27622: Introduction to Bioinformatics, Turbo version

Henrik Nielsen, Associate professor
Center for Biological Sequence Analysis

What is bioinformatics?

What are bioinformaticians up to, actually?

- **Manage** molecular biological data
  - Store in databases, organise, formalise, describe...
- **Compare** molecular biological data
- **Find patterns** in molecular biological data
  - phylogenies
  - correlations (sequence / structure / expression / function / disease)

Goals:
- **characterise** biological patterns & processes
- **predict** biological properties
  - low level data ⇒ high level properties
    (eg., sequence ⇒ function)
Bioinformatics: neighbour disciplines

- Computational biology
  - Broader concept: includes computational ecology, physiology, neurology etc...
- -omics:
  - Genomics
  - Transcriptomics
  - Proteomics
- Systems biology
  - Putting it all together...
  - Building models, identify control & regulation

A view of Systems Biology

Bioinformatics: prerequisites

- **Bio- side:**
  - Molecular biology
  - Cell biology
  - Genetics
  - Evolutionary theory
- **informatics** side:
  - Computer science
  - Statistics
  - Theoretical physics
Molecular biology data...

- DNA sequences

- Amino acid sequences
- Protein structure:
  - X-ray crystallography
  - NMR

Cell biology & proteomics data...

- Subcellular localization
Cell biology & proteomics data...

protein-protein interactions

Transcriptomics: DNA microarray technology

Phenotype data: human diseases
Prediction methods

• Homology / Alignment
• Simple pattern ("word") recognition
• Statistical methods
  – Weight matrices: calculate amino acid probabilities
  – Other examples: Regression, variance analysis, clustering
• Machine learning
  – Like statistical methods, but parameters are estimated by iterative training rather than direct calculation
  – Examples: Neural Networks (NN), Hidden Markov Models (HMM), Support Vector Machines (SVM)
• Combinations

The computer

• Everything can be reduced to bits (0 or 1)

Digital information

• A byte = 8 bits
  01000001
  Can be interpreted as
  • The number 65
  • The letter "A"
  • Part of a machine code instruction
  • Part of a colour specification
  • Part of a sound encoding
  • …
A text file is a file where every byte is interpreted as a character

Examples
Plain text	.txt
Program settings	.ini
C source code	.c
Python script	.py
TeX source	.tex
Web page source	.html
Sequences	.fasta

The ASCII table

Extended character sets

The are many ways to interpret characters with values above 127. Here, you see two of them.

Text files—line endings

- UNIX standard (including Mac OS X):
  - 10 — LF ("Line feed" char).
- Old Mac (System 9 and before):
  - 13 — CR ("Carriage Return" char).
- DOS/Windows:
  - 13, 10 — both CR and LF.

A good text editor can handle all three systems. Notepad for Windows cannot!