Interpolated Markov Models for Gene Finding

BMI/CS 776
www.biostat.wisc.edu/bmi776/
Spring 2017
Anthony Gitter
gitter@biostat.wisc.edu
Goals for Lecture

Key concepts
• the gene-finding task
• the trade-off between potential predictive value and parameter uncertainty in choosing the order of a Markov model
• interpolated Markov models
The Gene Finding Task

**Given:** an uncharacterized DNA sequence

**Do:** locate the genes in the sequence, including the coordinates of individual *exons* and *introns*
Sources of Evidence for Gene Finding

• **Signals**: the sequence *signals* (e.g. splice junctions) involved in gene expression

• **Content**: statistical properties that distinguish protein-coding DNA from non-coding DNA

• **Conservation**: signal and content properties that are conserved across related sequences (e.g., orthologous regions of the mouse and human genome)
Gene Finding: Search by Content

- Encoding a protein affects the statistical properties of a DNA sequence
  - some amino acids are used more frequently than others (Leu more prevalent than Trp)
  - different numbers of codons for different amino acids (Leu has 6, Trp has 1)
  - for a given amino acid, usually one codon is used more frequently than others
    - this is termed *codon preference*
    - these preferences vary by species
## Codon Preference in E. Coli

<table>
<thead>
<tr>
<th>AA</th>
<th>codon</th>
<th>/1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gly</td>
<td>GGG</td>
<td>1.89</td>
</tr>
<tr>
<td>Gly</td>
<td>GGA</td>
<td>0.44</td>
</tr>
<tr>
<td>Gly</td>
<td>GGU</td>
<td>52.99</td>
</tr>
<tr>
<td>Gly</td>
<td>GGC</td>
<td>34.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glu</td>
<td>GAG</td>
<td>15.68</td>
</tr>
<tr>
<td>Glu</td>
<td>GAA</td>
<td>57.20</td>
</tr>
<tr>
<td>Asp</td>
<td>GAU</td>
<td>21.63</td>
</tr>
<tr>
<td>Asp</td>
<td>GAC</td>
<td>43.26</td>
</tr>
</tbody>
</table>
Reading Frames

- A given sequence may encode a protein in any of the six reading frames

```
G C T A C G G A G C T T C G G A G C
C G A T G C C T C G A A G C C T C G
```
Open Reading Frames (ORFs)

- An ORF is a sequence that
  - starts with a potential start codon
  - ends with a potential stop codon, in the same reading frame
  - doesn’t contain another stop codon in-frame
  - and is sufficiently long (say > 100 bases)

- An ORF meets the minimal requirements to be a protein-coding gene in an organism without introns
Markov Models & Reading Frames

- Consider modeling a given coding sequence
- For each “word” we evaluate, we’ll want to consider its position with respect to the reading frame we’re assuming

\[
\begin{align*}
G & \text{ is in 3}^{\text{rd}} \text{ codon position} \\
C & \text{ is in 1}^{\text{st}} \text{ position} \\
A & \text{ is in 2}^{\text{nd}} \text{ position}
\end{align*}
\]

- Can do this using an inhomogeneous model
Inhomogeneous Markov Model

- **Homogenous Markov model**: transition probability matrix does not change over time or position

- **Inhomogenous Markov model**: transition probability matrix depends on the time or position
A Fifth Order Inhomogeneous Markov Model

start

position 2:
- AAAAA
- CTACA
- CTACC
- CTACG
- CTACT
- GCTAC
- TTTTT

position 3:
- AAAAA
- CTACA
- CTACC
- CTACG
- CTACT
- GCTAC
- TTTTT

position 1:
- AAAAA
- CTACA
- TACAA
- TACAC
- TACAT
- TACAG
- TACAT
- TTTTT

Trans. to states in pos. 2
Selecting the Order of a Markov Model

- Higher order models remember more “history”
- Additional history can have predictive value
- Example:
  - predict the next word in this sentence fragment
    “…you__” (are, give, passed, say, see, too, …?)
  - now predict it given more history
    “…can you___”
    “…say can you___”
    “…oh say can you___”
Selecting the Order of a Markov Model

• But the number of parameters we need to estimate grows exponentially with the order
  – for modeling DNA we need $O(4^{n+1})$ parameters for an $n$th order model

• The higher the order, the less reliable we can expect our parameter estimates to be

• Suppose we have 100k bases of sequence to estimate parameters of a model
  – for a 2nd order homogeneous Markov chain, we’d see each history 6250 times on average
  – for an 8th order chain, we’d see each history ~ 1.5 times on average
Interpolated Markov Models

• The IMM idea: manage this trade-off by interpolating among models of various orders
• *Simple* linear interpolation:

\[
P_{\text{IMM}}(x_i \mid x_{i-n}, \ldots, x_{i-1}) = \lambda_0 P(x_i) + \lambda_1 P(x_i \mid x_{i-1}) + \lambda_2 P(x_i \mid x_{i-2}) + \ldots + \lambda_n P(x_i \mid x_{i-n}, \ldots, x_{i-1})
\]

• where \( \sum_i \lambda_i = 1 \)
Interpolated Markov Models

- We can make the weights depend on the history
  - for a given order, we may have significantly more data to estimate some words than others
- General linear interpolation

\[
P_{\text{IMM}}(x_i \mid x_{i-n}, \ldots, x_{i-1}) = \lambda_0 P(x_i)
+ \lambda_1(x_{i-1}) P(x_i \mid x_{i-1})
+ \ldots
+ \lambda_n(x_{i-n}, \ldots, x_{i-1}) P(x_i \mid x_{i-n}, \ldots, x_{i-1})
\]

\(\lambda\) is a function of the given history
The GLIMMER System
[Salzberg et al., Nucleic Acids Research, 1998]

• System for identifying genes in bacterial genomes
• Uses 8th order, inhomogeneous, interpolated Markov chain models
IMMs in GLIMMER

• How does GLIMMER determine the values?
• First, let’s express the IMM probability calculation recursively

$$P_{IMM,n}(x_i | x_{i-n}, ..., x_{i-1}) =$$

$$\lambda_n(x_{i-n}, ..., x_{i-1})P(x_i | x_{i-n}, ..., x_{i-1}) +$$

$$[1 - \lambda_n(x_{i-n}, ..., x_{i-1})]P_{IMM,n-1}(x_i | x_{i-n+1}, ..., x_{i-1})$$

• Let $$c(x_{i-n}, ..., x_{i-1})$$ be the number of times we see the history $$x_{i-n}, ..., x_{i-1}$$ in our training set

$$\lambda_n(x_{i-n}, ..., x_{i-1}) = 1 \text{ if } c(x_{i-n}, ..., x_{i-1}) > 400$$
**IMMs in GLIMMER**

- If we haven’t seen $x_{i-n}, \ldots, x_{i-1}$ more than 400 times, then compare the counts for the following:

<table>
<thead>
<tr>
<th>$n$th order history + base</th>
<th>(n-1)th order history + base</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{i-n}, \ldots, x_{i-1}, a$</td>
<td>$x_{i-n+1}, \ldots, x_{i-1}, a$</td>
</tr>
<tr>
<td>$x_{i-n}, \ldots, x_{i-1}, c$</td>
<td>$x_{i-n+1}, \ldots, x_{i-1}, c$</td>
</tr>
<tr>
<td>$x_{i-n}, \ldots, x_{i-1}, g$</td>
<td>$x_{i-n+1}, \ldots, x_{i-1}, g$</td>
</tr>
<tr>
<td>$x_{i-n}, \ldots, x_{i-1}, t$</td>
<td>$x_{i-n+1}, \ldots, x_{i-1}, t$</td>
</tr>
</tbody>
</table>

- Use a statistical test ($\chi^2$) to get a value $d$ indicating our confidence that the distributions of $x_i$ depend on the order.
IMMs in GLIMMER

• Putting it all together

$$\lambda_n(x_{i-n}, \ldots, x_{i-1}) = \begin{cases} 
1 & \text{if } c(x_{i-n}, \ldots, x_{i-1}) > 400 \\
d \times \frac{c(x_{i-n}, \ldots, x_{i-1})}{400} & \text{else if } d \geq 0.5 \\
0 & \text{otherwise}
\end{cases}$$

where $d \in (0,1)$
IMM Example

• Suppose we have the following counts from our training set:

<table>
<thead>
<tr>
<th></th>
<th>ACGA</th>
<th>CGA</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>25</td>
<td>100</td>
<td>175</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-----</td>
<td>----</td>
</tr>
<tr>
<td></td>
<td>ACGC</td>
<td>CGC</td>
<td>GC</td>
</tr>
<tr>
<td>Count</td>
<td>40</td>
<td>90</td>
<td>140</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-----</td>
<td>----</td>
</tr>
<tr>
<td></td>
<td>ACGG</td>
<td>CGG</td>
<td>GG</td>
</tr>
<tr>
<td>Count</td>
<td>15</td>
<td>35</td>
<td>65</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-----</td>
<td>----</td>
</tr>
<tr>
<td></td>
<td>ACGT</td>
<td>CGT</td>
<td>GT</td>
</tr>
<tr>
<td>Count</td>
<td>20</td>
<td>75</td>
<td>120</td>
</tr>
</tbody>
</table>

\[\chi^2\text{ test: } d = 0.857 \]
\[\lambda_3(\text{ACG}) = 0.857 \times \frac{100}{400} = 0.214\]
\[\lambda_2(\text{CG}) = 0 \quad (d < 0.5, \quad c(\text{CG}) < 400)\]
\[\lambda_1(\text{G}) = 1 \quad (c(\text{G}) > 400)\]
Now suppose we want to calculate $P_{\text{IMM,3}}(T \mid ACG)$

$$P_{\text{IMM,1}}(T \mid G) = \lambda_1(G)P(T \mid G) + (1 - \lambda_1(G))P_{\text{IMM,0}}(T)$$

$$= P(T \mid G)$$

$$P_{\text{IMM,2}}(T \mid CG) = \lambda_2(CG)P(T \mid CG) + (1 - \lambda_2(CG))P_{\text{IMM,1}}(T \mid G)$$

$$= P(T \mid G)$$

$$P_{\text{IMM,3}}(T \mid ACG) = \lambda_3(ACG)P(T \mid ACG) + (1 - \lambda_3(ACG))P_{\text{IMM,2}}(T \mid CG)$$

$$= 0.214 \times P(T \mid ACG) + (1 - 0.214) \times P(T \mid G)$$

$$= 0.214 \times 0.2 + (1 - 0.214) \times 0.24$$
Gene Recognition in GLIMMER

• Essentially ORF classification
• For each ORF
  – calculate the probability of the ORF sequence in each of the 6 possible reading frames
  – if the highest scoring frame corresponds to the reading frame of the ORF, mark the ORF as a gene
• For overlapping ORFs that look like genes
  – score overlapping region separately
  – predict only one of the ORFs as a gene
Gene Recognition in GLIMMER
GLIMMER Experiment

- 8th order IMM vs. 5th order Markov model
- Trained on 1168 genes (ORFs really)
- Tested on 1717 annotated (more or less known) genes
GLIMMER Results

<table>
<thead>
<tr>
<th></th>
<th>TP</th>
<th>FN</th>
<th>FP &amp; TP?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>Genes found</td>
<td>Genes missed</td>
<td>Additional genes</td>
</tr>
<tr>
<td>GLIMMER IMM</td>
<td>1680 (97.8%)</td>
<td>37</td>
<td>209</td>
</tr>
<tr>
<td>5th-Order Markov</td>
<td>1574 (91.7%)</td>
<td>143</td>
<td>104</td>
</tr>
</tbody>
</table>

The first column indicates how many of the 1717 annotated genes in *H. influenzae* were found by each algorithm. The ‘additional genes’ column shows how many extra genes, not included in the 1717 annotated entries, were called genes by each method.

- GLIMMER has greater sensitivity than the baseline
- It’s not clear if its precision/specificity is better