582746 Modelling and Analysis in Bioinformatics

Lecture 2: Randomized Motif Finding

Juha Kärkkäinen

13.09.2016

Outline

Outline

DNA Sequence Motif

- A DNA sequence motif is a recurring pattern in DNA presumed to have some biological funtion.
- Often the occurences of the motif in DNA are binding sites where proteins such as transcription factors can attach.

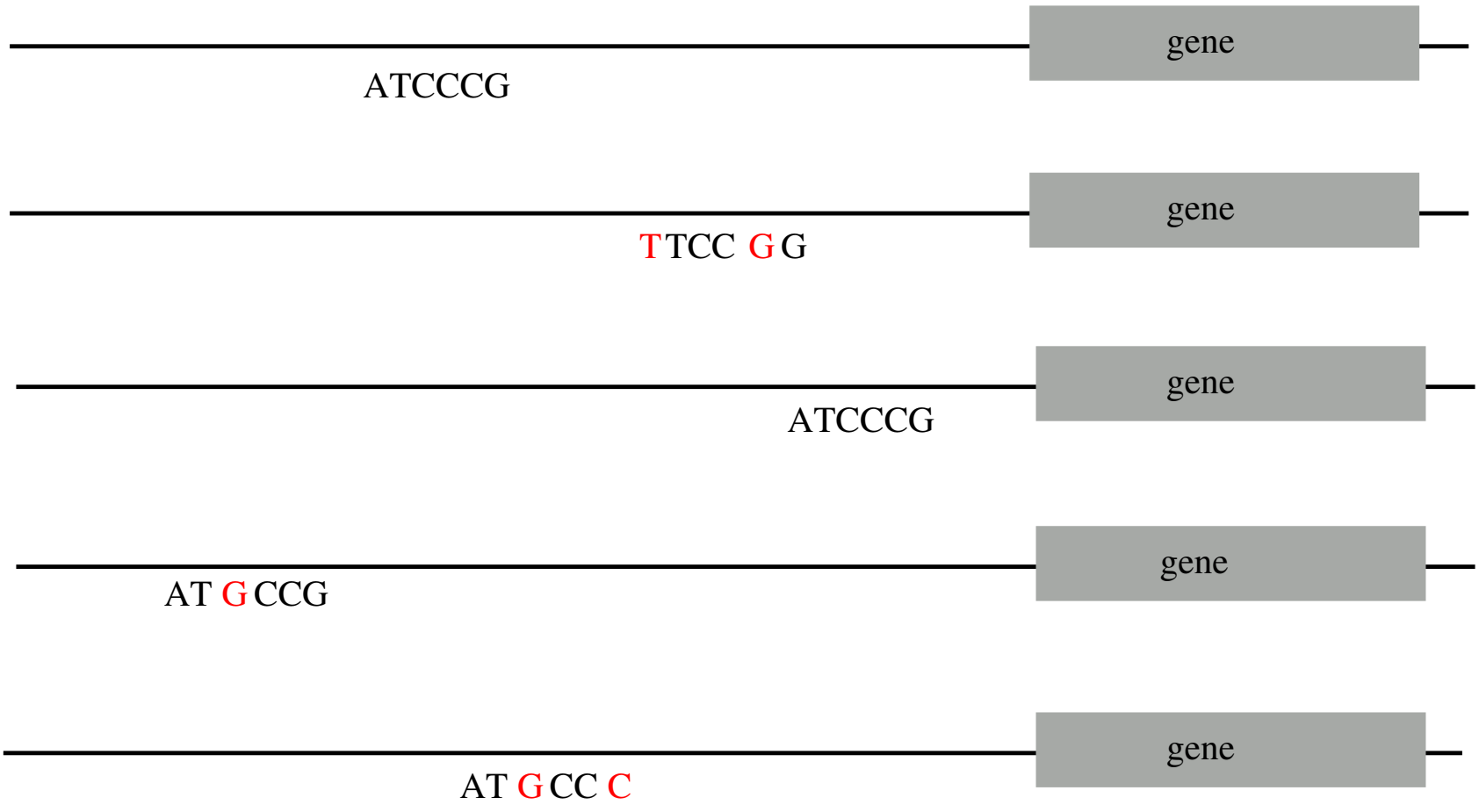
Transcription Factors

- Every gene contains a regulatory region typically stretching 100-1000 bp upstream of the transcriptional start site.
- Transcription factor (TF) is a regulatory protein that can bind to a regulatory region and promote or inhibit the transcription of the gene.
- A single TF typically regulates multiple (co-regulated) genes.

Transcription Factor Binding Sites and Motifs

- Transcription Factor Binding Site (TFBS) is the DNA location where the TF binds to.
  - May be located anywhere in the regulatory region.
- A TF binds to a specific kind of DNA sequence.
  - Similar but not identical in different co-regulated genes.
  - Some positions are more important and thus conserved while other positions allow variation.
- Motif is a pattern that describes the TFBS sequences of a given TF.

TBFSs


gene
ATCCCG
gene
TTCC G G
gene
ATCCCG
gene
AT G CCG
gene
AT G CC C

Motif Representations

- Set of sequences

TGGGGGA
TGAGAGA
TGGGGGA
TGAGAGA
TGAGGGA

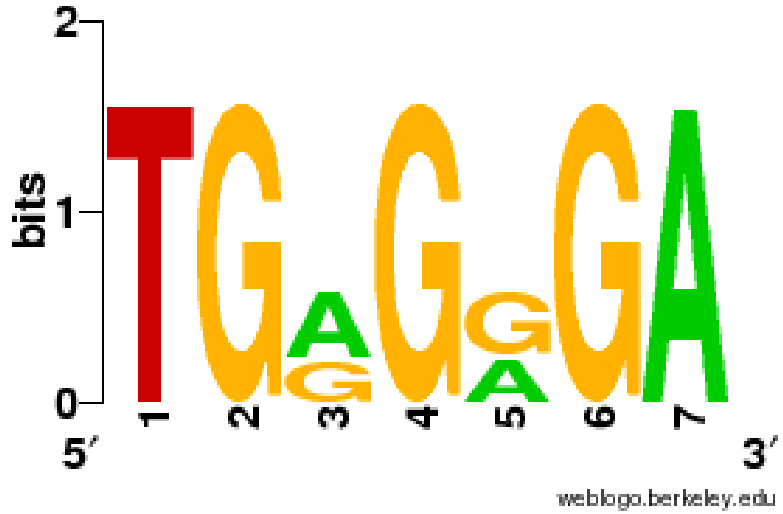
- Consensus sequence

TGAGGGA

- Profile matrix

A	0	0	3	0	2	0	5
C	0	0	0	0	0	0	0
G	0	5	2	5	3	5	0
T	5	0	0	0	0	0	0

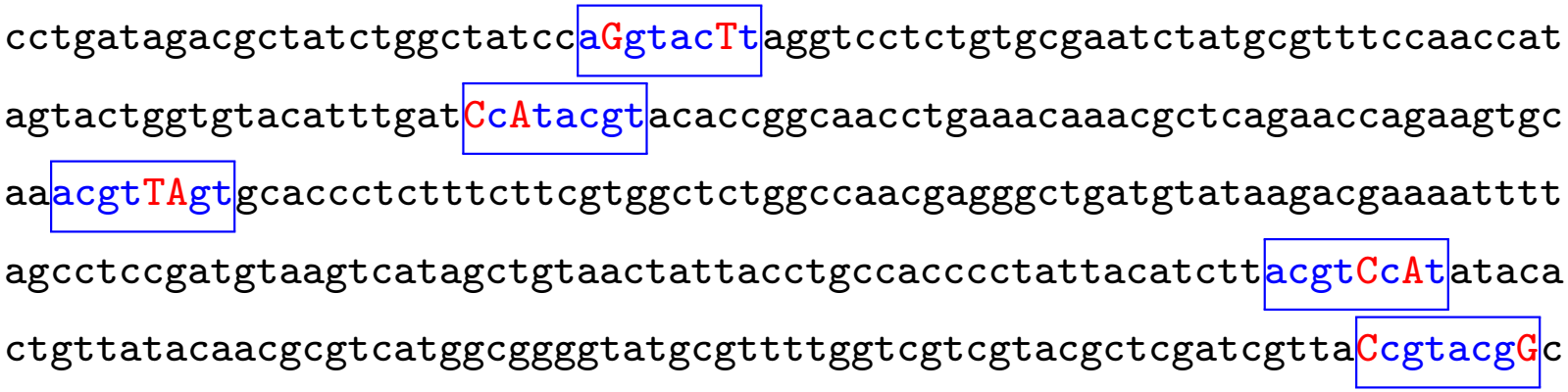
- Motif logo



Outline

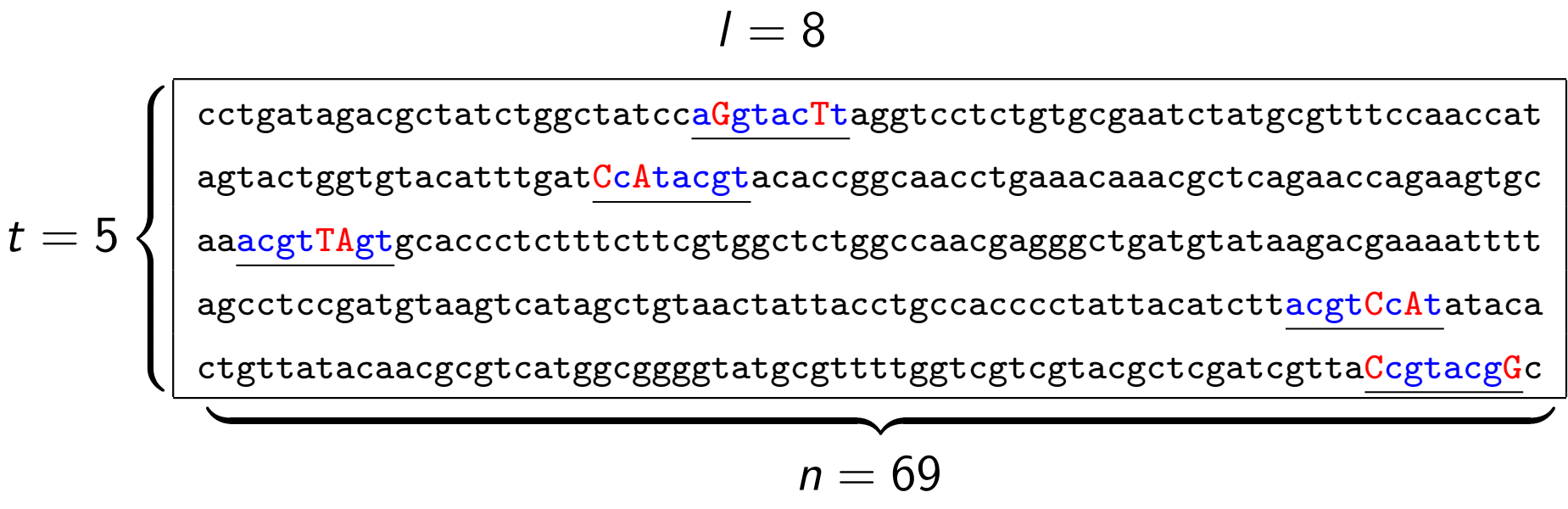
Identifying Motifs

- Identify a set of co-regulated genes.
- Extract their regulatory regions.
- Find a pattern shared by all of the regions: the motif.
- The motif can be used for finding other (previously unknown) genes regulated by the same TF.



The Motif Finding Problem: Formulation

- Goal: Find the best motif in a set of DNA sequences
- Input: A collection of t strings each of length n and an integer l
- Output: The collection of t l-mers, one from each input string, forming the best motif.
  - We will later consider what is the "best" motif.



Planted (l, d)-Motif Problem

Artificially generated problem instance

1. Generate a random l-mer M (the motif)
2. Generate t random sequences of length n
3. Implant M into a random position in each of the t sequences
4. In each implanted copy of M, make d random mutations

Example: Generate sequences

gatactgataccgtatttggcctaggcgtacacattagataaacgtatgaagtacgttagactcggcgccgcc

tttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatactgggcataaggtac

cctgggatgacttttgggaacactatagtgctctcccgatttttgaatatgtaggatcattcgccagggtccg

ttggatgaccttgtaagtgttttccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggag

gcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatggcccacttagtccacttata

tgttcttgtgaatggatttttaactgagggcatagaccgcttggcgcacccaaattcagtgtgggcgagcgca

gcccttgttagaggcccccgtactgatggaaactttcaattatgagagagctaatctatcgcgtgcgtgttca

ttggtttcgaaaatgctctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgt

ttggctaaaagcccaacttgacaaatggaagatagaatccttgcatttcaacgtatgccgaaccgaaagggaa

caacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttctgggtactgatagc

Example: Implant motif AAAAAAAAGGGGGGG

Where are the implanted motifs?

gatactgataaaaaaaaggggggggcgtacacattagataaacgtatgaagtacgttagactcggcgccgcc

tttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaataaaaaaaaggggggg

cctgggatgacttaaaaaaaaggggggggtgctctcccgattttgaatatgtaggatcattcgccagggtccg

ttggatgaaaaaaaaggggggggtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggag

gcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaataaaaaaaaggggggggcttata

tgttcttgtgaatggatttaaaaaaaaggggggggaccgcttggcgcacccaaattcagtgtgggcgagcgca

gcccttgttagaggcccccgtaaaaaaaaggggggggcaattatgagagagctaatctatcgcgtgcgtgttca

ttaaaaaaaaggggggggctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgt

ttggctaaaagcccaacttgacaaatggaagatagaatccttgcataaaaaaaaggggggggaccgaaagggaa

caacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttaaaaaaaaggggggg

Example: Add four mutations per motif occurrence

gatactgatAgAAgAAAGGttGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgcc

tttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatacAAtAAAAcGGcGGG

cctgggatgacttAAAAtAAtGGaGtGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccg

ttggatgcAAAAAAAGGGattGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggag

gcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAtAAtAAAGGaaGGGcttata

tgttcttgtgaatggatttAAcAAtAAGGGctGGgaccgcttggcgcacccaaattcagtgtgggcgagcgca

gcccttgttagaggcccccgtAtAAAcAAGGaGGGccaattatgagagagctaatctatcgcgtgcgtgttca

ttAAAAAAtAGGGaGccctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgt

ttggctaaaagcccaacttgacaaatggaagatagaatccttgcatActAAAAAGGaGcGGaccgaaagggaa

caacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttActAAAAAGGaGcGG

Where are the implanted motifs???

gatactgatagaagaaaggttgggggcgtacacattagataaacgtatgaagtacgttagactcggcgccgcc

tttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatacaataaaacggcggg

cctgggatgacttaaaataatggagtggtgctctcccgattttgaatatgtaggatcattcgccagggtccg

ttggatgcaaaaaaagggattgtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggag

gcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatataataaaggaagggcttata

tgttcttgtgaatggatttaacaataagggctgggaccgcttggcgcacccaaattcagtgtgggcgagcgca

gcccttgttagaggcccccgtataaacaaggagggccaattatgagagagctaatctatcgcgtgcgtgttca

ttaaaaaatagggagccctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgt

ttggctaaaagcccaacttgacaaatggaagatagaatccttgcatactaaaaaggagcggaccgaaagggaa

caacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttactaaaaaggagcgg

Why finding (15,4)-motifs is hard?

gatactgatAgAAgAAAGGttGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgcc

tttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatacAAtAAAAcGGcGGG

- Occurrences can differ in 8 positions

```
AgAAgAAAGGttGGG
..|..|||.|..|||
cAAtAAAAcGGcGGG
```

- Difficult to recognize similarity even if you know the positions

Planted (l, d)-Motif Problem

- Common benchmark or challenge problem for motif finding
- Adjustable difficulty
- For example, the planted $(15, 4)$-motif problem is difficult but not impossible

Real data is noisier and more complex

- No upper bound d on mutation
- Some positions have little variation, others have a lot

Outline

Scoring Motifs

- The output of motif finding is a set of l-mers, one from each input sequence
- In a desired output, the l-mers are as similar to each other as possible, representing a coherent motif
- How to measure the goodness of the motif?

Consensus Motif and Score

Motif occurrences

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

Counts

A	3	0	1	0	3	1	1	0
C	2	4	0	0	1	4	0	0
G	0	1	4	0	0	0	3	1
T	0	0	0	5	1	0	1	4

Consensus A C G T A C G T

Score 2+1+1+0+2+1+2+1=10

- t motifs occurrences, one from each sequence
- Count symbols in each column
- Consensus formed by most frequent symbols
- Score is the number of differences from consensus

Consensus Motif and Score

- Good model for planted motif problem
- In real data, mutations should be expensive in conserved positions but much cheaper in variable positions

Profile Matrix

Motif occurrences

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

Counts

A	3	0	1	0	3	1	1	0
C	2	4	0	0	1	4	0	0
G	0	1	4	0	0	0	3	1
T	0	0	0	5	1	0	1	4

Profile

A	.6	0	.2	0	.6	.2	.2	0
C	.4	.8	0	0	.2	.8	0	0
G	0	.2	.8	0	0	0	.6	.2
T	0	0	0	1	.2	0	.2	.8

Consensus

A C G T A C G T

- Profile represents the probability of each nucleotide in each position
- Profile summarizes the motif while keeping information about conserved and variable positions

Scoring with Profile

	A	.6	0	.2	0	.6	.2	.2	0
Profile P	C	.4	.8	0	0	.2	.8	0	0
	G	0	.2	.8	0	0	0	.6	.2
	T	0	0	0	1	.2	0	.2	.8

Probability or likelihood of l-mer given a profile

- $\Pr(\text{AGGTACTT} \mid P) = .6 \cdot .2 \cdot .8 \cdot 1 \cdot .6 \cdot .8 \cdot .2 \cdot .8 = 0.0073728$
- Measures how well the l-mer matches the motif

Scoring with Profile

Profile P

A	.6	0	.2	0	.6	.2	.2	0
C	.4	.8	0	0	.2	.8	0	0
G	0	.2	.8	0	0	0	.6	.2
T	0	0	0	1	.2	0	.2	.8

Probability or likelihood of l-mer given a profile

- $\Pr(\text{AGGTACTT} \mid P) = .6 \cdot .2 \cdot .8 \cdot 1 \cdot .6 \cdot .8 \cdot .2 \cdot .8 = 0.0073728$
- Measures how well the l-mer matches the motif

Likelihood ratio

- Does 0.0073728 imply a good match?
- Compare to a background model (Leccture 1)
- We assume uniform Bernoulli model:

$$\frac{\Pr(\text{AGGTACTT}) \mid P)}{\Pr(\text{AGGTACTT})} = \frac{0.0073728}{0.25^8} \approx 483$$

Problem: Zero Probabilities

	A	.6	0	.2	0	.6	.2	.2	0
Profile P	C	.4	.8	0	0	.2	.8	0	0
	G	0	.2	.8	0	0	0	.6	.2
	T	0	0	0	1	.2	0	.2	.8

Consensus	A	C	G	T	A	C	G	T

$$\Pr(\text{TCGTACGT} \mid P) = 0 \cdot .8 \cdot .8 \cdot 1 \cdot .6 \cdot .8 \cdot .6 \cdot .8 = 0$$

- Only one mismatch compared to consensus
- Should this likelyhood really be 0?

Problem: Zero Probabilities

	A	.6	0	.2	0	.6	.2	.2	0
Profile P	C	.4	.8	0	0	.2	.8	0	0
	G	0	.2	.8	0	0	0	.6	.2
	T	0	0	0	1	.2	0	.2	.8
Consensus		A	C	G	T	A	C	G	T

$$\Pr(\text{TCGTACGT} \mid P) = 0 \cdot .8 \cdot .8 \cdot 1 \cdot .6 \cdot .8 \cdot .6 \cdot .8 = 0$$

- Only one mismatch compared to consensus
- Should this likelyhood really be 0?

- The likelihood can never be zero for the l-mers from which the profile was formed
- But there can be other unknown occurrences of the motif elsewhere

Pseudocounts

- Add one to all counts
- Avoids zero counts

Count

A	3	0	1	0	3	1	1	0
C	2	4	0	0	1	4	0	0
G	0	1	4	0	0	0	3	1
T	0	0	0	5	1	0	1	4

PseudoCount

A	4	1	2	1	4	2	2	1
C	3	5	1	1	2	5	1	1
G	1	2	5	1	1	1	4	2
T	1	1	1	6	2	1	2	5

Laplace's Rule of Succession

- Use pseudocounts instead of counts to compute probabilities
- Avoids zero probabilities

PseudoCount

A	4	1	2	1	4	2	2	1
C	3	5	1	1	2	5	1	1
G	1	2	5	1	1	1	4	2
T	1	1	1	6	2	1	2	5

Profile

A	4/9	1/9	2/9	1/9	4/9	2/9	2/9	1/9
C	3/9	5/9	1/9	1/9	2/9	5/9	1/9	1/9
G	1/9	2/9	5/9	1/9	1/9	1/9	4/9	2/9
T	1/9	1/9	1/9	6/9	2/9	1/9	2/9	5/9

Motif Score

- Compute profile matrix from the motif l-mers
- Motif score is product of the l-mer likelihoods

Profile P

A	4/9	1/9	2/9	1/9	4/9	2/9	2/9	1/9
C	3/9	5/9	1/9	1/9	2/9	5/9	1/9	1/9
G	1/9	2/9	5/9	1/9	1/9	1/9	4/9	2/9
T	1/9	1/9	1/9	6/9	2/9	1/9	2/9	5/9

- $\Pr(\text{AGGTACTT} \mid P) = (4 \cdot 2 \cdot 5 \cdot 6 \cdot 4 \cdot 5 \cdot 2 \cdot 5)/9^8 = 0.00112$,
  $\Pr(\text{CCCTACGT} \mid P) = 0.00167$,
  $\Pr(\text{ACGTTAGT} \mid P) = 0.00112$,
  $\Pr(\text{ACGTCCAT} \mid P) = 0.00139$,
  $\Pr(\text{CCGTACGG} \mid P) = 0.00167$
- Motif score $\approx 4.85 \times 10^{-15}$
- (Could use logarithms to avoid very small numbers)

- Motif finding problem: Find the motif with the highest score

Outline

Motif Finding Algorithms

- Motif finding is NP-complete in general
- Algorithms that are guaranteed to find the best motif can be too slow
  - Brute force: try all possible motifs
  - Branch-and-bound: prune search space
- Use heuristics instead
  - May fail to find the best or any good motif
- Use randomization
  - Fails differently in each run
  - Repeat many times and keep the best result

Most Likely l-mer

- The l-mer in an input string with the highest profile likelihood
- Considered the best match to the motif

- Example: The most likely 8-mer in gcaaaAGGTACTTccaa is AGGTACTT
  - $\Pr(\text{AGGTACTT} \mid P) = 0.00112$

Profile P

A	4/9	1/9	2/9	1/9	4/9	2/9	2/9	1/9
C	3/9	5/9	1/9	1/9	2/9	5/9	1/9	1/9
G	1/9	2/9	5/9	1/9	1/9	1/9	4/9	2/9
T	1/9	1/9	1/9	6/9	2/9	1/9	2/9	5/9

Greedy Motif Search

1. Randomly choose t l-mers, one from each input string, as the initial motif
2. Compute the profile matrix P from the l-mers
3. Find the most likely l-mer in each input string to form a new motif
   - The new motif cannot have a lower score than the old one
4. Repeat steps 2 and 3 until the score stop improving

Greedy Motif Search: Analysis

- Local search algorithm that converges towards a local optimum
- Result depends on the initial random motif
- Good result is unlikely in a single run
- Run algorithm many times and hope that one of the runs gives a good result
- If the initial random set contains two or more l-mers from a good motif, the algorithm has a good chance of converging towards that motif

Outline

Gibbs Sampler

Randomized local search algorithm like Greedy Motif Search but with the following differences:

- In each step, only one (randomly chosen) l-mer is replaced with another l-mer in the same input sequence
- The new l-mer is chosen randomly using the likelihoods as weights (not as probabilities as they do not sum up to one)

Gibbs Sampler

1. Randomly choose t l-mers, one from each input string, as the initial motif
2. Randomly choose one, say ith, of the t input string and remove the corresponding l-mer from the motif
3. Compute the profile matrix P from the remaining l-mers
4. Randomly choose an l-mer in the ith input string using the likelihoods as weights and add the chosen l-mer to the motif
5. Repeat steps 2–4 until the best motif found has not improved for a while

Gibbs Sampler: Analysis

- If at some point the set contains two or more l-mers from a good motif, the algorithm has a good chance of converging towards that motif
- Otherwise it can escape (weak) local optima because the new l-mers are chosen randomly
- Can get stuck to a local optimum (for a long time)
- Better to restart fresh every now and then
- Even better: Determine a good initial motif by other means

Outline

Random Projection

- Typically a motif has several highly conserved positions
- If we only look at (a subset) of highly conserved positions, most motif occurrences are identical.
- We don't know the highly conserved positions but we can try many random subsets of positions.

a	g	g	T	a	c	t	T
c	C	a	T	a	c	g	T
a	C	g	T	t	a	g	T
a	C	g	T	c	c	a	T
c	C	g	T	a	c	g	g
	2		4				8

Projection

- Defined by a choice of k out of l positions
- Maps an l-mer into a k-mer by extracting the bases at the chosen positions
- Projection from an l-dimensional space to a k-dimensional space
- Projection is a kind of hash function
- Example: projection = (2,4,8)

$$\begin{array}{cccccccc} c & C & a & T & a & c & g & T \\ & 2 & & 4 & & & & 8 \end{array} \quad \Longrightarrow \quad C\ T\ T$$

Random Projection Algorithm

1. Randomly choose k out of ℓ positions
2. Distribute all l-mers in all input strings into buckets by their projected k-mer
3. If there is a bucket with enough l-mers, form a motif from those l-mers
4. Refine the motif if any
5. Repeat step 1–4 many times

Random Projection: Analysis

- Most l-mers are about evenly disributed over the buckets
- If the chosen position are highly conserved, most/many true motif occurrences map into the same bucket(s)
- Then, the motif formed from the l-mers in the bucket is similar to the true motif
- Requires appropriate choice of k and the threshold for large bucket (study group)

Motif Refinement

- Motif is formed from l-mers in a projection bucket
- Compute the profile for the motif
- Run a local search algorithm, such as Gibbs Sampler, initialized using the profile
  - For example, choose the initial l-mers in Gibbs Sampler using the profile-likelihoods

Gibbs Sampler and Random Projection

- Gibbs Sampler can use Random Projection for generating better initial motifs
- Random Projection can use Gibbs Sampler for refining motifs