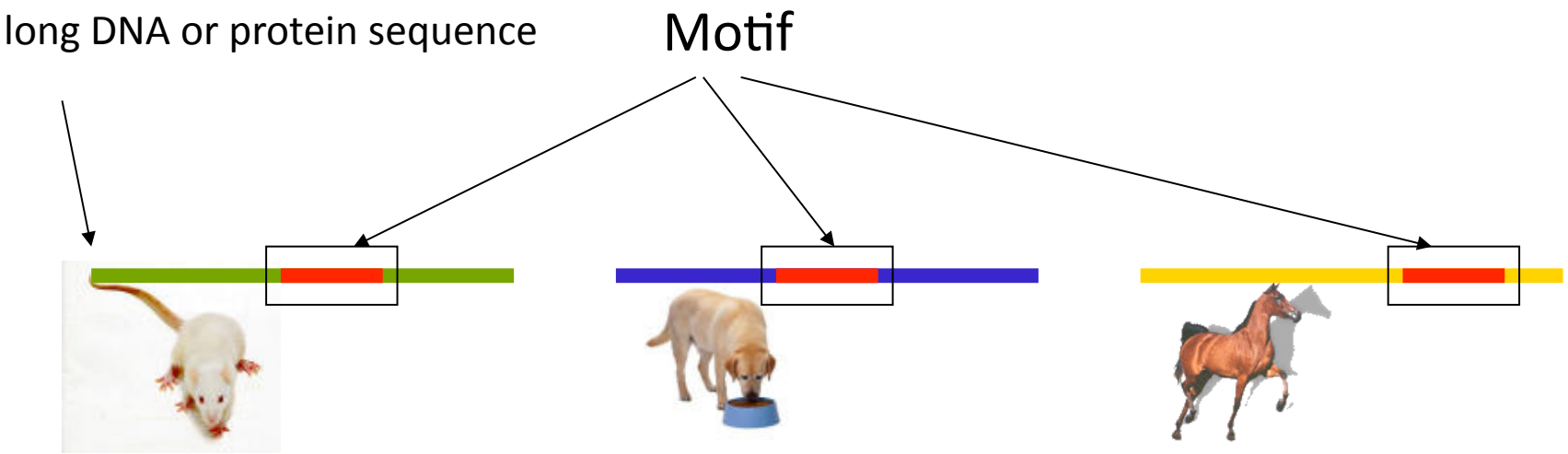
Motif Discovery

02-710 Computational Genomics

Seyoung Kim

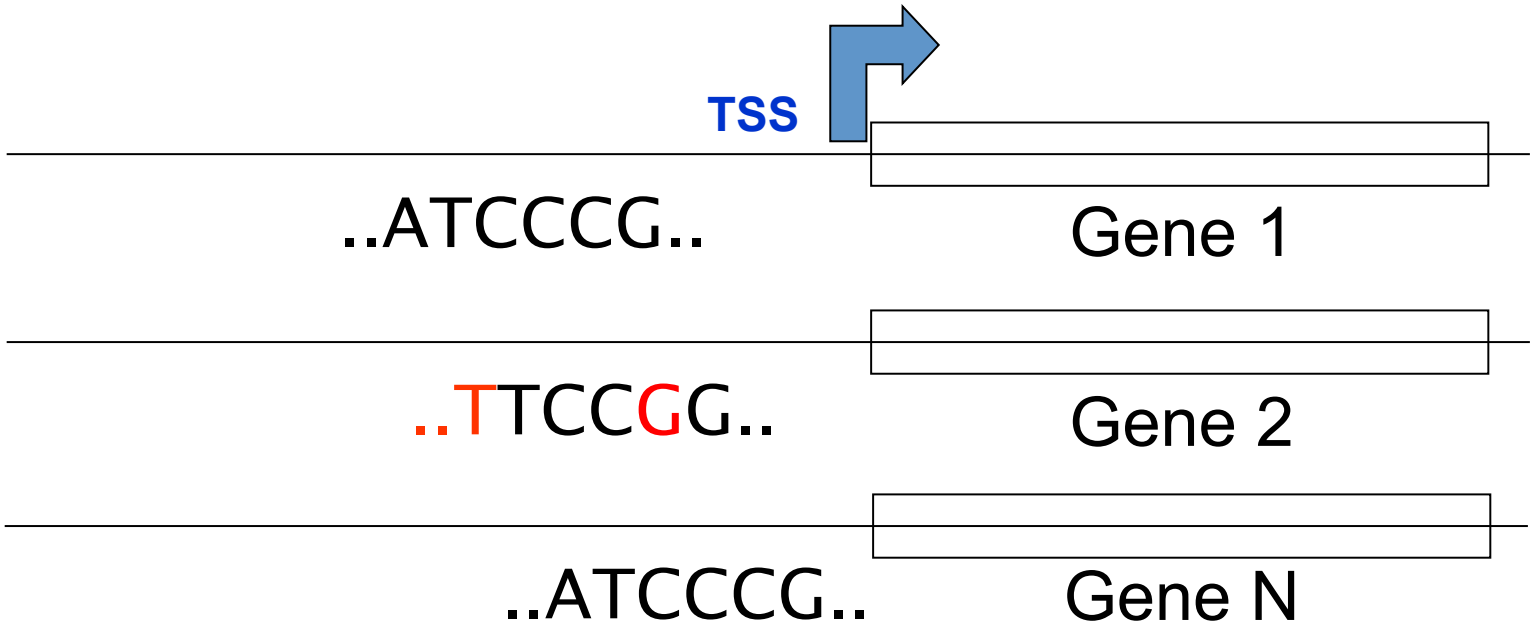
Motif

- Set of similar substrings, within a family of diverged DNA or protein sequences, which likely has a function



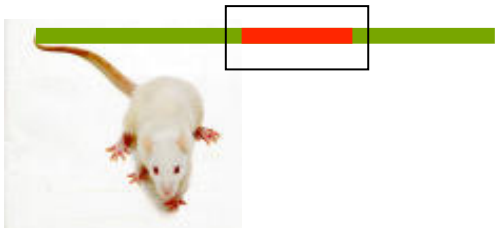
Transcription Factor Binding

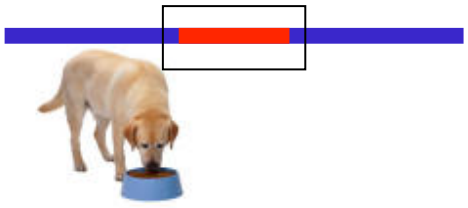
Transcription Factor Binding



Why identify motifs?

- In DNA
  - Discover how genes are regulated
- In proteins
  - Identify functionally important regions of a protein family
  - Find similarities to known proteins





Outline

- Motif models and position specific scoring matrix (PSSM)
- Assume PSSM is given and detect motif matches in a long sequence
- Assume PSSM is not given and learning motif models from a set of long sequences harboring motifs

Position Specific Scoring Matrix (PSSM)

```
AAGTGT
TAATGT
AATTGT
AATTGA
ATCTGT
AATTGT
TGTTGT
AAATGA
TTTTGT
```

```
A  1.32  1.32 -0.15 -3.32 -3.32 -0.15
C -3.32 -3.32 -1.00 -3.32 -3.32 -3.32
G -3.32 -1.00 -1.00 -3.32  1.89 -3.32
T  0.38 -0.15  1.07  1.89 -3.32  1.54
```

A set of related DNA sequences, each with 6 nucleotides

PSSM derived from the sequences, each entry for log-odds score

Log-odds Scores in PSSSM

1. Estimate the probability of observing each nucleotide.
2. Divide by the background probability of observing the same nucleotide.
3. Take the log so that the scores are additive.

Nucleotide "A" is observed.

Nucleotide was generated by the foreground model (i.e., the PSSM).

$$\log_2\left(\frac{\Pr(A|M)}{\Pr(A|B)}\right)$$

The nucleotide was generated by the background model (i.e., randomly selected nucleotide).

How to specify the foreground motif model M

How to specify the background model B

Learning the Foreground Motif Model Pr(A|M)'s

```
AAGTGT
TAATGT
AATTGT
AATTGA
ATCTGT
AATTGT
TGTTGT
AAATGA
TTTTGT
```

A set of related DNA sequences, each with 6 nucleotides

```
A 6 6 2 0 0 2
C 0 0 1 0 0 0
G 0 1 1 0 9 0
T 3 2 5 9 0 7
```

Create a matrix of counts for observing each nucleotide at each position

```
A 6.25 6.25 2.25 0.25 0.25 2.25
C 0.25 0.25 1.25 0.25 0.25 0.25
G 0.25 1.25 1.25 0.25 9.25 0.25
T 3.25 2.25 5.25 9.25 0.25 7.25
```

Add a pseudocount 0.25

```
A 0.625 0.625 0.225 0.025 0.025 0.225
C 0.025 0.025 0.125 0.025 0.025 0.025
G 0.025 0.125 0.125 0.025 0.925 0.025
T 0.325 0.225 0.525 0.925 0.025 0.725
```

Normalize each column to make it probabilities summing to 1 (Position Weight Matrix (PWM))

Learning the Background Model Pr(A|B)'s

```
A 0.25
C 0.25
G 0.25
T 0.25
```

Assume equal probabilities for all nucleotides

Even better, calculate from the frequency in sequences but discard positions

Putting Together PSSM

```
A 0.625 0.625 0.225 0.025 0.025 0.225
C 0.025 0.025 0.125 0.025 0.025 0.025
G 0.025 0.125 0.125 0.025 0.925 0.025
T 0.325 0.225 0.525 0.925 0.025 0.725
```

$Pr(A|M)$

```
A 0.25
C 0.25
G 0.25
T 0.25
```

$Pr(A|B)$

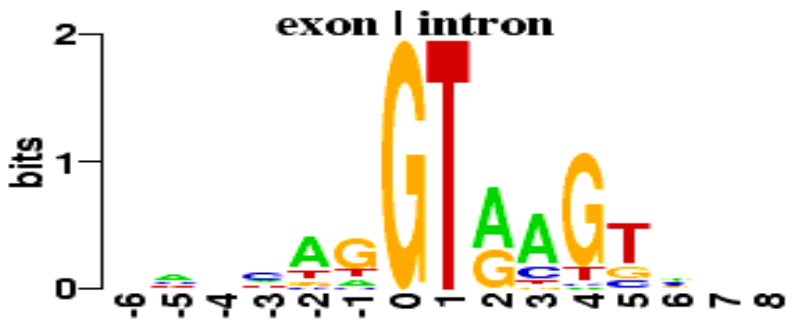
```
A 2.5 2.5 0.9 0.1 0.1 0.9
C 0.1 0.1 0.5 0.1 0.1 0.1
G 0.1 0.5 0.5 0.1 3.7 0.1
T 1.3 0.9 2.1 3.7 0.1 2.9
```

$\frac{Pr(A|M)}{Pr(A|B)}$

```
A  1.32  1.32 -0.15 -3.32 -3.32 -0.15
C -3.32 -3.32 -1.00 -3.32 -3.32 -3.32
G -3.32 -1.00 -1.00 -3.32  1.89 -3.32
T  0.38 -0.15  1.07  1.89 -3.32  1.54
```

$\log_2\left(\frac{Pr(A|M)}{Pr(A|B)}\right)$

Motif in Logo Format



Splice site motif

CTCF binding motif

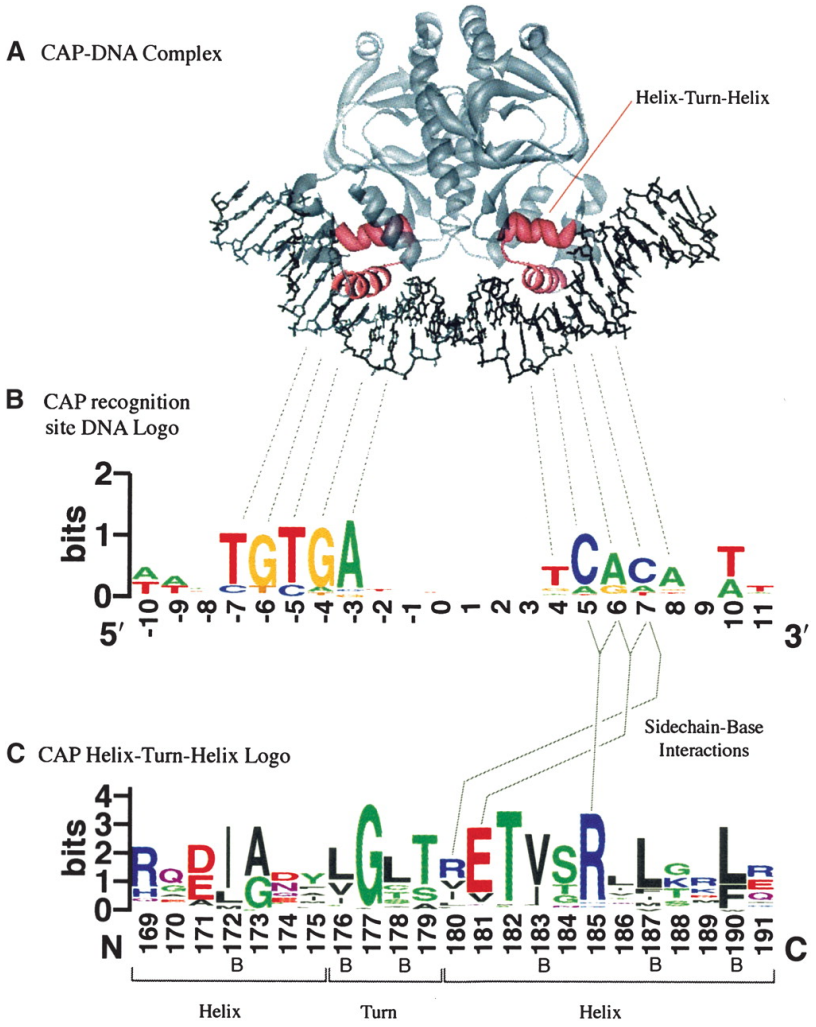
- Total height at each position: information content (log base 2)

$$\sum_{\alpha} p_{i,\alpha} \log \frac{p_{i,\alpha}}{b_{\alpha}} = \log 4 - (-\sum_{\alpha} p_{i,\alpha} \log p_{i,\alpha})$$

  - $p_{i,\alpha}$ = probability of α in matrix position *i*
  - b_{α} = background frequency of α

- Some positions have no information (random distribution)

CAP (Catabolite Activator Protein, also known as CRP) acts as a transcription activator by binding at more than 100 sites within the Escherichia coli genome



A palindromic sequence (w.r.t. both strands)

Motif Reveals the most important sequence-specific binding/interaction sites

Scanning for Motif Occurrences

- Given:
  - a long DNA sequence

    TAATGTTTGTGCTGGTTTTTGTGGCATCGGGCGAGAATAGCGCGTGGT
  - a DNA motif represented as a PSSM

```
A  1.32  1.32 -0.15 -3.32 -3.32 -0.15
C -3.32 -3.32 -1.00 -3.32 -3.32 -3.32
G -3.32 -1.00 -1.00 -3.32  1.89 -3.32
T  0.38 -0.15  1.07  1.89 -3.32  1.54
```

- Find:
  - occurrences of the motif in the sequence

Scanning for Motif Occurrences

A	1.32	1.32	-0.15	-3.32	-3.32	-0.15
C	-3.32	-3.32	-1.00	-3.32	-3.32	-3.32
G	-3.32	-1.00	-1.00	-3.32	1.89	-3.32
T	0.38	-0.15	1.07	1.89	-3.32	1.54

0.38 + 1.32 – 0.15 + 1.89 + 1.89 + 1.54 = 6.87

TAATGTTTGTGCTGGTTTTTGTGGCATCGGGCGAGAATAGCGCGTGGTGTGAAAG

Scanning for Motif Occurrences

```
A  1.32  1.32 -0.15 -3.32 -3.32 -0.15
C -3.32 -3.32 -1.00 -3.32 -3.32 -3.32
G -3.32 -1.00 -1.00 -3.32  1.89 -3.32
T  0.38 -0.15  1.07  1.89 -3.32  1.54
```

1.32 + 1.32 + 1.07 – 3.32 – 3.32 + 1.54 = -1.39

TAATGTTTGTGCTGGTTTTTGTGGCATCGGGCGAGAATAGCGCGTGGTGTGAAAG

Scanning for motif occurrences

```
A  1.32  1.32 -0.15 -3.32 -3.32 -0.15
C -3.32 -3.32 -1.00 -3.32 -3.32 -3.32
G -3.32 -1.00 -1.00 -3.32  1.89 -3.32
T  0.38 -0.15  1.07  1.89 -3.32  1.54
```

1.32 + 1.32 + 1.07 – 3.32 – 3.32 + 1.54 = -1.39

TAATGTTTGTGCTGGTTTTTGTGGCATCGGC

Essentially we are computing the log likelihood ratio:

$$\log\left(\frac{\prod_{i=1}^{6}\Pr(A_i|\mathrm{M})}{\prod_{i=1}^{6}\Pr(A_i|\mathrm{B})}\right)$$

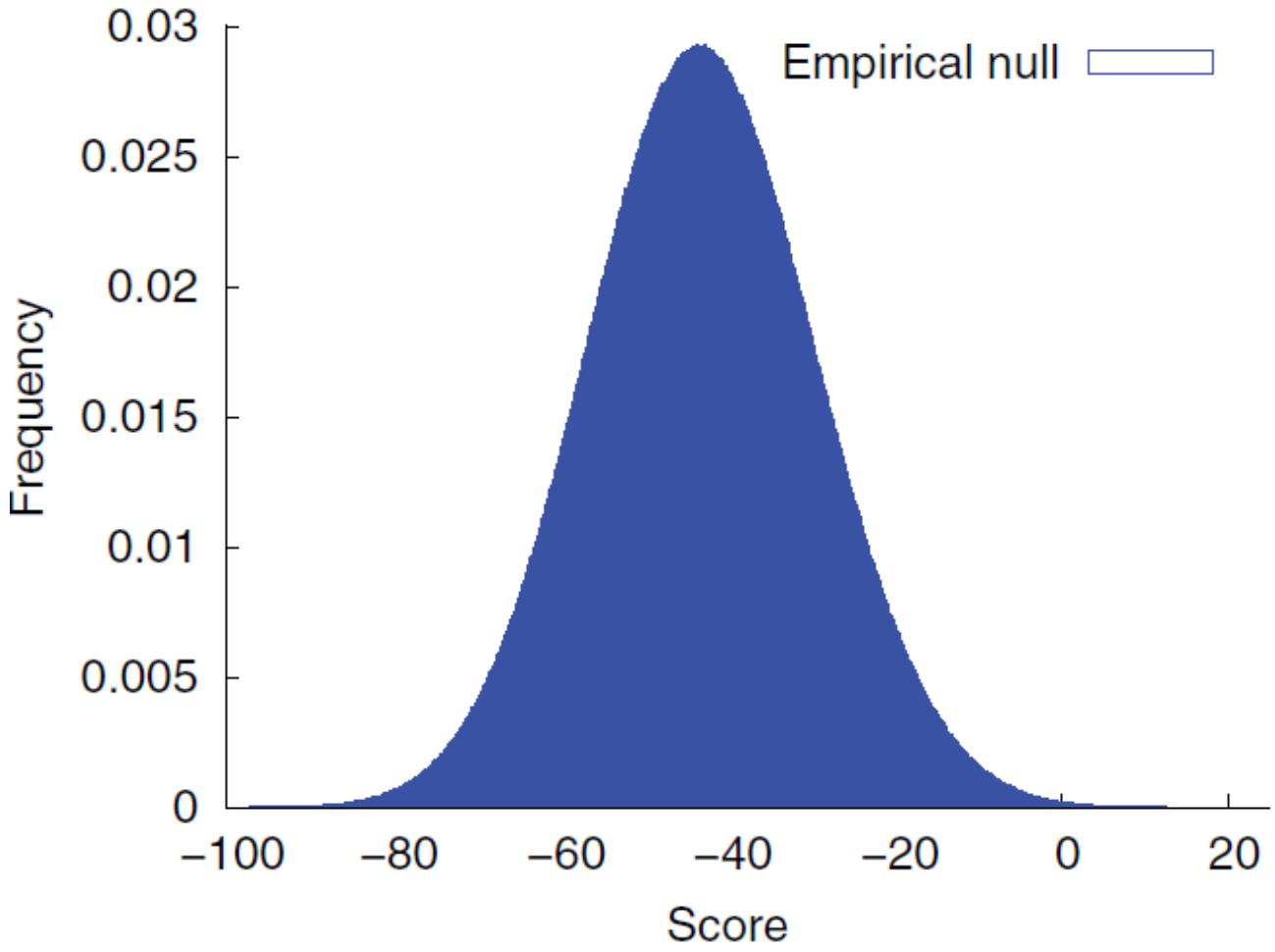
High score = motif occurrence

How to assess the significance of the high scores?

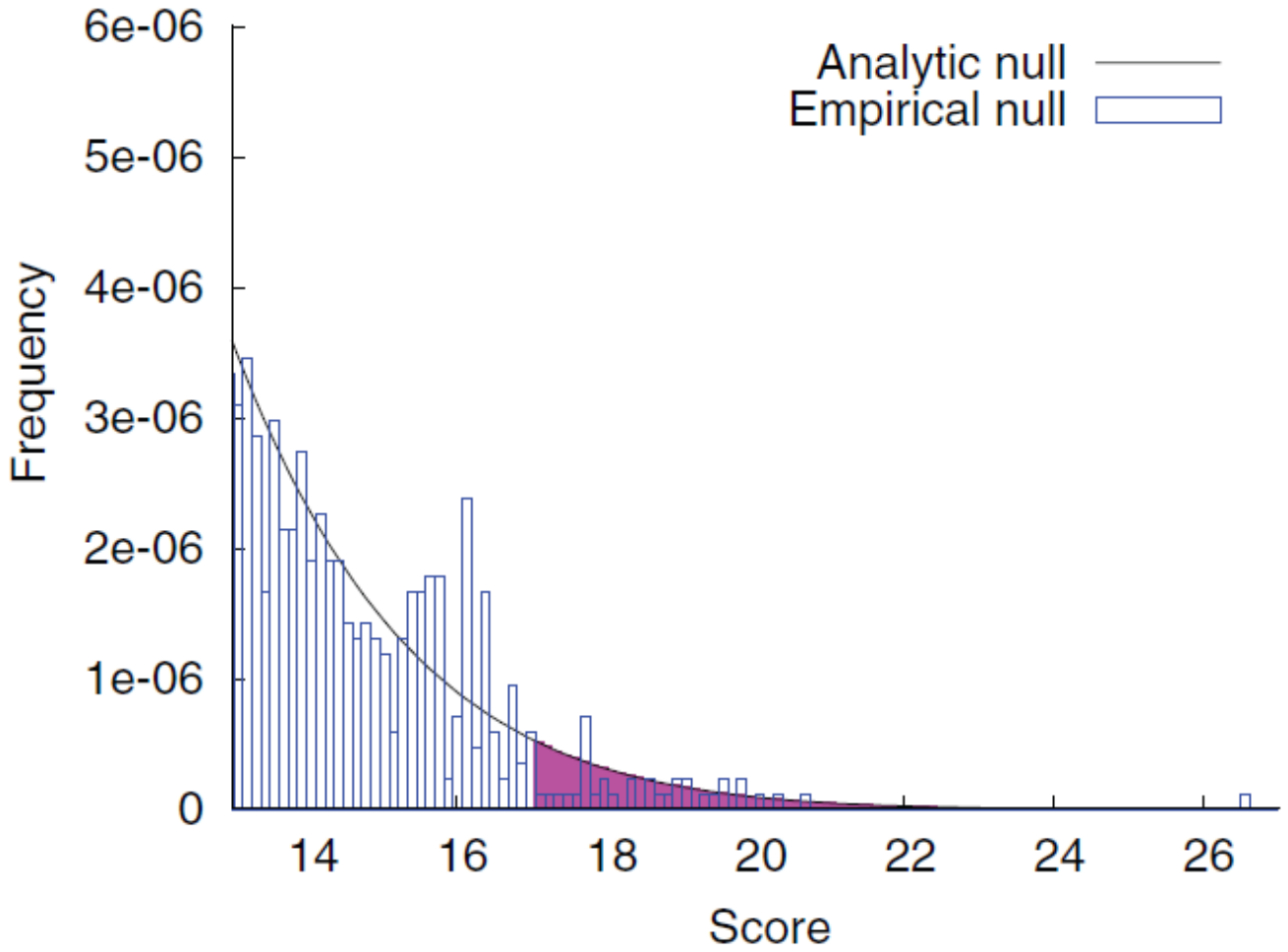
Two Ways to Assess Significance

1. Empirical
   - Randomly generate data according to the null hypothesis.
   - Use the resulting score distribution to estimate p-values.
2. Exact
   - Mathematically calculate all possible scores
   - Use the resulting score distribution to estimate p-values.

CTCF Empirical Null Distribution



Poor Precision in the Tail



Two Ways to Assess Significance

1. Empirical
   - Randomly generate data according to the null hypothesis.
   - Use the resulting score distribution to estimate p-values.
2. Exact
   - Mathematically calculate all possible scores
   - Use the resulting score distribution to estimate p-values.

Converting Scores to p-values

A	-2.3	1.7	1.1	0.1
C	1.2	-0.3	0.4	-1.0
G	-3.0	2.0	0.5	0.8
T	4.0	0.0	-2.1	1.5

A	10	67	59	44
C	60	39	49	29
G	0	71	50	54
T	100	43	13	64

- Linearly rescale the matrix values to the range [0,100] and integerize.

Converting Scores to p-values

A	-2.3	1.7	1.1	0.1
C	1.2	-0.3	0.4	-1.0
G	-3.0	2.0	0.5	0.8
T	4.0	0.0	-2.1	1.5

A	0.7	4.7	4.1	3.1
C	4.2	2.7	3.4	2.0
G	0.0	5.0	3.5	3.8
T	7.0	3.0	0.9	4.5

- Find the smallest value.
- Subtract that value from every entry in the matrix.
- All entries are now non-negative.

Converting Scores to p-values

A	0.7	4.7	4.1	3.1
C	4.2	2.7	3.4	2.0
G	0.0	5.0	3.5	3.8
T	7.0	3.0	0.9	4.5

A	10.00	67.14	58.57	44.29
C	60.00	38.57	48.57	28.57
G	0.00	71.43	50.00	54.29
T	100.00	42.86	12.85	64.29

100 / 7 = 14.2857

- Find the largest value.
- Divide 100 by that value.
- Multiply through by the result.
- All entries are now between 0 and 100.

Converting Scores to p-values

A	10.00	67.14	58.57	44.29
C	60.00	38.57	48.57	28.57
G	0.00	71.43	50.00	54.29
T	100.00	42.86	12.85	64.29

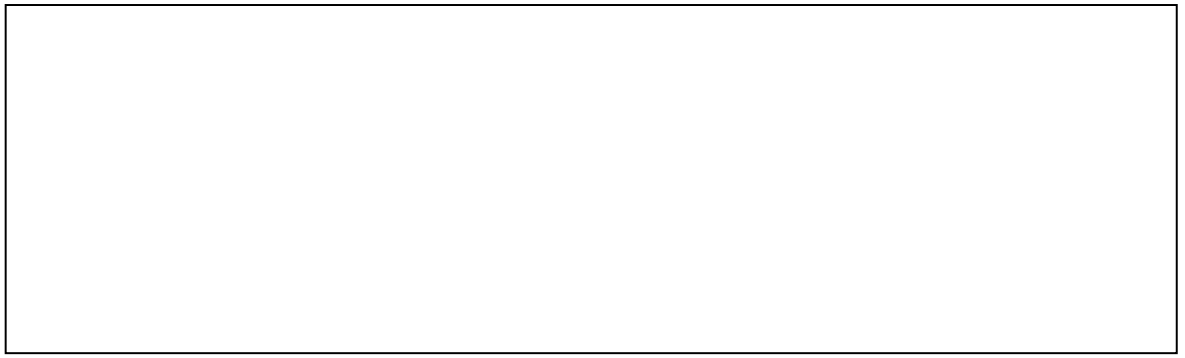
A	10	67	59	44
C	60	39	49	29
G	0	71	50	54
T	100	43	13	64

- Round to the nearest integer.

Now we compute the exact null distribution

Converting Scores to p-values

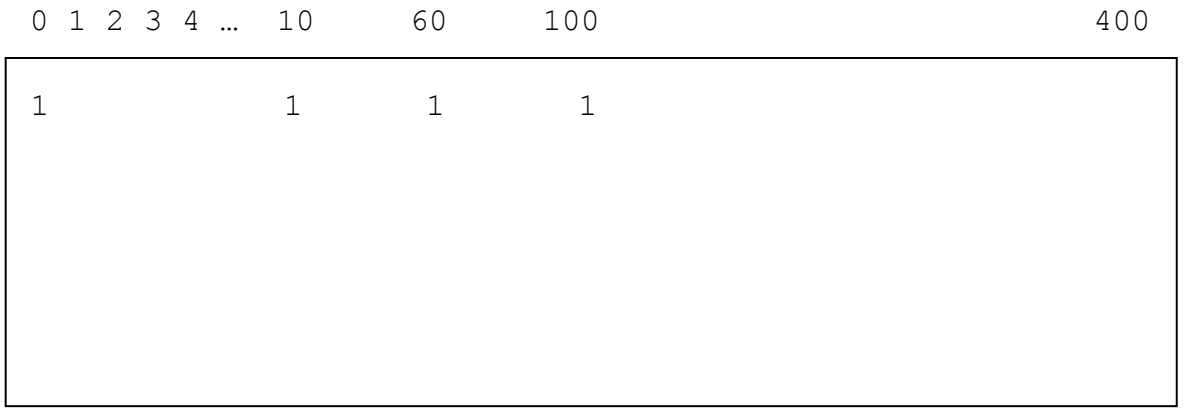
	0	1	2	3	4	…	400
A	10	67	59	44			
C	60	39	49	29			
G	0	71	50	54			
T	100	43	13	64			



- Say that your motif has N columns. Create a matrix that has N rows and 100N columns.
- The entry in row i, column j is the number of different sequences of length i that can have a score of j.

Converting Scores to p-values

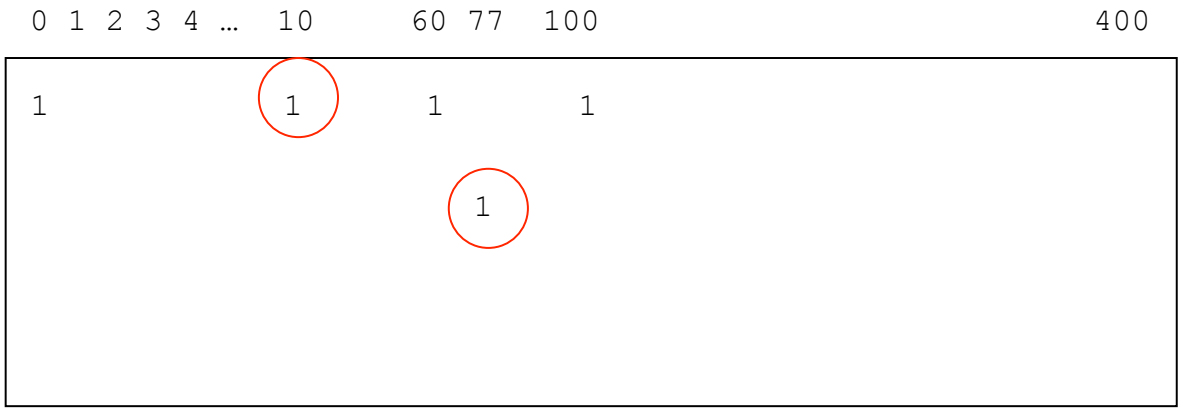
```
A  10 67 59 44
C  60 39 49 29
G   0 71 50 54
T 100 43 13 64
```



- For each value in the first column of your motif, put a 1 in the corresponding entry in the first row of the matrix.
- There are only 4 possible sequences of length 1.

Converting Scores to p-values

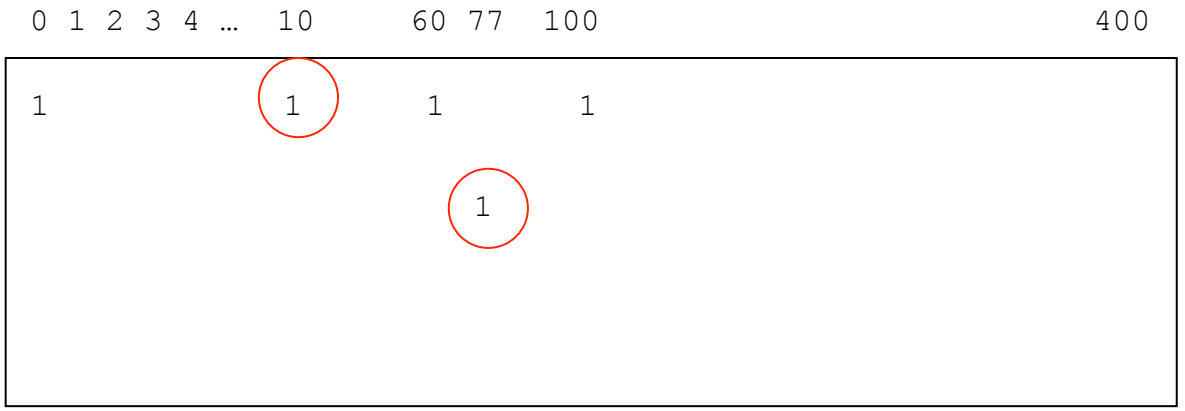
A	10	67	59	44
C	60	39	49	29
G	0	71	50	54
T	100	43	13	64



- For each value x in the second column of your motif, consider each value y in the zth column of the first row of the matrix.
- Add y to the x+zth column of the matrix.

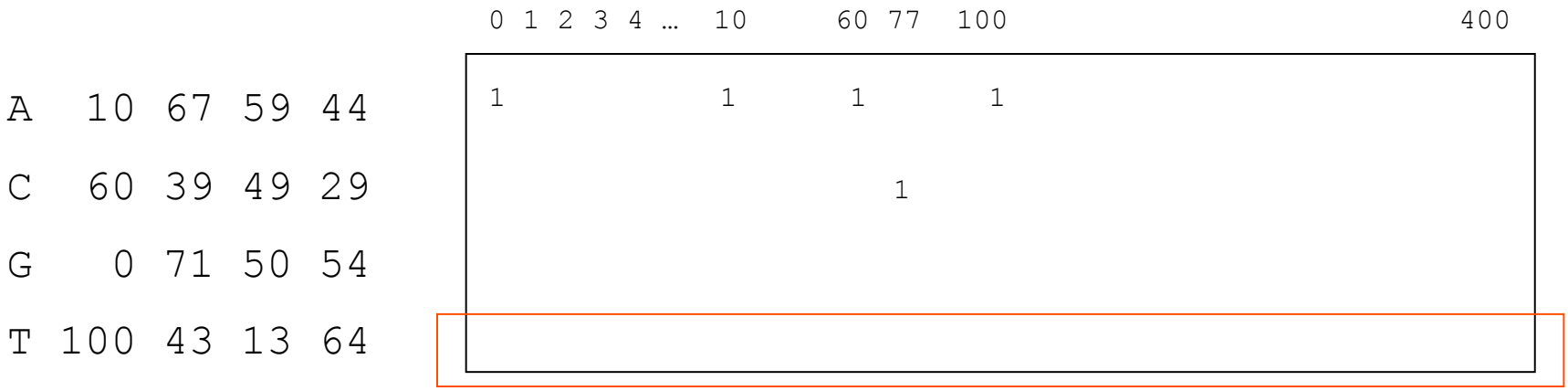
Converting Scores to p-values

A	10	67	59	44
C	60	39	49	29
G	0	71	50	54
T	100	43	13	64



- For each value x in the second column of your motif, consider each value y in the zth column of the first row of the matrix.
- Add y to the x+zth column of the matrix.
- What values will go in row 2?
  - 10+67, 10+39, 10+71, 10+43, 60+67, …, 100+43
- These 16 values correspond to all 16 strings of length 2.

Converting Scores to p-values



- In the end, the bottom row contains the scores for all possible sequences of length N.
- Use these scores to compute a p-value.

Multiple Testing Problems

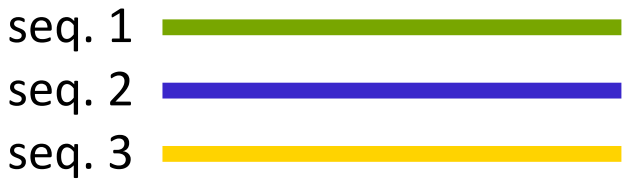
- When scanning a long sequence for motif match, we face multiple testing problem

- How to correct for this problem
  - Bonferroni correction
  - False discovery rate

So Far

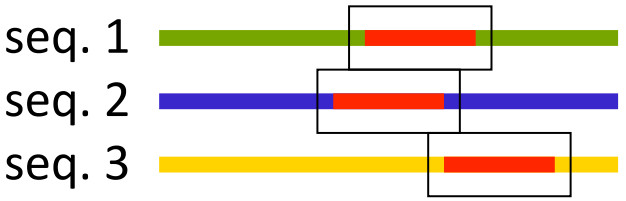
- We assumed the alignment of short sequences was given and derived PSSM

- What if neigher the alignment of short sequences nor motif is given
  - Given sequences



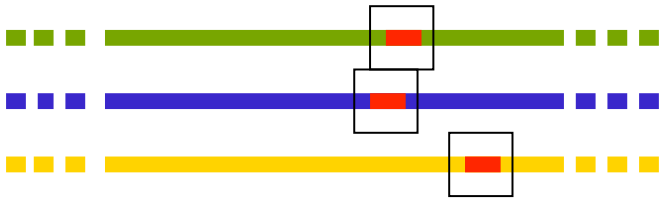
  - Find motif

IGRGGFGEVY at position 515
LGEGCFGQVV at position 430
VGSGGFGQVY at position 682

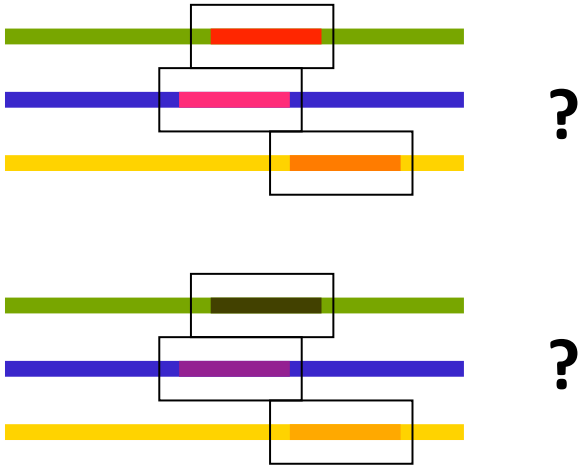


Why is This Hard?

- Input sequences are long (thousands or millions of residues).



- Motif may be *subtle*
  - Instances are short.
  - Instances are only slightly similar.



A Concrete Example: Transcription Factor Binding Sites

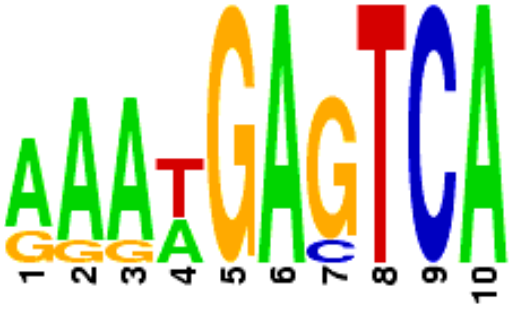
We are given a set of promoters from co-regulated genes. The sequence motif discovery problem is to discover the sites (or the motif) given just the sequences.

A Concrete Example: Transcription Factor Binding Sites

An unknown transcription factor binds to positions unknown to us, on either DNA strand.

A Concrete Example: Transcription Factor Binding Sites

The DNA binding motif of the transcription factor can be described by a position-specific scoring matrix (PSSM).




```
5′- TCTCTCTCCACGGCTAATTAGGTGATCATGAAAAAATGAAAAATTCATGAGAAAAGAGTCAGACATCGAAACATACAT  ...HIS7
5′- ATGGCAGAATCACTTTAAAACGTGGCCCCACCCGCTGCACCCTGTGCATTTTGTACGTTACTGCGAAATGACTCAACG  ...ARO4
5′- CACATCCAACGAATCACCTCACCGTTATCGTGACTCACTTTCTTTCGCATCGCCGAAGTGCCATAAAAAATATTTTTT  ...ILV6
5′- TGCGAACAAAAGAGTCATTACAACGAGGAAATAGAAGAAAATGAAAAATTTTCGACAAAATGTATAGTCATTTCTATC  ...THR4
5′- ACAAAGGTACCTTCCTGGCCAATCTCACAGATTTAATATAGTAAATTGTCATGCATATGACTCATCCCGAACATGAAA  ...ARO1
5′- ATTGATTGACTCATTTTCCTCTGACTACTACCAGTTCAAAATGTTAGAGAAAAATAGAAAAGCAGAAAAAATAAATAA  ...HOM2
5′- GGCGCCACAGTCCGCGTTTGGTTATCCGGCTGACTCATTCTGACTCTTTTTTGGAAAGTGTGGCATGTGCTTCACACA  ...PRO3
```

The MEME Algorithm

MEME uses expectation maximization (EM) to discover sequence motifs.

Slides courtesy of Tim Bailey

The MEME Algorithm

The positions (and strands) of the motif sites are the missing information variables, $\boldsymbol{Z} = \{Z_{i,j}\}$ for *i*th short sequence (at position *i*).

Motif indicator Z_{i1} = { 1, if motif; 0, otherwise }

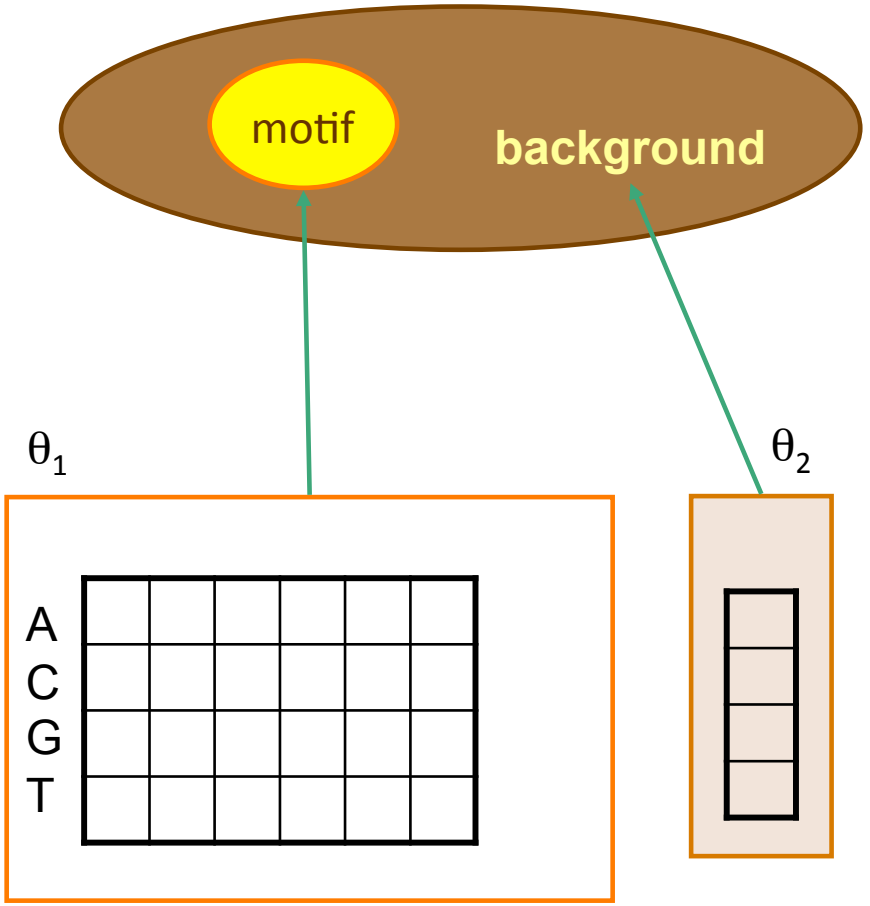
Background indicator Z_{i2} = { 1, if background; 0, otherwise }

The MEME Algorithm

M-step: If we knew the $Z = \{Z_{i,j}\}$, we could estimate the motif PSSM (both motif and background probability models) using maximum likelihood.

The MEME Algorithm: Parameters

- Motif/background model $\theta = \{\theta_1, \theta_2\}$
  - Define motif model θ_1:
    P_{ij} = Pr[letter i occurs in motif position j]
  - Define background model θ_2 :
    P_j = Pr[letter j in background sequence]
- Mixture proportion parameters $\lambda=\{\lambda_1, \lambda_2\}$
  λ_1: $Pr(Z_{i1}=1)$
  $\lambda_2 = (1-\lambda_1)$: $Pr(Z_{i2}=1)$



The MEME Algorithm

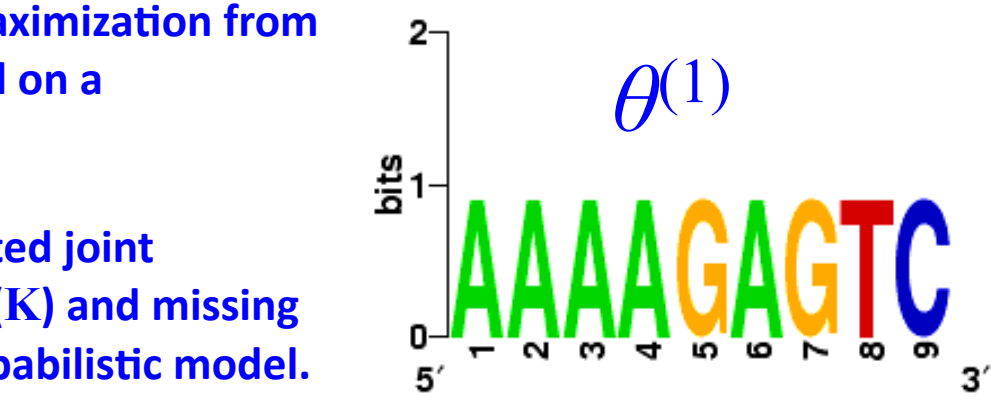
E-step: If we knew the motif PSSM, we could estimate $Z = \{Z_{i,j}\}$ by scoring the long sequence with the PSSM.

The MEME Algorithm

MEME starts Expectation Maximization from an initial estimate of θ based on a subsequence in the data

MEME maximizes the expected joint likelihood of the sequences (K) and missing information (Z) under a probabilistic model.

.




```
5′- TCTCTCTCCACGGCTAATTAGGTGATCATGAAAAAATGAAAAATTCATGAGAAAAGAGTCAGACATCGAAACATACAT ...HIS7
5′- ATGGCAGAATCACTTTAAAACGTGGCCCCACCCGCTGCACCCTGTGCATTTTGTACGTTACTGCGAAATGACTCAACG ...ARO4
5′- CACATCCAACGAATCACCTCACCGTTATCGTGACTCACTTTCTTTCGCATCGCCGAAGTGCCATAAAAAATATTTTTT ...ILV6
5′- TGCGAACAAAAGAGTCATTACAACGAGGAAATAGAAGAAAATGAAAAATTTTCGACAAAATGTATAGTCATTTCTATC ...THR4
5′- ACAAAGGTACCTTCCTGGCCAATCTCACAGATTTAATATAGTAAATTGTCATGCATATGACTCATCCCGAACATGAAA ...ARO1
5′- ATTGATTGACTCATTTTCCTCTGACTACTACCAGTTCAAAATGTTAGAGAAAAATAGAAAAGCAGAAAAAATAAATAA ...HOM2
5′- GGCGCCACAGTCCGCGTTTGGTTATCCGGCTGACTCATTCTGACTCTTTTTTGGAAAGTGTGGCATGTGCTTCACACA ...PRO3
```

MEME: EM Algorithm

- Given a set of *n* words (short sequences) $K_i = k_1 \ldots k_K$, where i=1, …, *n*, EM maximizes the expected complete data log likelihood in each iteration:

$$\log P(K_1 \ldots K_n, Z \mid \theta, \lambda) = \sum_{i=1}^{n} \sum_{j=1}^{2} Z_{ij} \log(\lambda_j P(K_i \mid \theta_j))$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{2} Z_{ij} \log P(K_i \mid \theta_j) + \sum_{i=1}^{n} \sum_{j=1}^{2} Z_{ij} \log \lambda_j$$

Define:

Mixture component parameters θ

θ_1: Motif probabilities;

θ_2: Background probablities

λ_1, λ_2: Mixture proportions

Expectation:

Find expected complete data log likelihood (impute missing Z):

$$E[\log P(K_1 \ldots K_n, Z \mid \theta, \lambda)]$$

Maximization:

Maximize expected the expected complete data log likelihood above over parameters θ, λ

MEME: E-step

Expectation:

Find expected complete data log likelihood (impute missing Z):

$$E[\log P(K_1 \ldots K_n, Z \mid \theta, \lambda)] =$$

$$\sum_{i=1}^{n} \sum_{j=1}^{2} E[Z_{ij}] \log P(K_i \mid \theta_j) + \sum_{i=1}^{n} \sum_{j=1}^{2} E[Z_{ij}] \log \lambda_j$$

where expected values of Z can be computed as follows:

$$E[Z_{ij}] = \frac{\lambda_j P(K_i \mid \theta_j)}{\lambda P(K_i \mid \theta_1) + (1-\lambda) P(K_i \mid \theta_2)} = Z^*_{ij}$$

Impute the missing Z!

MEME: M-step

Maximization:

Maximize expected value over θ and λ given the imputed Z^*_{ij}

For λ :

$$\lambda^{NEW} = \arg\max_{\lambda} \sum_{i=1}^{n} (Z^*_{i1} \log\lambda + Z^*_{i2} \log(1-\lambda)) \Rightarrow$$

$$\frac{\delta}{\delta\lambda} \sum_{i=1}^{n} (Z^*_{i1} \log\lambda + Z^*_{i2} \log(1-\lambda)) = \sum_{i=1}^{n} \frac{Z^*_{i1}}{\lambda} - \frac{Z^*_{i2}}{(1-\lambda)} = 0$$

$$\Rightarrow \lambda = \sum_{i=1}^{n} \frac{Z^*_{i1}}{n} \text{ since } Z^*_{i1} + Z^*_{i2} = 1$$

The maximum likelihood estimate of the parameters are relative frequencies in the full set of data

MEME: M-step

Maximization:

Maximize expected value over θ and λ given the imputed Z^*_{ij}

For θ for motif and background models:

c_{jk} = E[# times letter k appears in motif position j]

c_{0k} = E[# times letter k appears in background]

c_{ij} values are calculated easily from Z* values

$$P_{jk}^{NEW} = \frac{c_{jk}}{\sum_{k=1}^{4} c_{jk}} \qquad P_{0k}^{NEW} = \frac{c_{0k}}{\sum_{k=1}^{4} c_{0k}}$$

Problem: find a 6-mer motif in 4 sequences

S_1: GGCTATTGCAGATGACGAGATGAGGCCCAGACC

S_2: GGATGACNNTTATATAAAGGACGATAAGAGATGAC

S_3: CTAGCTCGTAGCTCGTTGAGATGCGCTCCCCGCTC

S_4: GATGACGGAGTATTAAAGACTCGATGAGTTATACGA

1. Initialization: MEME uses a heuristic to estimate the best starting-point matrix:

G	0.260.240.180.260.250.26
A	0.240.260.280.240.250.22
T	0.250.230.300.250.250.25
C	0.250.270.240.250.250.27

2. E-step: MEME scores the match of all 6-mers to current matrix

just consider the underlined 6-mers, Although in reality all 6-mers are scored

GGCTATTGCATATGACGAGATGAGGCCCAGACC

GGATGACNNTTATATAAAGGACCGTGATAAGAGATTAC

CTAGCTCGTAGCTCGTTGAGATGCGCTCCCCGCTC

GATGACGGAGTATTAAAGACTCGATGAGTTATACGA

Base heights corresponds to how much that 6-mer counts in calculating the new matrix

3. M- step: Reestimate the matrix based on the *weighted* contribution of all 6 mers

G	0.290.240.170.270.240.30
A	0.220.260.270.220.280.18
T	0.240.230.330.230.240.28
C	0.240.270.230.280.240.24

Iterate between Steps 2 and 3 until convergence

Summary

- PSSM captures the information on consensus sequence among multiple occurrences of motifs.
- Given PSSM, one can score a long sequence to detect motif occurrences
- MEME uses EM algorithm to learn PSSMs from a set of long sequences