Algorithms for genomic data analysis
Assignment 2: metagenomics sample classification

winter semester 2025/2026

Task

Implement an algorithm that will generate sketches of the provided large metagenomics samples from various environments, and then use the resulting sketches to classify new, smaller samples from these environments.

The training and testing data originate from DNA sequencing of metagenomic samples collected by the MetaSUB (Metagenomics and Metadesign of Subways and Urban Biomes) consortium. The intended application is in forensics: to classify samples collected from objects used in criminal activities.

Your program can be based on any approach to generate sketches from sets of sequences. In your program code you can use:

- code fragments from classes,
- libraries included in standard Python distribution, as well as NumPy, SciPy, Biopython and mmh3.

In your program you cannot use:

- programs and libraries to manipulate k-mer profiles, extract minimizers, generate sequence sketches etc.
- multiprocessing commands,
- subprograms written in other languages,
- commands using jit compilers.

The solution should include:

- program file `classifier.py` written in Python 3,
- slides with a short description of your approach.

For a fair comparison, please avoid submitting your solution with a packaged virtual environment.

Specification

Program should be executable using the following syntax:

```
python3 classifier.py training_data.tsv testing_data.tsv output.tsv
```

The first of the input files consists of two or more tab-separated columns. After the header line, each line corresponds to a dataset, with the name of the gzipped FASTA file in the first column, and the dataset class (name of the city from which the sample was derived) in the second column. Multiple datasets may have the same class. Further columns could be ignored.

The second of the input files follows a similar format, but only the first column is relevant, indicating names of the gzipped FASTA files containing the testing datasets to be classified.

The reads in FASTA files will be around 150 bp each. The full-size training data contain 20 datasets with 1 M sequencing reads, and the testing data – 100 datasets with 100 k sequencing reads. Your solution will be also evaluated using extended training data (containing the above 20 datasets plus 20 similar ones) and new testing data (containing 100 independent datasets similar to the ones above).

The output file should also be tab-separated, with a header line and one line for each testing dataset. The first column should indicate names of the FASTA files. Values in subsequent columns should quantify how likely the given dataset has each possible class indicated in the header line. These values do not have to be probabilities, but higher values should indicate higher probability.

The values returned by your classifier will be used to evaluate the solution through AUC-ROC. More specifically: for each possible class, the values returned for each dataset will be compared against the ground truth (a vector of 0s and 1s) to calculate the sensitivity and specificity of the classifier at all possible thresholds. The area under the Receiver Operating Characteristics curve will be averaged across all the classes.

Minimum performance requirements are following: your program should reach AUC-ROC ≥ 0.6 on the data provided, in the time limit of 5 hours (on the `students` server) and memory limit of 1 GB.

Additional items

A 2-person team is encouraged to implement one additional item, and a 3-person team – two additional items. You may wish to implement more of them. All items involve some kind of resampling; please *repeat the sampling multiple times* to estimate the variance, and plot the results in an informative manner (e.g. mean ± s.d.).

- Compare the performance of your solution when the *training* data are downsampled at various ratios, e.g. taking 10–100% of each training dataset.
- Compare the performance of your solution on downsampled *testing* data at various ratios, e.g. taking 10–100% of each testing dataset.
- Evaluate your solution on different subsets of 100 k reads sampled from the original larger datasets (available from `https://www.ncbi.nlm.nih.gov/sra` through their SRR*nnnnnnnn* identifiers).

Attached files

- The archive contains example input and output .tsv files, small training and testing FASTA files (downsampled to 1 k sequences), ground truth for the testing files, and a program to evaluate the classification.
- Full-size training and testing FASTA files are on the `students` server, in `/home/staff/iinf/ajank/adg`.

Terms and conditions

The assignment can be completed individually or in 2- or 3-person teams. Schedule:

- Submit your team by email to aleksander.jankowski@uw.edu.pl till November 31.
- Submit your solution to Moodle: code till December 14, slides till December 15.
- Present your approach in class on December 16.

Assessment

Every solution that meets the minimum requirements receives 2 points and can get additional points for

- classification performance on the provided training and testing data:
  2 points if average AUC-ROC ≥ 0.75, **1 points** if ≥ 0.7,
- classification performance on extended training and new testing data:
  4 points if average AUC-ROC ≥ 0.75, **3 points** if ≥ 0.7, **2 points** if ≥ 0.65, **1 point** if ≥ 0.6,
- runtime:
  3 points if ≤ 1 minute per 1 M reads, **2 point** if ≤ 2 minutes, **1 point** if ≤ 5 minutes,
- meeting deadlines and presentation quality:
  up to **2 points**,
- additional items:
  1 point per additional item
- team size:
  2 points 1 person, **1 point** 2 persons.

but not more than 15 points in total.