BINF200 Assignment 1

Pairwise sequence alignment, database search, gene expression

2023-09-04

1 Deadline, grading and report

The assignment is due **15 September 2023**.

The assignment is scored on **20 points** and counts towards **10% of the final grade**.

Your report should be a **single PDF file** that contains your report text, code, and figures in a single document. The easiest workflow is probably to run your analyses in a Jupyter or similar notebook, and save the final notebook as a PDF file.

You *may* work together, but you *must* declare it in your report. (You *cannot* work together if you are assigned to the *same* assignment group, see Section 4 below, but this affects only 2 people and can be fixed if required.)

Any use of ChatGPT or other generative AI tools *must* be declared in your report.

2 Background

In this assignment you will analyze data from the following paper:

S. Jayaraman *et al.* (2019), Application of long read sequencing to determine expressed antigen diversity in Trypanosoma brucei infections, PLOS Neglected Tropical Diseases 13(4): e0007262.

You will need a copy of the paper to answer some of the assignment questions.

The paper has an accompanying Github repository where you will find some useful commands.

3 Software

You will need the following software to solve the assignment:

1. **BLAST+**: Install the NCBI command line standalone BLAST+ programs following either the online instructions, or:

   - On Windows, installation in the Windows Subsystem for Linux will be easier than following the above instructions.
   - For Ubuntu (and hence also WSL), there exists a package "ncbi-blast+" not listed on the NCBI page, which you can install directly using `apt-get install`.

2. **GetORF**: Install the getorf tool. The installation instructions are a bit tedious, but you should get there in the end. Again, on Windows, installation in the WSL will be easier.

   - If at the configure step you get an error about "X11", then rerun the `configure` command with option `--without-x`

3. **Programming language**: You may use any language: Python, Julia, R, …

4. **Packages for reading FASTA-formatted files and importing and manipulating DataFrames**: Find and install them for your preferred programming language.

5. **Data analysis and visualization, report writing**: You can use JupyterLab (or Jupyter notebooks) and/or an IDE such as Visual Studio Code for data analysis and visualization. Using Quarto you can integrate your code, data visualization, and final report all in a markdown document. Pluto notebooks for Julia offer nice reactive features.

4 Sequence data

All data for the assignment are available on UiB OneDrive in the **PacBio VSG** folder. You will need to login with your UiB account to obtain access. Find and download the following files:

1. A database of known VSG genes. Download the entire folder named

   TREU927-v26_VSGTranscripts.

2. Sequencing data from **one** sample (**one file**) named

   PacBio_VSG_filtered_reads_*sample_name*.fasta.

   Twenty such files, for 20 samples (individual mice), are available in total. To find out which one *you* should download:

   1. Go to the **Groups** page on Mitt (https://mitt.uib.no/courses/42444/groups)

2. Select the **Compulsory Assignment 1 - Sample** tab and find to which group you have been (randomly) assigned.
3. Each group is named **Compulsory Assignment 1 - Sample *k***, where k is a number from 1 to 20 mapping to the sample names as follows:

Table 1: Mapping of group labels to sample names

k	Sample name	k	Sample name
1	balbc_3_0	11	balbc_10_0
2	balbc_3_1	12	balbc_10_1
3	balbc_3_2	13	balbc_10_2
4	balbc_3_3	14	balbc_10_4
5	balbc_3_4	15	balbc_10_5
6	balbc_6_0	16	balbc_12_1
7	balbc_6_1	17	balbc_12_2
8	balbc_6_2	18	balbc_12_3
9	balbc_6_4	19	balbc_12_4
10	balbc_6_5	20	balbc_12_5

5 Tasks

5.1 Briefly describe your programming experience (optional, ungraded)

Briefly describe your experience with:

- The Unix command line
- Python programming and package management
- Data input/output and DataFrames in Python
- Jupyter notebooks, VS Code

If you are comfortable with all of the above and did not experience any particular problems during the software installation or reading and processing of the sequence data, you can skip this task.

5.2 Show that you understand the BLAST+ package (2 points)

The BLAST+ package contains five **core blast search programs**.

1. List all five core blast search programs.
2. Explain the difference between **blastn** and **blastp**.

5.3 Show that you understand the biological experiment and data (2 points)

Write a short paragraph in your own words to explain the biological experiment that was done to generate the sequence data (see the publication), what sequences are contained in the **PacBio_VSG_filtered_reads_*sample_name*.fasta** files, and what sequences are contained in the **TREU927-v26_VSGTranscripts** database.

Answer the following question: What is the number of sequences in *your* input file? **Hint:** Most popular programming languages have packages that can parse FASTA files automatically.

5.4 BLAST the sample sequences against the reference VSG database (4 points)

Find the relevant BLAST command on the longread-application repository. Then adapt this command in the following ways:

1. Change the input file name to *your* assigned sample file (see Table 1).
2. Change the output file name to something containing *your* sample label (see Table 1).
3. What does the paramater `-max_target_seq` do and why was it set to 1?
4. Use a tabular output format *without* comments that includes *only* the following columns:
    - Query sequence id
    - Subject sequence id
    - Raw score
    - Bit score
    - E-value
    - Query sequence length
    - Subject sequence length
    - Alignment length
    - Start of alignment in subject
    - End of alignment in subject
    - Number of identical matches
    - Number of mismatches
    - Total number of gaps
    - Percentage of positive-scoring matches

Include the exact BLAST command you ran in your report.

5.5 Analyze the BLAST output (5 points)

Answer the following questions / perform the following tasks:

1. What is the number of sequences in *your* blast output file? Is it the same as in the input file? **Hint:** Import the BLAST output file into a DataFrame.
2. We define the alignment coverage as the percentage of the subject sequence covered by the alignment. Compute the alignment coverage for all sequences from the BLAST output.
3. Visualize the distribution of alignment coverages as a histogram.
4. Remove alignments with coverage less than 60% and verify that each query sequence is now aligned to at most one subject sequence.
5. The bit score S' is derived from the raw score S using the formula
$$S' = \frac{\lambda S - \ln K}{\ln 2}$$
Can you find the values of λ and K from your BLAST results? **Hint:** Plot the bit scores against the raw scores.

5.6 Count VSG expression levels (4 points)

Perform the following tasks:

1. Extract the unique VSG ids in *your* sample from your (filtered) BLAST results.
2. For each unique VSG: count the number of sequences aliging to that VSG, and the average number of identical matches, mismatches, and gaps for its alignments. **Hint:** Using the split-apply-combine strategy, these numbers can be computed in one line of code.
3. Identify the 10 most abundant VSGs in *your* sample, visualize their relative expression levels in a pie chart, and compare your result against Figure 3 of the paper.
4. Write a paragraph in your report that describes the figure and your interpretation of it.

5.7 Identify open reading frames (3 points)

Perform the following tasks:

1. Find the relevant `getorf` command on the longread-application repository and adapt it to work on *your* input sample. Include the exact `getorf` command in your report.
2. Count the percentage of reads in *your* sample that result in a predicted ORF with a minimum size of 1200 nucleotides.
3. Which explanation was proposed in the paper for this low percentage?

6 Bonus questions (ungraded)

If the previous tasks were too easy or you would like to have more fun with the data, try the following (ranked by increasing difficulty):

1. Repeat your analysis for a different sample and compare the results.
2. Reproduce Figure 2B,C from the paper using a day 3 or day 6 sample. This will require a new BLAST command where you keep the full alignment to compute the number of identical matches, mismatches, and gaps (deletions) *per nucleotide*.
3. Reproduce Figure 5 from the paper using a day 12 sample. This will require a new BLAST command where you keep *all* alignments for each query sequence and a few post-processing steps to identify likely mosaics as described in the paper.