Formations de Bacheliers et de Masters organisées par la Fédération Wallonie-Bruxelles

Bachelor thesis

enhancing the capacity of biogas production by understanding the microbiota communities involved in the process

Wilfart Lydiane

Bachelor in biotechnic - third year

HEH - Dep. of sciences and technologies

HEPH - Condorcet

Academic year 2021 - 2022

Bachelor thesis

enhancing the capacity of biogas production by understanding the microbiota communities involved in the process

Wilfart Lydiane

Bachelor in biotechnic - third year

HEH - Dep. of sciences and technologies

HEPH - Condorcet

Academic year 2021 - 2022

Acknowledgment

I would like to express my deepest gratitude to those who supported me in realizing my bachelor's thesis. First, I would like to thank my professor, Mr. Coornaert, for his support, precious advice, and encouragement throughout this project.

I would also like to thank my supervisor, Mr. Bongcam, for the opportunity he gave me to do my internship at SLU last year, leading me to this subject.

I want to thank Renaud for his help with the thesis. His expertise in bioinformatics and metagenomics was of excellent service to me, and I am highly grateful for his time and dedication.

I would also like to thank my family for their unwavering support, especially Jennifer and my Aunt Ada for their emotional support. I am grateful to always count on them to support me in difficult times.

Finally, I would like to express my gratitude to my partner for her constant love and support throughout this year. She has been a source of motivation and inspiration for me.

I am aware that this work would not have been possible without their help, support, and encouragement. Their support was crucial in helping me overcome the challenges I encountered throughout my graduate work. Once again, I thank you all for your unwavering support and contribution to the completion of this project.

Abstract

Biogas is a promising renewable energy source that can be produced from organic waste. However, to maximize its production and yield, it is essential to understand the biological mechanisms involved in its formation. Metagenomics is a method that allows the global analysis of microbial communities involved in biological processes such as biogas production.

This study analyzed six metagenomic samples from two different biogas reactors to determine the microbial species present and the genes involved in biogas formation. Bioinformatics tools were used to extract information from the sequencing data.

The results of this study allowed us to characterize the microbial species involved in biogas production and to discover genes potentially involved in this process. These results could improve biogas production by optimizing the environmental conditions of the reactor and selecting the most efficient microbial species for biogas production.

Le biogaz est une source d'énergie renouvelable prometteuse qui peut être produite à partir de déchets organiques. Cependant, pour maximiser sa production et son rendement, il est important de comprendre les mécanismes biologiques impliqués dans sa formation. La métagénomique est une méthode qui permet l'analyse globale des communautés microbiennes impliquées dans des processus biologiques tels que la production de biogaz.

Dans cette étude, six échantillons métagénomiques provenant de deux réacteurs à biogaz différents ont été analysés afin de déterminer les espèces microbiennes présentes et les gènes impliqués dans la formation du biogaz. Des outils bio-informatiques ont été utilisés pour extraire des informations des données de séquençage.

Les résultats de cette étude nous ont permis de caractériser les espèces microbiennes impliquées dans la production de biogaz et de découvrir des gènes potentiellement impliqués dans ce processus. Ces résultats pourraient être utilisés pour améliorer la production de biogaz en optimisant les conditions environnementales du réacteur et en sélectionnant les espèces microbiennes les plus efficaces pour la production de biogaz.

Table of contents

Table des matières

List of tables

List of Figures

Introduction

One of the significant problems of today is the treatment of waste, especially organic waste.

Their fate (Ngo et al., 2021) is diverse:

1. Landfilling

A certain quantity of organic waste is deposited in landfills. However, this method raises many problems, such as limited space disponibility, greenhouse gas production, and groundwater contamination. Nonetheless, with the help of some installation, landfill gas can be used to produce electricity or heat.

2. Incineration

This method is defined by the combustion of waste between 750 and 1100°C. It reduces the waste mass by up to 75%. A small portion can be included into electricity and heat generation. The main issue with this method is the emission of greenhouse gases.

3. Composting

it is defined as aerobic degradation of organic matter producing CO2. It is not included in any energy production, but composting has great agricultural value. One of the cons is that it demands to be aerated a lot to prevent the emission of greenhouse gases. Another problem is due to hygienic concerns in a dense population.

4. Anaerobic Digestion

This last one is the one we will talk about throughout this bachelor thesis.

Aerobic digestion is the biodegradation of organic matter via microorganisms in anaerobic conditions. This method allows the production of electricity, heat, and other valuable products. Anaerobic Digestion is the pathway to produce biogas.

I have six samples taken from a biogas reactor, which will be analysed to identify the species inside and the interesting genes for biogas production.

To understand the subject, I must explain multiple concepts first.

I will start with biogas concepts, how it is produced, and what it is used for. Then, I will describe metagenomics and how it works.

Biogas

How do we produce from organic waste?

The process to produce biogas is called anaerobic digestion. This process is using agricultural, industrial, and domestic waste.

Biogas is composed by:

- 50-70% of methane
- 30-70% of CO_2
- Small quantities of other gases

We can use biogas directly to produce heat and electricity or be enriched in bio-methane and used as vehicular fuel.

There are many ways to treat waste like:

- Gasification
- Pyrolisis
- Incineration

But those technologies sometimes demand more energy than they can produce. The use of anaerobic digestion is interesting.

The leading actors in this process are microorganisms. They are the ones who transform the substrate into biogas.

There are four steps to the biogas process:

1. Hydrolysis
   This first step involves decomposing large polymers into smaller molecules like simple sugar, amino acids, and fat acids using hydrolytic bacteria (e.g., Bacillus). Hydrolytic bacteria are microorganisms that can produce hydrolytic enzymes like proteinase, cellulase, lipase, and so on.
2. Acidogenesis
   Monomers are converted into volatile fatty acids(VFA), organic acids, and acid alcohols by fermentative bacteria (e.g. Acetivibrio)

3. Acetogenesis
   During that step, acetogens produce acetic acid, carbon and hydrogen.
4. Methanogenesis
   This is the last step of the process. It allows methane production from hydrogen, CO_2, and acetate by methanogens (archaea).
   The below equations are about the final step(Muzenda, 2014)

$$CO_2 + 4H_2 \longrightarrow CH_4 + 2H_2O$$

$$CH_3COOH \longrightarrow CH_4 + C_2O$$

Figure 1 – Equation of methanogenesis

I will end with a general equation of all the process

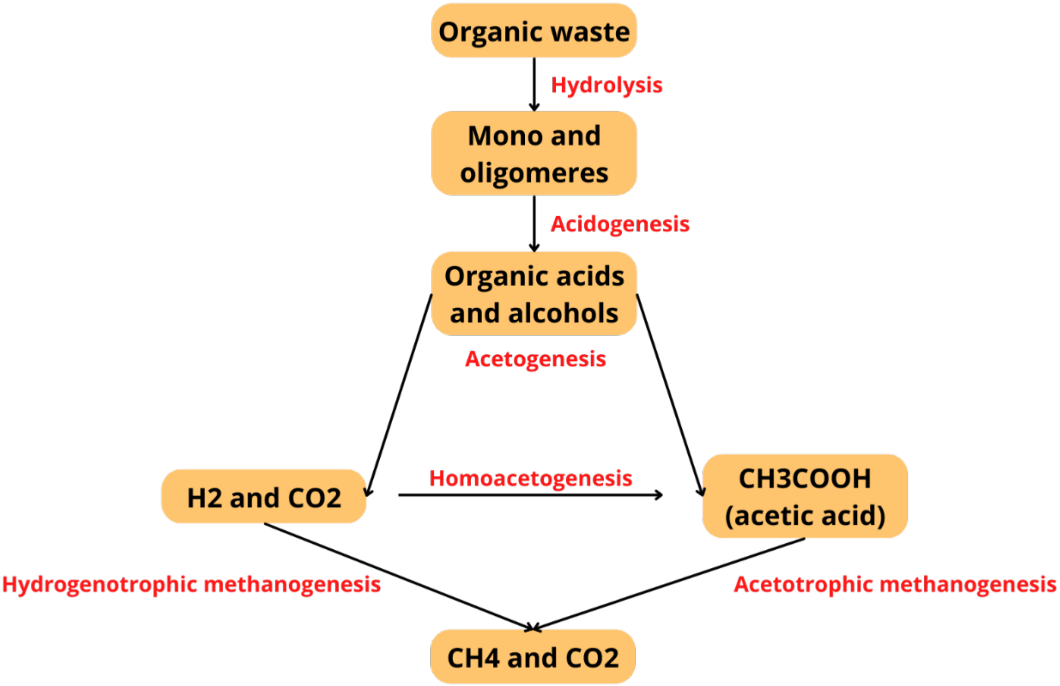


Figure 2 - General steps of biogas process

Biogas reactor

A Biogas reactor (Vögeli et al., s. d.), also called digester, is a closed tank with no oxygen. In it are put organic matter coming from all sorts of waste. It is kept at degrees and stirred continuously. In this anaerobic environment, bacteria will naturally grow.

In biogas plants, we often see four reactors, three main and one who finish the job.

here is a fixe dome type of biogas reactor

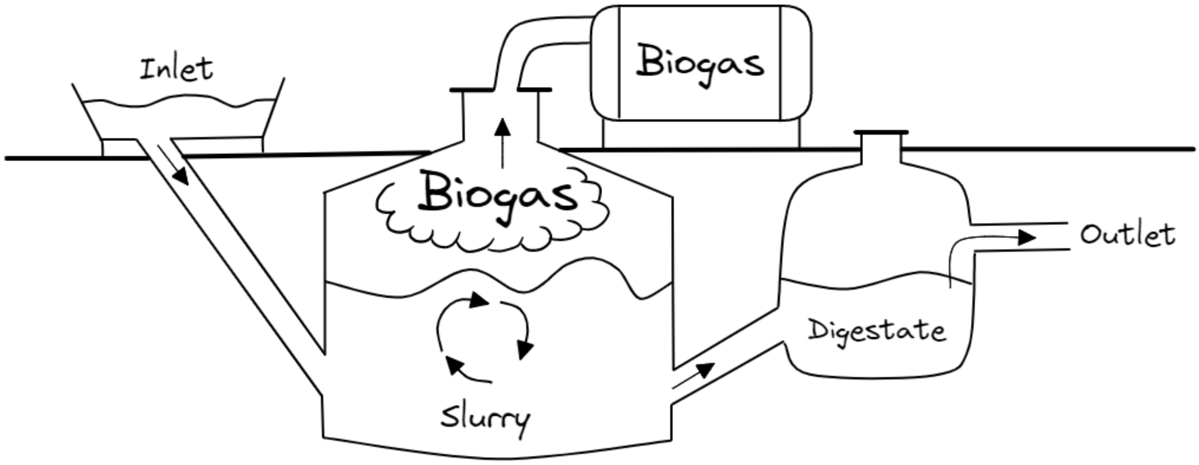


Figure 3 - Biogas reactor

Organic matter comes from Three main places :

- Municipality (houses, companies)
- Industrial
- Agricultural

The output are:

- electricity (-> electricity provider)
- Heat (heating the biogas reactor and local houses)
- Fertilizer (at the end of the process, there is left digestat, which can be used as fertilizer)
- Biofuel from methan (cars, buses)

Biogas Plant Input/Output Diagram:

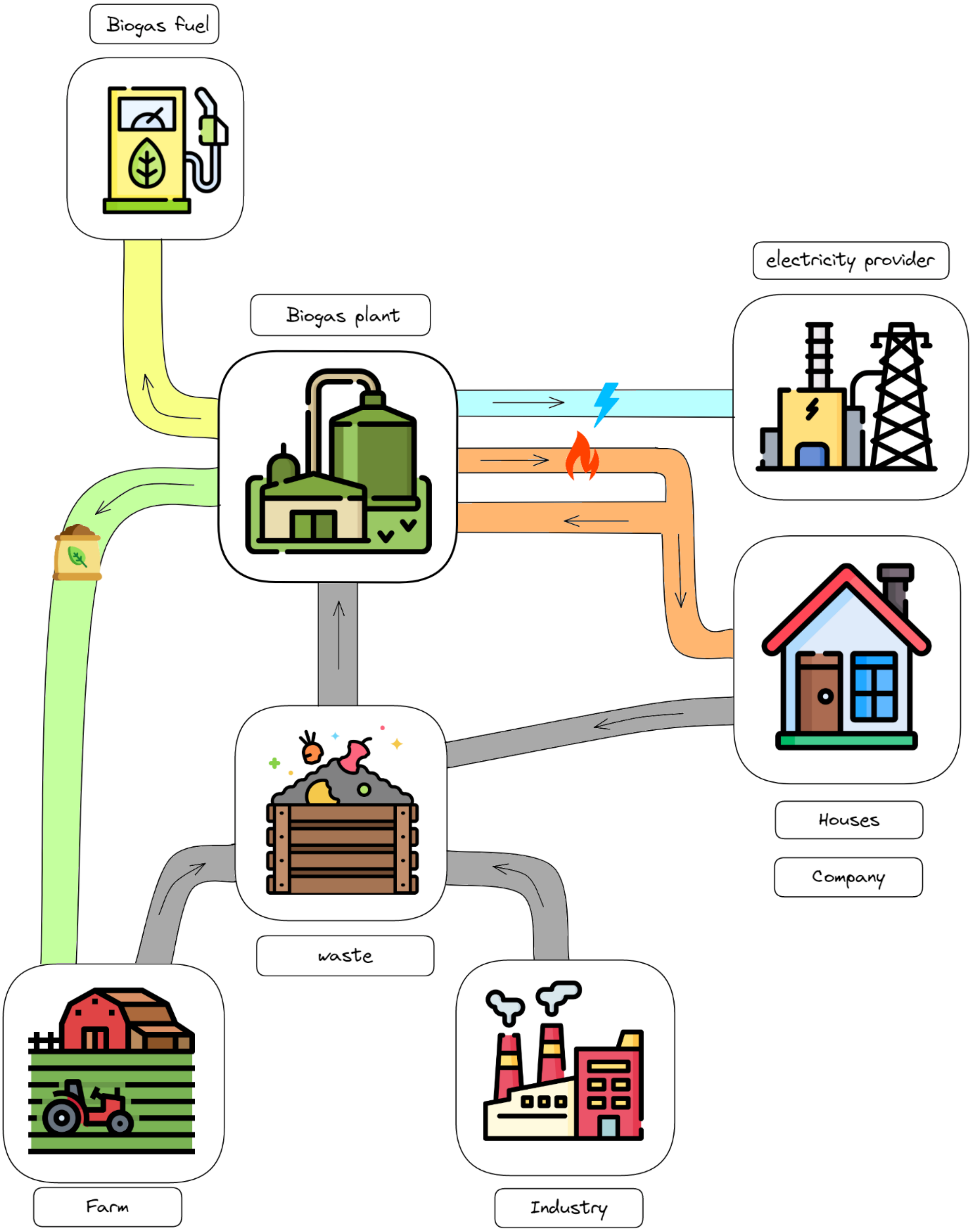


Figure 4 - Biogas Plant Input/Output Diagram

There are many different biogas technologies following different pathways. This is a diagram of these different pathways :

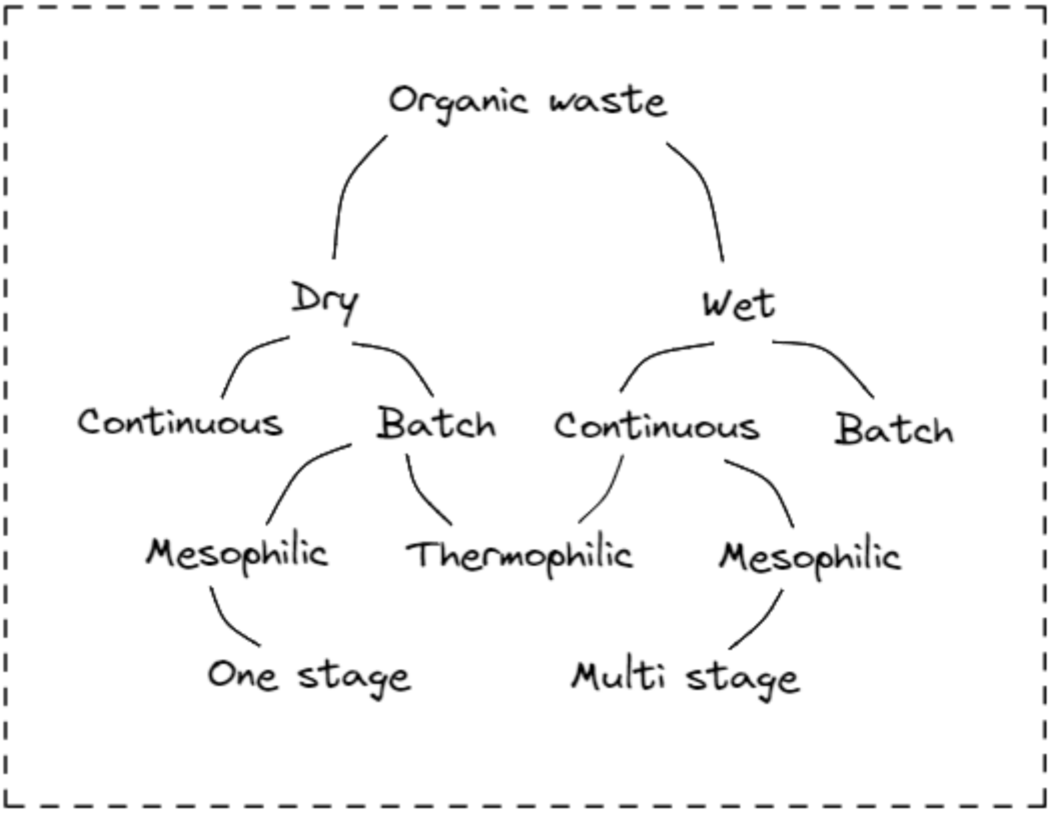


Figure 5 - Biogas technology pathway

Continuous means the feeding is done at regular intervals with equivalent volume. batch is the opposite of discontinuous feeding.

Wet means the total solid content will be 20% or less. The rest is water.

Mesophilic means the reactor operates at more or less 35 degrees Celsius. Thermophilic means a temperature around 55 degrees Celsius.

One stage means that all the process to produce methan is done in one tank. The process can be done in one, two, three, or four tanks. Therefore it's a multi-stage configuration.

Those technologies lead to different types of digesters like:

- Fixed dome (wet - continuous - mesophilic)
- Floating-drum
- Balloon-type
- Garage-type

Biofuel (biomethane upgrade)

The supply of petroleum fuels will gradually decrease. I need something to replace it.

Compared with other biomass-based vehicle fuels available, biogas often has several advantages from an environmental and resource-efficiency perspective and greenhouse gas reduction.
Bioethanol and biodiesel production routes have limitations in this regard.

Biogas has to be upgraded to natural gas quality to be used in standard vehicles designed to use natural gas.

The most common technologies for biogas upgrading are :(*biogas upgrading*, s. d.)

- Water scrubber technology
- PSA(Pressure Swing Adsorption) technology

Gas upgrading is usually performed in two steps. The primary step is the process that removes the CO2 from the gas and also minor contaminants.

Different types of sequencing

Illumina sequencing

Illumina sequencing involves fragmenting DNA, attaching short DNA fragments (adapters), and amplifying these fragments on a solid surface. Fluorescently labeled nucleotides are added during sequencing, and as each nucleotide is incorporated, its fluorescence is detected. This process generates millions of short DNA reads.

Nanopore sequencing

Nanopore sequencing is a method that directly reads DNA as it passes through a nanopore protein. This technology measures changes in electrical current caused by the passage of nucleotides through the pore, allowing real-time sequencing without the need for amplification. Nanopore sequencing offers portability.

RNA sequencing

RNA sequencing (RNA-seq) is a molecular biology technique used to study gene expression and analyze RNA molecules in a biological sample. It involves extracting RNA and converting it to complementary DNA (cDNA). RNA-seq provides insights into gene activities.

Metagenomics

What is it about?

Metagenomics is the study of microorganism communities in a given habitat (Hugenholtz & Tyson, 2008). It implies directly collecting microorganisms in a specific environment, sequencing DNA, and analyzing the data.

Metagenomics delivers information on microbial communities and their functioning within their environment.

Metagenomics uses next-generation sequencing (NGS). It empowers the sequencing of microorganisms that cannot be cultivated in a controlled environment.

There are two approaches to carrying out metagenomics:

Amplicon sequencing (Zhu et al., 2018) *(targeted)*

In this case, we magnify a portion of genes coding for 16S using PCR. We acquired an amplicon. 16S holds nine hypervariable regions unique to species. The amplicon from different samples receives a molecular barcode, and everything is sequenced at once. After sequencing, we juxtapose it to the 16S database. As a result, we get a taxonomic profile.
The 16S amplicon sequencing is restricted to the identification of bacteria and archaea.

- *Shotgun metagenomics (untargeted) (Quince et al., 2017)*
  We fragment the DNA of a sample, and we sequence them. We obtain reads, and then assemble them in a metagenomic pipeline. Like for 16S, you trim and compare to a database. In this case, we get more information. We manipulate data for assembly, binning, metabolic, and antibiotic gene profiling.
  Shotgun metagenomics allows the identification of all three domains of life (Eukarya, Archaea, and Bacteria) in the same process.

One of the metagenomics challenges today is orchestrating different tools and workflows to analyze the data.

Moreover, software installation is sometimes excessively complicated (missing libraries, dependency conflicts, OS, etc....). Sometimes, they also require extensive computing resources.

Typical Metagenomics Workflow (Shotgun)

The first step is to collect samples. One way to do that is to take one sample daily for a week. In that case, we will obtain more or less the same species daily. Another way would be to do a punctual assessment of the diversity, like deep ocean research.

In the following example, I will assume we are using the first way.

We sequence them and then do the analysis.

A typical bioinformatics workflow(Pérez-Cobas et al., s. d.) of metagenomics shotgun assembly based would be:

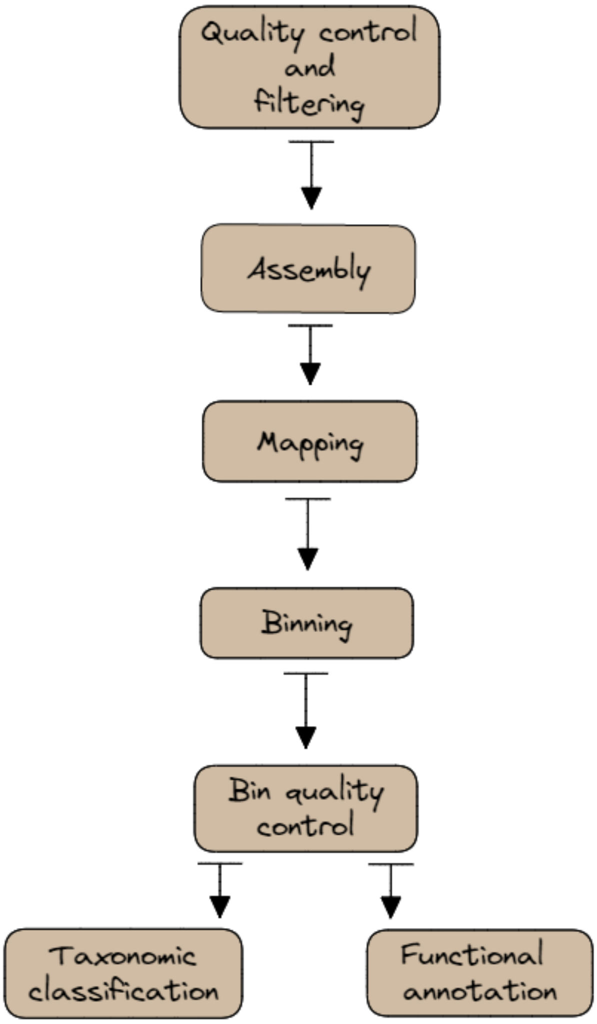


Figure 6 - Metagenomic workflow

Quality control and filtering

After sequencing our samples, we obtain reads that will pass to quality control. In this step, many things can happen to the reads:

- **Adapter trimming**
  Cutting off the adapter
- **Quality filter**
  filter reads by their quality to only have good quality reads
- **PolyG and PolyX tail trimming**
  PolyG is a common issue found with Illumina sequencing.
  X in polyX means any base of A/T/C/G
- **Duplication rate evaluation**
  This information is essential to profile the diversity of sequencing libraries.

Assembly

We take all the reads (fastq files) to create one large assembly. Tools like Spades or Flye can do this step.

There are three main ways to do assembly(*Genome Assembly - an overview | ScienceDirect Topics*, s. d.) :

- **Greedy**
  You take out a random sequence of your data and match it to all other sequences. If you find a match that extends your sequence, you include it. When you do not see any more possible extent, you stop, and what you have become a contig.

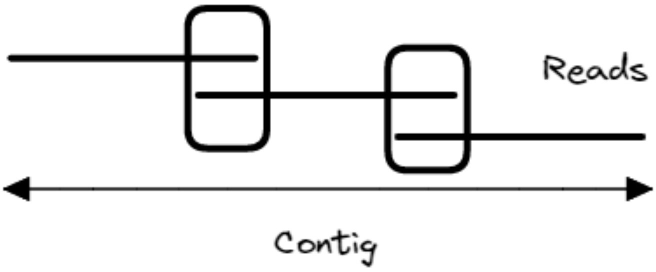


Figure 7 - greedy assembly

- **Overlap-layout-consensus (OLC)**
  This method maps all the reads against all the reads with short k-mers.
  A k-mer is a string of k length.

Figure 8 - OLC assembly

In this example, we have four k-mers with k=3
We then obtain an approximate mapping to create a layout of how we think the reads connect. After that, we need to resolve inconsistencies in the graph to have better contigs. This construction is called an overlap graph.

- **De Bruijn graph**
  In this case, we take k-mer out of reads and try to build an assembly by constructing a De Bruijn graph.

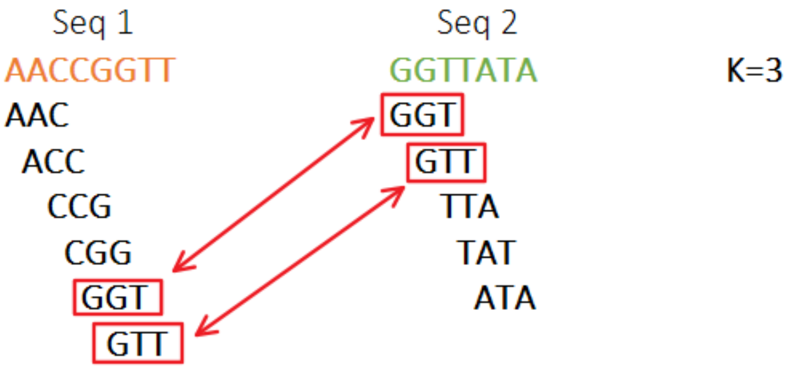


Figure 9 - De Bruijn graph explanation

We can see the two last k-mers of seq one overlap the two first k-mers of seq 2.

My sequence would be AACCGGTTATA.

On that, we can construct a De Bruijn graph (Sohn & Nam, 2016)

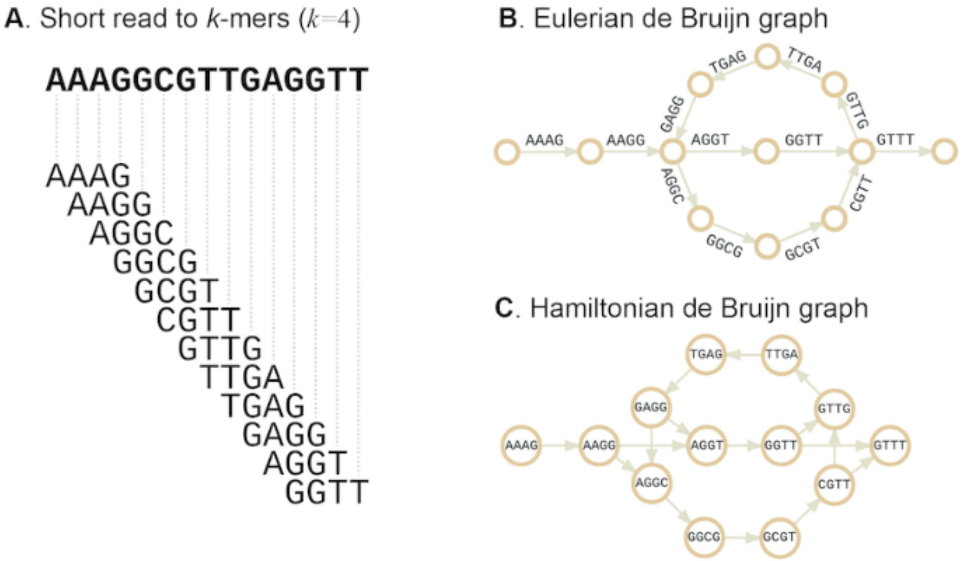


Figure 10 - Example of De Bruijn graphs

- **Hybrid assembly**
  Hybrid assembly combines short-read and long-read sequencing data to improve genome sequencing. Short reads (e.g., Illumina) provide accuracy, while long reads (e.g., PacBio, Nanopore). Hybrid assembly yields better genome quality by correcting errors in long reads and using short reads to link sequences.

Mapping

We then map all the reads against the assembly. Mapping compares every read with the reference genome or the assembled contigs used as reference. We have to do this step because the assembler doesn't track which reads contribute to which contigs.

Examples of mapping tools are bowtie2 and bwa.

Binning

In this step, we count the abundance of each contig in each sample. Then we plot a graph. This method is called differential coverage binning. Other methods exist but will not be discussed here.

We group the contigs based on similarity in different bins and other samples. The methods to group the contig vary, but one consists of counting the number of reads mapping to each contig in different samples. This gives us an estimation of the abundance of each contig and helps group contig of similar abundance across models to the same bin. Other methods exist and are often used together, but I will discuss them elsewhere.

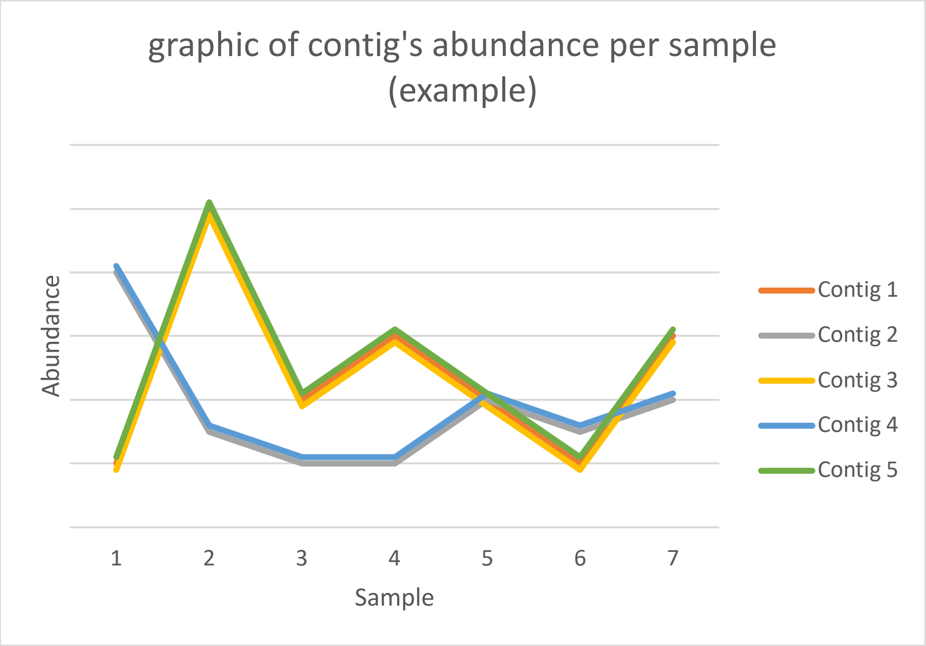


Figure 11 - Example of differential coverage binning graph

We can see in Figure 4 that the abundance of specific contigs is acting the same way. It means they belong to the same genome of the same organism.

All these contigs coming from the same genome are grouped in a bin.

We have several bin files representing each different genome at the end of this step.

The most used tools are concocted, metabat2, groopM, and crass.

Bin quality control

How do you know the bins you created are high-quality or represent one genome?

We can do that with tools like checkM. Here, you use a Collection of marker sets classified in a phylogenetic tree as a database to compare your bins with them. It will tell you if your bins are good quality, represent one genome, and contain contaminations.

Taxonomic classification

We then compare our bins with a database like GTDB to know what species we have in our sample.

Functional annotation

We use a database of orthologous groups to retrieve the genes in each bin. Those databases are also linked to functional databases like KEGG

KEGG (the most popular one). They are used to know what genes are involved. What metabolic pathway?

Bachelor thesis purpose

My work aimed to study metagenomic samples from two different biogas reactors to understand the species and genes involved in biogas formation. To achieve this goal, I used bioinformatics tools to analyze sequencing data obtained directly from the samples. This study was motivated by the environmental and economic issue of biogas, which is a renewable and sustainable energy source. By improving biogas production, we can contribute to reducing greenhouse gas emissions and the transition to a green economy. The results can be used to biogas production enhancement by identifying key species and genes involved in biogas formation. This research is therefore relevant to industry and the scientific community, as it provides valuable information to improve biogas production and contribute to the sustainability of our planet.

Materials and methods

Metagenomics is a potent approach for exploring the microbial richness and diversity within environmental samples. Among the multifaceted aspects of metagenomics, functional analysis emerges as a crucial method to characterize the genes and metabolic pathways present within a sample, delivering essential insights for optimizing biogas production.

Hardware Resources and IT Infrastructure

The execution of this functional analysis necessitated the deployment of specific hardware resources, particularly in computing systems. My laptop, an Asus Zenbook, equipped with 16 GB of RAM and an Intel Core i5 processor with eight threads, played a pivotal role. The Linux Mint operating system installed on this machine provided the conducive environment to conduct the analyses.

In addition to my laptop, I gained access to two dedicated servers, primarily for tasks demanding higher resource utilization. The first, known as the BIG Server at HEH, boasts the following specifications: 16 GB of RAM, 128 GB SWAP, and a processing power with 12 available threads. This server distinguishes itself through its round-the-clock availability and streamlined tool installation process, facilitating a smooth and efficient analytical workflow.

The second server, Planetsmasher, at SLU, possesses substantial computational capabilities. However, its usage depends on availability and other server users, resulting in queuing. The precise specifications of this server, such as RAM, swap space, and thread count, require verification. Nonetheless, its significant processing potential proves valuable for intricate tasks, albeit encountering challenges related to tool installation due to access restrictions and compatibility issues.

Samples overview

The data in question originates from two biogas reactors in Sweden, 02sw and 03sw. The sequencing process involved three techniques: Illumina (Ill), Nanopore (ONT), and RNA sequencing (RNAseq). Within the 02sw collection of four samples, each obtained at different time points, only the RNA sequencing data varies among the samples. Meanwhile, the Illumina and Nanopore sequencing data remain consistent across all four samples. Similarly, the 03sw collection, comprising two samples collected at separate time points, follows a parallel pattern. Therefore, we have six distinct scenarios. Unfortunately, due to data loss on the Planetsmasher server at SLU, the specific timeline for these samplings is still being determined.

The first step involves analyzing the fundamental data and facilitating consideration of the tools utilized. A set of tools, including seqKit stat, FastQC, and Nanoplot, will be employed to achieve this.

Tools Utilized

- seqKit stat: This tool, chosen for its rapid execution, provides fundamental insights into sequencing files. It quickly furnishes essential information such as read count and minimum and maximum read sizes, which is particularly beneficial for larger files. It applies to Ill, ONT, and RNAseq data.

- FastQC: A versatile analysis tool offering comprehensive data insights. FastQC is well-suited for this stage, as it can be applied to Ill, ONT, and RNAseq data.

- Nanoplot: Explicitly designed for Nanopore sequencing data, this tool addresses the technology's unique challenges. While FastQC excels for Illumina-based data, Nanopore's higher error rate necessitates specialized tools.

Exploration and Evaluation of Metagenomic Functional Analysis Pipelines

Initially, I considered the existence of multiple functional analysis pipelines, each boasting unique advantages and features. As part of this culminating study, I aimed to embark on an exploration of several of these pipelines – namely MG-Rast, QIIME, MetaPhlAn, Anvi'o, and MEGAN – with the ultimate goal of selecting the most suitable one for identifying organisms associated with biogas production from my sample data. In the subsequent sections, we will delve briefly into the inner workings of each pipeline, gaining a snapshot of their capabilities.

- MG-Rast: MG-Rast is an online platform that supports the processing of metagenomic data, including read assembly, functional annotation, and comparative analysis. Samples are uploaded to the platform and undergo preprocessing steps such as quality control, assembly, and taxonomic assignment. Subsequently, the data undergo functional analyses, such as metabolic pathway prediction and gene function characterization.

- QIIME: QIIME (Quantitative Insights Into Microbial Ecology) is an open-source toolkit for analyzing metagenomic data. It offers a range of tools for microbial diversity analysis, taxonomic attribution, co-occurrence network construction, and other advanced analyses. QIIME operates via a command-line interface and can perform customized analyses tailored to specific study needs.

- MetaPhlAn: MetaPhlAn is a taxonomic profiling tool that identifies microorganisms in metagenomic samples. It uses specific markers to characterize microbial communities and can provide information about the relative taxonomic composition of the samples.

- Anvi'o: Anvi'o is an analysis and visualization environment for metagenomic data. It supports assembly, functional annotation, taxonomic attribution, and interactive result visualization. Anvi'o also allows exploring data in more detail through its 3D visualization feature.

- MEGAN: MEGAN (Metagenome Analyzer) is a tool for visualizing and analyzing metagenomic data, enabling the characterization of the taxonomic composition of samples using functional and taxonomic annotation data.

After a comprehensive evaluation of multiple functional analysis pipelines, spanning MG-Rast, QIIME, MetaPhlAn, Anvi'o, and MEGAN, I extensively explored their intricacies, each pipeline revealing its distinctive potential. Over approximately two months, I extensively researched, comprehended the workflows, and attempted their implementation on the SLU server. Regrettably, unforeseen challenges emerged, impeding installing these pipelines on the SLU server. Complications stemming from compatibility issues and my restricted access to the server hindered my ability to resolve these challenges effectively.

Notably, even for MG-Rast, a platform of interest, the registration process necessitated a formal request, and it was only months later that I received a response. Faced with the complexities and hurdles encountered in my chosen approach to data analysis, I confronted the reality that a shift in strategy was imperative. Hence, I reevaluated my course of action and began formulating an alternative workflow that would better align with the constraints and demands of the project.

Exploration of a New Path: Building a Custom Workflow for Functional Analysis

Faced with the challenges encountered during pipeline installation, a new path gradually emerged. Recognizing the limitations imposed by various constraints, I decided to pivot towards an alternative approach, focused on constructing a workflow for functional analysis of my metagenomic samples. This approach led me to rethink how I handled my data completely. It sparked a profound questioning of the analysis steps, available methods, and suitable tools to achieve my objective.

Below, this workflow is detailed, highlighting the different stages, selected tools, and the considerations that guided my choices. This new direction has paved the way for an exciting exploration of functional analysis.

I first embarked on a research and information gathering phase, which took approximately two weeks to complete. Once this initial step was accomplished, I dedicated another week to crafting a theoretical workflow to realize my project.

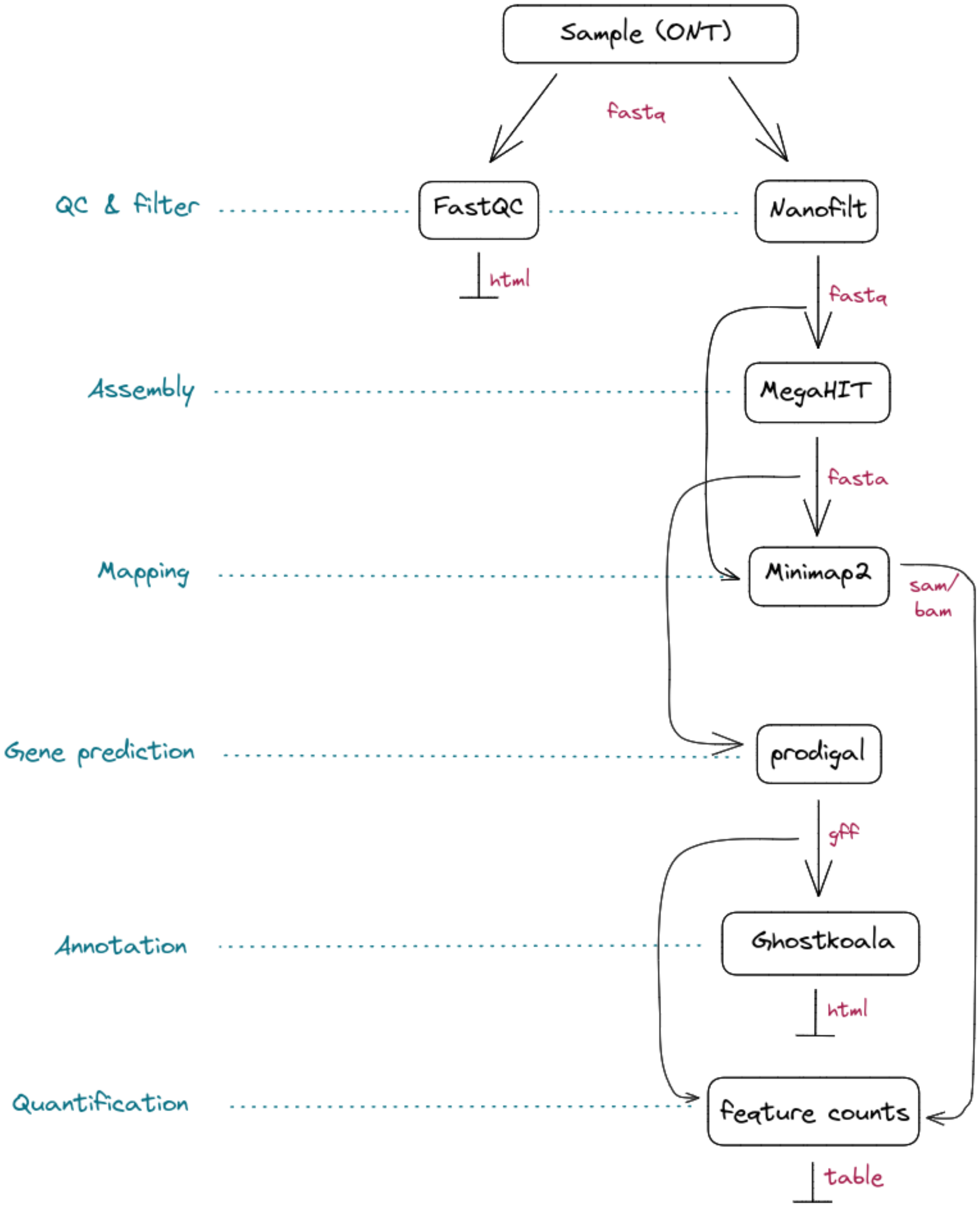


Figure 12 - First Workflow

Explanation of the first Workflow:

I intended to initiate the analysis by conducting quality control assessments on the nanopore dataset using FASTQC. This step would involve evaluating various quality metrics and identifying potential issues within the raw data. If deemed necessary, I planned to utilize nanofilt to enhance the data quality by filtering out low-quality reads and improving the overall dataset integrity.

Following the quality control phase, I aimed to perform assembly using MegaHIT. This assembly approach would reconstruct longer DNA sequences (contigs) from the fragmented nanopore reads. The resulting contigs would provide a comprehensive representation of the genomic content.

Once the assembly was completed, I intended to perform mapping using minimap2. This step would involve aligning the original nanopore reads back to the assembled contigs, enabling the determination of how well the reads correspond to the contig sequences and revealing potential variations or coverage discrepancies.

I planned to use prodigal, a gene prediction tool to predict potential genes within the assembled contigs. This process would involve identifying open reading frames (ORFs) and predicting the coding sequences within the contigs. These predicted genes would serve as a basis for subsequent functional annotation.

Functional annotation was another crucial step in my plan. I aimed to utilize ghostKOALA, a useful annotation tool, to assign potential functions to the predicted genes. GhostKOALA uses a database of known protein sequences to infer functional annotations based on sequence similarities. This step would provide insights into the genes' potential biological roles and functions within the metagenomic dataset.

For quantification analysis, I intended to use feature counts. This approach involves aligning the original nanopore reads to the predicted genes and quantifying the expression levels of each gene. This quantitative information would offer insights into the abundance and activity of specific genes in the metagenomic samples.

Lastly, I planned to integrate BLAST analysis into my workflow. BLAST would allow for a more in-depth investigation. This step would help identify organisms.

However, during the implementation of this workflow, I quickly needed a sufficient understanding of its operation, leading to confusion in my analysis. In the end this workflow needed to be corrected. This phase spanned three weeks. Through in-depth reflections and discussions with my mentor, Mr. Coornaert, it became evident that simplifying my approach and progressing step by step using more straightforward tools, while meticulously examining inputs and outputs was necessary.

As a result, the methodology evolved from a theoretical workflow followed by its implementation to an approach where the workflow was developed progressively as each tool was employed. This approach also prompted me to explore certain tools' specific performance, aiming to grasp their functioning better and maximize their utility for my analysis.

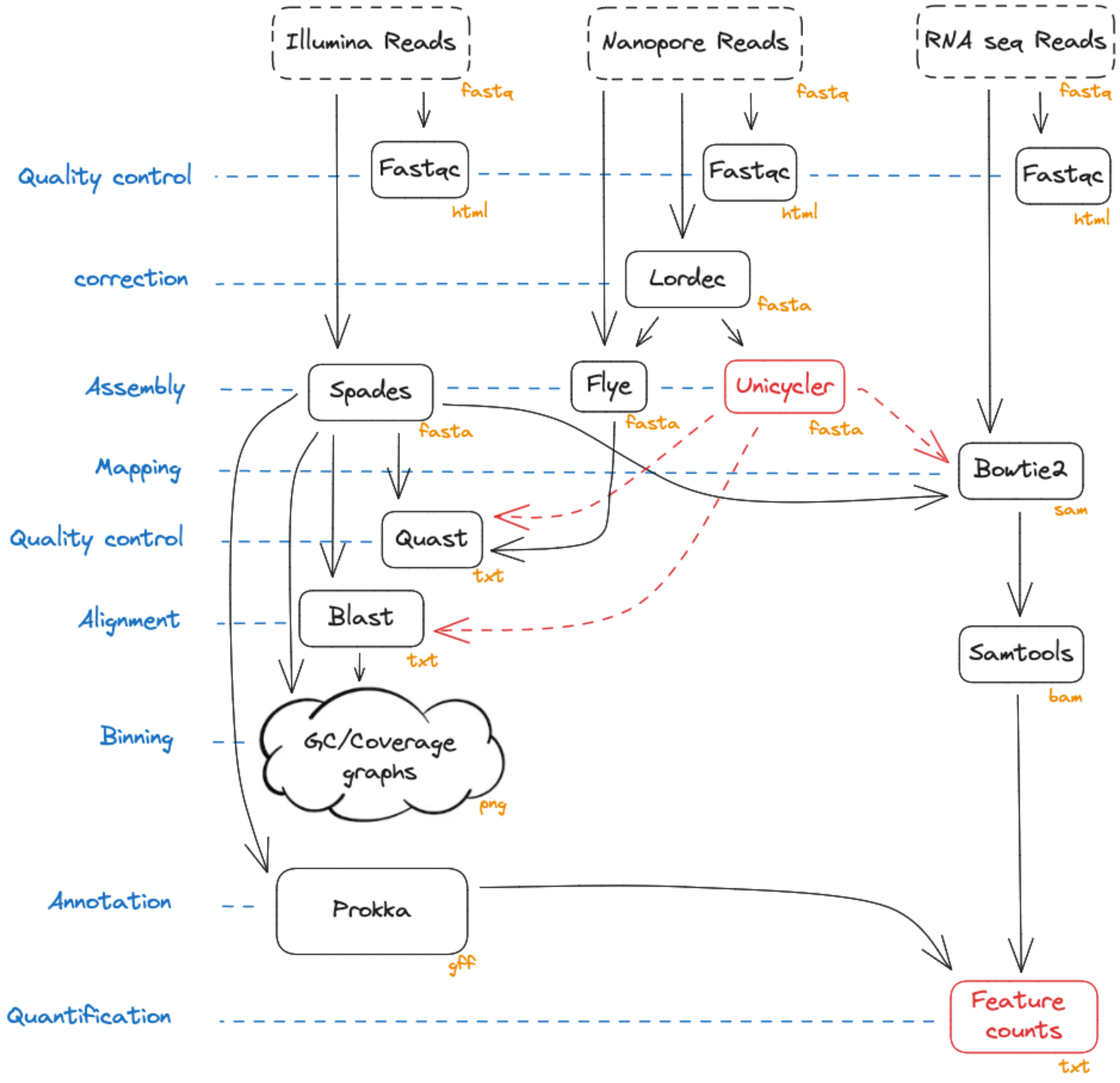


Figure 13 - Second workflow

Explanation of the new Workflow and List of Tools Used:

1. Quality control

   Quality control is a pivotal initial step in the workflow, crucial for comprehending the raw data obtained from various sequencing types, including Illumina, Nanopore, and RNA-seq. To execute this task, I opted for the tool FastQC.

   FastQC, short for Fast Quality Control, is a widely utilized bioinformatics tool that profoundly analyzes the quality of sequencing data. It furnishes a series of metrics and visual graphs to evaluate base quality distribution, quality score distribution, ambiguous bases, repeated base sequences, adapters, and other artifacts. This thorough analysis aids in grasping the overall quality of the raw data.

   In this context, FastQC was applied to all sequencing categories of the samples. As an output file, an HTML link is generated containing all the information gleaned through this tool.

2. Correction

   After evaluating my data using FastQC, an idea emerged. The notion was to enhance Nanopore reads by correcting them with Illumina sequencing data.

   I employed Lordec(Lin & Liao, 2015), a practical tool designed to correct long reads using short-read data to enhance this. Lordec follows a hybrid approach, where it corrects Nanopore reads, known for their length but lower reliability, using Illumina reads, which are shorter yet highly reliable. Lordec aims to diminish errors in Nanopore reads while preserving their characteristic length by merging these two types of data. I adopted this approach to prepare optimized reads for a hybrid assembly step.

   In the "Results" section, the observations of this process and the comparison between corrected and uncorrected reads will be presented in detail.

3. Assembly

   Regarding assembly, I thought about two ways to go. The first one is a simple assembly for short reads with SPAdes(Bankevich et al., 2012), and the second one is a hybrid assembly using Unicycler(Wick et al., 2017).

SPAdes, the St. Petersburg genome assembler, is a widely used software for assembling short reads into longer sequences known as contigs. For this process, Illumina reads are used as input. These contigs provide an overview of the genomic composition of the samples and are subsequently subjected to more in-depth analyses.

As for Unicycler, it's a specialized tool for hybrid assemblies that combines long reads (Nanopore) with short reads (Illumina) to produce more comprehensive and accurate assemblies, including circular sequences. In this case, Unicycler utilizes the Nanopore corrected file generated by Lordec (in fasta format) and the paired-end Illumina files as input. However, despite its potential, I encountered difficulties during its implementation.

Unfortunately, the assembly step with Unicycler failed due to hardware limitations. It proved to be too slow on the BIG server, and I could not make it work on the Planetsmasher server or the Galaxie platform in Slovakia. Consequently, I had to proceed with my analyses using only the data from SPAdes. In my report, I indicated the alternative path I would have taken had Unicycler functioned successfully, using dashed red arrows to illustrate this hypothetical sequence.

Furthermore, I analyzed the runtime of SPAdes under different conditions. Specifically, I investigated the impact of file length and the number of processing threads on the runtime of the assembly process using SPAdes. The results of this analysis are detailed in the "Results" section of this report. By exploring these factors, I aimed to optimize the computational efficiency of the assembly process, thereby contributing to the overall efficiency of the workflow.

4. Assembly quality control

Following the assembly with SPAdes, the next step involved evaluating the quality of the resulting assemblies. I employed QUAST, the Quality Assessment Tool for Genome Assemblies(Gurevich et al., 2013). It considers various metrics such as contig length, coverage, and assembly errors.

5. Alignment

Next, I performed alignment using BLAST on my SPAdes output files (contigs. fasta), utilizing the nt and env_nt databases.

BLAST, or Basic Local Alignment Search Tool, is a fundamental tool designed to compare and align biological sequences against a reference database.

This enables researchers to deduce functional annotations, homologies, and evolutionary relationships.

The nt database (nucleotide database) is an extensive repository of sequences compiled from various sources, including sequences from GenBank, EMBL, and DDBJ databases. It is a comprehensive reference for the vast diversity of genetic material across organisms. In contrast, the env_nt database focuses explicitly on environmental sequences, encompassing a wide range of genetic material obtained from various ecological niches, enhancing our understanding of the genomic landscape of diverse ecosystems. Observing these two databases proved relevant, although others, such as the plasmid database, would have been intriguing to explore.

The objective of using BLAST was to assign specific organisms to my contigs, with a particular emphasis on detecting organisms associated with biogas production within my samples.

6. Binning

I then transitioned to the binning phase, where I began the process without using specialized tools. To accomplish this, I compiled tables for each contig, capturing its GC percentage, the coverage attained through SPAdes, and the organism identified via BLAST. Subsequently, I generated graphical representations plotting %GC against coverage. To achieve this, I developed multiple scripts provided as appendices to this report. I conducted this procedure for both the nt and env_nt databases.

Binning, or grouping, aims to separate contigs into distinct sets that may originate from different organisms within the metagenomic sample.

The %GC measures the GC base pair content in a DNA sequence. It can vary significantly among different species and genera of organisms. Coverage, however, represents the number of times a specific sequence is represented in the sequencing data. When creating a graph with %GC plotted against coverage, a visualization is obtained that can reveal trends and clusters within the data.

By employing this approach, contigs with similar characteristics in terms of %GC and coverage tend to cluster in specific regions of the graph. These groupings can suggest the presence of groups of organisms sharing similar genomic features, potentially corresponding to distinct microorganisms. Thus, observing patterns in the %GC/Coverage graph makes it possible to identify and segregate contigs belonging to different organisms, which forms the core of the binning process.

7. Annotation

Genomic annotation involves assigning biological functions to an organism's genes by identifying coding regions and functional elements within its DNA sequences. To do so, Prokka(Seemann, 2014) was used.

This step is done from SPAdes output to continue toward quantification.

8. Mapping

After the assembly with SPAdes, using BLAST, and my %GC/coverage graphs, I focused on the fate of the RNAseq reads. I contemplated the possibility of mapping them onto the corresponding contigs.fasta assembly derived from SPAdes. For this purpose, I opted to employ Bowtie2, a tool to achieve efficient and accurate reading alignment to a reference genome or assembly.

However, using Bowtie2 encountered challenges, particularly on the Planetsmasher server. Despite my endeavors, the Bowtie2 process on this server inexplicably terminated after a certain period. To address this issue, I sought to reduce the processing load by segmenting my files into smaller fragments, hoping to make the task more manageable. Regrettably, this approach proved unsuccessful as Bowtie2 began to interpret my files as corrupted. This issue persisted on the BIG server as well.

To this date, I am still working towards resolving this issue.

9. Quantification

If I had successfully managed to run Bowtie2, I would have obtained a SAM file containing, among other things, precise alignments of RNAseq reads to the assembly. Each line of the SAM file would have described a specific alignment with detailed information about the position, alignment quality, read sequence, and other characteristics. Then, I would have needed to convert the SAM file to BAM using samtools. This BAM file would have served as an input for the tool FeatureCounts(Liao et al., 2014).

FeatureCounts is a bioinformatics tool designed to quantitatively assess gene or element expression levels from RNA sequencing (RNAseq) data. By aligning RNAseq reads to an assembly, FeatureCounts generates a counting table that quantifies the number of reads associated with each element. This counting table forms the basis for subsequent analyses, enabling researchers to compare gene expression levels across different samples, conditions, or experimental setups (differential analysis).

While BLAST provides valuable insights into the presence of specific organisms in the samples, FeatureCounts would have extended this understanding by quantifying the expression levels of their genes or elements. This combination would have allowed me to identify present organisms and evaluate their activity and potential contributions to biogas production.

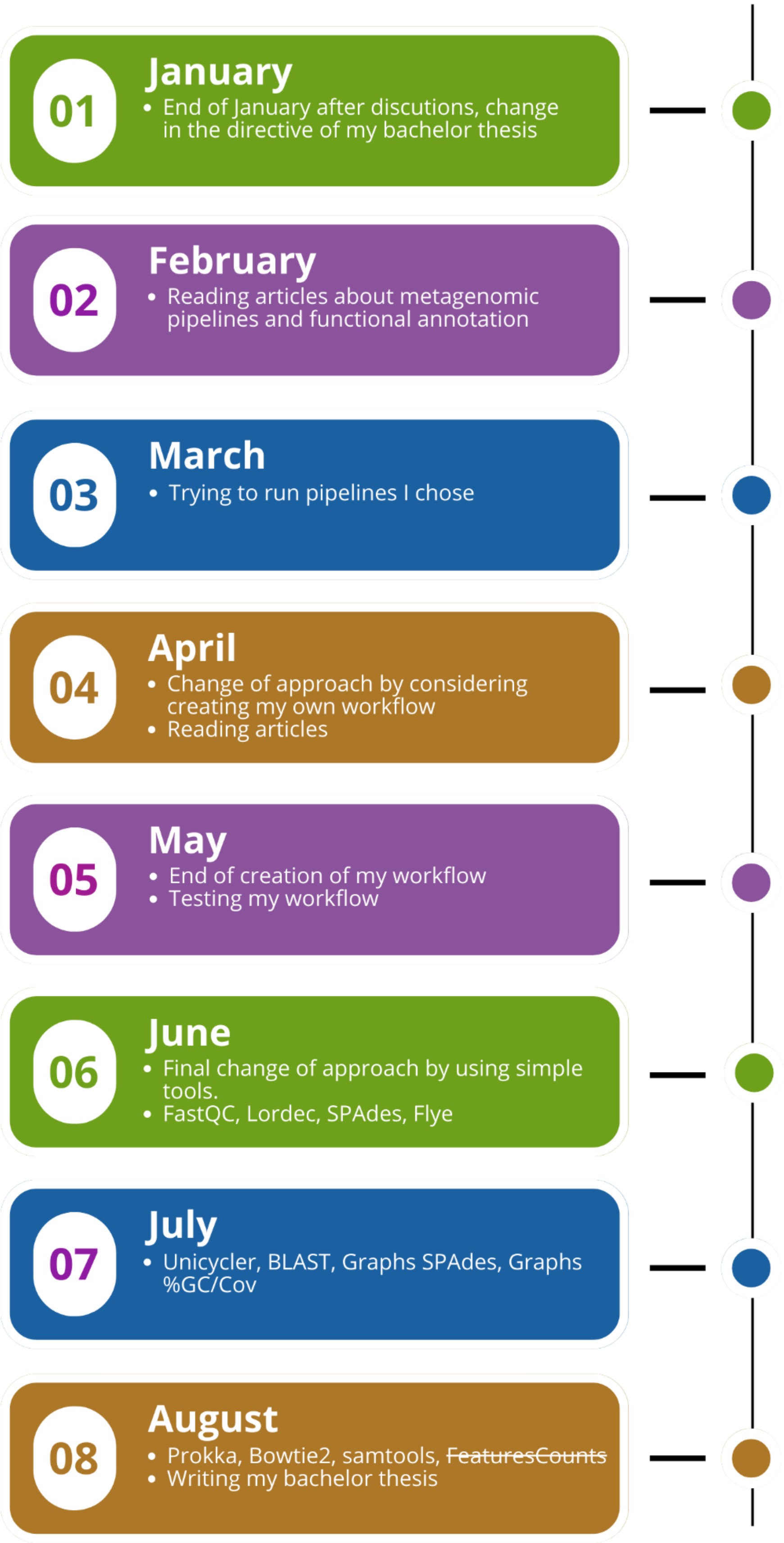
Analyzing the six different RNAseq samples this way would have been quite interesting.

10. Differential analysis

Looking ahead to the following steps, I have considered the possibility of conducting a differential analysis to identify genes or elements whose expression significantly varies across the samples. However, as I conclude my thesis work, this step remains to be seen due to time constraints. Before proceeding, I must understand the methods and parameters required to ensure relevant results.

We will now transition to the results generated by constructing and exploring the workflow. This section will showcase the insights gained from the meticulous application of chosen analysis tools and approaches, providing an overview of the organisms in my biogas production samples. Furthermore, we will assess the performance of selected tools.

Timeline of work



01 **January**

- End of January after discutions, change in the directive of my bachelor thesis

02 **February**

- Reading articles about metagenomic pipelines and functional annotation

03 **March**

- Trying to run pipelines I chose

04 **April**

- Change of approach by considering creating my own workflow
- Reading articles

05 **May**

- End of creation of my workflow
- Testing my workflow

06 **June**

- Final change of approach by using simple tools.
- FastQC, Lordec, SPAdes, Flye

07 **July**

- Unicycler, BLAST, Graphs SPAdes, Graphs %GC/Cov

08 **August**

- Prokka, Bowtie2, samtools, ~~FeaturesCounts~~
- Writing my bachelor thesis

Results

This section provides an overview of the achievements and insights from the various analyses.

Quality control with FastQC

The results presentation starts with quality control on our sample from distinct read types (Illumina, Nanopore, and RNAseq), conducted using the FastQC tool. This phase delivers a glimpse into the data quality and forms an indispensable bedrock for subsequent analysis.

a. Illumina reads

		Illumina (R1 ou R2)				
	# sample	**Taille(fq.gz)**	**# Reads**	**# total base**	**seq length**	**%GC**
03sw[SE]	bm01	2.3Go	30748424	4643012024	151	50
	bm02					
02sw[SE]	bm03 à 06	1.4Go	18946658	2860945358	151	46

Table 1 - Quality control on Illumina reads

As shown in Table 1, there is almost twice the quantity of reads for 03sw than for the 02sw dataset. Yet, it is a small volume of data to work with. This is good to know because fewer computer resources are needed with less data volume.

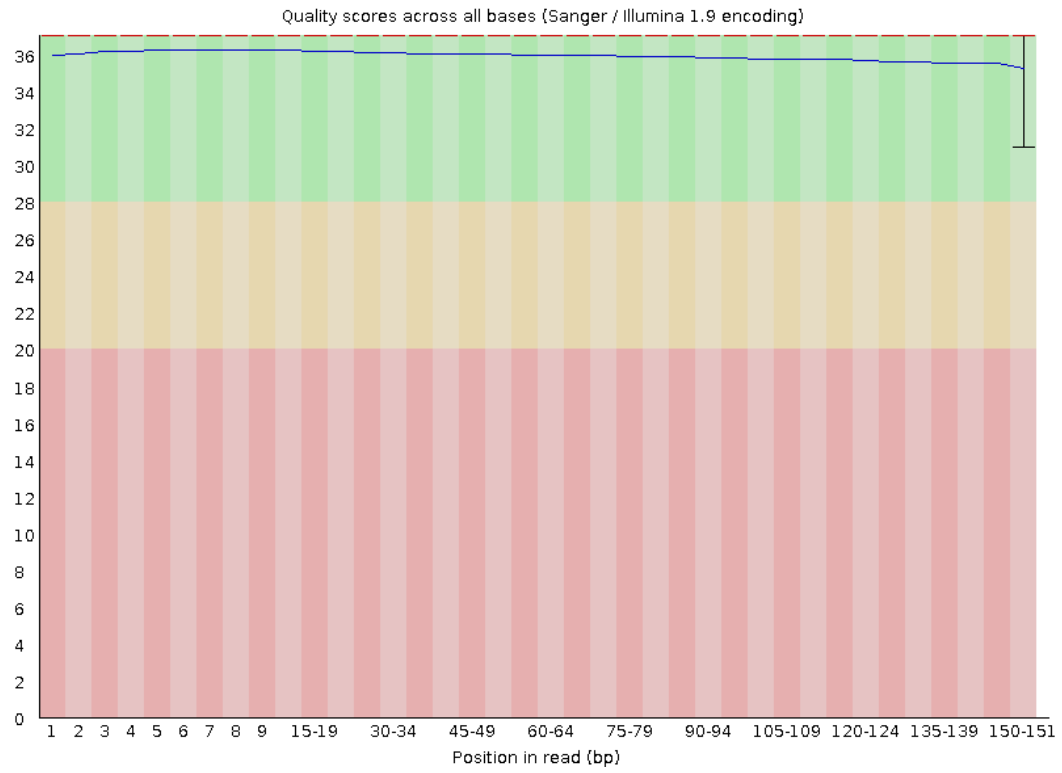
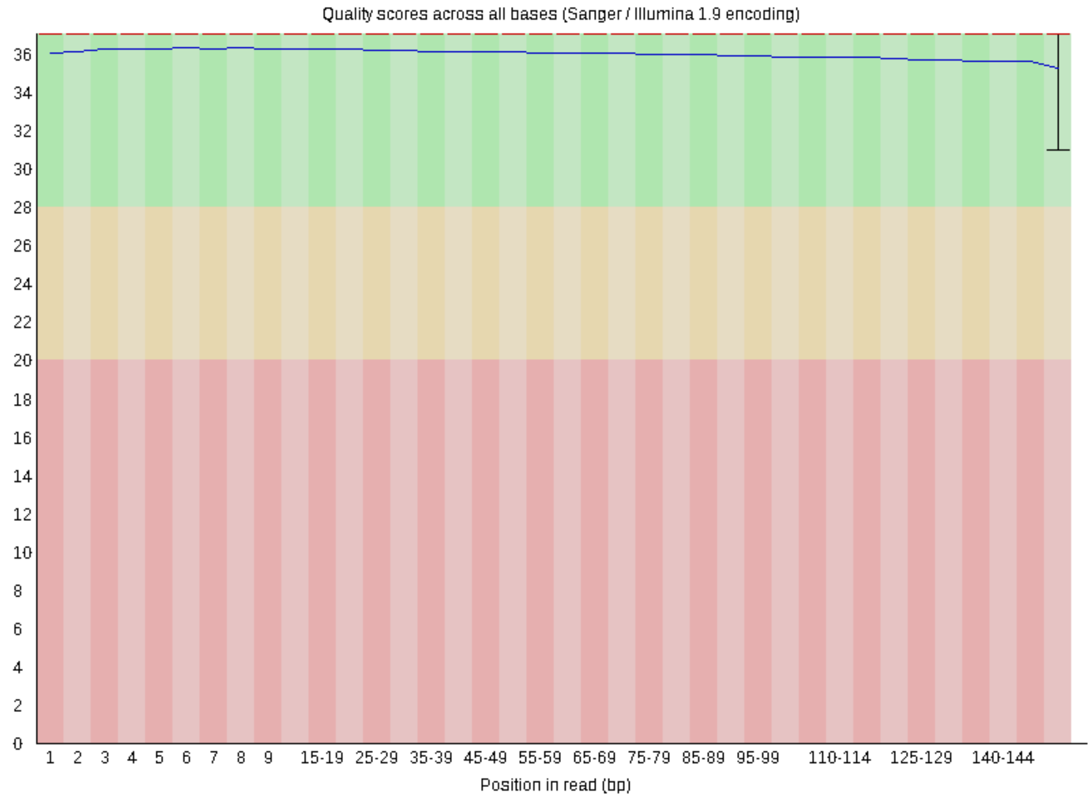
Graph Per base sequence quality of 03sw

Graph Per base sequence quality of 02sw


Table 2 - Graphs per base sequence quality of 02sw and 03sw Illumina reads

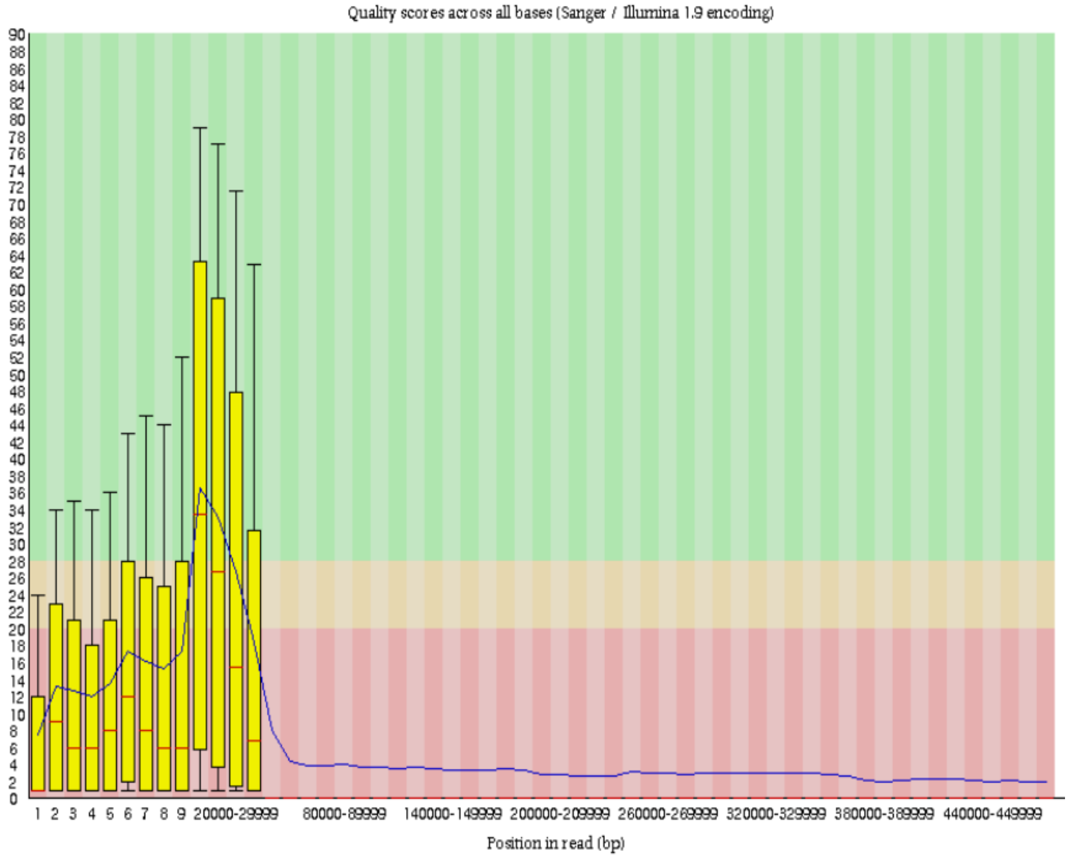
As we can observe from the above graphs in Table 2, the overall quality, whether for 02sw or 03sw, is excellent. Therefore, there is no need to correct the files. These files can be directly assembled via Spades.

b. Nanopore reads

Nanopore								
Samples		**Taille (fq.gz)**	**# reads**	**# total base**	**long min**	**long max**	**avg len**	**%GC**
03sw	bm01	7.7Go	2145259	7147012355	3	477830	3331	47
	bm02							
02sw	bm03 to bm06	20Go	3632882	20081076641	3	535220	5527	44

Table 3- Quality control on Nanopore reads

Per the base sequence quality of 03sw,



Per base sequence quality of 02sw

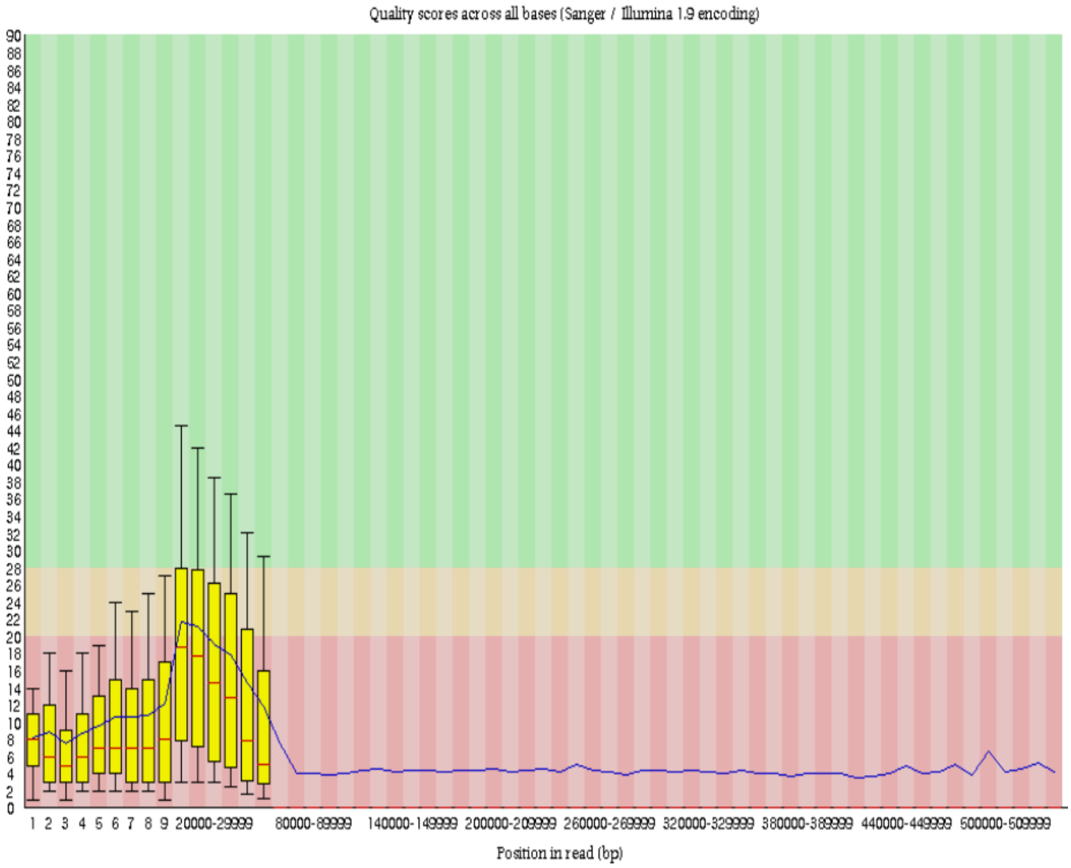


Table 4 - Graphs per base sequence quality of 02sw and 03sw Nanopore reads

In this case, we can observe that the nanopore sequences have an inferior quality. However, it's important to note that nanopore technology inherently yields higher error rates than Illumina. Additionally, FastQC's quality assessment is calibrated regarding Illumina sequencing, resulting in a higher quality threshold. For nanopore sequences, we can consider the quality of these files to be good. Therefore, they can directly be assembled with Flye, or Lordec can still improve them before assembly.

c. RNAseq

		RNAseq			
# sample		**Taille (fq.gz)**	**# Seq**	**seq length**	**%GC**
03sw[SE]	bm01	170Mo	3651202	76	49
	bm02	135Mo	3005628	76	52
02sw[SE]	bm03	235Mo	4973956	76	48

		RNAseq			
# sample		Taille (fq.gz)	# Seq	seq length	%GC
	bm04	219Mo	4630110	76	48
	bm05	170Mo	3559689	76	53
	bm06	180Mo	3815060	76	54

Table 5 - Quality control on RNAseq reads

Per the base sequence quality of 03sw,

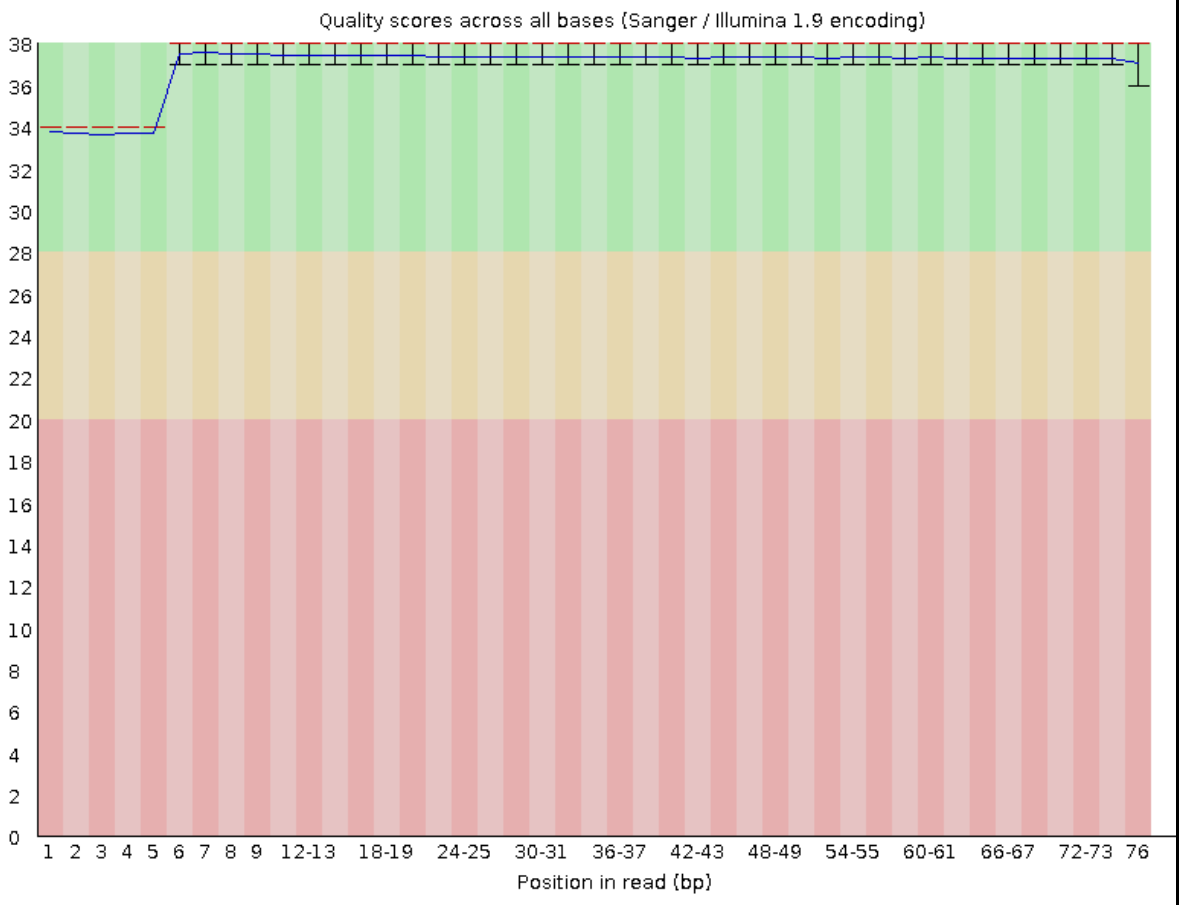


Table 6 - Graphs per base sequence quality of RNAseq reads

Regarding the RNAseq files, we can observe excellent base quality. This implies there is also no need to correct the reads.

Correction with Lordec

To enhance the quality of nanopore reads, I delved into the tool Lordec, as mentioned in the materials and methods section. Lordec is a hybrid correction tool that utilizes short reads to rectify long ones. The natural question was whether the use of Lordec indeed leads to a substantial difference. Consequently, a comparison was conducted between the approaches with and without correction, employing Flye for assembly and evaluating assembly quality using Quast. This inquiry sought to determine the concrete impact of Lordec on the accuracy and reliability of outcomes, thereby highlighting its potential role in improving nanopore analysis performance.

	Quality of assembly (by Flye)	
	Without Lordec	**With Lordec**
Total length	219271382	198305821
#contigs	6450	6049
# contigs (>= 1000 bp)	5951	5483
#contigs (>= 50000 bp)	931	857
%GC	46.43	46.36
N50	98807	109793
L50	408	340

Table 7 - Quality control of assembly on 02sw Nanopore reads with or without Lordec

Total length: The assembly with Lordec resulted in a slightly smaller full length (198,305,821) than without Lordec (219,271,382). This could indicate that Lordec's inclusion helped refine the assembly by possibly removing redundant or erroneous sequences.

Number of total contigs, short and large contigs: The assembly with Lordec produced fewer contigs (6,049) compared to the assembly without Lordec (6,450)

The count of contigs with a length of 50,000 bp or more was also lower with Lordec (857) than without Lordec (931). On the other hand, the count of short contigs (>= 1000 bp) with Lordec is also lower than without. This suggests that Lordec's correction likely contributed to the generation of longer and more significant contigs.

%GC content: Both GC content are similar, indicating that Lordec's corrections didn't affect the overall GC content.

N50 and L50: N50 represents the length at which half of the total sequence assembly is contained in contigs. L50, on the other hand, is the number of contigs needed to reach or exceed the N50 length.

In this case, the N50 value increased in the assembly with Lordec (109,793) compared to without Lordec (98,807). This suggests that the assembly with Lordec contains longer contigs that collectively cover a significant portion of the genome. The decrease in the L50 value implies that a smaller number of contigs contribute to most of the the assembly's length.

In summary, incorporating Lordec in the assembly process has positively impacted the contigs' quality, length, and coherence. This is reflected in the lower count of contigs and the increased N50 value, indicating an improvement in the overall assembly quality.

Assembly and quality control assembly

During this phase, I wanted to employ two distinct assembly strategies. The initial approach encompassed a more traditional assembly using Spades with short reads and Flye for long reads. The other method involved a hybrid assembly using Unicycler.

1. **Hybrid assembly (Unicycler)**

   Regrettably, I couldn't fully explore this hybrid assembly approach due to technical issues mentioned in the methods section. However, it's worth noting that it would have been interesting to pursue, given the previous success of Lordec in significantly improving assembly quality.

   Consequently, I let go of long reads since the hybrid method I had in mind stopped with Unicycler due to a lack of time to research other solutions.

2. Simple assembly (Spades)

Let's begin by shedding light on the performance of Spades. We will gain a concrete overview of its performance through an informative table and graph.

Number of thread	Time (min) With correction	Time (min) Without correction
1	250	129
2	187	84
4	95	56
6	68	45
8	58	40
10	51	39
12	46	36

Table 8 - Execution time of Spades as a function of threads with or without correction

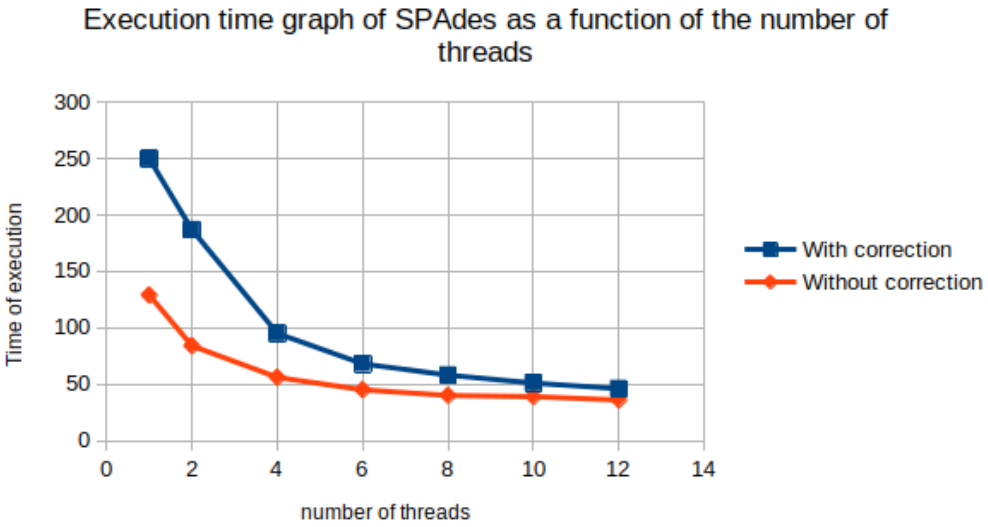


Figure 14 - Execution time graph of SPAdes as a function of the number of threads

Regarding the execution time of SPAdes in relation to the number of threads and whether the correction option is enabled, we observe a more pronounced increase in execution time under four threads. Beyond 6 threads, execution times show slight differences but decrease with increasing thread count. This leads me to conclude that on less powerful machines, four threads are sufficient. If the computer has 12 threads, execution time can be optimized by allocating resources to 6 threads, allowing for the concurrent execution of other tools and optimizing overall workflow efficiency.

Additionally, we notice that turning off the correction option yields a significant time saving when using six allocated threads. Beyond this point, time differences are marginal. This is particularly useful when working with high-quality data. However, it's worth noting that disabling correction does impact assembly quality, as indicated in the table below.

	02sw assembled with correction	02sw assembled without correction
Total length	36698079	37031184
# contigs	19920	21117
# contigs (>50000bp)	51	46
GC(%)	45.61	45.55
N50	3460	2964
L50	1345	1715

Table 9 - Quality control on assembly of 02sw with and without correction

Indeed, it's evident that the number of contigs is higher, the N50 value is lower, and the L50 value is higher. This indicates more significant fragmentation and, consequently, a less coherent assembly.

Quantity of reads (Million)	Time (min) when normal mode	Time (min) when meta mode
2	47	32
4	81	58
6	109	85
8	140	108
10	166	157
12	196	160

Table 10 - Execution time of Spades as a function of quantity of reads with normal and meta mode

Regarding the execution time of SPAdes based on the number of reads, whether opting for the standard or metagenomic mode, the execution time increases seemingly in direct proportion to the number of reads. Notably, using the metagenomic mode, explicitly designed for metagenomic data, results in lower execution times. This proves advantageous in our case.

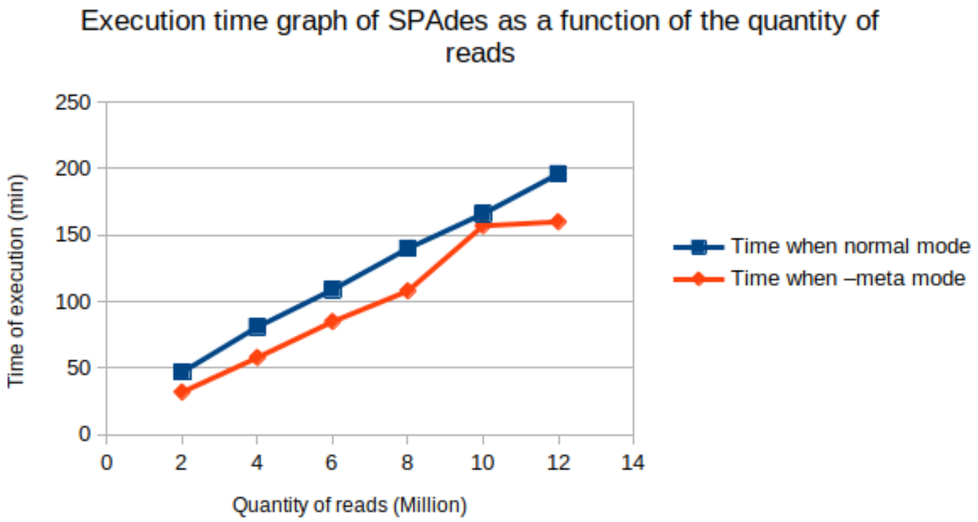


Figure 15 - Graph of execution time of SPAdes as a function of quantity of reads with normal or meta mode

I will now proceed to analyze the quality of the performed assemblies.

	Contigs de 02sw	Contigs de 03sw
Total length	132132700	497252448
# contigs	72413	35515
# contigs (>50000bp)	184	302
GC(%)	47.01	50.52
N50	3224	1781
L50	5583	47599

Table 11 - Quality control on assembly of 02sw and 03sw

The "02sw" assembly consists of 72,413 contigs, totaling 132,132,700 bp. The N50 is 3224 bp, signifying that half of the total length lies within contigs of at least this size. The L50 is 5583, indicating that 5583 contigs must reach half the total length. The GC percentage is 47.01%.

The "03sw" assembly is composed of 35,515 contigs, with a total length of 497,252,448 bp. The N50, at 1781 bp, suggests that half of the total length is within contigs of this size or more significant. The L50 is 47,599, meaning that 47,599 contigs must reach half the total length. The GC percentage is 50.52%.

Alignment

This new section addresses the alignment step using the BLAST tool to gain an in-depth insight into the organisms in my samples. This phase goes beyond being merely informative as it lays the groundwork for the subsequent pivotal step: binning.

The synergy between information gathered from Spades and data obtained through BLAST enabled binning, albeit in a rudimentary yet notably meaningful manner. This approach allowed me to identify specific individuals present in the sample. The results obtained after comparison with the env_nt database proved to be promising.

Below, I present the results for the first five NODES.

02sw aligned to env_nt		
NODE	Accession code	Organism
1	FZPS01000177.1	metagenome genome assembly, contig: metabat_cluster_49
2	FZPS01000177.1	metagenome genome assembly, contig: metabat_cluster_49
3	FZPS01000203.1	metagenome genome assembly, contig: metabat_cluster_72
4	FZPS01000177.1	metagenome genome assembly, contig: metabat_cluster_49
5	FZPS01000234.1	metagenome genome assembly, contig: metabat_unclustered_pseudo_scaffold_1

Table 12- Blast results on 02sw ; First 5 Nodes aligned to env_nt

Regarding Table 12, They are all part of the project PRJEB21678, associated with a publication titled: "Microbiome dynamics and adaptation of expression signatures during methane production failure and process recovery"(Grohmann et al., 2018). In other words, this indicates that we are on the right track, and methane-producing organisms will be present.

02sw aligned to nt		
NODE	Accession code	Organism
1	AP024619.1	Gelria sp. Kuro-4
2	CP094380.1	Pseudodesulfovibrio tunisiensis strain RB22 chromosome
3	CP032760.1	Halocella sp. SP3-1 chromosome, complete genome
4	AP014924.1	Limnochorda pilosa DNA, complete genome, strain: HC45
5	CP068564.1	Keratinibaculum paraultunense strain KD-1 chromosome, complete genome.

Table 13 - Blast results on 02sw; First 5 Nodes aligned to nt

- **Gelria sp. Kuro-4**

This mesophilic bacterium (Yamada et al., 2021) was isolated from a Thermophilic Anaerobic Digestion Reactor.

According to NCBI, Regarding its involvement in biogas formation, I found that Gelria doesn't produce methane. Still, it's a glutamate-degrading organism linked to converting glutamate into volatile fatty acids (VFAs) or acetate, implying its participation in acidogenesis and acetogenesis.

Gelria also possesses other enzymes, such as amido hydrolase, carbohydrate hydrolase, and glycoside hydrolase, which break down matter into simpler elements. These hydrolases are crucial in the initial step of biogas formation.

- **Pseudodesulfovibrio tunisiensis strain RB22 chromosome**

Pseudodesulfovibrio tunisiensis is a mesophilic anaerobic bacterium belonging to the Desulfovibrionaceae family. It was isolated from a wastewater refinery.

This organism doesn't produce methane. However, it utilizes lactate to convert it into acetate through lactic acid fermentation(Ben Ali Gam et al., 2009). This process also releases Hydrogen, which methanogenic bacteria can use to produce CH4 (methane).

- **Halocella sp. SP3-1 chromosome, complete genome**

Halocella is a cellulose and starch-degrading bacterium isolated from a hypersaline evaporation pond (Heng et al., 2019).
This fact means this organism is involved, at least in the first step of biogas production.

- **Limnochorda pilosa DNA, complete genome, strain: HC45**

Limnochorda Pilosa is a thermophilic bacterium. According to the BRENDA(Chang et al., 2021) database, this bacteria can produce methane.

- **Keratinibaculum paraultunense strain KD-1 chromosome, complete genome.**

  This thermophilic Tissierellaceae family bacterium was isolated from grassy marshland soil.

  According to BRENDA, this bacterium can perform acetate fermentation, which consequently results in the production of acetate. Methanogenic organisms can utilize this substrate to generate biogas. Therefore, this organism could be a promising candidate within the biogas production pathway.

By examining the first five organisms in the contig list, we identify two bacteria directly involved in methane production and three others involved in precursor steps of biogas production, such as hydrolysis and acetogenesis. This observation implies a substantial presence of methanogenic microorganisms within the studied sample, which is promising news. However, a more in-depth analysis and further investigations would be necessary to confirm their precise role in methane production.

03sw aligned to env_nt		
NODE	Accession code	Organism
1	AMWB02014822.1	Bioreactor metagenome contig_131279, whole genome shotgun sequence
2	CAMQFK010000013.1	sediment metagenome genome assembly
3	CAKYTT010000003.1	soil metagenome genome assembly, contig: bog_MAG_0369_000000000003, whole genome shotgun sequence
4	FPLM01004137.1	metagenome genome assembly, contig: 4137, whole genome shotgun sequence
5	CALOEM010000049.1	metagenome genome assembly, contig: contig_16947, whole genome shotgun sequence

Table 14 - Blast results on 03sw ; First 5 Nodes aligned to env_nt

Unlike the analysis of 02sw, the environments found for 03sw do not match the same projects; on the contrary, they seem to span various directions. For instance, CAMQFK010000013.1 is isolated from a benzene-degrading bioreactor, whereas FPLM01004137.1 has an unknown isolation source. The question arises whether this is a sample originating from a source other than a biogas reactor. As access to raw database information is unavailable, doubt remains.

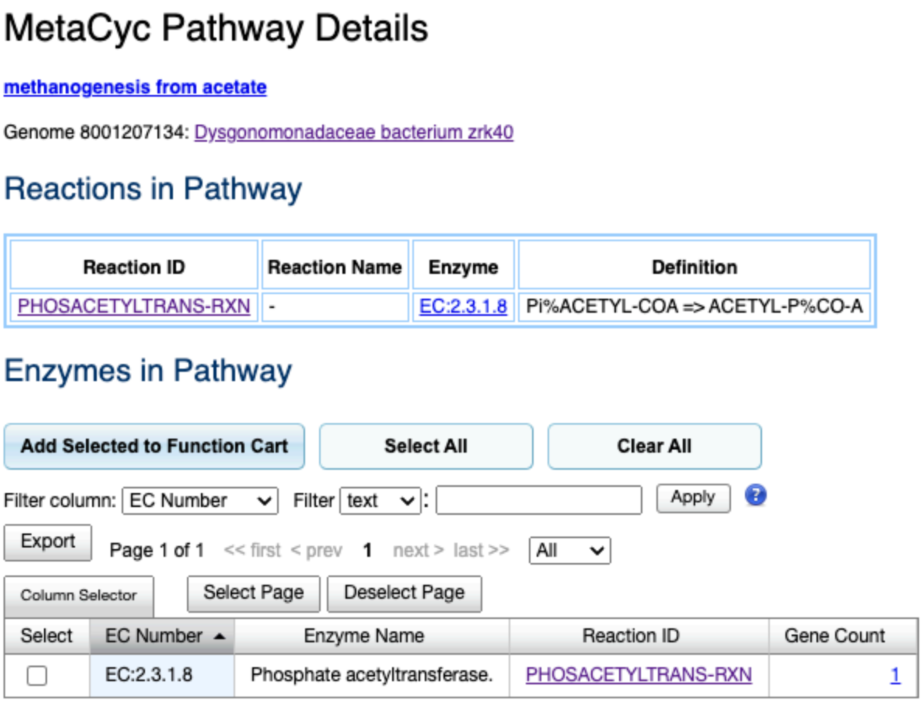
03sw aligned to nt		
NODE	Accession code	Organism
1	CP070276.1	Dysgonomonadaceae bacterium zrk40 chromosome
2	AP027081.1	Holophagaceae bacterium W786 DNA
3	CP007451.1	Draconibacterium orientale strain FH5T
4	LR634170.1	uncultured bacterium partial 16S rRNA gene
5	CP070763.1	Candidatus Latescibacteria bacterium isolate bin260 chromosome

Table 15 - Blast results on 03sw; First 5 Nodes aligned to nt

- **Dysgonomonadaceae bacterium zrk40 chromosome**

  It's a bacterial isolate from cold seep. Searching in MetaCyc, I could find a gene likely to be linked to methanogenesis from acetate. It's the only one connected to MetaCyc, knowing that many genes from this bacterium were not related to MetaCyc, meaning it could have other genes involved in this process.

  My doubt regarding this gene connected to methanogenesis relies on the enzyme name Phosphate acetyltransferase. This enzyme is involved in more extensive processes than methanogenesis.



MetaCyc Pathway Details

methanogenesis from acetate

Genome 8001207134: Dysgonomonadaceae bacterium zrk40

Reactions in Pathway

Reaction ID	Reaction Name	Enzyme	Definition
PHOSACETYLTRANS-RXN	-	EC:2.3.1.8	Pi%ACETYL-COA => ACETYL-P%CO-A

Enzymes in Pathway

Select	EC Number	Enzyme Name	Reaction ID	Gene Count
□	EC:2.3.1.8	Phosphate acetyltransferase.	PHOSACETYLTRANS-RXN	1

Figure 16 - MetaCyC pathway regarding Dysgonomonadaceae to methanogenesis

There is no further evidence that this bacterium is directly related to methane production. Nonetheless, It presents different hydrolases like amido hydrolase or glycoside hydrolase, meaning it can play a role in the precursor step of degrading polymer into monomer.

- **Holophagaceae bacterium W786 DNA**

  This organism is a bacterium isolated from river sediment, and there is little information regarding its involvement in methane production. Still, as much bacteria it possesses, hydrolase plays a role in degrading polymers.

- **Draconibacterium orientale strain FH5T**

  Draconibacterium(Du et al., 2014) is a mesophilic bacteria isolated from the marine environment in China. By looking on the BioCyc website, we can search pathways. I looked for pathways related to direct methane production but needed help finding something. On the other hand, the starch degradation pathway and hydrolysis can play a role in the precursor step of hydrolysis.

- **uncultured bacterium partial 16S rRNA gene**

  There is no information on NCBI except that it was isolated from a wastewater treatment system.

- **Candidatus Latescibacteria bacterium isolate bin260 chromosome**

  For this organism, there is not much to say. It is isolated from sediment and produces different hydrolases.

By examining the first five organisms of 03sw in the contig list, I identified zero bacterium directly involved in methane production. Four are involved in the precursor steps of biogas production, and one remains unknown.

Binning

In this new section, percentage GC content will be plotted against coverage, following an approach equivalent to manual binning. This step is crucial as it allows us to explore and highlight bins, thereby identifying distinct organisms. We aim to determine whether these organisms potentially contribute to biogas formation through research on NCBI.

Procedure

A series of steps were undertaken to generate the graphs, for which I prepared several scripts.

1. **The creation of a table from spades results in contigs.fasta**
   The script I used is tb_spades.py and can be found as Appendix 1
2. **The creation of a table with accession code from the blast file, then joining with the previous table**
   The script I used is tb_spades.py and can be found as Appendix 2
3. **A simple plot**
   The script I used is tb_spades.py and can be found as Appendix 3
4. **A plot with colors and legend**
   The script I used is tb_spades.py and can be found as Appendix 4
5. **A plot with specific accession code only**
   The script I used is tb_spades.py and can be found in Appendix 5

Quantity of reads and contigs

Before binning my samples, I needed to calibrate the appropriate number of reads and contigs to obtain interpretable graphs. This may seem counterintuitive; however, upon observing figures 16, 17, and 18, it becomes evident that as the number of contigs considered increases, the number of data points inevitably increases. Beyond a certain threshold, the graphs become unreadable. This led me to opt for 200 contigs. This way, I can observe clusters while maintaining reasonable visibility.

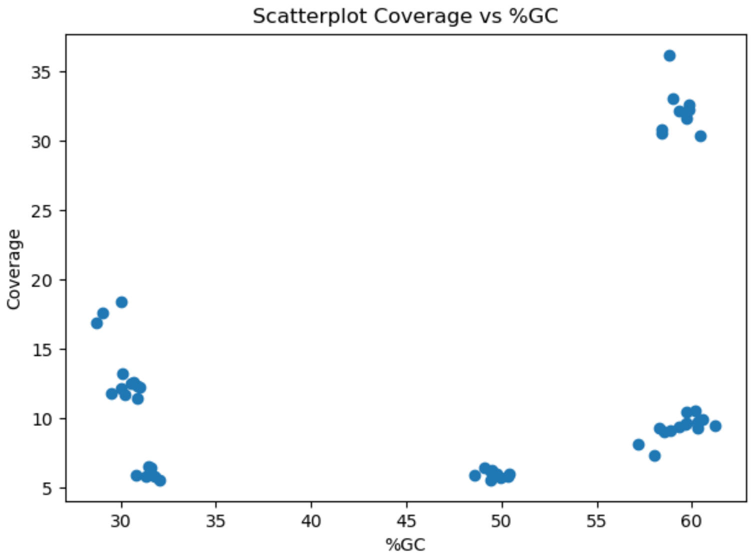


Figure 17 - Graph coverage as a function of %GC; 2M50

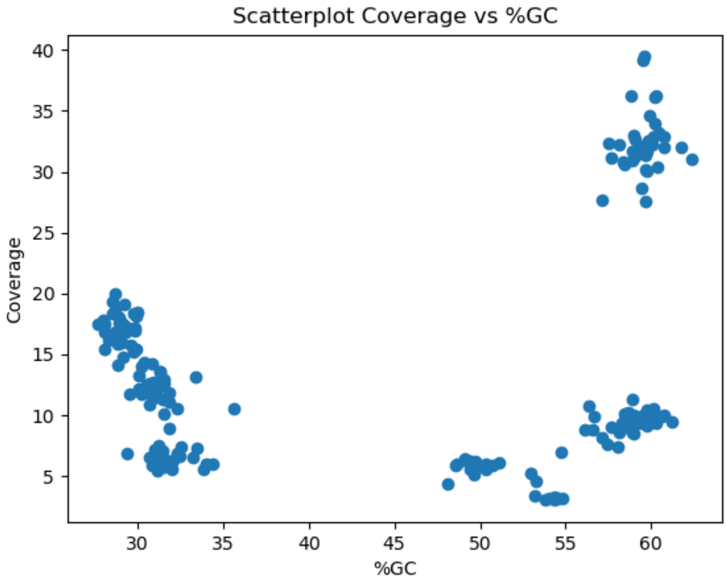


Figure 18 - Graph coverage as a function of %GC; 2M200

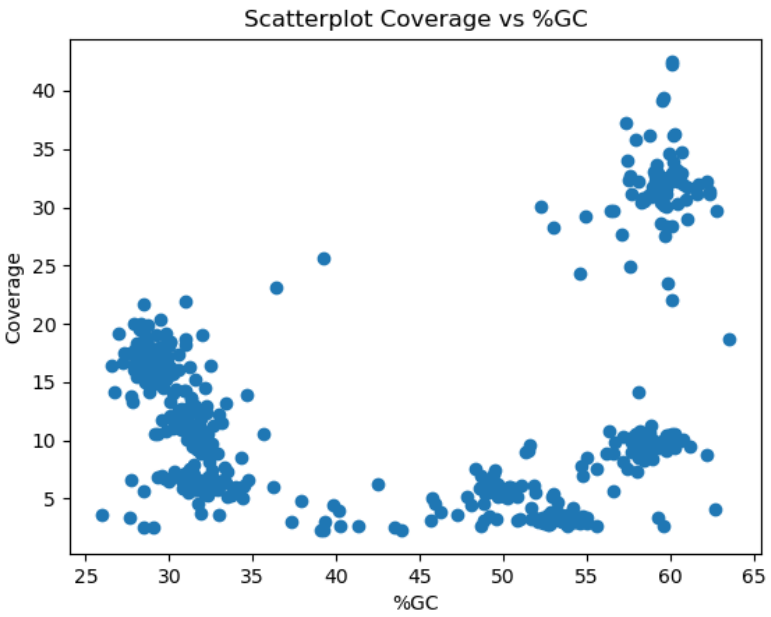


Figure 19 - Graph coverage as a function of %GC; 2M500

If we examine figures 17, 19, and 20, it becomes apparent that as the quantity of reads used for assembly increases, the coverage also increases, which is generally favorable. However, the readability of the graph is also impacted. Therefore, I opted for what appeared to be an optimal choice, namely 6 million reads.

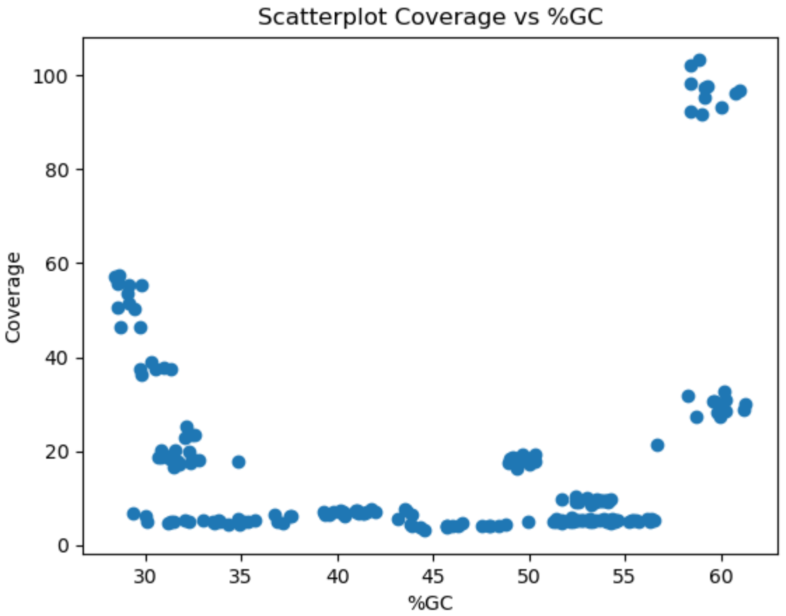


Figure 20 - Graph coverage as a function of %GC; 6M200

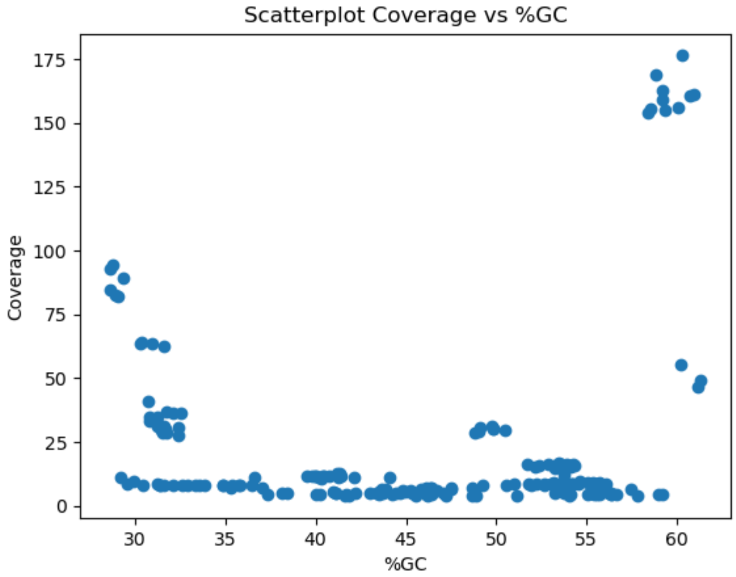


Figure 21 - Graph coverage as a function of %GC; 10M200

Hence, the binning of 02sw and 03sw was carried out using 6 million reads and considering 200 contigs.

02sw

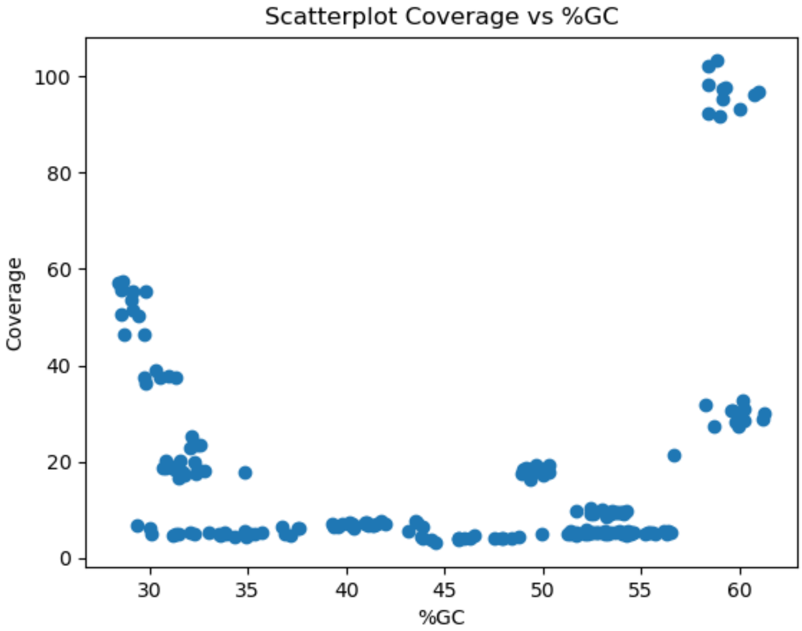


Figure 22 - 02sw graph coverage as a function of %GC ; first model

So, I initially constructed a preliminary graph, revealing the presence of around ten potential clusters. I subsequently created a graph associating accession codes with colors to confirm this. This graphical representation is depicted in Figure 22.

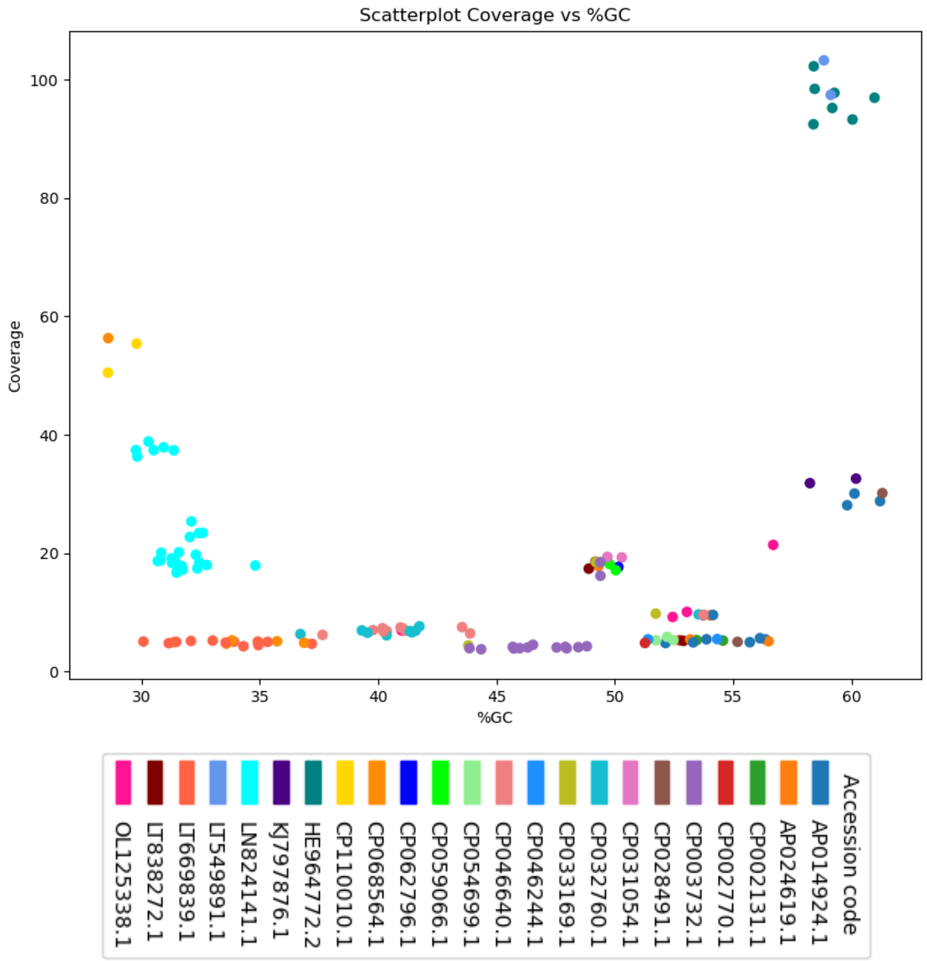


Figure 23- 02sw Graph coverage as a function of %GC colored by accession code (nt)

Immediately, it is evident that there are 23 organisms with more than 1 point on the graph (with others having been excluded beforehand). A distinct cluster is noticeable in the upper right corner (HE964772.2), which appears intriguing, and another light blue cluster in the lower left (LN824141.1). Research has been conducted on these organisms. Additionally, an investigation has been carried out on the organism in the lower middle, marked in purple (CP003732.1).

- **HE964772.2**

  It’s the accession code for Methanoculleus bourgensis MS2, an archea from the Methanoculleaceae family. It was isolated from a sewage sludge digester. Only by the name can we guess it is involved in biogas production. From Kegg, as shown in Figure 24, we can confirm that it is involved in methanogenesis and produce methane. Thanks to its enzyme, methyl coenzyme M reductase (MCR). We can see that the reaction catalyzed by MCR consists of reducing methyl-Coenzyme M to methane.

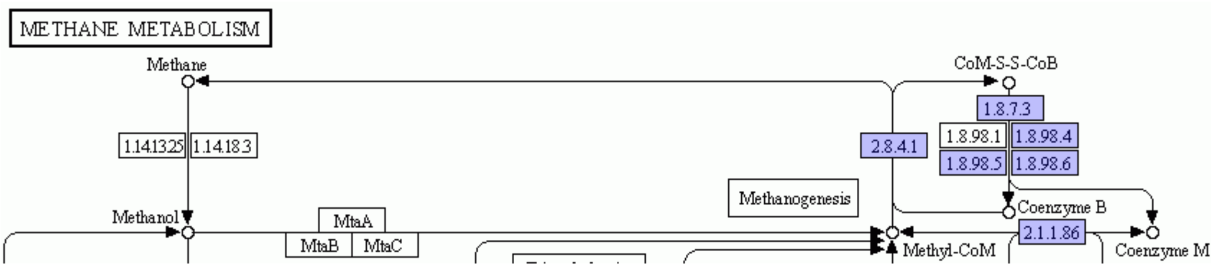


Figure 24 - Partial Methane metabolism of Methanoculleus bourgensis from KEGG

- **LN824141.1**
  It’s Defluviitoga tunisiensis, which doesn’t seem involved in direct biogas production. It’s an anaerobic thermophilic bacterium, but I didn’t find evidence of methane production in KEGG. On the other hand, it contains hydrolases, which can participate in precursor steps.
- **CP003732.1**
  The organism behind is Thermacetogenium phaeum. It’s a syntrophic acetate-oxidizing bacterium isolated from an anaerobic methanogenic reactor. It is a syntrophic acetate oxidizing bacterium, which oxidizes acetate in co-culture with a methanogen(Hattori et al., 2000), contributing to methanogenesis.

03sw

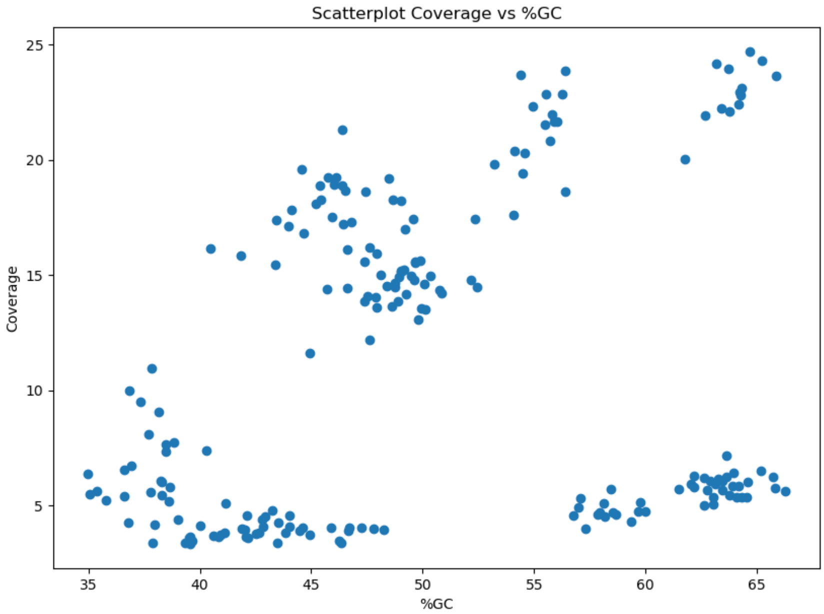


Figure 25 - 03sw graph coverage as a function of %GC ; first model

Like 02sw, I generated a basic plot for 03sw to identify potential clusters. However, the distribution appears more diffuse. It seems that there are six distinct groups. In Figure 24, there's a colored version of the plot based on accession codes, although the legend has been removed due to the presence of over fifty codes, which compromised the clarity of the image. As can be observed, the graph appears cluttered. Consequently, it seems essential to prune the data to enhance clarity.

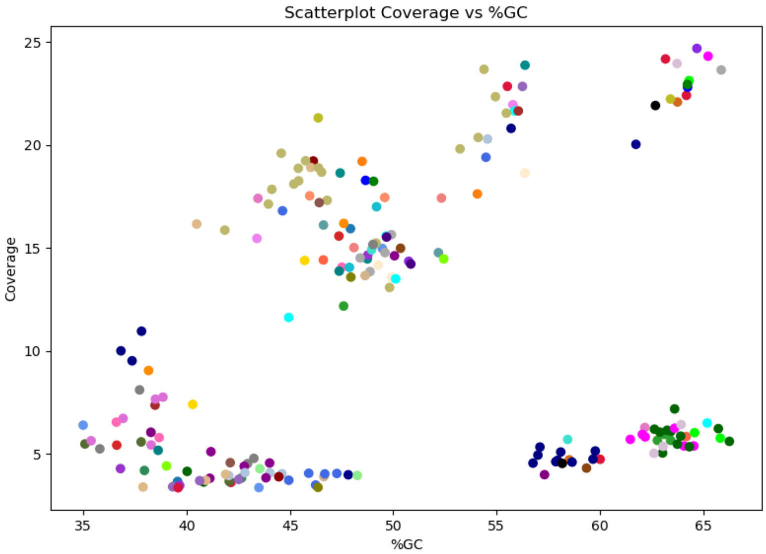


Figure 26 - 03sw graph coverage as a function of %GC ; colored by accession code (nt)

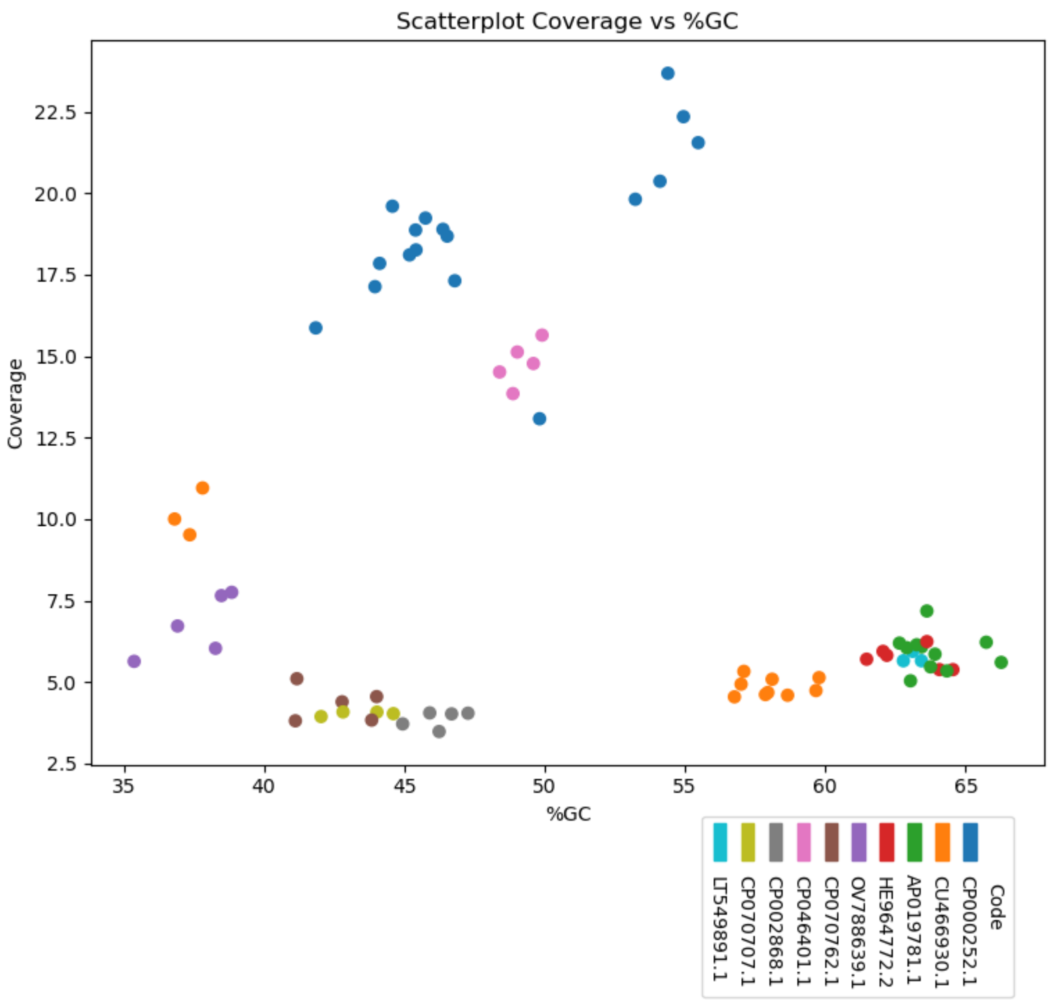


Figure 27 - 03sw Graph coverage as a function of %GC colored by accession code (nt)

- **CU466930.1**
  This organism is Candidatus Cloacimonas acidaminovorans. He was isolated from Evry mesophilic anaerobic digester (France). But it doesn't seem involved in the methanogenesis process.
- **CP000252.1**
  Syntrophus aciditrophicus bacterium can break down various substances such as fatty acids, benzoate, cyclohexane carboxylate, cyclohex-1-ene carboxylate, and crotonate. This degradation occurs when it interacts with hydrogen/formate-utilizing methanogenic or sulfate-reducing microorganisms in a cooperative environment (McInerney et al., 2007).
- **AP019781.1**
  The name of this organism is Methanoculleus chikugoensis. it's an archea isolated from a paddy field soil in Japan.. the first candidate I found for the 03sw able to directly produce methane (*Methanoculleus chikugoensis sp. nov., a novel methanogenic archaeon isolated from paddy field soil in Japan,*

and DNA-DNA hybridization among Methanoculleus species. | Microbiology Society, s. d.).

Mapping

As a result, I aligned the RNAseq data to the assembly generated by SPAdes, omitting the Lordec correction step as I decided to focus on the non-hybrid version. However, exploring this approach using Lordec and investigating potential differences would have been insightful.

Following this, I converted the SAM files into BAM format using Samtools.

Annotation

Furthermore, I should mention the annotation process using Prokka. The annotation was successful for sample 02sw but encountered an issue with sample "03sw" where the process terminated unexpectedly. Despite my efforts to troubleshoot the problem, I needed help identifying the root cause. The annotation with Prokka produces a GFF file, which would have been utilized for quantification with FeatureCounts.

Quantification

Unfortunately, due to time limitations, I could not proceed with the quantification process using FeatureCounts. This step, while not completed, holds the potential to offer valuable additional insights into the analysis of our RNAseq data. This quantification provided a deeper understanding of the expression profiles and potentially revealed new layers of information regarding the biological processes at play.

Interpretations

At the end of this study, some discoveries and observations have emerged from the 02sw and 03sw samples. Indeed, organisms participating in the biogas production process have been identified. Among these, I have highlighted three distinct organisms associated with biogas formation in the 02sw sample and another organism within the 03sw sample.

Moreover, the study has shed light on optimization parameters during the execution of SPADES. By exploring various combinations of threads, methods, and read counts, we have determined configurations that optimize the processing time of the assembly procedure. This nuanced understanding of parameters not only accelerates SPADES execution but also enhances the quality of obtained results.

An alternative approach to traditional binning has also been explored, where graphs depicting GC percentage in relation to coverage were employed to identify organism clusters. While this method might seem manual, it provides a rapid overview of sample composition, bypassing the complexity and time required to master more sophisticated tools and pipelines. This approach proves particularly valuable for initial data exploration before delving into more in-depth analyses.

In conclusion, this study has successfully pinpointed potentially involved organisms in biogas production, optimized SPADES execution parameters, and explored an alternative binning method.

Conclusion

As this study draws to a close, it is evident that it has yielded significant insights and revelations within the realms of my metagenomics samples and functional analysis. The identification of organisms intricately connected to the biogas production process is a good achievement I was far to get. I have highlighted three organisms associated with biogas formation within the 02sw sample and another within the 03sw sample. It seems nothing, but it's already something for further investigation. Some others were involved in the biogas process by synchronisms, and others are interested in precursor steps of methanogenesis, such as hydrolysis and acetogenesis.

Moreover, I've dived into optimizing SPADES execution parameters to optimize the execution time by experimenting with various combinations of threads, methods, and read counts. This understanding accelerates SPADES execution.

I glanced at an alternative approach to the conventional binning process, employing graphs depicting the GC percentage in relation to coverage to delineate organism clusters. While it might appear a manual procedure, it provides a rapid overview of sample composition, bypassing the intricacies and time investment required to master more sophisticated tools and pipelines. This approach is valuable as an initial foray into data exploration before delving into more intricate analyses.

In conclusion, this study has unveiled organisms with potential involvement in biogas production, refined SPADES execution parameters, and delved into an alternative binning method. It has also underscored the rewarding nature of diverse explorations and underlined the valuable lesson that what may seem simple, such as a pipeline, can be intricate. Starting analyses with essential tools to navigate effectively and efficiently toward the intended goals has proven invaluable.

Reflecting on this journey, I am reminded that unexpected turns often mark the path to understanding, and pursuing knowledge itself is a remarkable adventure. This study represents a little stepping stone in unraveling the complexities of biogas production and serves as a testament to the power of curiosity, adaptability, and perseverance through difficulties

perspective

In terms of perspectives, one particularly catches my attention: the exploration of hybrid assembly pathways. Regrettably, I could not pursue this route, which would have allowed me to compare the quality derived from hybrid assembly against that of more straightforward approaches. Additionally, time constraints prevented me from delving into FeatureCounts, a quantification tool I was eager to explore using real data thoroughly. Furthermore, the realm of differential analysis still needs to be expanded due to limitations, although both manual exploration and available tools hold promise. Finally, the broader journey of processing my data invites further investigation into the subsequent stages, promising a wealth of insights yet to be unearthed.

Bibliography

Ben Ali Gam, Z., Oueslati, R., Abdelkafi, S., Casalot, L., Tholozan, J. L., & Labat, M. (2009). Desulfovibrio tunisiensis sp. Nov., a novel weakly halotolerant, sulfate-reducing bacterium isolated from exhaust water of a Tunisian oil refinery. *International Journal of Systematic and Evolutionary Microbiology*, *59*(5), 1059-1063. https://doi.org/10.1099/ijs.0.000943-0

Biogas upgrading. (s. d.). Consulté 12 mai 2022, à l'adresse https://task37.ieabioenergy.com/files/daten-redaktion/download/publications/Workshops/7/06%20biogasupgrading.pdf

Chang, A., Jeske, L., Ulbrich, S., Hofmann, J., Koblitz, J., Schomburg, I., Neumann-Schaal, M., Jahn, D., & Schomburg, D. (2021). BRENDA, the ELIXIR core data resource in 2021 : New developments and updates. *Nucleic Acids Research*, *49*(D1), D498-D508. https://doi.org/10.1093/nar/gkaa1025

Du, Z.-J., Wang, Y., Dunlap, C., Rooney, A. P., & Chen, G.-J. (2014). Draconibacterium orientale gen. Nov., sp. Nov., isolated from two distinct marine environments, and proposal of Draconibacteriaceae fam. Nov. *International Journal of Systematic and Evolutionary Microbiology*, *64*(Pt 5), 1690-1696. https://doi.org/10.1099/ijs.0.056812-0

Genome Assembly—An overview | ScienceDirect Topics. (s. d.). Consulté 6 mai 2022, à l'adresse https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/genome-assembly

Grohmann, A., Fehrmann, S., Vainshtein, Y., Haag, N. L., Wiese, F., Stevens, P., Naegele, H.-J., Oechsner, H., Hartsch, T., Sohn, K., & Grumaz, C. (2018). Microbiome dynamics and adaptation of expression signatures during methane production failure and process recovery. *Bioresource Technology*, *247*, 347-356. https://doi.org/10.1016/j.biortech.2017.08.214

Gurevich, A., Saveliev, V., Vyahhi, N., & Tesler, G. (2013). QUAST : Quality assessment tool for genome assemblies. *Bioinformatics (Oxford, England)*, *29*(8), 1072-1075. https://doi.org/10.1093/bioinformatics/btt086

Hattori, S., Kamagata, Y., Hanada, S., & Shoun, H. (2000). Thermacetogenium phaeum gen. Nov., sp. Nov., a strictly anaerobic, thermophilic, syntrophic acetate-oxidizing bacterium. *International Journal of Systematic and Evolutionary Microbiology*, *50*(4), 1601-1609. https://doi.org/10.1099/00207713-50-4-1601

Heng, S., Sutheeworapong, S., Prommeenate, P., Cheevadhanarak, S., Kosugi, A., Pason, P., Waeonukul, R., Ratanakhanokchai, K., & Tachaapaikoon, C. (2019). Complete Genome Sequence of Halocella sp. Strain SP3-1, an Extremely Halophilic, Glycoside Hydrolase- and Bacteriocin-Producing Bacterium Isolated from a Salt Evaporation Pond. *Microbiology Resource Announcements*, *8*(7), 10.1128/mra.01696-18. https://doi.org/10.1128/mra.01696-18

Hugenholtz, P., & Tyson, G. W. (2008). Metagenomics. *Nature*, *455*(7212), Article 7212. https://doi.org/10.1038/455481a

Liao, Y., Smyth, G. K., & Shi, W. (2014). featureCounts : An efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics (Oxford, England)*, *30*(7), 923-930. https://doi.org/10.1093/bioinformatics/btt656

McInerney, M. J., Rohlin, L., Mouttaki, H., Kim, U., Krupp, R. S., Rios-Hernandez, L., Sieber, J., Struchtemeyer, C. G., Bhattacharyya, A., Campbell, J. W., & Gunsalus, R. P. (2007). The genome of Syntrophus aciditrophicus : Life at the thermodynamic limit of microbial growth. *Proceedings of* *the National Academy of Sciences of the United States of America*, *104*(18), 7600-7605. https://doi.org/10.1073/pnas.0610456104

Methanoculleus chikugoensis sp. Nov., a novel methanogenic archaeon isolated from paddy field soil in Japan, and DNA-DNA hybridization among Methanoculleus species. | Microbiology Society. (s. d.). Consulté 23 août 2023, à l'adresse https://www.microbiologyresearch.org/content/journal/ijsem/10.1099/00207713-51-5-1663

Muzenda, E. (2014). *Bio-methane Generation from Organic Waste : A Review*. 6.

Ngo, T., Ball, A., & Shahsavari, E. (2021). The Current Status, Potential Benefits and Future Prospects of the Australian Biogas Sector. *Journal of Sustainable Bioenergy Systems*, *11*, 14-32. https://doi.org/10.4236/jsbs.2021.111002

Pérez-Cobas, A. E., Gomez-Valero, L., & Buchrieser, C. 2020. (s. d.). Metagenomic approaches in microbial ecology : An update on whole-genome and marker gene sequencing analyses. *Microbial Genomics*, *6*(8), e000409. https://doi.org/10.1099/mgen.0.000409

Quince, C., Walker, A. W., Simpson, J. T., Loman, N. J., & Segata, N. (2017). Shotgun metagenomics, from sampling to analysis. *Nature Biotechnology*, *35*(9), Article 9. https://doi.org/10.1038/nbt.3935

Seemann, T. (2014). Prokka : Rapid prokaryotic genome annotation. *Bioinformatics (Oxford, England)*, *30*(14), 2068-2069. https://doi.org/10.1093/bioinformatics/btu153

Sohn, jang-il, & Nam, J.-W. (2016). The present and future of de novo whole-genome assembly. *Briefings in Bioinformatics*, *19*, bbw096. https://doi.org/10.1093/bib/bbw096

Wick, R. R., Judd, L. M., Gorrie, C. L., & Holt, K. E. (2017). Unicycler : Resolving bacterial genome assemblies from short and long sequencing reads. *PLoS Computational Biology*, *13*(6), e1005595. https://doi.org/10.1371/journal.pcbi.1005595

Yamada, T., Hamada, M., Kurobe, M., Narihiro, T., Tsuji, H., & Daimon, H. (2021). Complete Genome Sequence of Gelria sp. Strain Kuro-4, a Thermophilic Anaerobe Isolated from a Thermophilic Anaerobic Digestion Reactor Treating Poly(L-Lactic Acid). *Microbiology Resource Announcements*, *10*(33), e0054421. https://doi.org/10.1128/MRA.00544-21

Zhu, Q., Dupont, C. L., Jones, M. B., Pham, K. M., Jiang, Z.-D., DuPont, H. L., & Highlander, S. K. (2018). Visualization-assisted binning of metagenome assemblies reveals potential new pathogenic profiles in idiopathic travelers' diarrhea. *Microbiome*, *6*(1), 201. https://doi.org/10.1186/s40168-018-0579-0

Glossary

DNA – Deoxyribonucleic acid

RNA – Ribonucleic acid

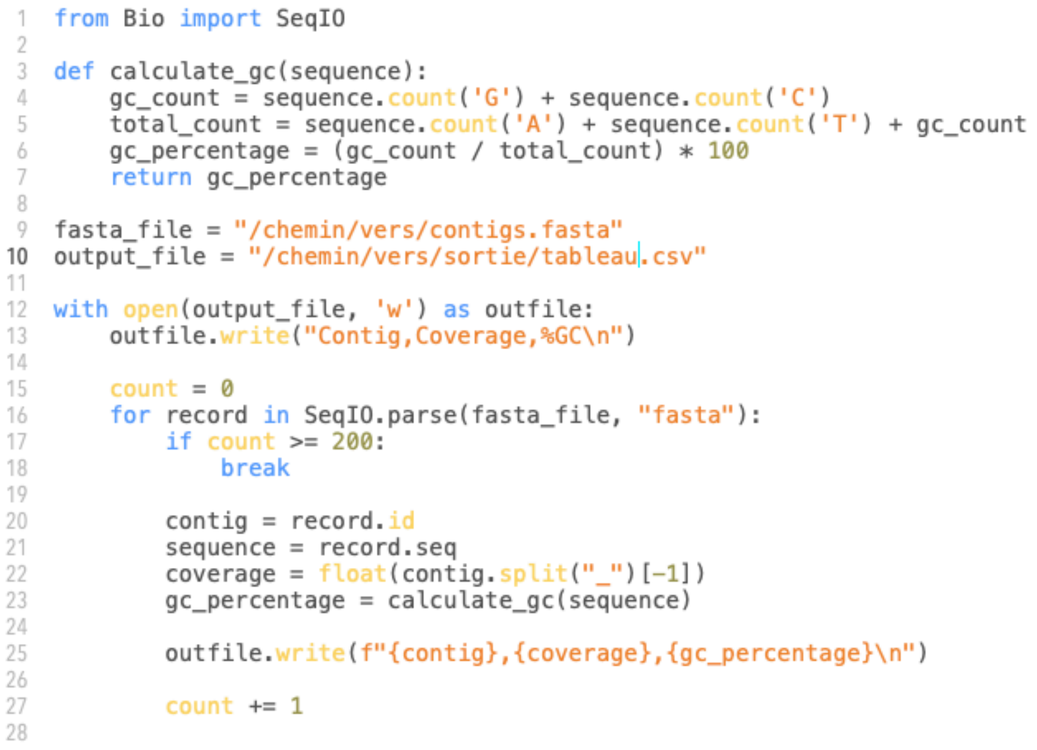
ONT – Oxford Nanopore Technology

GUI - graphical user interface

Appendix table

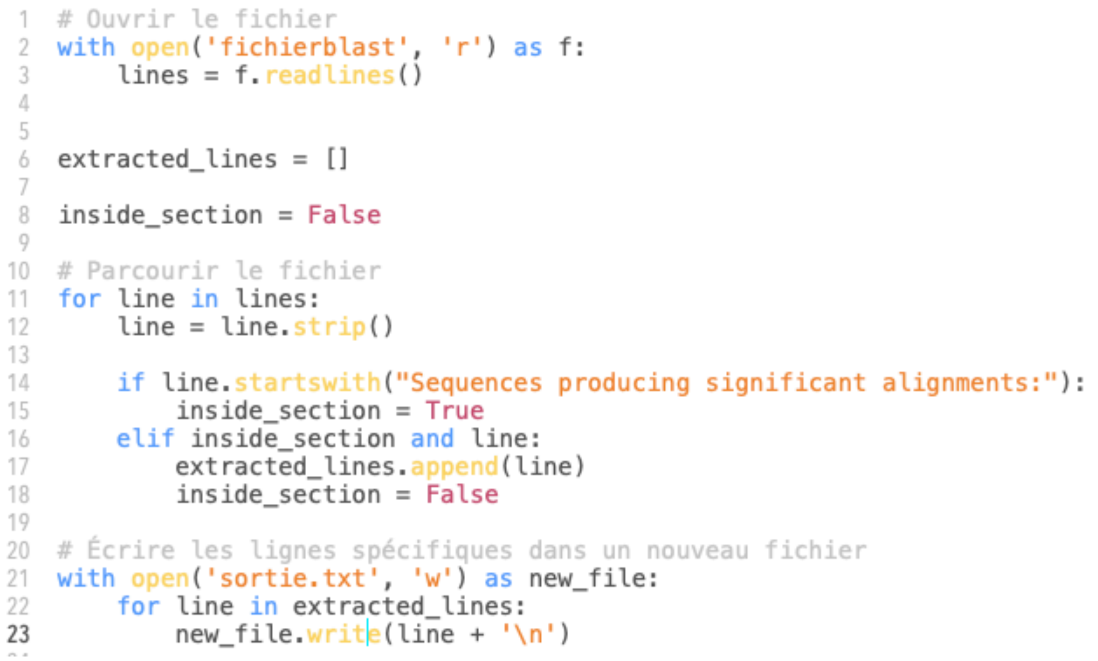
Appendix

Appendix 1 - tb_spades.py



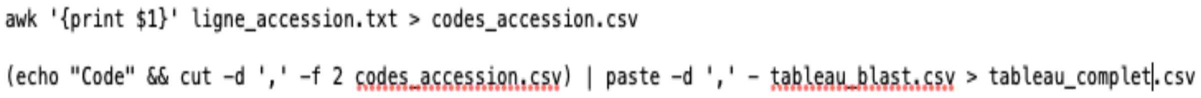
```python
from Bio import SeqIO

def calculate_gc(sequence):
    gc_count = sequence.count('G') + sequence.count('C')
    total_count = sequence.count('A') + sequence.count('T') + gc_count
    gc_percentage = (gc_count / total_count) * 100
    return gc_percentage

fasta_file = "/chemin/vers/contigs.fasta"
output_file = "/chemin/vers/sortie/tableau.csv"

with open(output_file, 'w') as outfile:
    outfile.write("Contig,Coverage,%GC\n")

    count = 0
    for record in SeqIO.parse(fasta_file, "fasta"):
        if count >= 200:
            break

        contig = record.id
        sequence = record.seq
        coverage = float(contig.split("_")[-1])
        gc_percentage = calculate_gc(sequence)

        outfile.write(f"{contig},{coverage},{gc_percentage}\n")

        count += 1

```

Appendix 2 - ligne_acc_blast.py



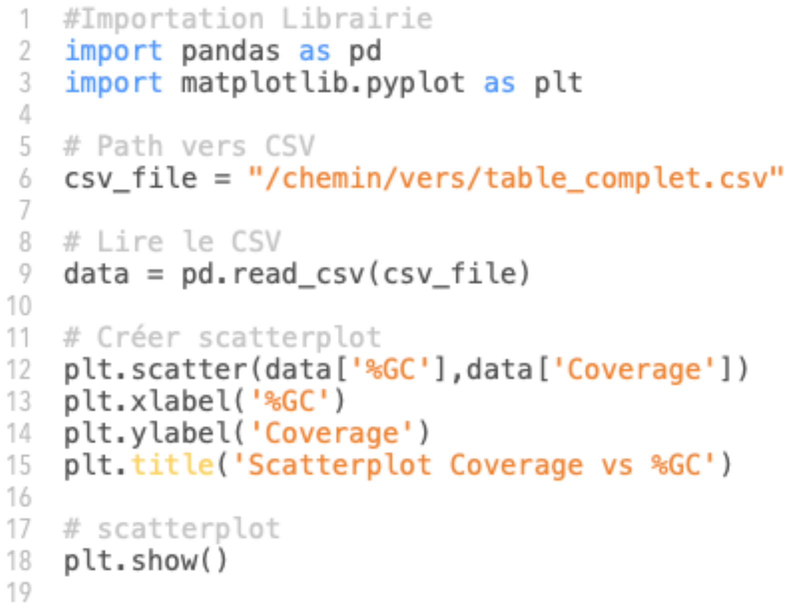
```python
# Ouvrir le fichier
with open('fichierblast', 'r') as f:
    lines = f.readlines()


extracted_lines = []

inside_section = False

# Parcourir le fichier
for line in lines:
    line = line.strip()

    if line.startswith("Sequences producing significant alignments:"):
        inside_section = True
    elif inside_section and line:
        extracted_lines.append(line)
        inside_section = False

# Écrire les lignes spécifiques dans un nouveau fichier
with open('sortie.txt', 'w') as new_file:
    for line in extracted_lines:
        new_file.write(line + '\n')
```

- Script bash pour faire une table des codes d’accession puis le rajouter à la table crée par Tb_blast.py



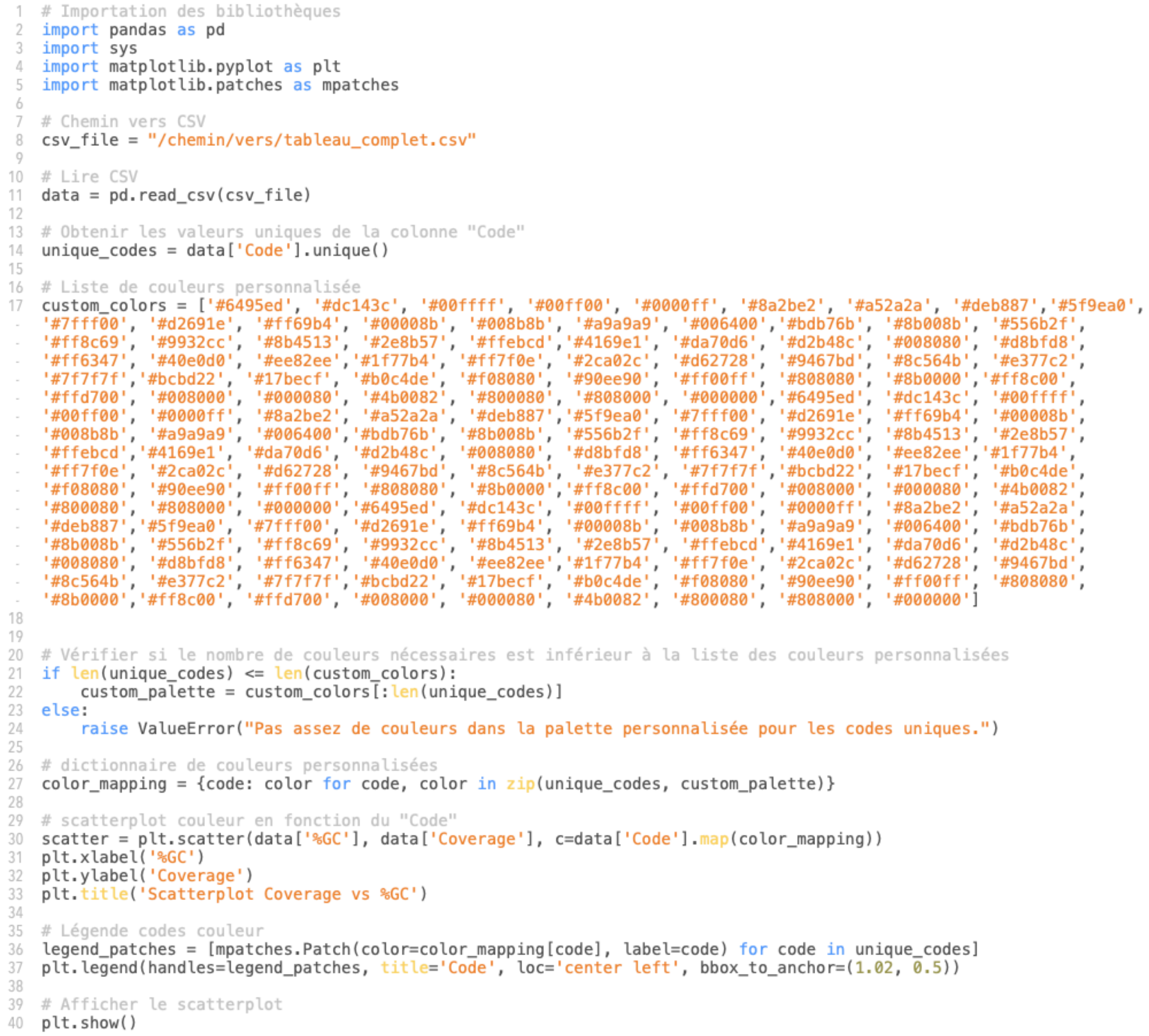
```
awk '{print $1}' ligne_accession.txt > codes_accession.csv

(echo "Code" && cut -d ',' -f 2 codes_accession.csv) | paste -d ',' - tableau_blast.csv > tableau_complet.csv
```

Appendix 3 - Simple_plot.py



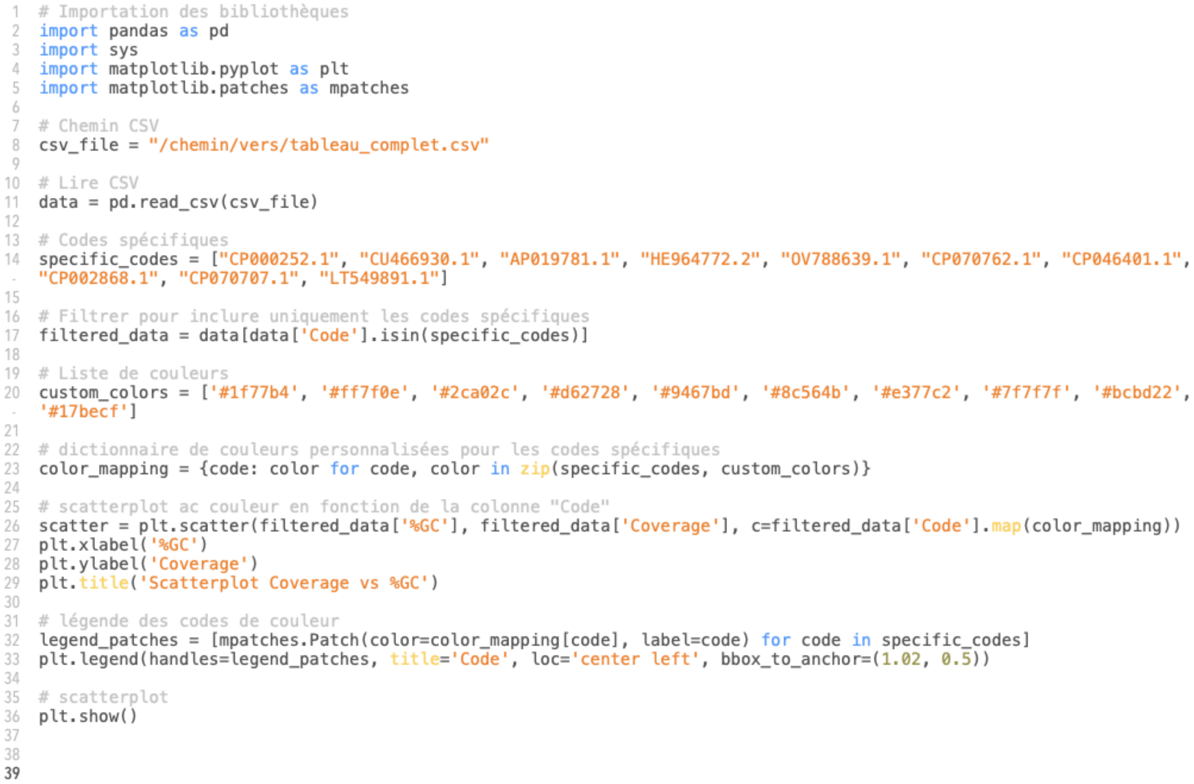
```python
#Importation Librairie
import pandas as pd
import matplotlib.pyplot as plt

# Path vers CSV
csv_file = "/chemin/vers/table_complet.csv"

# Lire le CSV
data = pd.read_csv(csv_file)

# Créer scatterplot
plt.scatter(data['%GC'],data['Coverage'])
plt.xlabel('%GC')
plt.ylabel('Coverage')
plt.title('Scatterplot Coverage vs %GC')

# scatterplot
plt.show()

```

Appendix 4 - Colored_leg_plot.py



```
# Importation des bibliothèques
import pandas as pd
import sys
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches

# Chemin vers CSV
csv_file = "/chemin/vers/tableau_complet.csv"

# Lire CSV
data = pd.read_csv(csv_file)

# Obtenir les valeurs uniques de la colonne "Code"
unique_codes = data['Code'].unique()

# Liste de couleurs personnalisée
custom_colors = ['#6495ed', '#dc143c', '#00ffff', '#00ff00', '#0000ff', '#8a2be2', '#a52a2a', '#deb887','#5f9ea0',
'#7fff00', '#d2691e', '#ff69b4', '#00008b', '#008b8b', '#a9a9a9', '#006400','#bdb76b', '#8b008b', '#556b2f',
'#ff8c69', '#9932cc', '#8b4513', '#2e8b57', '#ffebcd','#4169e1', '#da70d6', '#d2b48c', '#008080', '#d8bfd8',
'#ff6347', '#40e0d0', '#ee82ee','#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2',
'#7f7f7f','#bcbd22', '#17becf', '#b0c4de', '#f08080', '#90ee90', '#ff00ff', '#808080', '#8b0000','#ff8c00',
'#ffd700', '#008000', '#000080', '#4b0082', '#800080', '#808000', '#000000','#6495ed', '#dc143c', '#00ffff',
'#00ff00', '#0000ff', '#8a2be2', '#a52a2a', '#deb887','#5f9ea0', '#7fff00', '#d2691e', '#ff69b4', '#00008b',
'#008b8b', '#a9a9a9', '#006400','#bdb76b', '#8b008b', '#556b2f', '#ff8c69', '#9932cc', '#8b4513', '#2e8b57',
'#ffebcd','#4169e1', '#da70d6', '#d2b48c', '#008080', '#d8bfd8', '#ff6347', '#40e0d0', '#ee82ee','#1f77b4',
'#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2', '#7f7f7f','#bcbd22', '#17becf', '#b0c4de',
'#f08080', '#90ee90', '#ff00ff', '#808080', '#8b0000','#ff8c00', '#ffd700', '#008000', '#000080', '#4b0082',
'#800080', '#808000', '#000000','#6495ed', '#dc143c', '#00ffff', '#00ff00', '#0000ff', '#8a2be2', '#a52a2a',
'#deb887','#5f9ea0', '#7fff00', '#d2691e', '#ff69b4', '#00008b', '#008b8b', '#a9a9a9', '#006400', '#bdb76b',
'#8b008b', '#556b2f', '#ff8c69', '#9932cc', '#8b4513', '#2e8b57', '#ffebcd','#4169e1', '#da70d6', '#d2b48c',
'#008080', '#d8bfd8', '#ff6347', '#40e0d0', '#ee82ee','#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
'#8c564b', '#e377c2', '#7f7f7f','#bcbd22', '#17becf', '#b0c4de', '#f08080', '#90ee90', '#ff00ff', '#808080',
'#8b0000','#ff8c00', '#ffd700', '#008000', '#000080', '#4b0082', '#800080', '#808000', '#000000']


# Vérifier si le nombre de couleurs nécessaires est inférieur à la liste des couleurs personnalisées
if len(unique_codes) <= len(custom_colors):
    custom_palette = custom_colors[:len(unique_codes)]
else:
    raise ValueError("Pas assez de couleurs dans la palette personnalisée pour les codes uniques.")

# dictionnaire de couleurs personnalisées
color_mapping = {code: color for code, color in zip(unique_codes, custom_palette)}

# scatterplot couleur en fonction du "Code"
scatter = plt.scatter(data['%GC'], data['Coverage'], c=data['Code'].map(color_mapping))
plt.xlabel('%GC')
plt.ylabel('Coverage')
plt.title('Scatterplot Coverage vs %GC')

# Légende codes couleur
legend_patches = [mpatches.Patch(color=color_mapping[code], label=code) for code in unique_codes]
plt.legend(handles=legend_patches, title='Code', loc='center left', bbox_to_anchor=(1.02, 0.5))

# Afficher le scatterplot
plt.show()
```

Appendix 5 - Specific_code_plot_cl.py



```python
# Importation des bibliothèques
import pandas as pd
import sys
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches

# Chemin CSV
csv_file = "/chemin/vers/tableau_complet.csv"

# Lire CSV
data = pd.read_csv(csv_file)

# Codes spécifiques
specific_codes = ["CP000252.1", "CU466930.1", "AP019781.1", "HE964772.2", "OV788639.1", "CP070762.1", "CP046401.1", "CP002868.1", "CP070707.1", "LT549891.1"]

# Filtrer pour inclure uniquement les codes spécifiques
filtered_data = data[data['Code'].isin(specific_codes)]

# Liste de couleurs
custom_colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf']

# dictionnaire de couleurs personnalisées pour les codes spécifiques
color_mapping = {code: color for code, color in zip(specific_codes, custom_colors)}

# scatterplot ac couleur en fonction de la colonne "Code"
scatter = plt.scatter(filtered_data['%GC'], filtered_data['Coverage'], c=filtered_data['Code'].map(color_mapping))
plt.xlabel('%GC')
plt.ylabel('Coverage')
plt.title('Scatterplot Coverage vs %GC')

# légende des codes de couleur
legend_patches = [mpatches.Patch(color=color_mapping[code], label=code) for code in specific_codes]
plt.legend(handles=legend_patches, title='Code', loc='center left', bbox_to_anchor=(1.02, 0.5))

# scatterplot
plt.show()
```