CSE 6406: BIOINFORMATICS ALGORITHMS
CSE 463: INTRODUCTION TO BIOINFORMATICS

Motif finding problem

Random Sample

atgaccgggatactgataccgtatttggcctaggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg

acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatactgggcataaggtaca

tgagtatccctgggatgacttttgggaacactatagtgctctcccgatttttgaatatgtaggatcattcgccagggtccga

gctgagaattggatgaccttgtaagtgttttccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga

tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatggcccacttagtccacttatag

gtcaatcatgttcttgtgaatggatttttaactgagggcatagaccgcttggcgcacccaaattcagtgtgggcgagcgcaa

cggttttggcccttgttagaggcccccgtactgatggaaactttcaattatgagagagctaatctatcgcgtgcgtgttcat

aacttgagttggtttcgaaaatgctctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta

ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatttcaacgtatgccgaaccgaaagggaag

ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttctgggtactgatagca

Implanting Motif **AAAAAAAGGGGGGG**

atgaccgggatactgatAAAAAAAGGGGGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg

acccctatttttgagcagatttagtgacctggaaaaaaatttgagtacaaaactttccgaataAAAAAAAGGGGGGGa

tgagtatccctgggatgacttAAAAAAAGGGGGGGtgctctcccgattttgaatatgtaggatcattcgccagggtccga

gctgagaattggatgAAAAAAAGGGGGGGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga

tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAAAAAAAGGGGGGGcttatag

gtcaatcatgttcttgtgaatggatttAAAAAAAGGGGGGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa

cggttttggcccttgttagaggcccccgtAAAAAAAGGGGGGGcaattatgagagagctaatctatcgcgtgcgtgttcat

aacttgagttAAAAAAAGGGGGGGctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta

ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatAAAAAAAGGGGGGGaccgaaagggaag

ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttAAAAAAAGGGGGGGa

Where is the Implanted Motif?

atgaccgggatactgataaaaaaaaggggggggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg

acccctatttttgagcagatttagtgacctggaaaaaaatttgagtacaaaacttttccgaataaaaaaaaggggggga

tgagtatccctgggatgacttaaaaaaaaggggggtgctctcccgatttttgaatatgtaggatcattcgccagggtccga

gctgagaattggatgaaaaaaaaggggggtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga

tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaataaaaaaaaggggggcttatag

gtcaatcatgttcttgtgaatggatttaaaaaaaaggggggggaccgcttggcgcacccaaattcagtgtgggcgagcgcaa

cggttttggcccttgttagaggcccccgtaaaaaaaaggggggcaattatgagagagctaatctatcgcgtgcgtgttcat

aacttgagttaaaaaaaaggggggctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta

ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcataaaaaaaaggggggggaccgaaagggaag

ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttaaaaaaaaggggggga

Implanting Motif **AAAAAAGGGGGGG** with Four Mutations

atgaccgggatactgatAgAAgAAAGGttGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg

acccctatttttgagcagatttagtgacctggaaaaaaatttgagtacaaaacttttccgaatacAAtAAAAcGGcGGGa

tgagtatccctgggatgacttAAAAtAAtGGaGtGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccga

gctgagaattggatgcAAAAAAAGGGattGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga

tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAtAAtAAAGGaaGGGcttatag

gtcaatcatgttcttgtgaatggatttAAcAAtAAGGGctGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa

cggttttggcccttgttagaggcccccgtAtAAAcAAGGaGGGccaattatgagagagctaatctatcgcgtgcgtgttcat

aacttgagttAAAAAAtAGGGaGccctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta

ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatActAAAAAGGaGcGGaccgaaagggaag

ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttActAAAAAGGaGcGGa

Where is the Motif???

atgaccgggatactgatagaagaaaggttgggggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg

acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatacaataaaacggcggga

tgagtatccctgggatgacttaaaataatggagtggtgctctcccgatttttgaatatgtaggatcattcgccagggtccga

gctgagaattggatgcaaaaaaagggattgtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga

tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatataataaaggaagggcttatag

gtcaatcatgttcttgtgaatggatttaacaataagggctgggaccgcttggcgcacccaaattcagtgtgggcgagcgcaa

cggttttggcccttgttagaggcccccgtataaacaaggagggccaattatgagagagctaatctatcgcgtgcgtgttcat

aacttgagttaaaaaatagggagccctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta

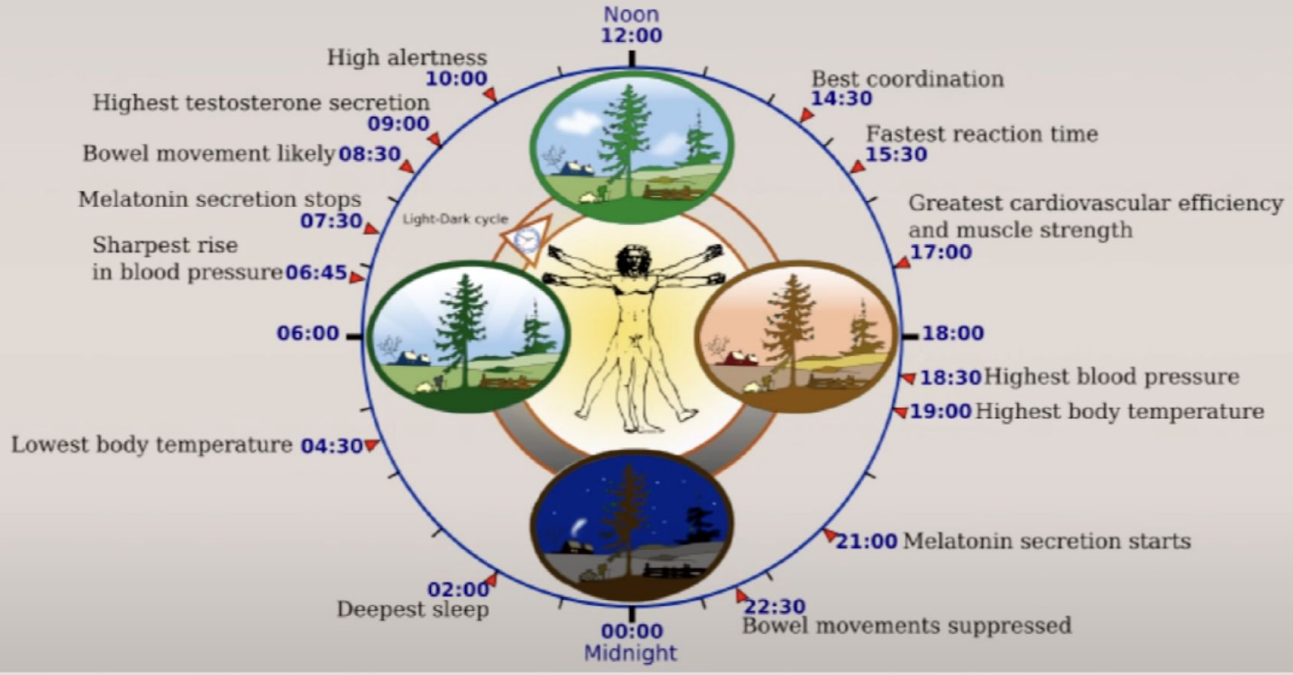
ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatactaaaaaggagcggaccgaaagggaag

ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttactaaaaaggagcgga

Why Finding (15,4) Motif is Difficult?

Circadian Clock

Our schedules are controlled by a molecular timekeeper called the **circadian clock**.



Challenge Problem

- Find a motif in a sample of
  - 20 "random" sequences (e.g. 600 nt long)
  - each sequence containing an implanted pattern of length 15,
  - each pattern appearing with 4 mismatches as (15,4)-motif.

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

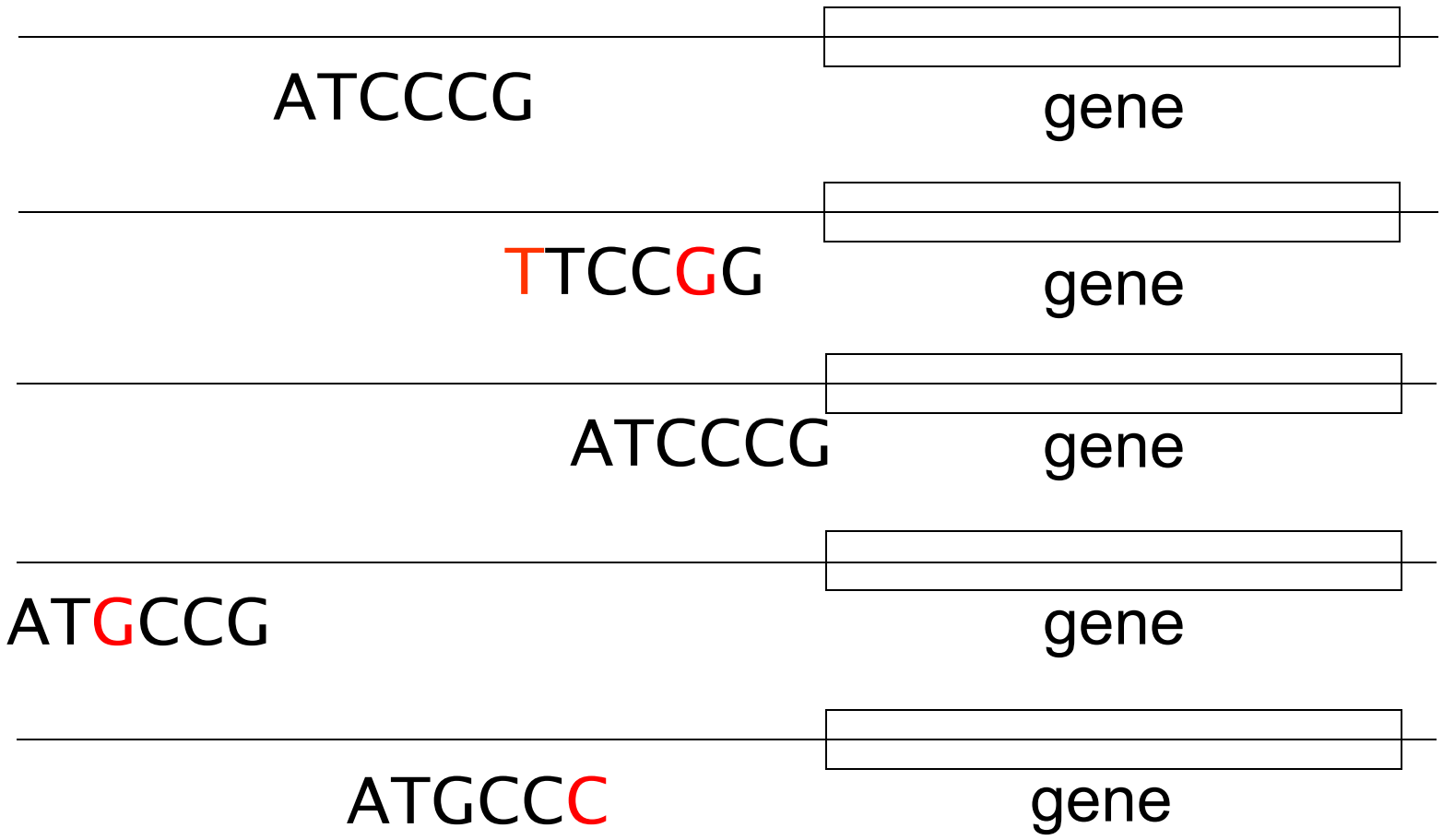
- TFs influence gene expression by binding to a specific location in the respective gene's regulatory region - TFBS

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

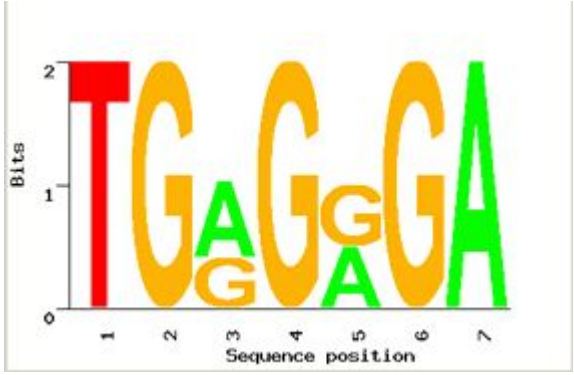
Motifs and Transcriptional Start Sites



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



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?

The Motif Finding Problem

- Given a random sample of DNA sequences:

```
cctgatagacgctatctggctatccacgtacgtaggtcctctgtgcgaatctatgcgtttccaaccat
agtactggtgtacatttgatacgtacgtacaccggcaacctgaaacaaacgctcagaaccagaagtgc
aaacgtacgtgcaccctctttcttcgtggctctggccaacgagggctgatgtataagacgaaaatttt
agcctccgatgtaagtcatagctgtaactattacctgccacccctattacatcttacgtacgtataca
ctgttatacaacgcgtcatggcggggtatgcgttttggtcgtcgtacgctcgatcgttaacgtacgtc
```

- Find the pattern that is implanted in each of the individual sequences, namely, the motif

The Motif Finding Problem (cont'd)

- Additional information:
  - The hidden sequence is of length 8
  - The pattern is not exactly the same in each array because random point mutations may occur in the sequences

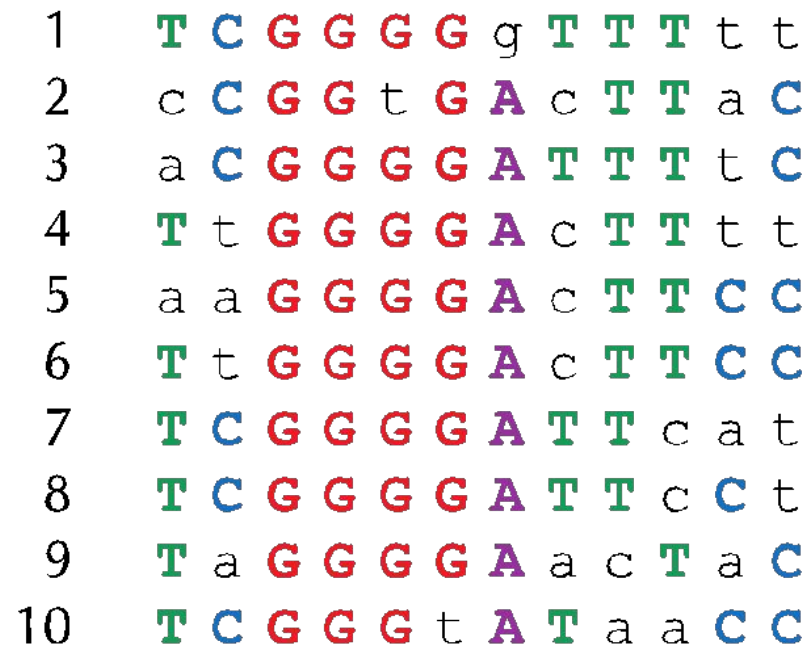
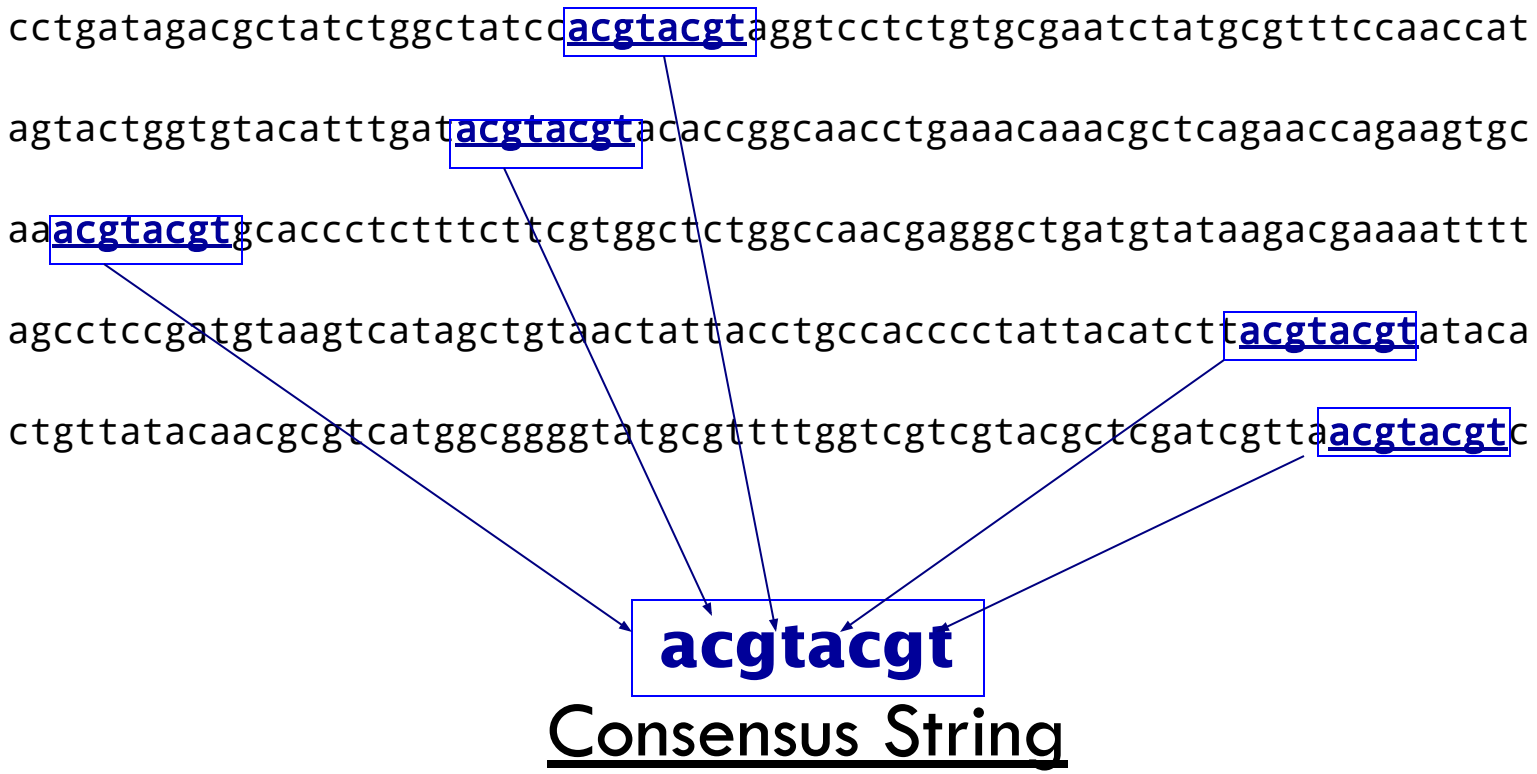


FIGURE 2.1 The ten candidate NF-$\varkappa$B binding sites appearing in the *Drosophila melanogaster* genome. The upper case colored letters indicate the most frequent nucleotide in each column.


1 at gaccgggat act gat AAAAAAAAGGGGGGGggcgt acacat t agat aaacgt at gaagt acgt t agact cggcgccgccg
2 acccct at t t t t t gagcagat t t agt gacct ggaaaaaaaat t t gagt acaaaact t t t ccgaat a AAAAAAAAGGGGGGGa
3 t gagt at ccct gggat gact t AAAAAAAAGGGGGGGt gct ct cccgat t t t t gaat at gt aggat cat t cgccagggt ccga
4 gct gagaat t ggat g AAAAAAAAGGGGGGGt ccacgcaat cgcgaaccaacgcggacccaaaggcaagaccgat aaaggaga
5 t ccct t t t gcggt aat gt gccgggaggct ggt t acgt agggaagccct aacggact t aat AAAAAAAAGGGGGGGct t at ag
6 gt caat cat gt t ct t gt gaat ggat t t AAAAAAAAGGGGGGGgaccgct t ggcgcacccaaat t cagt gt gggcgagcgcaa
7 cggt t t t ggccct t gt t agaggccccgt AAAAAAAAGGGGGGGcaat t at gagagagct aat ct at cgcgt gcgt gt t cat
8 aact t gagt t AAAAAAAAGGGGGGGct ggggcacat acaagaggagt ct t cct t at cagt t aat gct gt at gacact at gt a
9 t t ggcccat t ggct aaaagcccaact t gacaaat ggaagat agaat cct t gcat AAAAAAAAGGGGGGGaccgaaagggaag
10 ct ggt gagcaacgacagat t ct t acgt gcat t agct cgct t ccggggat ct aat agcacgaagct t AAAAAAAAGGGGGGGa
1 at gaccgggat act gat AgAAgAAAGGt t GGGggcgt acacat t agat aaacgt at gaagt acgt t agact cggcgccgccg
2 acccct at t t t t t gagcagat t t agt gacct ggaaaaaaaat t t gagt acaaaact t t t ccgaat a cAAt AAAAcGGcGGGa
3 t gagt at ccct gggat gact t AAAAt AAt GGaGt GGt gct ct cccgat t t t t gaat at gt aggat cat t cgccagggt ccga
4 gct gagaat t ggat gcAAAAAAAGGGatt Gt ccacgcaat cgcgaaccaacgcggacccaaaggcaagaccgat aaaggaga
5 t ccct t t t gcggt aat gt gccgggaggct ggt t acgt agggaagccct aacggact t aat At AAt AAAGGaaGGGct t at ag
6 gt caat cat gt t ct t gt gaat ggat t t AAcAAt AAGGGct GGgaccgct t ggcgcacccaaat t cagt gt gggcgagcgcaa
7 cggt t t t ggccct t gt t agaggccccgt At AAAcAAGGaGGGccaat t at gagagagct aat ct at cgcgt gcgt gt t cat
8 aact t gagt t AAAAAAt AGGGaGccct ggggcacat acaagaggagt ct t cct t at cagt t aat gct gt at gacact at gt a
9 t t ggcccat t ggct aaaagcccaact t gacaaat ggaagat agaat cct t gcat Act AAAAAGGaGcGGaccgaaagggaag
10 ct ggt gagcaacgacagat t ct t acgt gcat t agct cgct t ccggggat ct aat agcacgaagct t Act AAAAAGGaGcGGa
AgAAgAAAGGttGGG
cAAtAAAAcGGGGcG

The Motif Finding Problem (cont'd)

- The patterns revealed with no mutations:



The Motif Finding Problem (cont'd)

- The patterns with 2 point mutations:

cctgatagacgctatctggctatcc**aGgtacTt**aggtcctctgtgcgaatctatgcgtttccaaccat

agtactggtgtacatttgat**CcAtacgt**acaccggcaacctgaaacaaacgctcagaaccagaagtgc

aa**acgtTAgt**gcaccctctttcttcgtggctctggccaacgagggctgatgtataagacgaaaatttt

agcctccgatgtaagtcatagctgtaactattacctgccacccctattacatctt**acgtCcAt**ataca

ctgttatacaacgcgtcatggcggggtatgcgttttggtcgtcgtacgctcgatcgtta**CcgtacgG**c

The Motif Finding Problem (cont'd)

- The patterns with 2 point mutations:

cctgatagacgctatctggctatcc**aGgtacTt**aggtcctctgtgcgaatctatgcgtttccaaccat

agtactggtgtacatttgat**CcAtacgt**acaccggcaacctgaaacaaacgctcagaaccagaagtgc

aa**acgtTAgt**gcaccctctttcttcgtggctctggccaacgagggctgatgtataagacgaaaatttt

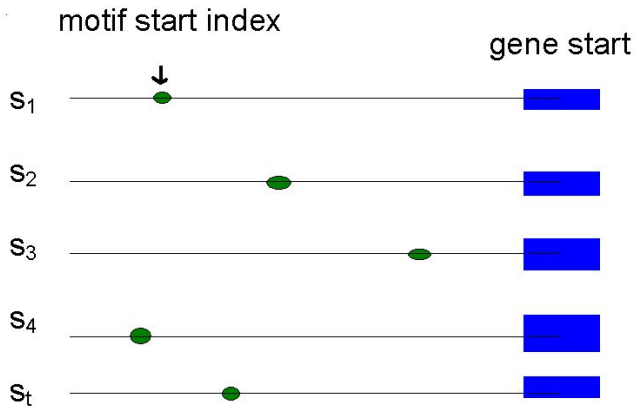
agcctccgatgtaagtcatagctgtaactattacctgccacccctattacatctt**acgtCcAt**ataca

ctgttatacaacgcgtcatggcggggtatgcgttttggtcgtcgtacgctcgatcgtta**CcgtacgG**c

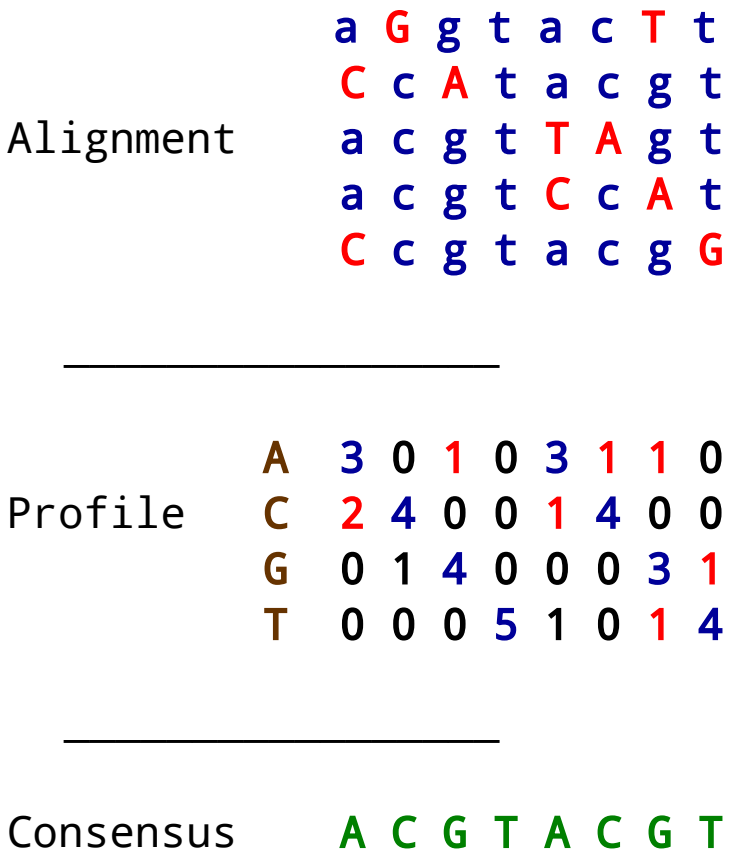
Can we still find the motif, now that we have 2 mutations?

Defining Motifs

- To define a motif, let us say we know where the motif starts in the sequence
- The motif start positions in their sequences can be represented as $s = (s_1, s_2, s_3, \ldots, s_t)$



Motifs: Profiles and Consensus



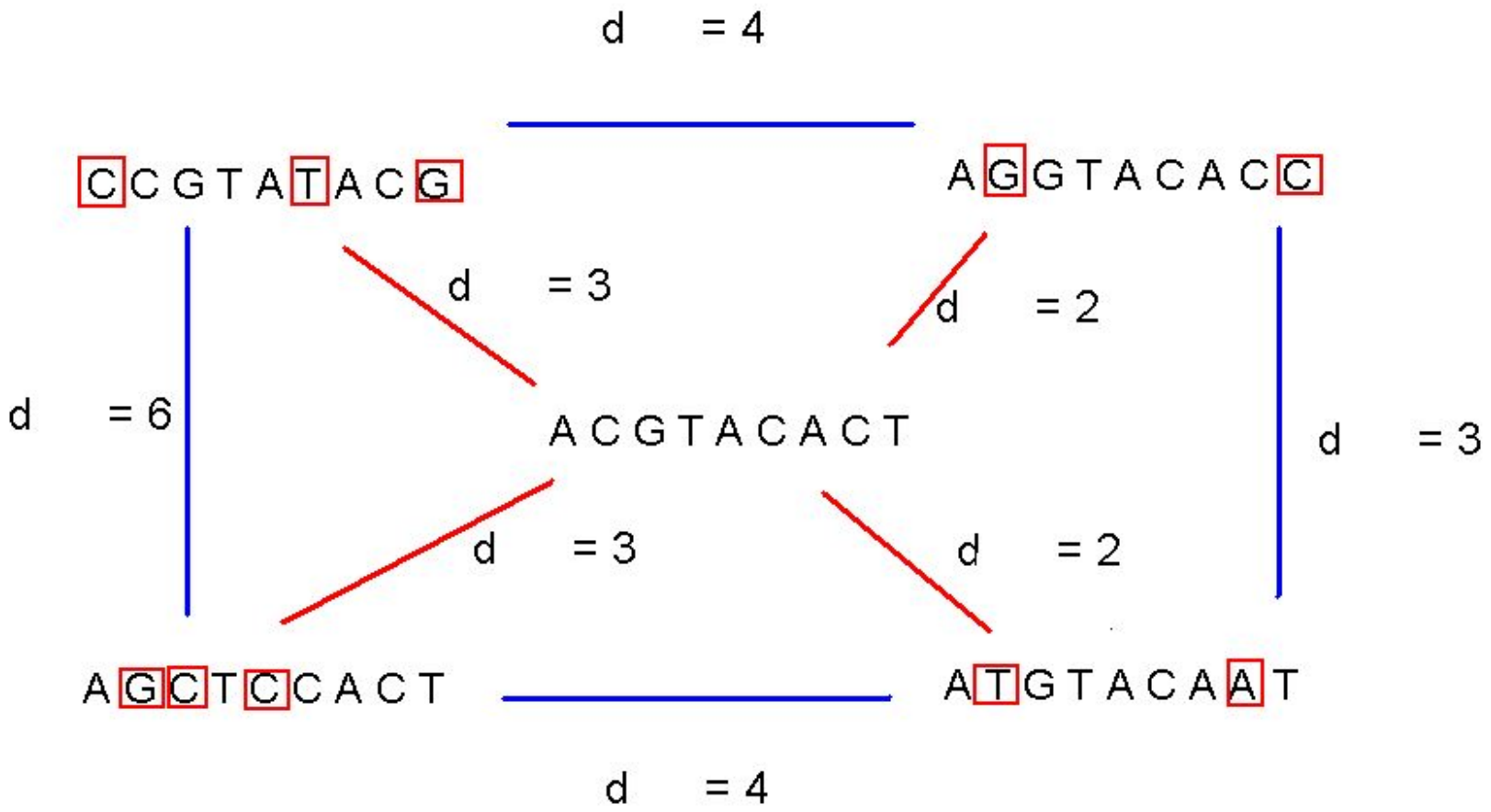
- Line up the patterns by their start indexes
  $\mathbf{s} = (s_1, s_2, \ldots, s_t)$
- Construct matrix profile with frequencies of each nucleotide in columns
- Consensus nucleotide in each position has the highest score in column

Consensus

- Think of consensus as an "ancestor" motif, from which mutated motifs emerged

- The *distance* between a real motif and the consensus sequence is generally less than that for two real motifs

Consensus (cont'd)



Evaluating Motifs

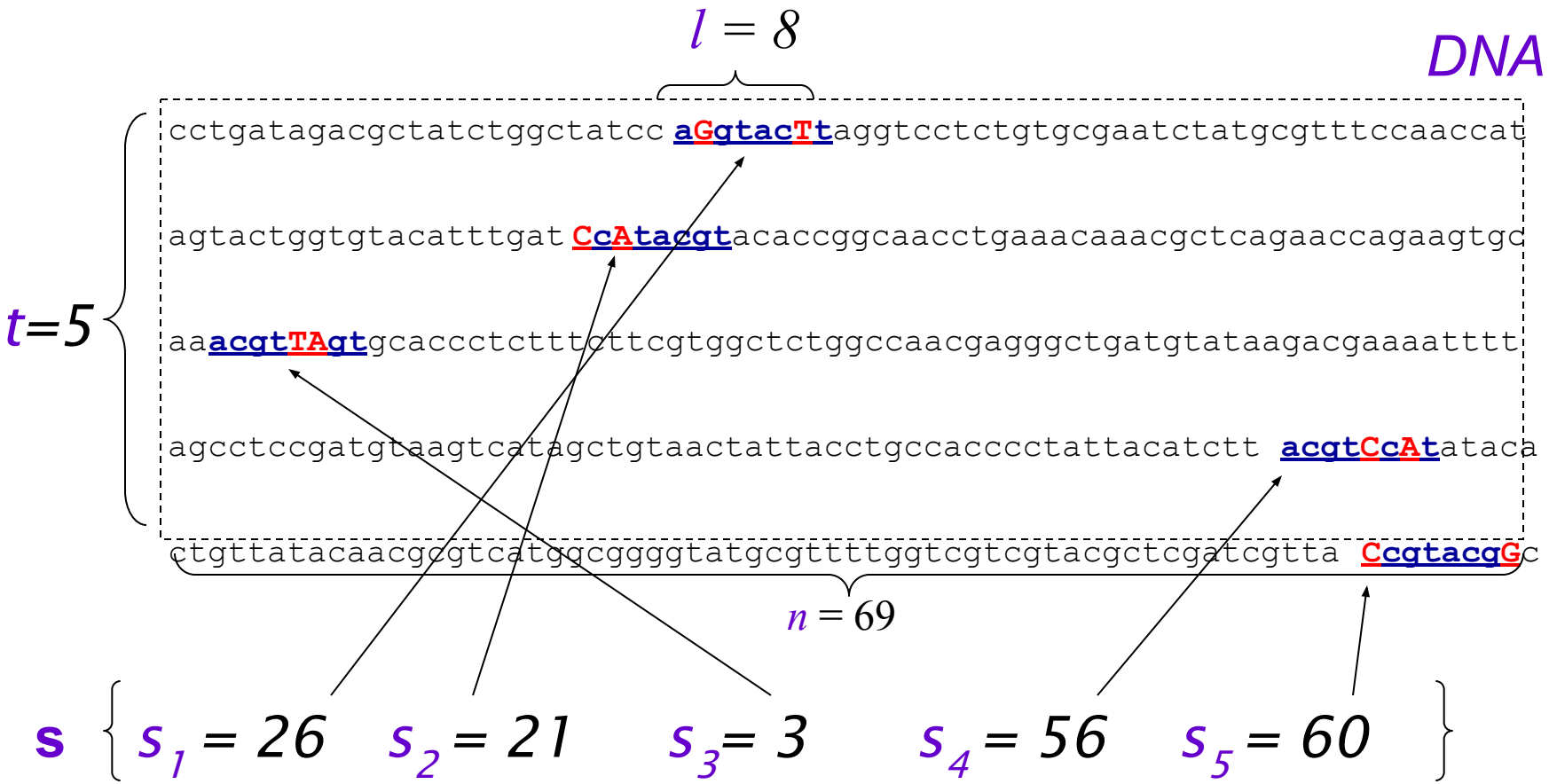
- We have a guess about the consensus sequence, but how "good" is this consensus?

- Need to introduce a scoring function to compare different guesses and choose the "best" one.

Defining Some Terms

- ***t*** - number of sample DNA sequences
- ***n*** - length of each DNA sequence
- ***DNA*** - sample of DNA sequences (*t* x *n* array)
- ***l*** - length of the motif (***l***-mer)
- s_i - starting position of an ***l***-mer in sequence *i*
- **s**=(s_1, s_2,… s_t) - array of motif's starting positions

Parameters


l = 8
DNA
cctgatagacgctatctggctatcc aGgtacTtaggtcctctgtgcgaatctatgcgtttccaaccat
agtactggtgtacatttgat CcAtacgtacaccggcaacctgaaacaaacgctcagaaccagaagtgc
aaacgtTAgtgcaccctctttcttcgtggctctggccaacgagggctgatgtataagacgaaaatttt
agcctccgatgtaagtcatagctgtaactattacctgccacccctattacatctt acgtCcAtatacа
ctgttatacaacgcgtcatggcggggtatgcgttttggtcgtcgtacgctcgatcgtta CcgtacgGc
t=5
n = 69
s
s1 = 26
s2 = 21
s3 = 3
s4 = 56
s5 = 60

Scoring Motifs

Motifs	T	C	G	G	G	G	g	T	T	T	t	t
	c	C	G	G	t	G	A	c	T	T	a	C
	a	C	G	G	G	G	A	T	T	T	t	C
	T	t	G	G	G	G	A	c	T	T	t	t
	a	a	G	G	G	G	A	c	T	T	C	C
	T	t	G	G	G	G	A	c	T	T	C	C
	T	C	G	G	G	G	A	T	T	c	a	t
	T	C	G	G	G	G	A	T	T	c	C	t
	T	a	G	G	G	G	A	a	c	T	a	C
	T	C	G	G	G	t	A	T	a	a	C	C

SCORE(*Motifs*) 3 + 4 + 0 + 0 + 1 + 1 + 1 + 5 + 2 + 3 + 6 + 4 = 30

COUNT(*Motifs*)												
A:	2	2	0	0	0	0	9	1	1	1	3	0
C:	1	6	0	0	0	0	0	4	1	2	4	6
G:	0	0	10	10	9	9	1	0	0	0	0	0
T:	7	2	0	0	1	1	0	5	8	7	3	4

PROFILE(*Motifs*)												
A:	.2	.2	0	0	0	0	.9	.1	.1	.1	.3	0
C:	.1	.6	0	0	0	0	0	.4	.1	.2	.4	.6
G:	0	0	1	1	.9	.9	.1	0	0	0	0	0
T:	.7	.2	0	0	.1	.1	0	.5	.8	.7	.3	.4

CONSENSUS(*Motifs*) T C G G G G A T T T C C

PROFILE(*Motifs*)	**A:**	.2	.2	0	0	0	0	**.9**	.1	.1	.1	.3	0
	C:	.1	**.6**	0	0	0	0	0	.4	.1	.2	.4	**.6**
	G:	0	0	**1**	**1**	**.9**	**.9**	.1	0	0	0	0	0
	T:	**.7**	.2	0	0	.1	.1	0	**.5**	**.8**	**.7**	.3	.4
CONSENSUS(*Motifs*)		**T**	**C**	**G**	**G**	**G**	**G**	**A**	**T**	**T**	**T**	**C**	**C**

$$H(p_1, \ldots, p_N) = -\sum_{i=1}^{N} p_i \cdot \log_2(p_i).$$

For example, the entropy of the probability distribution (0.2, 0.6, 0.0, 0.2) corresponding to the second column of the profile matrix in Figure 2.2 is

$$-(0.2\log_2 0.2 + 0.6\log_2 0.6 + 0.0\log_2 0.0 + 0.2\log_2 0.2) \approx 1.371,$$

whereas the entropy of the more conserved final column (0.0, 0.6, 0.0, 0.4) is

$$-(0.0\log_2 0.0 + 0.6\log_2 0.6 + 0.0\log_2 0.0 + 0.4\log_2 0.4) \approx 0.971,$$

and the entropy of the very conserved 5th column (0.0, 0.0, 0.9, 0.1) is

$$-(0.0\log_2 0.0 + 0.0\log_2 0.0 + 0.9\log_2 0.9 + 0.1\log_2 0.1) \approx 0.467.$$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C	G/C	G/T	T/A	C/T	G/C	C/G	A	T	G/T	C/G	A	T	C/T	C/T	G/T

FIGURE 2.3 The CSRE transcription factor binding site in *Saccharomyces cerevisiae* is 16 nucleotides long, but only five of these positions (1, 8, 9, 12, 13) are strongly conserved. The remaining 11 positions can take one of two different nucleotides.

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

FIGURE 2.4 Taking nucleotides in each column of the NF-kB binding site motif matrix from Figure 2.2 with frequency at least 0.4 yields a representation of the NF-kB binding sites with ten strongly conserved positions and two weakly conserved positions (8 and 12).

The Motif Finding Problem: Formulation

Motif Finding Problem:
Given a collection of strings, find a set of k-mers, one from each string, that minimizes the score of the resulting motif.

Input: A collection of strings *Dna* and an integer k.
Output: A collection *Motifs* of k-mers, one from each string in *Dna*, minimizing SCORE(*Motifs*) among all possible choices of k-mers.

The Motif Finding Problem: Brute Force Solution

1. BruteForceMotifSearch(***DNA***, ***t***, ***n***, ***l***)
2. ***bestScore*** □ 0
3. **for** each **s** = (s_1,s_2 , . . ., s_t) from (1,1 . . . 1)
   to (***n***-***l***+1, . . ., ***n***-***l***+1)
4. **if** (*Score*(**s**,***DNA***) < ***bestScore***)
5. ***bestScore*** □ *score*(**s**, ***DNA***)
6. **bestMotif** □ (s_1,s_2 , . . . , s_t)
7. **return bestMotif**

Running Time of BruteForceMotifSearch

- Varying $(n - \ell + 1)$ positions in each of t sequences, we're looking at $(n - \ell + 1)^t$ sets of starting positions

- For each set of starting positions, the scoring function makes ℓ operations, so complexity is $\ell(n - \ell + 1)^t = O(\ell n^t)$

- That means that for $t = 8$, $n = 1000$, $\ell = 10$ we must perform approximately 10^{24} computations – it will take billions years

Motifs

T	C	G	G	G	G	g	T	T	T	t	t
c	C	G	G	t	G	A	c	T	T	a	C
a	C	G	G	G	G	A	T	T	T	t	C
T	t	G	G	G	G	A	c	T	T	t	t
a	a	G	G	G	G	A	c	T	T	C	C
T	t	G	G	G	G	A	c	T	T	C	C
T	C	G	G	G	G	A	T	T	c	a	t
T	C	G	G	G	G	A	T	T	c	C	t
T	a	G	G	G	G	A	a	c	T	a	C
T	C	G	G	G	t	A	T	a	a	C	C

SCORE(*Motifs*) $3 + 4 + 0 + 0 + 1 + 1 + 1 + 5 + 2 + 3 + 6 + 4 = 30$

Motifs

T	C	G	G	G	G	g	T	T	T	t	t	**3**
c	C	G	G	t	G	A	c	T	T	a	C	**4**
a	C	G	G	G	G	A	T	T	T	t	C	**2**
T	t	G	G	G	G	A	c	T	T	t	t	**4**
a	a	G	G	G	G	A	c	T	T	C	C	**3**
T	t	G	G	G	G	A	c	T	T	C	C	**2**
T	C	G	G	G	G	A	T	T	c	a	t	**3**
T	C	G	G	G	G	A	T	T	c	C	t	**2**
T	a	G	G	G	G	A	a	c	T	a	C	**4**
T	C	G	G	G	t	A	T	a	a	C	C	**+ 3**

SCORE(*Motifs*) $3 + 4 + 0 + 0 + 1 + 1 + 1 + 5 + 2 + 3 + 6 + 4 = \mathbf{30}$

CONSENSUS(*Motifs*) T C G G G G A T T T C C

The Median String Problem

- Given a set of *t* DNA sequences find a pattern that appears in all *t* sequences with the minimum number of mutations

- This pattern will be the motif

Hamming Distance

- Hamming distance:
  - $d_H(\mathbf{v}, \mathbf{w})$ is the number of nucleotide pairs that do not match when **v** and **w** are aligned. For example:

$$d_H(\text{AAAAAA}, \text{ACAAAC}) = 2$$

Distance between a *k*-mer and a (longer) String

d(GATTCTCA, GCAAAGACGCTGACCAA) = ?

Distance: 7 6 **7**

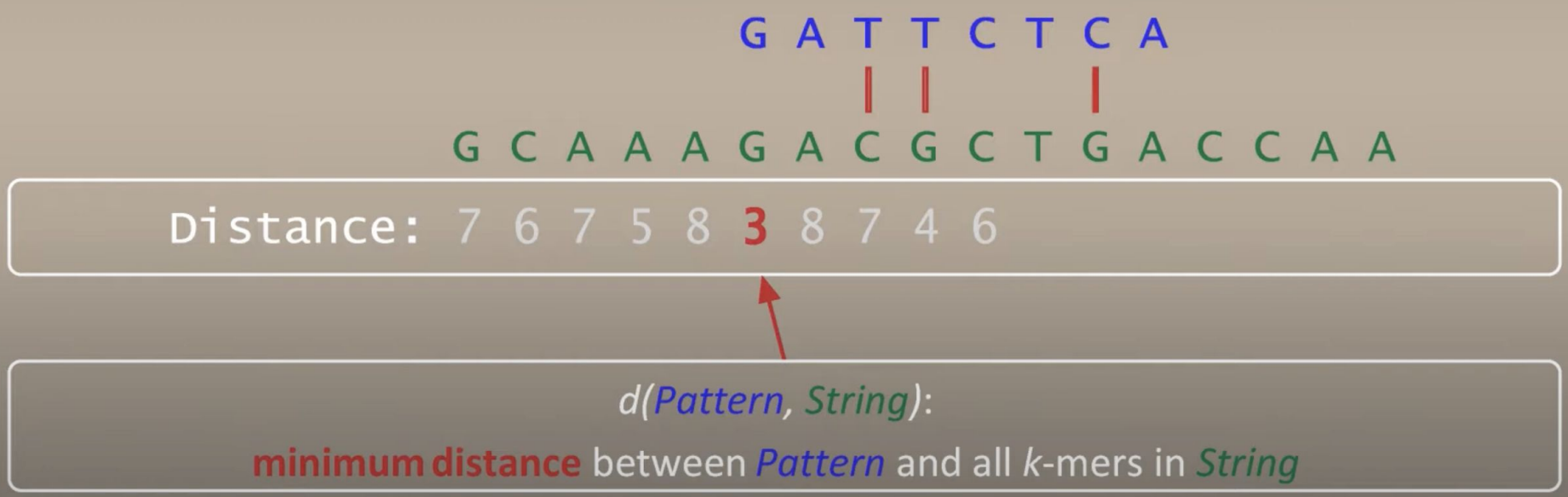
d(*Pattern*, *String*):

minimum distance between *Pattern* and all *k*-mers in *String*

https://www.youtube.com/watch?v=4wo2d2_OMyE

Distance between a *k*-mer and a (longer) String

d(GATTCTCA, GCAAAGACGCTGACCAA) = **3**



d(*Pattern*, *String*):

minimum distance between *Pattern* and all *k*-mers in *String*

Distance between a *k* – mer and a **Set** of Strings

Distance between a *k*-mer and *a set of strings* $Dna = \{Dna_1, \ldots, DNA_t\}$:

$d(k\text{-mer}, Dna) = \sum_{\text{all strings in } Dna} d(k\text{-mer}, Dna_i)$

Pattern = AAA

ttaccttAAc	1
gAtAtctgtc	1
Acggcgttcg	2
ccctAAAgag	0
cgtcAgAggt	1
d(AAA, *Dna*) =	5

A **median string** for the set of strings *Dna*:
a *k*-mer minimizing distance
d(*k*-mer, *Dna*)
over all possible k-mers.

Total Distance: An Example

- Given **v** = "acgtacgt" and **s**

v is the sequence in red, *x* is the sequence in blue

- *TotalDistance*(***v***,***DNA***) = 0

Total Distance: Example

- Given **v** = "acgtacgt" and **s**

v is the sequence in red, *x* is the sequence in blue

- *TotalDistance*(***v,DNA***) = 1+0+2+0+1 = 4

Total Distance: Definition

- For each DNA sequence ***i***, compute all $d_H(\mathbf{v}, \mathbf{x})$, where ***x*** is an ℓ-mer with starting position s_i
  ($1 \leq s_i \leq n - \ell + 1$)
- Find minimum of $d_H(\mathbf{v}, \mathbf{x})$ among all ℓ-mers in sequence ***i***
- *TotalDistance*(***v***, ***DNA***) is the sum of the minimum Hamming distances for each DNA sequence ***i***
- *TotalDistance*(***v***, ***DNA***) $= min_{\mathbf{s}}\, d_H(\mathbf{v}, \mathbf{s})$, where ***s*** is the set of starting positions $s_1, s_2, \ldots s_t$

The Median String Problem: Formulation

- Goal: Given a set of DNA sequences, find a median string
- Input: A ***t*** x ***n*** matrix DNA, and ℓ, the length of the pattern to find
- Output: A string **v** of ℓ nucleotides that minimizes *TotalDistance*(**v**,***DNA***) over all strings of that length

Median String Problem

Median String Problem. Finding a median string.

- **Input:** A set of sequences *Dna* and an integer *k*.
- **Output:** A *k*-mer minimizing distance *d*(*k*-mer, *Dna*) among all *k*-mers.

MedianString(*Dna*, *k*)

best-k-mer ← AAA · · · AA
for each *k*-mer from AAA · · · AA to TTT · · · TT
  if *d*(*k*-mer, *Dna*) < *distance*(*best-k-mer*, *Dna*)
    best-k-mer ← *k-mer*
return(*best-k-mer*)

Runtime: $4^k \cdot n \cdot t \cdot k$ (for *Dna* with *t* sequences of length *n*).

Motif Finding Problem
versus
Median String Problem

Runtime: $n^t \cdot k \cdot t$

Runtime: $4^k \cdot n \cdot t \cdot k$

Median String Search Algorithm

```
MEDIANSTRING(Dna, k)
    distance ← ∞
    for each k-mer Pattern from AA...AA to TT...TT
        if distance > d(Pattern, Dna)
            distance ← d(Pattern, Dna)
            Median ← Pattern
    return Median
```

with $k = 13$ in the hope that it will capture a substring of the correct 15-mer motif. The algorithm still requires half a day to run on our computer and returns the median string **AAAAAt AGa GGGG** (with distance 29). This 13-mer is not a substring of the implanted pattern **AAAAAAAAGGGGGGG**, but it does come close.

GREEDY MOTIF SEARCH

- Select most probable motifs based on profile
- Augment the profile
- How is the profile generated initially?
- How is the first motif chosen?

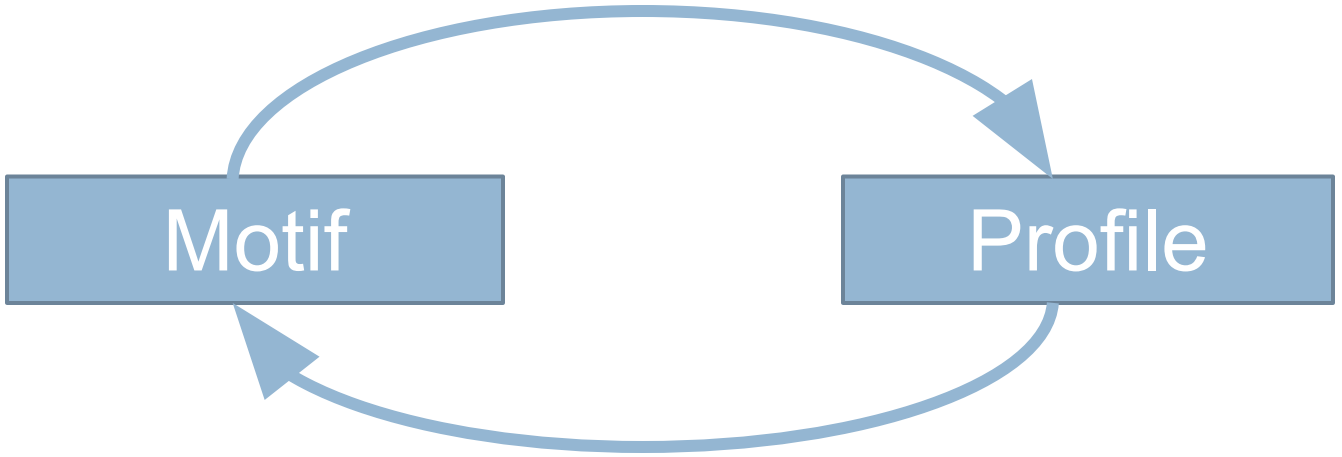
Profile

A:	.2	.2	.0	.0	.0	.0	.9	.1	.1	.1	.3	.0
C:	.1	.6	.0	.0	.0	.0	.0	.4	.1	.2	.4	.6
G:	.0	.0	1	1	.9	.9	.1	.0	.0	.0	.0	.0
T:	.7	.2	.0	.0	.1	.1	.0	.5	.8	.7	.3	.4

$$\Pr(\text{ACGGGGATTACC}|\textit{Profile})$$

$$= .2 \cdot .6 \cdot 1 \cdot 1 \cdot .9 \cdot .9 \cdot .9 \cdot .5 \cdot .8 \cdot .1 \cdot .4 \cdot .6 = 0.000839808$$

Circular Dependency



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

$\sum$ =?

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:

ctataaacct**tacatc**

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

GREEDY MOTIF SEARCH

```
GREEDYMOTIFSEARCH(Dna, k, t)
    BestMotifs ← motif matrix formed by first k-mers in each string from Dna
    for each k-mer Motif in the first string from Dna
        Motif_1 ← Motif
        for i = 2 to t
            form Profile from motifs Motif_1, ..., Motif_{i−1}
            Motif_i ← Profile-most probable k-mer in the i-th string in Dna
        Motifs ← (Motif_1, ..., Motif_t)
        if SCORE(Motifs) < SCORE(BestMotifs)
            BestMotifs ← Motifs
    return BestMotifs
```

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).

Dealing with Zeroes

Profile	A:	.2	.2	.0	.0	.0	.0	.9	.1	.1	.1	.3	.0
	C:	.1	.6	.0	.0	.0	.0	.0	.4	.1	.2	.4	.6
	G:	.0	.0	1	1	.9	.9	.1	.0	.0	.0	.0	.0
	T:	.7	.2	.0	.0	.1	.1	.0	.5	.8	.7	.3	.4

$$\Pr(\texttt{TCGTGGATTTCC}|Profile) = .7 \cdot .6 \cdot 1 \cdot .0 \cdot .9 \cdot .9 \cdot .9 \cdot .5 \cdot .8 \cdot .7 \cdot .4 \cdot .6 = 0$$

The fourth symbol of TCGTGGATTTCC causes $\Pr(\texttt{TCGTGGATTTCC}|Profile)$ to equal zero. As a result, the entire string is assigned a zero probability, even though TCGTGGATTTCC differs from the consensus string at only one position. For that matter, TCGTGGATTTCC has the same low probability as AAATCTTGGAA, which is very different from the consensus string.

Laplace's Rule of Succession

Motifs

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

COUNT(*Motifs*)

A:	2	1	1	1
C:	0	1	1	1
G:	1	1	1	0
T:	1	1	1	2

PROFILE(*Motifs*)

2/4	1/4	1/4	1/4
0	1/4	1/4	1/4
1/4	1/4	1/4	0
1/4	1/4	1/4	2/4

Laplace's Rule of Succession adds 1 to each element of COUNT(*Motifs*), updating the two matrices to the following:

COUNT(*Motifs*)

A:	2+1	1+1	1+1	1+1
C:	0+1	1+1	1+1	1+1
G:	1+1	1+1	1+1	0+1
T:	1+1	1+1	1+1	2+1

PROFILE(*Motifs*)

3/8	2/8	2/8	2/8
1/8	2/8	2/8	2/8
2/8	2/8	2/8	1/8
2/8	2/8	2/8	3/8

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

tgcattcaatagctta

tgtcctgtccactcac

ctccaaatcctttaca

ggtctaccttttatcct

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

Randomized Motif Search

```
RANDOMIZEDMOTIFSEARCH(Dna, k, t)
    randomly select k-mers Motifs = (Motif_1, ..., Motif_t) in each string from Dna
    BestMotifs ← Motifs
    while forever
        Profile ← PROFILE(Motifs)
        Motifs ← MOTIFS(Profile, Dna)
        if SCORE(Motifs) < SCORE(BestMotifs)
            BestMotifs ← Motifs
        else
            return BestMotifs
```

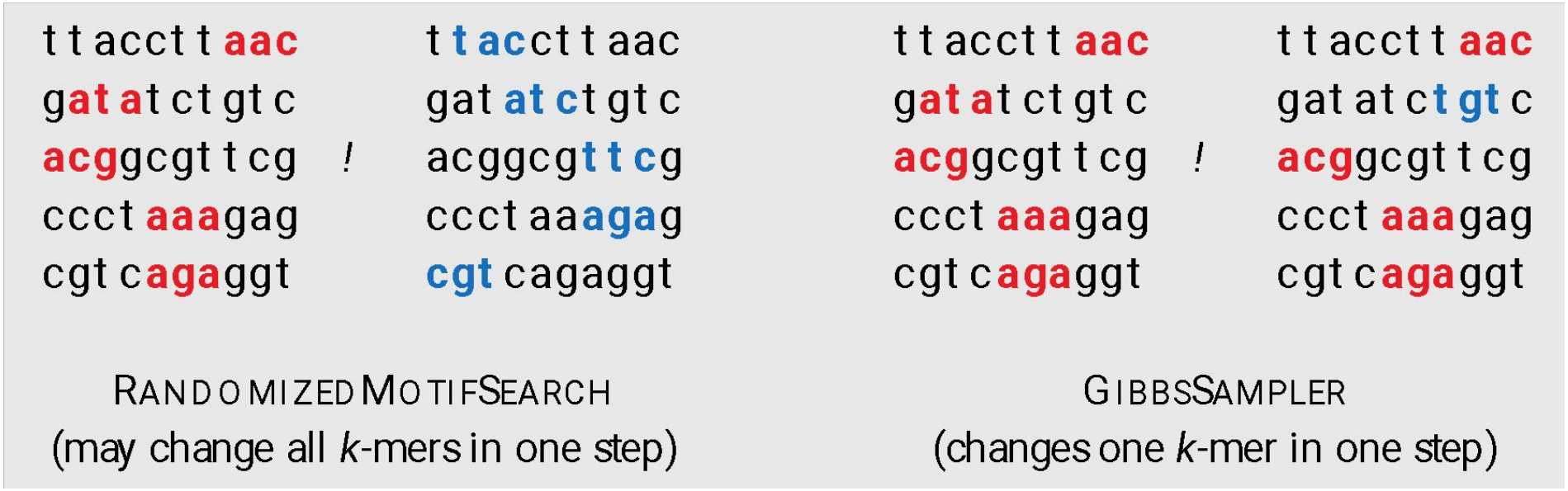
A:	0.25	0.25	0.25	0.25
C:	0.25	0.25	0.25	0.25
G:	0.25	0.25	0.25	0.25
T:	0.25	0.25	0.25	0.25

Randomized Motif Search

- 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

- 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.


t t a c c t t a a c
g a t a t c t g t c
a c g g c g t t c g
c c c t a a a g a g
c g t c a g a g g t
!
t t a c c t t a a c
g a t a t c t g t c
a c g g c g t t c g
c c c t a a a g a g
c g t c a g a g g t
RANDOMIZED MOTIF SEARCH
(may change all k-mers in one step)
t t a c c t t a a c
g a t a t c t g t c
a c g g c g t t c g
c c c t a a a g a g
c g t c a g a g g t
!
t t a c c t t a a c
g a t a t c t g t c
a c g g c g t t c g
c c c t a a a g a g
c g t c a g a g g t
GIBBS SAMPLER
(changes one k-mer in one step)

How Gibbs Sampling Works

1) Randomly choose starting positions

   $\mathbf{s} = (s_1,\ldots,s_t)$ 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 Sampler

```
GIBBSSAMPLER(Dna, k, t, N)
    randomly select k-mers Motifs = (Motif_1, ..., Motif_t) in each string from Dna
    BestMotifs ← Motifs
    for j ← 1 to N
        i ← RANDOM(t)
        Profile ← profile matrix formed from all strings in Motifs except for Motif_i
        Motif_i ← Profile-randomly generated k-mer in the i-th sequence
        if SCORE(Motifs) < SCORE(BestMotifs)
            BestMotifs ← Motifs
    return BestMotifs
```

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, $s=(s_1,s_2,s_3,s_4,s_5)$ 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 $prob(\boldsymbol{a}|\boldsymbol{P})$, and randomly select a new starting position based on this distribution.

a) To create this distribution, divide each probability $prob(\boldsymbol{a}|\boldsymbol{P})$ 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 anymore.

Gibbs Sampler in Practice

- Gibbs sampling needs to be modified when applied to samples with unequal distributions of nucleotides (*relative entropy* approach).

Dna

t **aaaa** GTCGa
ac GCTG **aaaa**
aaaa GCCTat
aCCCGa **at aa**
ag **aaaa** GGCG

- 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.

Thank you!

- Special thanks to Saifur sir