Modeling Biological Sequence and Hidden Markov Models
Challenges in Computational Biology

1. Gene Finding
2. Genome Assembly
3. Regulatory motif discovery
4. Comparative Genomics
5. Evolutionary Theory
6. Database lookup
7. RNA folding
8. Gene expression analysis
9. Protein network analysis
10. Cluster discovery
11. Gibbs sampling
12. Emerging network properties
13. Regulatory network inference
What have we learned so far?

- **Motif searching and counting**
  - Brute-force algorithm
  - Content-based indexing

- **Sequence alignment**
  - Dynamic programming, duality path ↔ alignment
  - Global / local alignment, general gap penalties

- **String comparison**
  - Exact string match, semi-numerical matching

- **Rapid database search**
  - Exact matching: Hashing, BLAST
  - Inexact matching: neighborhood search, projections

- **Problem set 1**
So, you find a new piece of DNA…
What do you do?

...GTACTCACC GGTTACAGGATTATGGGTTACAGGTAACCGTT...

• Align it to things we know about
• Align it to things we don’t know about
• Stare at it
  – Non-standard nucleotide composition?
  – Interesting k-mer frequencies?
  – Recurring patterns?
• Model it
  – Make some hypotheses about it
  – Build a ‘generative model’ to describe it
  – Find sequences of similar type
This week: Modeling biological sequences
(a.k.a. What to do with a huge chunk of DNA)

- Ability to emit DNA sequences of a certain **type**
  - Not exact alignment to previously known gene
  - Preserving ‘properties’ of **type**, not identical sequence
- Ability to recognize DNA sequences of a certain type (state)
  - What (hidden) state is most likely to have generated observations
  - Find set of states and transitions that generated a long sequence
- Ability to learn distinguishing characteristics of each state
  - Training our generative models on large datasets
  - Learn to classify unlabelled data
Computational Tool: Hidden Markov Models (HMMs)

• Today:
  – Simple example to introduce HMMs

• Computational framework:
  – Define Markov Chains
  – Define Hidden Markov Models

• Evaluation
  – Calculating forward probabilities

• First algorithm
  – Viterbi
First application of HMMs
The dishonest casino
Example: The Dishonest Casino

A casino has two dice:
- Fair die
  \[ P(1) = P(2) = P(3) = P(5) = P(6) = \frac{1}{6} \]
- Loaded die
  \[ P(1) = P(2) = P(3) = P(5) = \frac{1}{10} \\
  P(6) = \frac{1}{2} \]

Casino player switches between fair and loaded die on average once every 20 turns

**Game:**
1. You bet $1
2. You roll (always with a fair die)
3. Casino player rolls (maybe with fair die, maybe with loaded die)
4. Highest number wins $2
The dishonest casino model

**FAIR**
- P(1|F) = 1/6
- P(2|F) = 1/6
- P(3|F) = 1/6
- P(4|F) = 1/6
- P(5|F) = 1/6
- P(6|F) = 1/6

**LOADED**
- P(1|L) = 1/10
- P(2|L) = 1/10
- P(3|L) = 1/10
- P(4|L) = 1/10
- P(5|L) = 1/10
- P(6|L) = 1/2
Running the model: Probability of a sequence

What is the likelihood of

\( \pi = \text{Fair, Fair, Fair, Fair, Fair, Fair, Fair, Fair, Fair, Fair} \)

and rolls

\( x = 1, 2, 1, 5, 6, 2, 1, 6, 2, 4 \)

\[
p = \frac{1}{2} \times P(1 \mid \text{Fair}) P(\text{Fair}_{i+1} \mid \text{Fair}_i) P(2 \mid \text{Fair}) P(\text{Fair} \mid \text{Fair}) \ldots P(4 \mid \text{Fair})
\]

\[
= \frac{1}{2} \times (\frac{1}{6})^{10} \times (0.95)^9
\]

\[
= 0.5 \times 10^{-9}
\]

Why is \( p \) so small?
Running the model: Probability of a sequence

What is the likelihood of

\( \pi = \text{Load, Load, Load, Load, Load, Load, Load, Load, Load, Load, Loaded} \)

and rolls

\[ x = 1, 2, 1, 5, 6, 2, 1, 6, 2, 4 \]

emission transition emission transition emission transition emission

\[ p = \frac{1}{2} \times P(1 \mid \text{Load}) \times P(\text{Load}_{i+1} \mid \text{Load}_i) \times P(2 \mid \text{Load}) \times P(\text{Load} \mid \text{Load}) \times \ldots \times P(4 \mid \text{Fair}) \]

\[ = \frac{1}{2} \times \left(\frac{1}{10}\right)^8 \times \left(\frac{1}{2}\right)^2 \times (0.95)^9 \]

\[ = 7.9 \times 10^{-10} \]

Compare the two!
Comparing the two models

Two sequence paths:

- \( P( x, \text{all-Fair} ) = 0.5 \times 10^{-9} \) (very small)
- \( P( x, \text{all-Loaded} ) = 7.9 \times 10^{-10} \) (very very small)

Likelihood ratio:

- \( P( x, \text{all-Fair} ) \) is 6.59 times more likely than \( P( x, \text{all-Loaded} ) \)

It is 6.59 times more likely that the die is fair all the way, than loaded all the way.
Example: the dishonest casino

Let the sequence of rolls be:
\[ x = 1, 6, 6, 5, 6, 2, 6, 6, 3, 6 \]

Now, what is the likelihood \( \pi = F, F, \ldots, F? \)
\[
\frac{1}{2} \times (1/6)^{10} \times (0.95)^9 = 0.5 \times 10^{-9}, \text{ same as before}
\]

What is the likelihood \( \pi = L, L, \ldots, L? \)
\[
\frac{1}{2} \times (1/10)^4 \times (1/2)^6 \times (0.95)^9 = 0.5 \times 10^{-7}
\]

So, it is 100 times more likely the die is loaded

Model evaluation
What about partial runs and die switching

What is the likelihood of

\( \pi = \text{Fair, Fair, Fair, Fair, Load, Load, Load, Load, Fair, Fair} \)

and rolls

\( x = 1, 2, 1, 5, 6, 2, 1, 6, 2, 4 \)

\[
p = \frac{1}{2} \times P(1 \mid \text{Fair}) P(\text{Fair}_{i+1} \mid \text{Fair}_i) P(2 \mid \text{Fair}) P(\text{Fair} \mid \text{Fair}) \ldots P(4 \mid \text{Fair}) \\
= \frac{1}{2} \times (1/10)^2 \times (1/2)^2 \times (1/6)^5 \times (0.95)^7 \times (0.05)^2 \\
= 2.8 \times 10^{-10}
\]
Question # 1 – Evaluation

GIVEN

A sequence of rolls by the casino player

124552646214614613613666166466163661636163616515615115146123562344

QUESTION

How likely is this sequence, given our model of how the casino works?

This is the EVALUATION problem in HMMs
Question # 2 – Decoding

GIVEN

A sequence of rolls by the casino player

1245526462146146136136661664661636616366163616515615115146123562344

QUESTION

What portion of the sequence was generated with the fair die, and what portion with the loaded die?

This is the **DECODING** question in HMMs
Question # 3 – Learning

GIVEN

A sequence of rolls by the casino player

1245526462146146136136661664661636616366163616515615115146123562344

QUESTION

How “loaded” is the loaded die? How “fair” is the fair die? How often does the casino player change from fair to loaded, and back?

This is the LEARNING question in HMMs
**Markov Chain**

**Definition:** A *Markov chain* is a triplet \((Q, p, A)\), where:

- \(Q\) is a finite set of states. Each state corresponds to a symbol in the alphabet \(\Sigma\).
- \(p\) is the initial state probabilities.
- \(A\) is the state transition probabilities, denoted by \(a_{st}\) for each \(s, t\) in \(Q\).
- For each \(s, t\) in \(Q\) the transition probability is: \(a_{st} \equiv P(x_i = t | x_{i-1} = s)\)

**Output:** The output of the model is the set of states at each instant time \(\Rightarrow\) the set of states are observable.

**Property:** The probability of each symbol \(x_i\) depends only on the value of the preceding symbol \(x_{i-1}\): \(P(x_i | x_{i-1}, \ldots, x_1) = P(x_i | x_{i-1})\)

**Formula:** The probability of the sequence:

\[
P(x) = P(x_L, x_{L-1}, \ldots, x_1) = P(x_L | x_{L-1}) P(x_{L-1} | x_{L-2}) \ldots P(x_2 | x_1) P(x_1)
\]
HMM (Hidden Markov Model)

Definition: An **HMM** is a 5-tuple \((Q, V, p, A, E)\), where:

- \(Q\) is a finite set of states, \(|Q|=N\)
- \(V\) is a finite set of observation symbols per state, \(|V|=M\)
- \(p\) is the initial state probabilities.
- \(A\) is the state transition probabilities, denoted by \(a_{st}\) for each \(s, t\) in \(Q\).
  - For each \(s, t\) in \(Q\) the transition probability is: \(a_{st} \equiv P(x_i = t|x_{i-1} = s)\)
- \(E\) is a probability emission matrix, \(e_{sk} \equiv P(v_k \text{ at time } t | q_t = s)\)

**Output**: Only emitted symbols are observable by the system but not the underlying random walk between states -> “hidden”

**Property**: Emissions and transitions are dependent on the current state only and not on the past.
Markov Chains & Hidden Markov Models

- **Markov Chain**
  - $Q$: states
  - $p$: initial state probabilities
  - $A$: transition probabilities

- **HMM**
  - $Q$: states
  - $V$: observations
  - $p$: initial state probabilities
  - $A$: transition probabilities
  - $E$: emission probabilities
The three main questions on HMMs

1. **Evaluation**
   
   GIVEN a HMM M, and a sequence x,
   
   FIND \( \text{Prob}[ x \mid M ] \)

2. **Decoding**
   
   GIVEN a HMM M, and a sequence x,
   
   FIND the sequence \( \pi \) of states that maximizes \( P[ x, \pi \mid M ] \)

3. **Learning**
   
   GIVEN a HMM M, with unspecified transition/emission probs., and a sequence x,
   
   FIND parameters \( \theta = (e_i(\cdot), a_{ij}) \) that maximize \( P[ x \mid \theta ] \)
Back to Biology

1245526462146146136136661664661636616366163616515615115146123562344

ATGCGTGACCATGGTATCCGGCACATGGGTTAACCTTGATGGACAGACATGATAATGCTCGATGCTAGCAGATACAGAACA
Modeling biological sequences

- **Ability to emit DNA sequences of a certain type**
  - Not exact alignment to previously known gene
  - Preserving ‘properties’ of type, not identical sequence

- **Ability to recognize DNA sequences of a certain type (state)**
  - What (hidden) state is most likely to have generated observations
  - Find set of states and transitions that generated a long sequence

- **Ability to learn distinguishing characteristics of each state**
  - Training our generative models on large datasets
  - Learn to classify unlabelled data
Finding CpG islands

• Regions of regulatory importance in promoters of many genes
  – Defined by their methylation state (epigenetic information)

• Methylation process in the human genome:
  – Very high chance of methyl-C mutating to T in CpG
    ➔ CpG dinucleotides are much rarer
  – BUT it is suppressed around the promoters of many genes
    ➔ CpG dinucleotides are much more frequent than elsewhere
    • Such regions are called CpG islands
    • A few hundred to a few thousand bases long

• Problems:
  – Given a short sequence, does it come from a CpG island or not?
  – How to find the CpG islands in a long sequence
Training Markov Chains for CpG islands

- **Training Set:**
  - set of DNA sequences w/ known CpG islands
- **Derive two Markov chain models:**
  - + model: from the CpG islands
  - - model: from the remainder of sequence
- **Transition probabilities for each model:**

**Probability of C following A**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>.180</td>
<td>.274</td>
<td>.426</td>
<td>.120</td>
</tr>
<tr>
<td>C</td>
<td>.171</td>
<td>.368</td>
<td>.274</td>
<td>.188</td>
</tr>
<tr>
<td>G</td>
<td>.161</td>
<td>.339</td>
<td>.375</td>
<td>.125</td>
</tr>
<tr>
<td>T</td>
<td>.079</td>
<td>.355</td>
<td>.384</td>
<td>.182</td>
</tr>
</tbody>
</table>

\[
a_{st}^+ = \frac{c_{st}^+}{\sum_{t'} c_{st'}^+}
\]

\[
a_{st}^- = \frac{c_{st}^-}{\sum_{t'} c_{st'}^-}
\]

- \(c_{st}^+\) is the number of times letter \(t\) followed letter \(s\) inside the CpG islands
- \(c_{st}^-\) is the number of times letter \(t\) followed letter \(s\) outside the CpG islands
Using Markov Models for CpG classification

Q1: Given a short sequence $x$, does it come from CpG island (Yes-No question)

- To use these models for discrimination, calculate the log-odds ratio:

$$S(x) = \log \frac{P(x|\text{model } +)}{P(x|\text{model } -)} = \sum_{i=1}^{L} \log \frac{a_{x_{i-1}x_i}^+}{a_{x_{i-1}x_i}^-}$$

Histogram of log odds scores
Q2: Given a long sequence $x$, how do we find CpG islands in it

(Where question)

- Calculate the log-odds score for a window of, say, 100 nucleotides around every nucleotide, plot it, and predict CpG islands as ones w/ positive values
- Drawbacks: Window size

Use hidden states: CpG (+) or non-CpG (-)
HMM for CpG islands

• Build a single model that combines both Markov chains:
  – ‘+’ states: $A_+, C_+, G_+, T_+$
    • Emit symbols: A, C, G, T in CpG islands
  – ‘-’ states: $A_-, C_-, G_-, T_-$
    • Emit symbols: A, C, G, T in non-islands

• Emission probabilities distinct for the ‘+’ and the ‘-’ states
  – Infer most likely set of states, giving rise to observed emissions
  ➔ ‘Paint’ the sequence with + and - states
Finding most likely state path

- Given the observed emissions, what was the path?
Probability of given path $p$ & observations $x$

- **Known observations:** CGCG
- **Known sequence path:** C+, G-, C-, G+

![Diagram showing a probability model with nodes and edges representing transitions between nucleotides (A, T, G, C) with positive and negative labels. The diagram starts from a 'start' node and ends at an 'end' node, with paths indicated by arrows and color-coded paths from the known observations to the 'end' node.]
Probability of given path $p$ & observations $x$

- Known observations: CGCG
- Known sequence path: C+, G-, C-, G+
Probability of given path $p$ & observations $x$

- $P(p,x) = (a_{0,C^+} * 1) * (a_{C^+,G^-} * 1) * (a_{G^-,C^-} * 1) * (a_{C^-,G^+} * 1) * (a_{G^+,0})$

But in general, we don’t know the path!
How can we find the most likely path? (DECODING)

(using Dynamic Programming)
Finding the most likely path

Find path $\pi$ that maximizes total joint probability $P[ x, \pi ]$
Decoding using Dynamic Programming

- **Goal**
  - GIVEN $x = x_1x_2 \ldots x_N$
  - FIND $\pi^* = \pi_1, \ldots, \pi_N$
  - Such that $P[x, \pi]$ is maximized

- **Use Dynamic Programming**
  - Score can be defined recursively
  - Re-use of problem subparts

- **Viterbi algorithm**
  - Define $V_k(i) =$ Probability of the most likely path through state $\pi_i = k$
  - Compute $V_k(i+1)$ as a function of $\max_k \{ V_k(i) \}$
  - $V_k(i+1) = e_k(x_{i+1}) \times \max_j a_{jk} V_j(i)$
The Viterbi Algorithm

Input: \( x = x_1 \ldots x_N \)

**Initialization:**
\[
V_0(0) = 1 \\
V_k(0) = 0, \text{ for all } k > 0
\]

(0 is the imaginary first position)

**Iteration:**
\[
V_k(i+1) = e_k(x_{i+1}) \times \max_j a_{jk} V_j(i) \\
\text{Ptr}_j(i+1) = \text{argmax}_j a_{jk} V_j(i)
\]

(in practice use logs)

\[
V_k(i+1) = \log e_k(x_{i+1}) + \max_j [ V_j(i) + \log a_{jk} ]
\]

**Termination:**
\[
P(x, \pi^*) = \max_k V_k(N)
\]

**Traceback:**
\[
\pi_N^* = \text{argmax}_k V_k(N) \\
\pi_{i-1}^* = \text{Ptr}_{\pi_i}(i)
\]
The Viterbi Algorithm

Similar to “aligning” a set of states to a sequence

**Time:**
$O(K^2 N)$

**Space:**
$O(KN)$
Finding the optimal *parse* of sequence

Let $x$ be a sequence with a portion of $\sim 1/6$ 6’s, followed by a portion of $\sim 1/2$ 6’s…

$$x = 123456123456…123456626364656…1626364656$$

Then, it is not hard to show that optimal parse is (exercise):

```
  FFF........................F LLL..............................L
```

6 nucleotides “123456” parsed as F, contribute $0.95^6 \times (1/6)^6 = 1.6 \times 10^{-5}$

parsed as L, contribute $0.95^6 \times (1/2)^1 \times (1/10)^5 = 0.4 \times 10^{-5}$

“162636” parsed as F, contribute $0.95^6 \times (1/6)^6 = 1.6 \times 10^{-5}$

parsed as L, contribute $0.95^6 \times (1/2)^3 \times (1/10)^3 = 9.0 \times 10^{-5}$
What have we learned?

• Modeling biological sequences
  – Recognize a *type* of sequence

• Simple examples
  – The dishonest casino
  – Finding CpG islands

• Definitions
  – Markov Chains
  – Hidden Markov Models (HMMs)

• Our first computations
  – Evaluation: know model, emissions, states → \( p \)
  – Viterbi: know model, emissions → find optimal path
Next time: More on HMMs

- **HMM algorithms**
  - Viterbi algorithm
  - Forward / backward

- **Working with HMMs**
  - Posterior decoding
  - Baum-Welch training
  - Viterbi training

- **Applications**
  - HMMs for Gene Finding