Introduction to Genome Assembly with SPAdes

Overview

In this tutorial, we will guide you through the process of assembling genomic reads using the SPAdes assembler. Genome assembly is a critical step in bioinformatics, as it involves reconstructing the original genome from short DNA sequences generated by sequencing technologies. A well-assembled genome is essential for various applications, including comparative genomics, functional genomics, and evolutionary studies. For this practical we will be using the cleaned reads that we generated from our read_cleaning tutorial so if you haven't don't that tutorial i would advise that you do in order to complete this section.

Importance of Genome Assembly

Accurate genome assembly is vital for understanding the structure, function, and evolutionary relationships of genomes. High-quality genome assemblies allow researchers to:

- Identify genes and regulatory elements.
- Understand genetic variation within populations.
- Perform annotation and functional studies on the genome.
- Facilitate downstream analyses, such as variant calling and phylogenetic studies.

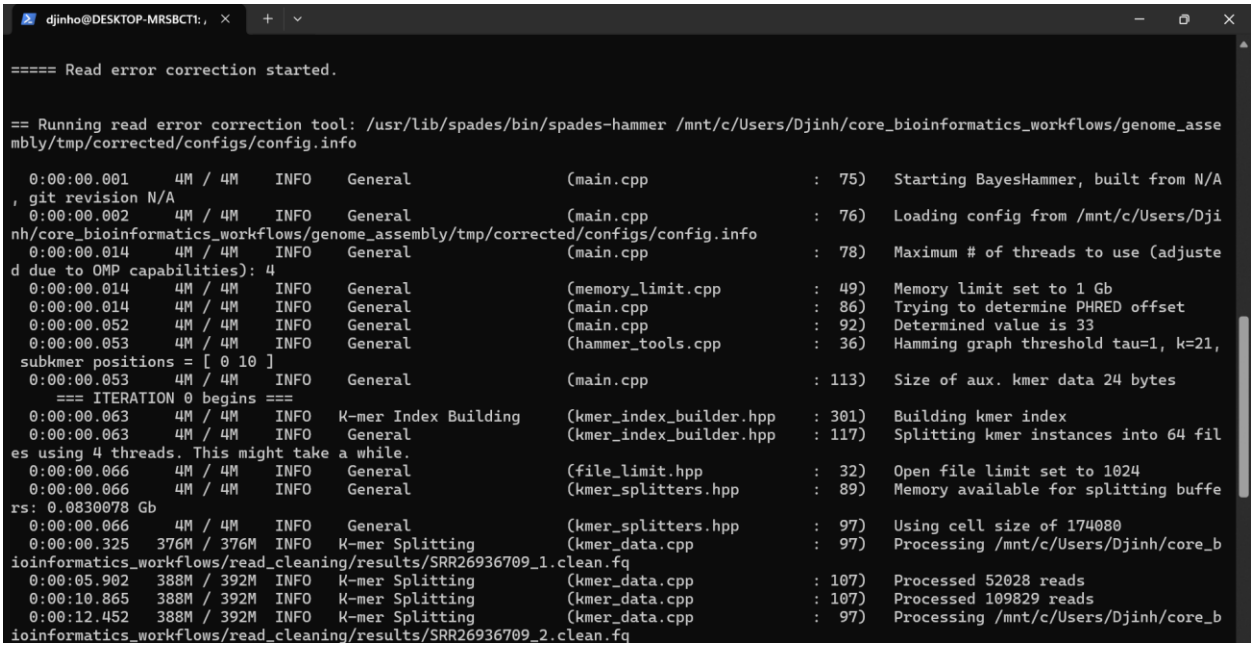
This tutorial emphasizes the basic operation of SPAdes and serves as a stepping stone toward more complex genome assembly tasks. For a comprehensive understanding of SPAdes features and capabilities, we encourage users to consult the official documentation and relevant scientific publications.

Key Steps Covered in the Tutorial

1. **Setting Up Directories**: We will create a structured directory to organize input data, temporary files, and results:
   mkdir genome_assembly
   mkdir input tmp results
   cd genome_assembly

   2) **Assembling Reads**: Use the SPAdes command to assemble cleaned reads.

Replace the paths with the correct ones from your read-cleaning tutorial. The image below is the output you should get.



3)
As with most assemblers, SPAdes generates multiple output files. One key file you'll likely use in further analyses is the `scaffolds.fasta` file. To prepare for the next steps, copy this file to the `results` directory: cp tmp/scaffolds.fasta results/

4) **Examining the Assembly Output**: Check the contents of the `scaffolds.fasta` file to understand its structure. You can view the first or last ten lines using:

```
head results/scaffolds.fasta
```

Or, to check the last 10 lines, use:

```
tail results/scaffolds.fasta
```

```
djinho@DESKTOP-MRSBCT1:/mnt/c/Users/Djinh/core_bioinformatics_workflows/genome_assembly$ head results/scaffolds.fasta
>NODE_1_length_449_cov_2.930108
CAGCAGCCGCGGTAATTCCAGCTCCAATAGCGTATATTAAAGTTGTTGCAGTTAAAAAGC
TCGTAGTCGAACTTCGGATCTGGCGGGATGGTCCGCCTTACGGTGTGTACTGTCCGGCCG
GATCTTACCTCTTGGTGAAGCCGTATGCCCTTTACTGGGTGTGCAGTGGAACCAGGAATT
TTACCTTGAGAAAATTAGAGTGTTCAAAGCAGGCAATTGCCCGAATACATTAGCATGGAA
TAATAGAATAGGACGTGCGGTTCTATTTTGTTGGTTTCTAGGATCGCCGTAATGATTAAT
AGGGACGGTCGGGGGCATTAGTATTCCGTTGCTAGAGGTGAAATTCTTAGATTTACGGAA
GACTAACATCTGCGAAAGCATTTGCCAAGGACGTTTTCATTGACCAAGGACGAAGGTTAG
GGGATCAAAAACGATTAGATACCCGCGTA
>NODE_2_length_448_cov_4.749326
djinho@DESKTOP-MRSBCT1:/mnt/c/Users/Djinh/core_bioinformatics_workflows/genome_assembly$
```

5) Assembly software typically provides statistics on the process, but the format of these outputs can vary across different assemblers. To generate a standardized report, you can use **Quast** (Quality Assessment Tool for Genome Assemblies). Run Quast on the `scaffolds.fasta` file without any additional options to obtain basic assembly statistics:

So install Quast first:

```
sudo apt install python3-pip
sudo pip3 install quast
```

6) The run Quast the default Quast command goes up to 500bp but these reads were short so we decided to use 400 bp: /usr/local/bin/quast.py scaffolds.fasta --min-contig 400

Conclusion

By following this tutorial, you will gain practical experience in genome assembly using SPAdes, equipping you with essential skills for bioinformatics research. For a deeper understanding, consider exploring the official documentation and recent studies that provide insights into assembly strategies and quality assessments.

The output in the terminal :