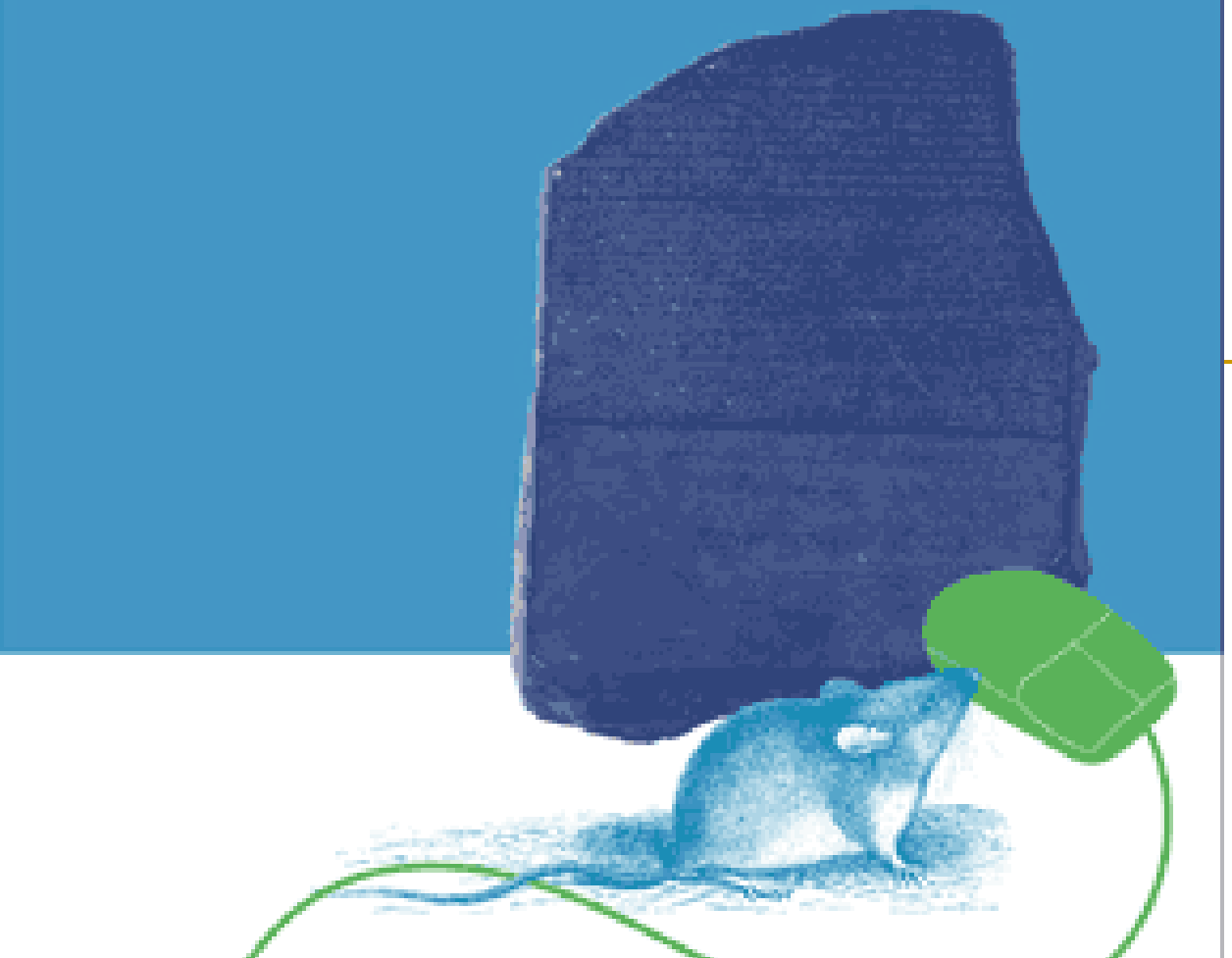
www.bioalgorithms.info

AN INTRODUCTION TO

BIOINFORMATICS ALGORITHMS

NEIL C. JONES AND PAVEL A. PEVZNER



Finding Regulatory Motifs in DNA Sequences

Regulatory Proteins

- Gene X encodes regulatory protein, a.k.a. a ***transcription factor*** (TF)
- The 20 unexpressed genes rely on gene X's TF to induce transcription
- A single TF may regulate multiple genes

Regulatory Regions

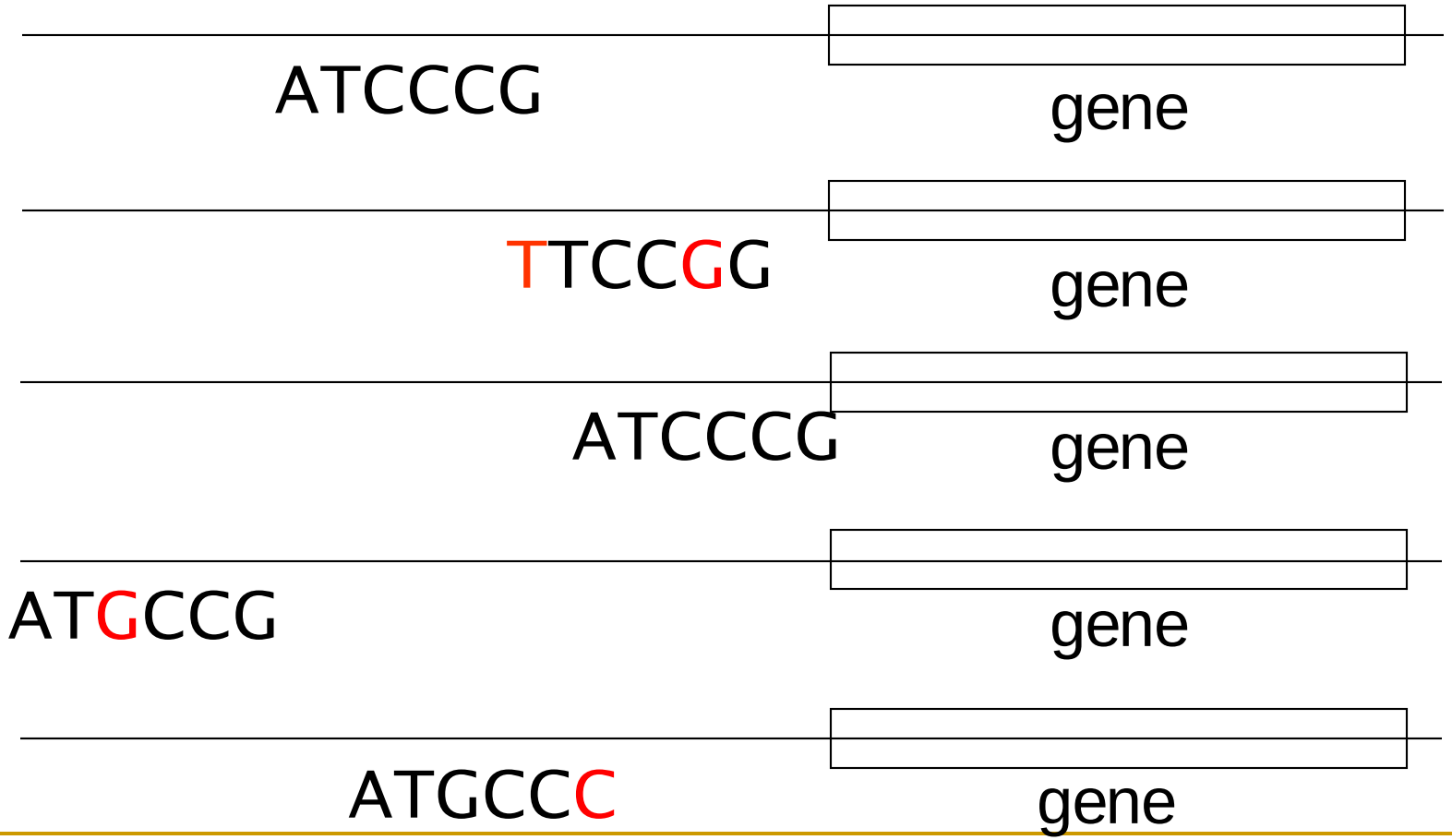
- Every gene contains a regulatory region (RR) typically stretching 100-1000 bp upstream of the transcriptional start site
- Located within the RR are the ***Transcription Factor Binding Sites*** (TFBS), also known as ***motifs***, specific for a given transcription factor
- TFs influence gene expression by binding to a specific location in the respective gene's regulatory region - TFBS

Transcription Factor Binding Sites

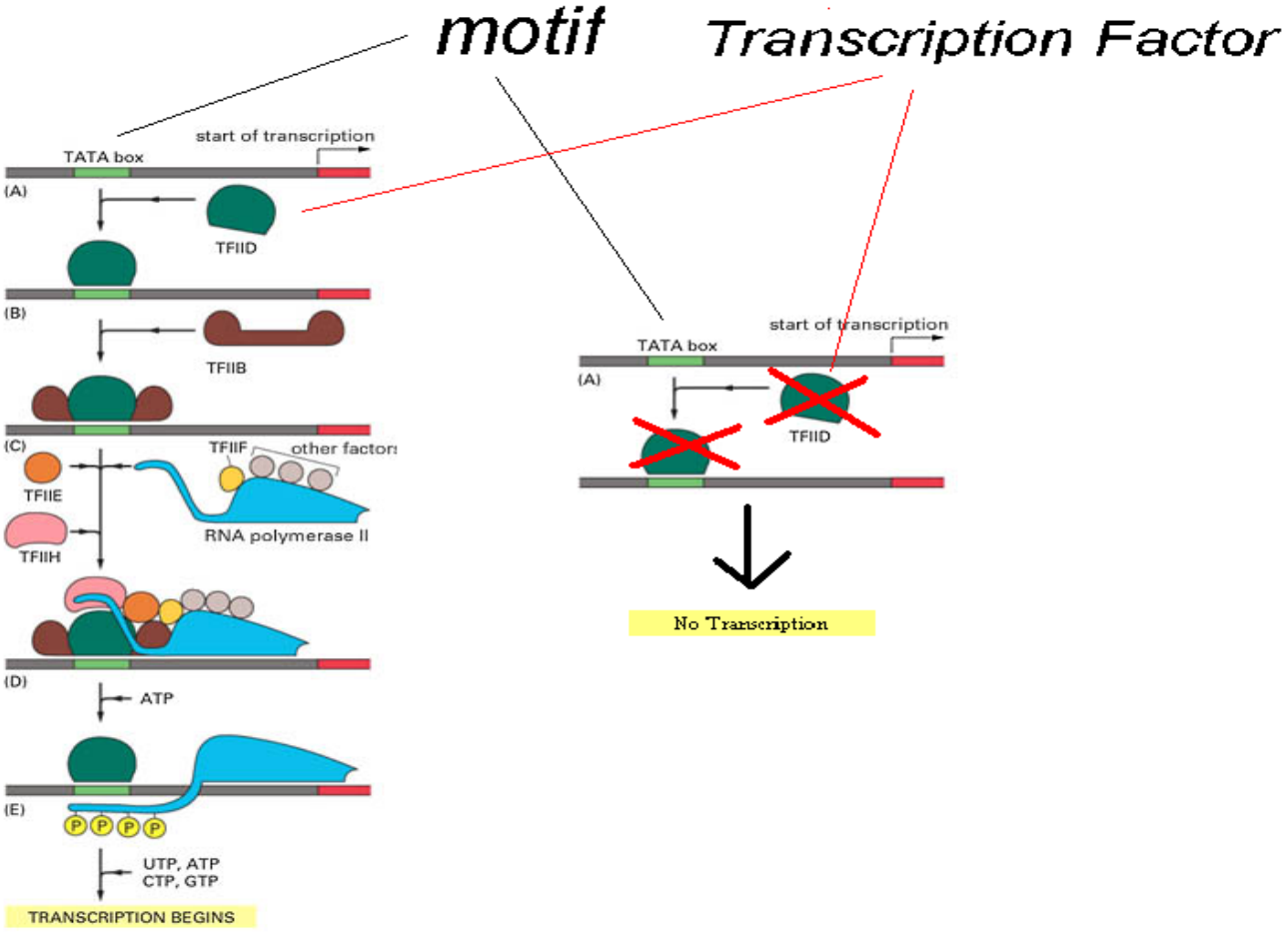
- A TFBS can be located anywhere within the Regulatory Region.

- TFBS may vary slightly across different regulatory regions since non-essential bases could mutate

Motifs and Transcriptional Start Sites



Transcription Factors and Motifs



TATA-box

	−3	−2	−1	0	1	2	3	4	5	6	7	8	9	10	11
A	61	16	352	3	354	268	360	222	155	56	83	82	82	68	77
C	145	46	0	10	0	0	3	2	44	135	147	127	118	107	101
G	152	18	2	2	5	0	20	44	157	150	128	128	128	139	140
T	31	309	35	374	30	121	6	121	33	48	31	52	61	75	71
A	−1·02	−3·05	0·00	−4·61	0·00	0·00	0·00	0·00	−0·01	−0·94	−0·54	−0·48	0·48	−0·74	−0·62
C	−0·28	−2·06	−5·22	−3·49	−5·17	−4·63	−4·12	−3·74	−1·13	−0·05	0·00	−0·05	−0·11	−0·28	−0·40
G	0·00	−2·74	−4·38	−4·61	−3·77	−4·73	−2·65	−1·50	0·00	0·00	−0·09	0·00	0·00	0·00	0·00
T	−1·68	0·00	−2·28	0·00	−2·34	−0·52	−3·65	−0·37	−1·40	−0·97	−1·40	−0·82	−0·66	−0·54	−0·61
	G	T	A	T	A	A	A	A	G	G	C	G	G	G	G
	C		T		T	T		T	A	C	G	C	C	C	C

Preferred region: center between −36 and −20.

Optimized cut-off value: −8·16 (79%)

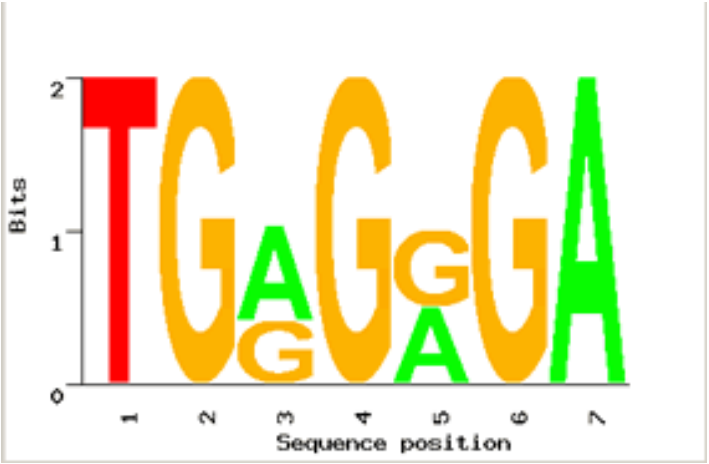
The above base frequency Table and weight matrix were obtained by iterative refinement starting from TATAAA with control parameters set as indicated in Table 1. The consensus sequences shown in Tables 3 to 6 list all bases that are assigned weights above column mean, the character size reflecting their relative importance.

Philipp Bucher, J.Mol. Biol., (1990) 212, 563-578

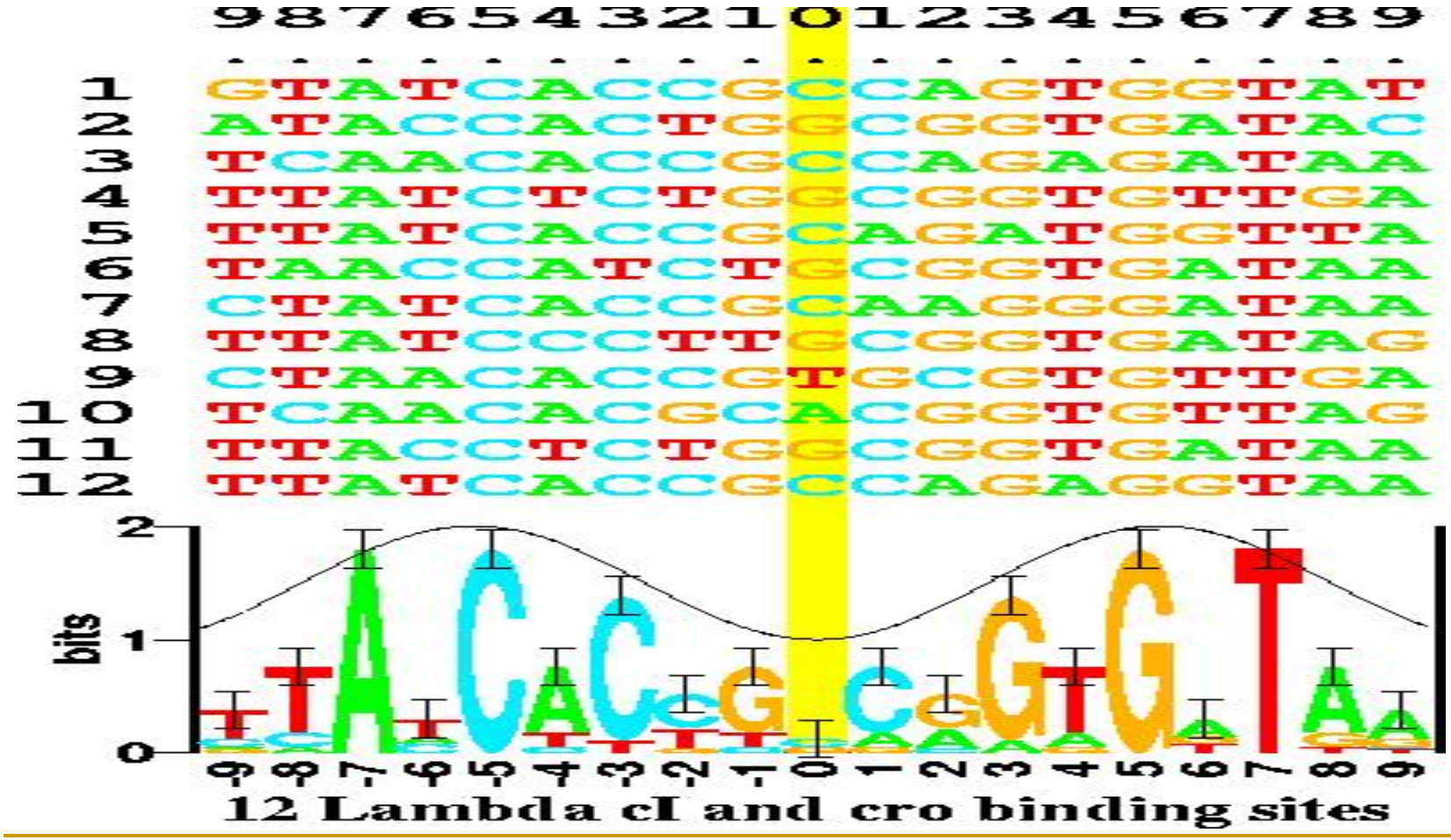
Motif Logo

- Motifs can mutate on non important bases
- The five motifs in five different genes have mutations in position 3 and 5
- Representations called *motif logos* illustrate the conserved and variable regions of a motif

TGGGGGA
TGAGAGA
TGGGGGA
TGAGAGA
TGAGGGA



Motif Logos: An Example



(http://www-lmmb.ncifcrf.gov/~toms/sequencelogo.html)

Identifying Motifs

- Genes are turned on or off by regulatory proteins
- These proteins bind to upstream regulatory regions of genes to either attract or block an RNA polymerase
- Regulatory protein (TF) binds to a short DNA sequence called a motif (TFBS)
- So finding the same motif in multiple genes' regulatory regions suggests a regulatory relationship amongst those genes

Identifying Motifs: Complications

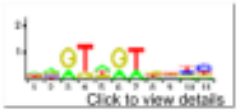
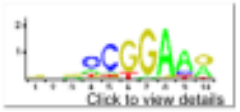
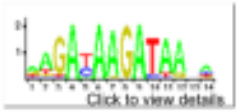
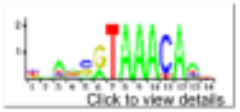
- We do not know the motif sequence
- We do not know where it is located relative to the genes start
- Motifs can differ slightly from one gene to the next
- How to discern it from “random” motifs?

Jaspar data base

- Contains weight matrices for bindings sites of many transcription factors
- http://jaspar.genereg.net/

SEARCH Name AND Species AND Class SEARCH ?

JASPAR matrix models:

TOGGLE	ID	name	species	class	family	Sequence logo
	MA0004.1	Arnt	Mus musculus	Zipper-Type	Helix-Loop-Helix	 Click to view details
	MA0006.1	Arnt::Ahr	Mus musculus	Zipper-Type	Helix-Loop-Helix	 Click to view details
	MA0009.1	T	Mus musculus	Beta-Hairpin-Ribbon	T	 Click to view details
	MA0017.1	NR2F1	Homo sapiens	Zinc-coordinating	Hormone-nuclear Receptor	 Click to view details
	MA0019.1	Ddit3::Cebpa	Rattus norvegicus	Zipper-Type	Leucine Zipper	 Click to view details
	MA0025.1	NFIL3	Homo sapiens	Zipper-Type	Leucine Zipper	 Click to view details
	MA0027.1	En1	Mus musculus	Helix-Turn-Helix	Homeo	 Click to view details
	MA0028.1	ELK1	Homo sapiens	Winged Helix-Turn-Helix	Ets	 Click to view details
	MA0029.1	Mecom	Mus musculus	Zinc-coordinating	BetaBetaAlpha-zinc finger	 Click to view details
	MA0030.1	FOXF2	Homo sapiens	Winged Helix-Turn-Helix	Forkhead	 Click to view details

ANALYZE selected matrix models:

CLUSTER ? selected models using STAMP

Create RANDOM matrix models based on selected models

Number of matrices: 200 Format: Raw

RANDOMIZE ?

Create models with PERMUTED columns from selected:

Type: Within each matrix Format: Raw

PERMUTE ?

SCAN this (fasta-formatted) sequence with selected matrix models

Relative profile score threshold 80 %

SCAN ?

Randomized Algorithms and Motif Finding

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.

A New Motif Finding Approach

- **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.
- **Previously:** we solved the Motif Finding Problem using a Branch and Bound or a Greedy technique.
- **Now**: **randomly** select possible locations and find a way to greedily change those locations until we have converged to the hidden motif.

Profiles Revisited

- Let **s**=(s_1,…,s_t) be the set of starting positions for *l*-mers in our *t* sequences.
- The substrings corresponding to these starting positions will form:
  - *t* x *l* ***alignment matrix*** and
  - 4 x *l* ***profile matrix**** **P**.

*We make a special note that the profile matrix will be defined in terms of the frequency of letters, and not as the count of letters.

Scoring Strings with a Profile

- *Prob*(**a**|**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 of **P** then *Prob*(**a**|**P**) will be high
- If **a** is very different, then *Prob*(**a**|**P**) will be low.

$$Prob(\mathbf{a}|\mathbf{P}) = \prod_{i=1}^{n} p_{a_i,\,i}$$

Scoring Strings 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

The probability of the consensus string:

Prob(**aaacct**|**P**) = ???

Scoring Strings 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

The probability of the consensus string:

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

Scoring Strings 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**

The probability of the consensus string:

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

Probability of a different string:

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

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**.

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

Given a sequence = ctataaaccttacatc, find the P-most probable *l*-mer

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

Find the *Prob*(**a**|**P**) of every possible 6-mer:

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**

-Continue this process to evaluate every possible 6-mer

P-Most Probable *l*-mer (cont'd)

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

String, Highlighted in Red	Calculations	*prob*(**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
ctataaaccttacat	1/2 x 1/8 x 3/8 x 0 x 1/8 x 0	0
ctataaaccttacat	1/8 x 7/8 x 3/8 x 0 x 3/8 x 0	0
ctataaaccttacat	1/2 x 7/8 x 3/8 x 5/8 x 3/8 x 7/8	.0336
ctataaaccttacat	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:

String, Highlighted in Red	Calculations	*Prob*(**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
ctataaaccttacat	1/2 x 1/8 x 3/8 x 0 x 1/8 x 0	0
ctataaaccttacat	1/8 x 7/8 x 3/8 x 0 x 3/8 x 0	0
ctataaaccttacat	**1/2 x 7/8 x 3/8 x 5/8 x 3/8 x 7/8**	**.0336**
ctataaaccttacat	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)

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

ctataaaccttacatc

because *Prob*(**aaacct**|**P**) = .0336 is greater than the *Prob*(**a**|**P**) of any other 6-mer in the sequence.

Dealing with Zeroes

- In our toy example *prob*(**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 small.
- To avoid many entries with *prob*(**a|P**)=0, there exist techniques to equate zero to a very small number so that one zero does not make the entire probability of a string zero (we will not address these techniques here).

P-Most Probable *l*-mers in Many Sequences

- Find the **P**-most probable *l*-mer in each of the sequences.

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

ctataaacgttacatc

atagcgattcgactg

cagcccagaaccct

cggtataccttacatc

tgcattcaatagctta

tatcctttccactcac

ctccaaatcctttaca

ggtcatcctttatcct

P-Most Probable *l*-mers in Many Sequences (cont'd)

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

tgcattcaatagcttа

tgtcctgtccactcac

ctccaaatccttтaca

ggtctaccttтatcct

P-Most Probable *l*-mers form a new profile

Comparing New and Old Profiles

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

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

Red – frequency increased, Blue – frequency descreased

Greedy Profile Motif Search

Use P-Most probable *l*-mers to adjust start positions until we reach a "best" profile; this is the motif.

1) Select random starting positions.
2) Create a profile **P** from the substrings at these starting positions.
3) Find the **P**-most probable *l*-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.

GreedyProfileMotifSearch Algorithm

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

GreedyProfileMotifSearch Analysis

- 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.
- It is unlikely that the random starting positions will lead us to the correct solution at all.
- 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.

Gibbs Sampling

- GreedyProfileMotifSearch is probably not the best way to find motifs.
- However, we can improve the algorithm by introducing **Gibbs Sampling**, an iterative procedure that discards one *l*-mer after each iteration and replaces it with a new one.
- Gibbs Sampling proceeds more slowly and chooses new *l*-mers at random increasing the odds that it will converge to the correct solution.

How Gibbs Sampling Works

1) Randomly choose starting positions **s** = (s_1,...,s_t) and form the set of *l*-mers associated with these starting positions.
2) Randomly choose one of the *t* sequences.
3) Create a profile **P** from the other *t* -1 sequences.
4) For each position in the removed sequence, calculate the probability that the *l*-mer starting at that position was generated by **P**.
5) Choose a new starting position for the removed sequence at random based on the probabilities calculated in step 4.
6) Repeat steps 2-5 until there is no improvement

Gibbs Sampling: an Example

Input:

t = 5 sequences, motif length l = 8

1. GTAAACAATATTTATAGC
2. AAAATTTACCTCGCAAGG
3. CCGTACTGTCAAGCGTGG
4. TGAGTAAACGACGTCCCA
5. TACTTAACACCCTGTCAA

Gibbs Sampling: an Example

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

$s_1=7$ GTAAACAATATTTATAGC

$s_2=11$ AAAATTTACCTTAGAAGG

$s_3=9$ CCGTACTGTCAAGCGTGG

$s_4=4$ TGAGTAAACGACGTCCCA

$s_5=1$ TACTTAACACCCTGTCAA

Gibbs Sampling: an Example

2) Choose one of the sequences at random:

Sequence 2: AAAATTTACCTTAGAAGG

s_1=7	GTAAACAATATTTATAGC
s_2=11	AAAATTTACCTTAGAAGG
s_3=9	CCGTACTGTCAAGCGTGG
s_4=4	TGAGTAAACGACGTCCCA
s_5=1	TACTTAACACCCTGTCAA

Gibbs Sampling: an Example

2) Choose one of the sequences at random:

Sequence 2: AAAATTTACCTTAGAAGG

s_1=7 GTAAACAATATTTATAGC

s_3=9 CCGTACTGTCAAGCGTGG

s_4=4 TGAGTAAACGACGTCCCA

s_5=1 TACTTAACACCCTGTCAA

Gibbs Sampling: an Example

3) Create profile ***P*** from *l*-mers in remaining 4 sequences:

1	A	A	T	A	T	T	T	A
3	T	C	A	A	G	C	G	T
4	G	T	A	A	A	C	G	A
5	T	A	C	T	T	A	A	C
A	1/4	2/4	2/4	3/4	1/4	1/4	1/4	2/4
C	0	1/4	1/4	0	0	2/4	0	1/4
T	2/4	1/4	1/4	1/4	2/4	1/4	1/4	1/4
G	1/4	0	0	0	1/4	0	3/4	0
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**)
AAAATTTACCTTAGAAGG	.000732
AAAATTTACCTTAGAAGG	.000122
AAAATTTACCTTAGAAGG	0
AAAATTTACCTTAGAAGG	0
AAAATTTACCTTAGAAGG	0
AAAATTTACCTTAGAAGG	0
AAAATTTACCTTAGAAGG	0
AAAATTTACCTTAGAAGG	.000183
AAAATTTACCTTAGAAGG	0
AAAATTTACCTTAGAAGG	0
AAAATTTACCTTAGAAGG	0

Gibbs Sampling: an Example

5) Create a distribution of probabilities of *l*-mers *prob*(***a***|***P***), and randomly select a new starting position based on this distribution.

a) To create this distribution, divide each probability *prob*(***a***|***P***) by the lowest probability:

Starting Position 1: *prob(* AAAATTTA | P) = .000732 / .000122 = 6

Starting Position 2: *prob(* AAATTTAC | P) = .000122 / .000122 = 1

Starting Position 8: *prob(* ACCTTAGA | P) = .000183 / .000122 = 1.5

Ratio = 6 : 1 : 1.5

Turning Ratios into Probabilities

b) Define probabilities of starting positions according to computed ratios

Probability (Selecting Starting Position 1): 6/(6+1+1.5)= 0.706

Probability (Selecting Starting Position 2): 1/(6+1+1.5)= 0.118

Probability (Selecting Starting Position 8): 1.5/(6+1+1.5)=0.176

Gibbs Sampling: an Example

c) Select the start position 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

Assume we select the substring with the highest probability – then we are left with the following new substrings and starting positions.

s_1=7	GTAAACAATATTTATAGC
s_2=1	AAAATTTACCTCGCAAGG
s_3=9	CCGTACTGTCAAGCGTGG
s_4=5	TGAGTAATCGACGTCCCA
s_5=1	TACTTCACACCCTGTCAA

Gibbs Sampling: an Example

6) We iterate the procedure again with the above starting positions until we cannot improve the score any more.

Gibbs Sampler in Practice

- Gibbs sampling needs to be modified when applied to samples with unequal distributions of nucleotides (*relative entropy* approach*).*
- Gibbs sampling often converges to locally optimal motifs rather than globally optimal motifs.
- Needs to be run with many randomly chosen seeds to achieve good results.