The assigned scientific paper is paper 1, RNA-seq and Tn-seq reveal fitness determinants of vancomycin-resistant *Enterococcus faecium* during growth in

human serum. *Enterococcus faecium* is a commensal bacterium in the human gastrointestinal tract, and the common cause of bloodstream infections in hospitalized patients. Moreover, it has recently acquired resistance to multiple antibiotics, which makes it more challenging for developing treatment of *E. faecium* bloodstream infections. Yet, the growth and survival mechanisms of this pathogen in blood have not been characterized. So, the aim of this project is to re-analyze the data in a similar way as the authors did on the original study, and re-evaluate their biological conclusions that genes involved in carbohydrate metabolism and nucleotide biosynthesis of E. faecium are essential for growth in human serum.

The vancomycin-resistant *E. faecium* strain E745 was used throughout the study. It was sequenced using a combination of Illumina sequencing, and long-read sequencing using PacBio and Nanopore. To re-analyze the sequences used in the study, the following analyses will be performed (in the same order as listed below):

1. Reads quality control

FastQC will be used to check the quality of Illumina short reads to make sure the quality of the data is good enough for any further analyses. Otherwise, data with bad quality can lead to errors or artifacts.

1. Reads preprocessing

Trimmomatic will be used to remove the low-quality base-calls and present adapters from the Illumina short-reads.

1. Genome assembly

Canu will be used to perform genome assembly for PacBio long-reads. Spades will be used for hybrid assemblies of Illumina and Nanopore reads.

1. Assembly evaluation

To evaluate the quality of the generated genome assemblies, QUAST, MUMmerplot and BCFtools will be used.

1. Aligning reads

BWA will be used as the aligner for this project. It can used to correct the assembly generated by Canu.

1. Structural and functional annotation

To predict what genetic elements the sequences encode, annotation will be performed by using Prokka.

1. Blast analysis

Blast against the NCBI database, the obtained results can be used for plasmid identification. The alignment file can also be used by Artemis Comparison Tool (ACT) for synteny analysis

1. Post-mapping analyses

* Use BCFtools to find SNPs
* Evaluate antibiotic resistance potential by ResFinder

1. RNA-Seq Analysis

BWA will be used to generate the required RNA-Seq reads alignment first.

Then, after sorting the alignments by SAMtools, use the alignment files to run HTSeq for generating the reads map, which will be used by DESeq2 for differential expression analysis.

1. Tn-Seq Analysis

* Use Bowtie to map the transposon sequences to the genome
* Sort and count the alignment by IGV
* Calculate read-count RPKM
* Use Cyber-T to compare the RPKM-values between different experimental conditions

Among all the analyses, genome assembly and differential expression analysis will require longer time to run, so they are the possible time bottlenecks for this project.

In terms of the time frame for this project, the plan is as follows:

|  |  |
| --- | --- |
| Week | Activity |
| 15 | Reads quality control and preprocessing |
| 16 | Genome assembly |
| 17 | Assembly Evaluation, aligning reads, annotation |
| 18 | Blast analysis, post-mapping analyses, RNA-Seq Analysis |
| 19 | Tn-Seq Analysis |

Therefore, by the end of week 19, I need to have finished running all the softwares so I can start analyzing the data.

The types of data that will be handling are DNA sequences, RNA sequences, short-reads from Illumina, long-reads from PacBio and Nanopore, alignment files, and different output files generated by sotfwares used in this project. The allocated storage space of the given account on UPPMAX is 32GB. In case of storage shortage, unnecessary files will be removed, and large files will be compressed.

As for data organization, I will:

* Use a spreadsheet to keep clear metadata
* Separate data and code, and never put big data files into my repository
* Keep data files and directories with unique and informative names
* Use symbolic links to create shortcuts when necessary
* Compress big data files
* Remove unnecessary files to avoid confusion