**Bilkent University**
CS481 | cs583 - Bioinformatics Algorithms

Fall 2022

---

# Homework Assignment #2

---

## GENERAL INSTRUCTIONS

**FAILURE TO FULFILL ANY OF THE FOLLOWING ITEMS WILL RESULT IN A GRADE SCORE OF 0 (zero) WITHOUT ANY CHANCE OF REDEMPTION.**

- You must **write your code yourself**. Sufficient evidence of plagiarism will be treated the same as for plagiarism or cheating.
- **C / C++, Python or Java** will be used as programming language. STL is allowed. The assignments are created with C / C++ in mind, so using other languages would require minimum tweaking.
- Your code must **compile**.
- Your code must **be complete**.
- Your code must **run on dijkstra.cs.bilkent.edu.tr** server.
- Your code must make use of **argument parsing**.
  Refer to: https://docs.google.com/presentation/d/1rU0DhBg6yVXbfEtNuK73pjc1yWSPG9OYnGNH-b-kklQ
- Submit your answers **ONLY** through the Moodle page.
- **Zip** your files and send them in only one zipped file.
- File name format `surname_name_hw#.zip`.
- The zip file must contain the following items:
  - All the source files.
  - `Makefile` to compile the source code and produce the binary. Even if you use Python, include this Makefile.
  - A `README.txt` file that briefly describes how your program works.
- All submissions must be made **strictly before the stipulated deadline**.
- A **bonus** will be given for the fastest code that solves the assignment successfully.

---

## 1) LOCAL AND GLOBAL ALIGNMENT

**Aim:** In this assignment, you will learn about the most common algorithms for sequence alignment, both local and global using Needleman-Wunsch and Smith-Waterman algorithms, respectively.

## 1.1 Data

- **Input**
  1. An option showing if the alignment to be performed is global or local. Given by the switch `-g` or `-l`.
  2. A FASTA file containing the pattern P to search. The file is given using the `-p` switch.
  3. A FASTA file containing the text T to be searched. The file is given using the `-t` switch.
  4. The scoring function for the match, mismatch and gap penalties. The scores are given using the `-s` switch, separated by "`|`".
- **Output**
  1. A text file containing the Dynamic Programming table used for the program and its final values. Given by the switch `-o`.
  2. The Standard Output must print the final alignment at the end of your program and its score.

## 1.2 Tasks

Using **each** data structure, perform pattern matching and along with the output file, print on the standard output, the status of the search while it is being carried out.

## 2) EXAMPLE

The numbers presented in the example are not necessarily correct. They just show how the program should output the information.

## 2.1 Input

```
user@dijkstra$ cat p.fasta
> Pattern P
GATTACA

user@dijkstra$ cat t.fasta
> Text T
GTCGACGCA
```

## 2.2 Execution

```
user@dijkstra$ ./hw2 -g -p pattern.txt -t text.txt -s +1|-1|-2 -o global.txt
GATTA--CA
GTCGACGCA
Score=-3
```

```
user@dijkstra$ ./hw2 -l -p pattern.txt -t text.txt -s +1|-1|-2 -o local.txt
GA
GA
Score=2
```

## 2.3 Output

```
user@dijkstra$ cat global.txt
                G       T       C       G       A       C       G       C       A
        0       -2      -4      -6      -8      -10     -12     -14     -16     -18
G       -2      1       -1      -3      -5      -7      -9      -11     -13     -15
A       -4      -1      0       -2      -4      -4      -6      -8      -10     -12
T       -6      -3      0       -1      -3      -5      -5      -7      -9      -11
T       -8      -5      -2      -1      -2      -4      -6      -6      -8      -10
A       -10     -7      -4      -3      -2      -1      -3      -5      -7      -7
C       -12     -9      -6      -3      -4      -3      0       -2      -4      -6
A       -14     -11     -8      -5      -4      -3      -2      -1      -3      -3
```

```
user@dijkstra$ cat local.txt
                G       T       C       G       A       C       G       C       A
        0       0       0       0       0       0       0       0       0       0
G       0       1       0       0       1       0       0       1       0       0
A       0       0       0       0       0       2       0       0       0       1
T       0       0       1       0       0       0       1       0       0       0
T       0       0       1       0       0       0       0       0       0       0
A       0       0       0       0       0       1       0       0       0       1
C       0       0       0       1       0       0       2       0       1       0
A       0       0       0       0       0       1       0       1       0       2
```