University of TEHRAN

Algorithms in Bioinformatics

Project 1- Read mapping and genome assembly

Department of Bioinformatics, IBB, University of Tehran

Spring 2023

Teaching assistants: Sajedeh Bahonar and Mohammad Sadegh Vafaei
Instructors: Alireza Fotuhi Siahpirani and Hesam Montazeri
Due date: 15 Ordibehesht 1402

Objective

In this project, you will learn how to analyze short-read and long-read sequencing data, perform quality control, create genome assemblies, and conduct read mapping and variant calling.

Part A - Downloading E. Coli WGS data, preliminary analyses and quality controls

First, download the data according to the following steps and then proceed with the subsequent analyses.

1. Download SRR8185316 (short-read WGS of E.coli) and SRR10538956 (long-read WGS of E.coli) from SRA using the SRA Toolkit in Linux or Windows. (Hint: you may use fastq-dump [options] <accession>).
2. In each file, find the strain of E.coli and the type of reads (single-end or paired-end) (refer to NCBI or EBI ENA (European Nucleotide Archive)). In addition, explain the difference between paired-end and interleaved paired-end files.
3. Answer the following questions about the short reads fastq file (use existing R packages such as ShortRead to answer the following questions):

    I. How many reads are in the fastq file?

    II. Print the identifier, quality, and sequence of the first read of the fastq file.

    III. How many times does the TTAAATGGAA subsequence appear in the file?

    IV. Extract the first 1000 sequences of the fastq files (4000 lines).

V. Plot the quality of the reads in the fastq file using a box plot.

VI. Show the distribution of read lengths using a density plot.

4. Perform quality control of the reads using FastQC and interpret the results. Complete this task using both the command-line function as well as the FastQC graphical interface. (you may download FastQC from here.)

Part B - De novo genome assembly

Install SPAdes, Canu, Quast, BWA, Samtools, Pilon, or any other alternative software of your choice. Afterwards, follow the steps provided below.

1. Run SPAdes to generate draft genome assemblies from short reads.
2. Run Canu to generate draft genome assemblies from long reads.
3. Assess the quality of the draft genome assembly (long reads) using Quast and compare it to the reference genome (Download the reference genome from here)
4. Long-read technology may produce various types of error, leading to low accuracy in genome assemblies. To address this challenge, Pilon can be used, a tool that detects and corrects errors in genome assemblies, including single nucleotide polymorphisms (SNPs), insertions, and deletions (indels). The goal of this part is to enhance the draft genome assembly by using the short-read data. To this end, map the Illumina short-read data to the PacBio assembly using BWA, and then use Pilon to identify and correct assembly errors.

Part C - Read mapping and variant calling

1. Print the head of the obtained SAM file in part B, question 4. Explain what you see for the first hit (you can do this step either in Linux or R).
2. Convert the SAM file to an indexed BAM file. Hint: use samtools view, samtools sort, samtools index.
3. Use the Integrative Genomics Viewer (IGV) to visualize the mapped reads in a 200-b genomic region of your choice. Select the reference genome fasta and GTF file (GTF is optional).
4. Determine the percentage of short reads that are mapped to the assembled genome from the long reads as well as to the reference genome. Hint: use samtools flagstat.
5. Get the read depth for the sorted BAM file at all positions of the assembled genome and report the mean of all reads. Hint: use samtools depth.
6. Make yourself familiar with the CIGAR format. How do you interpret the following expressions?

   “29S21=1X25=”

"20M2I1M1D10M"

"5M10N25M"

7. Perform variant calling on the obtained BAM file using the reference genome or refined assembled genome from the long reads and save the output as a VCF file. Hint: you may need the following commands/options: samtools mpileup, bcftools, multiallelic-caller algorithm

Important note: please ensure that all results are organized within a directory named Project1. This directory should contain subdirectories for the input data, QC results, the output of mapping to the reference genome, variant calling results, de novo assembly outputs, and the alignment results of the assembled genome with the reference genome.