVSRTLSKFHQEGLI
LPMCRRDIGDYLGLTLETVSRALSQLHTQGIL
LPMSRRDIADYLGLTVETVSRAVSQLHTDGVL
LPMSRQDIADYLGLTIETVSRTFTKLERHGAI
```

**Searching profiles: inference**

- Give a sequence S of length L, compute the likelihood ratio of being generated from this profile vs. from background model:
  \[ R(S|P) = \prod_{i=1}^{L} q(x_i) \]

- Searching motifs in a sequence: sliding window approach

**Components of profile HMMs**

- **Match states**
  - Emission probabilities \( e_M(a) \)

```
M1 M2 M3 M4 M5 M6 M7 M8
```

- **Delete states**
  - No emission prob.
  - Cost of a deletion
    - \( M_i \) to \( D_{i-1}, D_i \) to \( M_{i+1} \)
    - Each \( D_i \) to \( D_{i-1} \) might be different for different \( i \)

**Full structure of profile HMMs**

This is the structure implemented in Hmmer, slightly different from the structure described in the textbook: there is no transition allowed from \( D_{i+1} \) to \( I_i \) or from \( I_i \) to \( D_{i+1} \). As a result, the recursive equation for Viterbi algorithm is different from the one described in the book too.
Deriving HMMs from multiple alignments
• Key idea behind profile HMMs
  – Model representing the consensus for the alignment of sequence from the same family
  – Not the sequence of any particular member

Matching a sequence to a profile HMM (global alignment)
• Viterbi algorithm: pHMM $\theta$ (with L matching states) and a query sequence $x_1x_2...x_N$

Viterbi algorithm: trace back

Sequence conservation: entropy of the emission probability distributions
High entropy indicates non-conserved positions (note: here negative entropy is plotted.)
Matching a sequence to a profile HMM (global alignment)

- Forward: pHMM θ (with L matching states) and a query sequence Xᵢ₋₁: X₉

\[
F^θ_i(x_i) = \sum_{j=1}^{L} \left( F^θ_(i-1)(x_{i-1}) a_{x_{i-1}j} P^θ_j(x_i \mid a_{x_i}) \right)
\]

for i = 1, ..., N

Initialization: \( F^θ_0(x_0) = 1 \)

Termination: \( F^θ_N(x_N) = 0 \)

Probabilities: \( F^θ_i(x_i) \)

Note: this is slightly different from the textbook; no transition from \( D \) to \( I \) or \( D \) to \( D \).

Example: Viterbi algorithm

Probability matrices:

<table>
<thead>
<tr>
<th>Transition probability</th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-D</td>
<td>.0</td>
<td>.5</td>
<td>.0</td>
<td>.5</td>
</tr>
<tr>
<td>D-M</td>
<td>.0</td>
<td>.5</td>
<td>.0</td>
<td>.5</td>
</tr>
<tr>
<td>I-I</td>
<td>.5</td>
<td>.5</td>
<td>.7</td>
<td>.5</td>
</tr>
<tr>
<td>M-D</td>
<td>.2</td>
<td>0</td>
<td>0</td>
<td>.9</td>
</tr>
<tr>
<td>M-M</td>
<td>.8</td>
<td>1</td>
<td>.4</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td>.2</td>
<td>.2</td>
<td>.0</td>
<td>.2</td>
</tr>
<tr>
<td>G</td>
<td>.3</td>
<td>.3</td>
<td>0</td>
<td>.3</td>
</tr>
<tr>
<td>C</td>
<td>.2</td>
<td>.2</td>
<td>.9</td>
<td>.2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

Example: Viterbi algorithm

Probability matrices:

<table>
<thead>
<tr>
<th>Transition probability</th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-D</td>
<td>.0</td>
<td>.5</td>
<td>.0</td>
<td>.5</td>
</tr>
<tr>
<td>D-M</td>
<td>.0</td>
<td>.5</td>
<td>.0</td>
<td>.5</td>
</tr>
<tr>
<td>I-I</td>
<td>.5</td>
<td>.5</td>
<td>.7</td>
<td>.5</td>
</tr>
<tr>
<td>M-D</td>
<td>.2</td>
<td>0</td>
<td>0</td>
<td>.9</td>
</tr>
<tr>
<td>M-M</td>
<td>.8</td>
<td>1</td>
<td>.4</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td>.2</td>
<td>.2</td>
<td>.0</td>
<td>.2</td>
</tr>
<tr>
<td>G</td>
<td>.3</td>
<td>.3</td>
<td>0</td>
<td>.3</td>
</tr>
<tr>
<td>C</td>
<td>.2</td>
<td>.2</td>
<td>.9</td>
<td>.2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

Example: Viterbi algorithm

\[
V^θ_i(x_i) = a_{x_{i-1}j} P^θ_j(x_i \mid a_{x_i})
\]

Example: Viterbi algorithm

\[
V^θ_i(x_i) = a_{x_{i-1}j} P^θ_j(x_i \mid a_{x_i})
\]

Example: Viterbi algorithm

\[
V^θ_i(x_i) = a_{x_{i-1}j} P^θ_j(x_i \mid a_{x_i})
\]
Example: Viterbi algorithm

Transition
\[
\begin{array}{cccc}
A & 1 & 1 & 3 \\
G & 0 & 0 & 4 \\
C & 1 & 1 & 1 \\
T & 0 & 0 & 2
\end{array}
\]

Emission from M
\[
\begin{array}{cccc}
A & 0 & 0 & 0 \\
G & 0 & 0 & 0 \\
C & 0 & 0 & 0 \\
T & 0 & 0 & 0
\end{array}
\]

Emission from I
\[
\begin{array}{cccc}
A & 2 & 2 & 1 \\
G & 2 & 2 & 1 \\
C & 2 & 2 & 1 \\
T & 0 & 0 & 1
\end{array}
\]

Transition probabilities
\[
\begin{array}{cccc}
0-1 & 0 & 0 & 1 \\
0-0 & 0 & 0 & 0 \\
1-1 & 1 & 1 & 1 \\
1-0 & 0 & 0 & 0
\end{array}
\]

Traceback

Query: X:AGG (N=3)

Searching with profile HMMs

- Main usage of profile HMMs
  - Detecting potential sequences in a family
  - Core algorithm: matching a sequence to a profile HMM
    - Viterbi algorithm or forward algorithm
    - Comparing the resulting probability with random model (R): log-odd score
      \[ P(x | R) = \prod q_i \]
      where \( q_i \) is the frequency of observing \( x_i \).

Matching a sequence to a profile HMM (global alignment)

**Viterbi algorithm**

\[
V^i_j(i) = \max_{i=1,...L, j=1,...N} \left[ \log V^i_j(i-1) + \log a_{i,i} + \log q_i \right]
\]

**Initialization:**

\[ V^1_j(0) = \max_{j=1,...N} \left[ \log q_j \right] \]

**Termination:**

\[ V = \max_i \left[ V^i_j(N) V^i_j(N) \right] \]

for \( i=1,...,L \), \( j=1,...,N \)

Matching a sequence to a profile HMM (global alignment)

**Forward algorithm**

\[
\begin{align*}
P^0(0) &= -\log q_0 \\
P^1(0) &= -\log q_1 \\
P^1(0) &= -\log q_1 \\
P^{i+1}(0) &= -\log q_i \\
P^{i+1}(0) &= -\log q_i \\
\end{align*}
\]

**Initialization:**

\[ P^0(0) = -\log q_0 \]

**Termination:**

\[ P = \max_i \left[ \exp(P^i_j(N)) \exp(P^i_j(N)) \right] \]

Significance of HMM alignment

- The log-odd score of local Viterbi alignment (V) between a random sequence and a profile HMM follows a Gumbel (type I EVD) distribution
  \[ P(V > v) = 1 - \exp \left[ -e^v - \lambda v \right] \]
- With \( \approx 200 \) Viterbi, the location parameter \( \mu \) can be accurately estimated;
- \( \lambda \approx \log(2), z \) is the base of the log-odd score, e.g., \( z=2 \) when the sequence length approaches infinite
- The length effect can be corrected by \( \lambda = \log(2) + \frac{\Delta T}{\Delta x} \)
  - Where, \( N \) is the length, and \( h \) is the average relative entropy per match state in the pHMM;
  - For typical Pfam models, \( N \approx 140, h \approx 1.8, \lambda = \log(2) + 0.0057 \), a small correction.

Variants for non-global alignments

- Local alignment (Smith-Waterman type)
  - Emission prob. in flanking states use background values \( q_i \).
  - Looping prob. close to 1, e.g. \( 1 - \eta \) for some small \( \eta \).
Variants for non-global alignments

- Overlap (also called glocal or fit alignment)
  - The loop probability of the first and last insert states is much higher than the other insert states
  - When expecting to find either present as a whole or absent (e.g., of a protein domain within a protein)
  - Transition to first delete state allows missing first residue

Variants for non-global alignments

- Repeat alignments
  - Transition from right flanking state back to random model
  - Can find multiple matching segments in query string

Optimal model construction: different ways of marking columns

(a) Multiple alignment:

(b) Profile-HMM architecture:

Optimal model construction

- MAP (match-insert assignment)
  - Recursive calculation of a number $S_j$
    - $S_j$: log prob. of the optimal model for alignment up to and including column $j$, assuming $j$ is marked.
    - $S_j$ is calculated from $S_i$ and summed log prob. between $i$ and $j$.
    - $T_{ij}$: summed log prob. of all the state transitions between marked $i$ and $j$.
      - $T_{ij} = \sum_{x,y \in \{M,D,I\}} c_{xy} \log a_{xy}$

Optimal model construction

- Algorithm: MAP model construction
  - Initialization:
    - $S_0 = 0, M_{L+1} = 0$
  - Recurrence: for $j = 1, \ldots, L+1$:
    - $S_j = \max_{i \leq j} (S_i + T_{ij} + M_j + I_{i+1,j+1} + \lambda)$
    - $\sigma_j = \arg \max (S_i + T_{ij} + M_j + I_{i+1,j+1} + \lambda)$
  - Traceback: from $j = \sigma_{j+1}$, while $\sigma_j > 0$
    - Mark column $j$ as a match column
    - $j \leftarrow \sigma_j$

Weighting training sequences

- Input sequences are random?
- “Assumption: all examples are independent samples” might be incorrect

Solutions

- Weight sequences based on similarity: highly similar pair of training sequences receive lower weights
Multiple sequence alignment (MSA) by training profile HMM

- Sequence profiles can be represented as probabilistic models like profile HMMs.
  - ML methods for building (training) profile HMM are based on multiple sequence alignment.
  - Profile HMMs can also be trained from initially unaligned sequences using the Baum-Welch-like EM algorithm.
    - Simultaneously aligning multiple sequences and building the profile HMM from the multiple alignment.

Multiple alignment with a known profile HMM

- A step backward: to derive a multiple alignment from a known profile HMM model.
  - e.g., to align many sequences from the same family based on the HMM model built from the (seed) multiple alignment of a small representative set of sequences in the family.
- It just requires calculating a Viterbi alignment for each individual sequence.
  - Match a sequence to a profile HMM: Viterbi algorithm.
  - Residues aligned to the same match state in the profile HMM should be aligned in the same columns.
  - Given a preliminary alignment, HMM can align additional sequences.

Multiple alignment with a known profile HMM

- Comparing with other MSA program.
  - Profile HMM does not align inserts whereas other MSA algorithms align the whole sequences.

<table>
<thead>
<tr>
<th>Position</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>insert</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>P</td>
<td>H</td>
<td>F</td>
<td>-</td>
<td>D</td>
<td>L</td>
<td>S</td>
<td>H</td>
<td>O</td>
<td>S</td>
<td>A</td>
<td>G</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>D</td>
<td>L</td>
<td>L</td>
<td>T</td>
<td>P</td>
<td>F</td>
<td>A</td>
<td>M</td>
<td>G</td>
</tr>
<tr>
<td>F</td>
<td>R</td>
<td>F</td>
<td>K</td>
<td>H</td>
<td>L</td>
<td>T</td>
<td>A</td>
<td>A</td>
<td>R</td>
<td>L</td>
<td>K</td>
<td>E</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>G</td>
<td>A</td>
<td>C</td>
<td>K</td>
<td>L</td>
<td>K</td>
<td>E</td>
<td>K</td>
<td>G</td>
<td>T</td>
<td>A</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>C</td>
<td>L</td>
<td>T</td>
<td>A</td>
<td>L</td>
<td>K</td>
<td>Y</td>
<td>S</td>
<td>A</td>
</tr>
<tr>
<td>F</td>
<td>S</td>
<td>F</td>
<td>L</td>
<td>K</td>
<td>G</td>
<td>T</td>
<td>S</td>
<td>Y</td>
<td>F</td>
<td>Q</td>
<td>N</td>
<td>P</td>
</tr>
<tr>
<td>F</td>
<td>G</td>
<td>F</td>
<td>S</td>
<td>G</td>
<td>A</td>
<td>S</td>
<td>-</td>
<td>-</td>
<td>D</td>
<td>P</td>
<td>G</td>
<td>-</td>
</tr>
</tbody>
</table>

Training profile HMM from unaligned sequences

- Simultaneously aligning multiple sequences and building the profile HMM from the multiple alignment.
  - Initialization: choose the length of the profile HMM and initialize parameters of the model.
  - MSA: align all sequences to the final model using the Viterbi algorithm and build a multiple alignment as described in the previous section.
  - Training: estimate the model using the Baum-Welch algorithm.
  - Iterating until the model (and the MSA) converges.

Profile HMM training from unaligned sequences

- Initial Model
  - The only decision that must be made in choosing an initial structure for Baum-Welch estimation is the length of the model M.
  - A commonly used rule is to set M be the average length of the training sequence.
  - We need some randomness in initial parameters to avoid local maxima.

Multiple alignment by profile HMM training

- Avoiding Local maxima
  - Baum-Welch algorithm is guaranteed to find a LOCAL maxima.
    - Models are usually quite long and there are many opportunities to get stuck in a wrong solution.
  - Solution
    - Start many times from different initial models.
    - Use some form of stochastic search algorithm, e.g. simulated annealing.
Multiple alignment by profile HMM training--Model surgery

- We can modify the model after (or during) training a model by manually checking the alignment produced from the model.
  - Some of the match states are redundant
  - Some insert states absorb too many sequences
- Model surgery
  - If a match state is used by less than ½ of training sequences, delete its module (match-insert-delete states)
  - If more than ½ of training sequences use a certain insert state, expand it into n new modules, where n is the average length of insertions
  - ad hoc, but works well