Project S. 3-4: NGS – Project plan

Ahmed Sayadi

asayadi@univ-catholyon.fr

FISE2 – 2025/2026

Paper summary

RNA-seq and Tn-seq reveal fitness determinants of vancomycin-resistant *Enterococcus faecium* during growth in human serum

Xinglin Zhang, Vincent de Maat , Ana M. Guzmán Prieto , Tomasz K. Prajsnar , Jumamurat R. Bayjanov, Mark de Been , Malbert R. C. Rogers , Marc J. M. Bonten , Stéphane Mesnage , Rob J. L. Willems and Willem van Schaik

Enterococcus faecium is a commensal bacterium in the human gut that is associated with opportunistic bloodstream infections in immunocompromised hospitalized patients. Moreover, it has recently acquired resistance to multiple antibiotics, which represents a big public health concern. However, the growth and survival mechanisms of this opportunistic pathogen in the bloodstream have not been characterized. In this study you will identify what genes allow *E. faecium* to grow in human blood by different profiling techniques based on RNA-Seq and Tn-Seq.

Paper summary

Illumina PacBio Nanopore Genome Assembly RNA-Seq Tn-Seq Differential Expression

The main analyses included in this study are:

- Genome assembly on *E. faecium*
- Differential gene expression of *E. faecium* on human serum against rich medium using RNA-Seq data.
- Identification of genes that contribute to survival and growth in human serum using Tn-Seq data.

What you need to do

Analyses:

1. Genome assembly with PacBio reads.
2. Assembly evaluation.
3. Structural and functional annotation.
4. Reads preprocessing: trimming + quality check (before and after)
5. RNA-Seq reads alignment against assembled genome.
6. Differential expression analysis between rich medium and heat-inactivated serum conditions.

What you need to do

Extra analyses:

Genome assembly with Illumina and Nanopore reads.

- Assembly evaluation (extra methods).
- Plasmid identification.
- SNPs calling.
- Evaluate antibiotic resistance potential.
- Identify essential genes for growth in human serum based on the Tn-Seq data analysis.

Evaluation

La note est basée sur 2 parties :

- **GitHub Wiki** :

  (1) Vous résumerez l'article (problématique, méthodes utiliser et résultats).

  (2) Vous allez refaire une partie de l'analyse faite par les auteurs.

  Votre wiki doit comporter aussi:

  - Le plan du projet.
  - Les méthodes que vous avez utilisées pour votre projet.
  - Les interprétations et conclusions biologiques de vos résultats.
  - Vous devez enregistrer toutes les commandes, scripts et fichiers que vous interprétez ou produisez.

- **Réponse aux questions.**

Evaluation

- Pour chaque logiciel, il est nécessaire de consulter le manuel afin de prendre connaissance des instructions et d'écrire les lignes de commande appropriées pour l'exécuter. Vous expliquerez aussi les différents paramètres utilisés pour chaque ligne de commande.
- Il faut également préciser les types de fichiers d'entrée et de sortie de chaque logiciel.
- En raison du volume important des données de séquençage, j'ai dû refaire certaines analyses. Dans le dossier de résultats de chaque logiciel, vous trouverez les résultats finaux correspondants."
- Pour ces analyses, vous devez interpréter les résultats.

Github wiki

1. **Introduction**
   - Briefly introduce the scope and purpose of the paper.
2. **Paper Summary**
   - **Problem Statement**: Define the main research question or issue addressed by the paper.
   - **Methods Used**: Summarize the methodologies employed.
   - **Key Findings**: Highlight the main results and conclusions.
3. **Analysis Workflow**
   - **Workflow Overview**: Present a detailed figure of the analysis workflow, including tools used, and specify input and output file types at each step.

Github wiki

4. **Methods:**
   - **Type of Analysis Required**: Describe the type of analysis performed (e.g., genome assembly, RNA-Seq).
   - **Detailed Tool Explanation**:
     - For each analysis step, identify the tool used and explain its underlying principle.
     - Provide command-line examples with a breakdown of each parameter used.
5. **Results:**

   Interpret the output of each tool and discuss how these results contribute to answering the research question.
6. **Conclusion:** Summarize the overall findings and their implications based on the analysis conducted.

Introduction - Questions to answer

1) **Define and compare the roles of the chromosome and plasmids in *Enterococcus faecium***. Why are both included in the definition of the bacterium's genome?
2) **What types of genes are likely found on the chromosome versus the plasmids in *Enterococcus faecium*?** Give examples and explain how they benefit the bacterium.
3) **Explain why it is important to include both chromosomal and plasmid DNA when sequencing** and assembling the complete genome of *Enterococcus faecium*.
4) **Discuss the importance of plasmids in bacterial evolution**. How can studying plasmids in *Enterococcus faecium* inform us about the potential spread of antibiotic resistance genes to other bacterial species?

Genome Assembly - Tools to use

- **canu** is a *de novo* whole-genome shotgun (WGS) assembler especially designed for long-read sequencing technologies. (https://canu.readthedocs.io/en/latest/tutorial.html)
- **SOAPdenovo** is a short-read assembly method that can produce large genomes (human-size) (https://github.com/alekseyzimin/SOAPdenovo2/blob/master/MANUAL).
- **Spades** is an assembly toolkit capable of providing hybrid assemblies (combining short and long reads, i.e. Illumina + PacBio). It can work with paired-end reads, mate-pairs and unpaired reads (https://home.cc.umanitoba.ca/~psgendb/doc/spades/manual.html).

Genome Assembly - Analysis to do

- Use **canu** to assemble the **PacBio** reads.
  - Report the command line used
  - Explain each parameter used.
  - Interpret the output
- Use **Spades** to assemble **illumina** and **Nanopore** reads.
  - Report the command line used
  - Explain each parameter used.
  - Interpret the output

Canu can fail because of insufficient memory. Don’t worry, you can use the assembly file " E754.canu.contigs.fasta "

Genome Assembly - Questions to answer

- What's the genome size ?
- Report from the paper the accession numbers of the submitted sequences.
- How many contigs do you expect? How many do you obtain?
- What is the difference between a 'contig' and a 'unitig'?
- What is the difference between a 'contig' and a 'scaffold'?
- What are the k-mers? What k-mer(s) should you use? What are the problems and benefits of choosing a small kmer? And a big k-mer?
- Some assemblers can include a read-correction step before doing the assembly. What is this step doing?

Assembly evaluation – Tools to use

QUAST evaluates how good a newly generated genome assembly is (https://quast.sourceforge.net/docs/manual.html).

MUMmerplot uses the output from mummer, nucmer, promer (whole-genome alignment) , and generates an alignment dot-plot comparing two aligned assemblies of the similarities between them (https://mummer4.github.io/manual/manual.html).

BUSCO used to assess the genome completeness by evaluating the presence of conserved single-copy orthologs. It generates a report showing the proportion of complete, single-copy, duplicated, and missing genes, providing insights into the quality and completeness of the genome assembly (https://busco.ezlab.org/busco_userguide.html).

Assembly evaluation – Analysis to do

- Run **quast** to evaluate both assembled genomes.
  - Report the command line used
  - Explain each parameter used.
  - Interpret the output

Assembly evaluation - Questions to answer

- What do measures like N50, N90, etc. mean?
- How can they help you evaluate the quality of your assembly?
- Which measure is the best to summarize the quality of the assembly (N50, number of ORFs, completeness, total size, longest contig ...)
- How does your assembly compare with the reference assembly? What can have caused the differences?
- Why do you think your assembly is better/worse than the public one?

Improve Assembly - Tools to use

- **BWA** BWA is a software package for mapping low-divergent sequences against a large reference genome (https://bio-bwa.sourceforge.net/bwa.shtml).

- **Pilon** is a software tool that can improve draft assemblies and find variation among strains (https://github.com/broadinstitute/pilon/wiki).

Improve Assembly – Analysis to do

- Run **BWA** to mapp illumina reads back to the assembled genome with **CANU**.
  - Report the command line used
  - Explain each parameter used.
  - Interpret the output
- Run **Pilon** to improve the assembly done with **CANU**.
  - Report the command line used
  - Explain each parameter used.
  - Interpret the output
- Check if the assembly did improve using **Quast**
  - Report the command line used
  - Explain each parameter used.
  - Interpret the output

Improve Assembly – Analysis to do

- Run **BWA** to mapp illumina reads back to the assembled genome with **CANU**.
  - Illumina reads needs to be merged in one file. Use the shell command '**cat**' to do it

    `cat forward.A_1.fastq forward.B_1.fastq > forward_1.fastq` (you do the same for the reverse read)
  - Now you need to index your genome

    `bwa index genome_assembly.fasta`
  - Mapp the reads back to the assembled genome

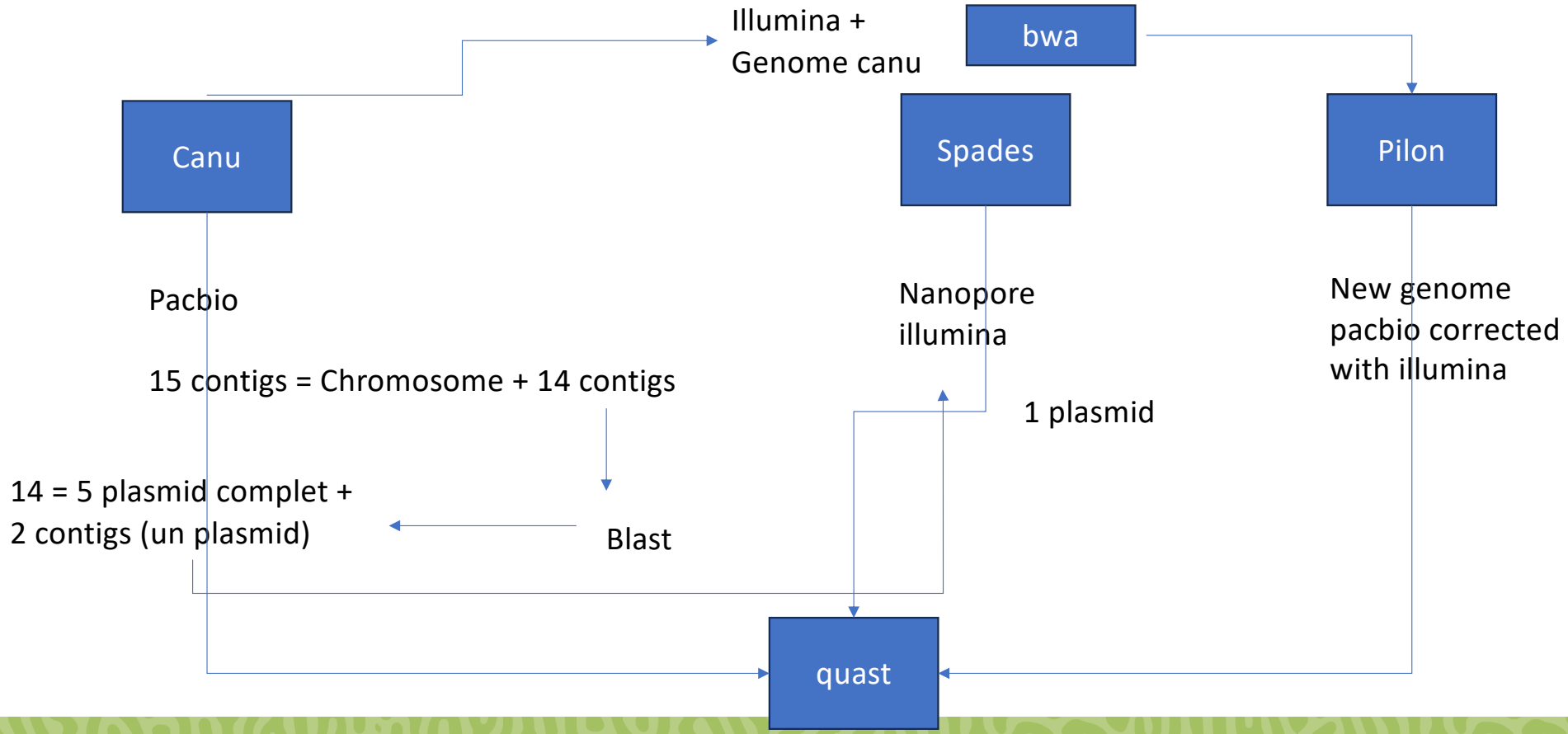
    `bwa mem genome_assembly.fasta forward_reads.fastq reverse_reads.fastq > aligned_reads.sam`
  - Convert SAM to BAM

    `samtools view -Sb aligned_reads.sam > aligned_reads.bam`
  - Sort and Index BAM file

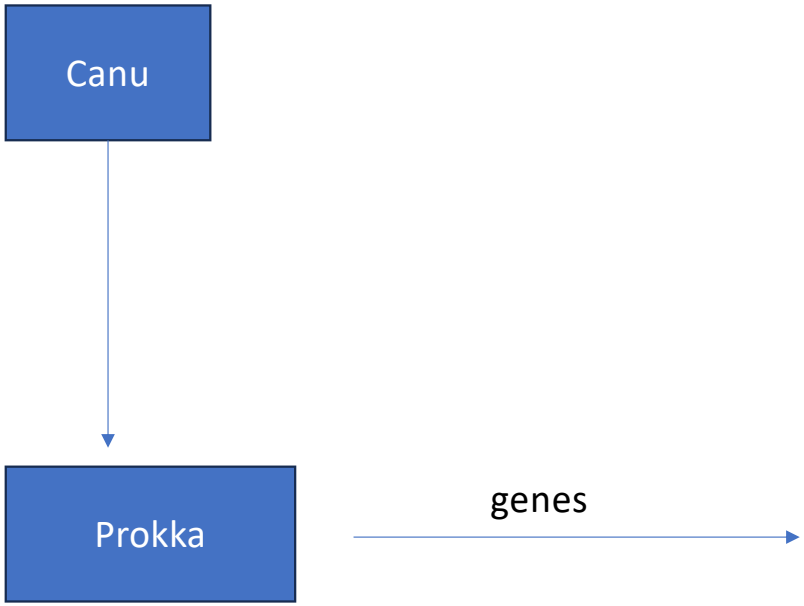
    `samtools sort aligned_reads.bam -o aligned_reads_sorted.bam`

    `samtools index aligned_reads_sorted.bam`

Improve Assembly – Analysis to do



Improve Assembly – Analysis to do



Improve Assembly – Analysis to do

- Run **Pilon** to improve the assembly done with **CANU**.
  - Report the command line used

    ```
    java -Xmx16G -jar pilon.jar --genome assembly.fasta --bam aligned_reads_sorted.bam --output polished_assembly --changes --threads 4
    ```

  - Explain each parameter used.
  - Interpret the output
- Check if the assembly did improve using **Quast**
  - Report the command line used
  - Explain each parameter used.
  - Interpret the output

Improve Assembly – Questions to answer

What type of reads were used for correction (short, paired-end, long reads)?

What parameters were used in **BWA** and **Pilon**, and how did they impact the results?

How computationally intensive was the correction process?

Compare both genomes before and after correction

Use **BLAST** to verify whether all the plasmids have been properly assembled.

Genome annotation – Tools to use

Prokka is a software pipeline that combines different tools for the annotation of prokaryotic genomes. It combines both structural and functional annotation by predicting and identifying genetic elements encoded in the genome (https://github.com/tseemann/prokka).

Maker2 is another structural annotation pipeline, designed for both eukaryotic and prokaryotic genomes (https://reslp.github.io/blog/My-MAKER-Pipeline/).

EggNOGmapper is a tool for fast functional annotation of already predicted sequences (works both for genes and proteins) using precomputed eggNOG-based orthology entries. It has an online web server (https://github.com/eggnogdb/eggnog-mapper).

Genome annotation – Analysis to do

- Use **Prokka** to annotate the assembly.
  - Report the command line used

```
prokka --outdir output_directory --prefix genome_prefix input_genome.fasta
```

  - Explain each parameter used.
  - Interpret the output

Prokka can be complicated to install – You can skip the installation and use the output files I prepared for you

Genome annotation – Questions to answer

- What information can you find in the **Prokka** summary file?
- How many genes were predicted?
- What's the number of rRNA, tRNA, coding sequences and pseudogenes predicted?
- What do you infer from the presence of pseudogenes in the summary file, if there is any?
- Are you satisfied with the annotation?