Bioinformatics LAB 1

FASTA and FASTQ file manipulation

Prof.ssa Elisa Ficarra
Prof.ssa Santa Di Cataldo
Eng. Marta Lovino
Eng. Alessio Mascolini

Politecnico di Torino
DAUIN
Dept. of Control and Computer Engineering

Organization

Schedule & Contacts

LABS

Tuesday	13.00 – 16.00
Thursday	11.30 – 14.30

Please check the **Teaching Portal** and the **Telegram group** to be updated.
https://t.me/joinchat/RykRgRtIMIASc8gLJtwA0g

CONTACT FOR LABS on GENOMICS

For any problems concerning the LABs, feel free to contact **Eng. Marta Lovino** during LABS class or by e-mail (**marta.lovino@polito.it**). Kindly indicate as the subject of the email “BIOINFO LABs“.

Schedule & Contacts

CONSULTING time

Each Tuesday 16.00-16.30
no need to book for a meeting

Use ALWAYS the following link:

Entra nella riunione in **Zoom**
https://us02web.zoom.us/j/2944811874?pwd=WUxTVmpGemdvM21Md0FJbzd1enBLQT09

ID riunione: 294 481 1874
Passcode: 6994523564

How to work

Start programming right now!

- Use **Telegram** group to ask for questions and/or the **consulting time**!
  https://t.me/joinchat/RykRgRtIMIASc8gLJtwA0g
  https://us02web.zoom.us/j/2944811874?pwd=WUxTVmpGemdvM21Md0FJbzd1enBLQT09

  ID riunione: 294 481 1874
  Passcode: 6994523564

- **Share your lab solutions** here
  LABXX_Surname_EsXX (LAB01_Lovino_Es04.py)
  https://drive.google.com/drive/folders/1-duA5R3ejTHOclRvuLWOL5d-egoDwgrE?usp=sharing

Setup your environment

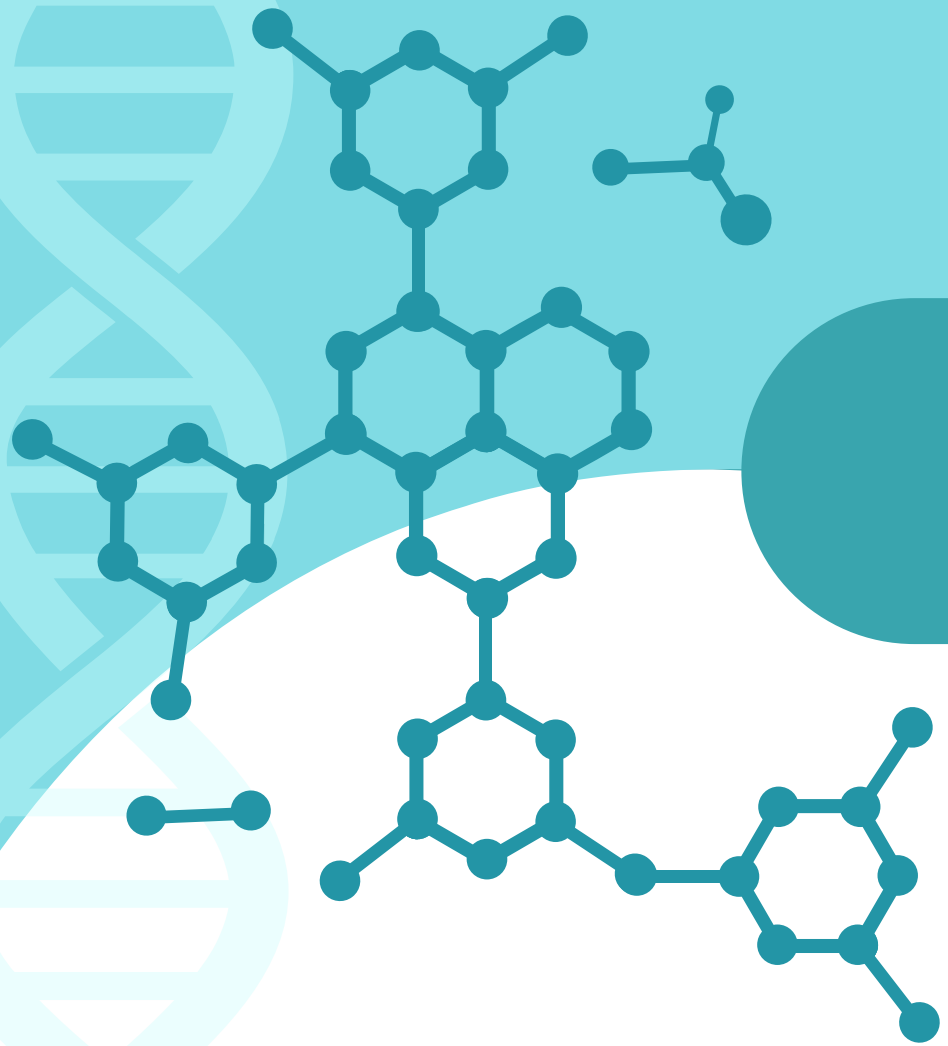
We strongly recommend **using Pycharm as Python IDE** and **a terminal with Anaconda distribution** installed.

Follow **the instructions in the "Setup for labs" files** uploaded on the teaching portal.

Feel free to ask us for problems in the installation process, we can provide you alternative solutions if you find issues with the suggested tools.

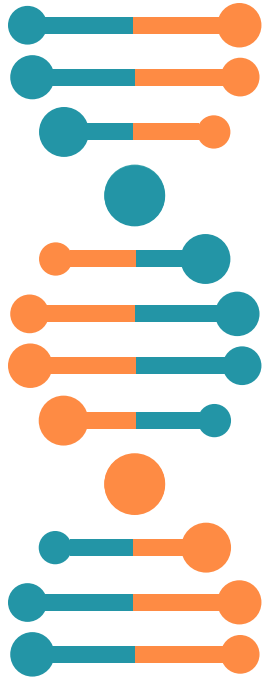
Structure of each LAB

1. **Definition of LAB objectives**
2. **Bio and Computational introduction to the lab**
3. **LAB assignments**
4. **Question & Answer session.** I am online to assist you



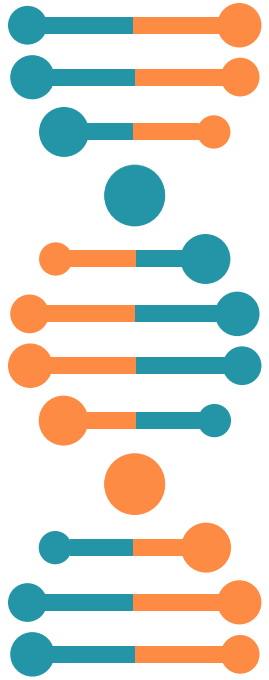
LAB 1 - Objectives

Objectives

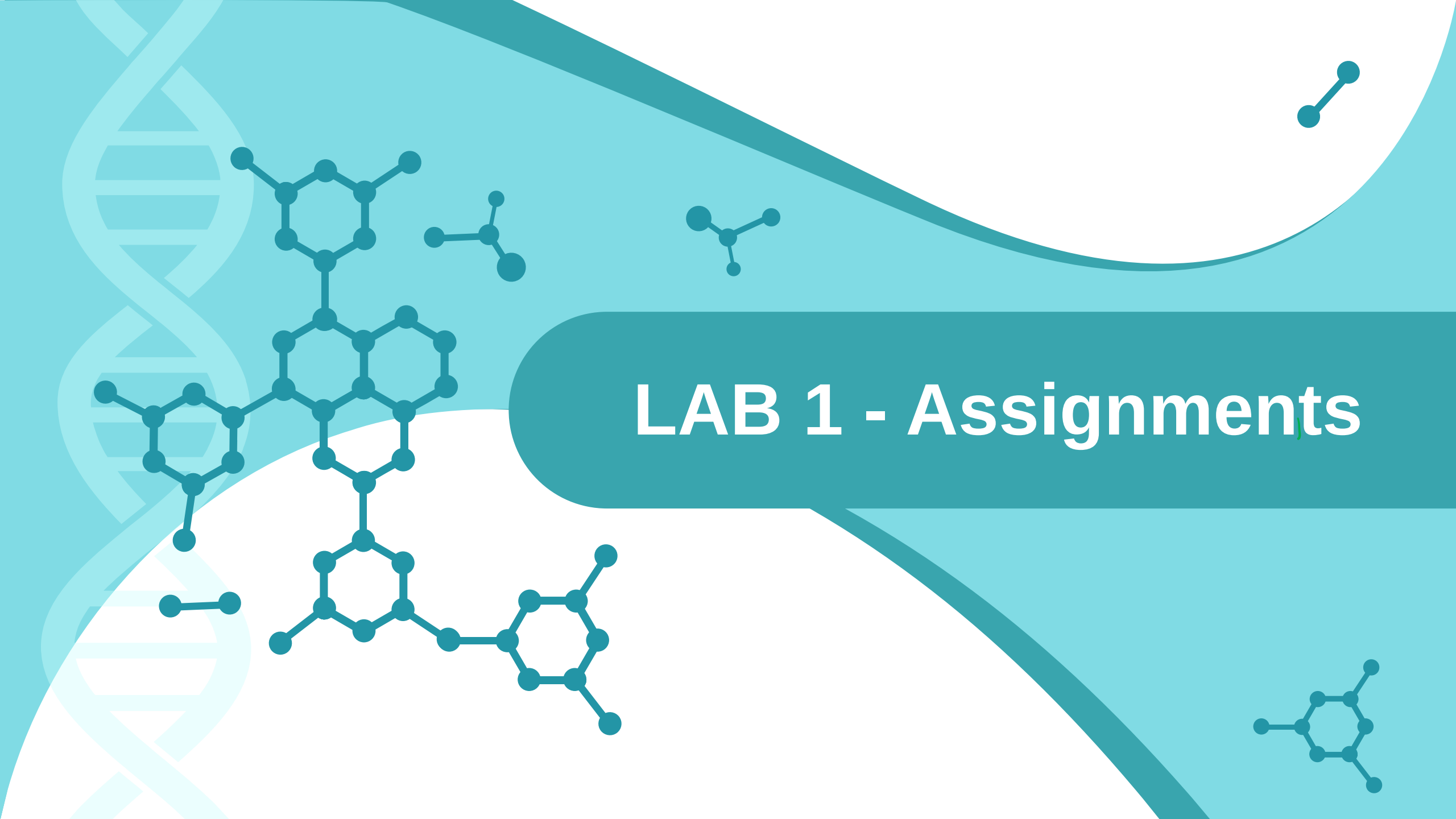


- Biological meaning of FASTA and FASTQ files
- Simulate genomics files (FASTA and FASTQ)
- Understanding and optimizing code

FASTA and FASTQ examples



- Quick bio recap
- FASTA
- FASTQ
- Single line/multi-line



LAB 1 - Assignments

Assignment 1: Random fasta and fastq file generator

Write a python program that generates both fasta and fastq files containing reads with the following characteristics:

-Read id contains a progressive number starting from 0.

-Sequences have length 50 bp

-Bases are randomly generated using a A,T,C,G alphabet, but probability of each base for each read should be given from the command line as a set of numbers (probA, probT, probC, probG)

- The number of reads should be passed as an argument from the command line

-The name of the fasta/fastq file should be passed as an argument from the command line

- For fastq files only: the quality of each base is randomly selected.

Example:

python read_generator simulatedfasta.fa 100 30 30 30 10

Assignment 2: Statistics extraction

Write a python program for extracting statistics from fasta/fastq files. The program must take as a first argument from the command line the name of the input fasta file to be analyzed and write to an output text file (whose name is passed as a second argument from the command line) a summary of the computed statistics.

The following are the expected output statistics:

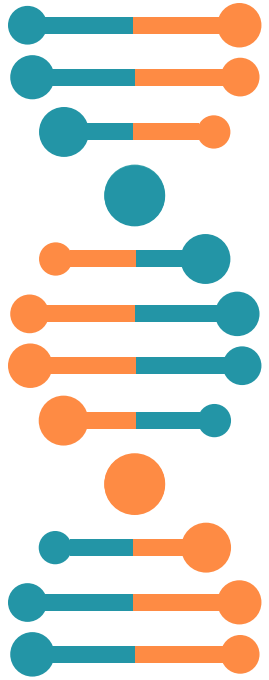
- Statistics of single bases across all the reads: Number of A,T,C,G

- Number of reads having at least one low complexity sequence: AAAAAA, TTTTTT, CCCCCC or GGGGGG.

- Number of reads having the number of GC couples (so called **GC content**) higher than a threshold GC_THRESHOLD passed as third argument from the command line

- For each read having a GC content higher than GC_THRESHOLD, report the read_id and the number of GC couples

Assignment 3: Fasta comparison



Write a python program to compare two fasta files. The two fasta files are passed as first and second argument from the command line.

The two fasta files have the following characteristics:

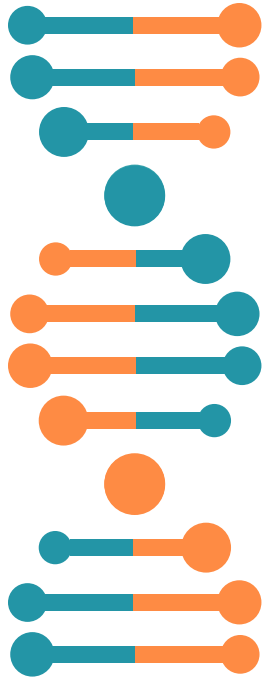
-The fasta format of the two files is correct (no need to check the format)

-Each read can take up one or multiple lines

-Each input file does not contain duplicated reads (i.e. identical reads)

The program must write as output a third fasta file containing only the reads that are in common between the input files. The read ids in the output file should be composed by the read id of the first file concatenated with the read id of the second file.

Assignment 4: Consensus Region



Write a python program that reconstructs the consensus regions on a specific chromosome starting from a tab-separated file called *alignments.txt* made up of three columns: the read ID, the sequence of the read and the alignment position of the read onto the reference genome. An example of *alignments.txt* is available in the following.

Exploiting the sequence and the alignment position of each read, build the consensus regions on the selected chromosome. Please note that all reads have the same length and that multiple consensus regions are allowed for the same chromosome.

***alignments.txt* example**

```
read_0    CAGCCATGACACTAAGCACG          15
read_1    TTTAAAAAATCCGTGGACAC          40
read_2    GCATTTAAAAAATCCTTGGA          37
read_3    ATTTCGGCGGCGACACCCCG          0
read_4    TTCGGCGGCGACACACCGAT          2
read_5    ATATTTGGACACAAATGCAT          48
```

Assignment 4: Consensus Region

Logic to build the Consensus regions

Reference genome:

```
ATTTCGGCGGCGACACAGGGATGACACAGGGCACGCAGCATTTAAAAAATTTTTGGACACAGCAGCAT
0         5         10        15        20        25        30        35        40        45        50        55        60        65
```

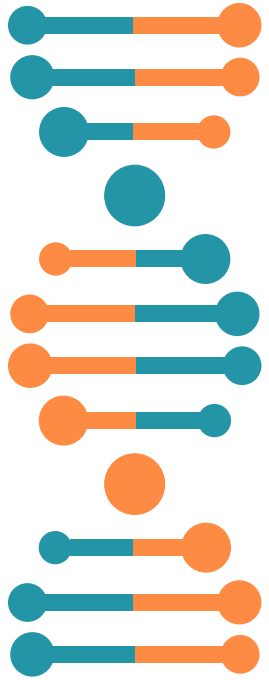
Reads:

```
ATTTCGGCGGCGACACCCCG                          TTTAAAAAATCCGTGGACAC
  TTCGGCGGCGACACACCGAT                    GCATTTAAAAAATCCTTGGA
                 CAGCCATGACACTAAGCACG                 ATATTTGGACACAAATGCAT
```

Consensus regions:

```
ATTTCGGCGGCGACACACCGATGACACTAAGCACG
GCATTTAAAAAATCCTTGGACACAAATGCAT
```

LAB1 – Take home message



- FASTA and FASTQ files usually can take up to 5/30 GB each
- Reads in a FASTA FASTQ file can span multiple lines
- FASTA and FASTQ files should be read line by line, especially if the dimension of the files are not known a priori
- Avoid repetitive and unnecessary instructions when possible (e.g. reading the same file multiple times)
- Optimize your code as much as possible (e.g. avoid unnecessary for loops, printing each line,…)

Questions?

Remember:
no question is
stupid