IITM-CS6024 : Algorithmic Approaches to Computational Biology Release Date: Aug 5

Assignment 1 Due Date : Aug 13, 23:55

Roll No: BE16B002 Name: Anoubhav Agarwaal

References: Q1. Eulerian path, Q2. K-Universal circular string

1. (6 points) [Choose your DS!]: We saw the algorithm below to construct an Eulerian cycle in a graph (with $n$ nodes and $m$ edges). What data structures would you choose to implement Lines 4,5,6 in the algorithm below, so that the overall algorithm has $O(m)$ running time complexity? Please assume that Graph is a balanced and strongly connected directed graph provided as an adjacency list.

```
1 EULERIAN_CYCLE(Graph)
2   form a cycle Cycle by randomly walking in Graph (don't revisit same edge!)
3   while there are unexplored edges in Graph
4     select a node newStart in Cycle with still unexplored edges
5     form a cycle Cycle' (starting at newStart) & then randomly walking
6     Cycle <- merge(Cycle, Cycle')
7   return Cycle
```

**Solution:** I would use a **stack** to implement the following lines, to ensure a time complexity of O(m). The basic algorithm using a stack is as follows:

1. Check whether there is a eulerian circuit in given graph.
2. Initialize a stack. Push a start vertex to stack.
3. While the stack is not empty:
   - 3.1. Obtain the vertex V at the top of stack.
   - 3.2. If degree of vertex V is zero:
     - i. Add V to the eulerian path.
     - ii. Pop the vertex V from stack.
   - 3.3. Else:
     - i. Pick any outgoing edge from vertex V
     - ii. Remove the edge from the graph
     - iii. Push the second end of this edge to the stack

2. (6 points) [Counting ambiguity]:
   (a) (3 points) Find a simple DNA sequence whose k-mer composition agrees with that of exactly 5 other DNA sequences. (Hint: What would its de Bruijn graph look like? Optional: Do "articulation points" help identify contigs in this graph, or do you need "maximal non-branching paths" as mentioned in the book/class?)

(b) (3 points) How many 3-universal circular binary strings are there? Justify. (Optional: How would you count the number of k-universal circular strings?)

**Solution:** There are **two** 3-universal circular binary strings. These are obtained by finding the unique circular Eulerian cycles in the De Bruijn Graph of all possible binary 2-mers by enumeration. The two strings are: 00010111 and 11101000. The complementary sequence of a De Bruijn sequence is also a De Bruijn sequence.[1]

3. (11 points) [Coding warmup]:

   (a) (8 points) Implement an algorithm to construct a de Bruijn graph from a $k$-mer collection as specified in the HackerRank contest at `https://www.hackerrank.com/assignment-1cs6024`. (Note: Create your `https://www.hackerrank.com` account using your smail id, and ROLL NUMBER as your user-name. Languages supported are C, C++, Java, Python, and R (use only one)).

   (b) (3 points) In addition, write here in this document the key data structure you used to implement the code, and associated time and space complexity of your implementation. Does you algorithm work for any value of $k$?

**Solution:** The key data structure I have used is a **list of tuples** containing the ends of each edge. The algorithm works for any length $k$. The algorithm is divided into three parts:

| Operation | Time complexity | Space complexity |
|---|---|---|
| Creation of edge list | O(n) | O(2*(k-1) *n) ≈ O(n) |
| Sorting of edge list | O(nlogn*2*(k-1)) ≈ O(nlogn), | in place |
| Printing of edge list | O(n) | No additional space |

For small k, Overall time complexity: $O(nlogn)$; Overall space complexity: $O(n)$

4. (11 points) [Dissecting a food-poisoning outbreak] Our campus doctor has obtained bacterial samples likely behind a recent food-poisoning outbreak in the campus (hypothetical of course!). As a bioinformatician, you are asked to assemble the genome of this bacteria from a set of whole-genome sequence reads obtained from these samples with an Illumina sequencer (see attached campusbacteria_R[12].fastq files). You can use Velvet, which is one of a number of $de\ novo$ assemblers that uses short read sets as input and construct de Bruijn graphs for genome assembly, inside the Galaxy framework.

   You can use any of these Galaxy Servers that host Velvet Assembler.

   - `https://usegalaxy.org.au/`
   - `http://bf2i-galaxy.insa-lyon.fr:8080/`
   - `https://usegalaxy.eu/`

   Helpful Link: `https://galaxyproject.github.io/training-material/topics/assembly/tutorials/general-introduction/tutorial.html` for genome assembly workflow of a different data.

(a) (2 points) What is the average number of wrong base calls in the last 5 positions of the $1^{st}$ read in campusbacteria_R2.fastq file? (Hint: Use quality scores provided by the Illumina 1.9 Next Generation Sequencer (NGS).

**Solution:** In Illumina 1.8+ the raw reads lie in the range(0, 41). The Phred score is the ASCII value - 33. The last 5 positions of the 1st read in R2.fastq are **D*CF2** Their respective phred scores are 35, 9, 34, 37, 17. A Phred score of a base:

$$Q_{phred} = -10 * \log_{10} e$$

where $e$ is the estimated probability of a base being wrong. The average number of wrong base calls in the last 5 positions of the 1st read is its expectation value. This is given by: $\Sigma\, e * no.of\ bases.$

$$Avg = 3.16 * 10^{-4} * 1 + 0.1259 * 1 + 3.98 * 10^{-4} * 1 + 1.995 * 10^{-4} * 1 + 0.01995 * 1$$

$Avg =$ **0.1468** average number of wrong base calls

(b) (2 points) The first step in any sequence analysis is to run FastQC tool. Using its output, report the length of each read, and calculate the average coverage of the genome by all reads (given a guess that the culprit food-poisoning bacteria is $Staphylococcus\ aureus$ with a genome size of 197,394 bp; note that average genome coverage is the average number of reads that span a nucleotide in the genome)?

**Solution:** The length of each read is **150 bp**. The number of sequences for the paired ends is 11167 + 11167 = 22334. Its given that the bacteria has a genome size of 197,394 bp. Hence, the genome coverage is given by:

$$Genome\ coverage = \frac{22334 * 150}{197394} \approx 17X\ coverage$$

(c) (3 points) Run velvet with a k-mer size of 29, and report how many contigs have been built, and what the mean, min and max length of the contigs are?

**Solution:** For K = 29; Number of contigs built = 246; Mean length= 740.24; Minimum length = 1; Maximum length = 9449

(d) (4 points) Rerun velvet with the following k-mer sizes: 23, 57, 100, and report for each case the above contig metrics in table format. Comment on the trade-off between small and large k-mer values, and which k value looks optimal to you.

**Solution:**

| K | Number of Contigs | Mean Length | Min Length | Max Length |
|---|---|---|---|---|
| 23 | 269 | 677.67 | 1 | 9449 |
| 57 | 123 | 1472.45 | 1 | 24019 |
| 100 | 223 | 733.53 | 2 | 4652 |

Larger values of k lead to a decrease in the number of contigs, which is desirable.

However, it also increases the chances of error in a k-mer.[2] Hence, on increasing k, assembly becomes easier due to fewer contigs, but error rates increase. Whereas, for smaller k, the genome assembly is more difficult, but the error rate is reduced. In this example, k = 57, is the optimal k value due to the few number of contigs and relatively lower read error rate compared to k = 100.

5. (6 points) [Research warmup]: Provide properly-formatted references for papers in this solution.

   (a) (3 points) Browse through the latest issue of top-ranked bioinformatics or systems biology journals (e.g., Bioinformatics, PLoS Computational Biology, Cell Systems, Molecular Systems Biology, etc.). Which one article caught your attention/interest the most based on only the titles and/or abstracts, and why?

**Solution: DeepPhos: prediction of protein phosphorylation sites with deep learning.**[3] This article caught my attention due to the following reasons:

- In my last semester, we had briefly touched upon cell signaling pathways through phosphorylation of proteins in my Structural Biology course. These signals can control essential tasks such as cell division and even cell death. Phosphorylation changes the structural conformation of a protein leading to change in function.
- It uses deep learning for the given task. I have explored the use of DL in more conventional computer vision tasks. However, I have not yet read about its applications in the field of genomics. I am curious to learn how this is framed as a deep learning problem.
- It presents a novel Deep Learning architecture specifically for the prediction of protein phosphorylation sites.
- The source code for the paper is publicly available.

   (b) (3 points) What is the latest research publication you could find on $de\ novo$ genome assembly (i.e., genome assembly as seen in class without the knowledge of a reference genome), and what key algorithm did it use?

**Solution: Yet another de novo genome assembler.**[4] *(posted online on May 31, 2019)*
The paper presents a novel tool called **Ra** (Rapid Assembler) for the de novo genome assembly problem for long uncorrelated reads. The underlying algorithm is based on sequence classification and assembly graphs. It is based upon the **Overlap-Layout-Consensus paradigm**.

# References


[1] S. Xie, "Notes on de bruijn sequences," *Discrete applied mathematics*, vol. 16, no. 2, pp. 157–177, 1987.

[2] R. Chikhi and P. Medvedev, "Informed and automated k-mer size selection for genome assembly," *Bioinformatics*, vol. 30, pp. 31–37, 06 2013.

[3] F. Luo, M. Wang, Y. Liu, X.-M. Zhao, and A. Li, "DeepPhos: prediction of protein phosphorylation sites with deep learning," *Bioinformatics*, vol. 35, pp. 2766–2773, 01 2019.

[4] R. Vaser and M. Šikić, "Yet another de novo genome assembler," *bioRxiv*, 2019.