# Computational methods of systems biology

**Esko Ukkonen**
Department of Computer Science
University of Helsinki
http://www.cs.helsinki.fi/u/ukkonen/

***Lectures in Fall 2009***

## Contents of the course (tentative)

- Lecture 0: Introduction
- Lecture 1: Hidden Markov Models
- Lecture 2: Applications of HMM – profile-HMMs for sequence families
- Lecture 3: Applications of HMM – modeling transcription binding sites with position weight matrices
- Lecture 4: Transcription networks – basics
- Lecture 5: Autoregulation – a network motif
- Lecture 6: The feed-forward loop network motif
- Lecture 7: Temporal programs and the global structure of transcription networks
- Lecture 8: Network motifs in developmental, signal transduction, and neuronal networks
- Lecture 9: Kinetic proofreading
- Lecture 10: Metabolic networks

## Comments on the contents

- The first part of the course (lectures 1-3) is an introduction to Hidden Markov models and their applications (profile models of sequence families, PWMs, cis-regulation) in biological sequence analysis. The text book by Durbin et al is the main source.
- The second part (lectures 4-10) is on biological networks, based on the recent text book by Uri Alon. The book has a novel approach to this very messy field. The idea is to understand various biological networks in terms of different regulatory 'motifs' that occur unexpectedly often in these networks (and hence should have been conserved in the evolution).
- One lecture (given by Esa Pitkänen) will be an introduction to metabolic networks and flux analysis

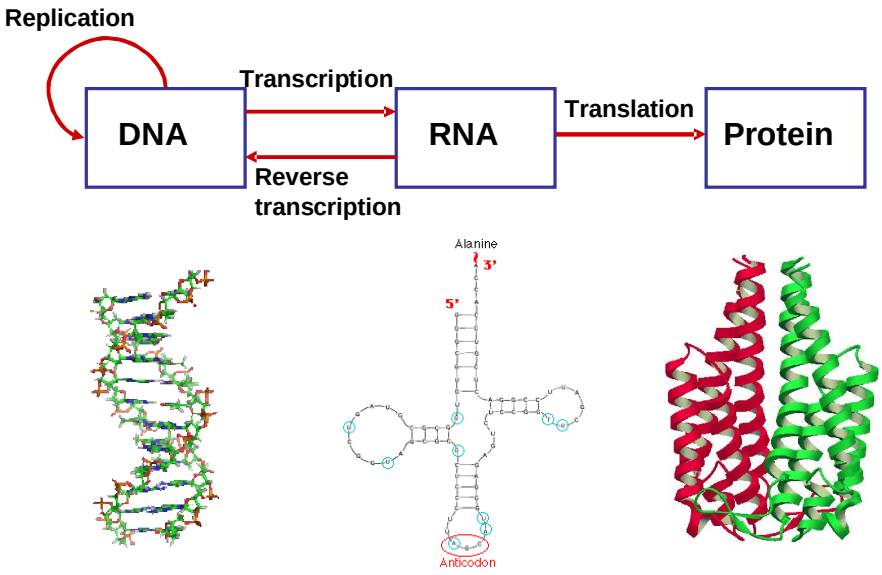
## Sources

- Text books
  - **R.Durbin, S.R.Eddy, A.Krogh & G.Mitchison: Biological sequence analysis. Cambridge University Press 1998**
    - Chapters 1.3, 3, 5 (lectures 1, 2, 3)
  - **U. Alon: An introduction to systems biology – Design principles of biological circuits. Chapman & Hall/CRC 2007**
    - Chapters 2, 3, 4, 5, 6, 9 (lectures 4-9)
  - **M. Zvelebil & J.O.Baum: Understanding Bioinformatics. Garland Science 2008**
    - Chapters 6.3 – 6.6 (lecture 3)
- Original articles

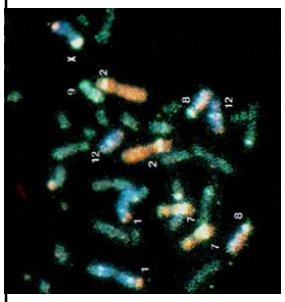
# Lecture 0: Introduction, background, summary of some topics

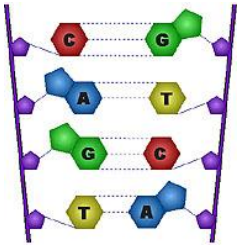
This lecture recalls some core bioinformatics problems and computational techniques that the participants are supposed to be familiar with. Some examples of the course content are also mentioned.

## Information flow in a cell



# DNA sequencing and genome projects






**cgccgagtgacag**
**agacgctaatcagg**
**ctgtgttctcaggat**
**gcgtaccgagtgg**
**agacagcagcacg**
**accag…**

# DNA fragment assembly problem

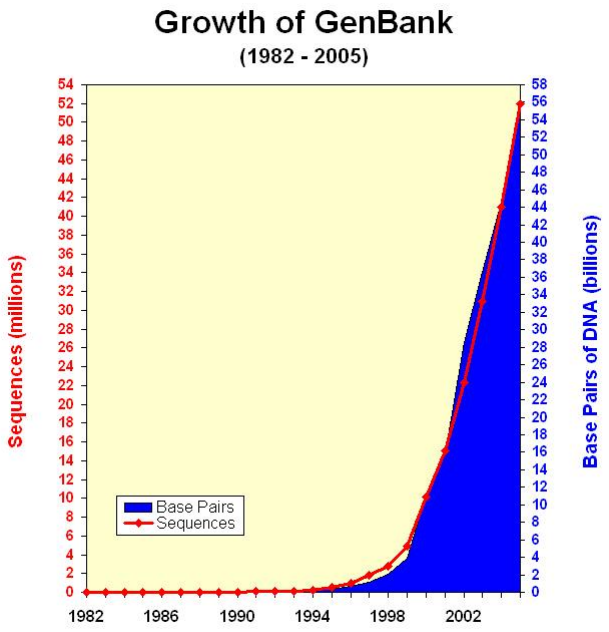
**cctcgagttaagtactgcccgcggcttcaacggatctgtcgggagtcg**

Re-assemble the puzzle?

**cctcgagtttaa**
**tacttaactcgag**
**cgggcagtacttaa**
**aagtactgcccgcg**
**gcccgcggcttcaacggat**
**cccgcggcttcaacggatctgtg**
**cccgacacagat**
**tgtgtcgggagtcg**

## Some numbers (Human Genome)

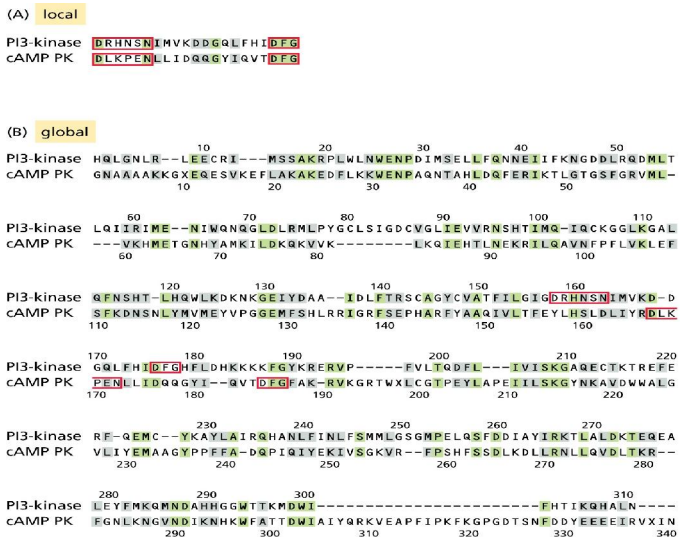
- Total length about 3 000 000 000 bp
- Celera's fragment data (Feb 2001):
  27 million fragments, each 150-800 bp
- at least 7-fold coverage by fragments
  needed => total data length 7 x 3 billion bp



# How does the DNA program work?

```
cgccgagtgacagagacgc
taatcaggctgtgttctca
ggatgcgtaccgagtggga
gacagcagcacgaccagcg
gtggcagagacccttgcag
acatcaagctctttgggaa
caagtggagcaccgatgat
gtacagccgatcaatgaca
tttccctaatgcaggatta
cattgcagtgcccaaggag
aagtatg...
```


# Pairwise alignment



$d_{i,j} = \min(\text{if } a_i = b_j \text{ then } d_{i-1,j-1} \text{ else } \infty,$
$d_{i-1,j} + 1,$
$d_{i,j-1} + 1)$

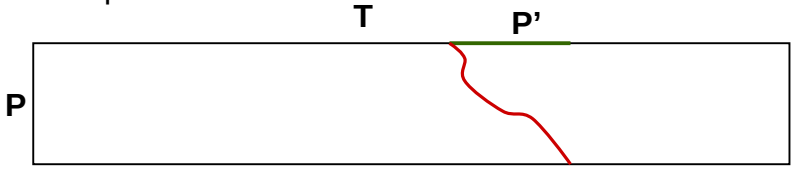
| A\B | | S | T | O | C | K | H | O | L | M |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| T | 1 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| U | 2 | 3 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| K | 3 | 4 | 3 | 4 | 5 | 4 | 5 | 6 | 7 | 8 |
| H | 4 | 5 | 4 | 5 | 6 | 5 | 4 | 5 | 6 | 7 |
| O | 5 | 6 | 5 | 4 | 5 | 6 | 5 | 4 | 5 | 6 |
| L | 6 | 7 | 6 | 5 | 6 | 7 | 6 | 5 | 4 | 5 |
| M | 7 | 8 | 7 | 6 | 7 | 8 | 7 | 6 | 5 | 4 |
| A | 8 | 9 | 8 | 7 | 8 | 9 | 8 | 7 | 6 | 5 |

optimal alignment by trace-back = **Viterbi!**

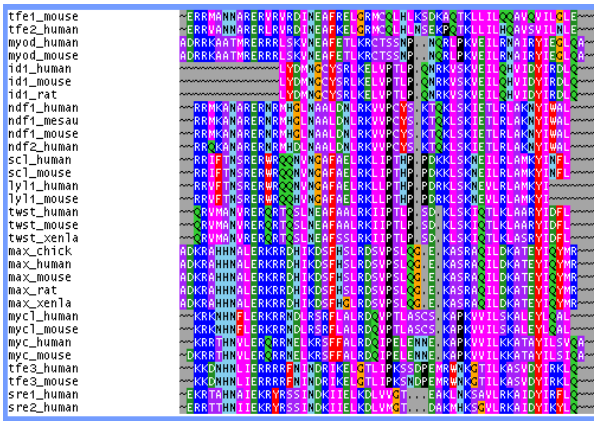
$d_{ID}(A,B)$

# Search problem

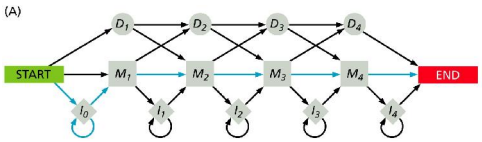
- find approximate occurrences P' of pattern P in text T such that d(P,P') small
- dyn progr with small modification: O(mn)
- lots of (practical) improvements:
  - distance bound k → O(kn) search
  - utilize regularities of the dp table
  - **filtration approach: BLAST (big success!)**
  - bit-parallelism



# Multiple alignment

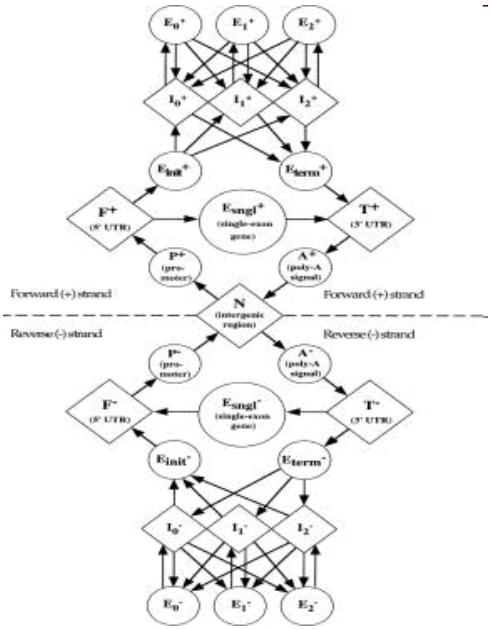



# Hidden Markov Models (HMM) for Sequence Families



## HMM architecture of GENESCAN

**Prediction of genes**




## Sequence motifs

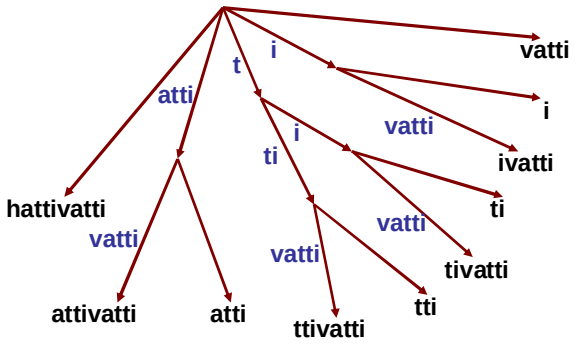
```
cgccgagtgacagagacgctaatcag
gcgacccttgcagacatcaagctctt
tgggaacaagtggagcaccgatgatg
tacagccgatcaatgacatttcccta
atgcaggattacattgcagtgcccaa
ggagaagtatgccaagtaatacctcc
ctcacagtg...
```

## Sequence motifs

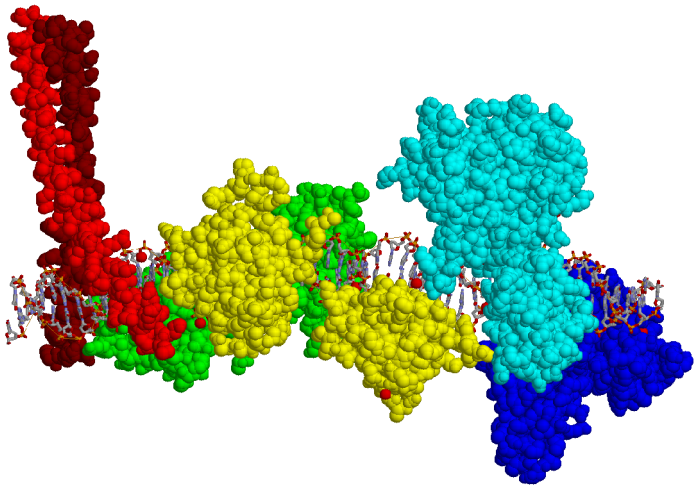
```
cgccgagtgacagagacgctaatcag
gcgacccttgcagacatcaagctctt
tgggaacaagtggagcaccgatgatg
tacagccgatcaatgacatttcccta
atgcaggattacattgcagtgcccaa
ggagaagtatgccaagtaatacctcc
ctcacagtg...
```

## Suffix-trees

hattivatti
attivatti
ttivatti
tivatti
ivatti
vatti
atti
tti
ti
i

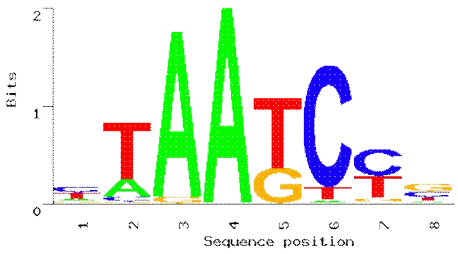


# Transcription factor binding sites

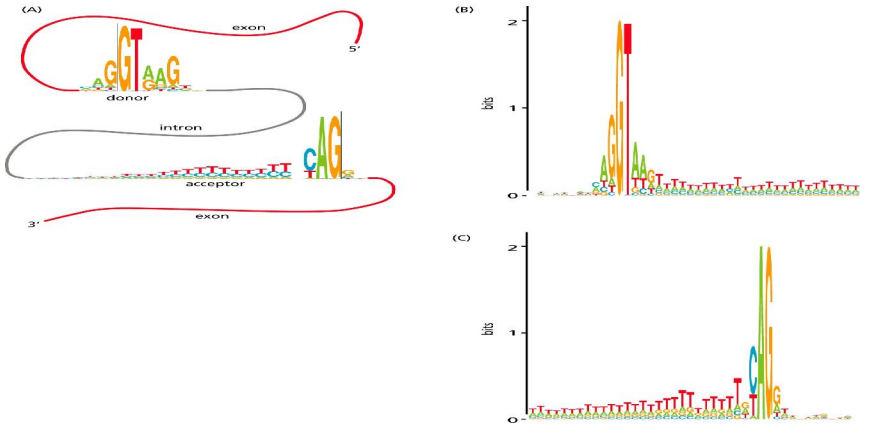


# Binding affinity matrices (PWMs)

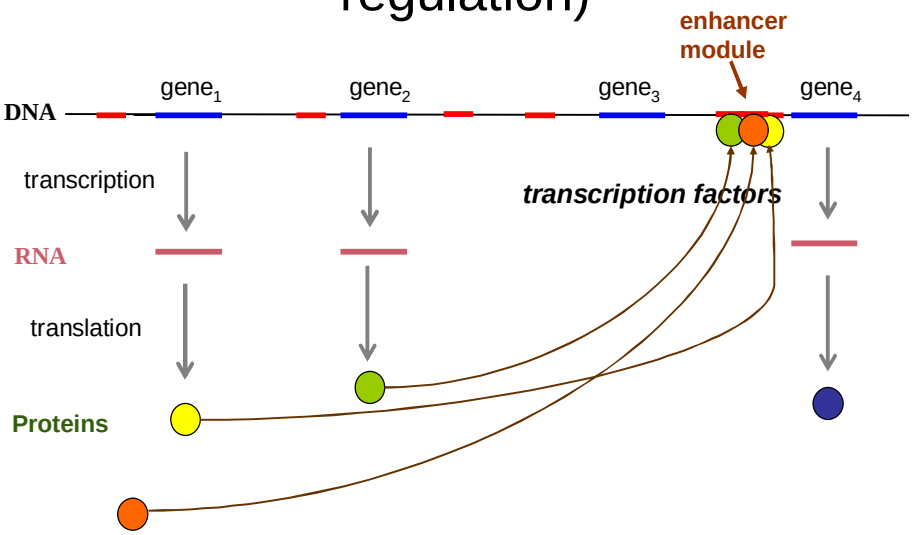
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **A** | 9 | 11 | 49 | 51 | 0 | 1 | 1 | 4 |
| **C** | 19 | 3 | 0 | 0 | 0 | 45 | 25 | 16 |
| **G** | 5 | 1 | 2 | 0 | 17 | 0 | 4 | 21 |
| **T** | 18 | 36 | 0 | 0 | 34 | 5 | 21 | 10 |



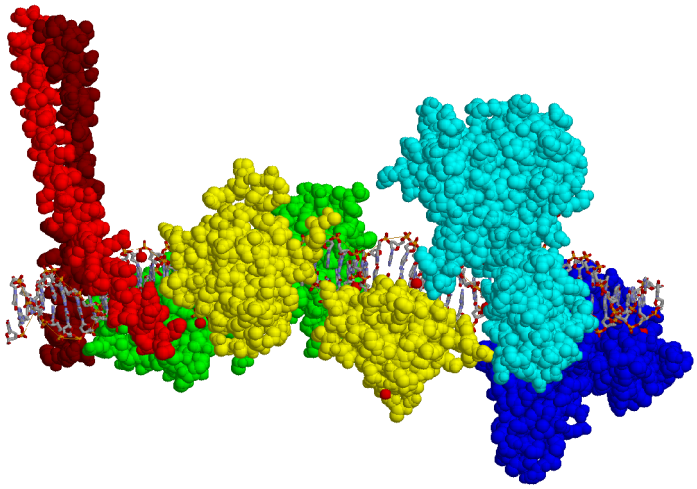
# Search for PWM occurrences




# Gene enhancer modules (cis-regulation)



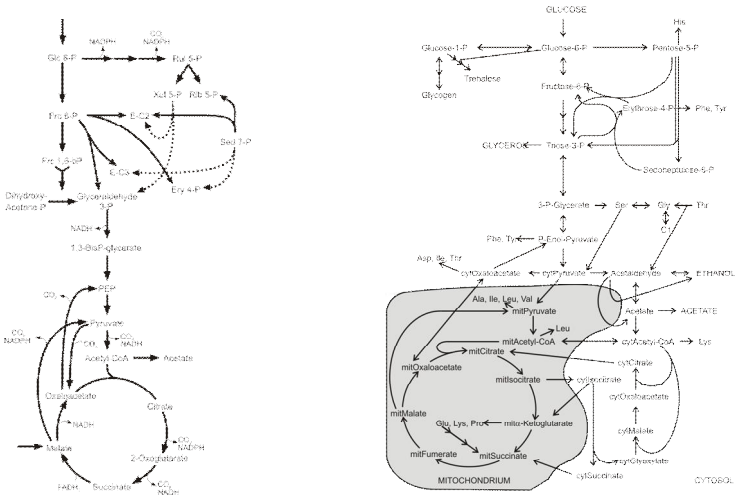
# Enhancer module



# *Drosophila* enhancer

- *Drosophila* even-skipped gene stripe 2 enhancer
- Score = 487.05

# Regulatory networks




# Computational techniques – Do you already know all these?

- Dynamic programming
- Needleman-Wunsch algorithm, Smith-Waterman algorithm, Viterbi algorithm
- Probabilistic modeling
- Hidden Markov Models
- Maximum likelihood estimation
- Expectation Maximization (EM) algorithm
- Combinatorial pattern matching
- Exact and approximate string matching
- Aho-Corasick & Boyer-Moore algorithms
- Index structures for sequential data
- Suffix-trees, suffix-arrays, BW transform