Hidden Markov Model and Gene Finding
HMM

• Hidden Markov Model was invented in speech recognition. However, it has tons of applications.
• HMM is widely used in Bioinformatics.
• HMM can be used to solve the following kind of problems:
  – Try to guess your thought from your face
A silly example of an HMM

How surfers speak.

• The surfer knows 4 words: “Dude,” “Bummer,” “Surf,” and “Yeah.”

• He does 3 things: surf, tan and swim

• Every 5 minutes, he changes what he’s doing, and says one word.

Both the change and the word depend only on what he’s doing right then.
A drawing of the surfer HMM

**Surf**
- \( \text{Pr["Dude"]} = .3 \)
- \( \text{Pr["Surf"]} = .6 \)
- \( \text{Pr["Bummer"]} = .05 \)
- \( \text{Pr["Yeah"]} = .05 \)

**Tan**
- \( \text{Pr["Dude"]} = .2 \)
- \( \text{Pr["Surf"]} = .1 \)
- \( \text{Pr["Bummer"]} = .65 \)
- \( \text{Pr["Yeah"]} = .05 \)

**Swim**
- \( \text{Pr["Dude"]} = .5 \)
- \( \text{Pr["Surf"]} = .1 \)
- \( \text{Pr["Bummer"]} = .05 \)
- \( \text{Pr["Yeah"]} = .35 \)
Keeping the example going: decoding

The surfer can be turned on, and go about his business.

Suppose you hear “Dude, yeah, bummer, yeah, dude, yeah, surf”

What was the *most likely* thing he was doing at each of these time steps?

In an HMM, that’s *hidden*, but can be estimated in time linear in the list of words.
Hidden Markov models

The most commonly used generative model in bioinformatics is the HMM. The basic idea: A Markov chain that emits symbols. What that means in practice:

A finite set of states, \( X \),

A finite alphabet/set of observations, \( O \)

For each state \( i \), the **transition** probability that from state \( i \) we go to each other state \( j \), and

For each state \( i \), the **emission** probability that we emit the symbol \( a \) for each symbol \( a \) in \( O \).
Represent HMM in Computer

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Transition prob.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.35</td>
<td>0.35</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

emission prob.
A review of the basic dogma

DNA sequence contains **genes**

which are transcribed and spliced into **mRNA**

which is translated into **protein**.

Every 3 bases of mRNA = 1 amino acid
Some more details about genes

In higher organisms, genes contain alternating regions of **exons**, which form the mature mRNA, and **introns**, which are spliced out.

![Diagram showing the process of transcription and splicing, followed by translation into a protein.](image-url)
How to do this, as a CS problem

• Given: A (potentially very long) string $S$ over the alphabet \{A,G,C,T\}

• Find: Intervals of that string which correspond to genes, and their intron/exon structure.

Example:

```
ACAGATAGATGCAGACGAGTGACAGTGACACAGATAGATGCAGACGAGTGACAGTGACACAGATAGATGCAGACGAGTGACAGTGAC
```

<table>
<thead>
<tr>
<th>exons</th>
</tr>
</thead>
<tbody>
<tr>
<td>introns</td>
</tr>
</tbody>
</table>

```
Two kinds of Cells

• Prokaryotes – no nucleus (bacteria)
  – Their genomes are circular

• Eukaryotes – have nucleus (animal, plants)
  – Linear genomes with multiple chromosomes in pairs. When pairing up, they look like

Middle: centromere
Top: p-arm
Bottom: q-arm
The difference that we concern about

• Genes of prokaryotes have no introns!
Prokaryotes
# Genetic code

<table>
<thead>
<tr>
<th>First letter</th>
<th>Second letter</th>
<th>Third letter</th>
<th>Protein Region</th>
<th>Amino Acid</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUG</td>
<td>U</td>
<td>C</td>
<td>I</td>
<td>Methionine; start codon</td>
</tr>
<tr>
<td>U</td>
<td>C</td>
<td>U</td>
<td>I</td>
<td>Phenylalanine</td>
</tr>
<tr>
<td>U</td>
<td>G</td>
<td>C</td>
<td>I</td>
<td>Leucine</td>
</tr>
<tr>
<td>CU</td>
<td>C</td>
<td>U</td>
<td>I</td>
<td>Leucine</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>U</td>
<td>I</td>
<td>Serine</td>
</tr>
<tr>
<td>A</td>
<td>U</td>
<td>C</td>
<td>I</td>
<td>Tyrosine</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>C</td>
<td>I</td>
<td>Stop codon</td>
</tr>
<tr>
<td>U</td>
<td>U</td>
<td>C</td>
<td>I</td>
<td>Glutamic acid</td>
</tr>
<tr>
<td>U</td>
<td>A</td>
<td>C</td>
<td>I</td>
<td>Aspartic acid</td>
</tr>
<tr>
<td>U</td>
<td>G</td>
<td>C</td>
<td>I</td>
<td>Alanine</td>
</tr>
<tr>
<td>G</td>
<td>U</td>
<td>C</td>
<td>I</td>
<td>Valine</td>
</tr>
<tr>
<td>G</td>
<td>U</td>
<td>C</td>
<td>I</td>
<td>Glutamic acid</td>
</tr>
<tr>
<td>G</td>
<td>G</td>
<td>C</td>
<td>I</td>
<td>Glutamine</td>
</tr>
</tbody>
</table>

© 2001 Strasser Associates, Inc.
For example

- ATG CAT ATT GAA CTT GCA TCG CCA GTT GCA CAT ATT TGG TTC TTA
- M H I E L A S P V A H I W F L
- TCA TTG CCG TCT CGT ATC GGT TTA CTT TTA GAT ATG CCA TTG CGC
- S L P S R I G L L L D M P L R
- GAC ATC GAA CGT GTA CTT TAT TTT GAA ATG TAC ATC GTG ACC TAG
- D I E R V L Y F E M Y I V T *
Formalization of the gene prediction problem

• Given a sequence of letters of \{A,C,G,T\}, label each position with labels \{I, T, P, G\}, where I means intergenic, G means internal codons, T means start of a gene, P means stop codon.

• Example:
  • ..TAGTCATGCACTTGCATCGCCAGTTGCACATATTUGGATTTCTTA..
  • ..IIIIITGGGGGGGGGPPIII..
An simple HMM for a prokaryote’ genome
## Parameters of the HMM

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
<th>ATG</th>
<th>TGA</th>
<th>TAA</th>
<th>TAG</th>
<th>AAA</th>
<th>AAC</th>
<th>...</th>
<th>...</th>
<th>...</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>¼</td>
<td>¼</td>
<td>¼</td>
<td>¼</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/61</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/61</td>
<td>1/61</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/3</td>
<td>1/3</td>
<td>1/3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The probability of a path

• Bayes’ rule

\[ \text{Pr}(\text{path}|\text{seq}) = \frac{\text{Pr}(\text{seq}|\text{path}) \times \text{Pr}(\text{path})}{\text{Pr}(\text{seq})} \]

• \( \text{Pr}(\text{seq}) \) is a fixed number. Therefore, to maximize \( \text{Pr}(\text{path}|\text{seq}) \), we need to maximize \( \text{Pr}(\text{seq}|\text{path}) \times \text{Pr}(\text{path}) \)

\[ \prod_{i=1}^{n} T[p_{i-1}, p_i] \]

\[ \prod_{i=1}^{n} E[p_i, seq_i] \]
Question?

• Suppose the genome was generated/output by the HMM. Observing the sequence, can we compute the most probable path of states that the HMM were through. I.e. maximize:

\[ \prod_{i=1..n} T[p_{i-1}, p_i] \times E[p_i, seq_i] \]

• Knowing the path, we can label the genome.
Answer – Dynamic Programming

• Yes, we can.
Dynamic Programming

- Suppose the sequence has length n.
- Let $DP[i, x]$ be the highest probability for a path generating the first i letters of the sequence, and last state being x. Then

\[
DP[i, x] = \max_y DP[i - l(x), y] \times T[y, x] \times E[x, s_{i-l(x)+1...i}]
\]
Dynamic Programming

\[\text{DP}[0,x] = 1 \text{ for any } x \text{ in } \{\text{I,T,G,P}\}\]

For \(i\) from 1 to \(n\)

For \(x\) in \(\{\text{I,T,G,P}\}\)

\[\text{DP}[i, x] = \max_y \text{DP}[i - l(x), y] \times T[y, x] \times E[x, s_{i-l(x)+1..i}]\]

Let \(x\) maximize \(\text{DP}[n,x]\). Output \(\text{DP}[n,x]\).

Backtracking.
How to train the parameters

• Suppose that we know a genome and all its genes, i.e., we know the labels – \{I,T,G,P\}

• Then we know a path of the HMM. Then we can compute \(\Pr(\text{one label } \rightarrow \text{ another})\), the transition probability.

• Also, for each label/state, we count the number of different letters in the genome with the same state, we can compute \(\Pr(\text{a letter } | \text{ a state})\), the emission probability.
What if we know nothing

- We start with an arbitrary values of the parameters.
- Then we predict the genes.
- Then we do statistics and change the parameters
- Then we predict the genes with new parameters.
- .......
- Until converge.
Problem

• The output letter of the HMM at one state only depends on the state itself. However, it should also depend on the previous output letter(s).
A more complex HMM

• Replace $\Pr(\text{output} \mid \text{current\_state})$ by $\Pr(\text{output} \mid \text{current\_state, previous\_output})$
Dynamic Programming

• Suppose the sequence has length n.
• Let \(\text{DP}[i, x]\) be the highest probability for a path generating the first \(i\) letters of the sequence, and last state being \(x\). Then

\[
\text{DP}[i, x] = \max_y \text{DP}[i - l(x), y] \times T[y, x] \times E[x, s_{i-l(x)+1..i}, s_{i-l(x)-l(y)+1..i-l(x)}]
\]
Dynamic Programming

\[ DP[0, x] = 1 \text{ for any } x \in \{I, T, G, P\} \]

For i from 1 to n

For \( x \) in \( \{I, T, G, P\} \)

\[ DP[i, x] = \max_y \]

\[ DP[i - l(x), y] \times T[y, x] \times E[x, s_{i-l(x)+1..i}, s_{i-l(x)-l(y)+1..i-l(x)}] \]

Let \( x \) maximize \( DP[n, x] \). Output \( DP[n, x] \).

Backtracking.
Effectiveness of HMM-based finders

- The best gene-finding HMM (GenScan, Burge and Karlin 1997) has ~80% sensitivity and ~80% specificity at the exon level. (That is, roughly 80% of true exons are entirely correctly found, and about 80% of the predicted exons are entirely correct.)
Gene Finding with Homology

• More and more EST (Expressed Sequence Tag) sequences have been collected.

• Complementary DNA (cDNA) is derived from RNA - usually messenger RNA (mRNA).
  – This is done using RNA as the template and the enzyme reverse transcriptase which is obtained from retroviruses

• Then those DNA segments are sequences.

• If a part of the genome is highly similar to an EST, it is highly possible the part is a part of a gene.
Some Gene Finding Programs

- FGENES
- GENSCAN
- Twinscan
- GenomeScan