

Global Multiple Alignment Local Multiple Alignment

Aleksandra Galitsyna
Bioinformatics seminar
10.09.2020

Skoltech

Skolkovo Institute of Science and Technology

Alignment

- Can be applied to any sequence (DNA, RNA, protein or other)
 - Pairwise alignments (2 sequences):

- Multiple alignments (≥ 3 sequences):

Alignment formats

Clustal W:

```
CPZANT ATGGGAGCGGGGGCGTCTGTTTGAGGGGAGAGAAGCTAGATAACATGGGA
U455    ATGGGTGCGAGAGCGTCAGTATTAAGCGGGAAAAAAATTAGATTACATGGGA

CPZANT AAGTATCAGGCTTCGGCCCGTGGCAAGAAAAAGTACATGATAAAACATC
U455    GAAAATT CGGTTAACGCCAGGGGGAAACAAAAAATATAGACTGAAACATT

CPZANT TGGTTTGGGCAAGATCGGAGCTGCAGCGTTGCGCTCAGCTCCCTCCCTT
U455    TAGTATGGGCAAGCAGGGAGCTGGAAAAATTCAACTAACCTGGCCTT

CPZANT CTAGAAACATCAGAAGGTTGTGAAAAGGCTATCCATCAATTGAGCCCTTC
U455    TTAGAAACAGCAGAAGGATGTCAGCAAATACTGGGACAATTACAACCAGC

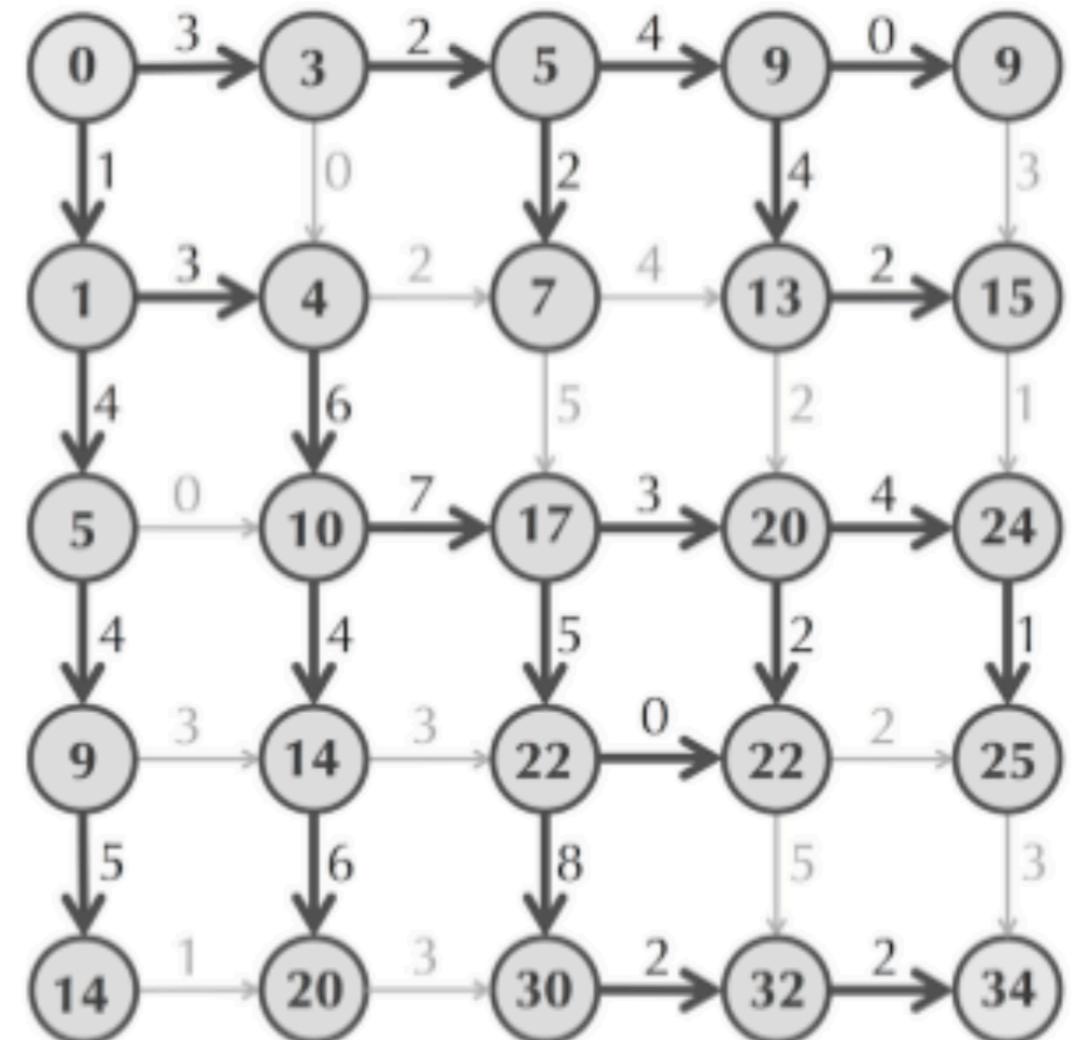
CPZANT CATAGAAATAAGATCCCCTGAAATAATATCTTGTAAACACCATTGTG
U455    TCTCCAGACAGGAACAGAAGAACTTAGATCATTATATAATACAGTAGCAG
```

FastA:

```
>CPZANT
ATGGGAGCGGGGGCGTCTGTTTGAGGGGAGAGAAGCTAGATAACATGGGA
AAGTATCAGGCTTCGGCCCGTGGCAAGAAAAAGTACATGATAAAACATC
TGGTTTGGGCAAGATCGGAGCTGCAGCGTTGCGCTCAGCTCCCTCCCTT
CTAGAAACATCAGAAGGTTGTGAAAAGGCTATCCATCAATTGAGCCCTTC
CATAGAAATAAGATCCCCTGAAATAATATCTTGTAAACACCATTGTG
>U455
ATGGGTGCGAGAGCGTCAGTATTAAGCGGGAAAAAAATTAGATTACATGGGA
GAAAATT CGGTTAACGCCAGGGGGAAACAAAAAATATAGACTGAAACATT
TAGTATGGGCAAGCAGGGAGCTGGAAAAATTCAACTAACCTGGCCTT
TTAGAAACAGCAGAAGGATGTCAGCAAATACTGGGACAATTACAACCAGC
TCTCCAGACAGGAACAGAAGAACTTAGATCATTATATAATACAGTAGCAG
```

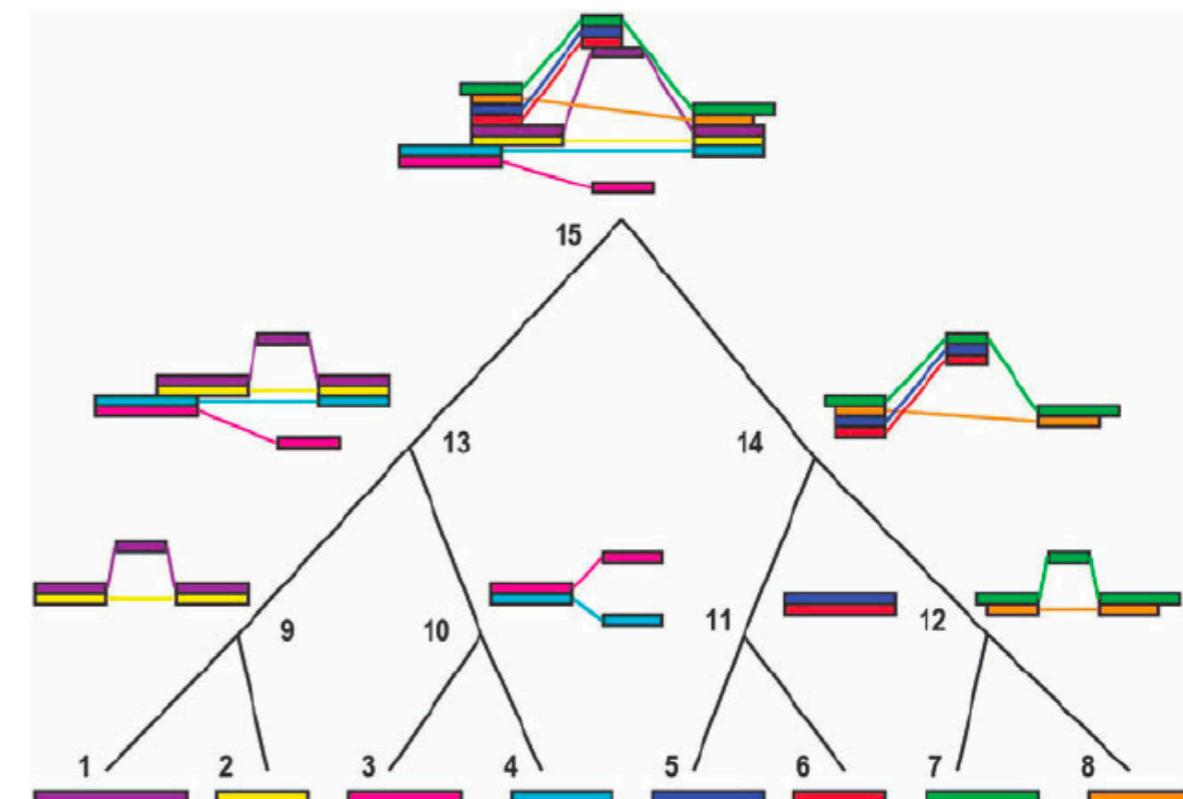
Algorithms for alignment

- Ideal: Dynamical Programming
- Heuristics:
 - progressive alignment construction
 - iterative methods
 - consensus methods
 - genetic algorithms



Progressive alignments

- finds a guiding tree using one of clustering methods
- builds a multiple alignment progressively, starting from most similar sequences, stacking them as in the guiding tree
 - + efficient enough to work with up to 1000 sequences
 - does not provide a global optimal alignment
 - errors in the first steps do propagate to the final alignment



Iterative methods

- works similar to progressive algorithms, but allows to realign the sequences in the alignment on each step
- optimizes a global metric
 - + less prone to error propagation, provides a more accurate result
 - + works fine with pairwise distant sequences
 - still heuristic
 - not as efficient as progressive algorithms

Popular aligners

Aligner Algorithm	Type	Input	Comments
MUSCLE	Iterative	DNA, RNA, proteins	Widely used. Allows a lot of options
CLUSTAL Omega	Progressive	DNA, RNA, proteins	$O(N \log N)$ guide tree production allows over 100 000 sequences to be aligned. Can reuse existing alignment and append new sequences to them
T-Coffee	Progressive	DNA, RNA, proteins, structures	Wide range of flavors for different situations, e.g. DNA, RNA, proteins. Different modes for fast, accurate, memory-efficient aligning
MAFFT	Iterative	DNA, RNA, proteins	One of the most accurate algorithms for less than 100 sequences. Allows large gaps, making it suitable for rRNA alignments

Popular aligners

- ClustalW and ClustalO
 - documentation, servers and download page: <http://www.clustal.org/>
 - try clustalw -INFILE=<fasta> and clustalo --auto --in <fasta> in terminal
- MUSCLE
 - documentation and download page: <http://www.drive5.com/muscle/>
 - server: <https://www.ebi.ac.uk/Tools/msa/muscle/>
 - try muscle -in <fasta> in terminal
- T-Coffee
 - Coffee family: <http://www.tcoffee.org/homepage.html>
 - documentation, servers and download page:
<http://www.tcoffee.org/Projects/tcoffee/>
- MAFFT
 - documentation, servers and download page:
<https://mafft.cbrc.jp/alignment/software/>

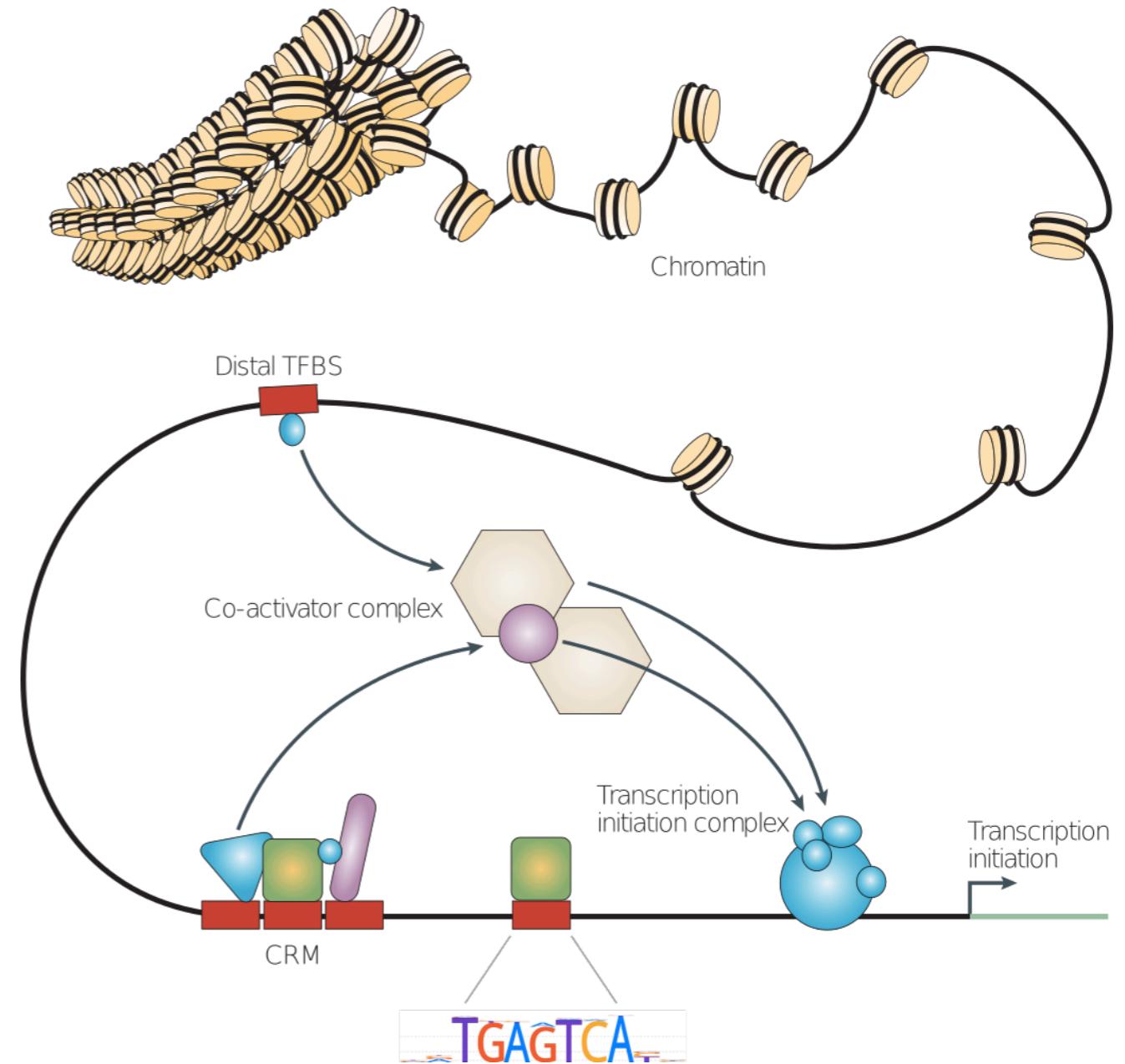
How to run aligners?

- Online Tools through Web Interface, for small tasks for manual curation:
 - <https://www.ebi.ac.uk/Tools/msa/>
- Standalone programs for larger tasks and manual curation:
 - JalView: <https://www.jalview.org/>
 - MEGA
- From bash terminal: Command Line Interface (CLI), for the large and time-consuming tasks
- From programming languages, for full control over input/output:
 - **BioPython** in Python
 - **SciKit-Bio** for simple alignments and files parsing in Python
 - **msa** package for R

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:

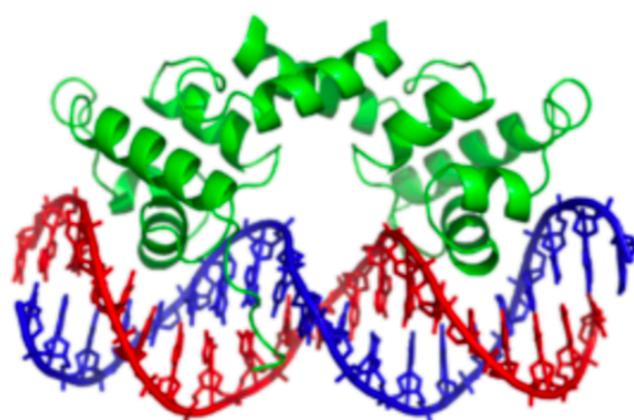
- **Protein-DNA:**
 - transcription
 - chromatin modification
 - replication/recombination/DNA-repair
 - packaging (histones)
- **Protein-RNA:**
 - regulation of RNA metabolism
 - processing: splicing, cleavage and polyadenylation, editing
 - nuclear-cytoplasmic transport
 - translation
 - degradation: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



DNA-protein interactions

Binding of protein is typically specific to DNA sequence:

DNA-protein binding

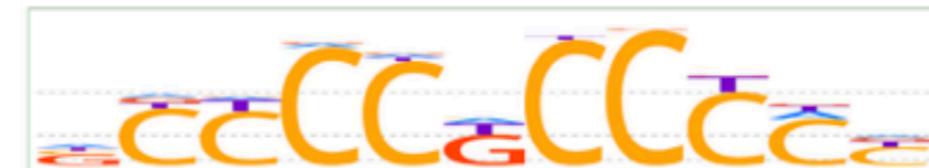


List of DNAs
with binding events

```
acgtgtactgCCCCCGCCCCGctgacgtgttagcgatgtcagtgaaaccc  
agcgtcgttagctgatcgtagctgaCCCCCGCCCCCTaaaaaaaaaaaa  
cgtagtcgttagctgaCCCCCGCCCCAaagtgcgtagaatacatagatcaa  
.....  
.....ACCCCGGGCCA.....
```



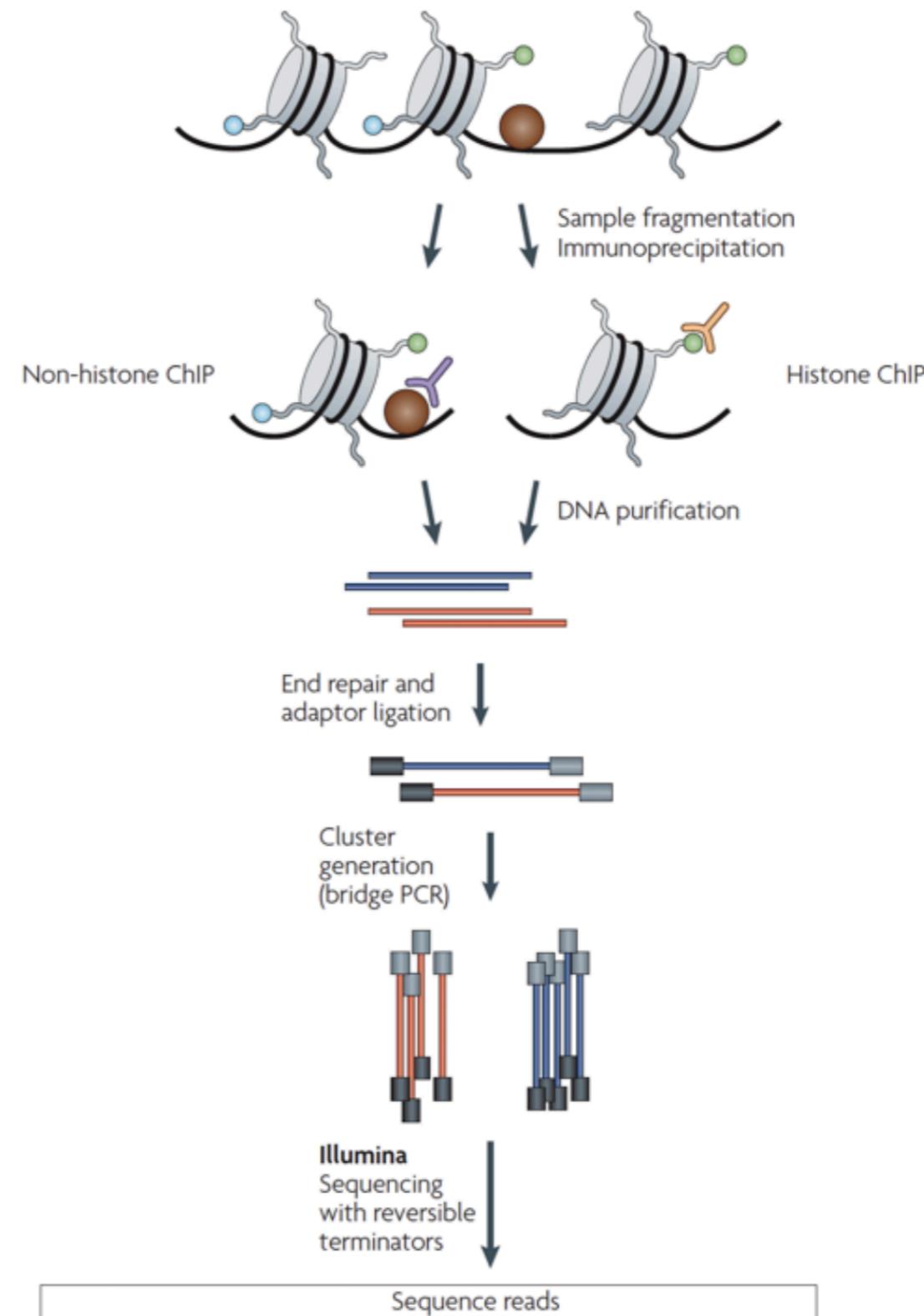
Alignment of binding sites



Binding pattern

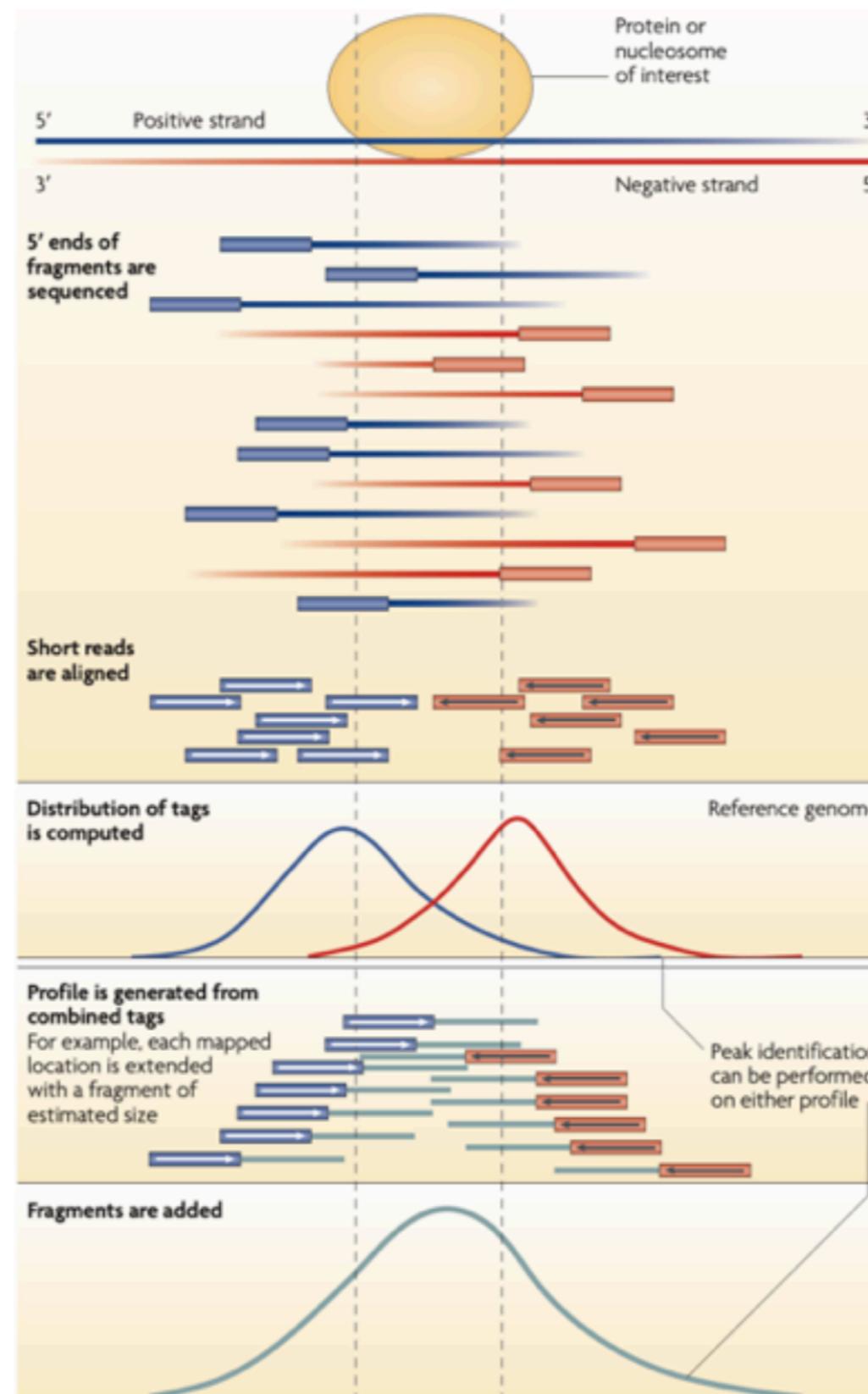
ChIP-Seq basics

Chromatin-
immunoprecipitation
followed by sequencing:



ChIP-Seq basics

Binding events:



Peak calling:

Motifs search: training attention

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

```
atgaccgggatactgataaaaaaaaagggggggggcgtacacattagataaacgtatgaagtacgttagactcgccgcgg  
acccctattttttagcagattttagtgacctggaaaaaaaaattttagtacaactttccgaataaaaaaaaaaggggggga  
ttagtatccctggatgactaaaaaaaaaggggggtgctctccgattttgaatatgttaggatcattgccagggtccga  
gctgagaattggatgaaaaaaaaaggggggtccacgcaatcgcaaccaacgcggacccaaaggcaagaccgataaaggaga  
tccctttgcgtaatgtgccggaggctggttacgttaggaagccctaacggacttaataaaaaaaaaggggggcttata  
gtcaatcatgttcttgtaatggattaaaaaaaaagggggggaccgcttggcgccccaaattcagtgtggcgagcgcaa  
cggtttggccctttagaggccccgtaaaaaaaaagggggggcaattatgagagagctaattatcgctgcgtttcat  
aactttagttaaaaaaaaaggggggctgggcacatacaagaggagtttcattcagttaatgttatgacactatgt  
ttggcccattggctaaagcccaacttgacaaatggaagatagaatccttcataaaaaaaaaaggggggaccgaaaggaaag  
ctggtagcaacgacagattttacgtcattagctcgcttccgggatctaatacgacgaaagctttaaaaaaaaagggggga
```

Motifs search: training attention

- Seems to be easy:

atgaccggatactgat **AAAAAAAAGGGGGGGG**ggcgtacacattagataaacgtatgaagtacgttagactcgccgcgg
acccctattttttagcagattttagtgacctggaaaaaaaaattttagtacaactttccgaata **AAAAAAAAGGGGGGGG**a
tgagtatccctggatgactt **AAAAAAAAGGGGGGGG**tgctctcccattttgaatatgttaggatcattcgccagggtccga
gctgagaattggatg **AAAAAAAAGGGGGGGG**tccacgcaatcgcaaccaacgcggacccaaaggcaagaccgataaaggaga
tccctttcggtaatgtgccggaggctggttacgttaggaagccctaacggacttaat **AAAAAAAAGGGGGGGG**cttata
gtcaatcatgttcttgtaatggat **AAAAAAAAGGGGGGGG**gaccgcttggcgccccaaattcagtgtggcgagcgcaa
cggtttggccctttagaggccccgt **AAAAAAAAGGGGGGGG**caattatgagagagactaatctatcgctgcgtttcat
aactttagtt **AAAAAAAAGGGGGGGG**ctggggcacatacaagaggagtcttcattatcagttaatgttatgacactatgt
ttggccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatt **AAAAAAAAGGGGGGGG**accgaaaggaaag
ctggtagcaacgacagattttacgtgcattagctcgcttccgggatctaatacgacgaaatctt **AAAAAAAAGGGGGGGG**a

Motifs search: training attention

- Let's introduce some substitutions:

atgaccggatactgat **A**g**A**g**AAAGGttGGG**ggcgtacacattagataaacgtatgaagtacgttagactcgccgcgg
accctat~~ttttt~~gagcagat~~ttt~~gacctggaaaaaaaat~~t~~tgagtaca~~aa~~act~~ttt~~ccgaata **C****A****A****t****AAAACGGCGGG**a
t~~g~~agtatccctggatgactt **AAAAtAAtGGaGtGG**tgctctcccattttgaatatgttaggatcattgccagg~~t~~ccga
gctgagaattggatg **c****AAAAAAAGGGattG**tccacgcaatcg~~cg~~gaaccaacgcggacc~~cc~~aaaggcaagaccgataaaggaga
tccctttgcgtaatgtgccggaggctgg~~t~~acgttag~~g~~gaagcccta~~c~~ggacttaat **A****tAAtAAAGGaaGGG**cttata~~g~~
gtcaatcatgttcttgt~~g~~aatggat~~t~~tt **AACAA****tAAGGGctGG**gaccgcttggcgacccaaattc~~ag~~tgtggcgagcgcaa
cggtttggcc~~tt~~gttagaggcccc~~cgt~~ **A****tAAA****CAAGGAGGGc**caattatgagagagcta~~at~~tatcg~~cgtgcgt~~ttcat
aactt~~g~~agtt **AAAAA****tAGGGaGcc**ctggggcacata~~ca~~agaggagtcttc~~tt~~atc~~ag~~ttaatg~~ct~~gtatg~~ac~~actatgta
ttggcccattggctaaaagcccaacttgacaaatggaagatagaatc~~ctt~~gc~~at~~t **A****c****t****AAAAGGAGC****GG**accgaaagg~~ga~~ag
ctgg~~t~~gagcaacgacagattcttacgtgcattagctcgcttccggggatcta~~at~~g~~c~~acgaagctt **A****c****t****AAAAGGAGC****GG**a

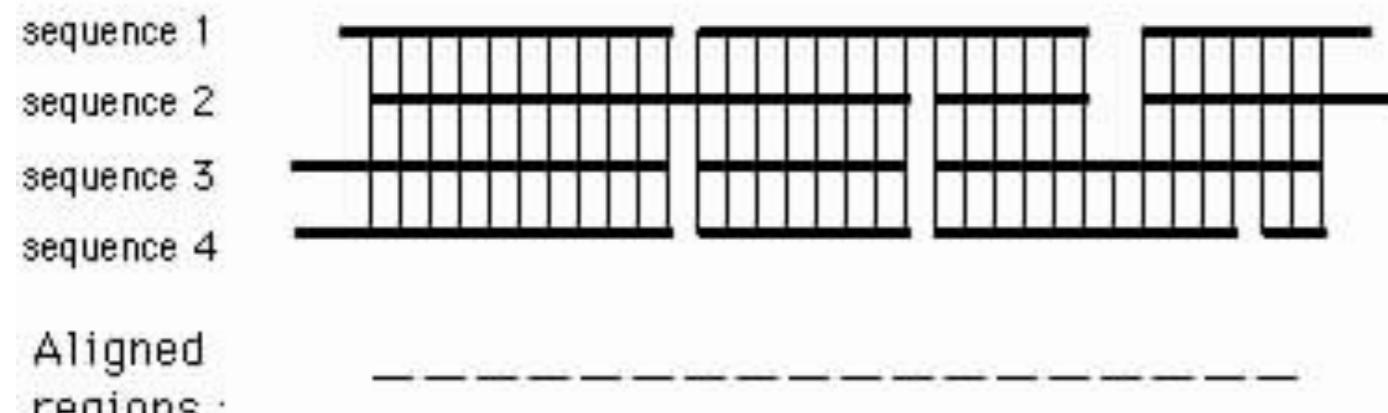
Motifs search: training attention

- Is everything easy if you know the answer?

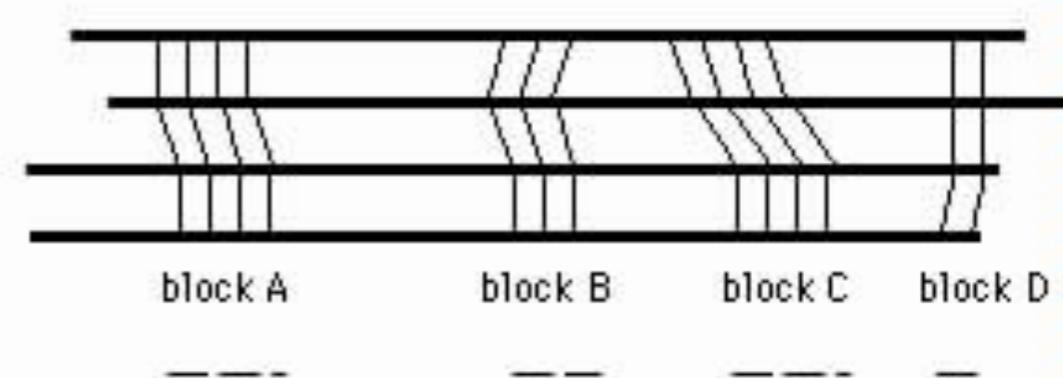
```
atgaccggatactgatagaagaaagggtggggcgtaacacattagataaacgtatgaagtacgttagactcggcgccg  
acccctattttttagcagattttagtgacctggaaaaaaaaattttagtacaaaaactttccgaatacaataaaacggcggga  
ttagtatccctggatgactaaaataatggagtggctctccgattttgaatatgttaggatcattgccagggtccga  
gctgagaattggatgcaaaaaaaggattgtccacgcaatcgcaaccaacgcggacccaaaggcaagaccgataaaggaga  
tccctttgcggtaatgtgccggaggctggttacgttaggaagccctaacggacttaatataataaaggaaggcattatag  
gtcaatcatgttcttgtaatggatttacaataaggctggaccgcttggcgccccaaattcagtgtggcgagcgcaa  
cggtttggccctttagaggccccgtataaaacaaggaggccaattatgagagagctaattatcgctgcgtgttcatt  
aacttgagtaaaaataggagccctgggcacatacaagaggagtcttcattatcagttaatgttatgacactatgta  
ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatactaaaaaggagcggaccgaaaggaaag  
ctggtagcaacgacagattttacgtcattagctcgcttccgggatctaatacgacgaagcttactaaaaaggagcggaa
```

Multiple alignment: global vs local

Global multiple alignment



Local multiple alignment



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

	frequency matrix						probability matrix					
123456	1	2	3	4	5	6	1	2	3	4	5	6
TATAAT	1	2	3	4	5	6	A	0.0	0.8	0.4	0.8	0.8
TAAAAT	A	0	4	2	4	4	0	0.0	0.0	0.0	0.0	0.0
TAATAT	C	0	0	0	0	1	0	0.0	0.0	0.0	0.2	0.0
TGTAAT	G	0	1	0	0	0	0	0.0	0.2	0.0	0.0	0.0
TATACT	T	5	0	3	1	0	5	1.0	0.0	0.6	0.2	0.0

$$M_{p,n} = \log_2 \left(\frac{p_{p,n}}{b_n} \right)$$

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

b_n is probability of nucleotide n in background

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	-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

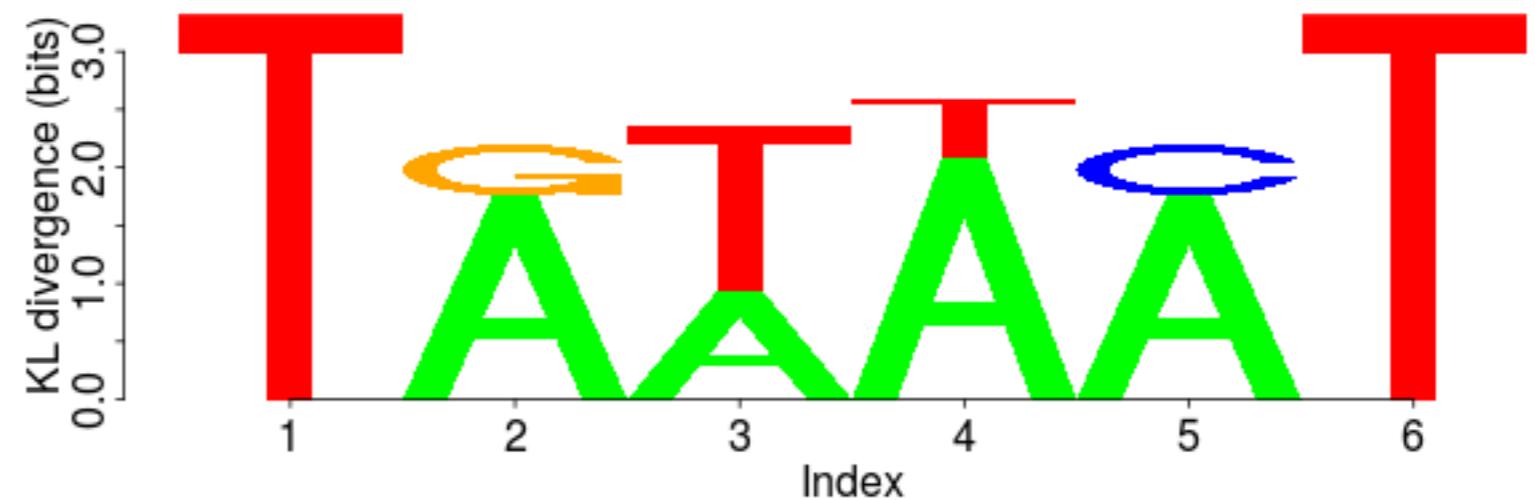
	1	2	3	4	5	6
A	-1.2	1.2	0.4	1.2	1.2	-1.2
C	-1.2	-1.2	-1.2	-1.2	-0.2	-1.2
G	-1.2	-0.2	-1.2	-1.2	-1.2	-1.2
T	1.4	-1.2	0.8	-0.2	-1.2	1.4

PWM visualisation: logo

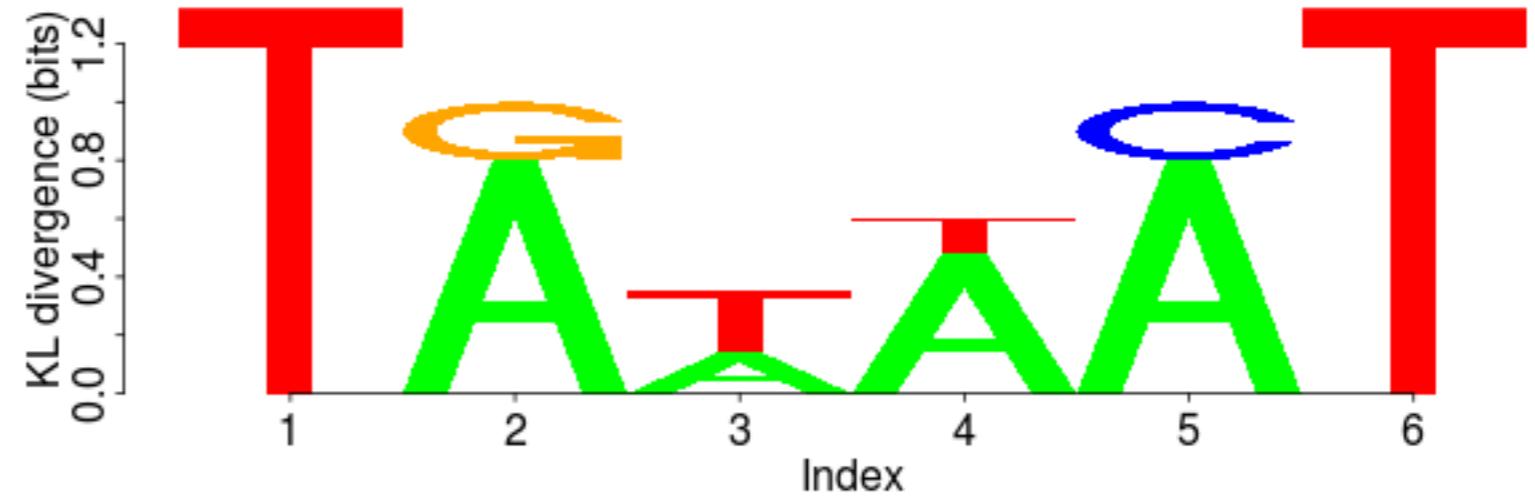
Height of each column is significance of the position (dissimilarity to background).

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

$$Score(tGATAc) = 5.4$$

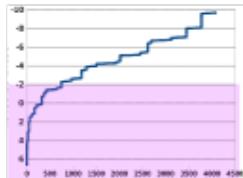
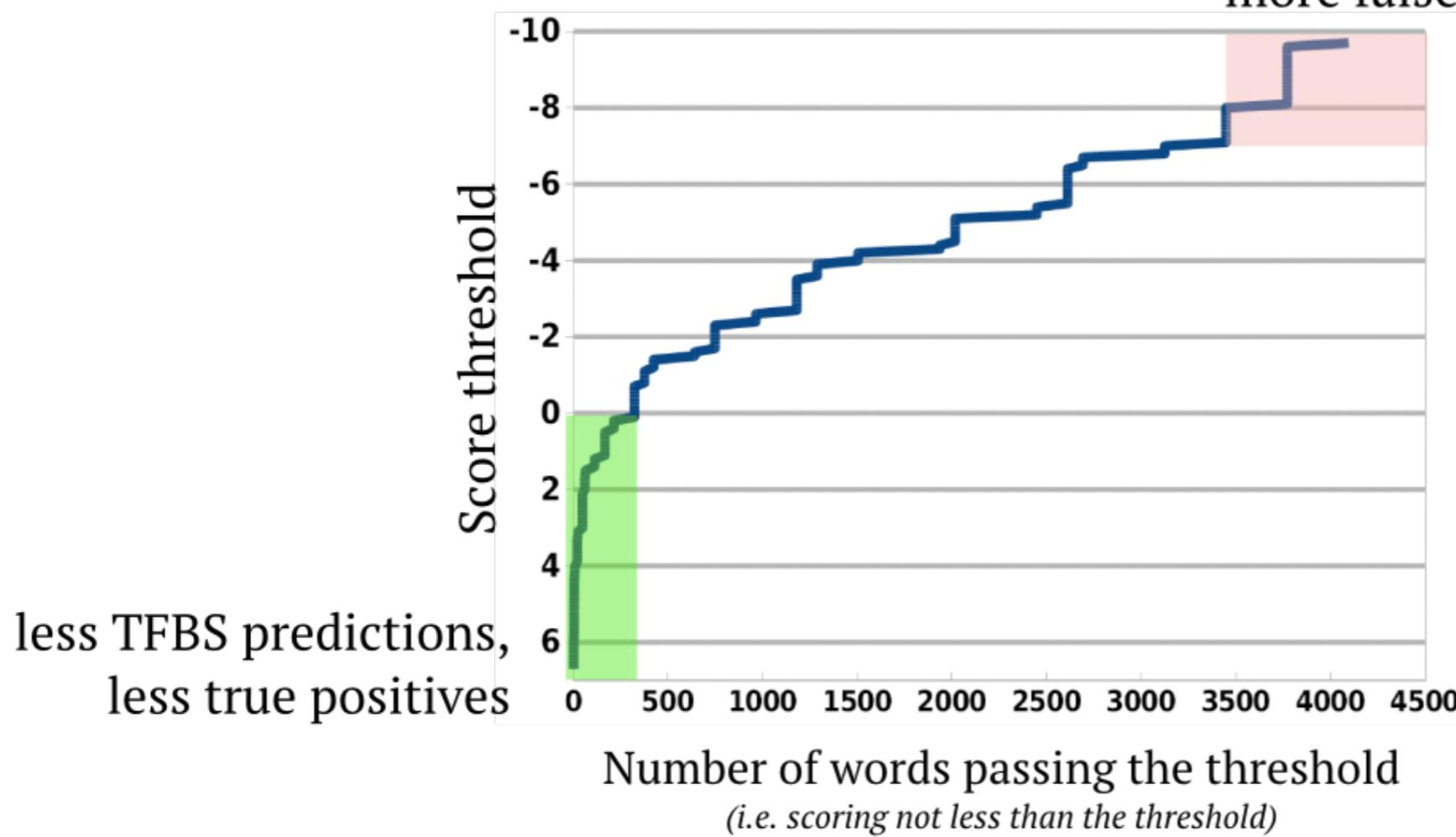
- Additivity assumption: score is larger for longer sequences!

Search of motif instances

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

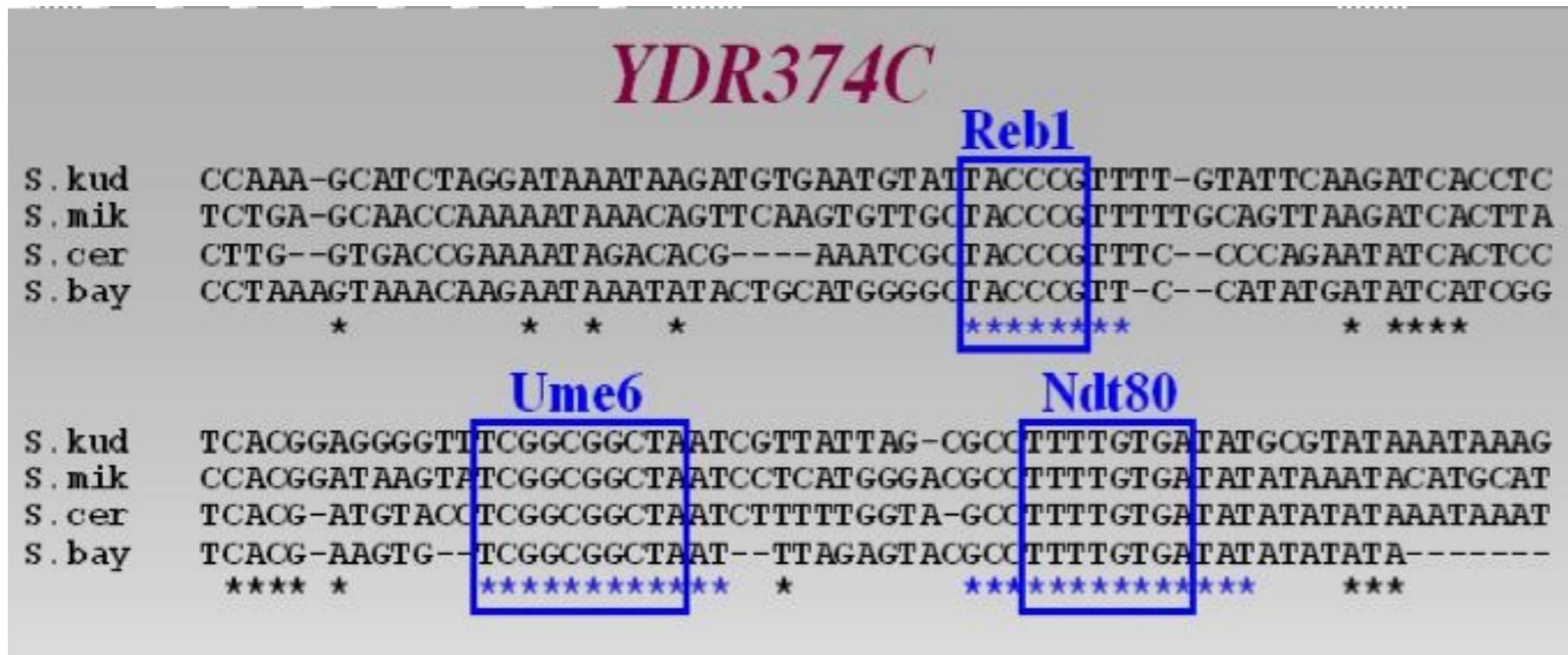
	1	2	3	4	5	6	PWM	
A	-1.6	-1.6	0.96	-1.6	-1.6	0.96	GGATTA	$S_{GGATTA} = 1.22 + 1.22 + 0.96 + 1.22 + 0.96 = \mathbf{6.8}$
C	-1.6	-1.6	0.00	-1.6	-1.6	-1.6		$S_{GGGGGG} = 2.44 - 6.4 = \mathbf{-3.96}$
G	1.22	1.22	-1.6	-1.6	-1.6	-1.6		S=-9.6 the worst score
T	-1.6	-1.6	-1.6	1.22	1.22	0.00		

more predicted TFBS,
more false positive predictions



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

Motifs search: generalized approach



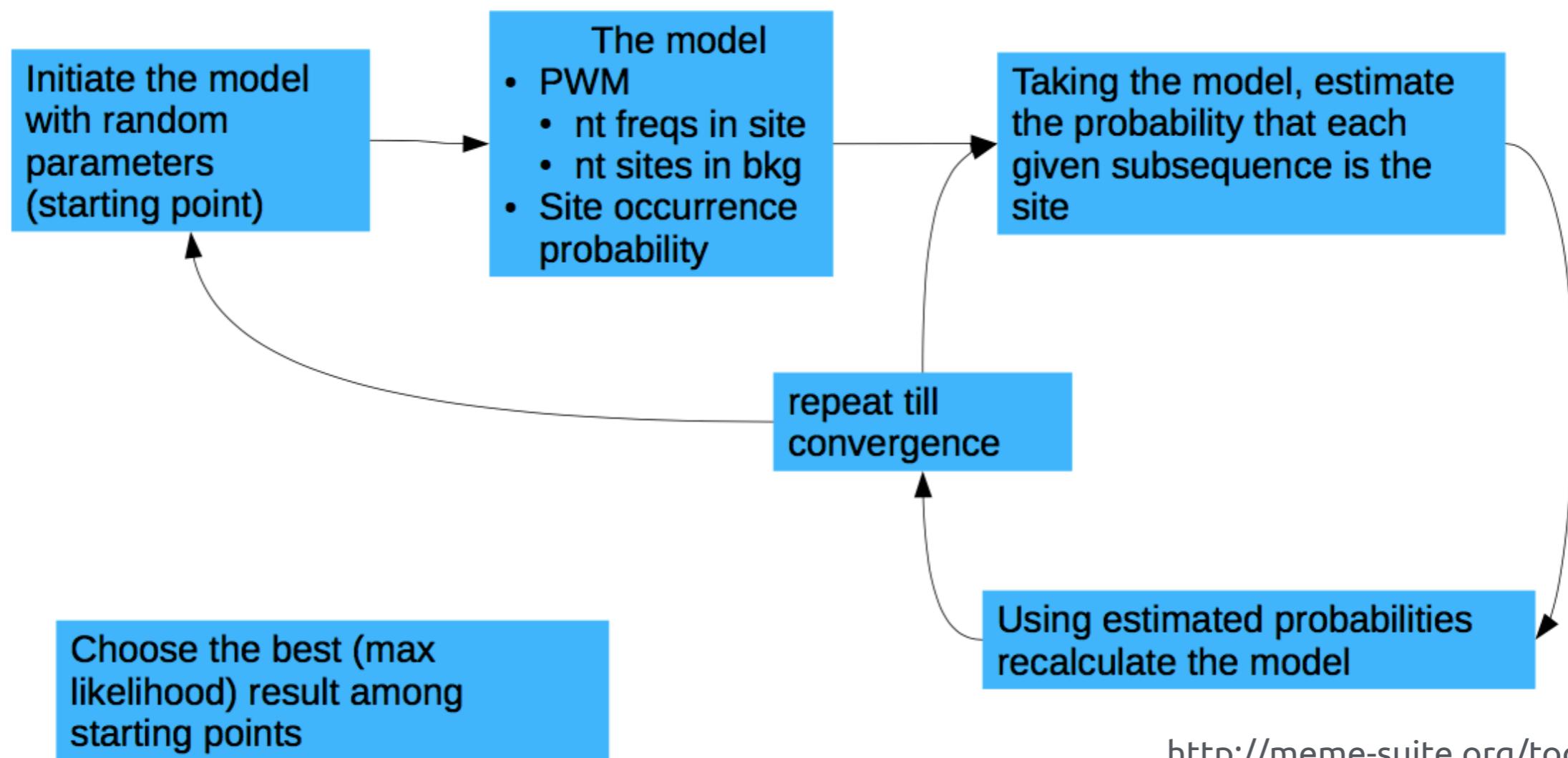
- Count number of occurrence of all possible kmers (or number of sequences that contain given kmer) and compare with random expectation (same count but in shuffled sequences).
- If we have background sequences we can compare kmer occurrence in test and in background set.

MEME

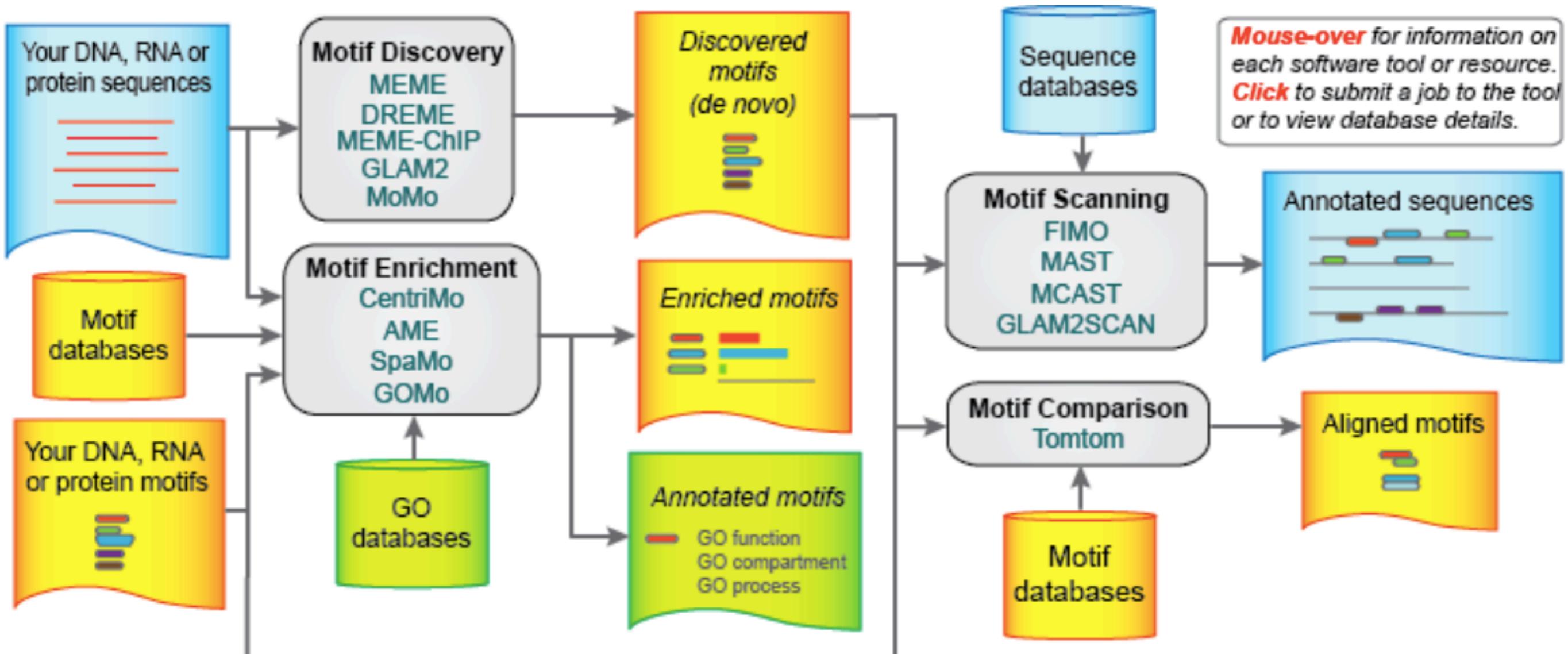
Input is unaligned sequences that can contain 0-n instances of some unknown motif:

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

Algorithm:



MEME-Suite



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 1-3. Multiple alignment

All the materials for this seminar are located on GitHub:

https://github.com/agalitsyna/Teaching_materials_Skoltech-Bioinformatics-2020/

Outline:

1. Download file with upstream regions of bacterial orthologs (`upstreams.fasta`). Create multiple alignment with T-COFFEE, MUSCLE and CLUSTALW. 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/> -> 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 4. Motif search in ChIP-Seq data

1. Download file with peaks sequences from the given **chicken** ChIP-Seq (peaks.fasta).
Find motifs with MEME-ChIP (<http://meme-suite.org/> -> MEME-ChIP).
What was the protein used for ChIP-Seq?
2. Repeat for your peak file assigned to you in Canvas.

About this seminar

- The seminar was inspired by great talk of Ivan Kulakovskiy on the nature of biological interactions.
- For the theoretical part, the seminars of Asya Mendelevich and Pavel Mazin were used as a starting point.