Parallel implementation of Sequence Alignment

Final project

Course 10324, Parallel and Distributed Computation

2020 Spring Semester

Sequence Alignment – a way to estimate a similarity of two strings of letters - is an important field in bioinformatics[1]. Sequence is a string of capital letters including hyphen sign (-), for example

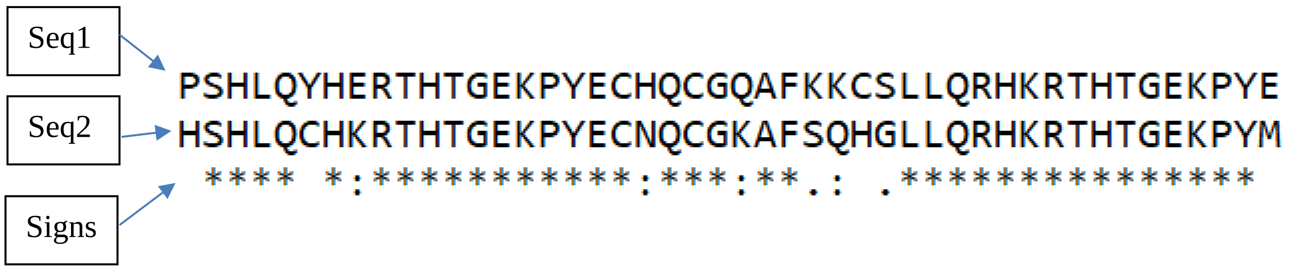
```
PSHLQYHERTHTGEKPYECHQCGQAFKKCSLLQRHKRTHTGEKPYE
```

Each letter in the sequence represents DNA, RNA, or protein. Identification of region of similarity of set of Sequences is extremely time consuming. This project deals with a simplified version of Sequence Alignment of two sequences. The purpose of the project is to parallelize the basic algorithm to produce an efficient computations within MPI, OpenMP and CUDA environment.

Alignment Score Definition with pair-wise comparison

1. Similarity of two sequences Seq1 and Seq2 of equal length is defined as follows:

- Two Sequences are places one under another:



```
PSHLQYHERTHTGEKPYECHQCGQAFKKCSLLQRHKRTHTGEKPYE
HSHLQCHKRTHTGEKPYECNQCGKAFSQHGLLQRHKRTHTGEKPYM
 **** *:***********:***:**.: .***************
```

- Each letter from Seq1 is compared with the correspondent letter from Seq2. If these letters are identical the pair is marked with Star sign (*).

[1] Mount DM. *Bioinformatics: Sequence and Genome Analysis* (2nd ed.). Cold Spring Harbor Laboratory Press: Cold Spring Harbor, NY., 2004

- Otherwise the additional check is provided. The letters are checked if they both present at least in one of 9 groups called Conservative Groups:

NDEQ	NEQK	STA
MILV	QHRK	NHQK
FYW	HY	MILF

In case that the pair is found in one of Conservative Group it is marked with Colon sign (**:**).

For example, the pair (E, K) is marked with sign **:** because they both were found in group **NEQK**

- If no Conservative Group is found, the pair is checked against 15 Semi-Conservative Groups

SAG	ATV	CSA
SGND	STPA	STNK
NEQHRK	NDEQHK	SNDEQK
	HFY	FVLIM

If the pair do presents in one of Semi-Conservative Groups, it is marked with Point sign (**.**).

For example, the pair (K,S) is marked with sign **.** because they both were found in group **STNK**

- If the letters in the pair are not equal, do not present both not in Conservative nor in Semi-Conservative groups – the pair is marked with Space sign (‘ ‘).

At the end of the check process the whole Sequence of Signs is obtained. This Sequence is used to estimate the similarity of two sequences – Seq1 and Seq2. For this project following formula is used to estimate the Alignment Score:

$$S = W_1{*}\text{NumberOfStars} - W_2{*}\text{NumberOfColons} - W_3{*}\text{NumberOfPoints} - W_4{*}\text{NumberOfSpaces}$$

where W_i are the given weight coefficients.

2. **Similarity of two sequences Seq1 and Seq2 in case that Seq2 is shorter than Seq1, is defined as follows:**

   - The Sequence Seq2 is places under the Sequence Seq1 with offset **n** from the start of the Sequence Seq1. The Sequence Seq2 do not allowed to pass behind the end of Seq1.
   - The letters from Seq1 that do not have a corresponding letter from Seq2 are ignored.
   - The Alignment Score is calculated according the pair-wise procedure described above.

For example, Sequence Seq2 is placed at different offsets under Seq1:

Score = -27

Offset n = 5

```
PSHLQYHERTHTGEKPYECHQCGQAFKKCSLLQRHKRTHTGEKPYE
     PYECNQCGKAFSQHGLLQRHKRTHTGEKPYM
      :* .: *:   :     :  :. :  : :
```

Score = 17

Offset n = 15

```
PSHLQYHERTHTGEKPYECHQCGQAFKKCSLLQRHKRTHTGEKPYE
               PYECNQCGKAFSQHGLLQRHKRTHTGEKPYM
               ****:***:**.: .***************
```

Mutant Sequence Definition

For a given Sequence S we define a Mutant Sequence MS(n) which is received by insertion of hyphen (-) after n-th letter in S. For example, for a Sequence S = PSHLQY a set of 6 Mutant Sequences

```
MS(1) = P-SHLQY
MS(2) = PS-HLQY
MS(3) = PSH-LQY
MS(4) = PSHL-QY
MS(5) = PSHLQ-Y
MS(6) = PSHLQY-
```

Problem Definition

Let Seq1 and Seq2 be a given Sequences of letters.

For all Mutant Sequences of Seq2 find an offset **n** which produce a maximum Alignment Score against Seq1. Among all results, choose the Mutant Sequence MS(**k**) with the best Alignment Score.

For example, for Seq1 = PSHLQY and Seq2 = SHQ, the Mutant Sequence MS(2) = SH-Q produce the best Alignment Score at offset n = 1

```
PSHLQY
 SH-Q
```

Input data and Output Result of the project

You will be provided with the input file **input.txt** with Weight Coefficients, Sequence Seq1 and few Sequences Seq2. The data is taken from[2]

The output must be written to file **output.txt.** For each Sequence Seq2 from input file provide an offset **n** of the Mutant Sequence MS(**k**) with the best Alignment Score and location of the hyphen sign (-) of this Mutant Sequence

Input File format

First line - weight coefficients W_1, W_2, W_3, W_4

Next line – Seq1 (not more than 3000 chars in line)

Next line – the number **NS2** of Sequences Seq2 to check against Seq1

Next **NS2** lines - Seq2 in each line (not more than 2000 chars in each line)

Output File format

This file contains **NS2** lines with **n** and **k** found for each Sequence Seq2 from the input file, in order Sequences Seq2 appear in the input file, where **n** is an Offset and **k** is a hyphen location of the Mutant Sequence MS(**k**) with the best Alignment Score.

[2] https://www.ncbi.nlm.nih.gov/genbank/

Requirements

- Implement the Simplified Sequence Alignment algorithm explained in the class (see above).
- The input file **input.txt** initially is known for one machine only. The results must be written to the file **output.txt** on the same machine
- The computation time of the parallel program must be faster than sequential solution.
- Be ready to demonstrate your solution running on two computers (if MPI is used)
- **No code sharing between students is allowed.** Each part of code, if any, which was incorporated to your project must be referenced according to the academic rules.
- Be able to explain each line of the project code, including those that was reused from any source.
- **The project that is not created properly (missing files, build or run errors) will be not accepted**

Grade Policy

- **60 points** for the effective **proper** parallel implementation of the problem with two components: ***MPI+OpenMP*** or ***OpenMP+ CUDA*** or ***MPI+CUDA***. The project that produce wrong results will not be accepted.
- **10 points** for implementation in ***MPI+OpenMP+CUDA*** configuration.
- **10 points** for the documentation of your solution – clear explanation what and how the problem was parallelized, what is a rational of choosing the specific architecture, complexity evaluation.
- **10 points** for the code quality – modularity, generality, self-explanatory, organization.
- **10 points** for the Load Balancing.

Additional Bonus for the project grade

5 points for implementation with OpenCL instead of CUDA

5 points for managing Mutant Sequences with more than one hyphen sign (discuss with the lecturer)

5 points for implementation of sophisticated variation of the algorithm (must be approved by lecturer).

5 points for your own proposal (must be approved by lecturer).

הפרויקט יוגדר כמטלת הקורס. הגשת התוכנה והתיעוד רק דרך מערכת Moodle לאחר ההגנה.

יישום והגשת הפרויקט <u>ביחידים בלבד</u>.

בהצלחה