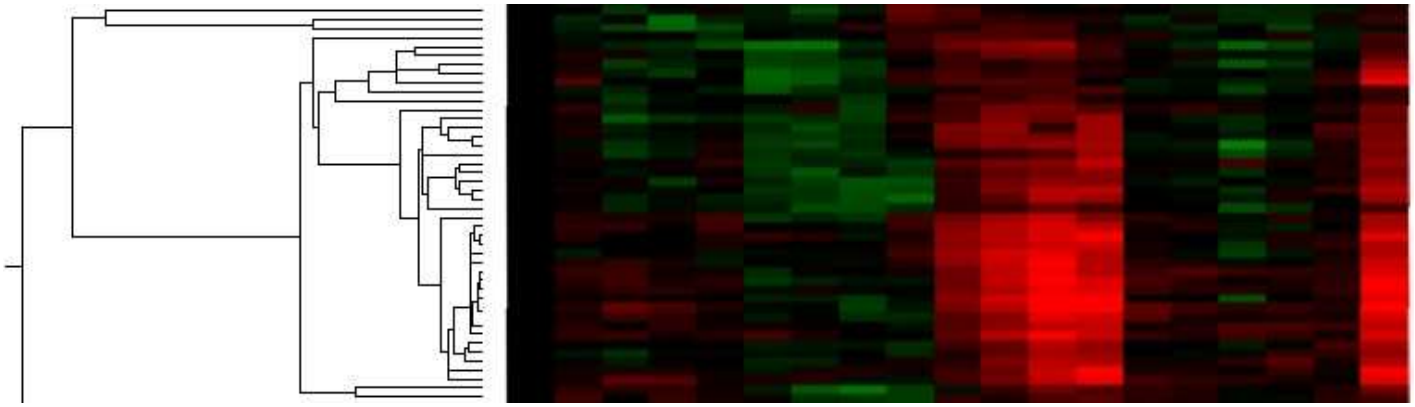
Announcements

- Homework 2 will be published today or tomorrow
- Homework 1 marks will eventually appear in Moodle
- Journal club meetings:
  group 4 done, group 2 agreed date,
  group 5 ??, group 6 looking for date

Regulation of Gene Expression

Broňa Brejová

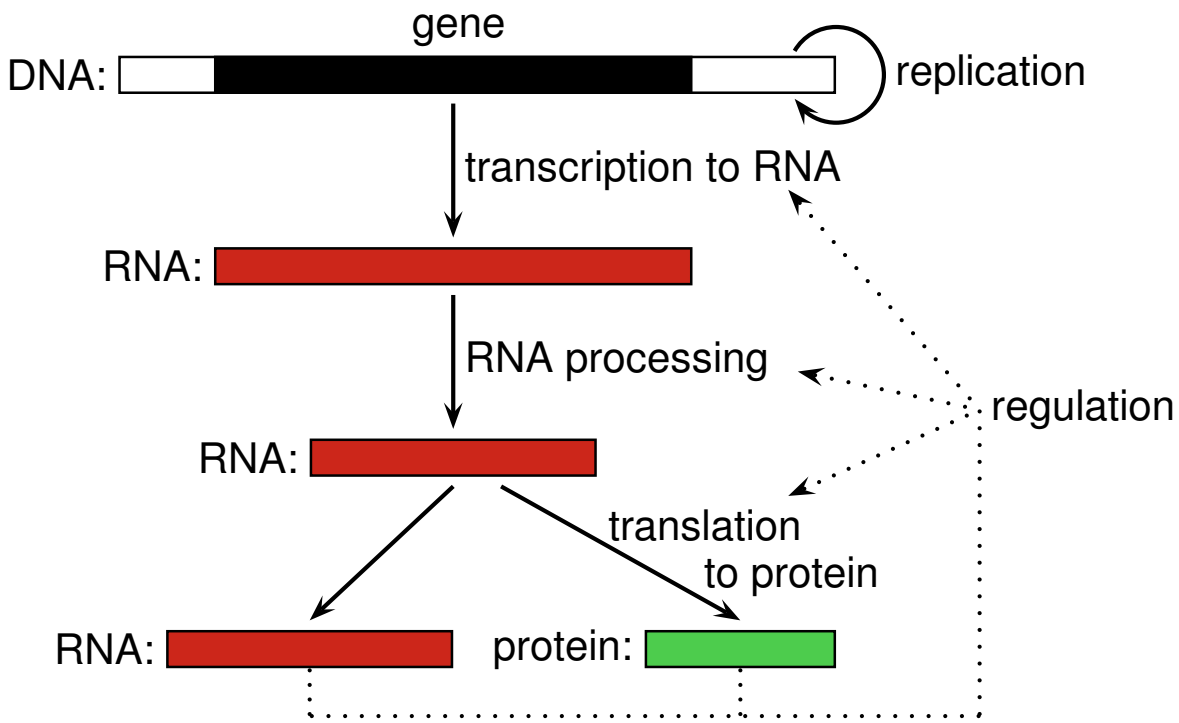
November 11, 2021



Recall: What information is stored in DNA?

Genes: Recipes for synthesis of proteins and functional RNAs.

Regulation of their expression: when and how much to synthesize

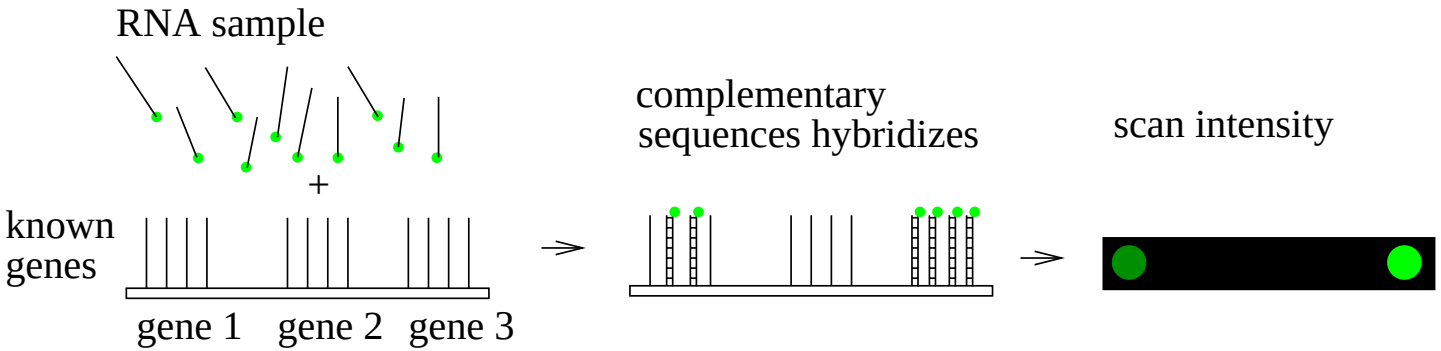


Regulation at the level of transcription, processing, translation, posttranslational modifications, ...

Goals

- Determine under which conditions a gene is expressed (related to gene function)
- Which genes regulate it
- Details of the regulatory mechanism (binding sites, expression levels,. . .)

Technology: expression array, microarray



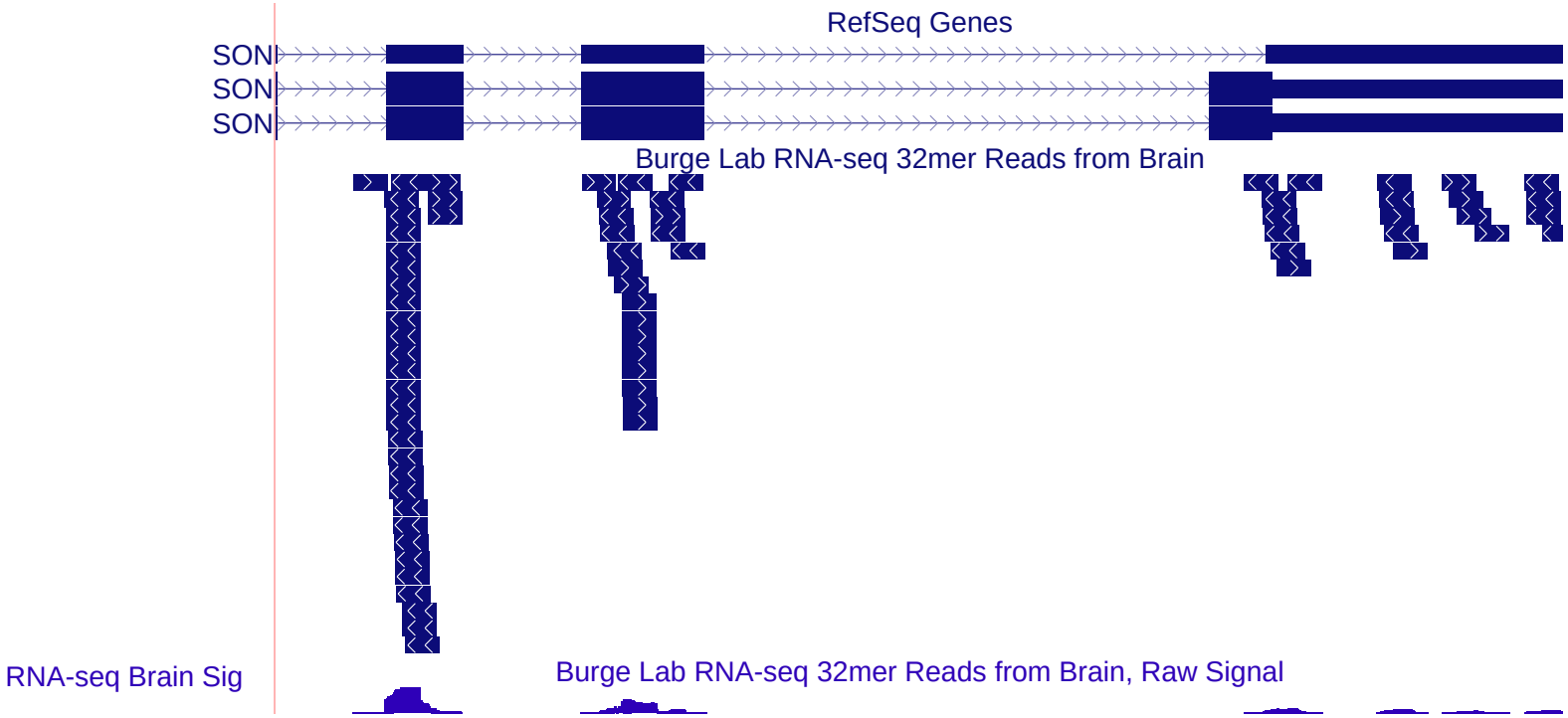
Measuring the amount of mRNA present in the sample for **many genes** at the same time.

Repeated under different conditions.

Technology: RNA-seq

Sequencing RNA extracted from the sample by NGS technologies, mapping reads to the genome.

The depth of coverage corresponds to the expression level



Example from the UCSC genome browser

Example of expression array data

Ratio of gene expression in sample and control fg/bg

	15min	30min	1h	2h	4h	...
W95909	0.72	0.1	0.57	1.08	0.66	
AA045003	1.58	1.05	1.15	1.22	0.54	
AA044605	1.1	0.97	1	0.9	0.67	
W88572	0.97	1	0.85	0.84	0.72	
AA029909	1.21	1.29	1.08	0.89	0.88	
AA059077	1.45	1.44	1.12	1.1	1.15	

...

Iyer et al 1999 The Transcriptional Program in the Response of Human Fibroblasts to Serum

Fibroblasts: cells synthesizing components of extracellular matrix.

To divide, they need growth factors added as "fetal bovine serum".

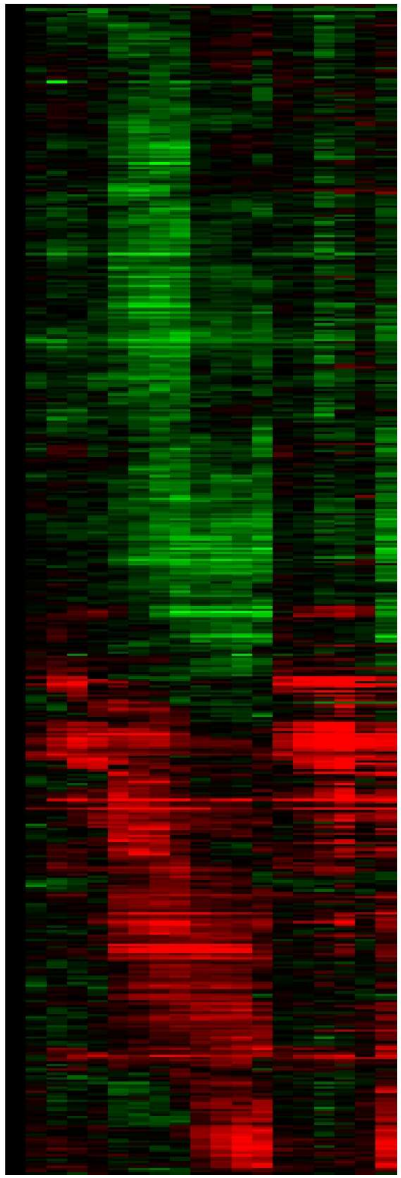
Visualization

Red: fg>bg

Green: fg<bg

517 genes (out of 8600)

19 experiments



This lecture: different type of data

Other lectures in this course: work with sequences

- genome assembly
- sequence alignment
- gene finding
- fylogenetic trees, population and comparative genomics
- structure and function of proteins and RNA

Today: table of numbers

- typical data in statistics
- we can use general methods of statistics and machine learning

The first set of problems: preprocessing data

- Read intensity from microarray images, detect invalid measurements
- Data aggregation from multiple measurements per gene
- Use of control probes
- Normalization to obtain data comparable across experiments

Microarray measurements are very noisy, many sources of errors

A simple result:

list of genes highly underexpressed/overexpressed
e.g. fg/bg> 2, or fg/bg< 0.5
often only these genes used for further analysis

Clustering (zhlukovanie)

Goal: find groups of genes with similar expression profiles.
If many genes in the group have the same function,
the remaining genes may participate as well

Measuring profile similarity: e.g. Pearson correlation coefficient
Profile of gene 1: $x_1, x_2, \ldots, x_n$, mean $\overline{x}$
Profile of gene 2: $y_1, y_2, \ldots, y_n$, mean $\overline{y}$

$$C(x,y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

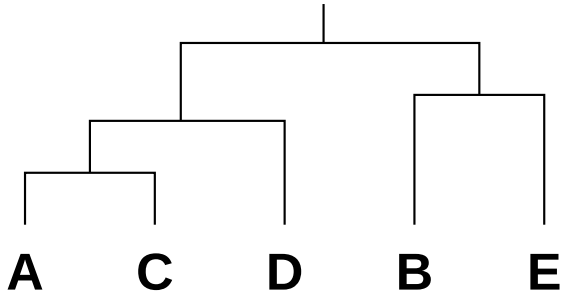
Number between -1 and 1, 1 for linearly correlated data
Distance $d(x,y) = 1 - C(x,y)$

Also other options, e.g. Euclidean distance

Hierarchical clustering

- Similar to neighbor joining method for building phylogenetic trees
- Start with each gene in a separate group
- Find two closest groups and join them to one
- Repeat until all genes are in one group
- Distance of two groups: e.g. distance of closest genes from one and the other group or average of distances over all pairs
- The result is a tree representing hierarchy of clusters

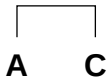
	A	B	C	D	E
gén A	0	0.6	0.1	0.3	0.7
gén B	0.6	0	0.5	0.5	0.4
gén C	0.1	0.5	0	0.6	0.6
gén D	0.3	0.5	0.6	0	0.8
gén E	0.7	0.4	0.6	0.8	0



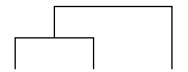
Hierarchical clustering - example

Distance of two groups: distance of closest genes from one and the other group (single linkage clustering)

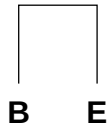
	A	B	C	D	E
gén A	0	0.6	0.1	0.3	0.7
gén B	0.6	0	0.5	0.5	0.4
gén C	0.1	0.5	0	0.6	0.6
gén D	0.3	0.5	0.6	0	0.8
gén E	0.7	0.4	0.6	0.8	0



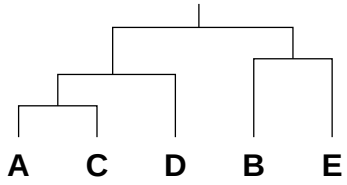
	A+C	B	D	E
A+C	0	0.5	0.3	0.6
B	0.5	0	0.5	0.4
D	0.3	0.5	0	0.8
E	0.6	0.4	0.8	0



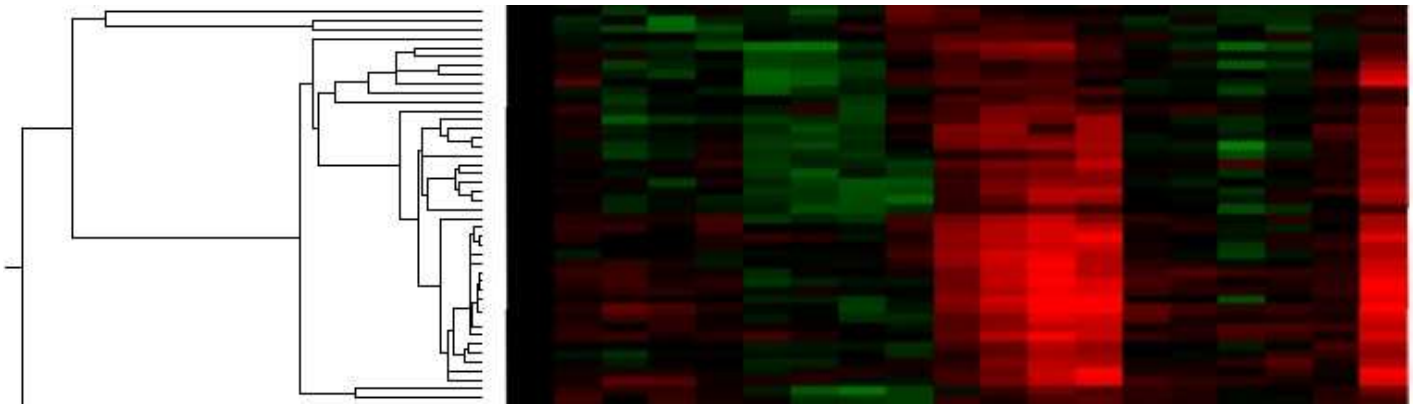
	A+C+D	B	E
A+C+D	0	0.5	0.6
B	0.5	0	0.4
E	0.6	0.4	0



	A+C+D	B+E
A+C+D	0	0.5
B+E	0.5	0

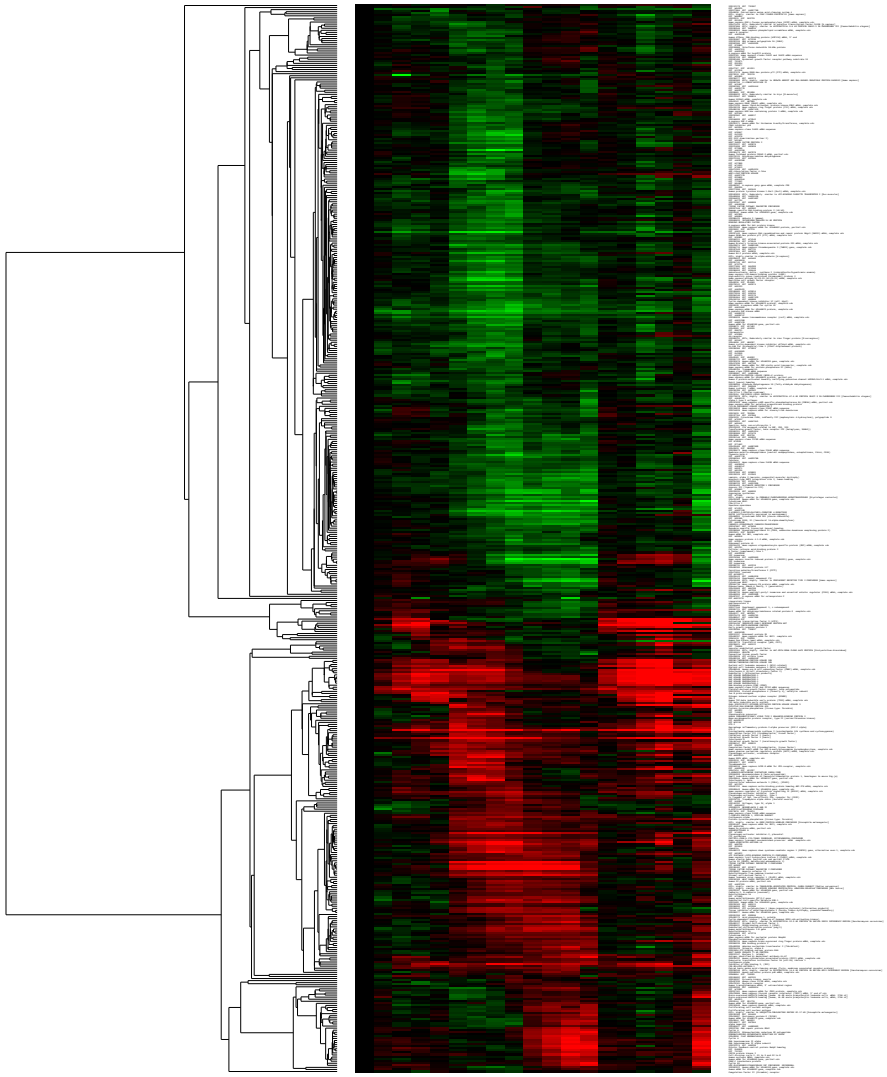


Example: part of the microarray data



Clustering helps to visualize data,

similar genes get close to each other

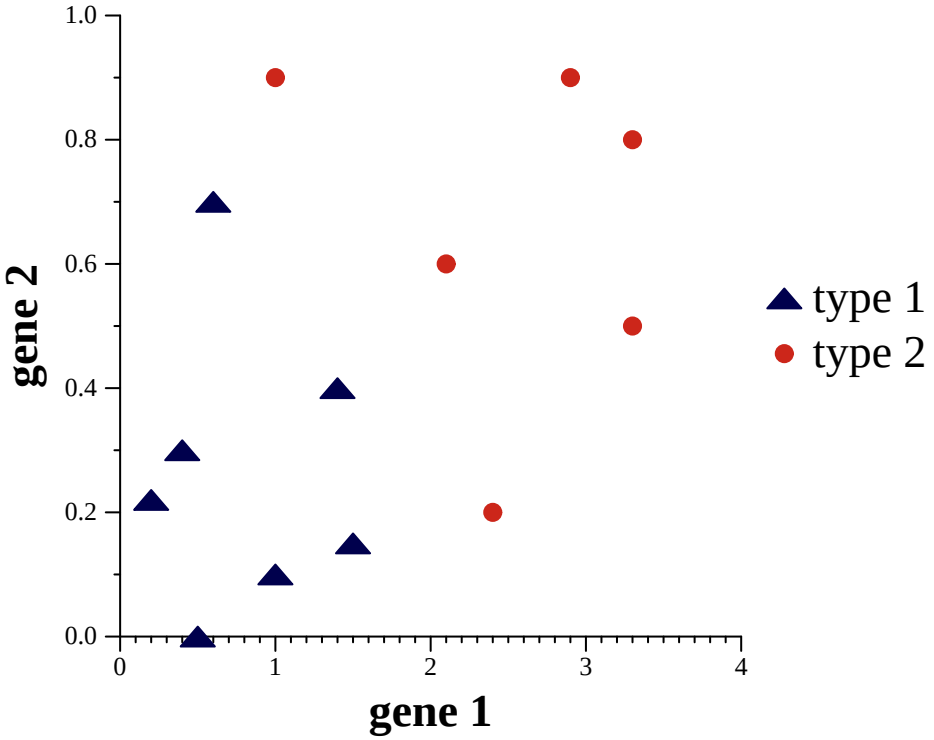


Classification

- A typical machine learning problem
- We might want to for example distinguish different types of tumors according to gene expression
- We are given examples with known expression and tumor type
- We want to find a formula which from the expression produces positive number for tumor type 1 and negative number for type 2
- We choose a family of functions with unknown parameters (hypothesis class)
- Find parameters that give the best accurracy on training data
- Accuracy of the resulting classifier tested on testing data (not used for training)
- The classifier then used on expression data with unknown tumor type

Toy example: expression of 2 genes

Training data with a known type:



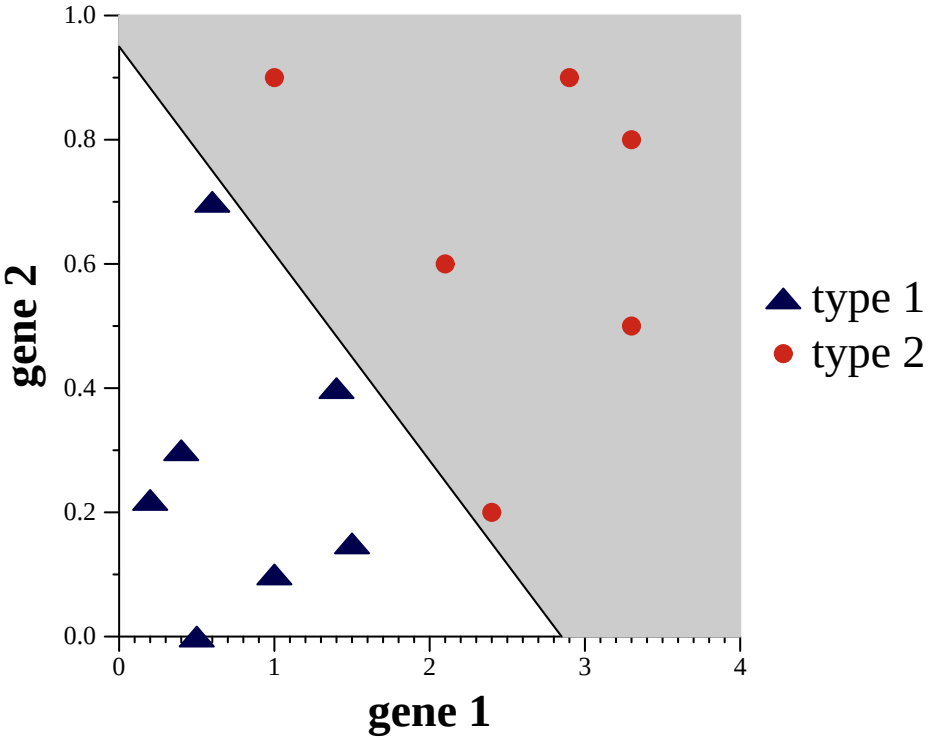
Hypothesis class: linear functions (linear discriminant)

type 1 tumor if $ax + by + c < 0$

The goal is to find a, b, c that work well on training data

Toy example: expression of 2 genes

Resulting classifier:



$a = 1, b = 3, c = -2.85$

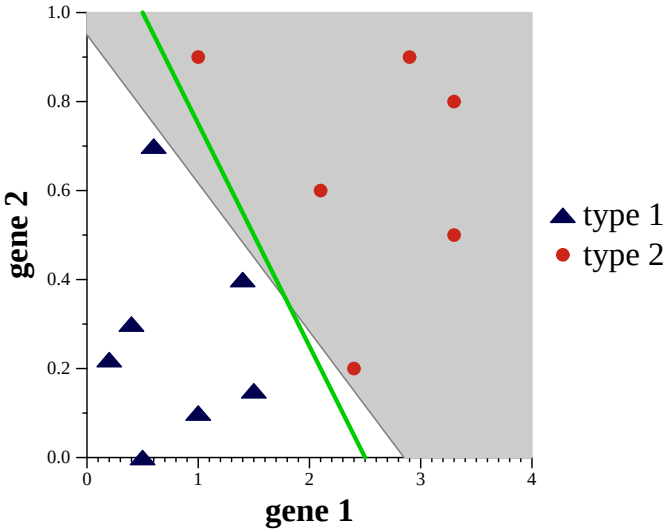
type 1 tumor if $x + 3y - 2.85 < 0$

Popular classification techniques

Logistic regression:

linear discriminant, assigns probability to each class, well-known method from statistics

Support vector machines (SVM): find linear discriminant with no training error which is most distant from all training examples



Can be generalized to non-linear functions by mapping vectors to a higher-dimensional space

Popular classification techniques

Neural networks:

"neurons" connected by "synapses",

output of each neuron is a weighted combination of its inputs

Bayesian networks:

probabilistic model generating random expression profiles

tumor type also a random variable in the model with unknown state

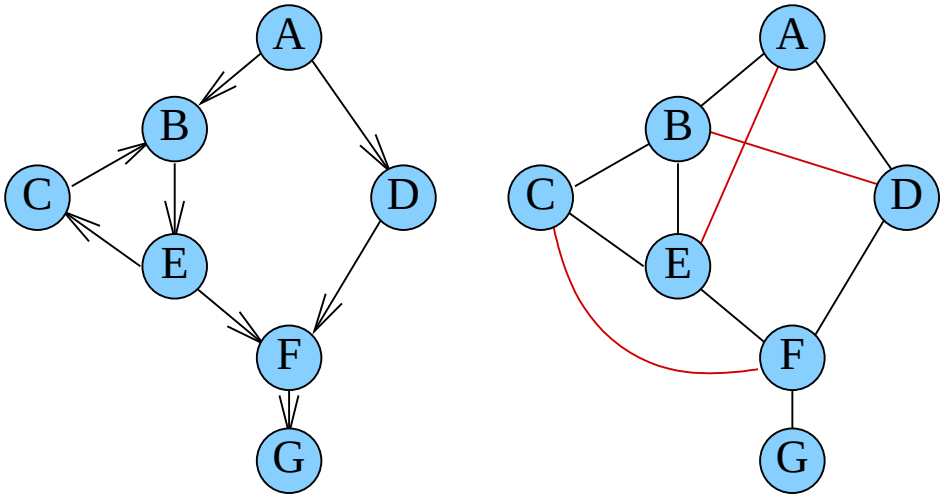
similarly to a state in an HMM

Gene regulation network from expression data

Input: Expression profile for each gene, perhaps under known conditions (time series, deletion mutants)

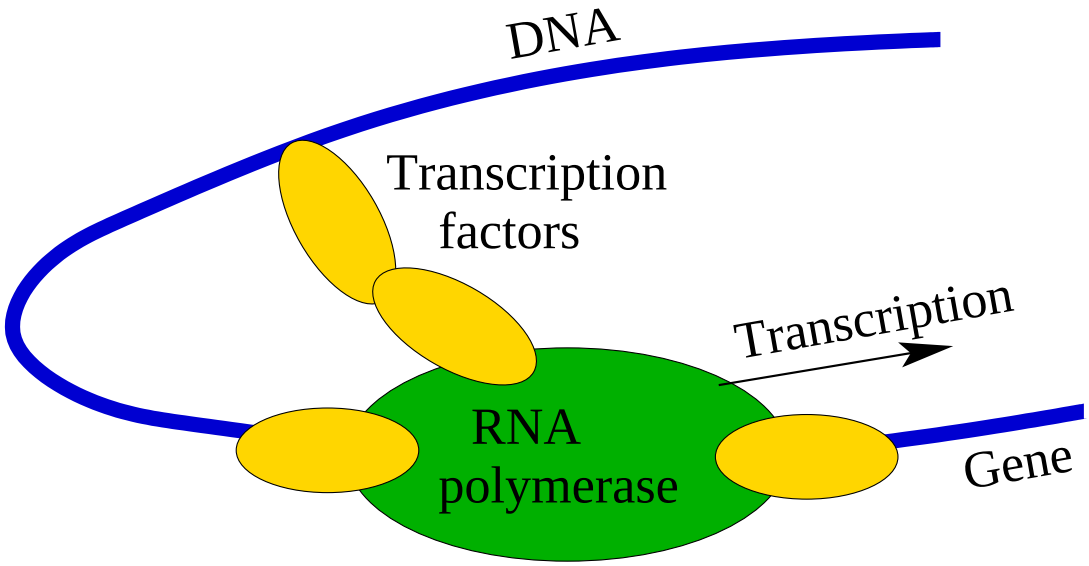
Output: Regulation network; nodes are genes,
directed edge $A \rightarrow B$ if A regulates B

Expression profile similarity may provide undirected edges
The goal is to remove edges resulting from transitivity
and to direct edges correctly (difficult)



Transcription factors (TFs)

Regulation of transcription initiation by transcription factors:
DNA binding proteins which help to attract RNA polymerase



Human genome has over 2000 TFs.

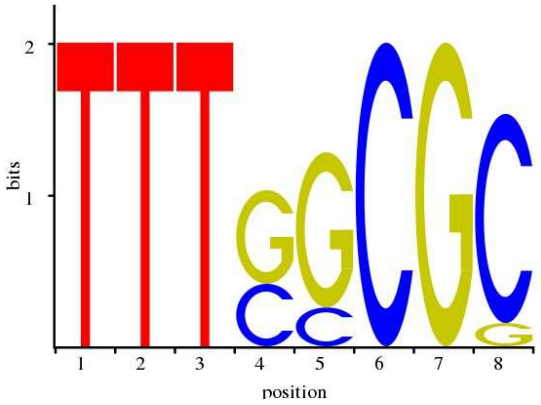
They can increase or decrese expression.

They can work in groups.

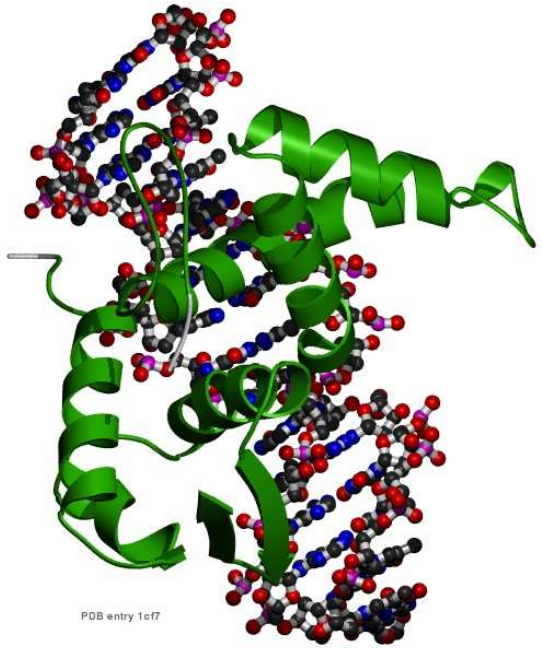
Example: E2F1 transcription factor

- Regulates cell cycle
- Binds TTTCCCGC, TTTCGCGC, and similar variants

A	0	0	0	0	0	0	0	0
C	0	0	0	4	2	10	0	9
G	0	0	0	6	8	0	10	1
T	10	10	10	0	0	0	0	0



- Goal: **represent** DNA sequences bound by a certain TF as a sequence **motif**, then search for **additional occurrences** in the genome



Representation of binding motifs

String with mismatches (consensus):

motif is a string, occurrences can have a certain number of mismatches given in advance

Example: motif TTTGGCGC + 1 mismatch

TTTGGCGC, TT**A**GGCGC, TTTG**C**CGC are motif occurrences

TTT**CC**CGC not an occurrence

Choosing motif: take the most frequent letter at each position

A	0	0	0	0	0	0	0	0
C	0	0	0	4	2	10	0	9
G	0	0	0	6	8	0	10	1
T	10	10	10	0	0	0	0	0

Representation of binding motifs 2

Regular expression:

some positions specify character sets

[GC] means position where C or G is allowed

N means any base

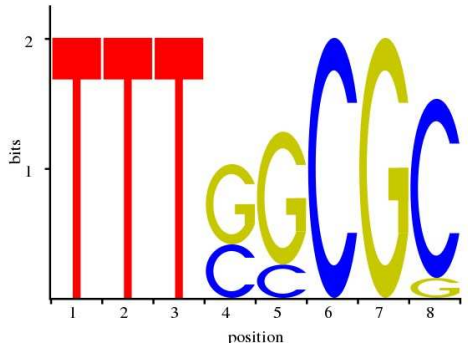
Example: motif TTT[CG][CG]CGC

TTTGGCGC, TTT**CC**CGC, TTTG**C**CGC are motif occurrences

TT**A**GGCGC is not an occurrence

Choosing motif: allow several most frequent letters at each position

A	0	0	0	0	0	0	0	0
C	0	0	0	4	2	10	0	9
G	0	0	0	6	8	0	10	1
T	10	10	10	0	0	0	0	0



Representation of binding motifs 3

Position specific scoring matrix (PSSM, PWM):

scoring matrix, score for each letter at each position

occurrences achieve score higher than threshold T

Example: $T = 8$

A	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0
C	-1.6	-1.6	-1.6	0.6	0.0	1.5	-1.6	1.4
G	-1.6	-1.6	-1.6	1.0	1.3	-1.6	1.5	-0.5
T	1.1	1.1	1.1	-2.0	-2.0	-2.0	-2.0	-2.0

TTT**CC**CGC is an occurrence: 1.1+1.1+1.1+0.6+0.0+1.5+1.5+1.4=8.3

TTTGGCG**G** is an occurrence: 1.1+1.1+1.1+1.0+1.3+1.5+1.5-0.5=8.1

TT**A**GGCGC is not: 1.1+1.1-2.0+1.0+1.3+1.5+1.5+1.4=6.4

Construction of PSSM: next lecture

Finding occurrences in the genome

- Consider motif in one of the representations:
  - Consensus, e.g. TTTGGCGC + 1 mismatch
  - Regular expression, e.g. TTT[CG][CG]CGC
  - Scoring matrix, e.g. threshold $T = 8$ and matrix:

```
A  -2.0  -2.0  -2.0 -2.0  -2.0  -2.0  -2.0  -2.0
C  -1.6  -1.6  -1.6  0.6   0.0   1.5  -1.6   1.4
G  -1.6  -1.6  -1.6  1.0   1.3  -1.6   1.5  -0.5
T   1.1   1.1   1.1 -2.0  -2.0  -2.0  -2.0  -2.0
```

- Test each position in the genome if it is an occurrence
- Occurrences are potential binding sites

Finding occurrences in the genome: problem

- Test each position in the genome if it is a motif occurrence
- Besides **binding sites**, often also many **random occurrences**
- E-value of a motif: how many occurrences are expected in a random sequence
- For example TTT[CG][CG]CGC appears about once in 30,000 bases
- To improve specificity, we can search for
  - clusters of binding sites
  - sites validated by experiments
  - evolutionarily conserved sites
- Motif databases, e.g. TRANSFAC, JASPAR

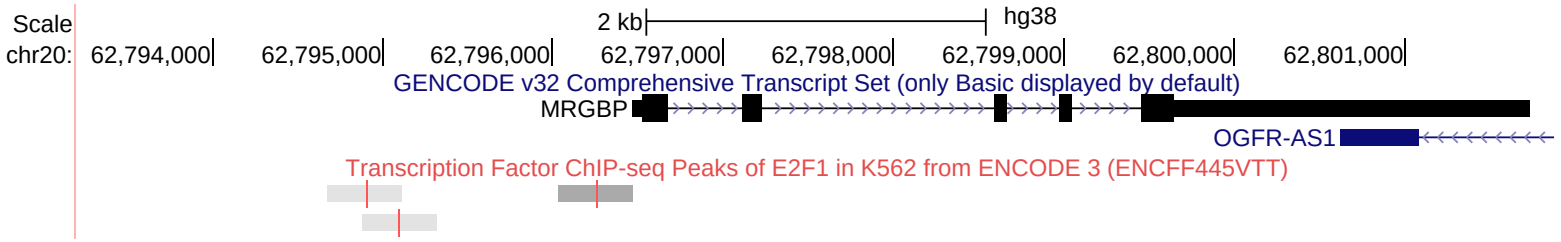
How to find binding sites experimentally?

Chromatin immunoprecipitation (ChIP)

Using an antibody specific to a given TF, we can determine approximate locations of its binding sites

- TF and DNA crosslinked by formaldehyde
- DNA cut to shorter segments
- Segments with crosslinked TF are bound by the antibody
- DNA is isolated and sequenced (**ChIP-seq**)

Problem: we find only approximate location of the binding site



How to find motifs by computational methods?

. . . without having several examples of a binding site

- Assume we have a group of sequences, each containing a binding site of the same TF, but binding preferences of this TF not known
- The goal is to find **the most specific** motif, occurring in all sequences or occurring more frequently than expected
- **Currently:** using ChIP-seq obtain regions of DNA surrounding binding sites, find motifs to refine the binding site position
- **Originally:** take a group of genes with similar expression profiles, thus possibly regulated by the same TF
  find motifs in DNA regions upstream of these genes

Consensus Pattern Problem (CPP)

Simple formulation of the motif finding problem

Input: motif length L, sequences $S_1, S_2, \ldots, S_k$

Output: motif (string) M of length L
and motif occurrence in each S_i (string s_i of length L)
such that the overall number of mismatches between M and s_i) is smallest possible

Example:

Input: CAAACAT, AGTAGC, TAACCA, TCTCCTC, $L = 4$

Output: motif TAAC

Occurrences and mismatches AAAC 1, TAGC 1, TAAC 0, TCTC 2

Total mismatches 4

Solving CPP

NP-hard problem

- **Idea 1:** Try all possible motifs of length L

  Problem: Not practical — why?

- **Idea 2:** Try all substrings of length L
  of input strings $S_1, \ldots, S_k$

  Problem: Sometimes gives wrong answer — why?
  But this always finds a solution
  with cost at most twice the optimum
  (2-approximation algorithm)

- **Further improvements:**
  Try consensus sequences
  of all samples of r substrings from input
  PTAS (polynomial-time approximation scheme)

Input: $L = 4$
CAAACAT,
AGTAGC,
TAACCA,
TCTCCTC

Output:
motif TAAC
Occurrences
and mismatches:
AAAC 1,
TAGC 1,
TAAC 0,
TCTC 2
Total mismatches 4

A more practical approach to motif finding

Probabilistc model generating sequence S
using matrix W of base frequences in the motif
and background frequences q outside the motif

```
A 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
C 0.01 0.01 0.01 0.39 0.19 0.97 0.01 0.01 0.89
G 0.01 0.01 0.01 0.59 0.79 0.01 0.97 0.97 0.09
T 0.97 0.97 0.97 0.01 0.01 0.01 0.01 0.01 0.01
```

$q(A) = 0.3$, $q(C) = 0.2$, $q(G) = 0.2$, $q(T) = 0.3$

Motif position in S is chosen randomly and each base is then generated according to q or one column of W

This model defines $\Pr(S \mid W)$.

Motif finding based on probabilistic models

Input: motif length L, sequences $S_1, \ldots, S_k$, frequencies q

Output: motif as a frequence matrix M maximizing likelihood $\Pr(S_1|W) \cdot \ldots \cdot \Pr(S_k|W)$

- Hard problem, addressed by heuristic algorithms
- For example EM (expectation maximalization)
- Local optimization, converging to a local maximum of likelihood
- Software: MEME

EM algorithm overview

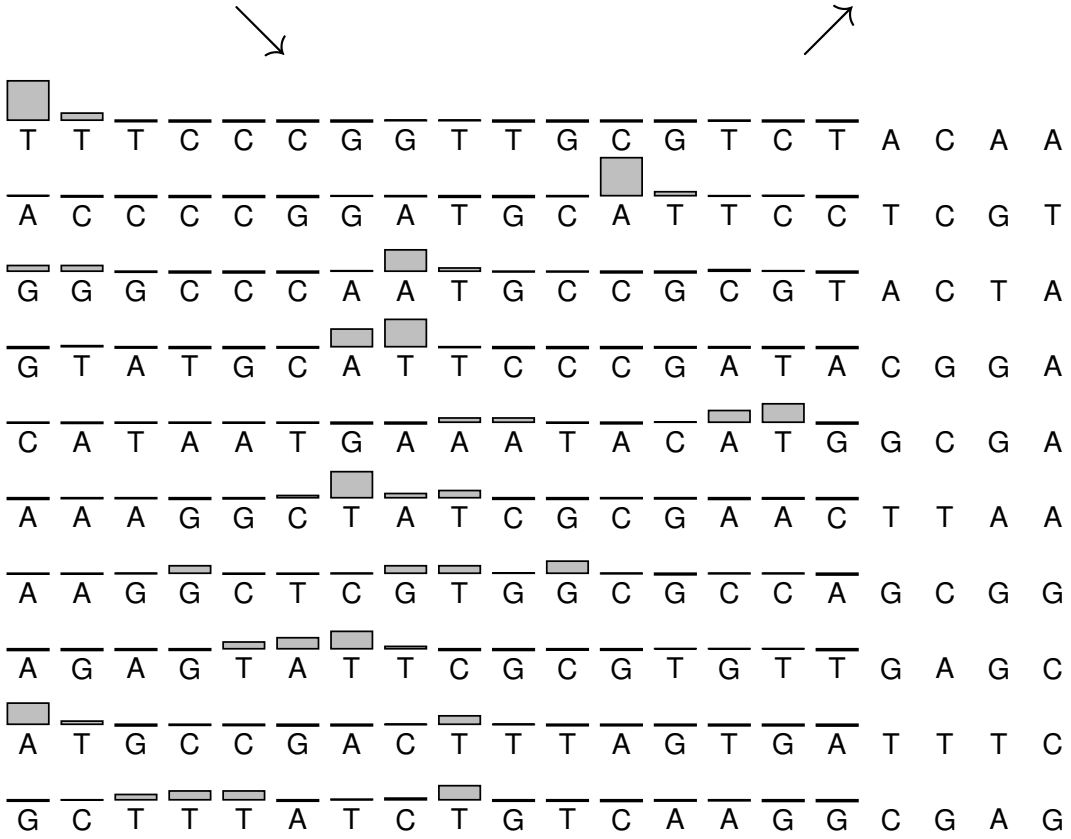
- **Initialization:**
  Choose initial matrix W
  (e.g. based on one input substring of length L)
- **Iteration:**
  1. Assign each position j in sequence S_i weight $p_{i,j}$ corresponding to probability that $S_i[j]$ is a start of the motif W.
  2. Compute W from all possible occurrences in $S_1, \ldots, S_k$ weighted by $p_{i,j}$

Iterations increase likelihood until convergence.

Repeat, starting from many different starting values W

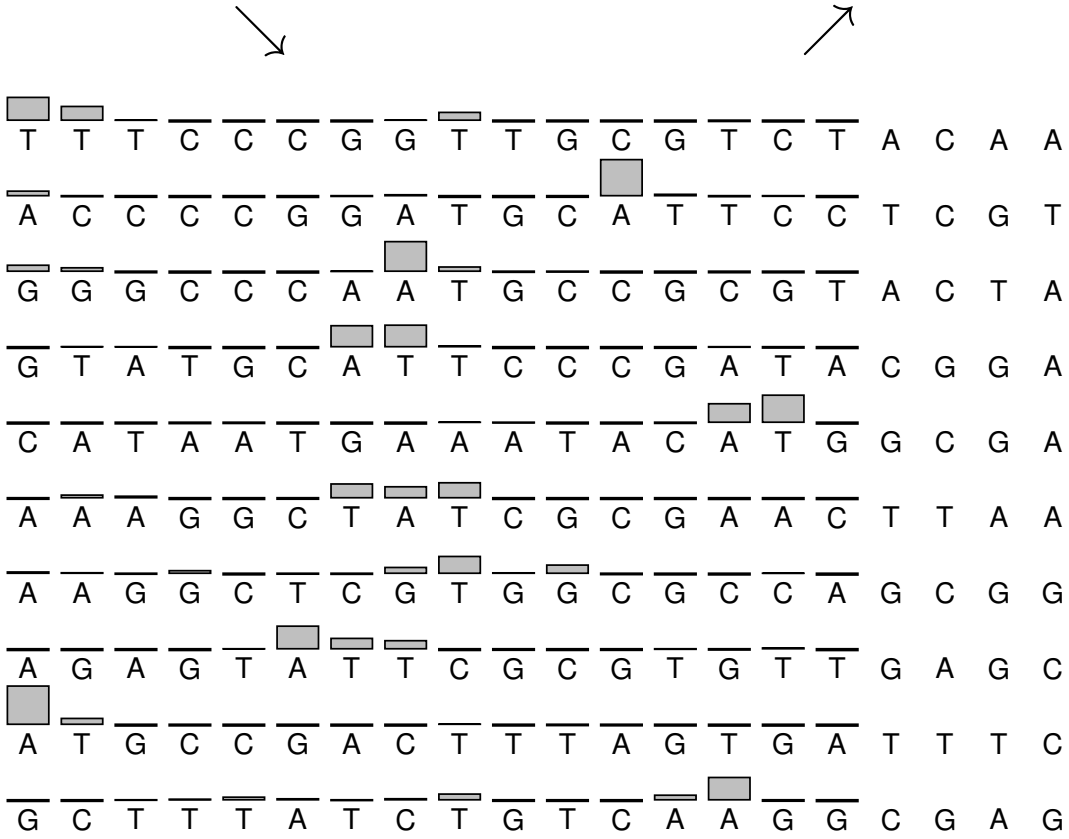
Example of the EM algorithm

A	0.10	0.10	0.10	0.10	0.10	A	0.31	0.14	0.06	0.07	0.07
C	0.10	0.10	0.10	0.70	0.70	C	0.06	0.10	0.19	0.71	0.61
G	0.10	0.10	0.10	0.10	0.10	G	0.12	0.17	0.29	0.14	0.25
T	0.70	0.70	0.70	0.10	0.10	T	0.51	0.60	0.46	0.08	0.07



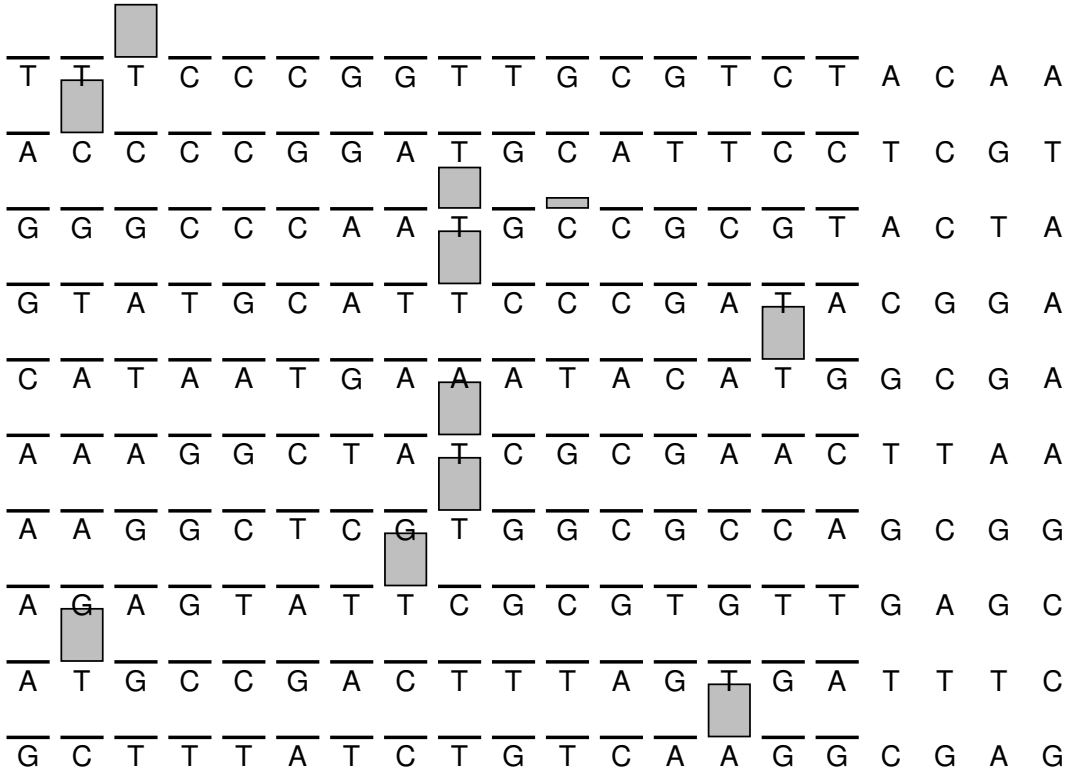
Example of the EM algorithm: next iteration

A	0.31	0.14	0.06	0.07	0.07	A	0.47	0.09	0.01	0.02	0.03
C	0.06	0.10	0.19	0.71	0.61	C	0.02	0.11	0.20	0.80	0.58
G	0.12	0.17	0.29	0.14	0.25	G	0.08	0.22	0.48	0.15	0.35
T	0.51	0.60	0.46	0.08	0.07	T	0.42	0.58	0.30	0.03	0.03



Example of the EM algorithm: after 20 iterations

A	0.10	ϵ	ϵ	ϵ	ϵ
C	0.12	0.52	0.48	$1-3\epsilon$	ϵ
G	ϵ	0.48	0.52	ϵ	$1-3\epsilon$
T	0.78	ϵ	ϵ	ϵ	ϵ



Summary

- Microarrays or RNA-seq can characterize expression levels of many genes at once, but produce noisy data
- Clustering (zhlukovanie) can find similar genes
  no prior training set is necessary (unsupervised learning)
- Classification can distinguish e.g. diseases according to expression
  needs training data with known answers (supervised learning)
- Expression data help to build regulatory networks
- Binding motifs can be represented in various forms
  (string, regular expression, scoring matrix)
- These motifs are not sufficiently specific, therefore it is hard to recognize binding sites in the genome
- EM algorithm for finding new motifs in sequences