Sequence motifs:
Representing, matching and discovering

Mikael Bodén
m.boden@uq.edu.au
<table>
<thead>
<tr>
<th>Chromosome</th>
<th>Start Position</th>
<th>End Position</th>
<th>Length</th>
<th>Average Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr3</td>
<td>45883549</td>
<td>45883824</td>
<td>1000</td>
<td>542.99</td>
</tr>
<tr>
<td>chr2</td>
<td>67877950</td>
<td>67878212</td>
<td>1000</td>
<td>486.30</td>
</tr>
<tr>
<td>chr4</td>
<td>163418813</td>
<td>163419060</td>
<td>1000</td>
<td>470.34</td>
</tr>
<tr>
<td>chr8</td>
<td>24151386</td>
<td>24151641</td>
<td>1000</td>
<td>464.89</td>
</tr>
<tr>
<td>chr12</td>
<td>25539128</td>
<td>25539379</td>
<td>1000</td>
<td>447.06</td>
</tr>
<tr>
<td>chr11</td>
<td>58941689</td>
<td>58941934</td>
<td>1000</td>
<td>439.78</td>
</tr>
<tr>
<td>chr21</td>
<td>178029389</td>
<td>178029617</td>
<td>1000</td>
<td>438.68</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
>('chr3', 45883549, 45883824)
TGGACTGCAATTATGCATTHTTTCATTGGTCCTCAGGATCACAACGCACTAGGATAT
TGCGTA
ACCGGGTTGACTGCACATGCACATTGGGTTCCTCCAGGGCGG

>('chr2', 67877950, 67878212)
CATTTGTAATTCTTTGACAAGGAGTGCAGATACAAATTGCGAATACAGAGAGCCATC
ACTCACCACAGCCCGCTGGAATTTCATGTGGAGCAACA

>('chr4', 163418813, 163419060)
TATTTCTTCATGTAAACCACATGATGGCAGCAAAAGGATAAGAGGACACAGGCATA
GCCATT
GCGCAATAAAAAAGCAAAAAACACAGTTATAGAATAAGTACA

>('chr8', 24151386, 24151641)
AGGACTTCCGCCTGCTGGGCCATTGTTGCAGGACCACAGCTTCGAGGTTGCGCAACACCT
TTAGTAAACACTTCCTGAATCAGAGACAAGAATGAACCAG

>('chr12', 25539128, 25539379)
ACACATCCGGCTCCCGATTGGCGGCGTAACCTCGCTTATTTGCATAGGCCGA
TTGCACT
ACCGGGCGGCACCTCAGCCCGGTGCTCGTGCCACGCCT
Short sequences with function: “motifs”

• Some functions are carried by only a few critical residues, e.g.
  – Active sites in enzymes
  – Post-translational modification sites
  – Localization signals in proteins (example 1)
  – Regulatory binding sites on DNA (example 2)

• Some experimental methods provide information about where signals in a sequence are, some only provide partial data; sometimes we rely on statistics alone

• We need methods that accurately represent, search for, and discover the occurrences of “sequence motifs”
A motivating example (1):
protein localization signals
**Signal peptides (SP)**
- Target proteins to the **secretory pathway** (the ER initially).
- 15-30 residues long.
- N-terminal domain positively charged, central hydrophobic region, small apolar residues -3, -1 relative cleavage site.

**Mitochondrial targeting peptides (mTP)**
- Target proteins to **mitochondria** (matrix, outer and/or inner membrane)
- 20-40 residues long.
- Arg, Ala and Ser overrepresented, while negative residues scarce.
- Weak conservation around cleavage site(s).

**Chloroplast transit peptides (cTP)**
- Target proteins to **chloroplast**.
- 20-120 residues long.
- Hydroxylated residues common, acidic uncommon.
- Semi-conserved motif around cleavage site.

**Peroxisomal targeting sequences (PTS)**
- Target proteins to the **peroxisome**
- Either a short C-terminal pattern or a weakly defined N-terminal pattern.

---

Nuclear localization signals

Importin α

NLS

ARM1
ARM2
ARM3
ARM4
ARM5
ARM6
ARM7
ARM8
ARM9
ARM10

Major Site

Minor Site

180°
A motivating example (2): transcription factor binding

- Transcription factors (TFs) bind to regulatory DNA sequences
- TFs modulate the efficiency by which RNA polymerase binds to DNA
- TF binding sites are usually described as “motifs”
TF-DNA binding
Constructing and visualising motifs

• By hand using expert knowledge, e.g. using protein structure to determine viable residues in a binding site, then construct/refine a multiple alignment accordingly

• By consensus, e.g. derive position-specific descriptor from a multiple alignment

• By automated discovery, e.g. use statistical analysis or machine learning on sequence data, e.g.
  – Over-represented “patterns”
  – Patterns associated with a “label”
GATA zinc fingers bound to DNA:
The C-terminal zinc finger of mouse GATA3 bound to DNA containing two variously arranged GATA binding sites.
Binding data (SELEX for GATA3)
Aligned binding data (GATA3)

s01  ---------GGGTCCGGGATAAGGTTCTTG------
s02  ---------GGGATGAGATTGAATCTGC------
s03  ---------AAAGGCAGATTGGGTTGT------
s04  ---------AGCATCCGATATTATCGTG------
s05  ---------ACTTTGAGATTGGGTTGT------
s06  ---------AGTGAGAACCAGCTTGTAGA-
s07  ---------AGGCCGAGATTGTGTGCGG------
s08  ---------AGATAAGCCCGCATGGTC------
s09  ---------AAACTGGCGGTACGAGATAT-------------
s10  ---------GGGCCGAGATTGTGTGCGG------
s11  ---------AGATAAGCCCGCATGGTC------
s12  ---------AGCCTAGCCCATACAGATAC-------------
s13  ---------TTGGCTCGATGCCATCTGCC------
s14  ---------GAGCTAAAGATTGTTGCTGC------
s15  ---------GGAGATAAAGCAGTGTATGC-
s16  ---------ATAGAGCGATATGGGGGTG------
s17  ---------TGTCGATGGTGCTGATTG------
s18  ---------ATAAGATAGGGGTGTG------
s19  ---------TGCCCGAGGAGATAGATGGT--------
s20  ---------AAGTGGCCGATAGGATTGCC------
s21  ---------TGGAGATAGGTCGTTTGTTT--
s22  ---------GGGATGAGATTGAATCTGCCG----
s23  ---------TGGATGATAAGGTTGGTGC----
s24  ---------GCTAATTGACTGATCGGGT----
s25  ---------GTCGGGCGTTGGTGTGATCG----
s26  ---------TGCCTAGCCCATACAGATAC-------------
s27  ---------TGCGCGAGATAGAGGAGCGC-----
s28  ---------TTGCATAGATAGAAGGGTGG-----
s29  ---------TACTCAGACACGCCAGTTAG-------------
s30  ---------ATAAGATAGGGGTGTG------
s31  ---------TGCCCGAGGAGATAGATGGT----
s32  ---------AAGTGAGAACCAGCTTGTAG-------
s33  ---------TGAGATAAGCCCGATGGCTGCT---------
s34  ---------TGAGATAAGCCCGATGGCTGCT---------
s35  ---------GGGATGAGATTGAATCTGC----
s36  ---------TGAGATAAGCCCGATGGCTGCT---------
s37  ---------GGGAGGAGATAGGGGTGT------
s38  ---------ATGCACGATATGTTGGTTA----
s39  ---------AGGGCTCTGATTAGCGTGGG------
s40  ---------TAACCTGGCGTTGCTGATGTA---------
s41  ---------AGGAGCGCTGATTAGCCGAT---------
s42  ---------CTTGGATGTGATAGGTTGGC-------
s43  ---------GCTTGGAGATAGAGGAGCGC------
s44  ---------TGATGGAGCTCTATCGATC---------
s45  ---------ATGCAGATATTGCTCCAGGT-------
s46  ---------CGACGGAACGATAGTGAGT---------
s47  ---------TGGGTTGGATAGGTCCAGG-------
s48  ---------TGAGATTGCTGATACAGGGT--------
s49  ---------GCTAATGATGAGTCAAGGCT-----
s50  ---------GTCAGGCGTTGGTGTGATCG---------
s51  ---------TGCCCGAGGAGATAGATGGT--------
s52  ---------TGCGCGATAGAGGAGCCC------
s53  ---------TTGCATAGATAGAAGGGTGG-----
s54  ---------TAACCTGGCGTTGCTGATGTA---------
s55  ---------GCTCGAGACGATGGTGCCTG-------
s56  ---------CACCTGGAACACTCATGATG--------
s57  ---------GGTGAGGAGGATGTTGGG-------
s58  ---------GGGAGGAGATAGGGGTGT------
p = IndepJoint([DNA_Alphabet for _ in range(aln.alignlen)])
for seq in aln.seqs:
    p.observe(seq)

p.displayMatrix()

    1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 ...
A  0.259 0.284 0.237 0.263 0.224 0.207 0.272 0.250 0.233 0.302 0.164 0.168 0.198 0.185 0.250 0.431 ...
C  0.259 0.216 0.237 0.280 0.259 0.241 0.185 0.233 0.284 0.198 0.216 0.168 0.129 0.289 0.250 0.224 ...
G  0.241 0.233 0.272 0.246 0.293 0.310 0.272 0.319 0.284 0.267 0.422 0.478 0.371 0.306 0.405 0.069 ...
T  0.241 0.267 0.254 0.211 0.224 0.241 0.272 0.198 0.198 0.233 0.198 0.185 0.302 0.220 0.095 0.276 ...

p.displayMatrix(count=True)

    1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 ...
A  15  16  13  15  13  12  15  14  13  17  9  9  11  10  14  25 ...
C  15  12  13  16  15  14  10  13  16  11  12  9  7  16  14  13 ...
G  14  13  15  14  17  18  15  18  16  15  24  27  21  17  23  4 ...
T  14  15  14  12  13  14  15  11  11  13  11  10  17  12  5  16 ...
Frequencies

\[ f_{u,a} = \frac{n_{u,a}}{N_{seq}} \]

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.25</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.05</td>
</tr>
<tr>
<td>C</td>
<td>0.10</td>
<td>0.15</td>
<td>0.00</td>
<td>0.00</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
<td>0.45</td>
</tr>
<tr>
<td>G</td>
<td>0.40</td>
<td>0.75</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.75</td>
<td>0.20</td>
</tr>
<tr>
<td>T</td>
<td>0.25</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.95</td>
<td>0.00</td>
<td>0.25</td>
<td>0.30</td>
</tr>
</tbody>
</table>

fu,a = nu,a

Nseq
Visualising motifs using “logo”

\[ H_u = \sum_a f_{u,a} \log_2 f_{u,a} \]  
(Shannon) entropy

\[ I_u = \log_2 | H_u | \]  
Information

\[ f_{u,a} I_u \]  
Frequency of residue
101001011001  1  \[0.5 \log_2 0.5 \quad 0.5 \log_2 0.5 = 1 \text{ bit}\]

111011011101  1  \[\frac{9}{12} \log_2 \frac{9}{12} \quad \frac{3}{12} \log_2 \frac{3}{12} = 0.81 \text{ bit}\]

001000010000  0  \[\frac{2}{12} \log_2 \frac{2}{12} \quad \frac{10}{12} \log_2 \frac{10}{12} = 0.65 \text{ bit}\]

**Entropy**

\[f_a \log_2 f_a\]

\[a \in A\]

where \(f_a\) is the frequency of a symbol \(a\) (from an alphabet \(A\)).

Wikipedia: Entropy is a measure of the information contained in a message, as opposed to the portion of the message that is predictable by inherent structures.
Nuclear localization signals

Importin α
ID    AKIR1_HUMAN       Reviewed;   192 AA.
AC    Q9H9L7; Q0VDB3; Q53FK8;
DT    06-FEB-2007, integrated into UniProtKB/Swiss-Prot.
DT    01-MAR-2001, sequence version 1.
DT    27-JUL-2011, entry version 60.
DE    RecName: Full=Akirin-1;
GN    Name=AKIRIN1; Synonyms=C1orf108;
OS    Homo sapiens (Human).
OC    Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
OC    Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini;
OC    Catarrhini; Hominidae; Homo.
OX    NCBI_TaxID=9606;...
DR    GO; GO:0005634; C:nucleus; IDA:UniProtKB.
PE    1: Evidence at protein level;
KW    Complete proteome; Nucleus; Phosphoprotein.
FT    CHAIN         1    192       Akirin-1.
FT    /FTId=PRO_0000274318.
FT    MOTIF        23     28       Nuclear localization signal.
FT    COMPBIAS     20     73       Pro-rich.
FT    MOD_RES      19     19       Phosphoserine.
FT    MOD_RES      22     22       Phosphoserine.
FT    MOD_RES      72     72       Phosphothreonine.
FT    CONFLICT     39     39       R -> G (in Ref. 1; BAD96996).
SQ    SEQUENCE   192 AA; 21867 MW; 21D7B2E6E026A802 CRC64;
       MACGATLKRP MEFEAALLSP GSPKRRRCAP LPGPTPGRLP PDAEPPPPFQ TQTTPQQSLQQ
       PAPPGSERRL PTEQIFQNI KQEYSRYQRW RHLEVVLNQS EACASESQPH SSALTAPSSP
       GSSWMKKDQP TFTLRQVGI2 CERLLKDYED KIREEYEQL NTKLAEQYES FVKFTHDQIM
       RRYGTRPTSY VS
//
ID AKIR1_HUMAN Reviewed; 192 AA.
AC Q9H9L7; Q0VDB3; Q53FK8;
DT 06-FEB-2007, integrated into UniProtKB/Swiss-Prot.
DT 01-MAR-2001, sequence version 1.
DT 27-JUL-2011, entry version 60.
DE RecName: Full=Akirin-1;
GN Name=AKIRIN1; Synonyms=C1orf108;
OS Homo sapiens (Human).
OC Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
OC Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini;
OC Catarrhini; Hominidae; Homo.
OX NCBI_TaxID=9606;...

DR GO; GO:0005634; C:nucleus; IDA:UniProtKB.
PE 1: Evidence at protein level;
KW Complete proteome; Nucleus; Phosphoprotein.
FT CHAIN 1 192 Akirin-1.
FT /FTId=PRO_0000274318.
FT MOTIF 23 28 Nuclear localization signal.
FT COMPBIAS 20 73 Pro-rich.
FT MOD_RES 19 19 Phosphoserine.
FT MOD_RES 22 22 Phosphoserine.
FT MOD_RES 72 72 Phosphothreonine.
FT CONFLICT 39 39 R -> G (in Ref. 1; BAD96996).

SQ SEQUENCE 192 AA; 21867 MW; 21D7B2E6E026A802 CRC64;
MACGATLKRP MEFEAALLSP GSPKRRCAP LPGPTPGLRP PDAEPPPFFQ TQTPPQSLQQ
PAPPGSERRL PTPEQIFQNI KQEYSRYQRW RHLEVVLNQS EACASESQPH SSALTAPSSP
GSSWMKKDQP TFLRQVGII CERLLKDYED KIREEYEQIL NTKLAEQYES FVKFTHDQIM
RRYGTRPTSY VS
//
| A | P | K | K | R | K | F |
| A4 | - | K | R | K | R | W |
| A24 | - | K | R | K | R | W |
| B113 | M | K | R | K | R | G |
| B31 | L | K | R | R | R | G |
| B161 | V | K | R | K | R | N |
| B199 | Y | K | R | K | R | D |
| B121 | T | K | R | K | R | A |
| B5 | L | K | R | K | R | G |
| B192 | K | K | R | K | R | D |
| B2 | G | K | R | K | R | L |
| B241 | G | K | R | K | R | L |
| B4 | R | K | R | K | R | E |
| B201 | R | K | R | K | R | D |
| B248 | R | K | R | K | R | D |
| B133 | R | K | R | K | R | D |
| B163 | R | K | R | K | R | S |
| B132 | R | K | R | K | W | D |
| B10 | R | K | R | K | L | A |
| A6 | R | K | R | R | R | E |
| B16 | G | K | R | R | R | R |

| B | K | 1 | 2 | 1 | 1 | 7 | 1 | 0 |
| R | 8 | 0 | 2 | 0 | 4 | 1 | 8 | 1 |
| S | 0 | 0 | 0 | 0 | 0 | 0 | 1 | .06 |
| L | 2 | 0 | 0 | 0 | 1 | 2 | .09 |
| P | 1 | 0 | 0 | 0 | 0 | 0 | .09 |
| I | 0 | 0 | 0 | 0 | 0 | 0 | .04 |
| G | 3 | 0 | 0 | 0 | 0 | 3 | .07 |

| D | RE1: [RG][K][R][KR][R][DG] |
| RE2: [K][R][KR][R] |

| E | K | -0.1 | +2.6 | -0.2 | +2.4 | -0.2 | -1.7 |
| R | +1.8 | -1.7 | +2.6 | +1.1 | +2.5 | -0.2 |
| S | -1.5 | -1.7 | -1.7 | -1.7 | -1.7 | -0.4 |
| L | +0.0 | -1.7 | -1.7 | -1.7 | -0.5 | +0.0 |
| P | -0.2 | -1.7 | -1.7 | -1.7 | -1.7 | -1.7 |
| I | -1.3 | -1.7 | -1.7 | -1.7 | -1.7 | -1.7 |
| G | +0.7 | -1.7 | -1.7 | -1.7 | -1.7 | +0.7 |

Models for finding motifs in sequences

- Plain frequencies fail to capture element of surprise of matches
- Alignments are not always obvious or even accessible
- We need a stronger theoretical framework to talk about “scoring” and “discovering” motifs
Probabilistic model

• A system that simulates the object or process under consideration
• Has parameters (that can be changed) to infer/assign different probabilities to possible outcomes
• Example model:
  – Process: a roll of a die
  – Outcomes: 1, 2, .., 6
  – Parameters: e.g. $\theta = p_1, p_2, ..., p_6$
  – (probabilities satisfy some conditions)
  – Inference: e.g. $P('1') = 1/6$
  – (model inference follows rules of probability)
Joint and conditional probability

Joint probability: \( P(X, Y) \)

*The probability of \( X \) and \( Y \)*

Conditional probability: \( P(X|Y) \)

*The probability of \( X \) given \( Y \)*
Joint and conditional probability

The joint probability for $X$ and $Y$ specifies the probability of each complete assignment of values for both $X$ and $Y$

Extends to any number of variables; we write

$$P(X, Y, Z, \ldots)$$

The joint probability function of a set of variables can be used to find a variety of other probability distributions, including marginal probability; e.g. we write

$$P(X, Y) = P(X, Y, Z)$$

A conditional probability can be calculated by taking the joint probability and dividing it by the marginal probability of one (or more) of the variables; we write

$$P(X|Y) = \frac{P(X, Y)}{P(Y)}$$
Conditional probability

Conditional probability can be defined as:

\[ P(X|Y) = \frac{P(X,Y)}{P(Y)} \]

\[ P(X,Y) = P(X|Y)P(Y) \]

Bayes’ rule can be used to reverse the roles of \( X \) and \( Y \):

\[ P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \]
Sequence models

• Observed biological sequences (DNA, RNA, protein) can be thought of as the outcomes of random processes
• It makes sense to model sequences using generative probabilistic models
• You can think of a probabilistic sequence model as a machine that randomly generates sequences
Finding model parameters

- Parameters can be estimated from sets of trusted examples, a “training set” \( D \)
- With large sets of fully observable examples, we expect measured frequencies to be reasonable estimates of underlying probabilities
- A model with parameters \( \theta \) and a set of data \( D \); the maximum likelihood estimate for \( \theta \) is the value that maximises \( P(D | \theta) \), i.e. the total probability of all data given a model
A simple sequence model

• Imagine a tetrahedral (four-sided) die with the letters A, C, G and T on its sides
• You roll the die 100 times and write down the letters that come up (down, actually)
• The frequencies of the outcomes are the maximum likelihood estimates
  – With these parameters the model has the best chance of re-producing the “training” sequence
Example

```python
>>> seq1 =
'CTGTCCGATATTGCAGCGTCTTAGTGTGAGAACGCTC
GCTGTATATCCGCAGGGAGATCCCCCTTCTTATTTCCT
TGGAAAGCATTTCACGACAGCAGTCC'

>>> p1 = Distrib(DNA_Alphabet)

>>> for s in seq1:
    p1.observe(s)

>>> print p1
< A=0.21  C=0.27  G=0.23  T=0.29 >
```
Example cntd

```python
>>> seq2 = ''
>>> for i in range(100):
    seq2 = seq2 + str(p1.generate())
>>> print seq2
AAGGACGACTACTGCGGTTTCTCCCCACGCAGATGCCCGCAGAGGCTAGTCCTAGTACTGTCAATAATACAGTGCTTTGGATTGCGTTTGTGGTC
>>> p2 = Distrib(DNA_Alphabet)
>>> for s in seq2:
    p2.observe(s)
>>> print p2
< A=0.21 C=0.26 G=0.25 T=0.28 >

>>> print p1
< A=0.21 C=0.27 G=0.23 T=0.29 >
>>> print seq1
CTGTCCGATATTGCAGCTCTTAGTGTGAGAACGCTCGCTGTATATCCGCAGGGAGATCCCCCTTCTTATTTCTTTGAAAGCATTTACGACAGCAGTCC
```
Position weight matrices (PWMs)

- A position specific probability matrix, \( q_{u,a} \) is the probability of a residue \( a \) at position \( u \)
- PRX-2 JASPAR Motif MA0075
- Constructed from observed binding sites
• Add pseudo-counts

• Frequency: $q_{u,a}$

• Background: $p_a$

• Ratio: $q_{u,a}/p_a$

• Log: $\log(q_{u,a}/p_a)$
Scoring motif \((i=1)\)

\[ S(i) = \log \frac{q_{u,R(u+i-1)}}{p_{R(u+i-1)}} \]

where \(R(i)\) is the residue index at sequence position \(i\), e.g. \(R(1)\) is ‘C’

\[ u = \begin{array}{ccccc}
    & 1 & 2 & 3 & 4 & 5 \\
    A & 1.852 & 2.031 & -3.876 & -3.876 & 2.007 \\
    G & -1.750 & -4.072 & -3.072 & -4.072 & -3.072 \\
\end{array} \]

\[ S(1) = -12.284 \]

\( i = 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \)
Scoring motif \((i=2)\)

\[
S(i) = \log \frac{q_{u,R(u+i-1)}}{p_{R(u+i-1)}u}
\]

\[u =
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 & S(2) \\
A & 1.852 & 2.031 & -3.876 & -3.876 & 2.007 & \textbf{-11.309} \\
G & -1.750 & -4.072 & -3.072 & -4.072 & -3.072 \\
\end{array}
\]

\[i = 1 2 3 4 5 6 7 8 9\]

\[S(i) = \begin{bmatrix}
A & 1.852 & 2.031 & -3.876 & -3.876 & 2.007 \\
C & -2.493 & -4.078 & -4.078 & -4.078 \\
G & -1.750 & -4.072 & -3.072 & -4.072 \\
T & -2.863 & -3.863 & 2.019 & 2.044 \\
\end{bmatrix}
\]
Scoring motif ($i=3$)

$$S(i) = \log \frac{q_{u,R(u+i-1)}}{\rho_{R(u+i-1)}}$$

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>S(3)=</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.852</td>
<td>2.031</td>
<td>-3.876</td>
<td>-3.876</td>
<td>2.007</td>
<td>5.238</td>
</tr>
<tr>
<td>C</td>
<td>-2.493</td>
<td>-4.078</td>
<td>-4.078</td>
<td>-4.078</td>
<td>-4.078</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-1.750</td>
<td>-4.072</td>
<td>-3.072</td>
<td>-4.072</td>
<td>-3.072</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>-2.863</td>
<td>-3.863</td>
<td>2.019</td>
<td>2.044</td>
<td>-3.863</td>
<td></td>
</tr>
</tbody>
</table>

C G T A T T A C G T ...

i = 1 2 3 4 5 6 7 8 9
Scoring motif \((i=4)\)

\[
S(i) = \sum_{u} \log \frac{q_{u,R(u+i-1)}}{p_{R(u+i-1)}},
\]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>S(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.852</td>
<td>2.031</td>
<td>-3.876</td>
<td>-3.876</td>
<td>2.007</td>
<td><strong>-7.945</strong></td>
</tr>
<tr>
<td>C</td>
<td>-2.493</td>
<td>-4.078</td>
<td>-4.078</td>
<td>-4.078</td>
<td>-4.078</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-1.750</td>
<td>-4.072</td>
<td>-3.072</td>
<td>-4.072</td>
<td>-3.072</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>-2.863</td>
<td>-3.863</td>
<td><strong>2.019</strong></td>
<td>2.044</td>
<td>-3.863</td>
<td></td>
</tr>
</tbody>
</table>

**CGTTATATTACGT...**

**i = 1 2 3 4 5 6 7 8 9**
Scoring motif

\[ S(i) = \log \frac{q_{u,R(u+i+1)}}{p_{R(u+i+1)}} \]
Problems with PWM scoring

• For long motifs PWMs are effective but long functional sites are unusual in DNA and RNA (note that proteins have “domains”)

• Most DNA binding motifs are quite short (6-12 residues) and short motifs lead to low recognition specificity:
  – Set the score threshold low enough to detect most known sites
    • Many incorrectly seen as true sites, i.e. false positives
    • In a typical genome, the vast majority of predictions are false positives (one false positive per 1-10 Kb)
  – Alternatively, set the score threshold high
    • Few true positives are found (low sensitivity)
Motif discovery

• A “profile” is a numeric representation (or “model”) of a sequence pattern, e.g. a set of *position-specific probability distributions*

• Discovery by enumeration—look at all possible
  – Count individual occurrences of sub-sequences, determine statistical enrichment (more if time)—can subsequently be used to determine a model

• Discovery by model—one size fits all
  – Find a probabilistic representation of a sub-sequence (e.g. profile) that maximises some objective based on data
  – We will start here...
General idea: Two (sets of) variables

- $Q$: model, e.g. profile or motif
- $Z$: label/s, e.g. match in a sequence

Discovery problem
- We would like to find the values for $Q$ and $Z$, that maximise $P(Q, Z)$
- From labels we can find a profile, i.e. we can compute $P(Q \mid Z)$, but we don’t know $Z$...
- With profile we can find label, i.e. we can compute $P(Z \mid Q)$, but we don’t know $Q$...
### General idea: Two variables

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.841</td>
<td>0.952</td>
<td>0.016</td>
<td>0.016</td>
<td>0.937</td>
</tr>
<tr>
<td>C</td>
<td>0.048</td>
<td>0.016</td>
<td>0.016</td>
<td>0.016</td>
<td>0.016</td>
</tr>
<tr>
<td>G</td>
<td>0.079</td>
<td>0.016</td>
<td>0.032</td>
<td>0.016</td>
<td>0.032</td>
</tr>
<tr>
<td>T</td>
<td>0.032</td>
<td>0.016</td>
<td>0.937</td>
<td>0.952</td>
<td>0.016</td>
</tr>
</tbody>
</table>

From labels we can find a profile
General idea: Two variables

With profile we can find label
Expectation Maximisation (EM)

- EM finds the values for $Q$ and $Z$ that maximise $P(Q, Z)$ given data, a joint probability distribution not known explicitly.

- EM iterates between
  - E-step: Estimate $P(Z \mid Q)$
  - M-step: Estimate $P(Q \mid Z)$
  - Both $Q$ and $Z$ improve gradually
  - Repeat until change in $Q$ is very small; deterministic and will converge but may get stuck in local optima.
Gibbs sampling

• Gibbs sampling samples from a joint probability distribution not known explicitly, e.g. P(Q, Z)
• Gibbs sampling is a stochastic analogue of EM:
  – Repeatedly samples from P(Q | Z) and P(Z | Q)
  – Typically data used for sampling are with-held from training
  – Q and Z improve gradually, but stochastically
  – Less susceptible to get stuck in local optima
  – Next-up an example: Find Q (means of two Gaussians); find Z (two labels of data points X)
Example: Find two $\mathcal{N}(\mu, \sigma = 1.00)$

<table>
<thead>
<tr>
<th>$X_0$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.9</td>
<td>0</td>
</tr>
<tr>
<td>3.3</td>
<td>1</td>
</tr>
<tr>
<td>4.3</td>
<td>1</td>
</tr>
<tr>
<td>6.1</td>
<td>0</td>
</tr>
<tr>
<td>6.4</td>
<td>0</td>
</tr>
<tr>
<td>7.5</td>
<td>1</td>
</tr>
<tr>
<td>8.1</td>
<td>1</td>
</tr>
<tr>
<td>8.8</td>
<td>1</td>
</tr>
</tbody>
</table>

1. Predictive update step: update model ($Q$) given current labeling ($Z$)
2. Sampling step: sample new labeling ($Z$) given model ($Q$)
3. Repeat from 1 until convergence
Hold-out $X_4 = 6.4$

Samples labeled $Z = 0$:
- $X_0 = 2.9$
- $X_3 = 6.1$
- New $P(Q|Z = 0) = 4.50;1.00$

Samples labeled $Z = 1$:
- $X_1 = 3.3$
- $X_2 = 4.3$
- $X_5 = 7.5$
- $X_6 = 8.1$
- $X_7 = 8.8$
- New $P(Q|Z = 1) = 6.40;1.00$

$X_4 = 6.4$: $P(Z|Q = 6.4) = <0.14 \ 0.86>$: $Z = 1$
Hold-out $X_1 = 3.3$

Samples labeled $Z = 0$:

$X_0 = 2.9$

$X_3 = 6.1$

New $P(Q|Z = 0) = 4.50;1.00$

Samples labeled $Z = 1$:

$X_2 = 4.3$

$X_4 = 6.4$

$X_5 = 7.5$

$X_6 = 8.1$

$X_7 = 8.8$

New $P(Q|Z = 1) = 7.02;1.00$

$X_1 = 3.3$: $P(Z|Q = 3.3) = <1.00 \ 0.00>$: $Z = 0$
Hold-out $X_3 = 6.1$

Samples labeled $Z = 0$:
- $X_0 = 2.9$
- $X_1 = 3.3$

New $P(Q|Z = 0) = 3.10;1.00$

Samples labeled $Z = 1$:
- $X_2 = 4.3$
- $X_4 = 6.4$
- $X_5 = 7.5$
- $X_6 = 8.1$
- $X_7 = 8.8$

New $P(Q|Z = 1) = 7.02;1.00$

$X_3 = 6.1$: $P(Z|Q = 6.1) = <0.02 \ 0.98>$: $Z = 1$
Hold-out $X_2 = 4.3$

Samples labeled $Z = 0$:
- $X_0 = 2.9$
- $X_1 = 3.3$

New $P(Q|Z = 0) = 3.10;1.00$

Samples labeled $Z = 1$:
- $X_3 = 6.1$
- $X_4 = 6.4$
- $X_5 = 7.5$
- $X_6 = 8.1$
- $X_7 = 8.8$

New $P(Q|Z = 1) = 7.38;1.00$

$X_2 = 4.3$: $P(Z|Q = 4.3) = <0.98 \ 0.02> : Z = 0$
| X_0  | \( P(Z|Q = 2.9) \) : \( Z = 0 \) |
| X_1  | \( P(Z|Q = 3.3) \) : \( Z = 0 \) |
| X_2  | \( P(Z|Q = 4.3) \) : \( Z = 0 \) |
| X_3  | \( P(Z|Q = 6.1) \) : \( Z = 1 \) |
| X_4  | \( P(Z|Q = 6.4) \) : \( Z = 1 \) |
| X_5  | \( P(Z|Q = 7.5) \) : \( Z = 1 \) |
| X_6  | \( P(Z|Q = 8.1) \) : \( Z = 1 \) |
| X_7  | \( P(Z|Q = 8.8) \) : \( Z = 1 \) |

1. Predictive update step: update model \((Q)\) given current labeling \((Z)\)
2. Sampling step: sample new labeling \((Z)\) given model \((Q)\)
3. Repeat from 1 until convergence
Motif discovery using Gibbs sampling

- We are given $N$ sequences $S_1, S_2, \ldots, S_N$
- We seek within each sequence mutually similar segments of specified width $W$, i.e. the most probable common pattern
- Initialise parameters (an alignment, i.e. a position in each sequence) randomly, then iterate
  - Predictive update step: update profile $(Q)$ given current alignment $(Z)$
  - Sampling step: sample new alignment $(Z)$ given profile $(Q)$
Motif discovery using Gibbs sampling

• Discovers a *motif* of length $W$ in a set of $N$ sequences

• Maintains two data structures:
  1. Iteratively updates $W+1$ residue distributions $q_{i,j}$ for residue $j \in \{A, C, G, T\}$ or $j \in \{A, R, N, \ldots\}$ in pattern position $i \in \{1, 2, \ldots, W\}$
     
     $p_j$ for a background distribution of residue $j$ (not inside the pattern)
  2. Keeps track of start position of pattern in each sequence $a_k$, $k \in \{1, \ldots, N\}$
  3. In its basic form the method assumes a single occurrence in each sequence
     • Provision for the absence or multiple presence of patterns exists
Motif discovery: Gibbs sampling

Step 0: Initialize sequence set

- Start with fixed $W$ and random set of locations $a_k$ in $N$ sequences $k \in \{1, \ldots, N\}$
Motif discovery: Gibbs sampling

**Step 1:** Select one sequence, look at the others

Predictive updating step:

a) Select a sequence $z$ at random

b) Determine $c_{i,j}$ is the number of times $j$ is found at position $a_{k+i}$ in the other $N-1$ sequences

c) Determine, $\pi_j$ is the whole dataset frequency of $j$, pseudo counts included by a weight $\beta$ (*often set to $\sqrt{N}$*)

\[
q_{i,j} = \frac{c_{i,j} + j}{N \cdot 1+}
\]
Motif discovery: Gibbs sampling

Step 1: Select one sequence, look at the others

Predictive updating step (cntd):

d) Background frequencies $p_j$ are updated analogously, using all positions not in the pattern:

$$m_{u,j}$$ is the number of times, a residue $j$ occurs in any position $u \not\in [a_k, a_k+W-1]$ in all sequences $k$ except $z$.

$M$ is the number of residues in all sequences.

$$p_j = \frac{m_{u,j} + \sum_j}{M}$$

![Diagram showing sequence patterns with motif highlighting]
Motif discovery: Gibbs sampling

Step 2: Adjust alignment in selected sequence

Sampling alignment step:

a) Calculate the motif likelihood for every stretch $x$ of $W$ residues in sequence $S_z$, starting at the $i$th residue:

b) $Q_x$ is obtained using $q_{ij}$: for every stretch $x$ of $W$ residues in $S_z$

c) $P_x$ is obtained using $p_j$: for every stretch $x$ of $W$ residues in $S_z$

d) $A_x$ is normalized to add up to 1.0 over the sequence (forming a distribution)

e) The new $a_z$ is chosen by random sampling over $A_x$

$$A_x \mu \frac{Q_x}{P_x}$$
Motif discovery: Gibbs sampling

**Step 3:** Go to 1 until convergence on a set of positions

- The procedure is repeated from step 1, until convergence (usually $100N$ times)
- Initially, selected positions are not more probable than any others, leading to exploratory (almost random) updating
- Later, correct positions (with patterns shared with the other sequences) will introduce a bias into updating, leading to convergence
Summary generative motif models

• Probabilistic models tolerate noise, but simple models face problems with inter-positional dependencies; complex models require large data sets, and/or assumptions

• Motif models can be found in training data, e.g. Lawrence et al. Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment, Science 262, 1993

• bioinf.scmb.uq.edu.au/pubs/vn/binfpy/src (gibbs.py)
Steps in a ChIP-seq experiment

- Cross-link proteins to DNA
- Fragment chromatin
- Immuno-precipitate with antibody to protein
- Size-select and ligate
- Amplify
- Sequence
ChIP-seq

- **Transcription factor binding sites** ("punctate narrow peaks")
- **Chromatin modifications** ("broad peaks")
How do we model pairs, triplets, etc. in observations?

• Joint probability \( P(X, Y) \)
  (of residues appearing next to one another)

• How determine?
  – by counting (we do not assume independence)
os.chdir('/Users/mikael/workspace/binf/data')
segs = readFastaFile('chipseq_2330.fa', DNA_Alphabet)

order = 0
j = Joint([DNA_Alphabet])
for s in segs:
    for i in range(len(s) - order):
        subseq = s[i:i + order + 1]
        j.observe(subseq)

for entry in j:
    print "%s \t %d \t %4.2f" % (str(entry), j.count(entry), j[entry])

('A',)    49557    0.21
('C',)    64098    0.28
('G',)    66011    0.28
('T',)    53334    0.23
order = 1
j = Joint([DNA_Alphabet, DNA_Alphabet])
for s in seqs:
    for i in range( len(s) - order):
        subseq = s[i:i + order + 1]
        j.observe(subseq)

for entry in j:
    print "%s \t %d \t %4.2f" % (str(entry), j.count(entry), j[entry])

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>('A', 'A')</td>
<td>9826</td>
<td>0.04</td>
</tr>
<tr>
<td>('C', 'A')</td>
<td>17692</td>
<td>0.08</td>
</tr>
<tr>
<td>('G', 'A')</td>
<td>13952</td>
<td>0.06</td>
</tr>
<tr>
<td>('T', 'A')</td>
<td>7612</td>
<td>0.03</td>
</tr>
<tr>
<td>('A', 'C')</td>
<td>11081</td>
<td>0.05</td>
</tr>
<tr>
<td>('C', 'C')</td>
<td>18620</td>
<td>0.08</td>
</tr>
<tr>
<td>('G', 'C')</td>
<td>19141</td>
<td>0.08</td>
</tr>
<tr>
<td>('T', 'C')</td>
<td>14589</td>
<td>0.06</td>
</tr>
<tr>
<td>('A', 'G')</td>
<td>19005</td>
<td>0.08</td>
</tr>
<tr>
<td>('C', 'G')</td>
<td>7158</td>
<td>0.03</td>
</tr>
<tr>
<td>('G', 'G')</td>
<td>19595</td>
<td>0.08</td>
</tr>
<tr>
<td>('T', 'G')</td>
<td>19617</td>
<td>0.09</td>
</tr>
<tr>
<td>('A', 'T')</td>
<td>9121</td>
<td>0.04</td>
</tr>
<tr>
<td>('C', 'T')</td>
<td>20016</td>
<td>0.09</td>
</tr>
<tr>
<td>('G', 'T')</td>
<td>12647</td>
<td>0.05</td>
</tr>
<tr>
<td>('T', 'T')</td>
<td>10998</td>
<td>0.05</td>
</tr>
</tbody>
</table>
```python
order = 2
j = Joint([DNA_Alphabet, DNA_Alphabet, DNA_Alphabet])
for s in seqs:
    for i in range(len(s) - order):
        subseq = s[i:i + order + 1]
        j.observe(subseq)

for entry in j:
    print "%s \t %d \t %4.2f" % (str(entry), j.count(entry), j[entry])

('A', 'A', 'A')  2182  0.01
('C', 'A', 'A')  2803  0.01
('G', 'A', 'A')  2891  0.01
('T', 'A', 'A')  1855  0.01
('A', 'C', 'A')  3883  0.02
('C', 'C', 'A')  5159  0.02
('G', 'C', 'A')  4845  0.02
('T', 'C', 'A')  3623  0.02
('A', 'G', 'A')  4620  0.02
('C', 'G', 'A')  1108  0.00
('G', 'G', 'A')  4269  0.02
('T', 'G', 'A')  3817  0.02
('A', 'T', 'A')  1901  0.01
('C', 'T', 'A')  2122  0.01
('G', 'T', 'A')  1499  0.01
('T', 'T', 'A')  2001  0.01
('A', 'A', 'C')  2292  0.01
('C', 'A', 'C')  4426  0.02
...
```

What is this a model of? What can this model be used for?
Enumerative method: motif enrichment

- ChIP-seq peaks are typically within +/- 50bp of the TF binding site.
- Regions centered on peaks should be enriched for binding motifs compared with flanking regions.
Discriminative motif discovery

• Detect motifs that are enriched in central regions vs. flanking regions
• Do this by counting the number of sequences containing a word in the two sets
• Apply Fisher’s Exact Test

<table>
<thead>
<tr>
<th>Word: CACACCC</th>
<th>Sequence Doesn’t Have Word</th>
<th>Sequence Has Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative Set</td>
<td>606</td>
<td>54</td>
</tr>
<tr>
<td>Positive Set</td>
<td>219</td>
<td>111</td>
</tr>
</tbody>
</table>

\( p\text{-value} = 9e^{-23} \)
Summary enumerative motif models

- Discrete models of sequence composition
- Sensitive to noise, and non-additive
- Space complexity during discovery is bad; limits the size of motifs that can be discovered