COSC 348: Computing for Bioinformatics

Lecture 7:
Sequence Motif Discovery

Lubica Benuskova

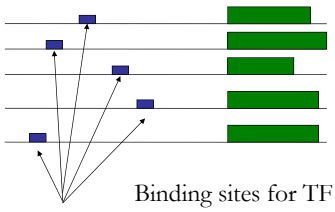
http://www.cs.otago.ac.nz/cosc348/

Sequence motif: definitions

- In Bioinformatics, a ***sequence motif*** is a nucleotide or amino-acid *sequence pattern* that is widespread and has been proven or assumed to have a biological significance.
- Once we know the sequence pattern of the motif, then we can use the search methods to find it in the sequences (i.e. Boyer-Moore algorithm, Rabin-Karp, suffix trees, etc.)
- The problem is to ***discover*** the motifs, i.e. what is the order of letters the particular motif is comprised of.

Examples of motifs in DNA

- The TATA *promoter* sequence is an example of a highly conserved DNA sequence motif found in eukaryotes.
- Another example of motifs: binding sites for transcription factors (TF) near promoter regions of genes, etc.



Sequence motif: notations

- An example of a motif in a protein: N, followed by anything but P, followed by either S or T, followed by anything but P
  - One convention is to write **`N{P}[ST]{P}`** where **`{X}`** means any amino acid except **`X`**; and **`[XYZ]`** means either **`X`** or **`Y`** or **`Z`**.
- Another notation: each '.' signifies any single AA, and each '*' indicates one member of a closely-related AA family:
  - **`WDIND*.*P..*...D.F.*W***.**.IYS**...A.*H*S*WAMRN`**
- In the 1st assignment we have motifs like **`A??CG`**, where the wildcard **`?`** Stands for any of **`A,U,C,G`**.

Sequence motif discovery from conservation

- Sequence motifs are *conserved sequences* of similar or identical *patterns* that may occur within nucleic acids (DNA, RNA) or proteins either
  - within different molecules produced by the same organism or
  - within molecules from multiple species of organisms
- In the case of cross-species conservation, conserved motif indicates that a particular sequence pattern may have been conserved during evolution to perform certain function, thus
  - Motif conservation is the basis of motif discovery by studying similar genes (or proteins) in different species;
  - A motif discovery program that considers phylogenetic conservation is named PhyloGibbs.

Motif discovery based on alignment

- *profile analysis* is another word for this. This is usually done by
  - first constructing a **local alignment** of multiple sequences,
  - after which the **highly conserved regions are isolated,** based on their high alignment scoring

Protein ID		Conserved region
HEM13	CCCATTGTTCTC	
HEM13	TTTCTGGTTCTC	
HEM13	TCAATTGTTTAG	
ANB1	CTCATTGTTGTC	
ANB1	TCCATTGTTCTC	
ANB1	CCTATTGTTCTC	
ANB1	TCCATTGTTCGT	
ROX1	CCAATTGTTTTG	

Motif discovery based on alignment

- After the highly conserved regions are isolated, they are used to construct profile matrices for each conserved region.
- The **profile matrix** for a given motif contains frequency counts for each letter at each position of the isolated conserved region.

```
A 002700000010
C 464100000505
G 000001800112
T 422087088261
```


Sequence logo and consensus sequence

- We can extract the so-called **consensus sequence**, i.e. the string of most frequent letters:

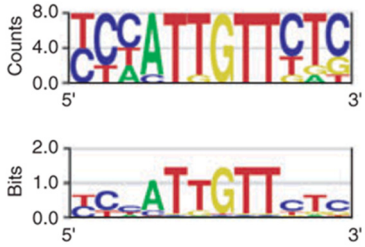
YCHATTGTTCTC

- A graphical representation of the consensus sequence is called a *sequence logo*:

- The height of different letters at the same position is proportional to their frequency in motifs: the better the base conservation is at that position, the higher the letters will be.

Sequence logo and information theory

- Sequence logo is often displayed using the information theory, i.e.



$$H = -\sum_{i=1}^{n} P(x_i)\log_2 P(x_i)$$

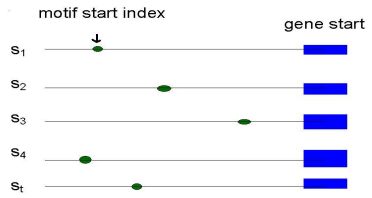
Shannon's information

- Shannon's information (in bits) is a measure of the information content $I(x_i)$ associated with the particular outcome x_i of a random variable X, which can have n values/outcomes, i.e. $x_1, x_2, \ldots x_n$
- $$I(x_i) = -\log_2 P(x_i) = \log_2 (1 / P(x_i))$$
  - When choosing from 4 nucleotides, the probability $P(x_i) = 1/4$. When particular nucleotide occurred, the amount of information is
  - I('nucleotide') = $\log_2 (1/(1/4)) = \log_2 (4) = 2$ bits.
- If the possible values x_i of variable X have probabilities $P(x_i)$ then *Shannon's expected information (entropy)* is:

$$H = -\sum_{i=1}^{n} P(x_i)\log_2 P(x_i)$$

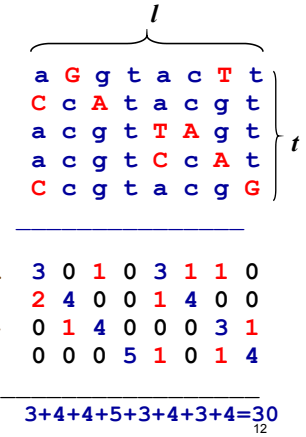
Discovering motifs without alignment

- First, let's assume *we know* where the motif starts in the reference sequence set.
- The motif start positions in sequences can be represented as the set $\boldsymbol{s} = (s_1, s_2, s_3, \ldots, s_t)$ where s_i is the position index



Scoring motifs

- Given $\boldsymbol{s} = (s_1, \ldots s_t)$, we align t motifs of length l from all sequences
- Construct profile matrix
- *Motif Score*($\boldsymbol{s}$) = $\sum_{i=1}^{l} \max_{k \in \{A,T,C,G\}} [count(k,i)]$



Motif finding problem

- The problem is to find the starting positions $s = (s_1, \ldots s_t)$ to maximize the Score(s) of the resulting profile matrix.
- Several kinds of profile matrices are used:
  - A **position frequency matrix** (PFM) records the position-dependent frequency f of each letter, i.e. how many times a letter occurs at a given position in N sequences.
  - A **position probability matrix** (PPM). When normalized to 1 this frequency turns into a probability, i.e. $P = f / N$.
  - A **position weight matrix** (PWM):

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

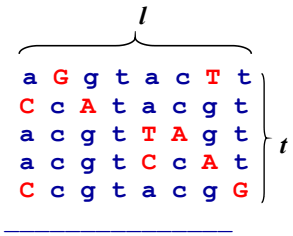
Let's start again

- 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.
- Let $\mathbf{s} = (s_1,\ldots,s_t)$ be the set of starting positions for l-mers in our t sequences.
- The strings corresponding to these starting positions will form:
  - 4 x l ***profile matrix**** **P** for DNA
  - and 20 x l ***profile matrix**** **P** for proteins

  * The profile matrix will be defined in terms of the ***probability*** of letters, and not as the count of letters.

Profile P

- Given **s** = (s_1, … s_t), we align t l-mers from all sequences and
- Construct the profile **P**



A	0.6	0.0	0.2	0.0	0.6	0.2	0.2	0.0
C	0.4	0.8	0.0	0.0	0.2	0.8	0.0	0.0
G	0.0	0.2	0.8	0.0	0.0	0.0	0.6	0.2
T	0.0	0.0	0.0	1.0	0.2	0.0	0.2	0.8

Probability of l-mers

- $Pr(\mathbf{a}|\mathbf{P})$ is defined as the probability that an l-mer **a** was created by the Profile **P**.
- If **a** is very similar to the consensus string (i.e. motif) then $Pr(\mathbf{a}|\mathbf{P})$ will be high.
- If $P_{ai,k}$ is the probability of letter a_i at position k, then the probablity of an l-mer **a** is equal to the product of individual probabilities $P_{ai,k}$, i.e.:

$$\Pr(\mathbf{a} \mid \mathbf{P}) = \prod_{k=1}^{l} P_{a_i,k}$$

Scoring l-mers with a profile

Given a profile **P** =

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

$Pr(\mathbf{aaacct}|\mathbf{P})$ = ???

Scoring l-mers with a profile (cont'd)

Given a profile **P** =

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

$Pr(\mathbf{aaacct}|\mathbf{P})$ = 1/2 x 7/8 x 3/8 x 5/8 x 3/8 x 7/8 = .033646

Scoring *l*-mers with a profile (cont'd)

Given a profile: **P** =

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**

Prob(**aaacct**|**P**) = 1/2 x 7/8 x 3/8 x 5/8 x 3/8 x 7/8 = .033646

Prob(**atacag**|**P**) = 1/2 x 1/8 x 3/8 x 5/8 x 1/8 x 1/8 = .001602

Motif – the P-most probable *l*-mer

- Define the **P**-most probable *l*-mer from a sequence as an *l*-mer in that sequence which has the highest probability of being created from the profile **P**.
- Task: given a sequence **ctataaaccttacatc** and the known profile **P**, find the P-most probable 6-mer:

P =

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

P-most probable *l*-mer (cont'd)

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

First try: c t a t a a a c c t t a c a t c

Second try: c t a t a a a c c t t a c a t c

Third try: c t a t a a a c c t t a c a t c

Slide the window to evaluate every possible 6-mer – brute force approach

P-most probable *l*-mer (cont'd)

Compute *Pr*(**a**|**P**) for every possible 6-mer:

Window, Highlighted Red	Calculations	*Pr*(**a**|**P**)
ctataaaccttacat	1/8 x 1/8 x 3/8 x 0 x 1/8 x 0	0
ctataaaccttacat	1/2 x 7/8 x 0 x 0 x 1/8 x 0	0
ctataaaaccttacat	1/2 x 1/8 x 3/8 x 0 x 1/8 x 0	0
ctataaaaccttacat	1/8 x 7/8 x 3/8 x 0 x 3/8 x 0	0
ctataaaaccttacat	1/2 x 7/8 x 3/8 x 5/8 x 3/8 x 7/8	.0336
ctataaaaccttacat	1/2 x 7/8 x 1/2 x 5/8 x 1/4 x 7/8	.0299
ctataaaccttacat	1/2 x 0 x 1/2 x 0 1/4 x 0	0
ctataaaccttacat	1/8 x 0 x 0 x 0 x 0 x 1/8 x 0	0
ctataaaccttacat	1/8 x 1/8 x 0 x 0 x 3/8 x 0	0
ctataaaccttacat	1/8 x 1/8 x 3/8 x 5/8 x 1/8 x 7/8	.0004

P-most probable *l*-mer (cont'd)

P-most probable 6-mer in the sequence is **aaacct**:

Window, Highlighted Red	Calculations	*Pr*(**a**|**P**)
ctataaaccttacat	1/8 x 1/8 x 3/8 x 0 x 1/8 x 0	0
ctataaaccttacat	1/2 x 7/8 x 0 x 0 x 1/8 x 0	0
ctataaaaccttacat	1/2 x 1/8 x 3/8 x 0 x 1/8 x 0	0
ctataaaaccttacat	1/8 x 7/8 x 3/8 x 0 x 3/8 x 0	0
ctataaaaccttacat	**1/2 x 7/8 x 3/8 x 5/8 x 3/8 x 7/8**	**.0336**
ctataaaaccttacat	1/2 x 7/8 x 1/2 x 5/8 x 1/4 x 7/8	.0299
ctataaaccttaacat	1/2 x 0 x 1/2 x 0 1/4 x 0	0
ctataaaccttacaat	1/8 x 0 x 0 x 0 x 0 x 1/8 x 0	0
ctataaaccttacaat	1/8 x 1/8 x 0 x 0 x 3/8 x 0	0
ctataaaccttacaat	1/8 x 1/8 x 3/8 x 5/8 x 1/8 x 7/8	.0004

Dealing with zeroes and small probabilties

- In our toy example *Pr*(**a**|**P**) = 0 in many cases. In practice, there will be enough sequences so that the number of elements in the profile with a frequency of zero is likely to be small but still we must ensure zeroes are taken care of.
- There exist several techniques to equate zero to a very small number so that one zero does not make the entire probability of a string zero.
  - The simplest one is to replace 0 with a small number, e.g. $1 / 10n$.
- Another problem is that the product of small probabilities is a very small number. Thus, we replace the product with the sum of logarithms:

$$\Pr(\mathbf{a} \mid \mathbf{P}) = \prod_{k=1}^{l} P_{a_i,k} \Rightarrow \log \Pr(\mathbf{a} \mid \mathbf{P}) = \sum_{k=1}^{l} \log\left(P_{a_i,k}\right)$$

P-most probable *l*-mers are motifs

- Task: Find the **P**-most probable *l*-mer in each of the sequences given profile **P**.
- The **P**-most probable *l*-mer is our motif.
- How do we find **P**?

```
ctataaacgttacatc
atagcgattcgactg
cagcccagaaccct
cggtataccttacatc
tgcattcaatagctta
tatcctttccactcac
ctccaaatcctttaca
ggtcatcctttatcct
```


Finding the profile **P** iteratively

1	a	a	a	c	g	t
2	a	t	a	g	c	g
3	a	a	c	c	c	t
4	g	a	a	c	c	t
5	a	t	a	g	c	t
6	g	a	c	c	t	g
7	a	t	c	c	t	t
8	t	a	c	c	t	t
A	5/8	5/8	4/8	0	0	0
C	0	0	4/8	6/8	4/8	0
T	1/8	3/8	0	0	3/8	6/8
G	2/8	0	0	2/8	1/8	2/8

```
ctataaacgttacatc
atagcgattcgactg
cagcccagaaccct
cggtgaaccttacatc
tgcattcaatagctta
tgctctgtccactcac
ctccaaatccttttaca
ggtctaccttatcct
```

- Let $l = 6$. Start at random positions (underlined) and calculate the initial profile **P.**

Finding the profile **P** iteratively

1	a	a	a	c	g	t
2	a	t	a	g	c	g
3	a	a	c	c	c	t
4	g	a	a	c	c	t
5	a	t	a	g	c	t
6	g	a	c	c	t	g
7	a	t	c	c	t	t
8	t	a	c	c	t	t
A	5/8	5/8	4/8	0	0	0
C	0	0	4/8	6/8	4/8	0
T	1/8	3/8	0	0	3/8	6/8
G	2/8	0	0	2/8	1/8	2/8

Use this initial profile to find the **P**-most probable *l*-mer in each sequence, and set the new starting positions according to the beginnings of **P**-most probable *l*-mer in each sequence.

Comparing new and old profiles

- **P**-most probable *l*-mers form *a new profile* by recalculating probabilities at all the positions

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

- According to this new profile **P**, find the most probable *l*-mers in each sequence, set new positions and re-iterate.

Algorithm for greedy profile motif search

Use **P**-most probable *l*-mers to adjust new start positions until we reach the "best" profile; this will be pronounced as the motif.

Select random starting positions, then:

1. Create a profile **P** from the *l*-mers at these starting positions.
2. Find the **P**-most probable *l*-mer **a** in each sequence and change the starting positions to the starting positions of **a'**s.
3. Go to step 1 and re-iterate until we cannot increase the score anymore.

Summary of greedy motif discovery

- 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 very 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 algorithm may be improved by heuristic knowledge, where approximately we should start or by more sophisticated statistical techniques, like *Gibbs sampling* that estimates the most probable start positions for motifs, where we ought to start our iterative process of motif discovery.