Applications of HMMs

HMM applications

- Gene finding
- Pairwise alignment (pair HMMs)
- Characterizing protein families (profile HMMs)
- Predicting membrane proteins, and membrane protein topology

Gene finding with HMMs

Using the gene finder

- Apply the gene finder to both the top strand and bottom strand
- Training using annotated genes
A simple eukaryotic gene finder

Pair HMMs and Sequence Alignment

Global alignment with affine gaps revisited

Suppose we're aligning sequences $x, y$ of lengths $m$ and $n$.
Dynamic Programming:

$M(i, j)$: Optimal alignment of $x_1 \ldots x_i$ to $y_1 \ldots y_j$ ending in a match

$I_x(i, j)$: Optimal alignment of $x_1 \ldots x_i$ to $y_1 \ldots y_j$ ending in a gap in $x$

$I_y(i, j)$: Optimal alignment of $x_1 \ldots x_i$ to $y_1 \ldots y_j$ ending in a gap in $y$

Alignment with affine gaps

Initialization:

- $M(0, 0) = 0$
- $M(i, 0) = M(0, j) = -\infty$, for $i, j > 0$
- $I_x(i, 0) = - (d + i \cdot e)$
- $I_y(0, j) = - (d + j \cdot e)$

Iteration:

- $M(i-1, j-1) + s(x_i, y_j)$
- $\max \{ I_x(i-1, j-1), I_y(i-1, j) - e, I_x(i, j-1) - d \}$
- $\max \{ I_y(i-1, j-1), I_x(i, j) - e, I_y(i, j-1) - d \}$

Termination:

Optimal alignment given by $\max \{ M(m, n), I_x(m, n), I_y(m, n) \}$

A state model for alignment

Alignments correspond
1-to-1 with sequences
of states $M, X, Y$

Scoring transitions

Alignments correspond
1-to-1 with sequences
of states $M, X, Y$
Probabilistic interpretation of an alignment

An alignment is a hypothesis that the two sequences are related by evolution

**Goal:**
- Produce the most likely alignment
- Compute the likelihood that the sequences are indeed related

A Pair HMM for alignments

Model generates two sequences simultaneously

**Emission probabilities:**
- Match/Mismatch state $M$
  - Emitted pair of symbols with distribution $P(x, y)$ that reflects substitution frequencies between pairs of amino acids
- Insertion states $X, Y$
  - Emit symbol + gap according to $P(x), P(y)$ that reflects amino acid frequencies of $x$ and $y$

Transition probabilities:
- $\delta$: set so that $1/2\delta$ is avg. length of match region
- $\varepsilon$: set so that $1/(1-\varepsilon)$ is avg. length of a gap

Viterbi for pair HMMs

**Initialization:**
- $v^M(0,0) = 1$, $v^*(i,0), v^*(0,j) = 0$

**Reurrence:**
- $v^M(i,j) = P_{x_0}, \max \left\{ (1 - 2\delta)P(x_i, y_j), (1 - \varepsilon)P(x_i), \varepsilon P(y_j) \right\}$
- $v^X(i,j) = q_x, \max \left\{ \delta P_M(x_i-1, y_j), \varepsilon P_M(x_i, y_j) \right\}$
- $v^Y(i,j) = q_y, \max \left\{ \delta P_M(x_i, y_j-1), \varepsilon P_M(x_i, y_j) \right\}$

**Termination:**
- $v^F = \max(v^M(m,n), v^X(m,n), v^Y(m,n))$

A model for unaligned sequences

The two sequences are independently generated from one another

$$P(x, y \mid R) = P(x_1) \cdots P(x_m) P(y_1) \cdots P(y_n) = \prod_i P(x_i) \prod_j P(y_j)$$

ALIGNMENT vs. RANDOM hypothesis

Every pair of letters contributes:
- $M$
  - $(1-2\delta)P(x_i, y_j)$ when matched
  - $\varepsilon P(x_i)$ or $\varepsilon P(y_j)$ when gapped
- $R$
  - $P(x_i)P(y_j)$ in random model

ALIGNMENT vs. RANDOM hypothesis

To compare the models we consider the log odds ratio under the two models, which can be expressed as a sum of terms:

- $s(x_i, y_j) = \log \frac{P(x_i, y_j)}{P(x_i) P(y_j)}$
- $d = -\log \frac{\delta (1-\varepsilon)}{(1-2\delta) P_M(x_i)}$
- $e = -\log \frac{\varepsilon P_M(x_i)}{P(x_i)}$
Log-odds Viterbi

- The Viterbi algorithm for Pair HMMs corresponds exactly to global alignment DP with affine gaps

\[
V_M(i, j) = \max \left( V_M(i-1, j-1), V_X(i-1, j-1), V_Y(i-1, j-1) \right) + s(x_i, y_j)
\]

\[
V_X(i, j) = \max \left( V_M(i-1, j) - d, V_X(i-1, j) - e \right)
\]

\[
V_Y(i, j) = \max \left( V_M(i-1, j) - d, V_Y(i-1, j) - e \right)
\]

Comments

- The pair HMM formulation provides a probabilistic interpretation for the pairwise alignment algorithm, and allows assigning values to parameters from first principles.
- Viterbi provides the probability that two sequences are related by a particular alignment according to the HMM.
- Using the forward algorithm we can compute the probability that the two sequences are related by any alignment:

\[
P(x; y) = \sum_{\text{alignments } \pi} P(x; y; \pi)
\]

Probability that two residues are aligned

- Notation: \( X_i \triangleq y_j \) means \( x_i \) and \( y_j \) are aligned

\[
P(x_i \triangleq y_j | x; y) = \frac{P(x, y, x_i \triangleq y_j)}{P(x, y)}
\]

\[
P(x, y, x_i \triangleq y_j) = P(x_1, ..., x_{i-1}, x_i \triangleq y_j, y_{j-1}, ..., y_m) = P(x_1, ..., x_{i-1}, y_{j-1}, ..., y_m)
\]

\[
f^M(i, j) = f^X(i, j) = f^Y(i, j)
\]

\( f^M(i, j) \) and \( f^B(i, j) \) are the forward and backward probs. that correspond to a match state

Algorithm: Forward calculation for pair HMMs

Initialisation:

\[
f^M(0, 0) = 1, f^X(0, 0) = f^Y(0, 0) = 0.
\]

All \( f^M(i, -1), f^X(-1, j), f^Y(-1, j) \) are set to 0.

Recursion: \( i = 0, ..., n, j = 0, ..., m \) except \( (0, 0) \);

\[
f^M(i, j) = p_{\text{match}} [1 - 2(1-\tau)f^M(i-1, j-1) + (1-\tau)(1-\epsilon)f^X(i-1, j-1) + f^Y(i-1, j-1)];
\]

\[
f^X(i, j) = q_a [\epsilon f^M(i-1, j) + \tau f^X(i-1, j)];
\]

\[
f^Y(i, j) = q_b [\epsilon f^M(i-1, j) + \tau f^Y(i-1, j)];
\]

Termination:

\[
f^M(n, m) = \epsilon [f^M(n, m) + f^X(n, m) + f^Y(n, m)].
\]

Pair HMM for local alignment

Describing protein families for prediction of protein structure and function

- Protein sequence and structure
- Protein classification
Primary Structure: Sequence

- The primary structure of a protein is its amino acid sequence.

Secondary Structure: α, β, & loops

- α helices and β sheets are stabilized by hydrogen bonds between backbone oxygen and hydrogen atoms.

Tertiary Structure: the Fold of a protein

- 3d structure is minimum free-energy conformation.

Structure Determines Function

- 3d structure is minimum free-energy conformation.
- How can we determine structure?
  - Experimental methods
  - Computational predictions

A useful mnemonic for the hydrophobic amino acids is "FAMILY VW"
Growth of the Protein Data Bank (PDB)

The number of folds in nature

Classifying folds

- Despite the large growth in the number of solved protein structures, the number of new folds identified is small
- Classifying proteins into folds
  - Generate overview of structure types
  - Detect similarities between proteins
  - Help predict 3D structure of new protein sequences

Structure Classification Databases

- SCOP
  - Manual classification (A. Murzin)
  - scop.berkeley.edu
- CATH
  - Semi manual classification (C. Orengo)
  - www.biochem.ucl.ac.uk/bsm/cath
- FSSP
  - Automatic classification (L. Holm)
  - www.ebi.ac.uk/dali/fssp/fssp.html

Major classes in SCOP

- Classes
  - All α proteins
  - All β proteins
  - α and β proteins (α/β)
  - Membrane and cell surface proteins
  - Small proteins

All α: Hemoglobin (1bab)
Protein Superfamily

- Consists of proteins which are (remotely) evolutionarily related
  - Sequence similarity low
  - Related function
  - Similar in structure
- Relationships between members of a superfamily hard to recognize from sequence alone

Fold

- Proteins in the same fold share the same general architecture, and have the same major secondary structures in the same topological arrangement
- No evolutionary relationship implied

All β: Immunoglobulin (8fab)

α/β: Triosephosphate isomerase (1hti)

α+β: Lysozyme (1jsf)
Protein Classification

- Given a new protein, can we assign it to its position within an existing protein hierarchy?

Methods
- BLAST / PsiBLAST
- Profile HMMs
- Supervised machine learning

PSI-BLAST

- Given a sequence query x, and database D
  1. Find all pairwise alignments of x to sequences in D
  2. Collect all matches of x to y with some minimum significance
  3. Construct a profile M
     - Each sequence y is given a weight so that many similar sequences cannot have much influence on a position (Henikoff & Henikoff 1994)
  4. Using M, search D for more matches
  5. Iterate 1–4 until convergence

Profiles & Profile HMMs

- Psi-BLAST builds a profile
- Profile HMM: a more elaborate version of a profile
  - Intuitively, a profile that models insertions and deletions

PWMs as HMMs

- A PWM can be described as a very simple HMM where the emissions associated with M_i correspond to the i-th column of the PWM

Insertions

- Insertion: a part of the sequence x that is not modeled by the match states
- The loop allows for insertions of multiple letters
- An insert state emits symbols with the background distribution
Insertions

- Log odds contribution of an insertion of length $k$:
  \[ \log a_{M_1} + \log a_{I_{j+1}} + (k - 1) \log a_{I_j} \]
- Looks like an affine gap penalty

Deletions

- Deletion states: silent states
- Cost of deletion of a sequence of letters: sum of log transition probabilities
- We could model deletions by allowing all forward transitions from match states: lots of transitions

Profile HMMs

- Each $M$ state has a position-specific pre-computed substitution table
- Each $I$ state has position-specific gap penalties (and in principle can have its own emission distributions)
- Each $D$ allows to skip its corresponding match state
  - In principle, $D-D$ transitions can also be customized per position

Estimating Profile HMMs from MSAs

- Columns that have less than a given number of gaps are used as match state of the HMM (starred columns)
- Amino acid frequencies of each column used to set emission probabilities for the corresponding match state

\[
a_{ij} = \sum A_{ij} e_{j}(a) \frac{E_{i}(a)}{\sum_{a} E_{i}(a)}
\]
Profile HMM for the globins example

Figure 5.4 from Durbin et al.

What you can do with a profile HMM

- **Classification**: given a protein, we can compute its probability under the model, or compare it to a random model (using a log-odds-ratio). Use forward/backward
- **Alignment**: Use the Viterbi algorithm to align a sequence to the HMM.
- **Learning**: set model parameters using a given MSA, or from scratch using EM.

Alignment of a protein to a profile HMM

To align sequence $x_1 \ldots x_n$ to a profile HMM:

We will find the most likely alignment with the Viterbi DP algorithm

- Define
  - $V_j^M(i)$: score of best alignment of $x_1 \ldots x_i$ to the HMM ending in $x_i$ being emitted from $M_j$
  - $V_j^I(i)$: score of best alignment of $x_1 \ldots x_i$ to the HMM ending in $x_i$ being emitted from $I_j$
  - $V_j^D(i)$: score of best alignment of $x_1 \ldots x_i$ to the HMM ending in $D_j$ ($x_i$ is the last character emitted before $D_j$)

- Denote by $q_a$ the frequency of amino acid $a$ in a ‘random’ protein

Log-odds Viterbi (termination)

Label the END state as $M_{n+1}$

Then

$$V_{n+1}^M(n) = \max \begin{cases} V_{n+1}^M(n) + \log(q_{M_M} q_{M_{n+1}}) \\ V_n^I(n) + \log(q_{I_{n+1}} a_{M_{n+1}}) \\ V_n^D(n) + \log(q_{D_{n+1}} a_{M_{n+1}}) \end{cases}$$

Gives the log-odds for the maximum scoring path.

By backtracing in the usual manner, we get the Viterbi Path.

Log-odds Viterbi

Define $V_0^M(0) = V_0^I(0) = V_0^D(0) = 0$

For all other allowable $i$

$$V_j^M(i) = \log(q_{M_j}(x_i)) + \max \begin{cases} V_j^M(i-1) + \log(q_{M_{n+1}} a_{M_{n+1}}) \\ V_j^I(i-1) + \log(q_{I_{n+1}} a_{M_{n+1}}) \\ V_j^D(i-1) + \log(q_{D_{n+1}} a_{M_{n+1}}) \end{cases}$$

$$V_j^I(i) = \log(q_{I_j}(x_i)) + \max \begin{cases} V_j^M(i-1) + \log(q_{M_{n+1}} a_{I_{n+1}}) \\ V_j^I(i-1) + \log(q_{I_{n+1}} a_{I_{n+1}}) \\ V_j^D(i-1) + \log(q_{D_{n+1}} a_{I_{n+1}}) \end{cases}$$

$$V_j^D(i) = \max \begin{cases} V_j^M(i-1) + \log(q_{M_{n+1}} a_{D_{n+1}}) \\ V_j^I(i-1) + \log(q_{I_{n+1}} a_{D_{n+1}}) \\ V_j^D(i-1) + \log(q_{D_{n+1}} a_{D_{n+1}}) \end{cases}$$

MSA using a profile HMM

- The optimal path assigns each letter in the sequence to one of the types of states: Match or Insert.
- If we align two or more sequences (independently) to a profile HMM, we end up with a multiple alignment of the sequences.
- Sequence 1: ATGC
  - TGCC
- Sequence 2: TGCC
- Sequence 3: ATCG
How to build a profile HMM

**Classification with Profile HMMs**

- **How generative models work**
  - Training examples: sequences known to be members of family
  - Tuning parameters with *a priori* knowledge
  - Idea: The sequences from the family (hopefully) yield a higher probability than sequences outside the family

- Log-likelihood ratio as score
  \[
  \log \frac{P(H_1|X)}{P(H_0|X)}
  \]

---

**Classification using Z score**

- The examples of a protein family are not a good sample from the sequences that belong to the family. Typically there are sequences that are very closely related. They can give the mistaken impression regarding the pattern of conservation in the family.
- Solution: assign weights to each sequence. Less weight to related sequences.
### Profile HMMs for local alignment

![Diagram](image)

### PFAM: profile HMMs for protein families

**PFAM method:**
- Use Blast to cluster a protein database into families of related proteins
- Construct a multiple alignment for each protein family.
- Construct a profile HMM from each alignment

Given a protein of unknown origin:
- Align the target sequence against each HMM to find the best fit

### PFAM

- **Pfam describes protein domains**
- Each protein domain family in Pfam has:
  - **Seed alignment**: manually verified multiple alignment of a representative set of sequences.
  - **HMM**: built from the seed alignment for further database searches.
  - **Full alignment**: generated automatically from the HMM.
- The distinction between seed and full alignments facilitates Pfam updates.
  - **Seed alignments** are a stable resource (manually inspected by an expert).
  - **HMMs** can be updated with new protein sequences.

### Generative Models

![Diagram](image)
Digression: Discriminative Models

Rather than model each class (red/green), construct a decision boundary between the classes.

Support Vector Machines (SVM) learn "large margin" decision boundaries.

Decision Rule: red: $w^T x > 0$

Margin

- For given word size $k$, and mismatch tolerance $m$, define
  $$K(x, y) = \# \text{distinct } k\text{-long word occurrences with } \leq m \text{ mismatches}$$
- SVM can be learned by supplying this kernel function

Let $k = 3, m = 1$

$$K(X, Y) = 4$$

- For given word size $k$, and mismatch tolerance $m$, define
  $$K(x, y) = \# \text{distinct } k\text{-long word occurrences with } \leq m \text{ mismatches}$$
- SVM can be learned by supplying this kernel function

Let $k = 3, m = 1$

$$K(X, Y) = 4$$

Benchmarks

- HMMer – a free profile HMM software
  - http://hmmer.janelia.org/
- SAM – another free profile HMM software
- PFAM – database of alignments and HMMs for protein families and domains
  - http://www.sanger.ac.uk/Software/Pfam/
- SCOP – a structural classification of proteins
  - http://scop.berkeley.edu/

Prediction of transmembrane proteins

Figure 3: Overall structure of bacteriorhodopsin as deduced from electron diffraction analyses of two-dimensional crystals of the protein in the bacterial membrane. The seven transmembrane $\alpha$ helices are labeled A-G. The retinal pigment is covalently attached to tyrosine 216 in helix G. The approximate position of the protein in the phospholipid bilayer is indicated. [Adapted from R. Henderson et al., 1990, J. Mol. Biol. 213:899.]

http://www.rcsb.org/pdb/structure-viewer/3br}
Transmembrane proteins

Properties of transmembrane proteins:
- hydrophobic α-helices
- connected by interfacial loops

Sonnhammer et al., ISMB, 6:175-182, 1998: TMHMM

- There are 3 main locations of a residue:
  - TM helix core (in hydrophobic tail of membrane)
  - TM helix cap (in head of membrane)
  - loops
    - cytoplasmic vs
    - non-cytoplasmic side of the helix core
    - non-cytoplasmic (short) vs
    - non-cytoplasmic (long)

Architecture of TMHMM