Set of Exercises

✓ **Main bibliographic sources:**

Algorithms on Strings, Trees and Sequences, D. Gusfield, Cambridge University Press, 10th edition 2007

Bioinformatics Algorithms: An Active Learning Approach by Phillip Compeau, Pavel Pevzner (2018), third edition, Active Learning Publishers

Arthur Lesk, Introduction to Bioinformatics, Oxford University Press, Fifth Edition, 2019

Jonathan Pevsner, Bioinformatics and Functional Genomics, 3rd Edition, Wiley 2015

Gonzalo Navarro, Compact Data Structures: A Practical Approach (2016), https://www.cambridge.org/core/books/compact-data-structures/68A5983E6F1176181291E235D0B7EB44 Biological Modeling: A Short Tour, January 25, 2023, by Phillip Compeau (Author, Editor), Mert Inan (Author), Noah Lee (Author), Shuanger Li (Author), Chris Lee (Author).

https://biologicalmodeling.org/

https://rosalind.info/problems/locations/

✓ **For implementation (where required) you can use any language you prefer. It is preferable to use Biopython (https://biopython.org/, https://en.wikipedia.org/wiki/Biopython)**

✓ **Exercises without a marking contribution (sub-question (iii) in exercise 2 and non-mandatory question in exercise 6) do not count towards the evaluation, they are just for consideration.**

✓ **In exercises 3 and 4 it is sufficient to investigate the relevant papers and techniques in 1-3 pages, while in questions 1 and 2 it is sufficient to investigate the relevant tools in 1-3 pages.**

Exercise 1

(i) The objective of this exercise is to investigate various software tools for handling bioinformatics problems. More specifically, you should investigate **(not solve)** the examples of the software tools listed on the page https://rosalind.info/problems/list-view/?location=bioinformatics-armory of Rosalind (https://rosalind.info/problems/locations/) and for each of them make a small report concerning their usage.

(ii) There are many freely accessible tools for multiple sequence alignment. In this report you will make a comparison of the tools in the NCBI and EBI Databases. Visit the NCBI and EBI website and report the key features of their multiple-alignment tools. For NCBI the key tools are in links: https://www.ncbi.nlm.nih.gov/projects/msaviewer/, https://www.ncbi.nlm.nih.gov/tools/cobalt/re_cobalt.cgi and for the EBI Sequence Manipulation Suite on the website https://www.ebi.ac.uk/jdispatcher/msa/ which ensures access to a large number of tools.

Hint: It is enough as a result TO REPORT SIMPLE USE OF THE VARIOUS TOOLS, NOT TO SOLVE THE MENTIONED PROBLEMS. That is, the purpose of the exercise is to get in touch with some ready-made tools NOT THE EXPERIENCED USE OF THEM.

Exercise 2[1]

(i) Access the NCBI database to study the SARS-CoV-2 coronavirus at the link https://www.ncbi.nlm.nih.gov/sars-cov-2/. Use the record with sequence data for SARS-CoV-2 https://www.ncbi.nlm.nih.gov/nuccore/NC_045512 to download the coronavirus spike protein sequence. Then from the link https://www.uniprot.org/uniprotkb/A0A6B9WHD3/entry download the Bat-RaTG13 coronavirus spike protein sequence (https://en.wikipedia.org/wiki/RaTG13) **and implement the classic dynamic programming global alignment algorithm with appropriate weights to identify their maximum common subsequence.** Report the final result.

(ii) View the structure of the two proteins of the previous query using the ab-initio swiss-modeller tool (https://swissmodel.expasy.org/interactive) and download the .pdb files (a textual file format describing the three-dimensional structures of molecules held in the Protein Data Bank (textual file of three-dimensional structures of in Protein Data Bank)). Then compare the structures of the two proteins using the Dali tool at http://ekhidna.biocenter.helsinki.fi/dali/. Make your observations about the correlation of sequences and structures.

Sub-question (iii) (sub-question without scoring contribution): if someone wants to delve deeper, they he can visit https://biologicalmodeling.org/coronavirus/home website with similar (but not identical) questions.

Subquestion (iv) (sub-query without scoring contribution): try to solve the protein structure prediction problem with various new machine learning (https://www.nature.com/articles/s41592-023-01790-6) algorithms such as **AlphaFold** (https://alphafold.ebi.ac.uk, https://www.ebi.ac.uk/Tools/sss/fasta/, https://colab.research.google.com/github/deepmind/alphafold/blob/main/notebooks/AlphaFold.ipynb) and **ESMFold** (https://www.science.org/doi/10.1126/science.ade2574, https://esmatlas.com/resources?action=fold).

Exercise 3 (research)

---

[1] Jonathan Pevsner, Bioinformatics and Functional Genomics, 3rd Edition, Wiley 2015

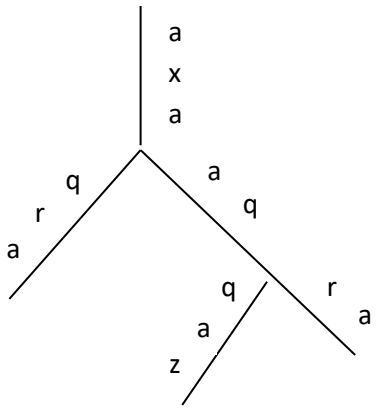
Let T be a generalized suffix tree for a set of k strings. Strings can be added or subtracted from this set. Describe the problems that arise for the dynamic preservation of this structure. Provide algorithms to support these operations.
Hint: search the world wide web (with search engines or digital library tools) with the query "dynamic suffix trees/dynamic string matching" and combine the results mentioned there**.**

Exercise 4 (research)
You are given a collection of *k* (*k>2)* sequences. Demonstrate an algorithm that detects ***repetitions*** (i.e. the occurrence of the same string twice) in each sequence, where the repeated string is the *same* in all sequences. Consider the problems that arise if we try to put constraints on the gaps between the two occurrences of the string in each sequence.
Hint: Use of a generalized suffix tree, explore relevant online solutions and the use of techniques with k-mers.

Related Bibliography:

- ✓ Gerth Stølting Brodal, Rune B. Lyngsø, Christian N. S. Pedersen, Jens Stoye: Finding Maximal Pairs with Bounded Gap. CPM 1999: 134-149
- ✓ Gerth Stølting Brodal, Christian N. S. Pedersen: Finding Maximal Quasiperiodicities in Strings. CPM 2000: 397-411
- ✓ A. Bakalis, Costas S. Iliopoulos, Christos Makris, Spyros Sioutas, Evangelos Theodoridis, Athanasios K. Tsakalidis, Kostas Tsichlas: Locating Maximal Multirepeats in Multiple Strings Under Various Constraints. Comput. J. 50(2): 178-185 (2007).

Exercise 5[2]
Consider a tree where each edge is labeled with one or more characters and a pattern P. Provide an algorithm that identifies all subpaths that start at the root and contain the pattern. Notice that although the subpath must be part of a path that starts at the root, the subpath itself does not have to start at the root (see scheme). Provide an algorithm for the specific problem running at a time proportional to the total number of characters at the edges of the tree plus the length of the pattern P.



Scheme*: the pattern P=aqra exists in two subpaths, starting at the root. These paths start at the root, but subpaths that contain the string aqra do not (there is also another subpath in the tree that is labeled aqra, and starts above the z character, but the requirement that it is a subpath starting at the root is violated).*

[2] Dan Gusfield, Algorithms on String Trees and Sequences, Cambridge University Press

Hint: Use generalized suffix tree

Exercise 6. Python exercise in bioinformatics.
Write a program using the Python programming language that will search for binding areas of the following **transcription factors** (Transcription Factor - TF). It will then display the positions of these areas.

Transcription Factor	Consensus Sequence
RUNX1	BHTGTGGTYW
TGIF1	WGACAGB
IKZF1	BTGGGARD

The sequence in which the search will be done is shown below. You can place it in a .fasta file and manage it with the help of Biopython.

- *Display occurrence positions for each factor and save them to the appropriate file.*
- *Present sequence statistics.*
- *Save in a file the statistics you found (number of bases, percentage of CG in relation to the sequence).*
- *Create a file that depicts the complementary sequence*

It is recommended to use string, Biopython, or re libraries (*which allows regular expressions*)

```
>Sequence

GACACCTCAGTACTAGGATGTATCAGCCTGAACTAGCAGGCCTGGTTCCAAATTTTTTTATCAACACTCG
TAGGGGGATTATCCTAGAGGGGGTCTGGGATTTCTTTGACATCAGAGTATTTTTGCCTTGCTCCTTCACA
ATTTGGGAACAAATAATTTAGTGGTTATTAACCCTGGCTACGCACTGGAAACTTTAAAAATAATGCTGGT
ATGAAATTTACACAGAGTATCGTGAAAATTTTCACTGAGTACCATGTGGTTATACATTGGATAAGGCTCC
AGGAAGCAGCTACTGGAAGACAGCCATGCCAAGAGTGGTTAGTGGTTGGAATTTTGGCAAGTCAGTTTTA
GTCTGCCTTATCAAATACATGGGCATACAGATAAATCCTTAGATGGCTCTCCTACTTACTGAAACATTTT
CTATCTATCTATCTATCTATCTATCTATTTGGGAAGCTATCTATCTATCTATCATTTATTTAAGGTAGTC
TCTATCTGCCTCTGTCTCTGTCTGTCTCTGTGTCTCTGTGTCTGTCTGCTCTCTCTCTCTCTGTGGGA
ATCTCTCTCTGTGTGTGTGTGTGTATGTGTGTGTGTGTGTGTGTGGTGTGCATGAACATGAGTAAAATCC
ATAAGGAAACTTTCAGAGTTGGTCCTCTCCTTATATCAAATGGATCCAGGAATTAAACTCAGGTTCAATT
CTTGGTGCCTTTACTAGTTGAGCCATCTCACTGGCTCTTCATCATCTTTAGAATAAACTCACTTTATTAC
ACACACACACACACACACAACCTGGGAGTACACACACACACACAACCAAAGCCCCAACGGAAAACTACAA
TATTATAATGAATACACAGGTTCTCAACATAGTCTCTGCCACGCTTGCAGACAAAGATGAGTAGAAGTAG
AAAGAACCAGGGAAACGTGGAGCAAGTCAGAAGGAATAACAGTCAGAAGGAATAACAGTCAGAAGGAATA
ACAGTCAGAAGGAGTAACAGTCAGAAGGAATAGCAGTCAGAAGGAATAACAGTCAGAAGACAGCACAGTC
AGAAGGAATAACAGTCAGAAGGAATAACAGTCAGAAGGAATAACAGTCAGAAGGAATAACAGTCAGAAGG
AATAGCAGTCAGAAGGAATAACAGTCAGAAGGAATAACAGTCAGAAGGAATAACAGTCAAAGAAATAGCA
GTCAGAAGGAATAGCAGTCAGAAGGAATAACAGTCAAAGGAGCAGTCAGAAGGAGTAACAGTCAGAAGGA
ATAACAGTCAGAAGGAATAACAGTCAAAGGAATAGCAGTCAGAAGGAGTAACAGTCAGAGCAAACACAGA
GATGACAAAGGCAATGGGGTCAGAGACTTCACCACTCTCCAAGATCTACTATATACTCTCTCTGTGT
```

You will notice that the sequences to consider include not only the bases A,T,G,C but also other characters. These are called ambiguity codes and are shown in the following table[3].

[3] https://www.dnabaser.com/articles/IUPAC%20ambiguity%20codes.html

Code	Represents
A	Adenine
G	Guanine
C	Cytosine
T	Thymine
Y	Pyrimidine (C or T)
R	Purine (A or G)
W	weak (A or T)
S	strong (G or C)
K	keto (T or G)
M	amino (C or A)
D	A, G, T (not C)
V	A, C, G (not T)
H	A, C, T (not G)
B	C, G, T (not A)

NON-MANDATORY PART OF THE EXERCISE[4]

- Suffix Tree application for search and time comparison with naïve algorithm.
- Can we apply KMP, Boyer-Moore;
- Performance comparison of regular expressions with exact matching algorithms.

[4] It doesn't count for evaluation