



Introduction to Bioinformatics

De novo sequence motif discovery

Lecturer: Jan Baumbach
Teaching assistant(s): Diogo Marinho

Lecture Overview

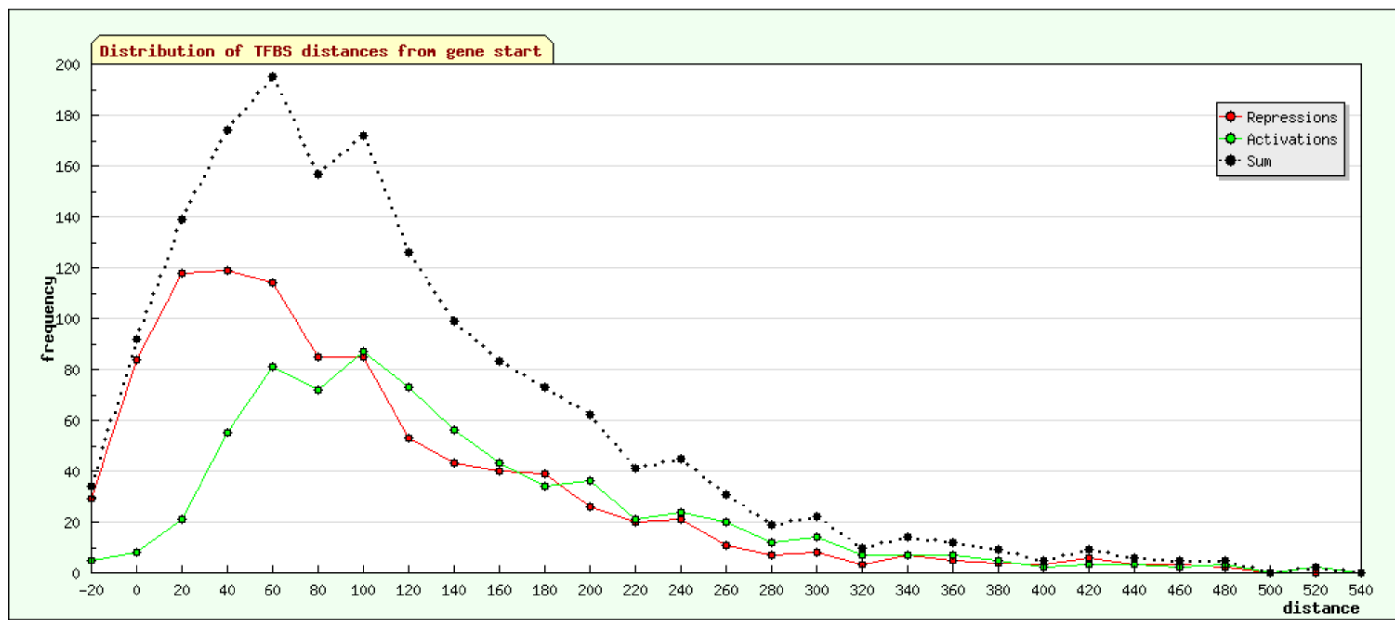
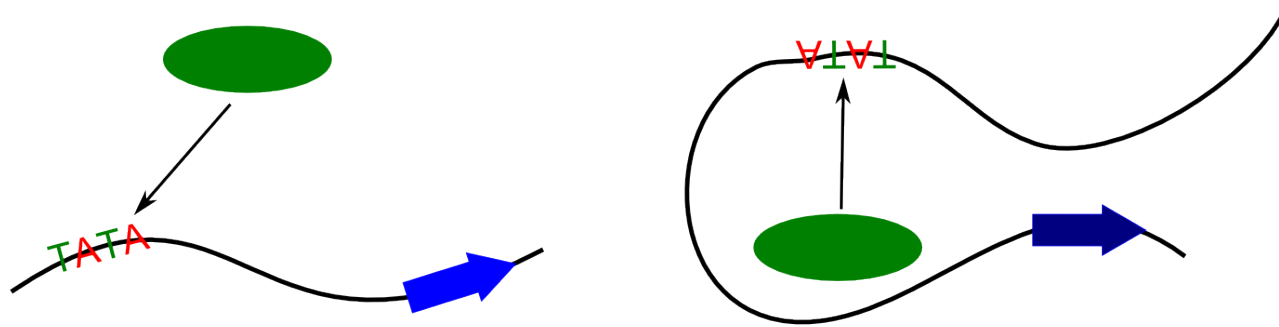
- ▶ What is all this good for?
- ▶ Exact Problem Statement & Definitions
- ▶ Methods
 - Simple Bruteforce Approach
 - Expectation Maximization
 - MEME
 - Gibbs Sampling
- ▶ Wrap Up

What is all this good for?

We start with a transcription factor

- ▶ Knock out a specific Transcription Factor
- ▶ See co-expressed genes
- ▶ They are most likely influenced by that specific transcription factor
- ▶ The sequence upstream of them are likely to contain the binding sites?
- ▶ Given: A set of long sequences likely to contain short, conserved sequence motif.
- ▶ Task: Find the most prominent motif.

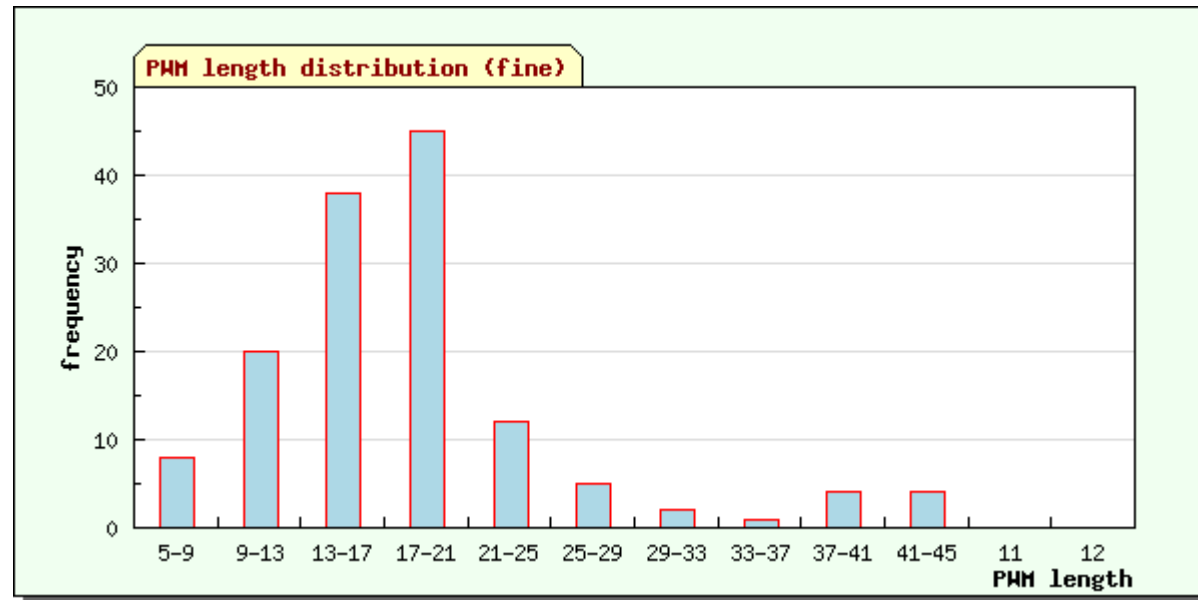
The position of binding sites can vary ...



Statistics taken from <http://www.coryneregnet.de/>

Common features of binding sites

- ▶ Located upstream of the gene
- ▶ Typically very short (5-15bp, sometimes up to 30)



Statistics taken from <http://www.coryneregnet.de/>

Wet-lab is not exact enough

- ▶ Experiments to verify binding sites are expensive and time-consuming
 - Mutation of the binding sites
 - Functional changes
- ▶ Furthermore, discovering the binding site accurate down to single-nucleotide resolution is hard, if not even impossible.

We need computational methods!

The starting point

- ▶ We already know the approximate position
- ▶ We assume that the binding sites are evolutionary conserved, i.e. they are quite similar

So, why don't we just look for the longest common nucleotide string?

Several issues make the problem hard!

- ▶ Not all co-regulated genes possess the same binding site. Hence, the motif might be different.
- ▶ Motifs can be slightly different.
- ▶ We are looking for very short motifs in quite large regions of a small set of DNA sequences.
- ▶ Bad signal (FG) to noise (BG) ratio.
 - But later more on that...

Exact Problem Statement & Definitions

Definitions!

Motif

We call a reoccurring nucleotide sequence a (l, d) -Motif if the sequence is of length l and has at most d mutations.

Planted Motif Problem

Given: A set S of sequences of length n and two ints l, d with $d < l < n$.

Wanted: A pattern M , such in each Sequence there is a substring which can be transformed to M with at most d substitutions.

Example

A**TCCTAGGCAGT**CCTGCATTGTCTTGCGGAATTGTA
TCCAGTCTATTTCAAACGG**TCCTAGT****CAGT**GCTCTG
TTGAATTTTCCTGTGTGGCTTCCTGTATAGAGGATA
GGTTA**TCCTAGGCAGT**ACAGCCCACCCCAGAGTCGG
CGGGTGCACCTCCAA**TCCA****AGGCAGT**GCGGGGCGGAAC
GTT**TCCTAGGCA****T**GGGTCAACGTTCCACACAGAACG

Example for the (11,1)-Motif **TCCTAGGCAGT**

Different problem formulations

- ▶ Find exactly one/several motifs
- ▶ In every upstream sequence, the motif needs to occur at least once, exactly once, or at most once in “most” of the sequences
- ▶ Motifs with/without gaps

Methods

Different approaches exist

- ▶ Bruteforce
- ▶ Statistical Approaches
 - MEME
 - Gibbs
- ▶ Combinatorial Approaches
 - SP-Star
 - Random projection

Bruteforce

Bruteforce Approach (1/2)...

- ▶ We have a set $S = S_1, \dots, S_m$ of sequences
- ▶ Let $d(S_i, M)$ the minimal edit distance between sequence S_i and a pattern M .
- ▶ We define $Score(M) = \sum_{i=1}^m d(S_i, M)$ and try to find a Motif of a given length l with minimal score.

Bruteforce Approach (2/2)...

```
set M_opt = AA...A
for every pattern M from AA...A to TT...T do
    for i = 1 to m do
        Compute d(S_i,M) .
        if d(S_i,M) > d, then try the next M
    compute score(M)
    if score(M) < score(M_opt) then set M_opt = M
return M_opt
```

... is bad!

- ▶ Calculating $d(S_i, M)$ for an (l, d) -Motif in an sequence of length n takes $O(nd)$
- ▶ We have m Sequences, so calculating $Score(M)$ takes $O(mnd)$
- ▶ In total we have 4^l possible motifs of length l over an alphabet Σ with $|\Sigma| = 4$.
- ▶ Total runtime: **$O(4^l mnd)$**

We need smarter approaches!

Expectation Maximization

Never expect too much ...

Coin Flip Experiment

- ▶ 2 Coins, A and B, biased (θ_A, θ_B)
- ▶ θ_A, θ_B denote the likeliness of seeing “head”
- ▶ 5 rounds of coin flips with either coin A or coin B
- ▶ 10 flips per round

	Coin A	Coin B
B HTTTHHTHTH		5H, 5T
A HHHHTHHHHH	9H, 1T	
A HTHHHHHHTHH	8H, 2T	
B HTHTTTHTTT		4H, 6T
A THHHTHHHTH	7H, 3T	

This and the following examples are taken from Chuong B Do & Serafim Batzoglou, “*What is the expectation maximization algorithm?*”, Nature Biotechnology, Vol. 26 No. 8, 2008

MLE, complete Information

- Now, we can apply a Maximum Likelihood Estimator (MLE).
With complete information, it is very easy:

	Coin A	Coin B
B HTTTHHTHTH		5H, 5T
A HHHHTHHHHH	9H, 1T	
A HTHHHHHTHH	8H, 2T	
B HTHTTTHTTT		4H, 6T
A THHHHTHHHTH	7H, 3T	

$$\hat{\theta}_A = \frac{24}{24 + 6} = 0.8$$

$$\hat{\theta}_B = \frac{9}{9 + 11} = 0.45$$

OK, that was easy ... but

- ▶ Now, let's assume we have incomplete information:
 - We still know that we have two biased coins
 - But we do not longer know, which coin produced which sample

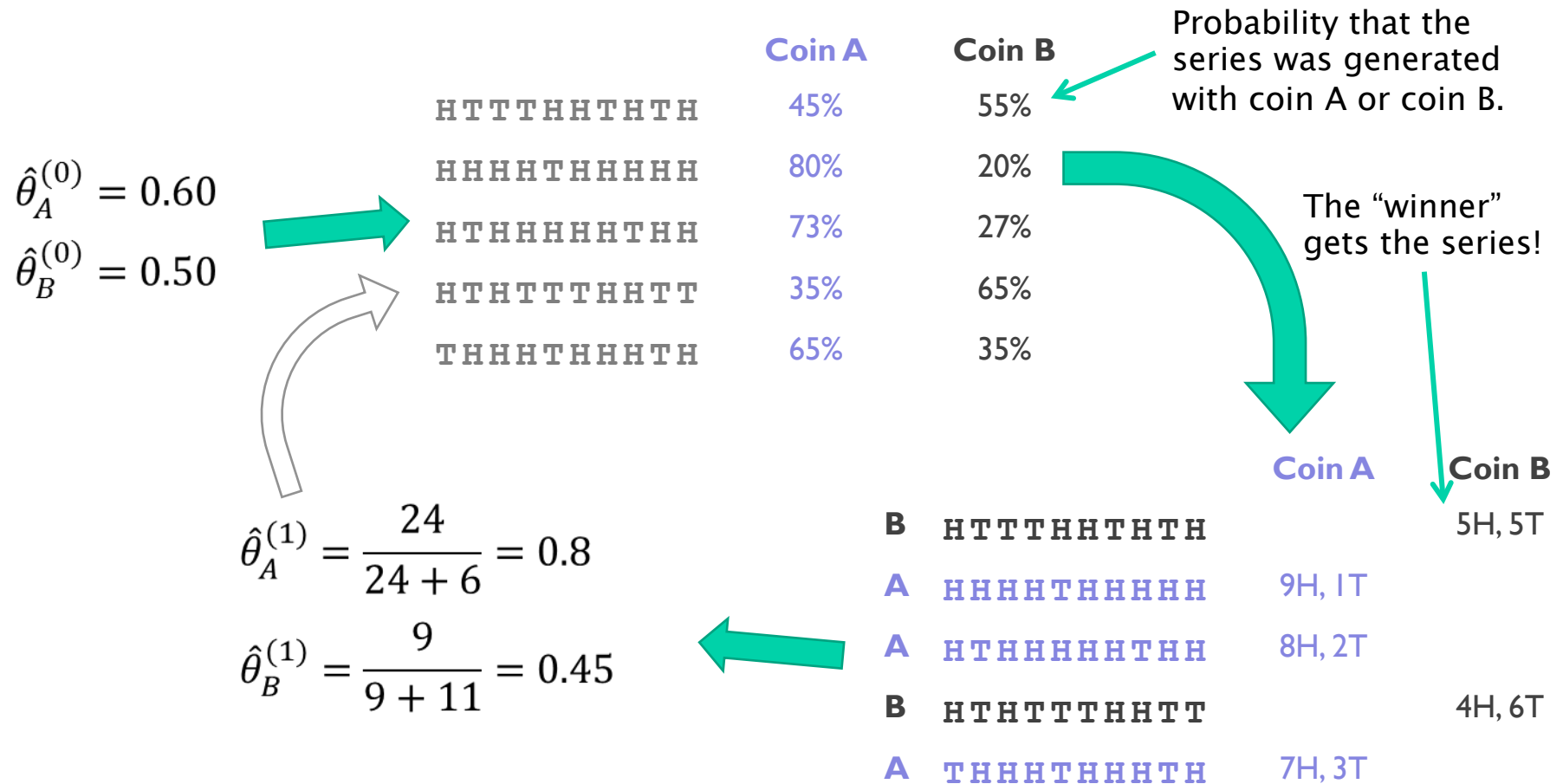
		Coin A	Coin B
?	HTTTHHTHTH	?	?
?	HHHHTHHHHH	?	?
?	HTHHHHHTHH	?	?
?	HTHTTTTHHTT	?	?
?	THHHTHHHTH	?	?

MLE, Incomplete Information

► Iterative process:

- Guess initial parameters $\theta_A^{(0)}$ and $\theta_B^{(0)}$
- Calculate which coin is the more likely one for what round (complete the information)
- Use that for calculating $\theta_A^{(0)}$ and $\theta_B^{(0)}$ using the same MLE as before
- Repeat until convergence

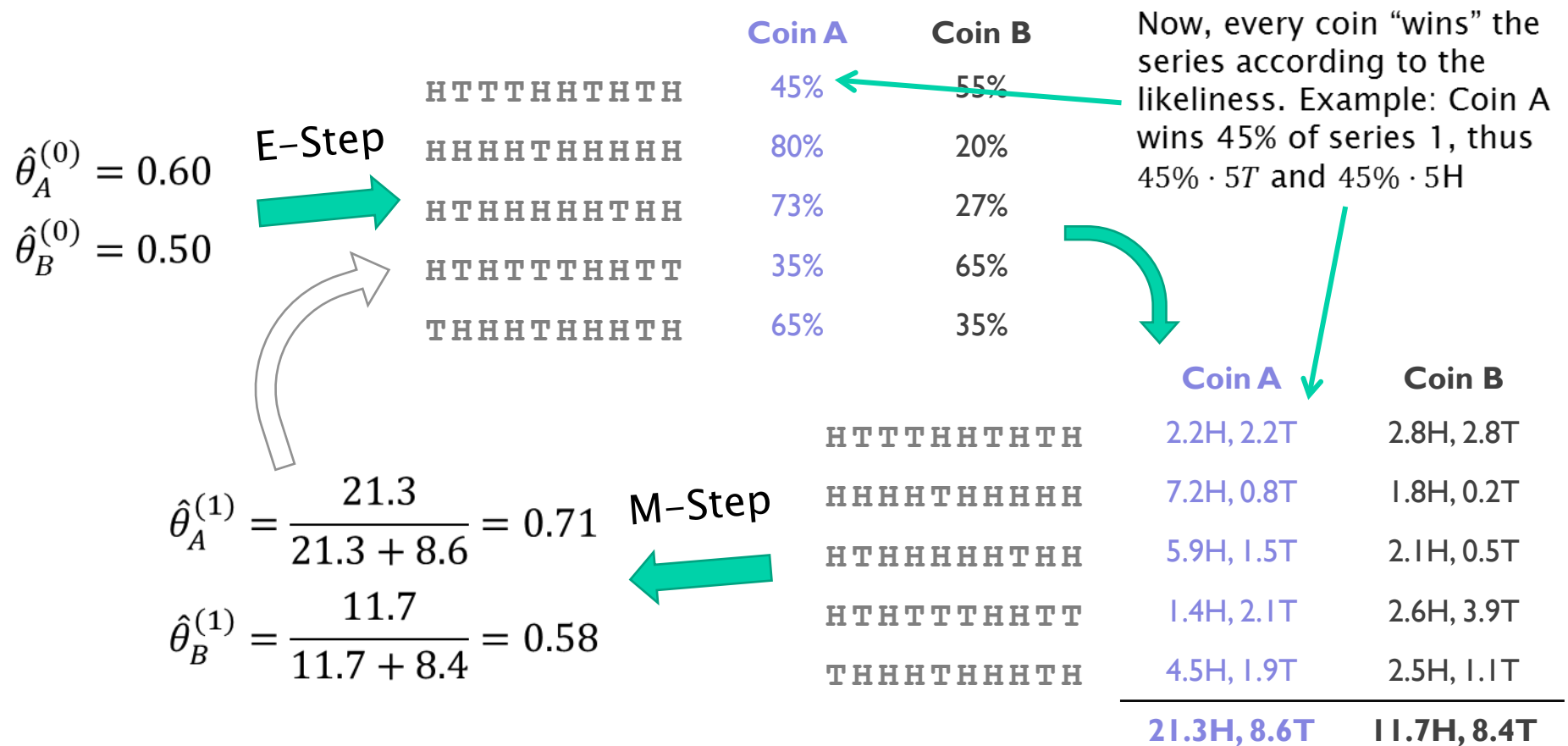
MLE, Incomplete Information



Expectation Maximization

- ▶ Refinement of this general idea
- ▶ Now, we stop using either coin A or coin B, but use probabilities for data completion (In our example the assignment of the coin flip series to the coins)
- ▶ Then we use a slightly adapted MLE for facilitating these probabilities.

Expectation Maximization



How to use EM for Motif discovery

- ▶ Assumptions for now:
 - The motifs have fixed length W
 - The Motif is generated by a probability function
 - The background is generated by a different probability function
- ▶ We have several unknown parameters
 - The Motif itself, so we have to find the different parameters of the probability function
 - The position of the Motif within the different sequences

The EM Algorithm

- ▶ Start with an initial guess of the Motif positions $Z^{(0)}$
- ▶ With that guess, calculate the first Motif representation $\theta^{(0)}$
- ▶ Use the motif model to update the position likelihoods ($Z^{(n+1)}$) (E-Step)
- ▶ Use these likelihoods to update the motif model ($\theta^{(n+1)}$) (M-Step)
- ▶ Iterate until convergence


Representation of a Motif

- ▶ Represented by a probability matrix $\theta = p_{c,k}$ with $p_{c,k}$ being the probability of character c at position k . (Position Weight Matrix)
- ▶ Same for the background

		0	1	2	3
$\theta =$	A	0.25	0.2	0.3	0.1
	C	0.27	0.3	0.5	0.2
	G	0.23	0.1	0.1	0.4
	T	0.25	0.4	0.1	0.3
		Background		Motif positions	

Likelihood of a Starting Position

- ▶ Represented by a matrix $Z = z_{i,j}$ with $z_{i,j}$ being the probability that the motif (given by θ) starts at position j in sequence i .



$$P(X_i | Z_{i,j} = 1, \theta) = \underbrace{\prod_{k=1}^{j-1} p_{c_{i,k},0}}_{\text{Before Motif}} \underbrace{\prod_{k=j}^{j+W-1} p_{c_{i,k},k-j+1}}_{\text{Motif}} \underbrace{\prod_{k=j+W}^L p_{c_{i,k},0}}_{\text{After Motif}}$$

X_i – i th sequence
 $Z_{i,j}$ – is 1 if motif starts at position j in sequence i .
 $c_{i,k}$ – is the character at position k in sequence i .

Can be read as the probability to find the Motif in sequence X_i at position j (the condition $Z_{i,j} = 1$) for the given Motif model θ .

Calculating the Z Matrix

- ▶ Basically calculate all likelihoods and scale them to sum up to 1 for every sequence

$$Z_{i,j}^{(t)} = \frac{P(X_i | Z_{i,j} = 1, \theta^{(t-1)}) P(Z_{i,j} = 1)}{\sum_{k=1}^{L-W+1} \underbrace{P(X_i | Z_{i,k} = 1, \theta^{(t-1)}) P(Z_{i,k} = 1)}}_{\text{Can be left out, if every starting position is equally likely!}}$$

Can be left out, if every starting position is equally likely!

Updating the Model

- ▶ Count and weight the amount of chars in the Motif

$$p_{c,k}^{(t)} = \frac{n_{c,k}}{\sum_b n_{b,k}}$$

$$n_{c,k} = \begin{cases} \sum_i \sum_{\{j | X_{i,j+k-1}=c\}} Z_{i,j}^{(t)} & k > 0 \\ n_c - \sum_{j=1}^W n_{c,j} & k = 0 \end{cases}$$

Updating the Model

- ▶ Count and weight the amount of chars in the Motif

Probability of char c at position k in the motif.

$$p_{c,k}^{(t)} = \frac{n_{c,k}}{\sum_b n_{b,k}}$$

At the moment, this is a overkill. Right now it is just the number of sequences

Sum over all motifs.

$$n_{c,k} = \begin{cases} \sum_i \sum_{\{j | X_{i,j+k-1}=c\}} Z_{i,j}^{(t)} & k > 0 \\ n_c - \sum_{j=1}^W n_{c,j} & k = 0 \end{cases}$$

Take those starting points, where char c appears at position k

Total number of char c in the entire dataset.

$k = 0$ means the background model.

Example ... enjoy the show!

- ▶ Guess initial starting position

TATAGACG
GGTATACC
CGCCTATA

Z

	P1	P2	P3	P4	P5
Seq1	0	0	0	0	1
Seq2	0	0	0	0	1
Seq3	0	0	0	0	1

Example ... enjoy the show!

- ▶ Leads to the model

TATAGACG
GGTATACC
CGCCTATA

θ	Backg.	P1	P2	P3	P4
A	0.25	0	1	0	0.33
C	0.25	0	0	0.67	0.33
G	0.25	0.33	0	0	0.33
T	0.25	0.67	0	0.33	0

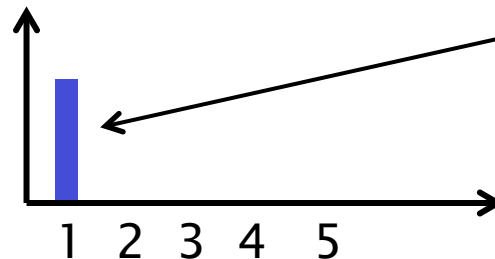
G T G
T A C A

Example ... enjoy the show!

- ▶ Likelihood for the model occurring at the first position of sequence 1

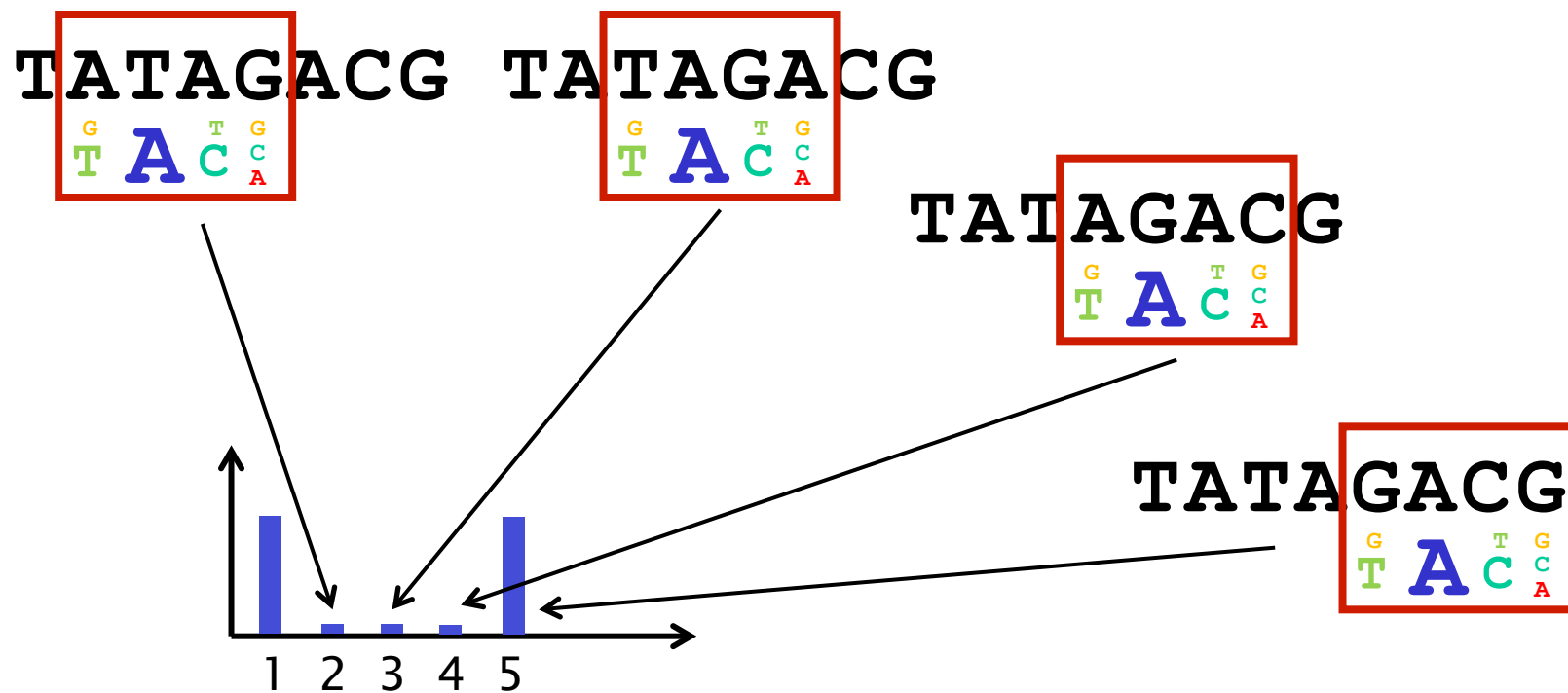
TATAGACG
G T G
T A C A

$$P(seq1|Z_{1,1} = 1, \theta) = \underbrace{p_{T,1} * p_{A,2} * p_{T,3} * p_{A,4}}_{\text{Motif}} * \underbrace{p_{G,0} * p_{A,0} * p_{C,0} * p_{G,0}}_{\text{After Motif}} = 2.89E^{-4}$$



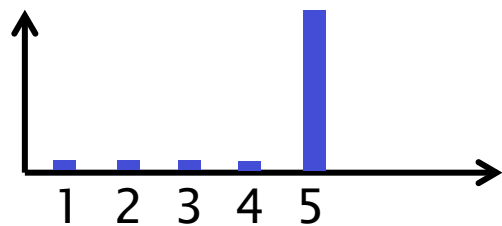
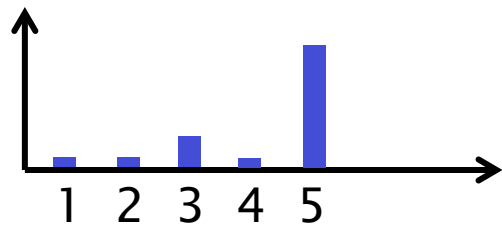
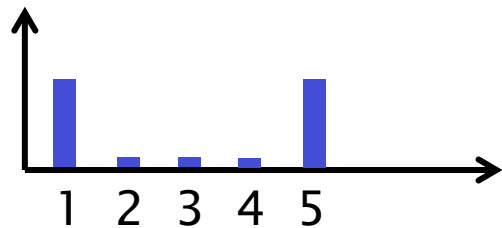
Example ... enjoy the show!

- ▶ The same with the remaining positions



Example ... enjoy the show!

- ▶ And the other sequences, normalize all to 1

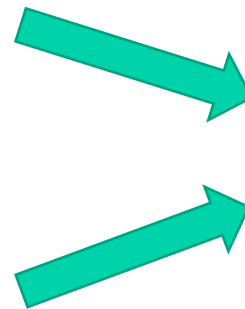


<i>Z</i>	P1	P2	P3	P4	P5
Seq1	0.5	0	0	0	0.5
Seq2	0	0	0.33	0	0.67
Seq3	0	0	0	0	1

Example ... enjoy the show!

- Refining the model

TATAGACG
GGTATACC
CGCCTATA



0.5 TATA
0.5 GACG
0.33 TATA
0.67 TACC
1 TACC

Z	P1	P2	P3	P4	P5
Seq1	0.5	0	0	0	0.5
Seq2	0	0	0.33	0	0.67
Seq3	0	0	0	0	1

Example ... enjoy the show!

- ▶ Calculate the new model

0.5	TATA	→	θ	Backg.	P1	P2	P3	P4
0.5	GACG							
0.33	TATA							
0.67	TACC							
1	TACC							
3								

Example for Position 1:

$$\frac{0.5T + 0.33T + 0.67T + 1T}{3} = 0.83T$$

$$\frac{0.5G}{3} = 0.17G$$

Example ... enjoy the show!

- ▶ Repeat until convergence

TATA

TATAGACG
GGTATAACC
CGCCTATA

Problems with EM

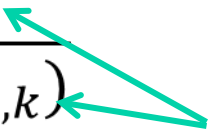
- ▶ If/once the letter probability of a certain letter equals 0, we will never find a motif with that letter at that position
 - Consequently, if the motif is not a combination of the initial guesses, EM would fail.
- ▶ As EM is a gradient decent method, we can (and will) be trapped in local maxima.
- ▶ There has to be exactly one motif in every sequence.
- ▶ We can't find more than one motif.
- ▶ We can't include previous knowledge.

Extensions for MEME (1/3)

- ▶ Inclusion of a pseudo count to avoid zero probabilities in the motif model

$$p_{c,k}^{(t)} = \frac{n_{c,k} + d_{c,k}}{\sum_b (n_{b,k} + d_{b,k})}$$

Pseudo-counts



$$n_{c,k} = \begin{cases} \sum_i \sum_{\{j | X_{i,j+k-1}=c\}} Z_{i,j}^{(t)} & k > 0 \\ n_c - \sum_{j=1}^W n_{c,j} & k = 0 \end{cases}$$

Extensions for MEME (2/3)

- ▶ Looking for a good starting point
 - To avoid being trapped in local maxima, the starting motif has to be as similar as possible to the actual motif.
 - For every distinct subsequence of length W
 - Calculate the motif model
 - Run just one EM step
 - Now, choose the model with the highest likelihood after that one EM step as start model

Extensions for MEME (3/3)

- ▶ Facilitate prior knowledge from deriving a good starting point
- ▶ For example, we know that binding site are T and A rich regions
- ▶ Then use a subsequence like 'TATA' and set the probability to some π . For example to 0.7

θ	Backg.	P1	P2	P3	P4
A	?	0.1	0.7	0.1	0.7
C	?	0.1	0.1	0.1	0.1
G	?	0.1	0.1	0.1	0.1
T	?	0.7	0.1	0.7	0.1

Allow Zero Occurrence in some Motifs

- ▶ At the moment, the Motif has to be in every Sequence!
- ▶ If we want to ease that condition, we have to introduce a prior:
 - λ prior probability that any position in a sequence is the start of a motif.
 - $\gamma = (L - W + 1)\lambda$ prior probability of a sequence containing a motif. ($L - W + 1$ is the number of possible starting positions of a motif of Length W within a sequence of length L)

Modified E-Step

$$Z_{i,j}^{(t)} = \frac{P(X_i | Z_{i,j} = 1, \theta^{(t-1)}) \lambda^{(t-1)}}{\underbrace{P(X_i | Q_i = 0, \theta^{(t-1)}) (1 - \gamma^{(t-1)})}_{\text{Likelihood of seeing the sequence } X_i \text{ and not containing a motif.}} + \sum_{k=1}^{L-W+1} \underbrace{P(X_i | Z_{i,k} = 1, \theta^{(t-1)}) \lambda^{(t-1)}}_{\text{Probability that a motif starts at that position.}}}$$

That denotes the likelihood of seeing the sequence X_i and not containing a motif. Q_i is a random variable which is 1 if sequence X_i contains a motif and 0 otherwise.

Probability that a motif starts at that position.

$$P(Q_i = 1) = \sum_{j=1}^{L-W+1} Z_{i,j}^{(t-1)}$$

$$P(X_i | Q_i = 0, \theta^{(t-1)}) = \prod_{j=1}^L \theta_{c_j, 0}^{(t-1)}$$

Update M-Step

- ▶ Updating the model $\theta^{(t)}$ remains the same!
- ▶ But we have to update $\lambda^{(t)}$ and $\gamma^{(t)}$:

Sum over all sequences

$$\gamma^{(t)} = (L - W + 1)\lambda^{(t)} = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^{L-W+1} Z_{i,j}^{(t)}$$

Sum over the likelihood for all possible starting positions for sequence i .

Gibbs Sampling

- ▶ EM and MEME can be trapped in local minima.
- ▶ This can be reduced by trying different starting positions
- ▶ Gibbs sampling makes greater use of random search
- ▶ Can be seen as a stochastic analog of EM
- ▶ Main difference:
 - In EM we kept a distribution for the starting points (Z_i)
 - In Gibbs, we assign a specific starting point a_i for each sequence X_i , but we will randomly resample these.

Algorithm

1. Choose initial guesses of motif locations.
2. Derive the model from these locations
3. Choose a random sequence. Calculate the likelihood of all possible motif starting positions
4. Randomly choose a motif position from the probability distribution
5. Repeat 2-4 until convergence.

Summary

What Else is Out There?

- ▶ We learned a probabilistic approach to *de novo* motif discovery.
- ▶ There are also combinatorial methods to solve that problem (as a discrete optimization problem)
 - Sometimes outperform probabilistic approaches
 - But they do not provide confidence measures
- ▶ Related problems:
 - Motif finding (if the motif is already known)
 - Motif adjustment

Overview of the Different Tools

Program	Method
AlignACE	Gibbs sampling
Consensus	Models motifs using weight matrices, searching for the matrix with maximum information content
The Improbizer	Uses EM to determine weight matrices of DNA motifs that occur more often than random in the input set.
MEME	See this lecture 😊
MotifSampler	Matrix-based, motif-finding algorithm that extends Gibbs sampling by modeling the background with a higher order Markov model

Tompa et al. *Assessing computational tools for the discovery of transcription factor binding sites*, Nature Biotechnology, vol 23 No 1, 137--144, 2005

Problem solved?

- ▶ Huge performance study:
 - sSn at most 0.22 (site sensitivity)
 - nCC at most 0.20 (nucleotide level correlation coefficient)
- ▶ Very bad signal to noise ratio
- ▶ Underlying molecular biology not fully understood
- ▶ How to really assess the correctness of such a tool? (Just because a tool returns a motif we have never seen before ... it is not necessarily wrong!)

Correlation coefficient (nCC) for all pairs of tools

	Quick score	GLAM	SeSi MCMC	MITRA	Consen	Improb	Align ACE	Motif sampler	MEME3	MEME	Oligo/dyad	ANN-Spec	YMF	Weeder
QuickScore	0.009	0.020	0.042	0.030	0.025	0.052	0.068	0.072	0.072	0.074	0.038	0.064	0.061	0.084
GLAM	0.031	0.016	0.060	0.037	0.039	0.068	0.066	0.084	0.088	0.086	0.052	0.082	0.090	0.113
SeSiMCMC	0.049	0.059	0.024	0.068	0.042	0.083	0.071	0.091	0.081	0.088	0.058	0.103	0.104	0.092
MITRA	0.042	0.041	0.072	0.031	0.054	0.082	0.084	0.097	0.106	0.105	0.070	0.101	0.103	0.131
Consensus	0.067	0.060	0.075	0.053	0.042	0.077	0.079	0.109	0.084	0.077	0.074	0.082	0.081	0.098
Improbizer	0.065	0.069	0.083	0.077	0.056	0.052	0.089	0.117	0.096	0.098	0.083	0.112	0.091	0.117
AlignACE	0.088	0.084	0.089	0.090	0.085	0.111	0.068	0.097	0.102	0.091	0.088	0.091	0.115	0.119
MotifSampler	0.071	0.092	0.107	0.097	0.077	0.103	0.099	0.068	0.112	0.119	0.103	0.127	0.130	0.134
MEME3	0.089	0.094	0.092	0.102	0.074	0.102	0.093	0.124	0.069	0.106	0.094	0.129	0.126	0.114
MEME	0.091	0.090	0.100	0.102	0.077	0.091	0.095	0.120	0.100	0.073	0.104	0.123	0.121	0.121
Oligo/dyad	0.073	0.088	0.111	0.088	0.082	0.082	0.099	0.136	0.119	0.112	0.071	0.106	0.107	0.130
ANN-Spec	0.085	0.091	0.111	0.094	0.090	0.100	0.085	0.122	0.114	0.110	0.089	0.074	0.118	0.117
YMF	0.094	0.095	0.112	0.101	0.093	0.100	0.114	0.146	0.121	0.129	0.092	0.131	0.084	0.137
Weeder	0.164	0.169	0.162	0.167	0.157	0.171	0.166	0.186	0.168	0.164	0.173	0.167	0.167	0.156

^aThe primary tool is listed in the row header and the secondary tool in the column header. The score shown for the same tool on both axes (that is, along the main diagonal) is the individual *nCC* score from **Figure 1**. Numerical values are categorized by color, ranging from dark blue (poorer predictions) to red (better predictions).

Tompa et al. *Assessing computational tools for the discovery of transcription factor binding sites*, Nature Biotechnology, vol 23 No 1, 137--144, 2005

Literature

- ▶ Lawrence et al., ***Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment***, Science, Vol. 262 No. 5131, 1993
- ▶ Bailey et al., ***MEME: discovering and analyzing DNA and protein sequence motifs***, Nucleic Acids Res 34 (Web Server issue):W369-373, 2006.
- ▶ Tompa et al., ***Assessing computational tools for the discovery of transcription factor binding sites***, Nature Biotechnology, Vol. 23 No. 1, 137-144, 2005
- ▶ Chuong et al., ***What is the expectation maximization algorithm?***, Nature Biotechnology, Vol. 26 No. 8, 2008
- ▶ Mark Craven's lecture BMI/CS 776 Advanced Bioinformatics.
- ▶ Irene Liu's lecture on Motif Finding
- ▶ And of course: www.bing.com (Use that to find "The Google")

Thank you!