Prof. Dr. Knut Reinert, Simon Gene Gottlieb

January 6, 2025

Introduction to Focus Areas

Winter term 23/24

Advanced Algorithms - Assignment 1
Due 2025-01-13, 10:00am

1 Implementing A Search

Imagine a fictional world in the future. A Virus has been spreading the world. You are working together with Virologists on a vaccine. To check if this vaccine is working the virologists need to find certain markers in the human genome. Your task is to evaluate different algorithms under python (easy) and c++ (fast) and figure out which one is the best.

You are given a reference text, which has parts of the first chromosome of the human genome 'hg38_partial.fasta.gz'. You also got a list of markers 'illumina_reads_XYZ.fasta.gz'. You need to figure out where these markers appear in the reference. Speed matters!

For your implementation in Python it is recommended to use `https://iv-project.github.io/IV2py`. For the c++ implementation code template can be found at `https://github.com/SGSSGene/ImplementingSearch`.

Hurry! The time window is closing!
Sincerely,
Humans of Earth

1. Give your group an awesome name. (A name like "Team-O(1)" or "DnaExtractornator").
2. Implement a naive search algorithm (don't use an index). (Python and C++)
3. Implement a suffix array based search. (Python and C++)
4. Benchmark (runtime and memory) your solutions for 1'000, 10'000, 100'000 1'000'000 queries of length 100.
5. Benchmark (runtime) queries of the length 40, 60, 80, and 100 with a suitable number of queries.

Deliverables:

1. Upload your source code (only the files you changed) to the Whiteboard or provide link to a public repo.
2. Give instructions on how to run your python code.
3. Create a report, show background, your methods, implementations details and benchmark results. Please include a section if and how you used AI.

Some hints:

1. Familiarize yourself with the FU-Berlin servers.
   (a) ssh - `https://github.com/seqan/seqan3/wiki/SSH` (connect to compute03.mi.fu-berlin.de)
   (or `http://www.mi.fu-berlin.de/w/IT/ItServicesSSHAccess` and `http://www.mi.fu-berlin.de/w/IT/ComputeServer`)
   (b) tmux - https://github.com/seqan/seqan3/wiki/tmux
   (c) Which editor? vim, emacs, nano .... or just scp to target?
2. When benchmarking suffix array based search method, separate the runtime of your index construction.
3. For memory consumption you can use '/usr/bin/time -v ./yourprogram'