236522 – Algorithms in computational Biology
Alternative assessment – Spring 2024

Genome Assembly Using Overlap Graphs*

Objective:

To develop a program that assembles the PhiX genome by simulating reads, generating overlap graphs, and measuring the assembly's performance under different error conditions.

Instructions:

1. Download Genome Sequence

Download the PhiX genome as a FASTA file from NCBI

2. Generate Error-Free Reads

Write a program to generate N error-free reads of length l from the PhiX genome. These reads should be randomly selected, covering the genome as uniformly as possible.
Note: NGS outputs usually contain $100 \leq N \leq 1{,}000{,}000$ reads of length $50 \leq l \leq 150$.
When generating the reads, consider the average coverage of the genome by the reads.

3. Generate Error-Prone Reads

Extend your program to generate N error-prone reads of length l, where each base has a mismatch probability p to simulate sequencing errors.
Note: The error probability in NGS outputs is usually in the range $0.001 \leq p \leq 0.1$

4. Assembly Using Overlap Graphs

Implement a method to assemble the generated reads using overlap graphs:

- Construct an overlap graph where nodes represent reads, and edges represent overlaps between reads.
- Traverse this graph to generate contigs or reconstruct the genome.
- Consider implementing a naïve solution and adding performance improvements

5. Testing the Assembly

Run your code to test the assembly algorithm on both error-free and error-prone reads.
You may want to start with a synthetic "toy genome" that you can control its size and complexity before moving on to the true PhiX genome.

6. Performance Measures

Define and compute the following performance measures for assembly quality. Explain your measures.

7. Experimentation with Parameters

Run experiments to evaluate how varying parameters N, l, and p (and any additional parameters you find relevant to your implementation) affect assembly performance. Document your findings.

8. Report

Prepare a comprehensive report as if for an academic paper, including:

- Abstract: Summary of the task and main findings (1-2 paragraphs).
- Introduction: Brief description of genome assembly, overlap graphs, and problem significance.
- Methods: Explain the technical aspects of your implementation, parameters used, and link to the code repository.
- Results: Present results of experiments with relevant plots/tables for clarity.
- Discussion: Summarize key insights, limitations, and suggestions for future improvements.
- References: Cite all external sources used in your work.

Submission Requirements

Upload your code as a GitHub repository.
Submit the report as a PDF, formatted as an academic paper.

Grading Criteria

The assignment will be assessed based on correctness, complexity, coding style, design and performance of the experiments and presentation.

Good luck!

* This assignment is inspired by the Coursera course:
Genome Assembly Programming Challenge