Motif Discovery

The Challenge

We are given a bunch of sequences and we wish to discover motifs embedded in sequences.

- There may be more or less than one motif in each sequence
- We don't know the location of the motifs within the sequences
- We don't know the composition of the motif
- We may or may not know the length of the motif
- There may be gaps in some of the motifs

Complexity

This is a (really) hard problem.

In fact, we have to settle for an heuristic, and might even have to settle for a greedy algorithm

This is NOT MSA

The content of each word to be aligned in an MSA is known (up to gap info)

Motif-finding does not know the content- it must discover it

Motif Elicitation
Standard Approaches

- EM
- HMM
- Home-spun

EM: A strategy

- We will discuss the EM approach
- For simplicity, we are going to constrain the problem:
  - *Given:* exactly one motif/sequence
  - *Given*: no gaps
  - *Given*: length is fixed, and known
  - *Given*: background sequence, as distinct from motif, will be identified. Restated, we will be *given* sequence that we know is ***not in the motif.***

Features of the Motif

When we find the motif, the SNR (motif to background) will be maximized

Restated: The motif will be unlikely in the context of its background

But we don’t know what content of the motif is, so how do we determine if it is unlikely?

If we know that the motif is, say, ATCC, and we know that the spelling in each sequence is identical, we could launch one of our many exact string match algorithms that we know about. The problem complexity would be no worse than quadratic, and could certainly be linear, given the right algorithm.

But consider these 6 sequences with embedded motifs (in red)

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

Well, certainly the consensus sequence is ATCC, but all the instances of the motif do not have the same spelling, nor do we even know the consensus *a priori,* so string matching, with or without wildcards, is not an option.

Profiles (PSSM)

A profile is a Position-Specific Scoring Matrix (PSSM)

The PSSM is a matrix with a column for the position of each letter in the motif, and a row for the frequency (probability) of each possible letter in that position, in the context of the alphabet (4 rows for DNA, 20 rows for amino acids).

If you think about it, each column is the pdf for the letters in that column

The background has but one column, since there is no concept of position

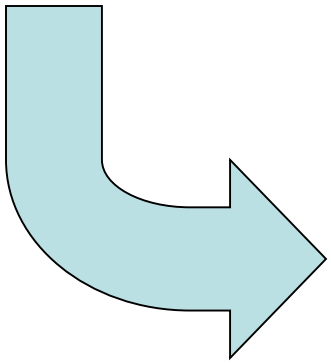
EXAMPLE OF A PSSM

Consider constructing a PSSM using our sample data, starting in position 2 of the sequence data

1	2	3	4	5	6	7	8	9	10	11	12
A	A	T	C	C	C	C	G	T	C	G	T
A	T	C	A	T	G	C	A	G	C	T	A
T	C	T	C	G	A	T	T	C	C	G	A
C	T	A	A	C	C	G	G	T	G	A	C
A	G	A	T	C	C	T	G	C	A	C	T
C	A	G	G	A	T	C	C	T	G	T	C

Sequence position

1 2 3 4 PSSM Position



	*Pos*1	*Pos*2	*Pos*3	*Pos*4
A	.334	.334	.334	.167
C	.167	.167	.334	.5
G	.167	.167	.167	.167
T	.334	.334	.167	.167

Background

A	.27
C	.25
G	.23
T	.25

The object is to determine the pdf of the motif

The pdf of the motif will be the most probable subsequence in the data.

Restated, the motif pdf will look exactly like the correct PSSM. (But we don't know what the 'correct' PSSM looks like, only how we guess when we start.)

Another viewpoint: the 'correct' pdf that is the one farthest from the psd of the background. (We do not know the psd farthest from the background.)

A Strategy

We are going to use EM to arrive at the correct PSSM

- We will take some initial PSSM and, based on its initial contents, compute the joint probability that the motif would start in each position of the sequence data.

- When we are done, we will use those probabilities to maximize the PSSM, but more about that later

- Our example will be with DNA, but this works on other alphabets, such as amino acids

Expectation Phase Recipe 1

Set a joint probability expression=1

Start in position 1 of the sequence data.

- If the first letter of the first sequence is an A, multiply the joint probability by the probability of an A as the first letter of a the motif, decided by the probability in the first position of the PSSM corresponding to an A
- If the first letter of the first sequence is a C, multiply the joint probability by the probability of a C as the first letter of a the motif, decided by the probability in the first position of the PSSM corresponding to an C in the first position of the PSSM
- If the first letter of the first sequence is a G, multiply the joint probability by the probability of a G as the first letter of a the motif, decided by the probability in the first position of the PSSM corresponding to an G
- If the first letter of the first sequence is a T, multiply the joint probability by the probability of a T as the first letter of a the motif, decided by the probability in the first position of the PSSM corresponding to an T

Expectation Phase Recipe-2

Look at the second position of the PSSM

Still in position 1 of the sequence data:

- If the 2nd letter of the first sequence is an A, multiply the joint probability by the probability of an A in the 2nd position of the PSSM
- If the 2nd letter of the first sequence is a C, multiply the joint probability by the probability of a C in the 2nd position of the PSSM
- If the 2nd letter of the first sequence is a G, multiply the joint probability by the probability of a G in the 2nd position of the PSSM
- If the 2nd letter of the first sequence is a T, multiply the joint probability by the probability of a T in the 2nd position of the PSSM

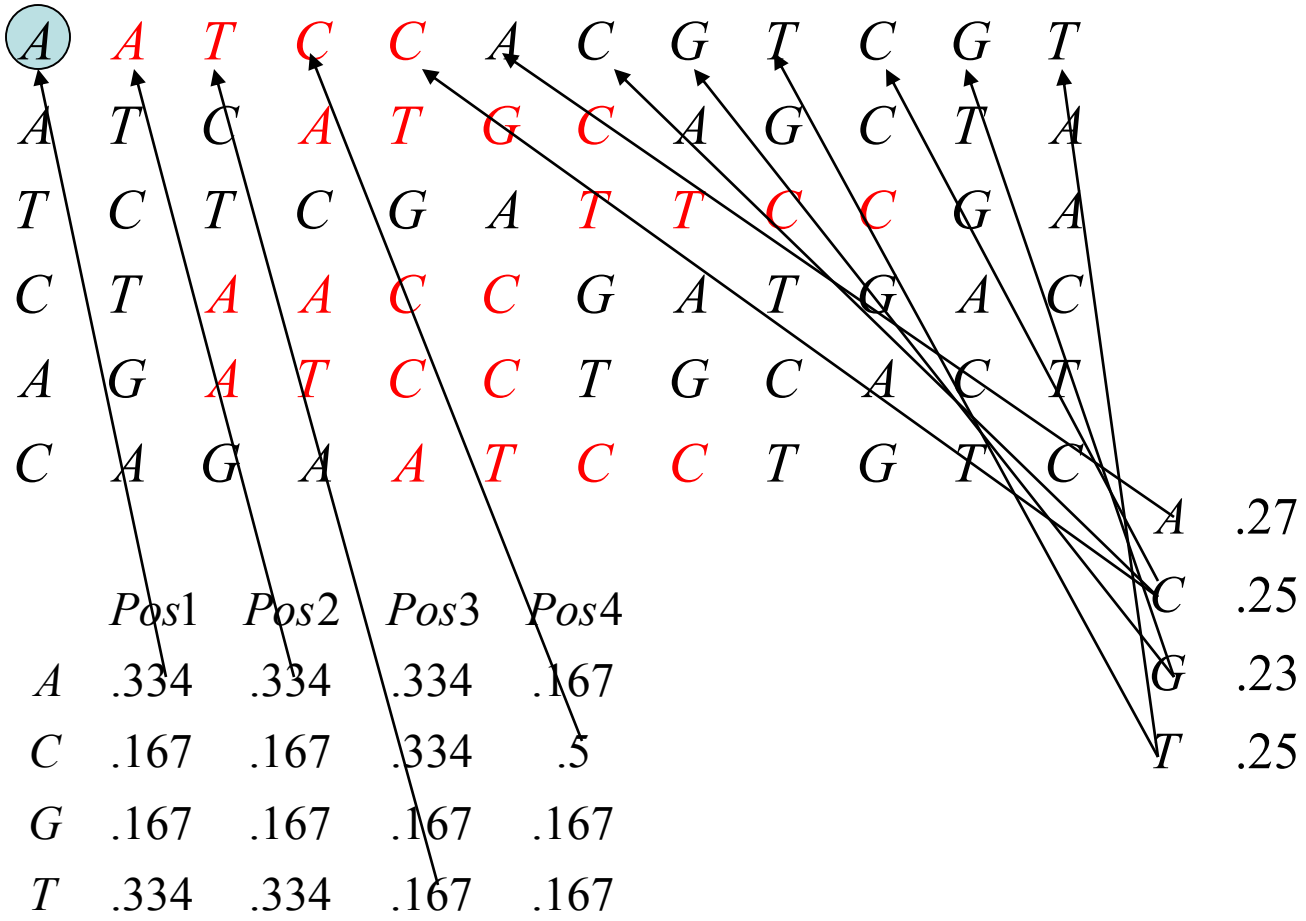
Expectation Phase Recipe-3

- Do this for all m letters of the motif
- For the remaining letters in the first sequence, by hypothesis, if they are not in the motif, they must be background, so continue to compute the joint probability using the frequencies of the background letters
- When all the letters of the first sequence are accounted for, there is an expression for the joint probability* of the first letter in the first sequence being the first letter of the motif

*This number promises to be very small; you may wish to use logs to prevent underflow

Example

Probability of an A in the first position of the first sequence



	Pos1	Pos2	Pos3	Pos4
A	.334	.334	.334	.167
C	.167	.167	.334	.5
G	.167	.167	.167	.167
T	.334	.334	.167	.167

$P(A_{1)}$=.334 x .334 x .167 x .5 x .25 x .27 x .25 x .23 x .25 x .25 x .23 x .25

Expectation Phase Recipe-4

- ‘Slide’ the PSSM one letter to the right in the sequence data, to position 2 of the sequence data and collect a new joint probability as before

- Do this for all n positions of the n-wide sequence data

- Do this for each sequence.

Expectation Phase Recipe-5

- When we are done, we have the joint probability of the motif if it were to start in each position of the data

- We want a relative probability that the motif starts in a position, so we need to normalize each of the joint probabilities, sequence by sequence, by dividing by the sum* of the joint probabilities. This gives us our expectation matrix

*Some variations use odds, rather than probability

Variation

Instead of the joint probability of the motif and background, we could ask “how ‘far’ is the motif pdf from the background pdf?”

The Kullback-Leibler distance measures how far pdfs are from each other

$$KL = \sum_{pos\ i} \left(\sum_{j \in \{A,C,G,T\}} p_{i,j} \log \frac{p_{i,j}}{q_j} \right)$$

where q is a letter in the background pdf

EXAMPLE WITH MOTIF LENGTH 4

Our motif data (red) are buried in sequences of random DNA background of length 80 (12 shown here)

0	1	2	3	4	5	6	7	8	9	10	11	12
A	A	**A**	**T**	**C**	**C**	T	C	A	T	A	C	C
A	C	C	T	**A**	**T**	**G**	**C**	C	G	A	C	T
A	T	C	G	G	G	G	**T**	**T**	**C**	**C**	T	C
C	T	G	**A**	**A**	**C**	**C**	C	G	A	A	C	T
G	G	C	**A**	**T**	**C**	**C**	T	G	A	T	T	G
T	C	G	C	T	**A**	**T**	**C**	**C**	A	C	T	C

Log p for our data in in the context of sequences of length 80

0	1	2	3	4	5	6	7	8	9	10	11	12	
-44.90	-44.38	-43.51	-44.26	-44.80	-45.89	-45.15	-45.60	-45.89	-45.89	-44.62	-45.46	-44.01	4
-44.53	-45.10	-44.82	-44.22	-45.05	-44.46	-46.31	-44.39	-45.27	-44.13	-43.68	-44.42	-44.75	4
-45.31	-45.56	-45.87	-43.90	-45.71	-45.26	-45.09	-44.52	-44.67	-45.21	-46.32	-44.54	-46.03	4
-45.99	-44.63	-45.47	-43.24	-43.72	-44.37	-44.60	-44.93	-44.69	-44.36	-43.84	-44.08	-44.32	4
-46.18	-46.18	-45.40	-43.46	-44.22	-44.79	-45.12	-46.20	-45.16	-45.40	-43.95	-45.36	-44.45	4
-43.89	-44.12	-44.34	-44.91	-44.46	-43.59	-44.34	-44.91	-44.46	-43.59	-44.36	-44.91	-46.30	4

Note

Normalized probability (in %) based on 80 letters per sequence

(12 shown here)

Normalized Probability Table

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	1
0.5	1.6	11.3	2.1	0.6	0.0	0.3	0.1	0.0	0.0	0.9	0.1	3.6	0.5	0.0	0.3	0.1	0.0	0.9	0
1.8	0.5	0.9	3.6	0.5	2.1	0.0	2.4	0.3	4.4	12.3	2.3	1.1	0.1	2.0	0.0	5.8	0.2	0.9	0
0.4	0.2	0.1	9.2	0.1	0.4	0.6	2.2	1.6	0.4	0.0	2.1	0.1	0.0	0.6	0.1	0.1	0.9	0.0	0
0.0	1.1	0.2	27.9	9.3	2.1	1.2	0.6	1.0	2.1	7.0	4.1	2.4	0.7	0.3	0.5	2.2	0.2	3.5	0
0.0	0.0	0.1	13.0	2.2	0.6	0.3	0.0	0.3	0.1	4.2	0.2	1.3	0.1	0.6	6.4	7.9	2.1	0.9	2
6.4	3.7	2.3	0.6	1.7	12.5	2.3	0.6	1.7	12.5	2.1	0.6	0.0	0.2	0.0	1.0	0.2	1.2	0.1	0

Maximization Phase

This completes the Expectation step.

For the Maximization step:

Now use the expectation matrix to generate a new PSSM by adding weights

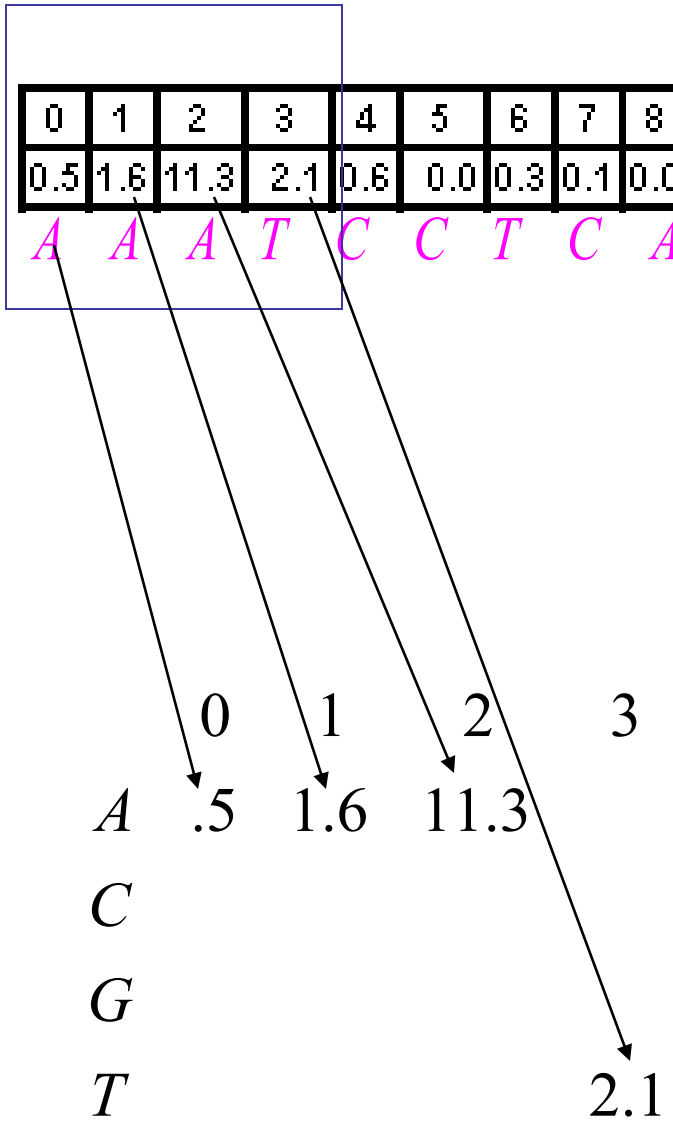
Maximization Phase Recipe-1

- Create an empty new PSSM
- Augment each base count in each position of the PSSM by a weight:
  - For each sequence in the data
    - For each cell ***i*** + ***j*** in the Expectation Matrix
    - For each column ***j*** in the new PSSM
      - Add the ***i*** + ***j*** th probability to the bin in the ***j*** th column of the PSSM appropriate to the base in the ***i*** + ***j*** th position of the original data

The above assumes zero-based indexing

Normalized Probability Table

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	1
0.5	1.6	11.3	2.1	0.6	0.0	0.3	0.1	0.0	0.0	0.9	0.1	3.6	0.5	0.0	0.3	0.1	0.0	0.9	0
A	*A*	*A*	*T*	*C*	*C*	*T*	*C*	*A*	*T*	*A*	*C*	*C*							

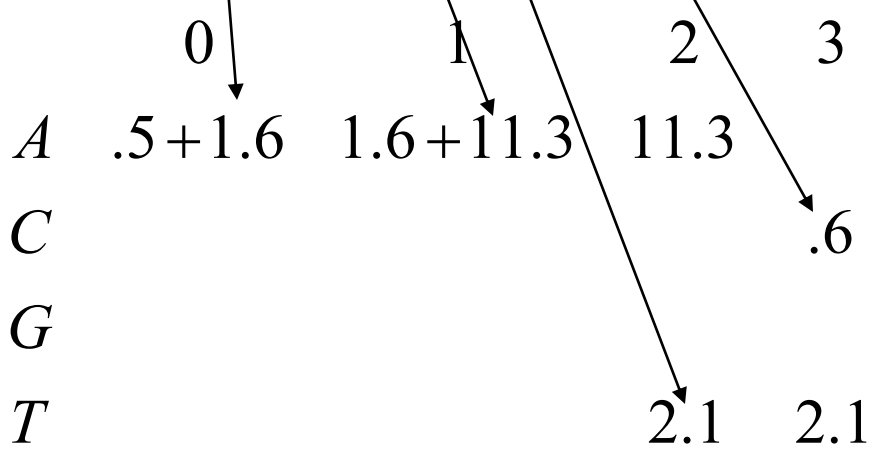


EXAMPLE

1st sequence, 0th position

Normalized Probability Table

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	1
0.5	1.6	11.3	2.1	0.6	0.0	0.3	0.1	0.0	0.0	0.9	0.1	3.6	0.5	0.0	0.3	0.1	0.0	0.9	0
A	*A*	*A*	*T*	*C*	*C*	*T*	*C*	*A*	*T*	*A*	*C*	*C*							



EXAMPLE

1st sequence, 1st position

Maximization Phase Recipe-2

- Exhaust all possible starting positions in the expectation matrix
- Do this for each sequence in the data

Maximization Phase
Recipe-3

- When all the weights are accumulated in the new PSSM, normalize them!
- The product is a refined PSSM

Motif Discovery by EM

Repeat

- Create a new Expectation matrix using the new PSSM
- Maximize PSSM by using the Expectation Matrix for weights

Stop: when things don't change.

Site Counts before this maximization step

	Col 0	Col 1	Col 2	Col 3
A	4.100	2.100	1.100	1.100
C	1.100	2.100	4.100	3.100
G	0.100	1.100	0.100	1.100
T	1.100	1.100	1.100	1.100

Augmented Site Counts after this maximization step

	Col 0	Col 1	Col 2	Col 3
A	3.850	1.750	0.973	0.996
C	1.098	1.939	3.729	3.376
G	0.184	1.098	0.199	0.943
T	1.269	1.613	1.500	1.085

UPDATED Site Frequencies after Maximization				
	Col 0	Col 1	Col 2	Col 3
A	0.602	0.273	0.152	0.156
C	0.171	0.303	0.583	0.527
G	0.029	0.172	0.031	0.147
T	0.198	0.252	0.234	0.170

Original (Hidden) Motif Site Frequencies				
	Col 0	Col 1	Col 2	Col 3
A	0.833	0.167	0.000	0.000
C	0.000	0.000	0.833	1.000
G	0.000	0.000	0.167	0.000
T	0.167	0.833	0.000	0.000

The Final Product

The PSSM will fully describe the elicited motifs in the data

Multiple Em Motif Elicitation

- The EM algorithm we have described discovers one motif in each sequence.
- There are sophisticated algorithms using the EM principle that can handle gaps and can find no, one, or many motifs in each sequence MEME is such an application available on the web from the San Diego supercomputing center
  http://meme.sdsc.edu/meme/intro.html

Other approaches to Motif Discovery

- Gibbs sampler
- HMM