BMI/CS 776
Lecture 8
Eukaryotic Gene Finding

Colin Dewey
2007.02.15
(adapted from slides by Mark Craven)
Each shape represents a functional unit of a gene or genomic region.

Pairs of intron/exon units represent the different ways an intron can interrupt a coding sequence (after 1st base in codon, after 2nd base or after 3rd base).

Complementary submodel (not shown) detects genes on opposite DNA strand.
The GENSCAN HMM

• for each sequence type, GENSCAN models
  • the length distribution
  • the sequence composition
• length distribution models vary depending on sequence type
  * nonparametric (using histograms)
  • parametric (using geometric distributions)
  • fixed-length
• sequence composition models vary depending on type
  • 5\textsuperscript{th}-order, inhomogeneous
  • 5\textsuperscript{th} -order homogenous
  • 1\textsuperscript{st}-order inhomogeneous
  * tree-structured variable memory (MDD)
Splice Signals

Figures from the Sanger Center.

**donor sites**

Sequence of U1 snRNA that base-pairs with donor site: GUCCAUUCA

**acceptor sites**

3,673 Chromosome 22 splice donor sites

3,673 Chromosome 22 splice acceptor sites
Motivation for MDD

• How can we model significant dependencies between non-adjacent positions?

ATGGGTCCATCTACATATACACATCCATT
TATCTCTACCCCGCTAGCTAGCTCGGATT
GCTACGACCACGAAGCTACGCTAGCTGGA
CCTTCGGCTATATATTATTCTTCTTATA
TCGAAATAGACTAGCTAAATCGCTAGCTA
TCCGCGCTCGCTAACACAGCTACCAAATAGA
CGTAGCTAGATCGAATCGAAAGCCCTACT
ACACCAGGCTTTCTAATCGATTAGATCCCA

\[ i \]

<table>
<thead>
<tr>
<th>pos ( i ) matches consensus</th>
<th>pos ( i ) does NOT match consensus</th>
</tr>
</thead>
<tbody>
<tr>
<td>( pos \ j = A )</td>
<td></td>
</tr>
<tr>
<td>( pos \ j = C )</td>
<td></td>
</tr>
<tr>
<td>( pos \ j = G )</td>
<td></td>
</tr>
<tr>
<td>( pos \ j = T )</td>
<td></td>
</tr>
</tbody>
</table>

• compute \( \chi^2 \) values using 2\( \times \)4 table
  
  **alternative hypothesis**: distribution for column \( j \) depends on what is in column \( i \)
  
  **null hypothesis**: distribution for column \( j \) is the same in both cases
Motivation for MDD

- Table shows $\chi^2$ values for pairs of positions around donor sites
- Values marked with * show statistically significant dependency

<table>
<thead>
<tr>
<th></th>
<th>Con</th>
<th>$j$:</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>+3</th>
<th>+4</th>
<th>+5</th>
<th>+6</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>c/a</td>
<td>—</td>
<td>61.8*</td>
<td>14.9</td>
<td>5.8</td>
<td>20.2*</td>
<td>11.2</td>
<td>18.0*</td>
<td>131.8*</td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>A</td>
<td>115.6*</td>
<td>—</td>
<td>40.5*</td>
<td>20.3*</td>
<td>57.5*</td>
<td>59.7*</td>
<td>42.9*</td>
<td>336.5*</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>G</td>
<td>15.4</td>
<td>82.8*</td>
<td>—</td>
<td>13.0</td>
<td>61.5*</td>
<td>41.4*</td>
<td>96.6*</td>
<td>310.8*</td>
<td></td>
</tr>
<tr>
<td>+3</td>
<td>a/g</td>
<td>8.6</td>
<td>17.5*</td>
<td>13.1</td>
<td>—</td>
<td>19.3*</td>
<td>1.8</td>
<td>0.1</td>
<td>60.5*</td>
<td></td>
</tr>
<tr>
<td>+4</td>
<td>A</td>
<td>21.8*</td>
<td>56.0*</td>
<td>62.1*</td>
<td>64.1*</td>
<td>—</td>
<td>56.8*</td>
<td>0.2</td>
<td>260.9*</td>
<td></td>
</tr>
<tr>
<td>+5</td>
<td>G</td>
<td>11.6</td>
<td>60.1*</td>
<td>41.9*</td>
<td>93.6*</td>
<td>146.6*</td>
<td>—</td>
<td>33.6*</td>
<td>387.3*</td>
<td></td>
</tr>
<tr>
<td>+6</td>
<td>t</td>
<td>22.2*</td>
<td>40.7*</td>
<td>103.8*</td>
<td>26.5*</td>
<td>17.8*</td>
<td>32.6*</td>
<td>—</td>
<td>243.6*</td>
<td></td>
</tr>
</tbody>
</table>
The Maximal Dependence Decomposition (MDD) Approach

• induce a tree that represents the dependency structure apparent in the data

• induce partial position weight matrices for each node and leaf of tree

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1</td>
<td>0.3</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.4</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>C</td>
<td>0.5</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.6</td>
<td>0.1</td>
<td>0.2</td>
<td>0.7</td>
</tr>
<tr>
<td>G</td>
<td>0.2</td>
<td>0.2</td>
<td>0.6</td>
<td>0.5</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>T</td>
<td>0.2</td>
<td>0.3</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
</tr>
</tbody>
</table>

• use the tree + weight matrices to calculate the probability of a given sequence
An MDD Learned Tree

A, C, or U at pos 5

Figure from Burge & Karlin, *Journal of Molecular Biology, 1997*
The MDD Algorithm: Finding the Tree

Given: a set of aligned training sequences $T$
positions $P = \{1, \ldots, k\}$
tree = find_MDD_subtree($T, P$)

\[
S_i = \sum_{j \neq i} \chi^2(C_i, x_j)
\]

find_MDD_subtree($T, P$)
for each position $i$
    determine the consensus base $C_i$
    calculate dependence between $C_i$ and other positions
if stopping criteria not met
    choose the value of $i$ such that $S_i$ is maximal
    make a node with $C_i$ as the test
$D_i^+ = \text{sequences in } T \text{ with base } C_i \text{ at position } i$
$D_i^- = \text{other sequences}$
left subtree = find_MDD_subtree($D_i^+, P - \{i\}$)
right subtree = find_MDD_subtree($D_i^-, P - \{i\}$)
Stopping Criteria for MDD Tree Learning

1. the \((k-1)^{th}\) level is reached; no further positions to split on

2. no significant dependencies between positions are detected

3. number of sequences in given subset is sufficiently small
Explaining a Sequence with an MDD Tree

- shown are selected position weight matrices for the tree
Explaining a Sequence with an MDD Tree

- calculate $\Pr(x_5)$

  if $x_5 \neq G$, use the weight matrix for $H_5$ subset
  else

    - calculate $\Pr(x_{-1})$ from $G_5$ subset
      if $x_{-1} \neq G$, use the WM for $G_5H_{-1}$ subset
      else

        - calculate $\Pr(x_{-2})$ from $G_5G_{-1}$ subset
we can represent the dependency structure of a sequence model as a graph

- nodes represent sequence positions
- edges represent dependencies in probability distribution

- the dependency structure of a 0\textsuperscript{th} order Markov chain of length 4 (e.g. a motif model inferred by MEME):

\begin{itemize}
  \item note: this is different than the transition graph
\end{itemize}
A Graphical View of Dependency Structure

- 1\textsuperscript{st} order model
  \[ \begin{align*}
  x_1 &\rightarrow x_2 \\
  x_2 &\rightarrow x_3 \\
  x_3 &\rightarrow x_4
  \end{align*} \]

- 2\textsuperscript{nd} order model
  \[ \begin{align*}
  x_1 &\rightarrow x_2 \\
  x_2 &\rightarrow x_3 \\
  x_3 &\rightarrow x_4
  \end{align*} \]

- For a fixed-length model, we could consider arbitrary dependencies
  \[ \begin{align*}
  x_1 &\leftrightarrow x_2 \\
  x_2 &\leftrightarrow x_3 \\
  x_3 &\leftrightarrow x_4
  \end{align*} \]
A Graphical View of Dependency Structure

- MDD allows arbitrary dependencies conditioned on *values* of certain variables

\[
x_3 = G \]

\[
x_4 = G
\]

- Graphical representation of dependencies with variables \(x_1, x_2, x_3, x_4\)
Duration Modeling in HMMs

• suppose we have a type of sequence for which the base distribution is the same regardless of length

• the simplest way to model it:

\[
p \quad 1-p
\]

\[
\begin{array}{c}
A & 0.4 \\
C & 0.1 \\
G & 0.2 \\
T & 0.3 \\
\end{array}
\]

• this encodes a \textit{geometric} distribution (shifted by 1) on the length of sequences
Duration Modeling in HMMs

- min length = 5; geometric distribution over longer sequences

- any distribution over length 2 to 6
Length Distributions of Introns/Exons

geometric dist. provides good fit
Semi-Markov HMMs
(a.k.a. Generalized HMMs)

- key idea: decouple length from composition
- represent a parse $\Pi$, as a sequence of states and associated lengths (durations)

$$\tilde{q} = \{q_1, q_2, \ldots, q_n\}$$

$$\tilde{d} = \{d_1, d_2, \ldots, d_n\}$$

Diagram:
- $N \rightarrow P^+$
- $P^+ \rightarrow F^+$
- $E_{\text{init}}^+$

Lengths:
- 307
- 52
- 254
- 410
Semi-Markov Models

• representing a parse $\Pi$, as a sequence of states and associated lengths (durations)
  \[ \tilde{q} = \{q_1, q_2, \ldots, q_n\} \quad \tilde{d} = \{d_1, d_2, \ldots, d_n\} \]

• the joint probability of generating parse $\Pi$ and sequence $x$

\[
Pr(x, \pi) = a_{\text{start}, 1} \ Pr(d_1 \mid q_1) Pr(x_1 \mid q_1, d_1) \times \prod_{k=2}^{n} a_{k-1,k} \ Pr(d_k \mid q_k) Pr(x_k \mid q_k, d_k)
\]
DP with Semi-Markov Models

states

sequence positions

complexity of Viterbi/Forward/Backward in standard HMMs is $O(S^2L)$ where $S$ = number of states, $L$ = sequence length

complexity in semi-Markov HMMs is $O(S^2LD)$ where $D$ = maximum length of a segment
DP with Semi-Markov Models

- review: Forward algorithm recurrence for HMMs
  \[
  f_l(i) = \sum_k f_k(i-1) \ a_{kl} \ \Pr(x_i \mid q_l)
  \]
  transition from \(k\) to \(l\); prob. of emitting \(x_i\) from \(l\)

- for semi-Markov models: each Forward value assumes we’re ending a segment in the given state
  \[
  f_l(i) = \sum_k \sum_{d=1}^{D} \left[ f_k(i-d) \ a_{kl} \ \Pr(d \mid q_l) \ \prod_{j=i-d+1}^{i} \Pr(x_j \mid q_l) \right]
  \]
  prob. of length \(d\) segment from \(l\); prob. of emitting \(x_{i-d+1} \ldots x_i\) from \(l\)
GENSCAN Conclusions

• HMMs readily enable background knowledge to be incorporated into the model
  • state topology
  • length distributions
  • order of Markov chains
• key technical ideas
  • semi-Markov models (old): can represent arbitrary length distributions
  • MDD (new): can represent context-specific dependencies