Motif finding with Gibbs sampling

CS 466
Saurabh Sinha
Regulatory networks

- Genes are switches, transcription factors are (one type of) input signals, proteins are outputs.
- Proteins (outputs) may be transcription factors and hence become signals for other genes (switches).
- This may be the reason why humans have so few genes (the circuit, not the number of switches, carries the complexity).
Decoding the regulatory network

• Find patterns (“motifs”) in DNA sequence that occur more often than expected by chance
  – These are likely to be binding sites for transcription factors
  – Knowing these can tell us if a gene is regulated by a transcription factor (i.e., the “switch”)
Transcriptional regulation

TRANSCRIPTION FACTOR

GENE

ACAGTGA

PROTEIN
Transcriptional regulation

TRANSCRIPTION FACTOR

ACAGTG

GENE
A motif model

- To define a motif, let's say we know where the motif starts in the sequence.
- The motif start positions in their sequences can be represented as \( s = (s_1, s_2, s_3, \ldots, s_t) \).
Motifs: Matrices and Consensus

<table>
<thead>
<tr>
<th>Alignment</th>
<th>Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong> G g t a c T t</td>
<td><strong>A</strong> 3 0 1 0 3 1 1 0</td>
</tr>
<tr>
<td><strong>C</strong> c A t a c g t</td>
<td><strong>C</strong> 2 4 0 0 1 4 0 0</td>
</tr>
<tr>
<td><strong>A</strong> c g t T A g t</td>
<td><strong>G</strong> 0 1 4 0 0 0 3 1</td>
</tr>
<tr>
<td><strong>A</strong> c g t C C A t</td>
<td><strong>T</strong> 0 0 0 5 1 0 1 4</td>
</tr>
<tr>
<td><strong>C</strong> c g t t a c g G</td>
<td></td>
</tr>
</tbody>
</table>

- Line up the patterns by their start indexes
  
  \[ s = (s_1, s_2, \ldots, s_t) \]

- Construct “position weight matrix” with frequencies of each nucleotide in columns

- Consensus nucleotide in each position has the highest frequency in column
Position weight matrices

• Suppose there were $t$ sequences to begin with
• Consider a column of a position weight matrix
• The column may be $(t, 0, 0, 0)$
  – A perfectly conserved column
• The column may be $(t/4, t/4, t/4, t/4)$
  – A completely uniform column
• “Good” profile matrices should have more conserved columns
Information Content

• In a PWM, convert frequencies to probabilities
• PWM W: $W_{\beta k}$ = frequency of base $\beta$ at position $k$
• $q_\beta$ = frequency of base $\beta$ by chance
• Information content of W:

$$\sum_{k} \sum_{\beta \in \{A,C,G,T\}} W_{\beta k} \log \frac{W_{\beta k}}{q_\beta}$$
Information Content

• If $W_{\beta k}$ is always equal to $q_\beta$, i.e., if $W$ is similar to random sequence, information content of $W$ is 0.

• If $W$ is different from $q$, information content is high.
Detecting Subtle Sequence Signals: a Gibbs Sampling Strategy for Multiple Alignment

Lawrence et al. 1993
Motif Finding Problem

• Given a set of sequences, find the motif shared by all or most sequences, while its starting position in each sequence is unknown

• Assumption:
  – Each motif appears exactly once in one sequence
  – The motif has fixed length
Generative Model

• Suppose the sequences are aligned, the aligned regions are generated from a motif model

• Motif model is a PWM. A PWM is a position-specific multinomial distribution.
  – For each position i, a multinomial distribution on (A,C,G,T):
    $q_{iA}, q_{iC}, q_{iG}, q_{iT}$

• The unaligned regions are generated from a background model: $p_A, p_C, p_G, p_T$
Notations

- Set of symbols: $\Sigma$
- Sequences: $S = \{S_1, S_2, \ldots, S_N\}$
- Starting positions of motifs: $A = \{a_1, a_2, \ldots, a_N\}$
- Motif model ($\theta$): $q_{ij} = P(\text{symbol at the } i\text{-th position} = j)$
- Background model: $p_j = P(\text{symbol} = j)$
- Count of symbols in each column: $c_{ij} = \text{count of symbol, } j, \text{ in the } i\text{-th column in the aligned region}$
Probability of data given model

\[
P(S \mid A, \theta) = \prod_{i=1}^{W} \prod_{j=1}^{\mid \Sigma \mid} q_{ij}^{c_{ij}} \quad P(S \mid A, \theta_0) = \prod_{i=1}^{W} \prod_{j=1}^{\mid \Sigma \mid} p_{ij}^{c_{ij}}
\]
Scoring Function

• Maximize the log-odds ratio:

\[
P(S \mid A, \theta) = \prod_{i=1}^{W} \prod_{j=1}^{\Sigma} q_{ij}^{c_{ij}} \quad P(S \mid A, \theta_0) = \prod_{i=1}^{W} \prod_{j=1}^{\Sigma} p_{ij}^{c_{ij}}
\]

\[
F = \log \frac{P(S \mid A, \theta)}{P(S \mid A, \theta_0)} = \sum_{i=1}^{W} \sum_{j=1}^{\Sigma} c_{ij} \log \frac{q_{ij}}{p_j}
\]

• Is greater than zero if the data is a better match to the motif model than to the background model
Optimization and Sampling

• To maximize a function, \( f(x) \):
  – Brute force method: try all possible \( x \)
  – Sample method: sample \( x \) from probability distribution: \( p(x) \sim f(x) \)
  – Idea: suppose \( x_{\text{max}} \) is \( \text{argmax} \) of \( f(x) \), then it is also \( \text{argmax} \) of \( p(x) \), thus we have a high probability of selecting \( x_{\text{max}} \)
Markov Chain sampling

• To sample from a probability distribution \( p(x) \), we set up a Markov chain s.t. each state represents a value of \( x \) and for any two states, \( x \) and \( y \), the transitional probabilities satisfy:

\[
p(x) \Pr(x \rightarrow y) = p(y) \Pr(y \rightarrow x)
\]

• This would then imply that if the Markov chain is “run” for “long enough”, the probability thereafter of being in state \( x \) will be \( p(x) \)

\[
\lim_{N \to \infty} \frac{1}{N} C(x) = p(x)
\]
Gibbs sampling to maximize F

• Gibbs sampling is a special type of Markov chain sampling algorithm
• Our goal is to find the optimal $A = (a_1, \ldots a_N)$
• The Markov chain we construct will only have transitions from $A$ to alignments $A'$ that differ from $A$ in only one of the $a_i$
• In round-robin order, pick one of the $a_i$ to replace
• Consider all $A'$ formed by replacing $a_i$ with some other starting position $a'_i$ in sequence $S_i$
• Move to one of these $A'$ probabilistically
• Iterate the last three steps
Algorithm

Randomly initialize $A^0$;
Repeat:
(1) randomly choose a sequence $z$ from $S$;
$A^* = A^t \setminus a_z$; compute $\theta^t$ from $A^*$;
(2) sample $a_z$ according to $P(a_z = x)$, which is proportional to $Q_x/P_x$; update $A^{t+1} = A^* \cup x$;

Select $A^t$ that maximizes $F$;

$Q_x$: the probability of generating $x$ according to $\theta^t$;
$P_x$: the probability of generating $x$ according to the background model

$q_{ij} = \frac{c_{ij}}{\sum_k c_{ik}}$
Algorithm

Current solution $A^t$
Algorithm

Choose one $a_z$ to replace
For each candidate site $x$ in sequence $z$, calculate $Q_x$ and $P_x$:
Probabilities of sampling $x$ from motif model and background model resp.
Among all possible candidates, choose one (say $x$) with probability proportional to $Q_x/P_x$. 
Algorithm

Set $A^{t+1} = A^* \cup x$
Algorithm

Repeat

x

Repeat
Local optima

• The algorithm may not find the “global” or true maximum of the scoring function

• Once “A_t” contains many similar substrings, others matching these will be chosen with higher probability

• Algorithm will “get locked” into a “local optimum”
  – all neighbors have poorer scores, hence low chance of moving out of this solution