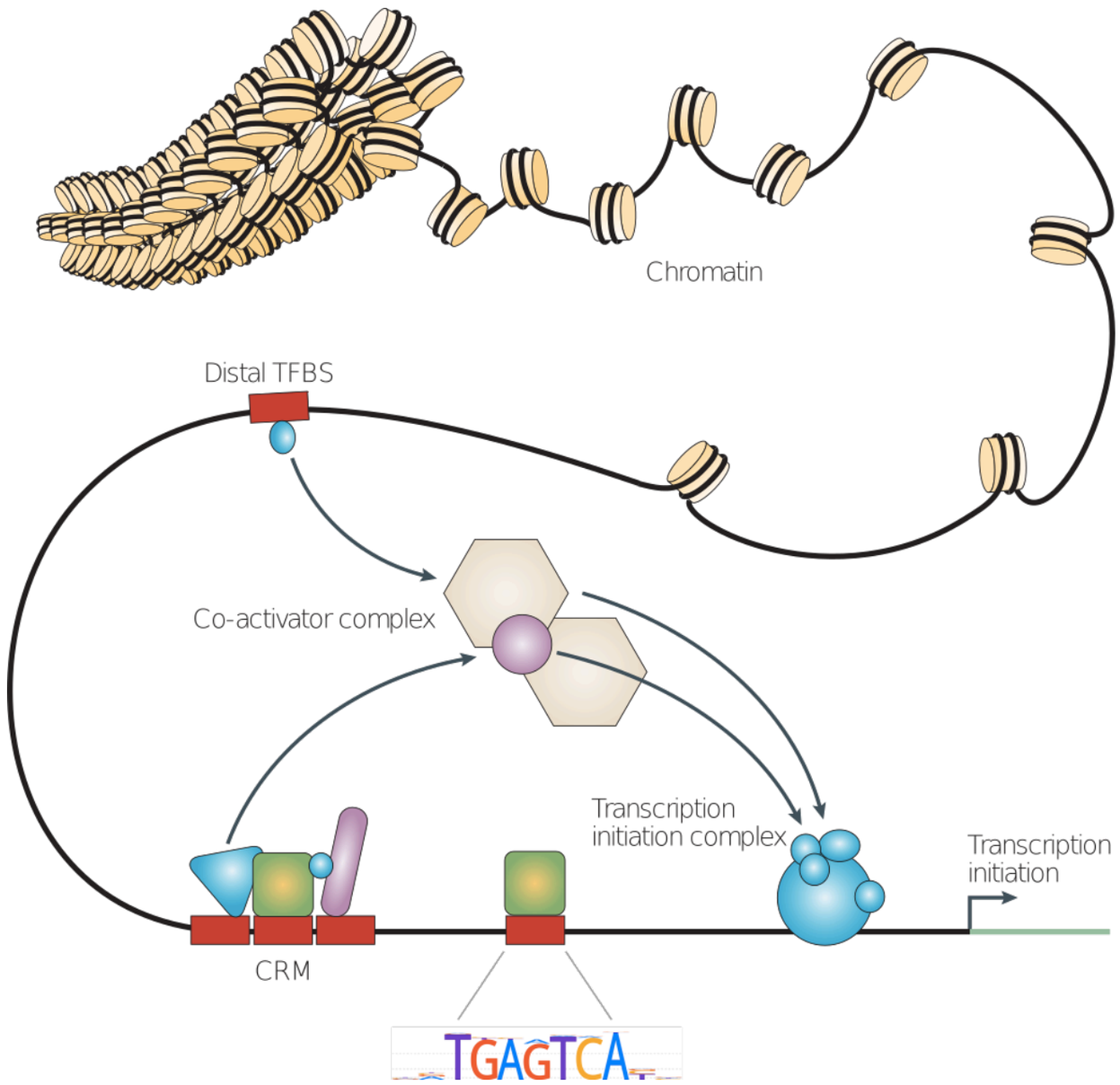
Bioinformatics Seminar: Local Multiple Alignment

Aleksandra Galitsyna
Skoltech
16.11.2017

Biological background

Proteins and nucleic acid comprise the main acting components of the cell. Major living processes of the cells are regulated via interactions between them:

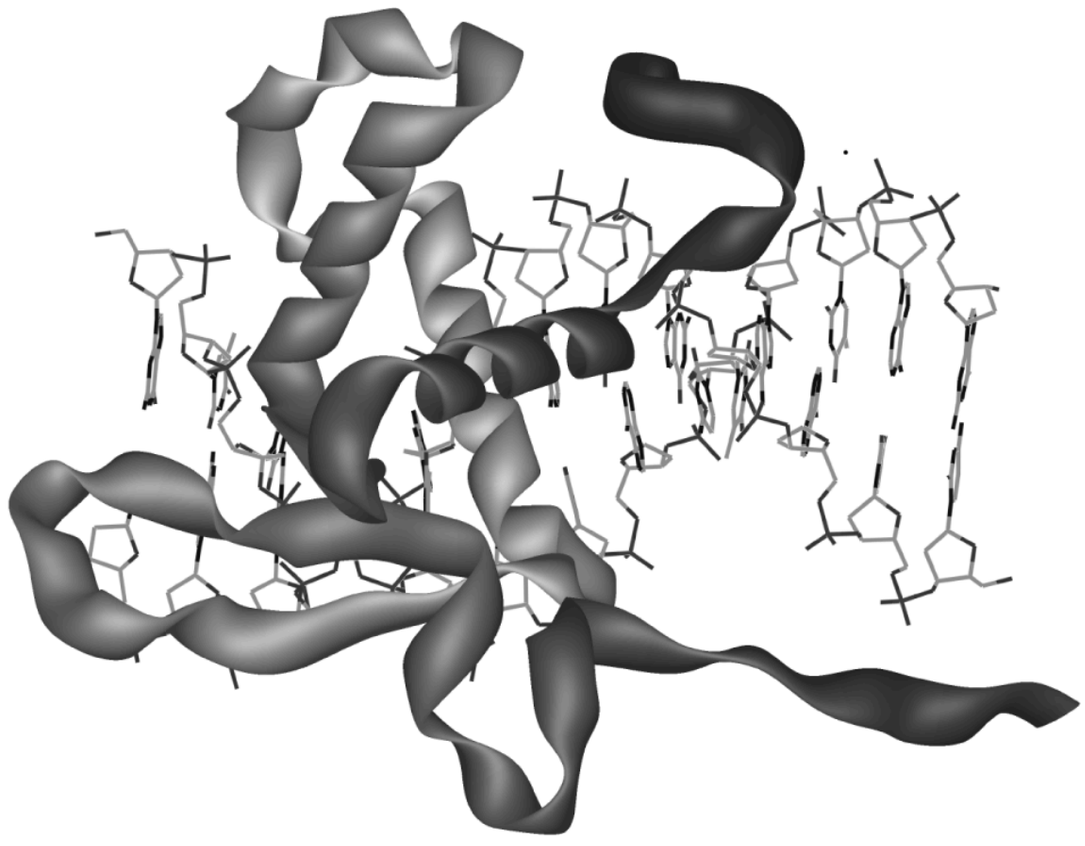
- Protein-DNA:
  - transcription
  - chromatin modification
  - replication/recombination/ DNA-repair
  - packaging (histones)
- Protein-RNA:
  - regulation of RNA metabolism
  - processing: splicing, cleavage and polyadenylation, editing
  - nuclear-cytoplasmic transport
  - translation
  - degradation: microRNA, NMD
- RNA-DNA interactions:
  - various chromatin regulation



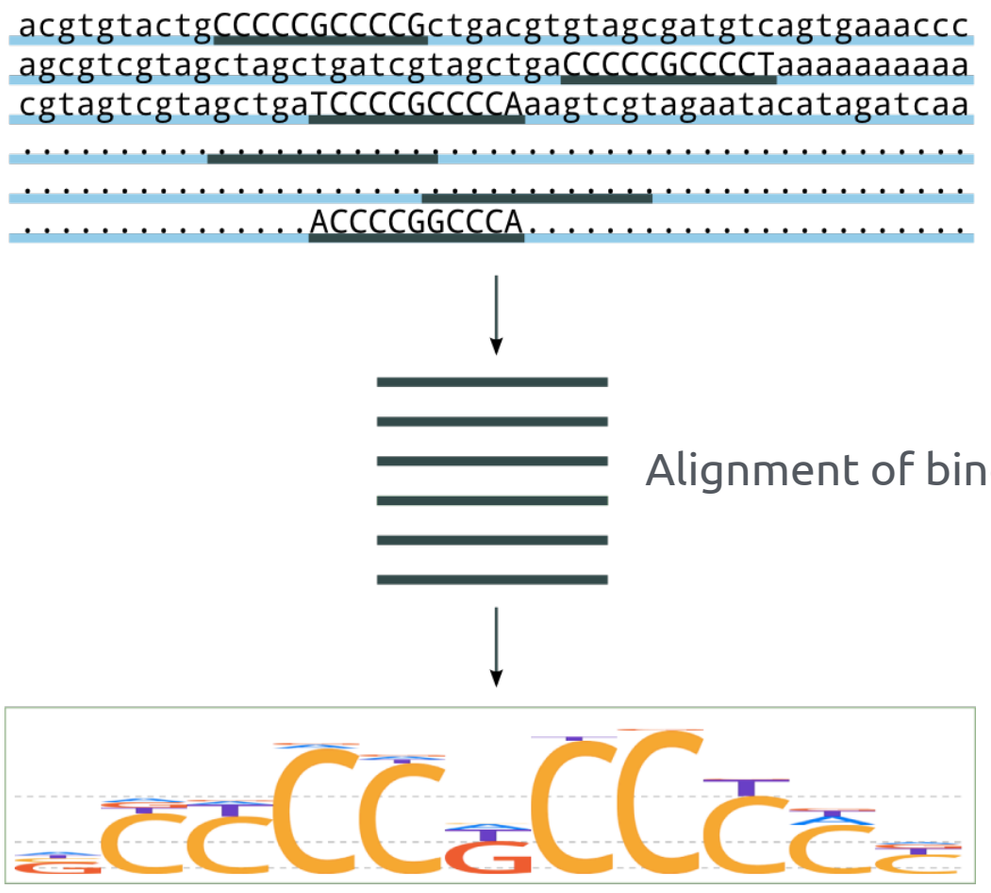
Wasserman & Sandelin, 2004

DNA-protein interactions

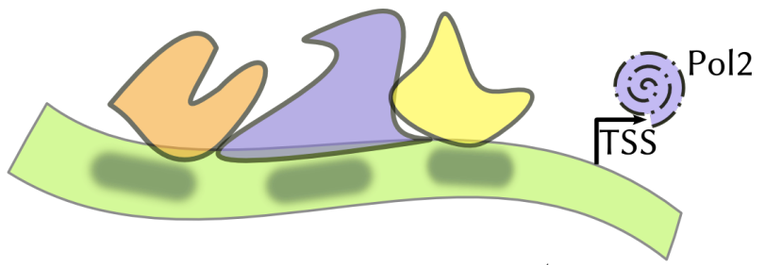
DNA-protein complex



Genomic positioning of binding sites

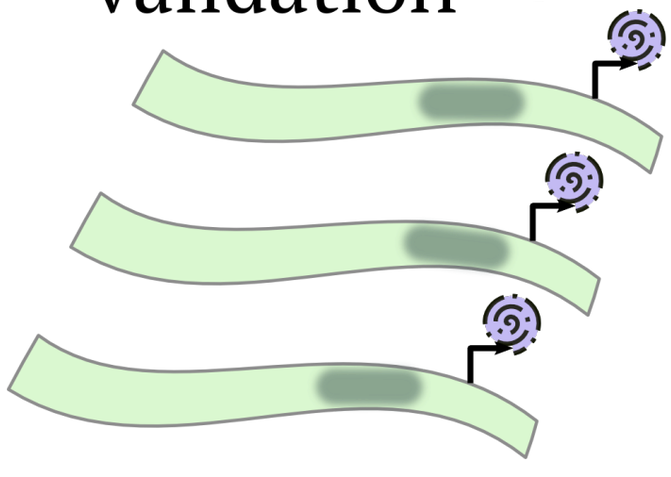


How can the information about binding sites be used?

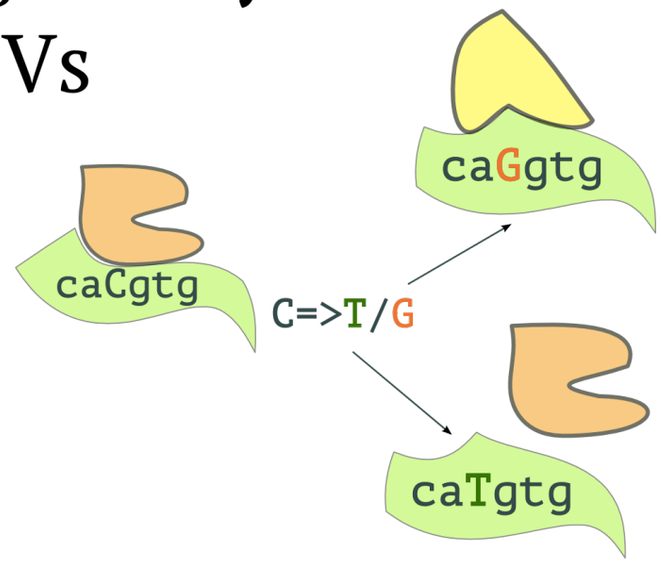


positional preferences of binding

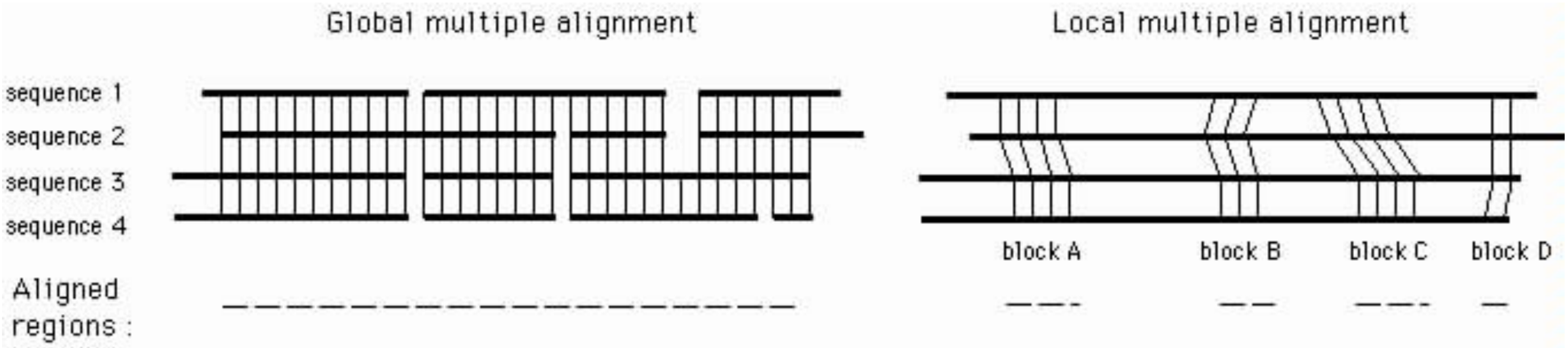
binding site & target gene prediction & validation



regulatory SNVs



Multiple alignment: global vs local



Motifs search: training attention

- Try to predict what is the regulatory motif in the following set of sequences:

atgaccgggatactgataaaaaaaaggggggggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg

acccctatttttgagcagatttagtgacctggaaaaaaatttgagtacaaaactttccgaataaaaaaaaggggggga

tgagtatccctgggatgacttaaaaaaaaggggggtgctctcccgatttttgaatatgtaggatcattcgccagggtccga

gctgagaattggatgaaaaaaaaggggggtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga

tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaataaaaaaaaggggggcttatag

gtcaatcatgttcttgtgaatggatttaaaaaaaaggggggggaccgcttggcgcacccaaattcagtgtgggcgagcgcaa

cggttttggcccttgttagaggcccccgtaaaaaaaaggggggcaattatgagagagctaatctatcgcgtgcgtgttcat

aacttgagttaaaaaaaaggggggctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta

ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcataaaaaaaaggggggggaccgaaagggaag

ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttaaaaaaaaggggggga

Motifs search: training attention

- Seems to be easy:

atgaccgggatactgatAAAAAAAAGGGGGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg

acccctatttttgagcagatttagtgacctggaaaaaaatttgagtacaaaacttttccgaataAAAAAAAAGGGGGGGa

tgagtatccctgggatgacttAAAAAAAAGGGGGGGtgctctcccgattttgaatatgtaggatcattcgccagggtccga

gctgagaattggatgAAAAAAAAGGGGGGGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga

tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAAAAAAAAGGGGGGGcttatag

gtcaatcatgttcttgtgaatggatttAAAAAAAAGGGGGGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa

cggttttggcccttgttagaggccccgtAAAAAAAAGGGGGGGcaattatgagagagctaatctatcgcgtgcgtgttcat

aacttgagttAAAAAAAAGGGGGGGctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta

ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatAAAAAAAAGGGGGGGaccgaaagggaag

ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttAAAAAAAAGGGGGGGa

Motifs search: training attention

- Let's introduce some substitutions:

atgaccgggatactgatAgAAgAAAGGttGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg

acccctatttttgagcagatttagtgacctggaaaaaaatttgagtacaaaacttttccgaatacAAtAAAAcGGcGGGa

tgagtatccctgggatgacttAAAAtAAtGGaGtGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccga

gctgagaattggatgcAAAAAAAGGGattGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga

tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAtAAtAAAGGaaGGGcttatag

gtcaatcatgttcttgtgaatggatttAAcAAtAAGGGctGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa

cggttttggcccttgttagaggcccccgtAtAAAcAAGGaGGGccaattatgagagagctaatctatcgcgtgcgtgttcat

aacttgagttAAAAAAtAGGGaGccctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta

ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatActAAAAAGGaGcGGaccgaaagggaag

ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttActAAAAAGGaGcGGa

Motifs search: training attention

- Is everything easy if you know the answer?

```
atgaccgggatactgatagaagaaaggttgggggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg
acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatacaataaaacggcggga
tgagtatccctgggatgacttaaaataatggagtggtgctctcccgatttttgaatatgtaggatcattcgccagggtccga
gctgagaattggatgcaaaaaaagggattgtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatataataaaggaagggcttatag
gtcaatcatgttcttgtgaatggatttaacaataagggctgggaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
cggttttggcccttgttagaggcccccgtataaacaaggagggccaattatgagagagctaatctatcgcgtgcgtgttcat
aacttgagttaaaaaatagggagccctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatactaaaaaggagcggaccgaaagggaag
ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttactaaaaaggagcgga
```

Motif representation: consensus sequence

Consensus sequence lists nucleotides that are allowed in given position.
Consider following gapless block of local alignment:

```
TATAAT
TAAAAT
TAATAT
TGTAAT
TATACT
```

Its consensus:

```
T[AG][AT][AT][AC]T
```

Problems:

- Doesn't allow to incorporate different preferences for different nucleotides,
- Doesn't allow to account for background nucleotides frequencies.

Motif representation: position weight matrix

```
123456
TATAAT
TAAAAT
TAATAT
TGTAAT
TATACT
```

frequency matrix

	1	2	3	4	5	6
A	0	4	2	4	4	0
C	0	0	0	0	1	0
G	0	1	0	0	0	0
T	5	0	3	1	0	5

probability matrix

	1	2	3	4	5	6
A	0.0	0.8	0.4	0.8	0.8	0.0
C	0.0	0.0	0.0	0.0	0.2	0.0
G	0.0	0.2	0.0	0.0	0.0	0.0
T	1.0	0.0	0.6	0.2	0.0	1.0

$$M_{p,n} = \log 2\left(\frac{p_{p,n}}{b_n}\right)$$

$p_{p,n}$ is probability of nucleotide n in position p

b_n is probability of nucleotide n in background

	1	2	3	4	5	6
A	-Inf	1.6	0.6	1.6	1.6	-Inf
C	-Inf	-Inf	-Inf	-Inf	-0.3	-Inf
G	-Inf	-0.3	-Inf	-Inf	-Inf	-Inf
T	2	-Inf	1.2	-0.3	-Inf	2

Add pseudocounts (for example, 1), to frequency matrix to evade infinity in PWMs. Pseudocounts reflect the fact, that any sequence can be bound by the protein. But some of them are bound with very low probability

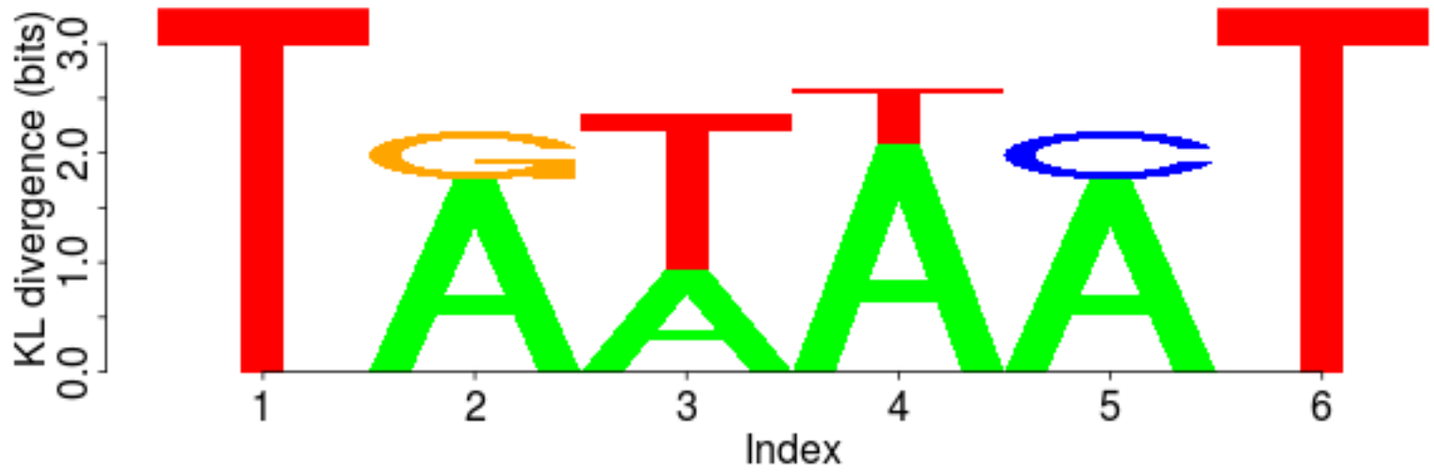
	1	2	3	4	5	6
A	-1.2	1.2	0.4	1.2	1.2	-1.2
C	-1.2	-1.2	-1.2	-1.2	-0.2	-1.2
G	-1.2	-0.2	-1.2	-1.2	-1.2	-1.2
T	1.4	-1.2	0.8	-0.2	-1.2	1.4

PWM visualisation: logo

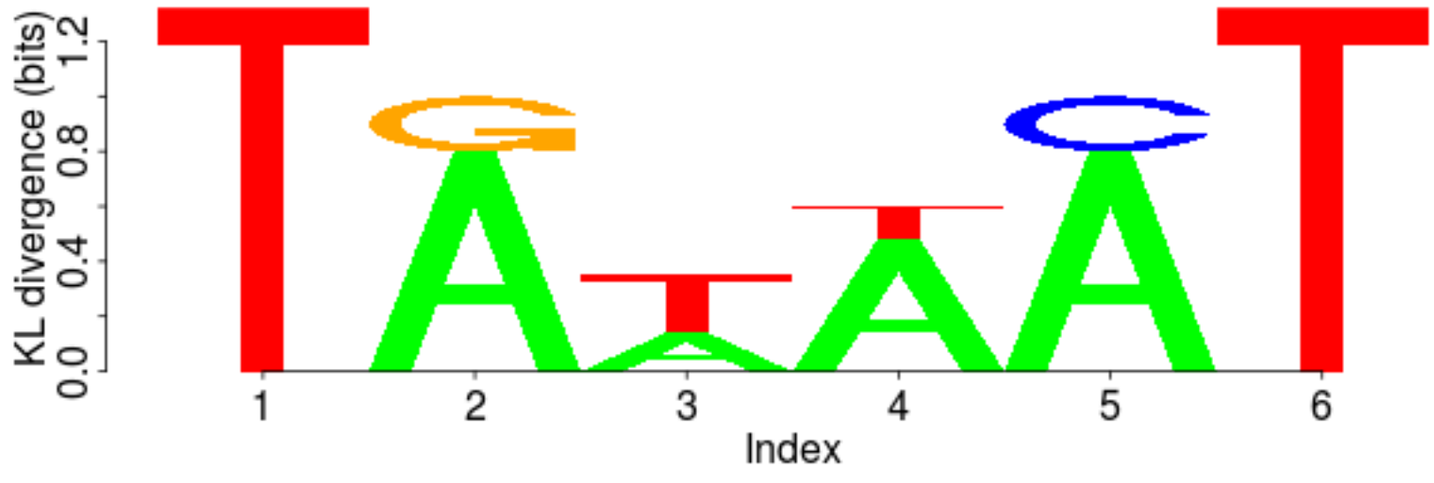
Height of each column is significance of the position (dissimilarity to background).

Relative size of the letter is a frequency of the nucleotide.

GC-rich background:



AT-rich background:



Search of motif instances in a given sequence

- Let's imagine that we know particular motif and its PWM for some protein. How can we find the binding sites of this protein in the genome?

motif logo

T G A T A C

given sequence (e.g. genome)

A	-0.2	-1.8	1.2	-1.8	1.0	-0.2
C	-1.8	-1.8	-1.8	-1.8	-1.8	0.4
G	0.4	1.2	-1.8	-1.8	-1.8	-0.2
T	0.4	-1.8	-1.8	1.2	-0.2	-0.2
	1	2	3	4	5	6

motif PWM

$$\sum 0.4 + 1.2 + 1.2 + 1.2 + 1.0 + 0.4$$

score

$$\textit{Score}(\text{tGATAc}) = 5.4$$

- Additivity assumption: score is larger for longer sequences!

Search of motif instances

Motif model (e.g. positional weight matrix, PWM)

	1	2	3	4	5	6
A	-1.6	-1.6	0.96	-1.6	-1.6	0.96
C	-1.6	-1.6	0.00	-1.6	-1.6	-1.6
G	1.22	1.22	-1.6	-1.6	-1.6	-1.6
T	-1.6	-1.6	-1.6	1.22	1.22	0.00

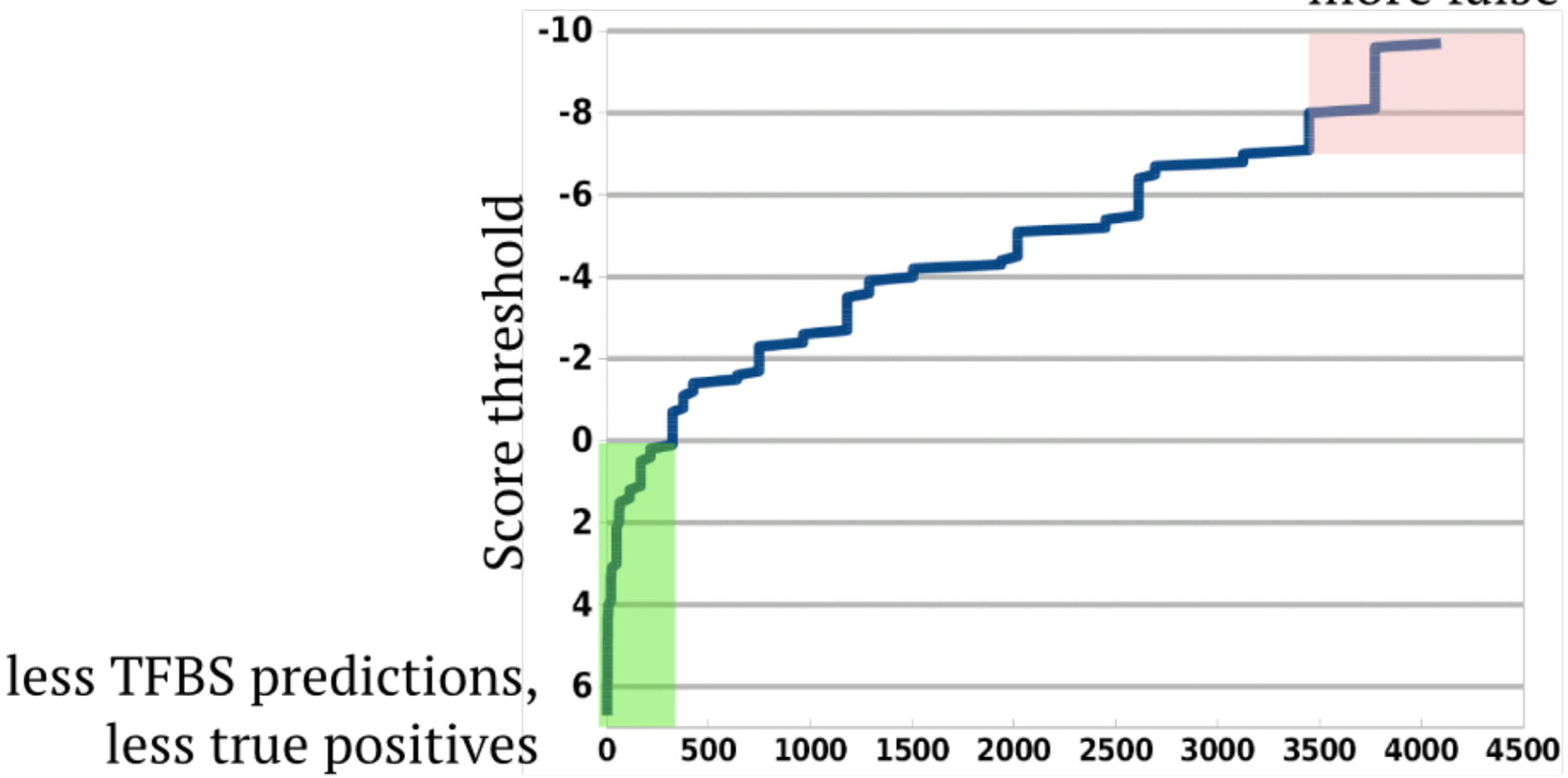
PWM

GGATTA → S_{GGATTA}=1.22+1.22+0.96+1.22+1.22+0.96=**6.8** the best score

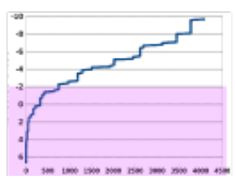
S_{GGGGGG}=2.44-6.4=**-3.96**

S=**-9.6** the worst score

more predicted TFBS,
more false positive predictions

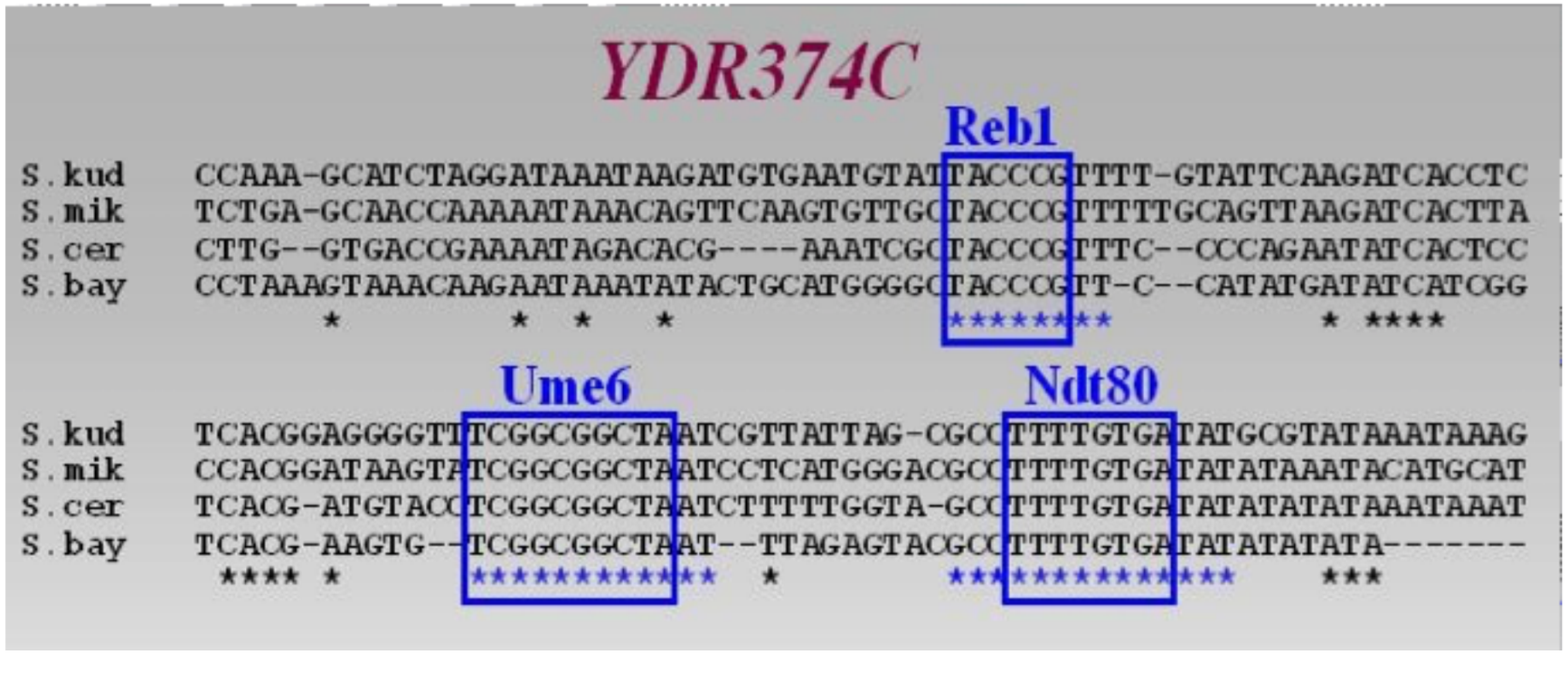


less TFBS predictions,
less true positives



Score threshold turns a motif model into a binary "yes/no" classifier!

Motifs search: generalized approach



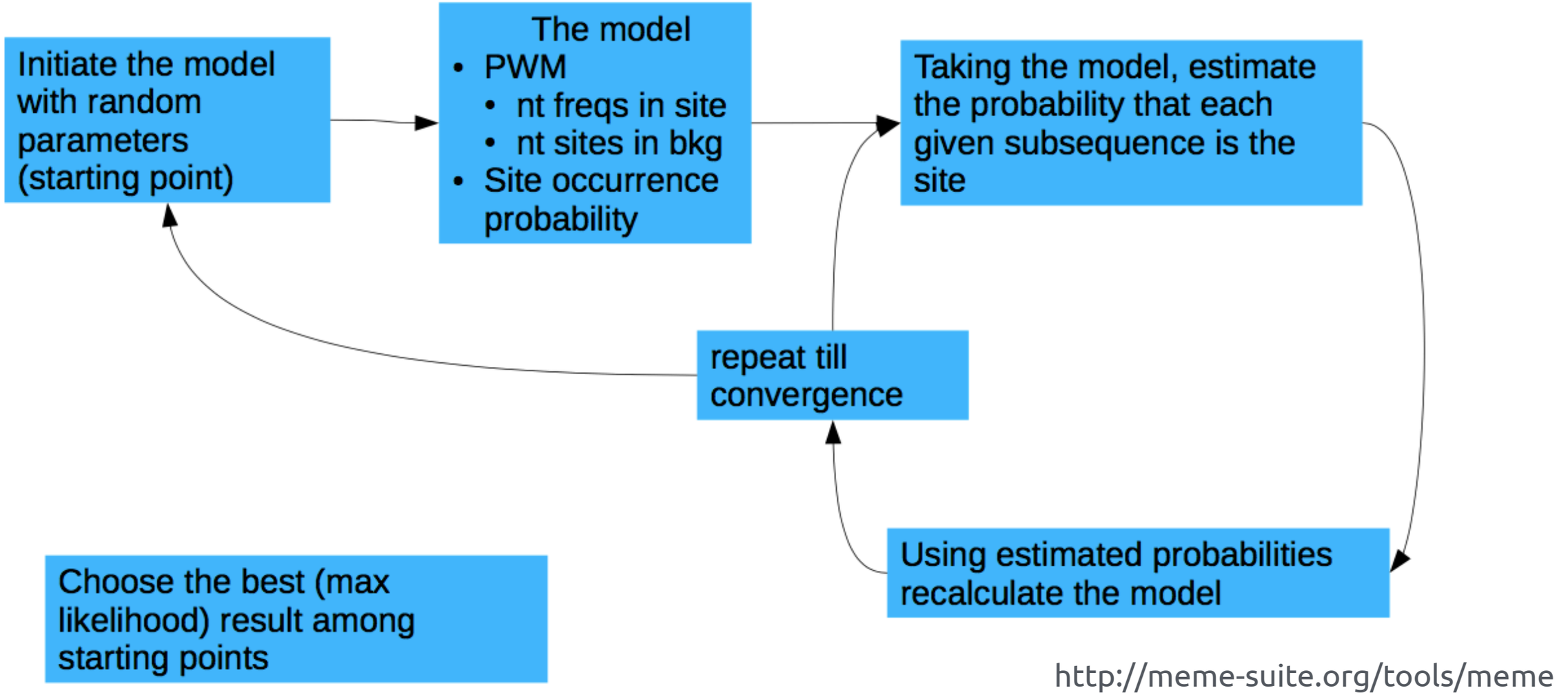
- Count number of occurrence of all possible kmers (or number of sequences that contain given kmer) and compare with random expectation (same count but in shuffled sequences).
- If we have background sequences we can compare kmer occurrence in test and in background set.

MEME

Input is unaligned sequences that can contain 0-n instances of some unknown motif:

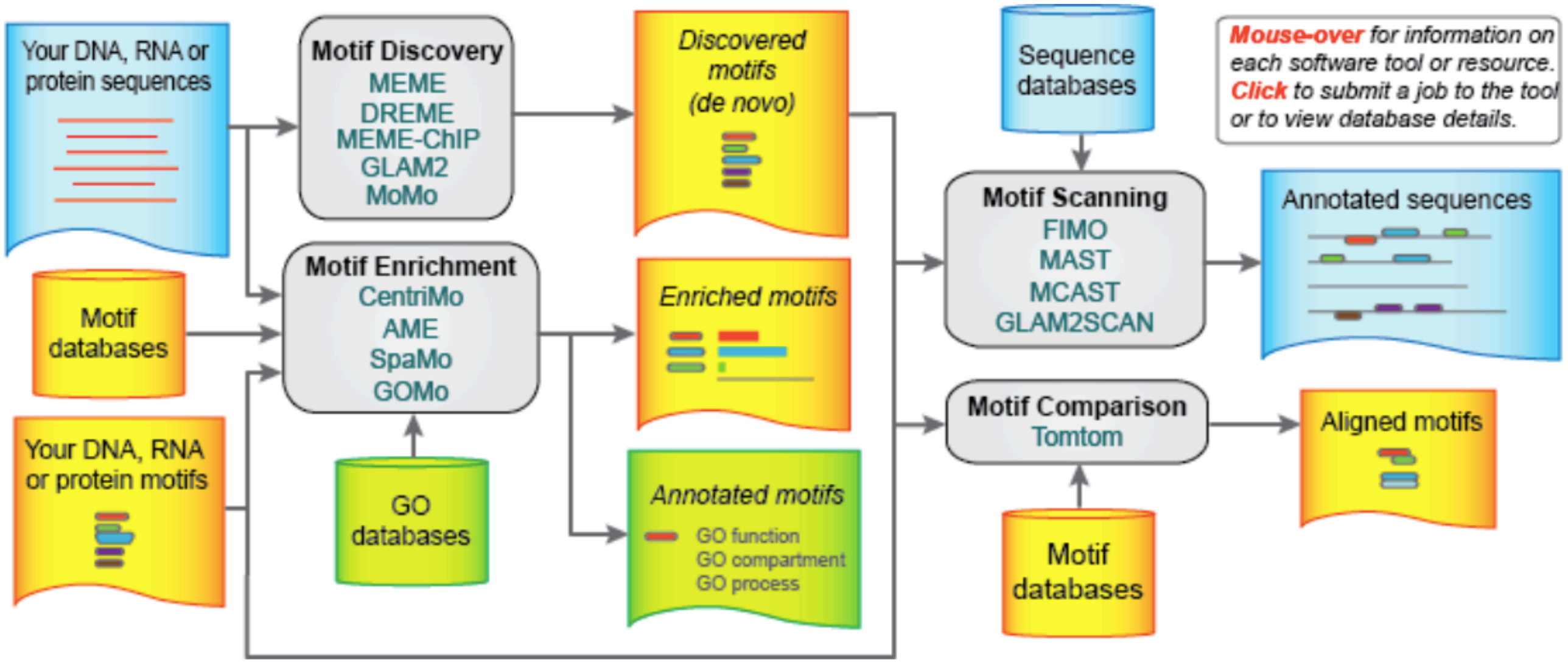
- ChIP-seq sites
- Promoters of co-regulated genes
- Promoters of orthologous genes

Algorithm:



http://meme-suite.org/tools/meme

MEME-Suite



http://meme-suite.org/tools/meme

Some tools for motifs search and manipulation

- Web server tools:
- http://rsat.eu/
- http://meme-suite.org/

- Console tools:
- http://autosome.ru/

- Tools embedded in programming languages:
- BioPython motifs

- For more info see: https://omictools.com/motif-discovery-category

Task 1. From multiple alignment to motif search

Directory with files: http://makarich.fbb.msu.ru/agalicina/2017/Skoltech/

1. Download file from previous seminar with upstream regions of bacterial orthologs (upstreams.fasta). Create multiple alignment with T-Coffee (http://tcoffee.crg.cat/apps/tcoffee/do:mcoffee), manually select the most conservative gapless region and save it into .fasta file.
2. Create counts, frequencies, weights matrices and logo from gapless alignment with RSAT tools: http://embnet.ccg.unam.mx/rsat/ -> Procaryotes RSAT -> Matrix tools.
3. Process the same set of sequences upstreams.fasta with MEME: http://meme-suite.org/. Set possible length of motif from 5 to 10. Is the result similar to what you found manually?

Task 2. Motif search in ChIP-Seq data

Directory with files: http://makarich.fbb.msu.ru/agalicina/2017/Skoltech/

1. Download file with peaks sequences from the given **chicken** ChIP-Seq (peaks.fasta) and find motifs with MEME-ChIP (http://meme-suite.org/ -> MEME-ChIP). What was the protein used for ChIP-Seq?
2. Log in to the server and analyse this data with ChIP-Munk console program provided at autosome.ru. Retrieve PWM from program output and save it to the file called gallus_ctcf.pwm in simple tabular format.
3. Search resulting motif against vertebrates motifs database (TOMTOM, http://meme-suite.org/tools/tomtom). Does it produce the same result as MEME?
4. Check this DNA binding protein in JASPAR database of various profiles (http://jaspar.genereg.net/). For what species is it available? What are the differences between profiles?

Task 3. Motifs analysis, comparison and search

1. Log in to the server and download gallus_chr1.fasta file. Create plain tabular files with CTCF profiles obtained in previous task (gallus_ctcf.pwm from ChIP-Munk, homo_ctcf.pfm and dros_ctcf.pfm from JASPAR).
2. Switch to Python terminal and with BioPython packages read the motifs, compare them, search gallus_chr1.fasta for instances. Follow the recommendations provided in the next slides.

BioPython: reading motifs from files

- "Motifs" module in Bio package is dedicated to motifs analysis. import it:

```
from Bio import motifs
```

- Read the motifs for different species into separate variables, for example:

```
m_dros = motifs.read(open(“dros_ctcf.pfm"),"pfm")
```

- m_dros is BioPython object with all the information about this motif and its profile:

```
m_dros.consensus
len(m_dros.consensus)
m_dros.pwm
m_dros.pssm
```

- PSSM has some "Inf", let's add pseudocounts to get rid of it:

```
m_dros.pseudocounts = {'A': 0.1, 'C': 0.1, 'T': 0.1, 'G': 0.1}
m_dros.pssm
```

- Repeat the same for Homo and Gallus profiles.

BioPython: comparing motifs

- We can simply calculate the Pearson correlation between profiles:

```
m_homo.pssm.dist_pearson(m_dros.pssm)
```

- Repeat comparisons between all profiles. What are most similar ones? Is it coherent with phylogenetic relationship between species?

BioPython: search chicken genome for motif instances

- First of all, we need to read input fasta file with fragment of chromosome 1. For that we need modules SeqIO and Alphabet:

```
from Bio import SeqIO
from Bio.Alphabet.IUPAC import IUPACUnambiguousDNA
seq = SeqIO.read("gallus_chr1.fasta", "fasta",
alphabet=IUPACUnambiguousDNA())
```

- Now we can simply search for instances with arbitrary score threshold 5.0:

```
[(x,y) for (x,y) in m_dros.pssm.search(seq.seq, threshold=5.0)]
```

- Repeat this action for all the profiles with threshold 5.0. For what species number of occurrences is higher? Can it be explained by profile length?

BioPython: search chicken genome with FPR-fixed threshold

- The problem is that the raw score is not comparable between instances for different profiles. We have to bind it to some statistical property, for example, False Positive Rate (FPR). Let's set it to 0.0001 and calculate corresponding score for each profile:

```
m_gallus.pssm.distribution().threshold_fpr(0.0001)
```

- Set the search threshold corresponding to 0.0001 FPR for this particular profile:

```
[(x,y) for (x,y) in m_dros.pssm.search(seq.seq,
threshold=9.96)]
```

Has the number of occurences changed?

- Repeat this for all the species and compare the number of motif instances between different species and for different thresholds.

Home tasks: TBA

1. The tasks will be available on Canvas soon, please, check Announcements.
2. Contact me in case of any troubles: aleksandra.Galitsyna@skoltech.ru

About this seminar

- Lectures of Pavel Mazin, Ivan Kulakovskiy and materials from bioalgorithms.info were used for the theoretical part.
- Tasks were inspired by BioPython manual: http://biopython-cn.readthedocs.io/zh_CN/latest/en/chr14.html