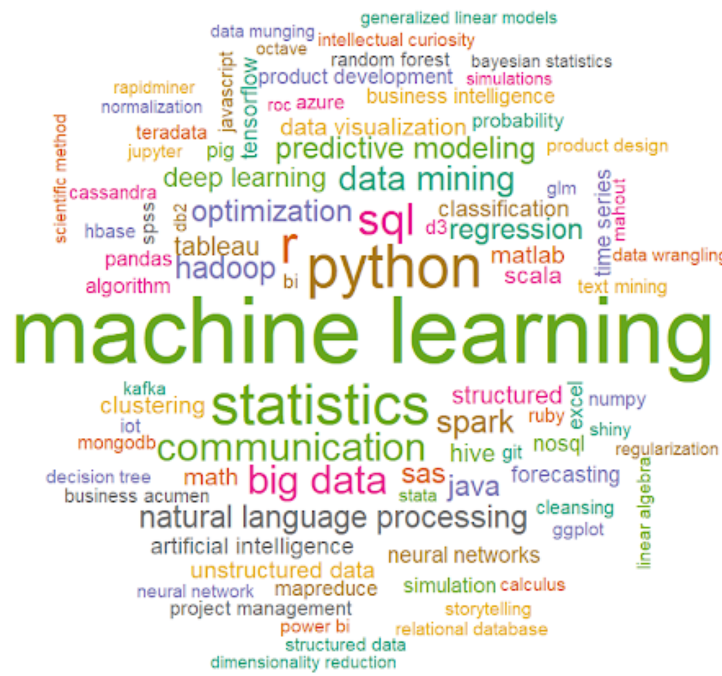
Programming in Biology

MCDB 170

Instructor: Sung Soo Kim (MCDB)

Computer Science

Data Analysis



Course objectives

- General working knowledge in Python (3 wks)
- DNA sequence analysis using string and Biopython (2 wks)
- Using Numpy and Scipy to simulate biological system (2 wks)
- Using Numpy and Scipy to perform statistical analyses of biological data (2 wks)
- Biological image analysis using scikit-image (will be covered if time allows)

Logistics

- Lectures are asynchronous
- TA sections will be synchronous
- Exams will be synchronous
- Homework assignment is due Sunday 11:59 pm.
- Quiz: Unlimited number of attempts allowed. Due is Wednesday 11:59pm. No more attempt is allowed after the due.

See syllabus for the schedule

Grading

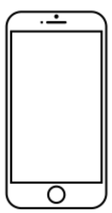
- Will be based on the absolute points
- Homework assignments: 90 points
- Quiz: 45 points
- TA sections: 20 points (for attendance)
- Exams: 1 mid-term (30 points), 1 final (50 points)
- Total: 235 points (Grading table is in the syllabus)

MISC

- Instructor office hours: Thursdays 11 am – 12 pm
- TA (Jon Luntzel) office hours: Fridays 4:30 pm – 6 pm
- Email: Use a title beginning with MCDB170. Will be answered in 1–2 days

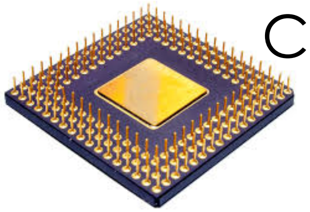
What is programming?

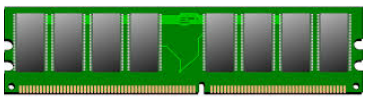
- Writing instructions for computers to perform tasks



CPU

Execute instructions



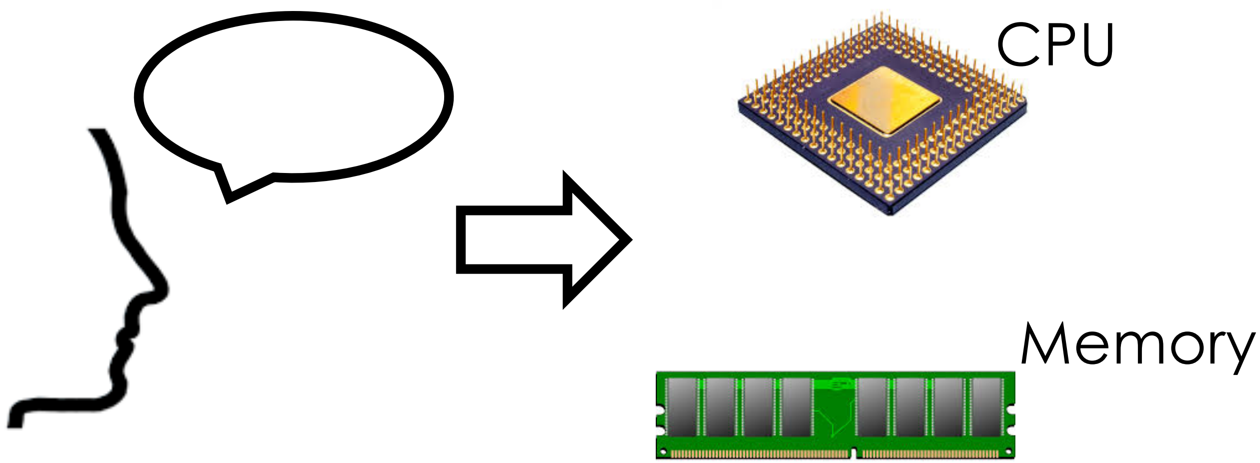


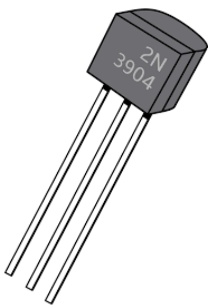
Memory

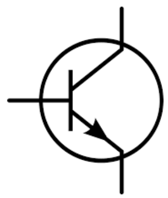
Data
Instructions

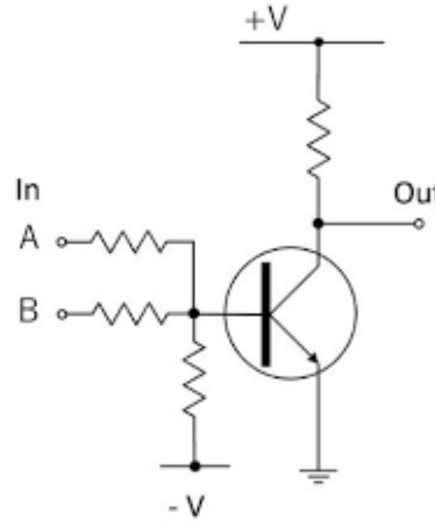
What is programming Language?

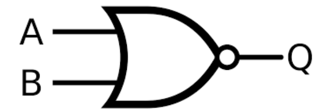
- Writing instructions for computers to perform tasks

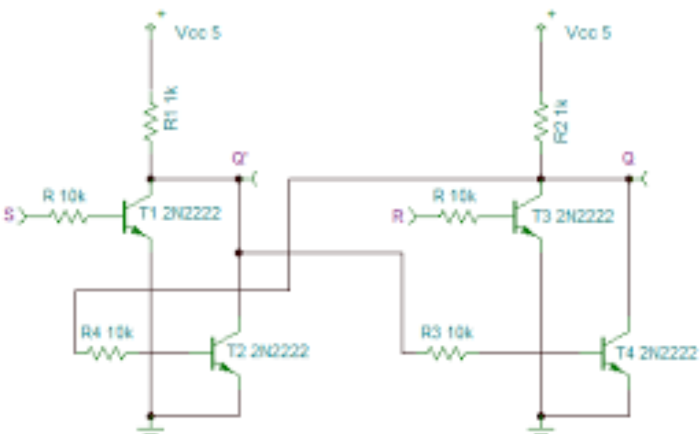






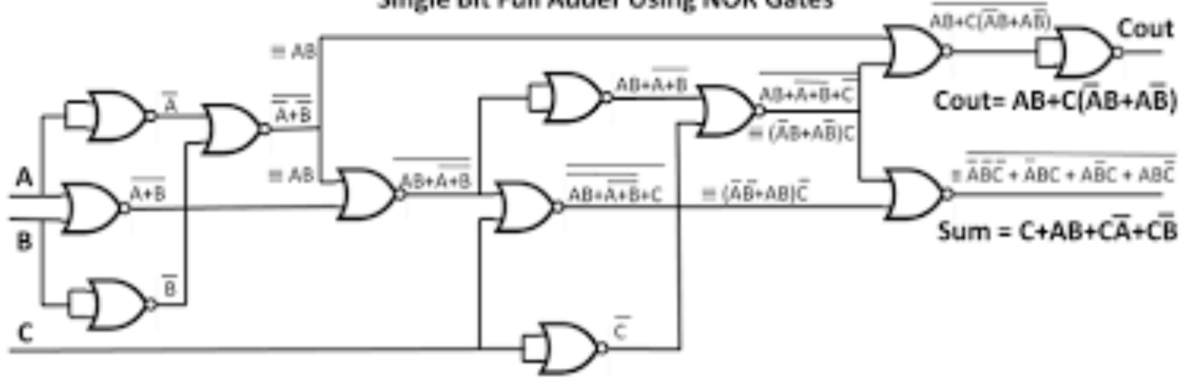






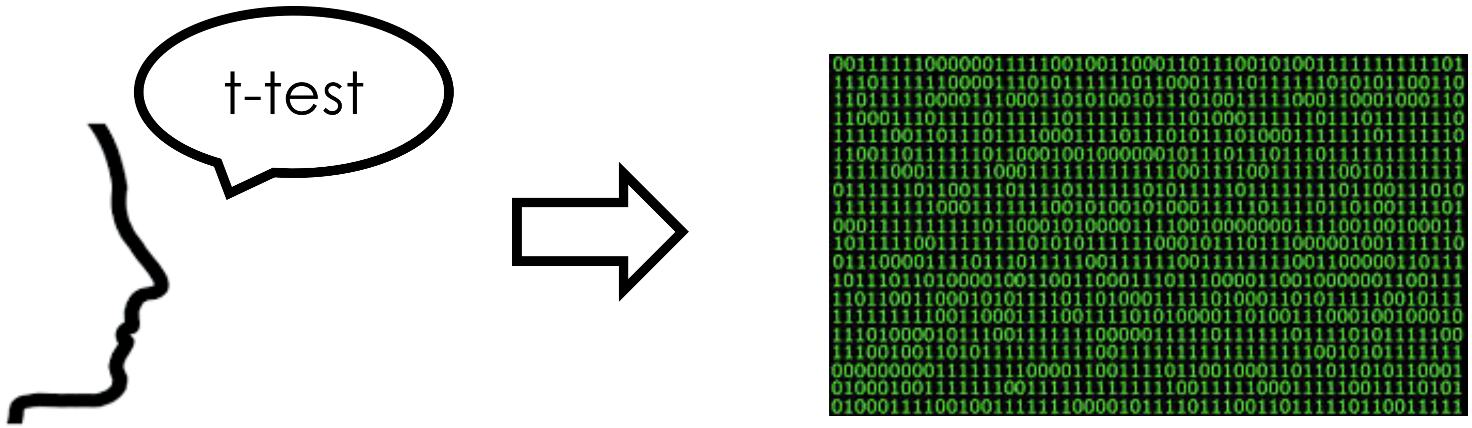
1-bit Memory

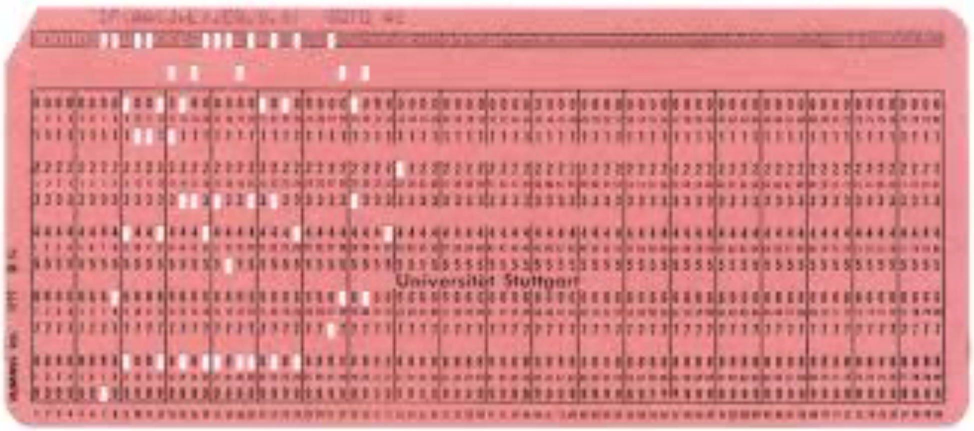
Single Bit Full Adder Using NOR Gates

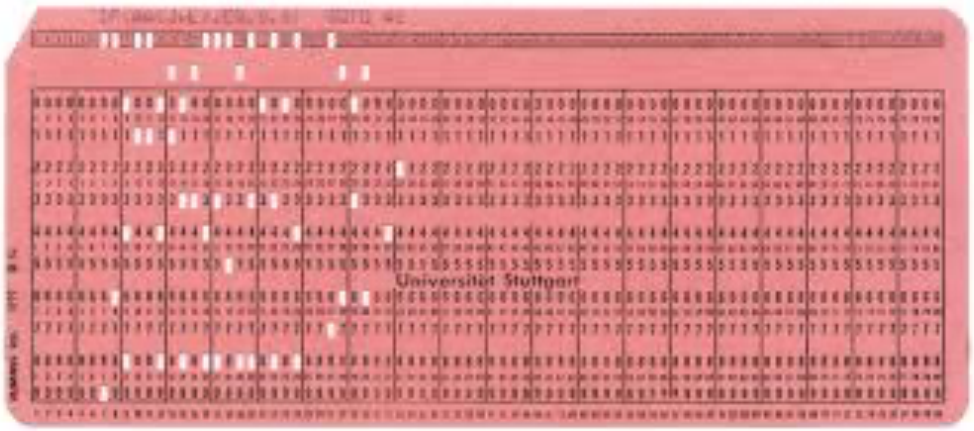


Unit of information: Bit 0 , 1

Computer operations are extremely simple (and bits)




Universität Stuttgart



```
push ebp
mov ebp, esp
mov eax, [ebp+0x08]
fld tword [eax]
mov ecx, [ebp+0x0C]
fld tword [ecx]
faddp
mov edx, [ebp+0x10]
fstp tword [edx]
fld tword [eax+0x0A]
fld tword [ecx+0x0A]
faddp
fstp tword [edx+0x0A]
pop ebp
ret 0x000C
```

```
PIN=0.02
IF(DDT.NE.0.0)  THEN
DT=DDT
ELSE
DT=PIN
ENDIF
WRITE(*,'(A)') '    PLEASE ENTER NAME OF OUTPUT FILE
*  B:ZZ.DAT)'
READ(*,'(A)') FNAMEO
OPEN(6,FILE=FNAMEO,STATUS='UNKNOWN')
PV=WFLX/TH
RS=NEQ*ROU*KD/TH
C0=CS
```

```
#include <stdio.h>

main()
{
    printf("hello, world\n");
}
```

```
push ebp
mov ebp, esp
mov eax, [ebp+0x08]
fld tword [eax]
mov ecx, [ebp+0x0C]
fld tword [ecx]
faddp
mov edx, [ebp+0x10]
fstp tword [edx]
fld tword [eax+0x0A]
fld tword [ecx+0x0A]
faddp
fstp tword [edx+0x0A]
pop ebp
ret 0x000C
```

Complied Languages

C/C++
Fortran
Java
etc

Interpretable Languages

Python – in almost all science
Matlab – Most engineering and some biology
R – in almost all statistics
Perl – Extensively used in Human Genome project
Ruby
etc

Why Python?

- Easy to learn: Syntax is very human friendly
- Extremely popular in every field of science
- Mature scientific libraries (Numpy, Scipy)
- Easy-to-learn programming environment (JupyterLab)

What you need to do Python programming

- Chrome browser
- UCSB NetID
- Connect to http://mcdb170.lsit.ucsb.edu/

Required reading (& watching)

- JupyterLab basics
  - https://youtu.be/A5YyoCKxEOU

- Interface, file operations, and notebook basics.
  - https://jupyterlab.readthedocs.io/en/stable/user/interface.html
  - https://jupyterlab.readthedocs.io/en/stable/user/files.html
  - https://jupyterlab.readthedocs.io/en/stable/user/file_editor.html
  - https://jupyterlab.readthedocs.io/en/stable/user/notebook.html