Probabilistic models of biological sequence motifs

Discovery of new motifs

Master in Bioinformatics UPF
2014-2015

Eduardo Eyras
Computational Genomics
Pompeu Fabra University - ICREA
Barcelona, Spain

Based on lectures by
Chris Burge
Mark Craven (http://www.biostat.wisc.edu/~craven/cs776.html
Enrique Blanco (http://rantamplan.bio.ub.es/~eblanco)
Pavel Pevzner (http://bioalgorithms.info)
Sequence Motifs

what is a sequence *motif*?

a sequence (or set of sequences) of biological significance (usually represented together as a pattern/matrix/etc)

Examples:

protein binding sites in DNA
protein sequences corresponding to common functions
conserved pieces of structure
Sequence Motifs

Pattern Matching:
search known motifs in a sequence (e.g. using a PWM)

Pattern Discovery:
search new motifs that are common to many sequences
(Expectation Maximization algorithm)
Biological Relevance of Motif finding:

Find a pattern or similar motif in a set of sequences that play a similar biological role can serve as evidence that this motif has a specific function

E.g.: promoter sequences of co-expressed genes in Yeast:
Motifs and Weight Matrices

Given a set of aligned sequences, we already know that we can construct a weight matrix to characterize the motif (see lecture on PWMs)
Motifs and Weight Matrices

how can we construct a weight matrix if the sequences are not “aligned”?

Moreover, in general we do not know what the motif looks like
But, what is the problem? Let’s consider the reverse situation

Consider 10 DNA sequences

1. atgaccgggatactgataaccgtattttgccctagggctacacatattagataaactgatagttagactcgccgagtacccgagcctgacatcctctggttaatagttgtttgtataaatggctacatctgttttccacgcaacatcgcaggaacccaaagcaagaccgataaaaggaga
2. acccctatattttttagcagagettagtacctggaatatttttttggcataatactgataacggtcataaaggtacagtacacattttttgtagctccttccgagttattttaaatattgtgttacatgtgcctcgcaggttaatattttccacgcaacatcgcaggaacccaaagcaagaccgataaaaggaga
3. tgggttctgctttgtaatgctgaggtacggttacgtaggaagcccaacacgcaatcgcaggaacccaaagcaagaccgataaaaggaga
4. gctgagaattggatgcctttgtaagttgttttccacgcaacatcgcaggaacccaaagcaagaccgataaaaggaga
5. tccctttttcggtaatattgctgccgaggctggttacgtaggaagcccaacacgcaatcgcaggaacccaaagcaagaccgataaaaggaga
6. gtcaaatcacttgtctttgtgagtagggcgctgcttacgtaggaagcccaacacgcaatcgcaggaacccaaagcaagaccgataaaaggaga
7. cgggtttggcctttctttgtagaggccccctctgctgtatgaggtggaatatttttccacgcaacatcgcaggaacccaaagcaagaccgataaaaggaga
8. aacctttaggttttgcagaaatgtcttgtgggcacatacagagaggtcttttaccatcagtatgtgcttgtgtagtttctctggtmggttaatagttgttttccacgcaacatcgcaggaacccaaagcaagaccgataaaaggaga
9. ttggccaataggctaaaggccccacttgacaaatggaagatagaccttgatctcggcttccggttttagctttcgtgcctggttctgtgatagca
10. ctggttagacagacagattccttaagctcgtgatcttggtggtctctcggcttccggttttagctttcgtgcctggttctgtgatagca
We implant the motif $\text{AAAAAAAGGGGGGG}$
Where is the implanted motif?

```
atgaccgggatactgataaaaaaaaggggagccgtacacattagataaaacgtatgaagtacgttagactgacggccg
acccctatatttttgacgcagatttagtgacctggaaaaaatgttgaatcaaaactttgtaataaaaaaaggggga
 tgagtatcccctgggatgacctaaaaaaaggggggtggtctctcccgatttttgaatatgtgagatcatttcgacagggtccga
 gctgagaattggatgaaaaaaaaggggggtccacgcaacgcgagccaccaagccagacccgataaaaggaga
 tccccctttgcggtaatgtgcccgggaggtctgggtacgtgaggaagccctataaccccttaacttattaaaaaaagggggtttatag
 gtaaatcatgttctttgtgatggattttaaaaaaaaggggggggaccgcttggcgcacccaaattccagtgtgtggtggcagcgcaag
 cgggttgggcttgggttagagggcccccgttaaaaaaagggggaattatgagagatgctaatctcattatcgcgctgtgtcttcat
 aacttgagttaaaaaaaaggggggctgggacatcaacaagaggtctttctttatcattatatgctgtatgacacatgtgta
 ttgggcccattggctaaagcccaacttgacaaatggaagatagaatccctatgaaaaaaaagggggggacccgaaaggggaag
 ctttgtgagccagcttattacgtgtacactgcatccgttccgggatctaatagcagacggagtttttaggggga
```
We implant the motif $AAAAAAGGGGGGG$ with four mutations
Where is the implanted motif now?

atgaccgggatactgatagagaagaaagttgaggctacacatttagataaactaagttatcagtttagacctgatggccgacggtcga
accctatatttttgacgagatttaggtacctggaatatagttaaactttaaaattttgaatatagtcattcctcaggttcgacg
ctgagatttgatgcaaaaaagggatttgtaaccgcagacacgcgaaccgacccaaagggcaagagacccgataaagagacgtcgggatgacttagttgaattggatgcaaaaaagggattgtccacagcaatccgagcagcttcaggggctggtgatgacgtgacgtctcctcaggtgacg
Why is motif finding difficult?

Motifs can mutate on non important bases, but these are a priori unknown
The Motif Finding Problem

Motif Finding Problem:

Given a list of $t$ sequences each of length $n$, find the “best” pattern of length $l$ that appears in each of the $t$ sequences.
The Motif Finding Problem

We use HEURÍSTICS: rules derived from our own experience that provide quick approximate solutions to complex problems.
The motif finding (pattern discovery) problem

INPUT:

N DNA sequences biologically related (e.g. promoters of co-expressed genes)

OUTPUT:

1. The motif: Representation (e.g. Weight matrix) of a motif common to all (or some of the) sequences. This can be any type of model

2. The positions: The best occurrence of the motif in each sequence
Motif finding (Iterative) Algorithms

while (convergence condition FALSE){
    # calculate model
    # modify model
    # evaluate new model
    # (convergence?)
    # go to next movement
}

If after many iterations, the “quality” of the solution does not improve, we say that given the INPUT, the program has converged to a solution (supposedly the best)
Before motif finding

How do we obtain a set of sequences on which to run motif finding? i.e. how do we get genes that we believe are regulated by the same transcription factor? E.g.

**Expression data:**
- Microarrays, RNA-Seq, ...
- Measures activity of thousands of genes in parallel under one or more conditions
- Collect set of genes with similar expression (activity) profiles and do motif finding on these.

**Protein binding data:**
- ChIP-on-chip, ChIP-Seq, RNA IP (RIP), CLIP, etc...
- Measure whether a particular factor binds to each of the sequences
- Provides hundreds or thousands of DNA/RNA target sequences
- Collect the set to which protein binds, do motif finding on these
Greedy Algorithm for Motif Search
• find a way to greedily change possible motif locations until we have converged to a highly likely hidden motif.
Greedy Algorithm

Consider t input sequences

Let $s = (s_1, ..., s_t)$ be the set of starting positions for $k$-mers in our $t$ sequences.

select arbitrary positions in the sequences

This defines $t$ substrings corresponding to these starting positions and will form:

- $t \times k$ alignment matrix and
- $4 \times k$ profile matrix $P$.

The profile matrix $P$ will be defined in terms of the frequency of letters, and not as the count of letters.
Greedy Algorithm

Profile matrix from random 6-mers in the $t$ input sequences:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1/2</td>
<td>7/8</td>
<td>3/8</td>
<td>0</td>
<td>1/8</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>1/8</td>
<td>0</td>
<td>1/2</td>
<td>5/8</td>
<td>3/8</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>1/8</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
<td>1/4</td>
<td>7/8</td>
</tr>
<tr>
<td>G</td>
<td>1/4</td>
<td>0</td>
<td>1/8</td>
<td>3/8</td>
<td>1/4</td>
<td>1/8</td>
</tr>
</tbody>
</table>

Positions in motif
Define the most probable \textit{k-mer} from each sequence using the built the profile \textbf{P}.

e.g. given a sequence = ctataaaccttacatc, find the most probable \textit{k-mer} according to \textbf{P}:

\begin{center}
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
  & A  & C  & T  & G  \\
\hline
A & 1/2 & 7/8 & 3/8 & 0  & 1/8  & 0  \\
C & 1/8 & 0   & 1/2 & 5/8 & 3/8  & 0  \\
T & 1/8 & 1/8 & 0   & 0  & 1/4  & 7/8 \\
G & 1/4 & 0   & 1/8 & 3/8 & 1/4  & 1/8 \\
\hline
\end{tabular}
\end{center}
Greedy Algorithm

Compute \( \text{prob}(a|P) \) for every possible 6-mer:

| String, Highlighted in Red          | Calculations                                   | \( \text{prob}(a|P) \) |
|-------------------------------------|------------------------------------------------|--------------------------|
| ctataaacctttacat                   | \( 1/8 \times 1/8 \times 3/8 \times 0 \times 1/8 \times 0 \) | 0                        |
| ctataaacctttacat                   | \( 1/2 \times 7/8 \times 0 \times 0 \times 1/8 \times 0 \) | 0                        |
| ctataaacctttacat                   | \( 1/2 \times 1/8 \times 3/8 \times 0 \times 1/8 \times 0 \) | 0                        |
| ctataaacctttacat                   | \( 1/8 \times 7/8 \times 3/8 \times 0 \times 3/8 \times 0 \) | 0                        |
| ctataaacctttacat                   | \( 1/2 \times 7/8 \times 3/8 \times 5/8 \times 3/8 \times 7/8 \) | .0336                    |
| ctataaacctttacat                   | \( 1/2 \times 7/8 \times 1/2 \times 5/8 \times 1/4 \times 7/8 \) | .0299                    |
| ctataaacctttacat                   | \( 1/2 \times 0 \times 1/2 \times 0 \times 1/4 \times 0 \) | 0                        |
| ctataaacctttacat                   | \( 1/8 \times 0 \times 0 \times 0 \times 0 \times 1/8 \times 0 \) | 0                        |
| ctataaacctttacat                   | \( 1/8 \times 1/8 \times 0 \times 0 \times 3/8 \times 0 \) | 0                        |
| ctataaacctttacat                   | \( 1/8 \times 1/8 \times 3/8 \times 5/8 \times 1/8 \times 7/8 \) | .0004                    |
Greedy Algorithm

Most Probable 6-mer in the sequence is aaacctt:

| String, Highlighted in Red | Calculations | $\text{Prob}(a|P)$ |
|----------------------------|--------------|-------------------|
| ctataaaacctttacat         | $1/8 \times 1/8 \times 3/8 \times 0 \times 1/8 \times 0$ | 0 |
| ctataaaacctttacat         | $1/2 \times 7/8 \times 0 \times 0 \times 1/8 \times 0$ | 0 |
| ctataaaacctttacat         | $1/2 \times 1/8 \times 3/8 \times 0 \times 1/8 \times 0$ | 0 |
| ctataaaacctttacat         | $1/8 \times 7/8 \times 3/8 \times 0 \times 3/8 \times 0$ | 0 |
| ctataaaacctttacat         | $1/2 \times 7/8 \times 3/8 \times 5/8 \times 3/8 \times 7/8$ | .0336 |
| ctataaaacctttacat         | $1/2 \times 7/8 \times 1/2 \times 5/8 \times 1/4 \times 7/8$ | .0299 |
| ctataaaacctttacat         | $1/2 \times 0 \times 1/2 \times 0 \times 1/4 \times 0$ | 0 |
| ctataaaacctttacat         | $1/8 \times 0 \times 0 \times 0 \times 1/8 \times 0$ | 0 |
| ctataaaacctttacat         | $1/8 \times 1/8 \times 0 \times 0 \times 3/8 \times 0$ | 0 |
| ctataaaacctttacat         | $1/8 \times 1/8 \times 3/8 \times 5/8 \times 1/8 \times 7/8$ | .0004 |
Greedy Algorithm

- Find the \( P \)-most probable \( k \)-mer in each of the sequences.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>T</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>P=</td>
<td>1/2</td>
<td>1/2</td>
<td>1/8</td>
<td>1/4</td>
</tr>
<tr>
<td></td>
<td>7/8</td>
<td>0</td>
<td>1/8</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3/8</td>
<td>1/2</td>
<td>0</td>
<td>1/8</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>5/8</td>
<td>0</td>
<td>3/8</td>
</tr>
<tr>
<td></td>
<td>1/8</td>
<td>3/8</td>
<td>1/4</td>
<td>7/8</td>
</tr>
<tr>
<td></td>
<td>1/8</td>
<td>1/8</td>
<td>3/8</td>
<td>1/4</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1/8</td>
<td>1/4</td>
<td>1/8</td>
</tr>
</tbody>
</table>

ctataacggttacatc
atagcgattcgactg
cagccccagaacccct
cgggtgaacctttacatc
tgcattcaatagctta
tgtcctgtccactcac
cuccaaatccttttaca
ggtctaccttttatcct
### Greedy Algorithm

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>7/8</th>
<th>3/8</th>
<th>0</th>
<th>1/8</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1/2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1/8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>1/8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>1/4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ctataaacgttacatc
atagcgattcgactg
cagcccagaacccct
cggtgaaccttacatc
tgcattcaatagctta
tgtcctgtccactcac
tcccaaatccttttaca
ggtctacctttatatcct
The most Probable $k$-mers define a new profile
**Greedy Algorithm**

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>a</th>
<th>a</th>
<th>c</th>
<th>g</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>c</td>
<td>g</td>
<td>t</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>t</td>
<td>a</td>
<td>g</td>
<td>c</td>
<td>g</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>t</td>
</tr>
<tr>
<td>4</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>c</td>
<td>t</td>
</tr>
<tr>
<td>5</td>
<td>a</td>
<td>t</td>
<td>a</td>
<td>g</td>
<td>c</td>
<td>t</td>
</tr>
<tr>
<td>6</td>
<td>g</td>
<td>a</td>
<td>c</td>
<td>c</td>
<td>t</td>
<td>g</td>
</tr>
<tr>
<td>7</td>
<td>a</td>
<td>t</td>
<td>c</td>
<td>c</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>8</td>
<td>t</td>
<td>a</td>
<td>c</td>
<td>c</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>ctataaaccgtttacatc</th>
<th>atagcgaattcgactg</th>
<th>cagccccagaaaccct</th>
<th>cgggtgaacctttacatc</th>
<th>tgcattcaatagcatta</th>
<th>tgtcctgtccactcac</th>
<th>ctccaaatccttttaca</th>
<th>ggtctacctttatatcct</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5/8</td>
<td>5/8</td>
<td>4/8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>4/8</td>
<td>6/8</td>
<td>4/8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>1/8</td>
<td>3/8</td>
<td>0</td>
<td>0</td>
<td>3/8</td>
<td>6/8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>2/8</td>
<td>0</td>
<td>0</td>
<td>2/8</td>
<td>1/8</td>
<td>2/8</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The **t** most Probable *k*-mers define a **new profile**
## Greedy Algorithm

<table>
<thead>
<tr>
<th>1</th>
<th>a</th>
<th>a</th>
<th>a</th>
<th>c</th>
<th>g</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>a</td>
<td>t</td>
<td>a</td>
<td>g</td>
<td>c</td>
<td>g</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>t</td>
</tr>
<tr>
<td>4</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>c</td>
<td>t</td>
</tr>
<tr>
<td>5</td>
<td>a</td>
<td>t</td>
<td>a</td>
<td>g</td>
<td>c</td>
<td>t</td>
</tr>
<tr>
<td>6</td>
<td>g</td>
<td>a</td>
<td>c</td>
<td>c</td>
<td>t</td>
<td>g</td>
</tr>
<tr>
<td>7</td>
<td>a</td>
<td>t</td>
<td>c</td>
<td>c</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>8</td>
<td>t</td>
<td>a</td>
<td>c</td>
<td>c</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>

A
- 1/2
- 7/8
- 3/8
- 0
- 1/8
- 0

C
- 0
- 1/2
- 5/8
- 3/8
- 0

T
- 1/8
- 1/8
- 0
- 0
- 1/4
- 7/8

G
- 1/4
- 0
- 1/8
- 3/8
- 1/4
- 1/8

Red – frequency increased,

Blue – frequency decreased
The Steps

1) Select random starting positions.
2) Create a profile $P$ from the substrings at these starting positions.
3) Find the most probable $k$-mer $a$ in each sequence and change the starting position to the starting position of $a$.
4) Compute a new profile based on the new starting positions after each iteration and proceed until we cannot increase the score anymore.
Greedy Algorithm

The pseudocode:

1. GreedyProfileMotifSearch (DNA, t, n, l)
2. Randomly select starting positions $s=(s_1,...,s_t)$ from DNA
3. bestScore $\leftarrow$ 0
4. while Score($s$, DNA) $>$ bestScore
5. Form profile $P$ from $s$
6. bestScore $\leftarrow$ Score($s$, DNA)
7. for $i \leftarrow 1$ to $t$
8. Find the most probable k-mer $a$ from the $i^{th}$ sequence
9. $s_i \leftarrow$ starting position of $a$
10. return bestScore
Greedy Algorithm

• We have described it here with a probability matrix, but a more correct description would be a weight matrix model (log-likelihood ratios).

• Note that the model could be other e.g., a Markov model of any order.

• Since we choose starting positions randomly, there is little chance that our guess will be close to an optimal motif, meaning it will take a long time to find the optimal motif.

• In practice, this algorithm is run many times with the hope that random starting positions will be close to the optimum solution simply by chance.
The Gibbs sampling algorithm
Gibbs Sampling

- **Gibbs Sampling** is an iterative procedure, similar to the Greedy algorithm, but it discards one *k-mer* after each iteration and replaces it with a new one.

- Gibbs Sampling proceeds more slowly than the Greedy algorithm. However, it chooses one new *k-mer* at each iteration at random (using the score distribution), which increases the chance of finding the correct solution.
Gibbs Sampling

1. Select a random position in each sequence
Gibbs Sampling

1. Select a **random** position in each sequence

Sequence set

motif instance

Can we make an educated guess?
Gibbs Sampling

2. Select a sequence at random
3. Build a weight matrix with the other sequences

\[ M_{\text{temp}} \]
Gibbs Sampling

4. Score possible sites in seq using weight matrix

Likelihood (probability)

\[ M_{\text{temp}} \]
Gibbs Sampling

5. Sample a new site proportional to likelihood

Likelihood (probability)

$M_{\text{temp}}$
Gibbs Sampling

6. Update weight matrix

Likelihood (probability)

$M'_{\text{temp}}$
7. Iterate until convergence (no change in sites or $M_{\text{temp}}$)
Given N sequences of length L and desired motif width W:

Step 1) Choose a starting position in each sequence at random: $a_1$ in seq 1, $a_2$ in seq 2, ..., $a_N$ in seq N

Step 2) Choose a sequence at random from the set (say, seq 1).

Step 3) Make a weight matrix model of width W from the sites in all sequences except the one chosen in step 2.

Step 4) Assign a probability to each position in seq 1 using the weight matrix model constructed in step 3: $p = \{ p_1, p_2, p_3, ..., pL-W+1 \}$

Step 5) Sample a starting position in seq 1 based on this probability distribution and set $a_1$ to this new position.

Step 6) Update weight matrix model

Step 7) Repeat from step 2) until convergence

Lawrence et al., Science 1993
Gibbs Sampling: an Example

**Input:**

\[ t = 5 \text{ sequences, motif length } l = 8 \]

1. GTAAACAATATTTATAGC
2. AAAATTTACCTCGCAAGG
3. CCGTACTGTCAAGCGTGG
4. TGAGTAAACGACGTCCCA
5. TACTTAACACCCTGTCAAA
Gibbs Sampling: an Example

1) Randomly choose starting positions, $s=(s_1, s_2, s_3, s_4, s_5)$ in the 5 sequences:

- $s_1=7$  GTAAACAATATTTATAGC
- $s_2=11$  AAAATTTACCTTAGAAGG
- $s_3=9$  CCGTACTGTCAGCGTG
- $s_4=4$  TGAAGTAACGAGCGTCCCA
- $s_5=1$  TACTTAAACTCCCTGATCAA
Gibbs Sampling: an Example

2) Choose one of the sequences at random:

**Sequence 2:** AAAATTACCTTAGAAGG

\[
\begin{align*}
  s_1 &= 7 & \text{GTAAACA} & \text{AATATTT} & \text{TATAGC} \\
  s_2 &= 11 & \text{AAAATTTAC} & \text{CCTTAGAAGG} \\
  s_3 &= 9 & \text{CCGTACTG} & \text{TCAAGCGTGG} \\
  s_4 &= 4 & \text{TGA} & \text{GTAAACGACGTCCCA} \\
  s_5 &= 1 & \text{TACTTAAC} & \text{ACCCTGTCAAA}
\end{align*}
\]
3) Build a motif description model with the remaining ones:

\[ s_1 = 7 \quad \text{GTAAACAATATTTATAGC} \]
\[ s_3 = 9 \quad \text{CCGTACTGTCAAGCGTGG} \]
\[ s_4 = 4 \quad \text{TGAGTAACGCAGTCCCA} \]
\[ s_5 = 1 \quad \text{TACTTTAACACCCTGTCAA} \]
Gibbs Sampling: an Example

Create model (e.g. profile $P$) from $k$-mers in remaining 4 sequences:

<table>
<thead>
<tr>
<th></th>
<th>1/A</th>
<th>2/A</th>
<th>3/T</th>
<th>4/A</th>
<th>5/T</th>
<th>6/T</th>
<th>7/T</th>
<th>8/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T</td>
<td>A</td>
<td>A</td>
<td>T</td>
<td>A</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>2</td>
<td>T</td>
<td>C</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>G</td>
<td>C</td>
<td>G</td>
</tr>
<tr>
<td>3</td>
<td>G</td>
<td>T</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>G</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>T</td>
<td>A</td>
<td>C</td>
<td>T</td>
<td>T</td>
<td>A</td>
<td>A</td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>T</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1/4</td>
<td>2/4</td>
<td>2/4</td>
<td>3/4</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1/4</td>
<td>1/4</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>2/4</td>
<td>1/4</td>
<td>1/4</td>
<td>1/4</td>
</tr>
<tr>
<td>G</td>
<td>1/4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Consensus String: T A A A T C G A
Gibbs Sampling: an Example

4) Calculate the $prob(a|P)$ for every possible 8-mer in the removed sequence:

| Strings Highlighted in Red                          | $prob(a|P)$ |
|-----------------------------------------------------|------------|
| AAAATTTTACCTTAGAAGG                                  | .0000732   |
| AAAATTTTACCTTAGAAGG                                  | .000122    |
| AAAATTTACCTTAGAAGG                                   | 0          |
| AAAATTTACCTTAGAAGG                                   | 0          |
| AAAATTTACCTTAGAAGG                                   | 0          |
| AAAATTTACCTTAGAAGG                                   | 0          |
| AAAATTTACCTTAGAAGG                                   | 0          |
| AAAATTTACCTTAGAAGG                                   | 0          |
| AAAATTTACCTTAGAAGG                                   | .0000183   |
| AAAATTTACCTTAGAAGG                                   | 0          |
| AAAATTTACCTTAGAAGG                                   | 0          |
| AAAATTTACCTTAGAAGG                                   | 0          |
| AAAATTTACCTTAGAAGG                                   | 0          |
5) Create a distribution of probabilities of \( k\)-mers \( \text{prob}(\mathbf{a} | \mathcal{P}) \), and randomly select a new starting position based on this distribution.

Define probabilities of starting positions according to ratios

Probability (Selecting Starting Position 1): \( \frac{P1}{P1 + P2 + P8} = 0.706 \)
Probability (Selecting Starting Position 2): \( \frac{P2}{P1 + P2 + P8} = 0.118 \)
Probability (Selecting Starting Position 8): \( \frac{P8}{P1 + P2 + P8} = 0.176 \)
Gibbs Sampling: an Example

Select the start position (in seq 2) according to computed ratios:

P(selecting starting position 1):     .706
P(selecting starting position 2):     .118
P(selecting starting position 8):     .176
Gibbs Sampling: an Example

We select the $k$-mers with the highest probability – then we are left with the following new $k$-mers and starting positions.

$\mathbf{s}_1 = 7$ \quad GTAAACA\textcolor{blue}{AATATTTATAGC} \\
$\mathbf{s}_2 = 1$ \quad \textcolor{red}{AAAATTTTA}\textcolor{blue}{CCTCGCAAGG} \\
$\mathbf{s}_3 = 9$ \quad CCGTACTG\textcolor{red}{TCAAGCGT}GG \\
$\mathbf{s}_4 = 4$ \quad TGAG\textcolor{red}{TAAACG}ACGTC\textcolor{blue}{CCA} \\
$\mathbf{s}_5 = 1$ \quad TACT\textcolor{red}{TTAAC}\textcolor{blue}{ACCCTGTCAAA}
Gibbs Sampling: an Example

6) Update the matrix: select one sequence and random and build the matrix with the rest, and repeat calculation.

7) We iterate until we cannot improve the score any more. Stop condition: overlap score does not change, relative entropy of the model (matrix), etc....
Gibbs Sampling: summary

Gibbs sampling needs to be modified when applied to samples with unequal distributions of residues.

Gibbs sampling often converges to locally optimal motifs rather than globally optimal motifs.

Not guaranteed to converge to same motif every time – Needs to be run with many randomly chosen seeds to achieve good results.
Gibbs Sampling: summary

Randomized algorithms make random rather than deterministic decisions.

The main advantage is that no input can reliably produce worst-case results because the algorithm runs differently each time.

These algorithms are commonly used in situations where no exact and fast algorithm is known.

Works for protein, DNA and RNA motifs.
The Expectation – Maximization (EM) algorithm
What we must specify:

$W$: Motif length

$M$: Motif description (probability matrix)

$B$: Background (probability vector)

$Z$: where the motif appears in the sequences
Motif Representation

A motif is assumed to have a fixed width, $W$

A motif is represented by a matrix of probabilities: $M_{ck}$ represents the probability of character $c$ in column $k$

This matrix $M$ estimates the probabilities of the nucleotides in positions likely to contain the motif

Example: DNA motif with $W=3$

$$M = \begin{array}{ccc}
1 & 2 & 3 \\
A & 0.1 & 0.5 & 0.2 \\
C & 0.4 & 0.2 & 0.1 \\
G & 0.3 & 0.1 & 0.6 \\
T & 0.2 & 0.2 & 0.1 \\
\end{array}$$
Background Representation

We also represent the “background” (i.e. sequence outside the motif) probability of each character

\[ B_c \] represents the probability of character \( c \) in the background

This vector \( B \) estimates the probabilities of the nucleotides in positions outside the likely motif

Example:

\[
B = \begin{matrix}
A & 0.26 \\
C & 0.24 \\
G & 0.23 \\
T & 0.27
\end{matrix}
\]
**Motif Position representation**

$Z_a$ is a location indicator for each sequence $S_a$

$Z_{aj} = \text{probability that the motif starts at position } j \text{ in sequence } S_a$

$$Z_{aj} = P(w_{aj} = 1 | M, B, S_a)$$

The elements $Z_{aj}$ of the matrix $Z$ are updated to represent the probability that the motif starts in position $j$ in sequence $a$

Example: given 4 DNA sequences of length 6, where $W=3$

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>seq1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>seq2</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>seq3</td>
<td>0.3</td>
<td>0.1</td>
<td>0.5</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>seq4</td>
<td>0.1</td>
<td>0.5</td>
<td>0.1</td>
<td>0.3</td>
<td></td>
</tr>
</tbody>
</table>

Normalized per sequence
EM Algorithm

given: length parameter $W$, training set of sequences

set initial values for $M$ and $B$
do

estimate $Z$ from $M$ and $B$ \hspace{1cm} (E–step)

estimate $M$ and $B$ from $Z$ \hspace{1cm} (M-step)

until change in $p <$ threshold

return: $B$, $M$, $Z$
EM Algorithm

Simplest model:

We will assume that each sequence $S_a$ contains a single instance of the motif and that the starting position of the motif is chosen uniformly at random from among the positions of $S_a$.

The positions of motifs in different sequences are independent.
we need to calculate the probability $P$ of a training sequence $S_a$ of length $L$, given a hypothesized starting position $j$:

$$P(S_a \mid w_{aj} = 1, M, B) = \prod_{i=1}^{j-1} B_{c_i} \prod_{i=j}^{j+W-1} M_{c_i,i-j+1} \prod_{i=j+W}^{L} B_{c_i}$$

before motif          motif          after motif

$S_a$ is one of the training sequences

$w_{aj}$ is 1 if motif starts at position $j$ in sequence $a$

$c_i$ is the character at position $i$ in the sequence $S_a$
EM Algorithm

we can also use a score using log-likelihood that the motif in sequence $S_a$ (of length $L$) starts at position $j$:

$$L(S_a \mid w_{aj} = 1, M, B) = \log \left( \prod_{i=j}^{j+W-1} M_{c_{i},i-j+1} \prod_{i=j}^{j+W-1} B_{c_{i}} \right) = \sum_{i=j}^{j+W-1} \log \left( \frac{M_{c_{i},i-j+1}}{B_{c_{i}}} \right)$$

$S_a$ is one of the training sequences

$w_{aj}$ is 1 if motif starts at position $j$ in sequence $a$

$c_i$ is the character at position $i$ in the sequence $S_a$
Initialization

W=3

We calculate M and B from the randomly selected start positions:

\[
\begin{align*}
A & : 0.25 & A & : 0.1 & 0.5 & 0.2 \\
B = C & : 0.25 & M = C & : 0.4 & 0.2 & 0.1 \\
G & : 0.25 & G & : 0.3 & 0.1 & 0.6 \\
T & : 0.25 & T & : 0.2 & 0.2 & 0.1
\end{align*}
\]

Remember:

\[S_a = \text{G C T G T A G}\]

Probability that the motif starts at position 3 in sequence \(S_a\):

\[
P(S_a \mid w_{a3} = 1, M, B) = B_G B_C M_{T,1} M_{G,2} M_{T,3} B_A B_G
\]

\[
= 0.25 \times 0.25 \times 0.2 \times 0.1 \times 0.1 \times 0.25 \times 0.25
\]
Initialization

W=3
We calculate M and B from the randomly selected start positions:

\[
\begin{array}{c|c|c|c}
   & 1 & 2 & 3 \\
\hline
A & 0.25 & A & 0.1 \quad 0.5 \quad 0.2 \\
B = & C & 0.25 & M = & C & 0.4 \quad 0.2 \quad 0.1 \\
G & 0.25 & G & 0.3 \quad 0.1 \quad 0.6 \\
T & 0.25 & T & 0.2 \quad 0.2 \quad 0.1 \\
\end{array}
\]

Remember:
\[S_a = \text{G C [T G T A G]}\]

The log-likelihood that the motif starts at position 3 in sequence \(S_a\):

\[
L(S_a \mid w_{a3} = 1, M, B) = \log \left( \frac{M_{T,1}}{B_T} \right) + \log \left( \frac{M_{G,2}}{B_G} \right) + \log \left( \frac{M_{T,3}}{B_T} \right)
\]

\[
= \log \frac{0.2}{0.25} + \log \frac{0.1}{0.25} + \log \frac{0.1}{0.25}
\]
Estimating $Z$ (E-step):

$$Z_{aj} = P(w_{aj} = 1 \mid M, B, S_a)$$

Probability that the motif starts at position $j$ in sequence $S_a$
Estimating $Z$ (E-step):

$$Z_{aj} = P(w_{aj} = 1 \mid M, B, S_a)$$

$$= \frac{P(S_a, w_{aj} = 1 \mid M, B)}{P(S_a \mid M, B)}$$

$$= \frac{P(S_a \mid w_{aj} = 1, M, B)P(w_{aj} = 1)}{\sum_{k=1}^{L-W+1} P(S_a \mid w_{ak} = 1, M, B)P(w_{ak} = 1)}$$

This follows from Bayes’ theorem

We assume that \textit{a priori} it is equally likely that the motif starts at any position of the sequence

Probability that the motif starts at position $j$ in sequence $S_a$

These are the probabilities that we were calculating before

$Z$ values must be normalized over the possible start positions
Estimating $Z$ (E-step):

The E-step computes the expected value of every $Z_{aj}$ using the matrix $M$ and vector $B$:

$$B = \begin{pmatrix}
A & 0.25 \\
C & 0.25 \\
G & 0.25 \\
T & 0.25 \\
\end{pmatrix} \quad \text{and} \quad M = \begin{pmatrix}
A & 1 & 2 & 3 \\
C & 0.1 & 0.5 & 0.2 \\
G & 0.4 & 0.2 & 0.1 \\
T & 0.3 & 0.1 & 0.6 \\
\end{pmatrix}$$

$$S_a = \text{G C T G T A G}$$

- $Z_{a1} = 0.3 \times 0.2 \times 0.1 \times 0.25 \times 0.25 \times 0.25 \times 0.25$
- $Z_{a2} = 0.25 \times 0.4 \times 0.2 \times 0.6 \times 0.25 \times 0.25 \times 0.25$
- $Z_{a3} = 0.25 \times 0.25 \times 0.2 \times 0.1 \times 0.1 \times 0.25 \times 0.25$
- $Z_{a4} = 0.25 \times 0.25 \times 0.25 \times 0.3 \times 0.2 \times 0.2 \times 0.25$
- $Z_{a5} = 0.25 \times 0.25 \times 0.25 \times 0.25 \times 0.2 \times 0.5 \times 0.6$

(before normalizing)
Estimating $Z$ (E-step):

The E-step computes the expected value of every $Z_{aj}$ using the matrix $M$ and vector $B$:

$$B = \begin{pmatrix} A & 0.25 \\ C & 0.25 \\ G & 0.25 \\ T & 0.25 \end{pmatrix} \quad M = \begin{pmatrix} A & B & C & D \\ 1 & 0.1 & 0.5 & 0.2 \\ 2 & 0.4 & 0.2 & 0.1 \\ 3 & 0.3 & 0.1 & 0.6 \end{pmatrix}$$

$S_a = G\ C\ T\ G\ T\ A\ G$

We can work with (log-likelihood ratio) scores:

$$Z_{a1} = \log\left(\frac{0.3}{0.25}\right) + \log\left(\frac{0.2}{0.25}\right) + \log\left(\frac{0.1}{0.25}\right)$$

$$Z_{a2} = \log\left(\frac{0.4}{0.25}\right) + \log\left(\frac{0.2}{0.25}\right) + \log\left(\frac{0.6}{0.25}\right)$$

$$Z_{a3} = \log\left(\frac{0.2}{0.25}\right) + \log\left(\frac{0.1}{0.25}\right) + \log\left(\frac{0.1}{0.25}\right)$$

$$Z_{a4} = \log\left(\frac{0.3}{0.25}\right) + \log\left(\frac{0.2}{0.25}\right) + \log\left(\frac{0.2}{0.25}\right)$$

$$Z_{a5} = \log\left(\frac{0.2}{0.25}\right) + \log\left(\frac{0.5}{0.25}\right) + \log\left(\frac{0.6}{0.25}\right)$$
Estimating $Z$ (E-step):

We **normalize**, dividing each value by the total sum of them:

$$Z_{aj} \rightarrow \frac{1}{Z_{a1} + \ldots + Z_{a5}} Z_{aj}$$

so that

$$\sum_{j=1}^{L-W+1} Z_{aj} = 1$$

$Z_{aj}$ represents now the normalized score/probability associated to the motif starts at $j$ in sequence $S_a$

For 3 sequences, using probabilities:

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$A \ C \ A \ G \ C \ A$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$A \ G \ G \ C \ A \ G$</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$T \ C \ A \ G \ T \ C$</td>
</tr>
</tbody>
</table>

Because the positions of motifs in different sequences are independent, the expectation of $Z_a$ depends only on the sequence $S_a$
Estimating $M$ and $B$ (M-step)

We want to estimate $M$ from $Z$, where $M_{ck}$ is the probability that character $c$ occurs at the $k^{th}$ position of the motif:

$$n_{ck} = \sum_{a} \sum_{j/S_a(j+k-1)=c} Z_{aj}$$

We sum every $Z_{aj}$ such that the character in the sequence $S_a$ at position $k^{th}$ of the motif is $c$.

Is the expected number of occurrences of the character $c$ at the $k^{th}$ position of the motif.

We obtain a probability by normalizing (on the characters) these values and using pseudocounts:

$$M_{ck} = \frac{n_{ck} + 1}{\sum_{b \in \{A, C, G, T\}} (n_{bk} + 1)}$$

$$\sum_{c \in \{A, C, G, T\}} M_{ck} = 1$$

pseudocounts
Estimating $M$ and $B$ (M-step)

\[ Z_{aj} = \begin{array}{cccc}
S_1 & 1 & 2 & 3 & 4 \\
S_2 & 0.1 & 0.7 & 0.1 & 0.1 \\
S_3 & 0.4 & 0.1 & 0.1 & 0.4 \\
S_3 & 0.2 & 0.6 & 0.1 & 0.1 \\
\end{array} \]

\begin{align*}
S_1 &= ACAAGCA \\
S_2 &= AGGCCAG \\
S_3 &= TCAGTCC
\end{align*}

We estimate the probabilities in $M$ and $B$, from the occurrences in $Z$:

E.g. estimate probability of finding an A in the first position of the motif:
Estimating $M$ and $B$ (M-step)

We estimate the probabilities in $M$ and $B$, from the occurrences in $Z$:

E.g. estimate probability of finding an A in the first position of the motif:
Estimating $M$ and $B$ (M-step)

We estimate the probabilities in $M$ and $B$, from the occurrences in $Z$:

E.g. estimate probability of finding an $A$ in the first position of the motif:

$$M_{A,1} \text{ is proportional to the (sum of) probabilities of the motif start positions in the sequences at positions with an } A$$
Estimating $M$ and $B$ (M-step)

We estimate the probabilities in $M$ and $B$, from the occurrences in $Z$:

E.g. estimate probability of finding an A in the first position of the motif:

Sum the probabilities of A at position 1 of motif in all sequences:

$$M_{A,1} = \frac{Z_{1,1} + Z_{1,3} + Z_{2,1} + Z_{3,3} + 1}{Z_{1,1} + Z_{1,2} + \ldots + Z_{3,3} + Z_{3,4} + 4}$$
Estimating *M and B* (M-step)

To calculate B from Z, we estimate the probability that each character occurs outside the motif.

\[ n_{c, bg} = n_c - \sum_{k=1}^{W} n_{ck} \]

Expected number of character c outside the motifs.

Recall that:
\[ n_{ck} = \sum_{a} \sum_{j/S_a(j+k-1)=c} Z_{aj} \]

We obtain a probability by normalizing (on the characters) these values and using pseudocounts:

\[ B_c = \frac{n_{c, bg} + 1}{\sum_{b} (n_{b, bg} + 1)} \]

\[ \sum_{c \in \{A,C,G,T\}} B_c = 1 \]
EM Algorithm (summarized)

0. Initialisation:
Select randomly a motif of $W$ positions in each sequence
Align the motifs to calculate for the first time the matrix $M$ and the vector $B$.

1. E step:
Use $M$ and $B$ to score each possible candidate segment (candidate_score).
For each sequence, normalize the candidate_scores with respect to all candidates of the same sequence
store the candidate segments and its scores

2. M step:
Update the matrices $M$ and $B$ using the best scores:
   - $M$: for each nucleotide, update its corresponding position in the matrix according to its occurrence in the candidates using the associated scores.
   - $B$: for each nucleotide outside a candidate, update the vector $B$

Normalize $M$ and $B$
Intuitively, the new motif model \((M,B)\) derives its character frequencies directly from the sequence data by counting the (expected) number of occurrences of each character in each position of the motif.

Because the location of the motif is uncertain, the derivation uses a weighted sum of frequencies given all possible locations, with more favorable locations getting more weight.

If a strong motif is present, repeated iterations of EM will hopefully adjust the weights to favor of the true motif start sites.
EM Algorithm

EM converges to a local maximum in the likelihood of the data given the model:

$$\prod_a P(S_a \mid B, M)$$

Usually converges in a small number of iterations
Sensitive to initial starting point (i.e. values in $p$)
MEME builds on the basic EM approach in the following ways:

• trying many starting points

• not assuming that there is exactly one motif occurrence in every sequence

• allowing multiple motifs to be learned

• incorporating Dirichlet prior distributions (instead of pseudocounts)
Starting Points in MEME

For every distinct subsequence of length $W$ in the training set

• derive an initial $p$ matrix from this subsequence

• run EM for 1 iteration

choose motif model (i.e. $p$ matrix) with highest likelihood

run EM to convergence
The ZOOPS Model

The approach outlined before, assumes that each sequence has exactly one motif occurrence per sequence; this is the OOPS model.

The ZOOPS model assumes zero or one occurrences per sequence.
The TCM Model

The TCM (two-component mixture model) assumes *zero or more* motif occurrences per sequence.
References

**Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids**
Cambridge University Press, 1999

**Problems and Solutions in Biological Sequence Analysis**
Mark Borodovksy, Svetlana Ekisheva
Cambridge University Press, 2006

**An Introduction to Bioinformatics Algorithms**
(Computational Molecular Biology)