# Basic concepts in genome analysis

VELI MÄKINEN

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF HELSINKI

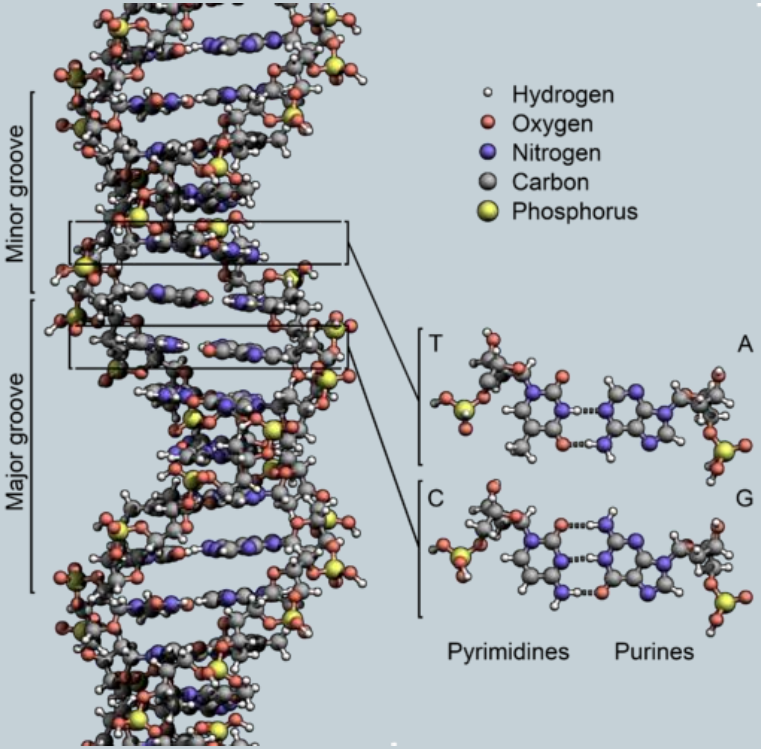
# History

- This material is based on the lecture slides for the textbook
  - R. C. Deonier, M. S. Waterman, S. Tavaré. Computational Genome Analysis: An Introduction. Springer, 2005.
- The material has evolved through several courses given at the University of Helsinki, with the first edition prepared by Esa Pitkänen for the Introduction to Bioinformatics course.

# Part I

## SOME CONCEPTS OF MOLECULAR BIOLOGY

# DNA



Source: Wikipedia

# One slide recap of molecular biology

Nucleotides A, C, G, T

gene

DNA

…TACCTACATCTACACATC…AGCTACGTTCCCCGACTACGACATGGTGATT

5’ …ATGGATGTAGATGTGTAG…TCGATGCAAGGGGCTGATGCTGTACCACTAA… 3’

exon

intron

exon

transcription

mRNA

codon

…AUGGAUGUAGAUGGGCUGAUGCUGUACCACUAA

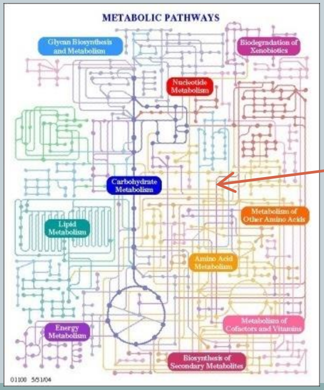
translation

Protein

MDVDGLMLYH

Gene regulation

entsyme


METABOLIC PATHWAYS
Glycan Biosynthesis and Metabolism
Biodegradation of Xenobiotics
Nucleotide Metabolism
Carbohydrate Metabolism
Lipid Metabolism
Metabolism of Other Amino Acids
Amino Acid Metabolism
Energy Metabolism
Metabolism of Cofactors and Vitamins
Biosynthesis of Secondary Metabolites
01100 5/31/04

recombination

Mother DNA

AGCTAGGCTAGC
AGCCAGGATCGC

Father DNA

AGTCAGGCTAAC
AGCTAGGCTATC

Child DNA

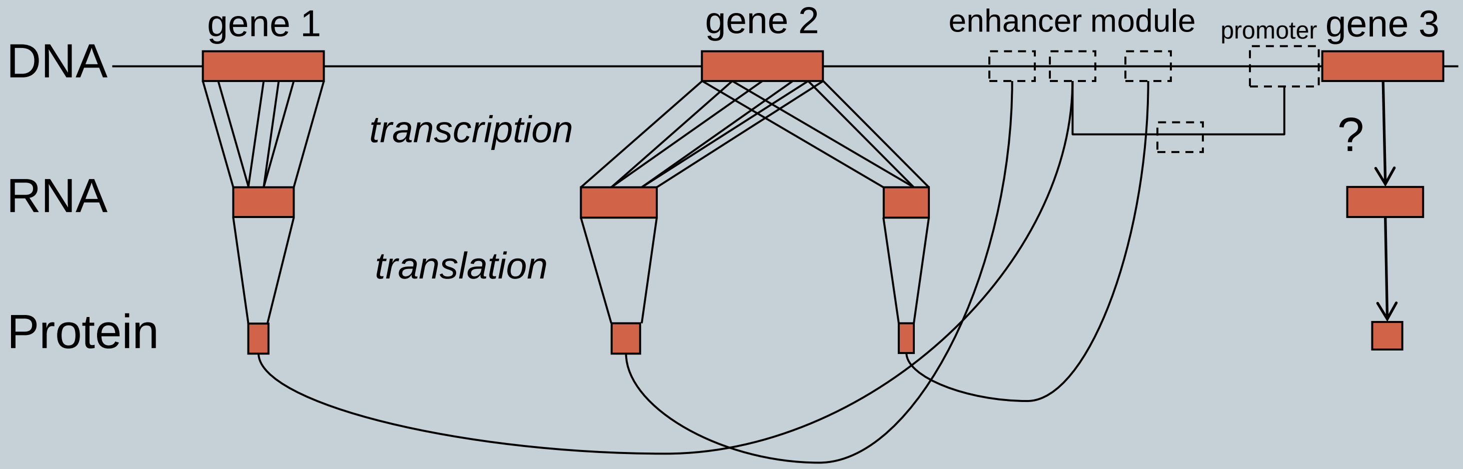
AGCCAGGCTAGC
AGTCAGACTATC

# Part II

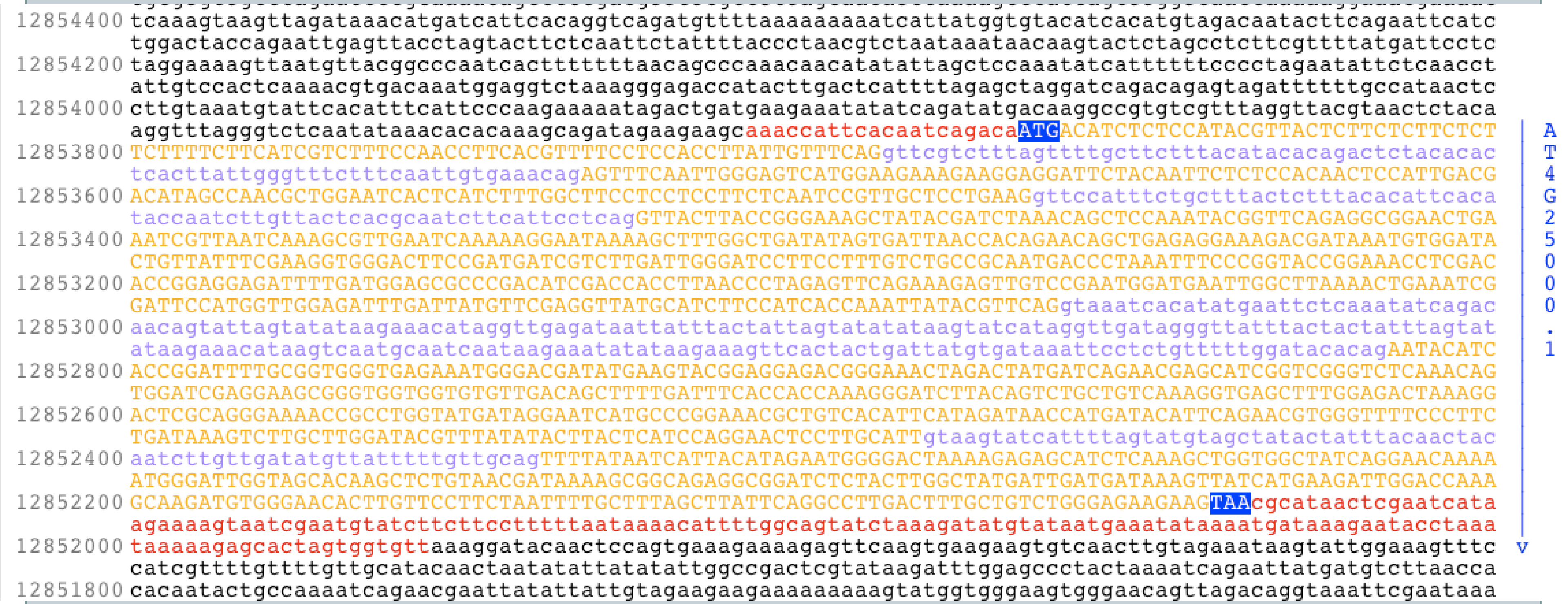
## SIGNALS IN DNA

# Signals in DNA

- Genes
- Promoter regions
- Binding sites for regulatory proteins (*transcription factors, enhancer modules, motifs*)



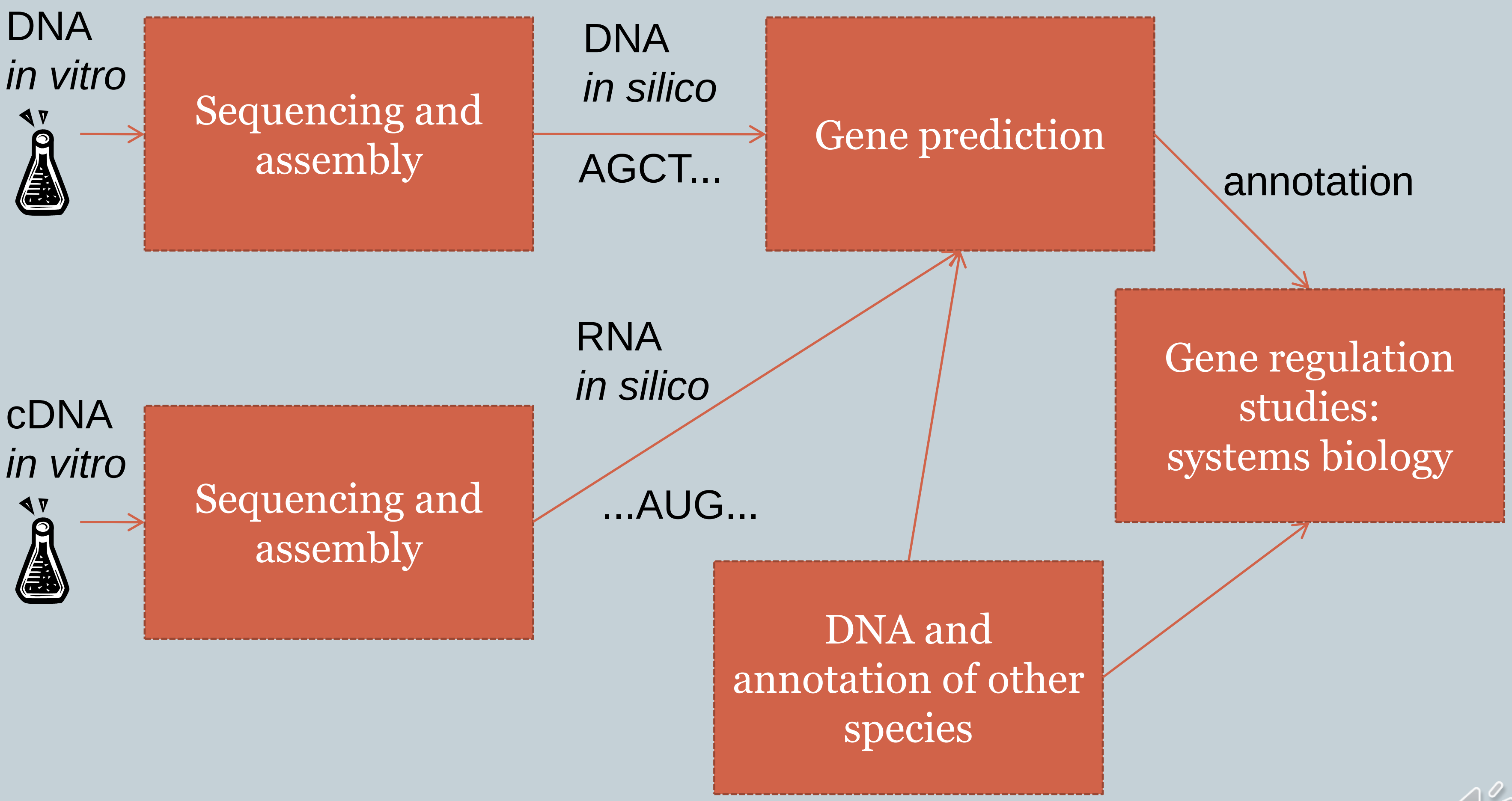
# Typical gene



```
12854400 tcaaagtaagttagataaacatgatcattcacaggtcagatgttttaaaaaaaaatcattatggtgtacatcacatgtagacaatacttcagaattcatc
         tggactaccagaattgagttacctagtacttctcaattctattttaccctaacgtctaataaataacaagtactctagcctcttcgttttatgattcctc
12854200 taggaaaagttaatgttacggcccaatcactttttttaacagcccaaacaacatatattagctccaaatatcatttttttcccctagaatattctcaacct
         attgtccactcaaaacgtgacaaatggaggtctaaagggagaccatacttgactcattttagagctaggatcagacagagtagatttttttgccataactc
12854000 cttgtaaatgtattcacatttcattcccaagaaaaatagactgatgaagaaatatatcagatatgacaaggccgtgtcgtttaggttacgtaactctaca
         aggtttagggtctcaatataaacacacaaagcagatagaagaagcaaaccattcacaatcagacaATGACATCTCTCCATACGTTACTCTTCTCTTCTCT
12853800 TCTTTTCTTCATCGTCTTTCCAACCTTCACGTTTTCCTCCACCTTATTGTTTCAGgttcgtctttagttttgcttctttacatacacagactctacacac
         tcacttattgggtttctttcaattgtgaaacagAGTTTCAATTGGGAGTCATGGAAGAAAGAAGGAGGATTCTACAATTCTCTCCACAACTCCATTGACG
12853600 ACATAGCCAACGCTGGAATCACTCATCTTTGGCTTCCTCCTCCTTCTCAATCCGTTGCTCCTGAAGgttccatttctgctttactctttacacattcaca
         taccaatcttgttactcacgcaatcttcattcctcagGTTACTTACCGGGAAAGCTATACGATCTAAACAGCTCCAAATACGGTTCAGAGGCGGAACTGA
12853400 AATCGTTAATCAAAGCGTTGAATCAAAAAGGAATAAAAGCTTTGGCTGATATAGTGATTAACCACAGAACAGCTGAGAGGAAAGACGATAAATGTGGATA
         CTGTTATTTCGAAGGTGGACTTCCGATGATCGTCTTGATTGGGATCCTTCCTTTGTCTGCCGCAATGACCCTAAATTTCCCGGTACCGGAAACCTCGAC
12853200 ACCGGAGGAGATTTTGATGGAGCGCCCGACATCGACCACCTTAACCCTAGAGTTCAGAAAGAGTTGTCCGAATGGATGAATTGGCTTAAAACTGAAATCG
         GATTCCATGGTTGGAGATTTGATTATGTTCGAGGTTATGCATCTTCCATCACCAAATTATACGTTCAGgtaaatcacatatgaattctcaaatatcagac
12853000 aacagtattagtatataagaaacataggttgagataattatttactattagtatatataagtatcataggttgatagggttatttactactatttagtat
         ataagaaacataagtcaatgcaatcaataagaaatatataagaaagttcactactgattatgtgataaattcctctgtttttggatacacagAATACATC
12852800 ACCGGATTTTGCGGTGGGTGAGAAATGGGACGATATGAAGTACGGAGGAGACGGGAAACTAGACTATGATCAGAACGAGCATCGGTCGGGTCTCAAACAG
         TGGATCGAGGAAGCGGGTGGTGGTGTGTTGACAGCTTTTGATTTCACCACCAAAGGATCTTACAGTCTGCTGTCAAAGGTGAGCTTTGGAGACTAAAGG
12852600 ACTCGCAGGGAAAACCGCCTGGTATGATAGGAATCATGCCCGGAAACGCTGTCACATTCATAGATAACCATGATACATTCAGAACGTGGGTTTTCCCTTC
         TGATAAAGTCTTGCTTGGATACGTTTATATACTTACTCATCCAGGAACTCCTTGCATTgtaagtatcattttagtatgtagctatactatttacaactac
12852400 aatcttgttgatatgttatttttgttgcagTTTTATAATCATTACATAGAATGGGGACTAAAAGAGAGCATCTCAAAGCTGGTGGCTATCAGGAACAAAA
         ATGGGATTGGTAGCACAAGCTCTGTAACGATAAAAGCGGCAGAGGCGGATCTCTACTTGGCTATGATTGATGATAAAGTTATCATGAAGATTGGACCAAA
12852200 GCAAGATGTGGGAACACTTGTTCCTTCTAATTTTGCTTTAGCTTATTCAGGCCTTGACTTTGCTGTCTGGGAGAAGAAGTAAcgcataactcgaatcata
         agaaaagtaatcgaatgtatcttcttcctttttaataaaacattttggcagtatctaaagatatgtataatgaaatataaaatgataaagaatacctaaa
12852000 taaaaagagcactagtggtgttaaaggatacaactccagtgaaagaaaagagttcaagtgaagaagtgtcaacttgtagaaataagtattggaaagtttc
         catcgttttgttttgttgcatacaactaatatattatatattggccgactcgtataagatttggagccctactaaaatcagaattatgatgtcttaacca
12851800 cacaatactgccaaaatcagaacgaattatattattgtagaagaagaaaaaaaaagtatggtgggaagtgggaacagttagacaggtaaattcgaataaa
```

http://en.wikipedia.org/wiki/File:AMY1gene.png

# Genome analysis pipeline



DNA *in vitro*

Sequencing and assembly

DNA *in silico*

AGCT...

Gene prediction

annotation

cDNA *in vitro*

Sequencing and assembly

RNA *in silico*

...AUG...

Gene regulation studies: systems biology

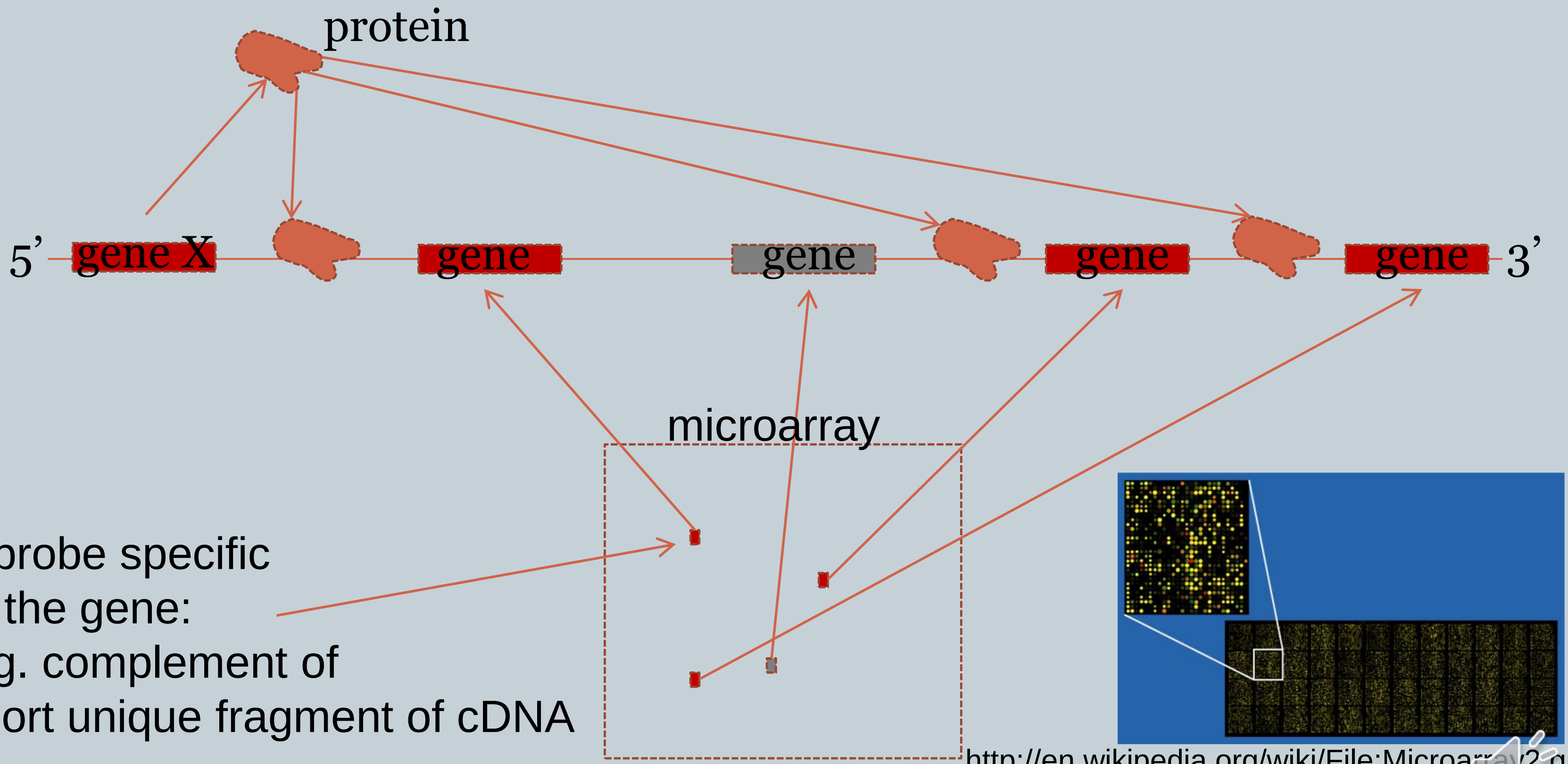
DNA and annotation of other species

# Gene regulation

- Let us assume that gene prediction is done.
- We are interested in signals that influence gene regulation:
  - How much mRNA is transcriped, how much protein is translated?
  - How to measure those?
    - 2D gel electrophoresis (traditional technique to measure protein expression)
    - Microarrays (the standard technique to measure RNA expression)
    - RNA-sequencing (a new technique to measure RNA expression, useful for many other purposes as well, including gene prediction)

# Microarrays and gene expression

- Idea:



protein

5’

gene X

gene

gene

gene

gene

3’

microarray

a probe specific
to the gene:
e.g. complement of
short unique fragment of cDNA

http://en.wikipedia.org/wiki/File:Microarray2.gif

# Time series expression profiling

- It is possible to make a series of microarray experiments to obtain a time series expression profile for each gene.

- *Cluster* similarly behaving genes.

# Analysis of clustered genes

- Similarly expressing genes may share a common transcription factor located upstream of the gene sequence.
  - Extract those sequences from the clustered genes and search for a common motif sequence.
  - This approach is called *motif discovery*.
- We concentrate now on the structure of upstream region, representation of motifs, and the simple tasks of locating the occurrences of already known motifs.

# Promoter sequences

- Immediately before the gene.
- Clear structure in prokaryotes, more complex in eukaryotes.
- An example from *E coli* is shown in next slide (from Deonier et al. book).

# Promoter example

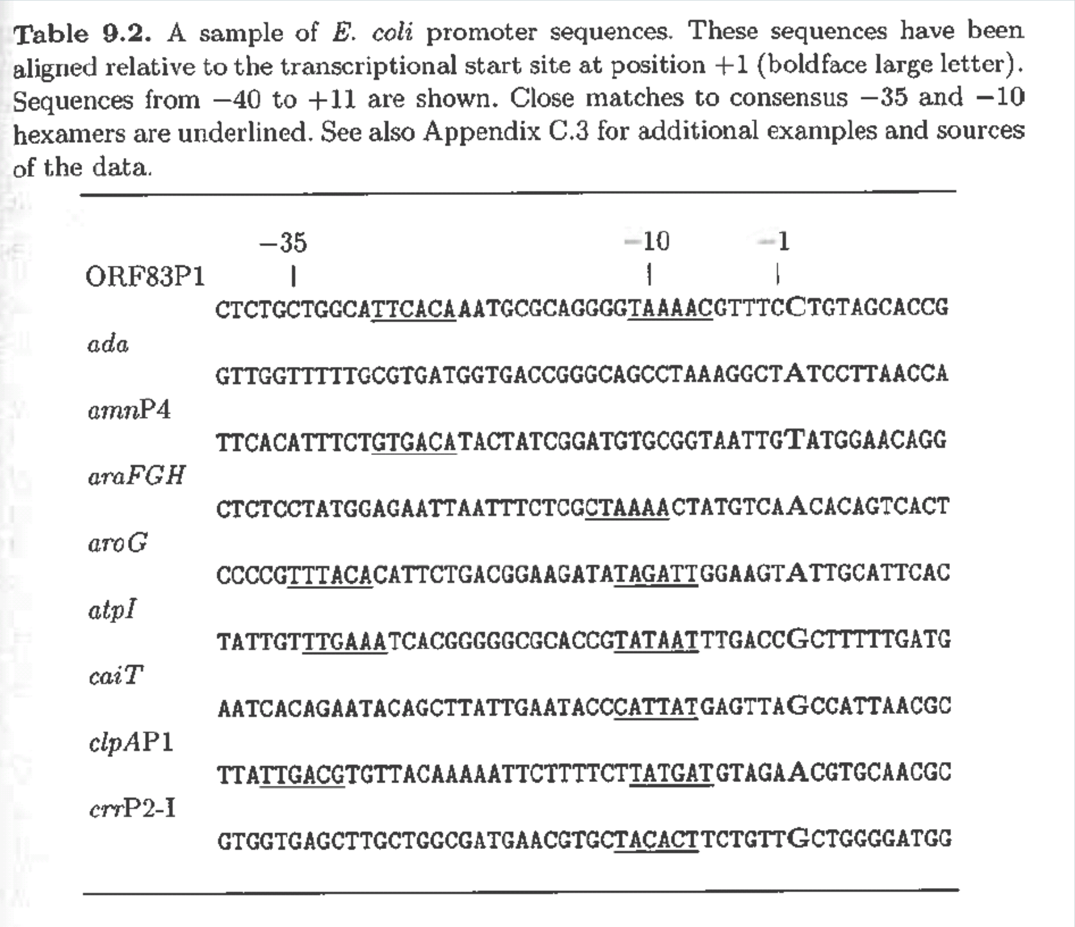


**Table 9.2.** A sample of *E. coli* promoter sequences. These sequences have been aligned relative to the transcriptional start site at position +1 (boldface large letter). Sequences from −40 to +11 are shown. Close matches to consensus −35 and −10 hexamers are underlined. See also Appendix C.3 for additional examples and sources of the data.

| | −35 −10 −1 |
|---|---|
| ORF83P1 | CTCTGCTGGCATTCACAAATGCGCAGGGGTAAAACGTTTC**C**TGTAGCACCG |
| *ada* | GTTGGTTTTTGCGTGATGGTGACCGGGCAGCCTAAAGGCT**A**TCCTTAACCA |
| *amn*P4 | TTCACATTTCTGTGACATACTATCGGATGTGCGGTAATTG**T**ATGGAACAGG |
| *araFGH* | CTCTCCTATGGAGAATTAATTTCTCGCTAAAACTATGTCA**A**CACAGTCACT |
| *aroG* | CCCCGTTTACACATTCTGACGGAAGATATAGATTGGAAGT**A**TTGCATTCAC |
| *atpI* | TATTGTTTGAAATCACGGGGGCGCACCGTATAATTTGACC**G**CTTTTTGATG |
| *caiT* | AATCACAGAATACAGCTTATTGAATACCCATTATGAGTTA**G**CCATTAACGC |
| *clpA*P1 | TTATTGACGTGTTACAAAAATTCTTTTCTTATGATGTAGA**A**CGTGCAACGC |
| *crr*P2-I | GTGGTGAGCTTGCTGGCGATGAACGTGCTACACTTCTGTT**G**CTGGGGATGG |

# Representing signals in DNA

- Consensus sequence:
  - -10 site in E coli: TATAAT
  - GRE half-site consensus: AGAACA
- Simple regular expression:
  - A(C/G)AA(C/G)(A/T)
- Positional weight matrix (PWM):

$$\begin{array}{c} \mathrm{A} \\ \mathrm{C} \\ \mathrm{G} \\ \mathrm{T} \end{array} \begin{bmatrix} 1.00 & 0.00 & 1.00 & 1.00 & 0.00 & 0.86 \\ 0.00 & 0.14 & 0.00 & 0.00 & 0.86 & 0.00 \\ 0.00 & 0.86 & 0.00 & 0.00 & 0.14 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.14 \end{bmatrix}$$

GRE half-sites:

AGAACA
ACAACA
AGAACA
AGAAGA
AGAACA
AGAACT
AGAACA

consensus: AGAACA

# Position-specific scoring matrix (PSSM)

- PSSM is a log-odds normalized version of PWM.[1]
- Calculated by $\log(p_{ai}/q_a)$, where
  - $p_{ai}$ is the frequency of a at column i in the samples.
  - $q_a$ is the probability of a in the whole organism (or in some region of interest).
- Problematic when some values $p_{ai}$ are zero.
- Solution is to use pseudocounts:
  - add 1 to all the sample counts where the frequencies are calculated.

[1] In the following log denotes base 2 logarithm.

# PWM versus PSSM

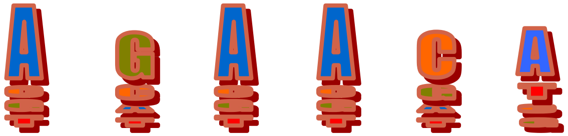
counts

$$\begin{tikzcd}
\begin{bmatrix} 7 & 0 & 7 & 7 & 0 & 6 \\ 0 & 1 & 0 & 0 & 6 & 0 \\ 0 & 6 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \arrow[r, "\text{PWM}"] \arrow[d, "\text{pseudocounts}"] & \begin{bmatrix} 1.00 & 0.00 & 1.00 & 1.00 & 0.00 & 0.86 \\ 0.00 & 0.14 & 0.00 & 0.00 & 0.86 & 0.00 \\ 0.00 & 0.86 & 0.00 & 0.00 & 0.14 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.14 \end{bmatrix} \\
\begin{bmatrix} 8 & 1 & 8 & 8 & 1 & 7 \\ 1 & 2 & 1 & 1 & 7 & 1 \\ 1 & 7 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 2 \end{bmatrix} \arrow[r, "\text{PSSM}", "\text{(position-specific scoring matrix)}"'] & \begin{bmatrix} 1.54 & -1.46 & 1.54 & 1.54 & -1.46 & 1.35 \\ -1.46 & -0.46 & -1.46 & -1.46 & 1.35 & -1.46 \\ -1.46 & 1.35 & -1.46 & -1.46 & -0.46 & -1.46 \\ -1.46 & -1.46 & -1.46 & -1.46 & -1.46 & -0.46 \end{bmatrix}
\end{tikzcd}$$

log((8/11)/(1/4))
log((1/11)/(1/4))
log((2/11)/(1/4))
log((7/11)/(1/4))

assuming $q_a$=0.25 for all $a$

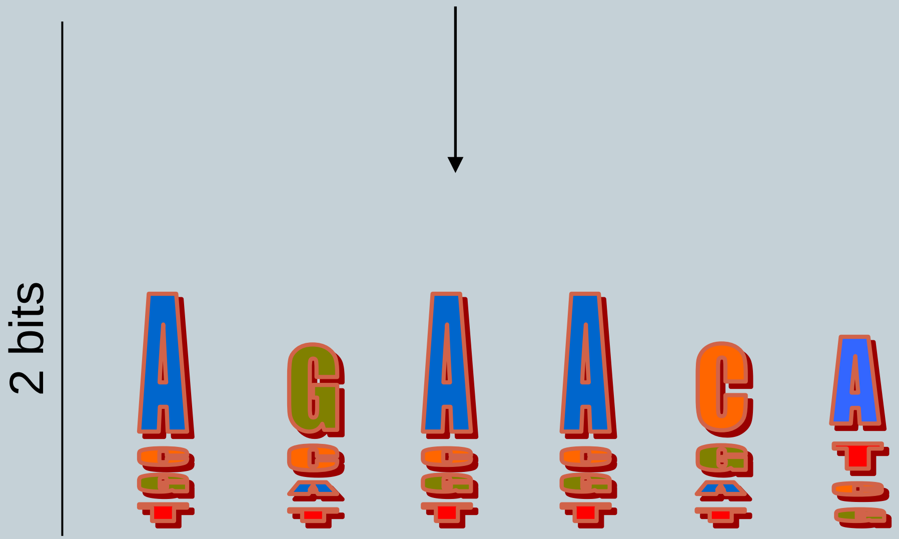
# Sequence logos



- Many known transcription factor binding site PWM:s can be found from JASPAR database (http://jaspar.cgb.ki.se/).
- PWM:s are visualized as *sequence logos*, where the height of each nucleotide equals its proportion of the relative entropy (expected log-odds score) in that column.
  - $$\mathrm{E}(S_i) = \sum_a p_{ai} \log(p_{ai} / q_a)$$
  - Height of a at column i is $p_{ai}\mathrm{E}(S_i)$

# Example sequence logo

$$\begin{bmatrix} 1.54 & -1.46 & 1.54 & 1.54 & -1.46 & 1.35 \\ -1.46 & -0.46 & -1.46 & -1.46 & 1.35 & -1.46 \\ -1.46 & 1.35 & -1.46 & -1.46 & -0.46 & -1.46 \\ -1.46 & -1.46 & -1.46 & -1.46 & -1.46 & -0.46 \end{bmatrix}$$



# Searching PSSMs

- As easy as naive exact text search (see next slide).
- Much faster methods exist. For example, one can apply branch-and-bound technique on top of suffix tree (omitted here).
- Warning:
  - Good hits for any PSSM are too easy to find!
  - Search domain must be limited by other means to find anything statistically meaningful with PSSMs only.
    - Typically used on upstream regions of genes clustered by gene expression profiling.

```
#!/usr/bin/env python
import sys
import time
# naive PSSM search
matrix = {'A':[1.54,-1.46,1.54,1.54,-1.46,1.35],
          'C':[-1.46,-0.46,-1.46,-1.45,1.35,-1.46],
          'G':[-1.46,1.35,-1.46,-1.46,-0.46,-1.46],
          'T':[-1.46,-1.46,-1.46,-1.46,-1.46,-0.46]}
count = {'A':0,'C':0,'G':0,'T':0}
textf = open(sys.argv[1],'r')
text = textf.read()
m=len(matrix['A'])
bestscore = -m*2.0
t1 = time.time()
for i in range(len(text)-m+1):
   score = 0.0
   for j in range(m):
      if text[i+j] in matrix:
         score = score + matrix[text[i+j]][j]
         count[text[i+j]] =  count[text[i+j]]+1
      else:
        score = -m*2.0
      if score > bestscore:
         bestscore = score
         bestindex = i
t2 = time.time()
totalcount = count['A']+count['C']+count['G']+count['T']
expectednumberofhits = 1.0*(len(text)-m+1)
for j in range(m):
   expectednumberofhits = expectednumberofhits*float(count[text[bestindex+j]])/float(totalcount)
print 'best score ' + str(bestscore) + ' at index ' +str(bestindex)
print 'best hit: ' + text[bestindex:bestindex+m]
print 'computation took ' + str(t2-t1) + ' seconds'
print 'expected number of hits: ' + str(expectednumberofhits)
```

```
pssm.py hs_ref_chrY_nolinebreaks.fa
best score 8.67 at index 397
best hit: AGAACA
computation took 440.56187582 seconds
expected number of hits: 18144.7627936
```

no sense in
this search!

# Refined motifs

- Our example PSSM (GRE half-site) represents only half of the actual motif: the complete motif is a palindrome with consensus:
  - ○ AGAACAnnnTGTTCT

```
pssmpalindrome.py hs_ref_chrY_nolinebreaks.fa
best score 17.34 at index 17441483
best hit: AGAACAGGCTGTTCT
computation took 1011.4800241 seconds
expected number of hits: 5.98440033042
total number of maximum score hits: 2
```

  - ○ Exercise: modify pssm.py into pssmpalindrome.py
    … or learn biopython to do the same in few lines of code

# Discovering motifs

- *Principle:* discover over-represented motifs from the promotor / enhancer regions of co-expressing genes.
- How to define a motif?
  - Consensus, PWM, PSSM, palindrome PSSM, co-occurrence of several motifs (enhancer modules),…
  - Abstractions of protein-DNA chemical binding.
- Computational challenge in motif discovery:
  - Almost as hard as (local) multiple alignment.
  - Exhaustive methods too slow.
  - Lots of specialized pruning mechanisms exist.
- New sequencing technologies will help (ChIP-seq).