University of Rouen Normandy - Science and Technology faculty
First year Master's degree in bioinformatics – BIMS
2023-2024

Internship dissertation

Development of a metagenome annotation workflow with the graph-based annotation tool, ggCaller

Author :
MAOULOUD Lale

EMBL's European Bioinformatics Institute
Pathogen informatics and modelling group

Supervisors:
Dr. Samuel Horsfield, *Post-doctoral researcher*
Dr. John Lees *Group Leader and Co-chair of Infection Biology Transversal Theme*

AMBASSADE DE FRANCE AU ROYAUME-UNI
Liberté Égalité Fraternité

Erasmus+

EMBL-EBI

University of Rouen Normandy - Science and Technology faculty
First year Master's degree in bioinformatics – BIMS
2023-2024

Internship dissertation

Development of a metagenome annotation workflow with the graph-based annotation tool, ggCaller

Author :
MAOULOUD Lale

EMBL's European Bioinformatics Institute
Pathogen informatics and modelling group

Supervisors:
Dr. Samuel Horsfield, *Post-doctoral researcher*
Dr. John Lees *Group Leader and Co-chair of Infection Biology Transversal Theme*

AMBASSADE DE FRANCE AU ROYAUME-UNI
Liberté Égalité Fraternité
Erasmus+
EMBL-EBI

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my professor, Mrs. Hélène Dauchel from the University of Rouen Normandy, for her invaluable guidance and support during my internship search. Her insightful advice and encouragement played a pivotal role in targeting suitable opportunities, ultimately leading to the discovery of an internship that perfectly aligned with my aspirations.

I am deeply indebted to my supervisors, John Lees and Samuel Horsfield, for providing me with an enriching and supportive environment to undertake this internship. Their mentorship and expertise have not only broadened my understanding across various domains but have also instilled in me the importance of patience, effective communication, and the acceptance of failure as an inherent part of the research journey. I extend my heartfelt gratitude to Samuel Horsfield for his unwavering attention and invaluable guidance throughout the entirety of the internship period.

Furthermore, I extend my sincere appreciation to the Embassy of France in London for their generous financial support, which enabled me to maximize the benefits of this internship experience in Cambridge. I am hopeful that this collaborative partnership will continue to empower and inspire many other aspiring researchers in their pursuit of knowledge.

To my parents, you should know that your support and encouragement was worth more than I can express on paper. This accomplishment would not have been possible without them. Thank you.

Contents

List of Figures

List of Acronyms

- **EBI**: European Bioinformatics Institute
- **EMBL**: European Molecular Biology Laboratory
- **ETR**: Extra Tree Regressor
- **HPC**: High Performance Compute Cluster
- **IDE**: Integrated Development Environment
- **KNN**: K Nearest Neighbour
- **KNR**: K Neighbours Regressor
- **MAG**: Metagenome Assembled Genome
- **NGS**: Next-Generation Sequencing
- **ORF**: Open Reading Frame
- **SLURM**: Simple Linux Utility for Resource Management
- **VSC**: Visual Studio Code
- **ML**: Machine Learning
- **SI**: Simple Imputer
- **BR**: Bayesian Ridge
- **CF**: Collaborative Filtering
- **DBG**: De Bruijn Graph
- **DFS**: Depth First Search
- **DNA**: Deoxyribonucleic Acid
- **DTR**: Decision Tree Regressor
- **BacPop**: Bacterial Population Genetics Group, EMBL-EBI

1. Introduction

1.1 The Organisation

This internship takes place at the European Bioinformatics Institute (EBI), situated in the Wellcome Genome Campus, Hinxton, United Kingdom. Founded in 1994, the EBI is part of the European Molecular Biology Laboratory (EMBL). The EMBL is an intergovernmental organisation founded in 1974 by Léo Szilàrd, James Watson, and John Kendrew (Thakur et al., 2022), and is currently supported by 28 different states. The research conducted at EMBL is site-specific, with six different sites.

The aim of the EBI 1.1 is to provide freely available data, bioinformatics services, and training. This involves maintaining databases such as Expression Atlas, or tools like Clustal Omega. Collaborations are strongly encouraged, facilitated by the presence of an Elixir hub at the EBI, and the Wellcome Sanger Institute, also situated in the Wellcome Genome Campus (Figure 1.1). Collaborations extend globally, exemplified by EBI's partnership with DeepMind to create and update the AlphaFold Database (Varadi et al., 2021). The Pathogen Informatics and Modeling Group 15 at EMBL-EBI, in collaboration with the Robert Finn group 16, is at the forefront of Metagenomics Analysis. The goal of this internship is to enhance the annotation methodology for Metagenome-Assembled Genomes (MAGs) by leveraging the capabilities of the state-of-the-art graph-based tool, ggCaller 10.

Figure 1.1: Photograph of EMBL's European Bioinformatics Institute, one of the six sites of the European Molecular Biology Laboratory (EMBL). With 28 member states and thousands of scientists and engineers working together, EMBL is a powerhouse of biological expertise.

1.2 Scientific context

1.2.1 Pangenomics

From the inception of genome research, the scientific community has traditionally depended on a singular 'reference' genome for each species to conduct a broad spectrum of genetic analyses, including

the examination of genetic variations both within and between species (Eizenga et al., 2020). However, with the significant reduction in sequencing costs, the sequencing of thousands of new genomes has illuminated the limitations of relying solely on a single reference genome for comprehensive genetic studies. Recognizing these shortcomings has led to the development of the concept of a pan-genome (Tettelin et al., 2005a). Conceptually, the pan-genome is defined as the entirety of genes or sequences present within a population or clade (e.g., species, species complexes) (Sherman & Salzberg, 2020). In other words, it represents the union of sequences from all genomes within the considered group. it can be represented using different data structures, and the most commonly used one is De Bruijn graphs (see Figure 1.2). These graphs help in identifying similarities and differences between gene sets by representing overlapping sequences, thereby facilitating the detection of genetic variations and the structural organization of genomes (Compeau et al., 2011).

The concept of a pangenome was first used in 2005 in bacteria (Tettelin et al., 2005b) but can now be applied to all species in the living domain (Golicz et al., 2020). Pangenomics seeks to capture the full genetic diversity within a population. This comprehensive approach identifies core genes, which are conserved across different species, core genes enable the reconstruction of phylogenetic trees (Parks et al., 2018) to understand the evolutionary connections and origins of various species (Baumdicker et al., 2012). On the other hand, accessory genes, which vary among different strains or species, are instrumental in genotype-phenotype association studies (Brynildsrud et al., 2016). These genes help identify traits that may confer ecological advantages or adaptations, allowing researchers to link specific genetic variations to observable phenotypic differences (Golicz et al., 2016).

The study of pangenomics has become especially relevant in microbial genomics, where the vast diversity between even closely related strains can significantly affect their behavior and interactions (Computational Pan-Genomics Consortium, 2018). Additionally, pangenomics is applied in agriculture to breed crops with desirable traits by understanding gene variations that confer advantages like drought resistance or increased yield (Golicz et al., 2016). In human genetics (Computational Pan-Genomics Consortium, 2018), pangenomics helps to understand the genetic underpinnings of diseases better by highlighting how variations from the reference genome can influence disease susceptibility and treatment responses (Abondio et al., 2023).

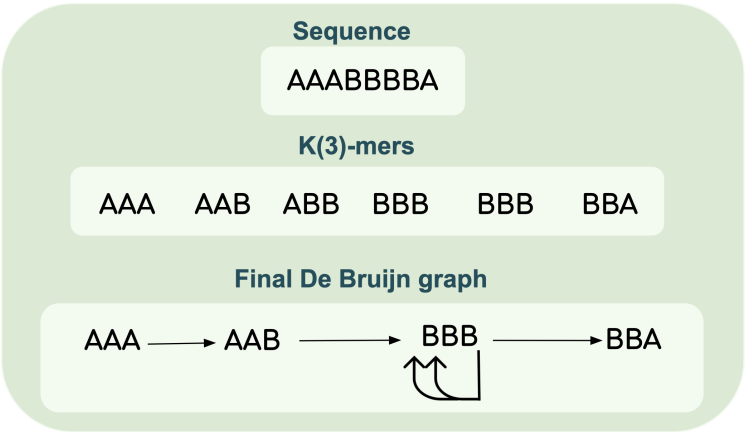

Figure 1.2: Diagram showing the construction of a De Bruijn graph from a sequence of DNA. Nodes: Represent k-mers (subsequences of length k) and edges Indicate overlaps of length k-1 between k-mers.

This visual representation helps in understanding how sequences can be reconstructed and analyzed using De Bruijn graphs, which are essential in various computational biology applications, including genome assembly. De Bruijn graphs simplify the complexity of assembling short read sequences by breaking them into k-mers, which are sequences of length k where sequential k-mers have an overlap of k-1 nucleotides. This allows for more efficient and accurate reconstruction of the original genome. The first DBG assembler, Velvet (Kleftogiannis et al., 2013), pioneered this approach, significantly improving the accuracy and speed of genome assemblies.

Furthermore, De Bruijn graphs play a crucial role in pangenome representation. Tools like Pantools and ggCaller utilize De Bruijn graphs to manage and analyze the vast amounts of genomic data efficiently. Pantools focuses on the representation, storage, and exploration of pan-genomic data, switching from reference-centric to comprehensive pangenomic approaches. It enables analyses at various levels, such as nucleotide, gene, and genome structure, facilitating the identification of genetic variations across multiple genomes (Jonkheer et al., 2022). ggCaller, on the other hand, is designed for pangenome annotation and clustering.

1.2.2 Metagenomics

The term "metagenomics" was first used by Jo Handelsman and others in the University of Wisconsin Department of Plant Pathology, and first appeared in publication in 1998 (Handelsman et al., 1998).The term metagenome represented the idea that a collection of genes sequenced from the environment could be analyzed in way analogous to the study of a single genome. The exploding interest in environmental genetics has resulted in the broader use of metagenomics to describe any sequencing of genetic material from environmental (i.e. uncultured) samples. In 2005 researchers at the University of California, Berkeley, Kevin Chen and Lior Pachter defined metagenomics as "the application of modern genomics techniques to the study of communities of microbial organisms directly in their natural environments, bypassing the need for isolation and lab cultivation of individual species" (Chen & Pachter, 2005).

Modern genomic techniques, especially next-generation sequencing (NGS), have revolutionized metagenomics by enabling the analysis of entire microbial communities directly from environmental samples without the need for culturing individual species. Key techniques include whole-genome shotgun sequencing, which involves sequencing all the DNA present in a sample, and targeted amplicon sequencing, which focuses on sequencing specific regions of the genome, such as the 16S rRNA gene in bacteria, to identify and classify microorganisms within the community.

This field has transformed our understanding of microbial ecosystems, uncovering vast numbers of previously unknown microorganisms and their functions (Pavlopoulos et al., 2023). Metagenomics not only provides a snapshot of microbial diversity but also illuminates how microorganisms interact with their environment, including their roles in nutrient cycling, climate regulation, and pollutant degradation (Wooley et al., 2010). Furthermore, metagenomics has important implications in health and disease; for instance, analyzing the human microbiome can lead to breakthroughs in understanding diseases like obesity, diabetes, and various autoimmune disorders (Martín et al., 2014). Metagenomic techniques have also been pivotal in biotechnological applications such as the development of new pharmaceuticals and the bioremediation of contaminated environments (Thomas et al., 2012).

1.2.3 The issue of Incompleteness in MAGs and Its Impact on Pangenome Analysis

Metagenomic sequencing produces an array of sequence reads from the various organisms in a microbial community (Pavlopoulos et al., 2023) (see Figure 1.3). Complications often arise during the assembly of these reads into longer DNA segments called contigs, due to shared conserved regions across different species (Tettelin & Medini, 2020). By categorizing these contigs through characteristics like GC content, tetramer frequency, and sequence coverage, researchers can identify groups of contigs corresponding to specific species (Nurk et al., 2017). This process has led to the conceptualization of Metagenome-Assembled Genomes (MAGs), which encompass all contigs or scaffolds associated with a single species or closely related strains.

When constructing pangenomes from metagenomic data, researchers aim to obtain a comprehensive overview of all genes present in a microbial community, encompassing both core and accessory genes. This involves two primary methods: assembling reads into contigs to create complete and precise MAGs for mapping predicted genes onto known sequences, or directly aligning individual reads with known gene sequences (Tettelin & Medini, 2020). This approach is critical as it provides deeper insights into microbial diversity, ecology, and potential applications (Chen & Pachter, 2005). However, the use of MAGs introduces significant challenges, primarily due to their inherent incompleteness, contamination, and fragmentation (Eisenhofer et al., 2023).

Incomplete assemblies might fail to capture the full array of core and accessory genes, which are vital for understanding microbial adaptability and niche specialization (Tettelin et al., 2005a). Missing these genes can lead to an incomplete picture of microbial diversity and function. Tools like CheckM quantify the completeness of MAGs by evaluating the presence of a set of universal single-copy genes, providing an estimate of both the completeness and contamination levels of a genome (Parks et al., 2015). This gives an idea of how 'good' the assemblies are, indicating whether the full pangenome diversity from a set of MAGs is being captured.

Contamination occurs when sequences from different organisms are incorrectly assembled together, misrepresenting the genetic makeup of the microbial community. CheckM also quantifies contamination by identifying sequences that deviate from the expected genomic properties of the target genome (Parks et al., 2015). Reducing contamination is essential for accurate functional annotation and downstream analysis.

Fragmentation refers to the incomplete assembly of genomes into contiguous sequences, resulting in a fragmented genomic landscape. Fragmented assemblies hinder the ability to link genes with their functional contexts, complicating the interpretation of metabolic pathways and ecological interactions (Eisenhofer et al., 2023). Advances in assembly algorithms and tools aim to reduce fragmentation and improve the contiguity of MAGs. Tools like PPanGGOLiN (Gautreau et al., 2020) and mOTUpan (Buck et al., 2022) refine the quality of MAGs by estimating a likely threshold for core genes, even in the presence of missing data. These tools lower the frequency threshold for core gene identification, helping to identify which genes are considered essential across multiple genomes. However, these tools do not provide information on which specific genomes the genes are missing from. CELEBRIMBOR (Core ELEment Bias Removal In Metagenome Binned ORthologs)13, co-developed by the Lees group at EMBL-EBI, enhances the accuracy of pangenomes constructed from metagenome data by identifying and correcting biases in core gene identification (Hellewell et al., 2024). This tool addresses some of the limitations of earlier methods by focusing on bias removal, but it still does not infer from which specific genomes genes are missing. MAGgIMPUTE represents an advancement over CELE-

BRIMBOR by providing more sophisticated methods for imputing missing genes in MAGs, thereby improving the completeness and reliability of pangenomic analyses. This workflow leverages advanced statistical techniques to predict missing gene content, offering a comprehensive view of the microbial community's genetic landscape. By addressing the issue of missing genes in specific genomes, MAGgIMPUTE enhances our understanding of the full pangenome diversity captured from a set of MAGs.

The importance of constructing accurate pangenomes from metagenomes is underscored by their applications in ecological and functional genomics. As highlighted by (Parks et al., 2017) , their research significantly expanded the known diversity of life by recovering nearly 8,000 MAGs from over 1,500 public metagenomic datasets. This monumental effort not only increased the phylogenetic breadth of known microbial life but also revealed novel lineages and metabolic pathways, illustrating the vast, previously uncharted genetic diversity within microbial communities.

(Parks et al., 2017) demonstrated that accurate recovery and assembly of MAGs are essential for understanding the full spectrum of microbial diversity and function. Incomplete assemblies can lead to missing critical accessory genes, which are vital for comprehending microbial adaptability and niche specialization, and core genes, which are required for phylogenetic analysis as they are used to build trees. Contamination, where sequences from different organisms are incorrectly assembled together, and fragmentation, resulting in incomplete genome assemblies, further complicate pangenomic analyses.

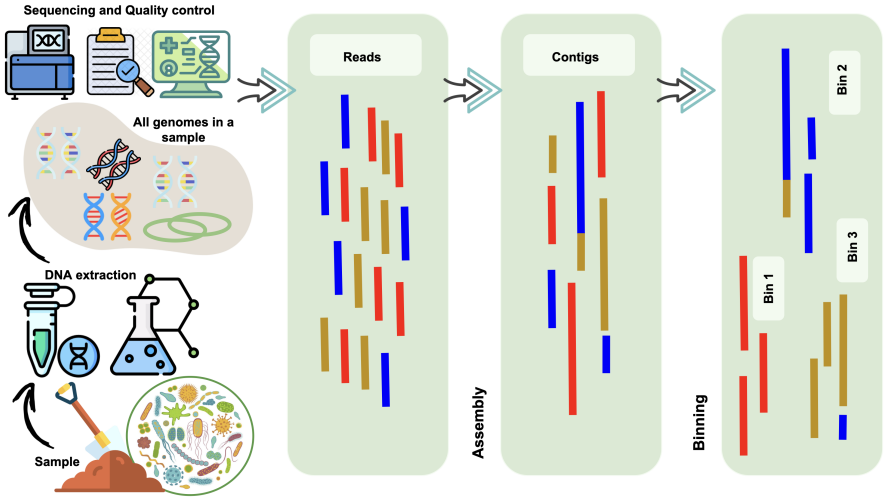

Figure 1.3: The process of MAG generation : The process starts with DNA extraction and quality control from a sample containing multiple genomes. These genomes are sequenced to generate reads, which are then assembled into contigs. The contigs are grouped through a process called binning, where each bin represents a distinct genome. This method allows for the separation and analysis of individual species within the sample. MAGs are the result of this binning process, providing insights into the genetic composition of microbial communities.

1.2.4 Graph-based gene identification and pangenome analysis tool - ggCaller

ggCaller 1.4 is an open-source software graph-gene-caller specifically designed for pangenome analysis. ggCaller emerges as an innovative tool in this context, combines gene prediction, functional annotation, and clustering into a single workflow using population-wide de Bruijn Graphs, removing redundancy in gene annotation, and resulting in more accurate gene predictions and orthologue clustering (Horsfield et al., 2023).

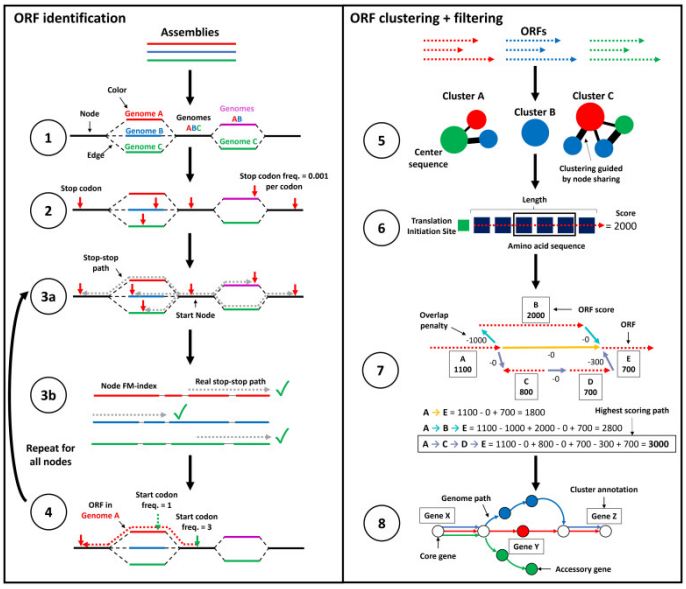

Figure 1.4: This figure summarizes ggCaller's workflow, divided into two phases: ORF identification (Steps 1-4) and ORF clustering and filtering (Steps 5-8). In the identification phase, a De Bruijn Graph (DBG) is created using Bifrost, stop codons are identified and paired, and ORFs are confirmed by start codons and translation initiation sites. In the clustering phase, ORFs are grouped into COGs, evaluated by Balrog (Sommer & Salzberg, 2021), and the highest scoring paths are determined using the Bellman-Ford algorithm (Bellman, 1958), followed by gene graph construction and refinement with Panaroo (Tonkin-Hill et al., 2020). *Figure source (Horsfield et al., 2023): Samuel T. Horsfield, Gerry Tonkin-Hill, Nicholas J. Croucher, and John A. Lees. Genome Research*

1.3 Aim of my project

This work focused on developing a streamlined and effective workflow for metagenome annotation utilizing ggCaller, aiming to optimize the annotation process in terms of both efficiency and accuracy. The integration of ggCaller significantly enhanced the analytical capabilities of researchers, providing deeper insights into the functional dynamics of microbial communities. ggCaller was used to generate a representation of a pangenome with missing data, which was then used to guide the identification of missing genes. My project unfolds in three key phases:

1.3.1 Evaluation of ggCaller on simulated Metagenome DATA

My first task involves deploying ggCaller on simulated MAGs, I will meticulously assess ggCaller's performance in the presence of missing data, providing valuable insights into its adaptability and robustness.This initial phase will establish a baseline for further improvements.

1.3.2 Implementation of a Missing Data Imputation methods

Imputation is a crucial step in the preprocessing and quality control protocol of genetic studies, playing a pivotal role in data analysis by recovering missing data. This technique is broadly applied across various fields, including recommender systems like the Netflix challenge (**Netflix**), imputation of unmeasured epigenomics datasets (Ernst & Kellis, 2015), and gene expression recovery in single-cell RNA-sequencing data (Cheng et al., 2023).
While extensively applied in human genetics, imputation methods are not exclusive to this field. These algorithms are optimized to handle high levels of genetic diversity, making them applicable across var-

ious domains. Haplotype phasing, which involves predicting the combination of alleles (haplotypes) inherited from each parent, can be determined through laboratory experiments or estimated computationally (Garg, 2021).
In the context of metagenomics, imputation methods are applied to Operational Taxonomic Units (OTUs), typically using 16S rRNA gene sequencing to identify bacteria. These methods are used to impute the presence of species within a microbiome that might have been missed, based on the presence of other species (Calle, 2019). Recent advancements include the use of machine learning techniques, such as those described by Ruochen Jiang et al. (Jiang et al., 2021) in 2021, to enhance the prediction of missing OTUs, thus improving microbial community analysis. However, imputation has not been previously applied to predict the presence or absence of missing genes in a metagenome, marking a significant opportunity for advancing metagenomic analyses.

A particularly promising area of research is the application of haplotype imputation methods to MAGs. This innovative approach, still in its early stages, has the potential to revolutionize our understanding of microbial genomes within environmental samples, offering new insights into microbial evolution and interactions.

1.3.3 Comprehensive Analysis and Comparison:

The final phase involves conducting a thorough analysis to evaluate the enhancements with the integrated imputation algorithms. Through benchmarking against existing data, I will measure improvements in gene annotation. This comprehensive evaluation will provide us a quantitative understanding of the impact of the proposed enhancements.

2. Resources

2.1 Computing Environment

The majority of this internship was conducted using the EMBL-EBI cluster, managed by the Slurm workload manager. To facilitate this, an EMBL-EBI Apple M1 MacBook Air with macOS, 16GB of memory, and 512GB of storage was employed. Connection to the cluster was established via a VPN. Initially, I utilized the LSF codon cluster but transitioned to Slurm in alignment with EMBL-EBI's organizational shift to this platform.
Throughout the internship, various Integrated Development Environments (IDEs) were utilized to enhance productivity. For Python development, the PyCharm IDE 1 by JetBrains was predominantly used. Most Python coding was performed remotely, although the Professional Edition of PyCharm, which is freely available to students and offers a more comprehensive feature set, was occasionally used. For R programming, the RStudio IDE by Posit was utilized locally on the company laptop. Additionally, VS Code 4 was employed for scripts and other programming languages, particularly for writing Nextflow 14 script. Both a locally installed version and the cluster-hosted version 22.10.1 of Nextflow were utilized to ensure compatibility and efficiency in workflow management.

2.2 Professionnal practices

2.2.1 Bibliographic and Technological Monitoring

To stay updated with the latest research and developments in the field, I initially conducted manual searches and later utilized the StorkApp 7 alert system. I configured the app to send me weekly notifications of newly published articles containing key phrases such as 'Imputation methods', 'Metagenomic Analysis Tools', 'Pathogen evolution and statistical genetics', 'Genome Assembly', 'Pangenomics', and 'Bioinformatics Techniques for MAGs'...
To efficiently manage and organize the scholarly articles relevant to my project, I employed Zotero 8, a reference management software. This tool was invaluable in collecting and formatting references, which I used to substantiate the methodologies and findings in my internship report.

2.2.2 Good computing practices

The codes was written with the intention of being more understandable. Each function was thoroughly documented with comments, and complex code blocks were clearly annotated. Comprehensive user documentation was also developed and made available on the project's GitHub page 6. To avoid version conflicts, Conda 5 was installed on the cluster. Given the pipeline's extensive dependencies and the need to integrate multiple programs, a Conda environment (v24.3.0) named "monenvie" was created to manage these dependencies. This environment is activated by the pipeline's main script upon each use. Additionally, a corresponding YML file was generated to facilitate quick setup on other machines.
Furthermore, since the cluster was managed using Slurm 3, the software was employed to monitor

resource allocation for each script and to prevent potential resource leaks. This approach ensured efficient utilization of computational resources and maintained the smooth operation of the workflow.

2.2.3 Communication of Work

Throughout the internship, we maintained a consistent meeting schedule, convening twice a week for approximately thirty minutes per session. These meetings were instrumental in delineating the next steps based on the progress and results obtained. A significant portion of the code developed during this internship has been made publicly accessible on GitHub 6, promoting transparency and collaboration.

2.3 Tools for imputation and workflow development

2.3.1 Python libraries

For this project, the python's scikit-learn package 11was used for its six imputation methods. This Scikit-learn library based imputation methods leverages simple yet effective techniques to infer missing genetic information in MAGs. This approach utilizes straightforward algorithms, which are readily available through scikit-learn librarie, to impute missing genes.

a. Simple Imputer: This method replaces missing values with the most frequent value in the column.

b. K-Nearest Neighbors Imputer (KNN): looks at the 'k' closest genome (in terms of gene presence/absence similarity) to the genome with missing data. Note that KNeighborsRegressor is different from KNN imputation, which learns from samples with missing values by using a distance metric that accounts for missing values, rather than imputing them. This approach assumes that genomes with similar genetic makeup will exhibit similar traits in their genes.

c. Matrix Factorization, SVD: decomposes the original data matrix into three smaller matrices. These matrices capture underlying patterns in the interactions between two different entities (like users and items in recommendation systems). Pattern Extraction: The idea is that much of the data is redundant, and the missing values can be predicted through the relationships implicit in the reduced dimensional space. Imputation: By reconstructing the original matrix from the decomposed matrices, SVD fills in missing values based on the patterns it has learned. This method was implemented using a package called Surprise.

d. Multivariate Imputer (Iterative Imputer): This more sophisticated technique employs multivariate imputation algorithms that consider all other columns (or genes) in the genome to estimate the missing values. It iteratively models each feature with missing values as a function of other features, and uses that estimation for imputation.

- **Bayesian Ridge**: This method applies Bayesian principles to ridge regression, which involves adding a regularization term to the regression model to prevent overfitting. Regularization works by shrinking the coefficients of less important features towards zero, which effectively reduces their influence on the model. In the context of imputing a gene presence/absence matrix, Bayesian Ridge treats the problem as a regression task where the presence (1) or absence (0) of genes is predicted based on patterns in other genes in the matrix.

- **Decision Tree Regressor** : A Decision Tree Regressor builds a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. For imputing

gene presence, it creates rules that decide whether a gene is present based on the presence or absence of other genes. This method splits the data into branches to make predictions based on feature values.

- **Kneighborsregressor**: This method imputes missing data by averaging the values of the nearest neighbors in the multidimensional space of available data. For gene presence/absence, genomes are treated as points in a multidimensional space where each dimension represents a gene's presence. The method finds the closest genomes (those with similar gene patterns) and uses their values to predict the missing gene's status.

- **Extra tree regressor**: Also known as Extremely Randomized Trees, this method predicts gene presence by averaging results from multiple trees built using random subsets of the data. Unlike regular decision trees, Extra Trees Regressors use random splits of the data to create diverse trees, which are then averaged to improve prediction accuracy and reduce overfitting.

For the matrix factorization method, the Surprise package was installed and used. Surprise 12 is a Python library designed for building and analyzing recommender systems, particularly those dealing with explicit rating data. It supports various matrix factorization algorithms, these algorithms decompose the gene-MAG interaction matrix into lower-dimensional representations, making it easier to predict missing entries.

2.3.2 R packages

The data visualizations for this analysis were conducted using RStudio 2, leveraging the resources of a SLURM cluster. By utilizing SLURM 3, we were able to efficiently manage and execute computational tasks across multiple nodes, ensuring that the data processing and visualization tasks were handled effectively. R packages such as `ggplot2` and `pheatmap` were employed to create comprehensive visual representations of the imputation results. The integration of RStudio with the SLURM workload manager allowed for seamless execution of parallel jobs, significantly improving the efficiency and scalability of our data analysis workflow.

2.4 Statistical metrics and Data

In this section, I present a detailed analysis of the performance metrics used to evaluate the imputation methods. These metrics are crucial for understanding the effectiveness of the models in predicting genes presence or absence accurately. The confusion matrix, forms the foundation for calculating these metrics and is estimated as follows:

True Positives (TP): The MAG is 0, and the imputed value is 1, with the reference indicating the gene is present (1). True Negatives (TN): The MAG is 0, and the imputed value is 0, with the reference indicating the gene is absent (0). False Positives (FP): The MAG is 0, and the imputed value is 1, incorrectly, as the reference indicates the gene is absent (0). False Negatives (FN): The MAG is 0, and the imputed value is 0, incorrectly, as the reference indicates the gene is present (1). By focusing on sites where the MAG was initially 0, we aim to reduce the similarity in measures, expecting a significant number of 1s, especially in the core genome. These definitions help to calculate the various performance metrics as follows:

- **Precision**: This indicates the proportion of true positive instances out of the total instances predicted as positive. Precision of a model can be assessed by asking, "Among all the instances where the model predicted a 'Positive' outcome, how often was this prediction accurate?" The goal is to minimize errors in identifying positive labels. It is important to note that this metric does not consider negative labels. It is calculated as:

$$Precision = \frac{TruePositives(TP)}{TruePositives(TP) + FalsePositives(FP)} \tag{2.1}$$

- **F1-Score**: This is the harmonic mean of precision and recall (sensitivity) and provides a balance between them. It is calculated as:

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.2}$$

where Recall (Sensitivity) is defined as:

$$Recall(Sensitivity) = \frac{TruePositives(TP)}{TruePositives(TP) + FalseNegatives(FN)} \tag{2.3}$$

- **Sensitivity (Recall)**: This measures the proportion of actual positive instances that are correctly identified. A model's sensitivity is determined by asking, "When the true outcome class was 'Positive', how frequently did the model correctly predict it?" It is calculated as:

$$Sensitivity(Recall) = \frac{TruePositives(TP)}{TruePositives(TP) + FalseNegatives(FN)} \tag{2.4}$$

- **Specificity**: This measures the proportion of actual negative instances that are correctly identified. The specificity of a model can be determined by asking, "Out of all the instances where the true outcome was 'Negative', how frequently did the model correctly identify it as such?" It is calculated as:

$$Specificity = \frac{TrueNegatives(TN)}{TrueNegatives(TN) + FalsePositives(FP)} \tag{2.5}$$

2.4.1 Genomes datasets

Three distinct datasets representing different microbial species were used in this internship: 90 capsular biosynthetic locus from all *pneumococcal serotypes* (Bentley et al., 2006) (SP PRJEB2632), 162 representative genomes of *Escherichia coli* (bsac collection ref) (Kallonen et al., 2017), and 616 genomes of *Streptococcus pneumoniae* (Croucher et al., 2015). However, only the results of the workflow on *Streptococcus pneumoniae* are presented in this report.
These datasets collectively provided a comprehensive basis for evaluating and refining the workflow, ensuring its applicability and reliability across different microbial species and dataset sizes.

3. Results

3.1 Simulating MAGs: motivation and method

The MAG simulation process involves generating synthetic, incomplete metagenome-assembled genomes (MAGs) to reflect the challenges faced in metagenomic studies, such as incomplete genomic data. The purpose of this process is to provide a realistic testing environment for metagenomic analysis tools. To achieve this, I used the remove-sequence.py script developed by my supervisor. The script randomly removes sections from each genome according to the specified parameters. The simulated MAGs are saved in a specified output directory with a completeness file listing details about each simulated MAG, including filename, completeness level, and removed sections.

3.1.1 Analysis of Simulated MAGs on Different Datasets

The assessment of simulated MAGs focuses on their resemblance to real-world incomplete genomes. Several key metrics are evaluated to ensure the validity of the simulation: Completeness Distribution: The distribution of completeness levels in the simulated MAGs is represented in figure 3.1, as illustrated in the histograms on the right side of each figure. These histograms show a range of completeness levels from highly fragmented to nearly complete genomes, indicating that the simulation successfully generates a realistic spectrum of MAG completeness.
Randomness of Section Removal: The script's effectiveness in randomly removing genome sections is evaluated through density plots shown on the left side of each figure. These plots compare the density of contig lengths for both complete genomes and simulated MAGs. The differences in contig lengths indicate that sections of the assemblies have been removed, meaning the simulation is working as expected. The difference in total assembly size, with MAGs being smaller than the true assemblies, confirms that the simulated incompleteness realistically represents the randomness seen in actual metagenomic samples. By thoroughly analyzing these aspects, the simulation process ensures that the generated MAGs are realistic and useful for testing our workflow.

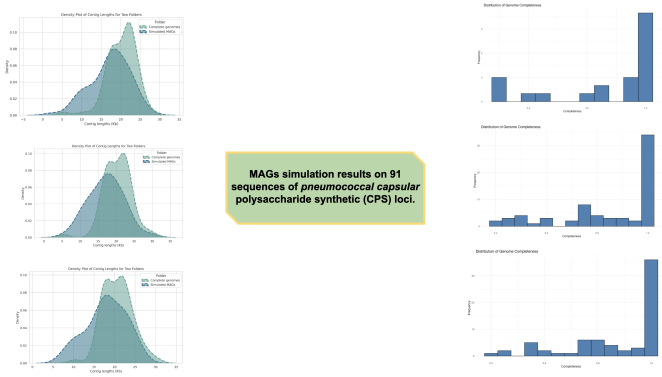

Figure 3.1: Each plot shows the density of contig lengths for both complete genomes and simulated MAGs. The overlapping curves indicate that the simulation closely mimics the length distribution of real contigs. Each histogram displays the distribution of genome completeness in the simulated MAGs, showing a range of completeness levels from highly fragmented to nearly complete genomes. Each plot represents a different run.

3.2 Adaptation and modifications of the original workflow

3.2.1 Original ggCaller's graph traversal function

Representing a collection of genomes in a graph can significantly reduce the size of the data structure while enhancing the efficiency of various genomic analyses. Methods that leverage graph traversal of an edge-induced de Bruijn graph (DBG), which include all possible edges between nodes, offer a powerful approach for managing the complexity and redundancy inherent in large genomic datasets. The DBG representation captures the sequence overlaps between different genomes, allowing for a compact and comprehensive depiction of the pangenome.

ggCaller 10 scales with both the number of edges and nodes in a DBG due to its innovative use of depth-first search (DFS) during stop codon pairing and gene adjacency identification. By utilizing depth-first search, *ggCaller* efficiently navigates through the DBG, ensuring that each potential gene is examined in the context of its neighboring sequences. This method not only enhances the prediction's accuracy but also significantly reduces the computational time compared to traditional linear genome-based workflows. The performance improvements observed with *ggCaller* are particularly noteworthy when dealing with large-scale genomic data, where linear methods can become prohibitively slow and resource-intensive.

The query function in *ggCaller* is designed to efficiently retrieve genes from reference sequences within the pangenome graph with high precision. This function operates by searching for k-mers, which are short sequences of nucleotides, within the graph. When a query is made, the function scans the graph to find matches between the query k-mers and the k-mers associated with annotated genes in the graph. If a match is found, the corresponding gene is returned as a hit. This method ensures that genes are accurately identified based on sequence similarity, leveraging the structure of the graph to provide rapid and reliable results.

3.2.2 Adaptation of the graph traversal function to MAGs

I focused on making significant modifications to the existing graph traversal function, enhancing its ability to process and analyze MAGs within the context of the pangenome graph. Below, I detail the specific enhancements made.

Originally, the *ggCaller* query function was limited to querying genes from reference sequences within the pangenome graph. This approach was effective for well-annotated reference genomes but lacked the flexibility needed to handle the complex and often incomplete data characteristic of MAGs. To address this limitation, I modified the query function to allow entire MAGs to be queried within the ggCaller graph. This modification enables a more comprehensive analysis by matching each gene predicted in the reference sequences against the k-mers from the MAGs. The enhanced function works by querying k-mers from each MAG and searching them within the graph. If any query k-mers match a gene annotated in the graph, the gene is returned as a hit. This approach ensures efficient identification of genes within the metagenomic data.

Gene Detection and Matrix Generation

After Modification: For every gene detected or matched in a MAG, the updated version now

generates a gene presence/absence matrix. This provides a clear representation of which genes are present or absent across different MAGs in the dataset. This functionality facilitates a deeper understanding of gene distribution and variability within MAGs.

Implementation Details

To implement these enhancements, I introduced a new argument for the *ggCaller* command line tool. This argument specifies the path to the gene graphs post-quality control with Panaroo. generated by *ggCaller*, along with the intermediate data structures created using the `--save` argument. The process for each MAG in the dataset is as follows:

1. **Query the Entire MAG:** The modified *ggCaller* graph query function processes the entire MAG, matching each gene previously predicted in the reference sequences to the MAG. By default, this matching is based on a single k-mer overlap.

2. **Generate Gene Presence/Absence Matrix:** Once the query is complete, the tool generates a gene presence/absence matrix. This matrix is instrumental in comparative genomics studies, as it highlights the genetic similarities and differences between various MAGs.

Command for Execution

The new command structure for executing these queries includes the paths to all MAGs and the Panaroo graph of the reference sequence. An example command might look like this:

```
ggcaller --query MAGs_folder/input.txt --panaroo_graph ggCaller_output/final_graph.gml

--graph reference_files/input.gfa --colours ggCaller_output/ggc_data --threads 4
```

This command ensures that *ggCaller* processes the specified MAGs using the updated query function and generates the necessary gene presence/absence matrices for further analysis.

3.2.3 MAGGImpute Workflow

In this section, I present the MAGGImpute workflow that I developed to annotate simulated MAG's genomes using ggCaller. Below, I describe each step of the workflow, as illustrated in Figure 3.2.

Workflow Overview

The workflow begins with the dataset being processed by a Nextflow pipeline, which orchestrates the preprocessing and preparation of the genomic data by splitting the genomes and repeating the process for 10 runs. The dataset is then randomly divided into two subsets: 25% of the genomes are designated for training, and 75% for simulation. This division mimics real metagenome analysis, ensuring a robust basis for gene prediction while reflecting the complexity and variability of real-world metagenomic samples. The 25% subset is processed using ggCaller for initial gene calling, generating gene graphs and a gene presence/absence matrix for the reference genomes. Next, a set of pre-simulated MAGs is prepared as a ground truth for benchmarking. MAGs are simulated based on varying levels of completeness, and the updated graph traversal function of ggCaller matches queries from both pre-simulated and post-simulated MAGs to the reference matrix, with a single k-mer overlap as the default threshold. Gene presence/absence matrices are generated for both the reference genomes and the simulated MAGs, capturing gene content across varying levels of completeness. Models are trained on the 25% subset of the original reference data to predict and impute missing

gene presence/absence information, with seven different imputation methods applied to the simulated MAGs' gene presence/absence matrix. Finally, the imputed MAGs' matrix is compared against the ground truth matrix to assess the accuracy and effectiveness of the imputation methods, and the results are analyzed to determine the best approaches for handling incomplete MAG data.

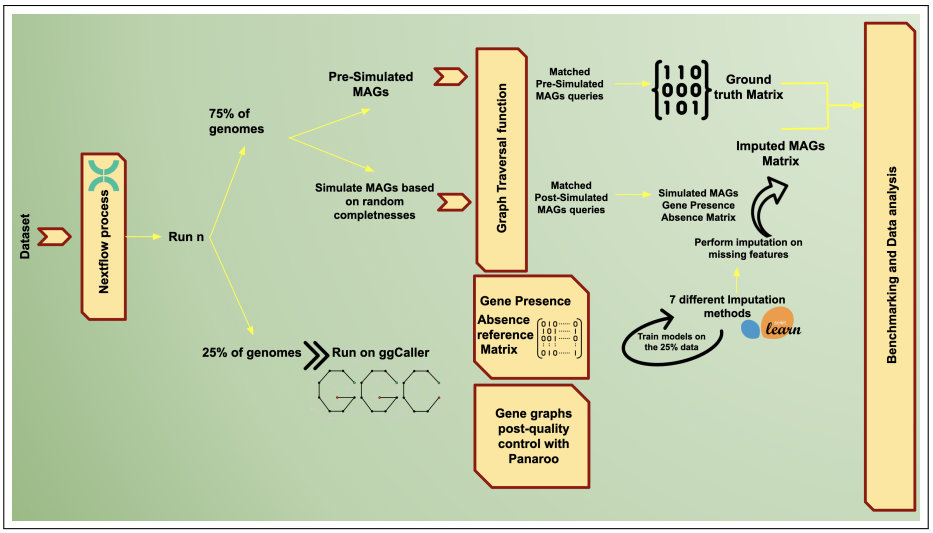

Figure 3.2: Simplified representation of MAGGImpute workflow.

3.3 Workflow's performances on 492 genomes of *Streptococcus pneumoniae*

In this results section, we meticulously evaluate the performance of the MAGGImpute workflow applied to a dataset consisting of 492 genomes of *Streptococcus pneumoniae*. This extensive dataset allowed for a thorough evaluation of the workflow's scalability and performance. To assess the variability across runs, the workflow was executed 10 times.

This analysis includes examining the workflow's effectiveness across different imputation methods. The key performance metrics evaluated are precision, F1-score, sensitivity, specificity. The results are illustrated in the figures below.

3.3.1 Sensitivity

The sensitivity of the various imputation methods is shown in the first box plot. These imputation methods are, respectively, Bayesian Ridge (BR), Matrix Factorization (MF), Extra Tree Regressor (ETR), Decision Tree Regressor (DTR), K-Nearest Neighbors (KNN), KNeighbors Regressor (KNR), and Simple Imputer (SI). Sensitivity measures the proportion of true positive results correctly identified by the model. Here 3.3, we observe that all methods exhibit variability in sensitivity across the 10 runs. Mags Run, representing the matrix before any imputation, has the lowest sensitivity, consistently remaining at 0.

3.3.2 Precision

Precision is depicted in the second box plot 3.4 and represents the proportion of positive identifications that are actually correct.

Precision is depicted in the second box plot and represents the proportion of positive identifications

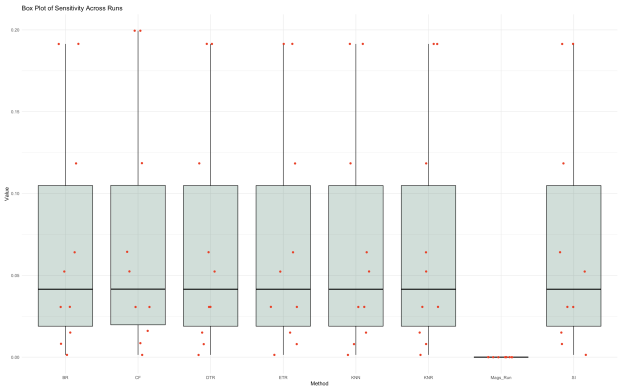

Figure 3.3: Sensitivity of Different Methods Across Runs

Figure 3.4: Precision of Different Methods Across Runs

that are actually correct. From the plot, most imputation methods have very low precision values, clustering near zero. There is noticeable variation across runs, but the precision remains generally low across all methods. This indicates that while some positive results are correctly identified, many are not, leading to a high false positive rate.

3.3.3 Specificity

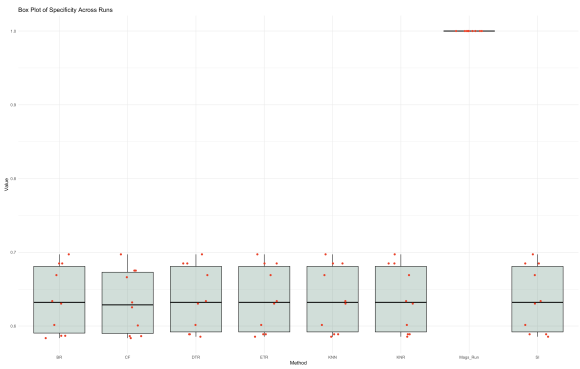

Figure 3.5: Specificity of Different Methods Across Runs

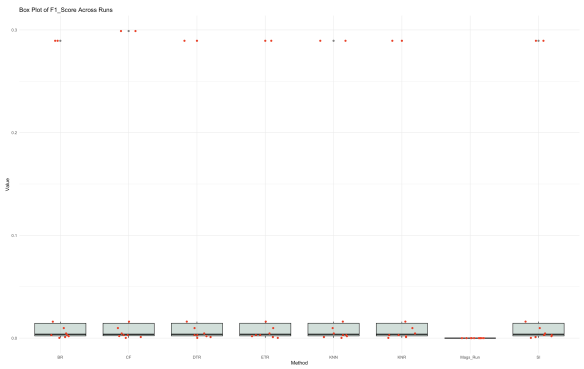

Figure 3.6: F1 Score of Different Methods Across Runs

Specificity, shown in the third box plot 3.5, measures the proportion of true negative results correctly identified. Key observations include that specificity values are higher compared to sensitivity and precision, indicating that the methods are better at identifying true negatives. The Mags before any imputation has the highest specificity, close to 1. the other methods show moderate specificity with some variability.

3.3.4 F1 Score

The F1 score, represented in the fourth box plot 3.6, is the harmonic mean of precision and sensitivity, providing a balance between the two. Analysis of the plot reveals that F1 scores are generally low, reflecting the low precision and sensitivity observed. There is variability across methods and runs, with no method consistently outperforming others. The low F1 scores suggest that the imputation methods struggle to balance precision and sensitivity effectively.

3.3.5 Relationship Between Performance Metrics and Gene Frequency

The analysis of the matrix factorization-based imputation method 3.7 revealed that True Positives (TP) initially increase with gene frequency, indicating good model performance at mid-range frequencies, but decline at very high frequencies. True Negatives (TN) decrease as gene frequency increases, which is expected because there are fewer 0s in genomes for core genes as they are present in more genomes initially. False Positives (FP) decrease as gene frequency increases, implying improved precision at higher frequencies. Conversely, False Negatives (FN) are low in accessory genes but high in core genes. This is because core genes are expected to be present (1s) in more genomes, so when they are missing, we get more FNs, but also more TPs. In accessory genes, where more 0s are expected, it is more likely to get more TNs, but also more FPs. These trends suggest that while the method performs better with higher gene frequencies, it faces challenges with low frequencies, highlighting areas for potential optimization and threshold adjustment.

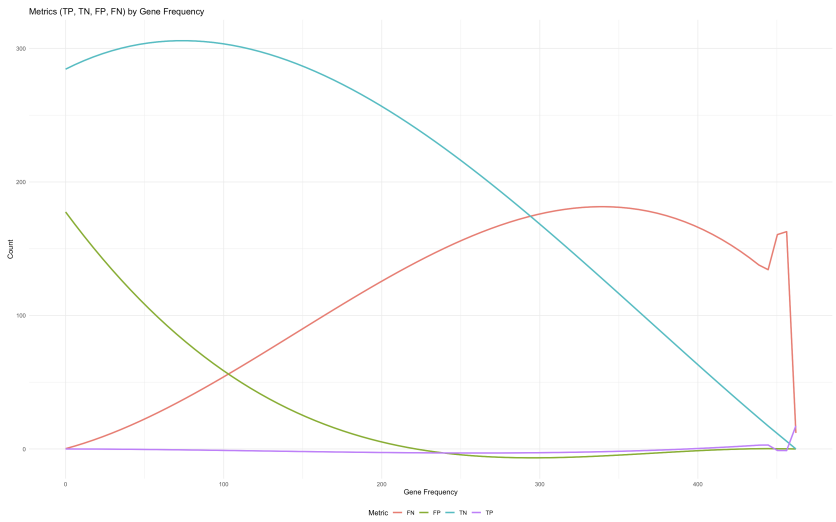

Figure 3.7: This graph illustrates the variation in True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) with gene frequency. The imputation method shows improved accuracy at higher gene frequencies, with challenges noted at very low and very high frequencies.

The performance evaluation of the MAGGImpute workflow across 492 genomes of Streptococcus pneumoniae demonstrates variability in key metrics across different imputation methods. Sensitivity and precision are generally low, indicating challenges in correctly identifying true positives and minimizing false positives. However, specificity is higher, suggesting that the methods are more effective in correctly identifying true negatives, mainly in accessory genome. The low F1 scores further highlight the need for improvements in balancing precision and sensitivity. Upon comparison, none of the imputation methods demonstrated superior performance over the others; they all performed equivalently.

4. Discussion

The comprehensive evaluation of the MAGGImpute workflow across multiple datasets has yielded several important insights. One of the key insights from this evaluation is that no imputation method performed better than any other. While all methods performed reliably across multiple metrics, there was no single method that consistently outperformed the others. The high sensitivity values across all methods indicate that the workflow is proficient in correctly identifying true negatives, an essential aspect for reliable genomic analysis.
The evaluation reveals that imputation methods generally perform better with core genes than with accessory genes. Core genes are typically more conserved and present across multiple genomes, making them easier to predict and impute. In contrast, accessory genes are more variable and less consistently present, posing a greater challenge for imputation algorithms. This disparity in performance may also be due to the choice of reference datasets. The reference panel can have a significant impact on performance, as demonstrated by the variability in results for the same method using different reference datasets. This highlights the importance of ensuring that the reference panel is representative of the population diversity.

To improve the reference panel, it is crucial to include a diverse set of genomes that capture the full spectrum of genetic variation within the population. This can be achieved by incorporating samples from different geographical locations, environments, and lineages. Tools like PopPUNK (Lees et al., 2019) allow for the generation of a representative panel of genomes, ensuring comprehensive coverage of genetic diversity. Additionally, having a sufficiently large reference panel increases the likelihood of capturing rare and unique genes, which are often missed in smaller panels, thus improving the imputation of accessory genes that might be underrepresented. It is also essential to avoid bias in the reference panel by ensuring that it is not skewed towards specific lineages or clonal groups. Including a variety of strains helps train the imputation model on a broad range of genetic backgrounds, enhancing its generalizability.

To address the challenges posed by accessory genes, more sophisticated machine learning techniques and customized algorithms can be employed. For example, deep learning models such as Convolutional Neural Networks (CNNs), Hidden-markov models (HMM) and Recurrent Neural Networks (RNNs) can model complex relationships in genomic data, capturing intricate patterns and dependencies suitable for imputing variable accessory genes. Ensemble methods like Gradient Boosting Machines (GBMs) and XGBoost combine multiple models to make predictions, reducing the likelihood of errors that might occur from any single model. Transfer learning involves a method that can be used universally across neural networks. Additionally, autoencoders can learn efficient representations of the data, capturing essential features while reducing dimensionality, which can be used to identify and impute missing genes by learning the underlying structure of the genomic data.
An interesting avenue for further exploration involves measuring contamination levels and fragmentation to evaluate the fidelity of the simulated MAGs and ensure they closely resemble real-world scenarios.
A critical aspect of improving the MAGGImpute workflow involves adjusting the proportion of k-mers

matched to a gene. We only match single k-mers to identify a match. Increasing this number may increase the specificity and precision of gene calls. This adjustment ensures that only sequences with substantial matches are considered, improving the overall accuracy of annotations and reducing erroneous detections.
A captivating aspect of ggCaller is its ability to detect overlapping open reading frames (ORFs). This feature, while advantageous for identifying novel genes, may also lead to the detection of "spurious genes" after MAG simulation. These spurious genes result from overlapping ORFs that were not initially detected as genes pre-simulation due to their complex nature. During the simulation process, the removal of certain sequences can lead to these previously overlapping ORFs being wrongly identified as genes. This phenomenon underscores the need for careful validation and quality control to differentiate between true gene calls and artifacts of the simulation process. However, this aspect was not explicitly tested in this internship and warrants further investigation to confirm its impact on gene prediction accuracy. Future studies should focus on assessing the extent to which these overlapping ORFs, now identified as genes post-simulation, affect the overall accuracy of gene predictions. This will involve detailed analysis and validation steps to distinguish between genuine gene calls and artifacts introduced during the sequence removal and simulation processes.

In addition to the points discussed above, it is crucial to consider the scalability of the workflow. While this has not been explicitly tested, it is something we aim to test in the future. Some bacterial species, such as *Streptococcus pneumoniae*, have tens of thousands of genomes. Ensuring that MAGGIMPUTE can maintain high performance across such large datasets is essential for future applications in metagenomics, where large and complex datasets are becoming increasingly common.

5. Conclusion

The Lees group, renowned for its innovative contributions to microbial genomics and bioinformatics, has continually advanced the field through the development of cutting-edge software tools like ggCaller. The primary objective of this report was to evaluate and enhance the performance of ggCaller within the context of metagenomics, focusing on its application to MAGs and ensuring high accuracy and reliability in gene prediction and imputation. The primary challenge in metagenomic analysis lies in the handling of incomplete and fragmented genomic data, which complicates gene annotation and pangenome construction. Traditional methods often fall short in providing accurate and reliable results due to the variability and complexity of MAGs. This problematic landscape necessitated the development of a robust pipelines capable of efficiently imputing missing genes and improving the accuracy of gene predictions.

The objective of my internship was to develop a workflow that not only handles incomplete genomic data with high accuracy but also improves the overall quality of MAG annotations. The specific aims included simulating MAGs with varying completeness, and exploring different imputation methods to identify the most effective strategies for metagenomic data analysis. Thus, I first adapted, improved, and secured the pipeline scripts to make them easier to use, especially on a large scale, by reorganizing the inputs and outputs in particular. I also wrote the complete documentation for this tool and set up a GitHub repository 6 to share it.

The comprehensive evaluation demonstrated that no imputation method performed better than any other. This workflow, which I named MAGGImpute, is efficient for handling incomplete genomic data with high reliability. However, there is always room for improvement. The next important step is to determine how large the reference panel needs to be to improve performance, particularly for accessory genes. Additionally, fine-tuning the workflow parameters to address variability in sensitivity will be crucial for consistently capturing all true positives. Validating the workflow across a broader range of microbial datasets will ensure its applicability and robustness in diverse research scenarios.

In terms of future work over the next two months, a thorough analysis with existing tools like Celebrimbor could be conducted to highlight the effectiveness of new imputation methods. Celebrimbor, known for refining core gene identification in metagenomes, serves as a foundation, and this work represents the next step in evolving imputation strategies for more accurate and reliable metagenomic analyses. If pursued as a PhD project, the focus could extend to integrating new methods with real-time data analysis, exploring co-assembly techniques, and developing novel algorithms tailored to the unique challenges of accessory genes. The work presented here represents a next step in the evolution of metagenomic analysis tools, building on the foundation laid by tools like Celebrimbor. While Celebrimbor has contributed significantly to the field, MAGGImpute aims to advance this further by addressing specific challenges in gene imputation and prediction within MAGs. The insights gained from this comprehensive evaluation pave the way for future improvements, shaping ggCaller and MAGGImpute into pioneering tools in metagenomic annotation methodologies, ensuring robust analyses in microbial genomics.

Bibliography

Abondio, P., Cilli, E., & Luiselli, D. (2023). Human pangenomics: Promises and challenges of a distributed genomic reference. *Life (Basel)*, *13*(6).

Baumdicker, F., Hess, W. R., & Pfaffelhuber, P. (2012). The infinitely many genes model for the distributed genome of bacteria. *Genome Biol. Evol.*, *4*(4), 443–456.

Bellman, R. (1958). On a routing problem. *Quart. Appl. Math.*, *16*(1), 87–90.

Bentley, S. D., Aanensen, D. M., Mavroidi, A., Saunders, D., Rabbinowitsch, E., Collins, M., Donohoe, K., Harris, D., Murphy, L., Quail, M. A., Samuel, G., Skovsted, I. C., Kaltoft, M. S., Barrell, B., Reeves, P. R., Parkhill, J., & Spratt, B. G. (2006). Genetic analysis of the capsular biosynthetic locus from all 90 pneumococcal serotypes. *PLoS Genet.*, *2*(3), e31.

Brynildsrud, O., Bohlin, J., Scheffer, L., & Eldholm, V. (2016). Rapid scoring of genes in microbial pan-genome-wide association studies with scoary. *Genome Biol.*, *17*(1).

Buck, M., Mehrshad, M., & Bertilsson, S. (2022). mOTUpan: A robust bayesian approach to leverage metagenome-assembled genomes for core-genome estimation. *NAR Genom. Bioinform.*, *4*(3), lqac060.

Calle, M. L. (2019). Statistical analysis of metagenomics data. *Genomics Inform.*, *17*(1), e6.

Chen, K., & Pachter, L. (2005). Bioinformatics for whole-genome shotgun sequencing of microbial communities. *PLoS Comput. Biol.*, *1*(2), 106–112.

Cheng, Y., Ma, X., Yuan, L., Sun, Z., & Wang, P. (2023). Evaluating imputation methods for single-cell RNA-seq data. *BMC Bioinformatics*, *24*(1), 302.

Compeau, P. E. C., Pevzner, P. A., & Tesler, G. (2011). How to apply de bruijn graphs to genome assembly. *Nat. Biotechnol.*, *29*(11), 987–991.

Computational Pan-Genomics Consortium. (2018). Computational pan-genomics: Status, promises and challenges. *Brief. Bioinform.*, *19*(1), 118–135.

Croucher, N. J., Finkelstein, J. A., Pelton, S. I., Parkhill, J., Bentley, S. D., Lipsitch, M., & Hanage, W. P. (2015). Population genomic datasets describing the post-vaccine evolutionary epidemiology of streptococcus pneumoniae. *Sci. Data*, *2*(1), 150058.

Eisenhofer, R., Odriozola, I., & Alberdi, A. (2023). Impact of microbial genome completeness on metagenomic functional inference. *ISME Commun.*, *3*(1), 12.

Eizenga, J. M., Novak, A. M., Sibbesen, J. A., Heumos, S., Ghaffaari, A., Hickey, G., Chang, X., Seaman, J. D., Rounthwaite, R., Ebler, J., Rautiainen, M., Garg, S., Paten, B., Marschall, T., Sirén, J., & Garrison, E. (2020). Pangenome graphs. *Annu. Rev. Genomics Hum. Genet.*, *21*(1), 139–162.

Ernst, J., & Kellis, M. (2015). Large-scale imputation of epigenomic datasets for systematic annotation of diverse human tissues. *Nat. Biotechnol.*, *33*(4), 364–376.

Garg, S. (2021). Computational methods for chromosome-scale haplotype reconstruction. *Genome Biol.*, *22*(1), 101.

Gautreau, G., Bazin, A., Gachet, M., Planel, R., Burlot, L., Dubois, M., Perrin, A., Médigue, C., Calteau, A., Cruveiller, S., Matias, C., Ambroise, C., Rocha, E. P. C., & Vallenet, D. (2020). PPanGGOLiN: Depicting microbial diversity via a partitioned pangenome graph. *PLoS Comput. Biol.*, *16*(3), e1007732.

Golicz, A. A., Batley, J., & Edwards, D. (2016). Towards plant pangenomics. *Plant Biotechnol. J.*, *14*(4), 1099–1105.

Golicz, A. A., Bayer, P. E., Bhalla, P. L., Batley, J., & Edwards, D. (2020). Pangenomics comes of age: From bacteria to plant and animal applications. *Trends Genet.*, *36*(2), 132–145.

Handelsman, J., Rondon, M. R., Brady, S. F., Clardy, J., & Goodman, R. M. (1998). Molecular biological access to the chemistry of unknown soil microbes: A new frontier for natural products. *Chem. Biol.*, *5*(10), R245–9.

Horsfield, S. T., Tonkin-Hill, G., Croucher, N. J., & Lees, J. A. (2023). Accurate and fast graph-based pangenome annotation and clustering with ggcaller. *Genome Res.*, *33*(9), 1622–1637.

Jiang, R., Li, W. V., & Li, J. J. (2021). Mbimpute: An accurate and robust imputation method for microbiome data. *Genome Biol.*, *22*(1), 192.

Jonkheer, E. M., van Workum, D.-J. M., Sheikhizadeh Anari, S., Brankovics, B., de Haan, J. R., Berke, L., van der Lee, T. A. J., de Ridder, D., & Smit, S. (2022). PanTools v3: Functional annotation, classification and phylogenomics. *Bioinformatics*, *38*(18), 4403–4405.

Kallonen, T., Brodrick, H. J., Harris, S. R., Corander, J., Brown, N. M., Martin, V., Peacock, S. J., & Parkhill, J. (2017). Systematic longitudinal survey of invasive escherichia coli in england demonstrates a stable population structure only transiently disturbed by the emergence of ST131. *Genome Res.*, *27*(8), 1437–1449.

Kleftogiannis, D., Kalnis, P., & Bajic, V. B. (2013). Comparing memory-efficient genome assemblers on stand-alone and cloud infrastructures. *PLoS One*, *8*(9), e75505.

Lees, J. A., Harris, S. R., Tonkin-Hill, G., Gladstone, R. A., Lo, S. W., Weiser, J. N., Corander, J., Bentley, S. D., & Croucher, N. J. (2019). Fast and flexible bacterial genomic epidemiology with PopPUNK. *Genome Res.*, *29*(2), 304–316.

Martín, R., Miquel, S., Langella, P., & Bermúdez-Humarán, L. G. (2014). The role of metagenomics in understanding the human microbiome in health and disease. *Virulence*, *5*(3), 413–423.

Nurk, S., Meleshko, D., Korobeynikov, A., & Pevzner, P. A. (2017). metaSPAdes: A new versatile metagenomic assembler. *Genome Res.*, *27*(5), 824–834.

Parks, D. H., Chuvochina, M., Waite, D. W., Rinke, C., Skarshewski, A., Chaumeil, P.-A., & Hugenholtz, P. (2018). A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. *Nat. Biotechnol.*, *36*(10), 996–1004.

Parks, D. H., Imelfort, M., Skennerton, C. T., Hugenholtz, P., & Tyson, G. W. (2015). CheckM: Assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Res.*, *25*(7), 1043–1055.

Parks, D. H., Rinke, C., Chuvochina, M., Chaumeil, P.-A., Woodcroft, B. J., Evans, P. N., Hugenholtz, P., & Tyson, G. W. (2017). Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. *Nat. Microbiol.*, *2*(11), 1533–1542.

Pavlopoulos, G. A., Baltoumas, F. A., Liu, S., Selvitopi, O., Camargo, A. P., Nayfach, S., Azad, A., Roux, S., Call, L., Ivanova, N. N., Chen, I. M., Paez-Espino, D., Karatzas, E., Novel Metagenome Protein Families Consortium, Iliopoulos, I., Konstantinidis, K., Tiedje, J. M., Pett-Ridge, J., Baker, D., … Kyrpides, N. C. (2023). Unraveling the functional dark matter through global metagenomics. *Nature*, *622*(7983), 594–602.

Sherman, R. M., & Salzberg, S. L. (2020). Pan-genomics in the human genome era. *Nature Reviews Genetics*, *21*(4), 243–254. https://doi.org/10.1038/s41576-020-0210-7

Sommer, M. J., & Salzberg, S. L. (2021). Balrog: A universal protein model for prokaryotic gene prediction. *PLoS Comput. Biol.*, *17*(2), e1008727.

Tettelin, H., Masignani, V., Cieslewicz, M. J., Donati, C., Medini, D., Ward, N. L., Angiuoli, S. V., Crabtree, J., Jones, A. L., Durkin, A. S., Deboy, R. T., Davidsen, T. M., Mora, M., Scarselli, M., Margarit y Ros, I., Peterson, J. D., Hauser, C. R., Sundaram, J. P., Nelson, W. C., ... Fraser, C. M. (2005a). Genome analysis of multiple pathogenic isolates of streptococcus agalactiae: Implications for the microbial "pan-genome". *Proc. Natl. Acad. Sci. U. S. A.*, *102*(39), 13950–13955.

Tettelin, H., Masignani, V., Cieslewicz, M. J., Donati, C., Medini, D., Ward, N. L., Angiuoli, S. V., Crabtree, J., Jones, A. L., Durkin, A. S., Deboy, R. T., Davidsen, T. M., Mora, M., Scarselli, M., Margarit y Ros, I., Peterson, J. D., Hauser, C. R., Sundaram, J. P., Nelson, W. C., ... Fraser, C. M. (2005b). Genome analysis of multiple pathogenic isolates of streptococcus agalactiae: Implications for the microbial "pan-genome". *Proc. Natl. Acad. Sci. U. S. A.*, *102*(39), 13950–13955.

Thakur, M., Bateman, A., Brooksbank, C., Freeberg, M., Harrison, M., Hartley, M., Keane, T., Kleywegt, G., Leach, A., Levchenko, M., Morgan, S., McDonagh, E. M., Orchard, S., Papatheodorou, I., Velankar, S., Vizcaino, J. A., Witham, R., Zdrazil, B., & McEntyre, J. (2022). Embl's european bioinformatics institute (embl-ebi) in 2022. *Nucleic Acids Research*, *51*(D1), D9–D17. https://doi.org/10.1093/nar/gkac1098

Thomas, T., Gilbert, J., & Meyer, F. (2012). Metagenomics - a guide from sampling to data analysis. *Microb. Inform. Exp.*, *2*(1), 3.

Tonkin-Hill, G., MacAlasdair, N., Ruis, C., Weimann, A., Horesh, G., Lees, J. A., Gladstone, R. A., Lo, S., Beaudoin, C., Floto, R. A., Frost, S. D. W., Corander, J., Bentley, S. D., & Parkhill, J. (2020). Producing polished prokaryotic pangenomes with the panaroo pipeline. *Genome Biol.*, *21*(1), 180.

Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., Žídek, A., Green, T., Tunyasuvunakool, K., Petersen, S., Jumper, J., Clancy, E., Green, R., Vora, A., Lutfi, M., ... Velankar, S. (2021). Alphafold protein structure database: Massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, *50*(D1), D439–D444. https://doi.org/10.1093/nar/gkab1061

Wooley, J. C., Godzik, A., & Friedberg, I. (2010). A primer on metagenomics. *PLoS Comput. Biol.*, *6*(2), e1000667.

Webography

1. PyCharm: A Python IDE for professional developers.
   Accessed on 12 March, at https://www.jetbrains.com/pycharm/.

2. RStudio: An integrated development environment for R.
   Accessed on 12 March, at https://rstudio.com/.

3. SLURM Cluster: An open-source workload manager designed for Linux clusters of all sizes.
   Accessed on 12 March, at https://slurm.schedmd.com/.

4. Visual Studio Code: A free source-code editor made by Microsoft.
   Accessed on 12 March, at https://code.visualstudio.com/.

5. Conda: An open-source package management and environment management system.
   Accessed on 06 March, at https://docs.conda.io/.

6. A significant portion of the code developed during this internship has been made publicly accessible on GitHub.
   Accessed on 23 June, at https://github.com/Lalemaouloud/MAGGImpute.

7. StorkApp alert system: An app designed to provide timely notifications and alerts for various events.
   Accessed on 12 March, at https://www.storkapp.me.

8. Zotero: A free, easy-to-use tool to help you collect, organize, cite, and share research.
   Accessed on 12 March, at https://www.zotero.org.

9. The Netflix challenge: A competition to improve the accuracy of the recommendation algorithm used by Netflix. (2018).
   Accessed on 12 March, at https://www.researchgate.net/publication/326694752$_{T}he_{N}etflix_{C}hallenge$.

10. ggCaller: A tool for gene calling and annotation in bacterial genomes.
    Accessed on 12 March, at https://github.com/bacpop/ggCaller.

11. Scikit-learn library: A Python module integrating a wide range of state-of-the-art machine learning algorithms.
    Accessed on 12 March, at https://scikit-learn.org/stable/modules/impute.html.

12. Surprise: A Python scikit for building and analyzing recommender systems that deal with explicit rating data.
    Accessed on 12 March, at https://surprise.readthedocs.io/en/stable/matrix$_{f}actorization.html$.

13. CELEBRIMBOR's GitHub repository: pipeline to create pangenomes from metagenome assembled genomes (MAGs), using completeness information to adjust observed frequencies.
    Accessed on 16 May, at CELEBRIMBOR.

14. Nextflow: A workflow manager that eases the writing of data-driven pipelines.
Accessed on 26 March, at https://www.nextflow.io/.

15. Lees Group GitHub repository: Repository hosting projects related to bacterial population genomics.
Accessed on 4 June, at https://github.com/bacpop.

16. Finn Group GitHub repository: Repository hosting projects related to Computational metagenomics and analysis.
Accessed on 4 June, at https://github.com/bacpop.

SUMMARY

Metagenomics and pangenomics offer profound insights into the genetic diversity and functional potential of microbial communities, yet they face significant challenges, particularly due to the incompleteness of metagenome-assembled genomes (MAGs). The MAGGImpute workflow, leveraging the advanced gene-calling capabilities of ggCaller, provides a robust solution to these challenges. By accurately imputing missing genes, MAGGImpute enhances the reliability of genomic annotations, ensuring comprehensive pangenomic analyses. This work builds upon and extends the foundational methodologies established by the state-of-the-art tool Celebrimbor, paving the way for more precise and scalable metagenomic research.

Keywords: *MAGs, Bacterial pangenomics, Workflow, Benchmarking, Imputation methods, Gene-Graph caller, FAIR.*

Résumé

La métagénomique et la pangénomique offrent des perspectives profondes sur la diversité génétique et le potentiel fonctionnel des communautés microbiennes, mais elles sont confrontées à des défis significatifs, notamment en raison de l'incomplétude des génomes assemblés par métagénome (MAGs). Le flux de travail MAGGImpute, tirant parti des capacités avancées d'appel de gènes de ggCaller, fournit une solution robuste à ces défis. En imputant avec précision les gènes manquants, MAGGImpute améliore la fiabilité des annotations génomiques, assurant des analyses pangénomiques complètes. Ce travail s'appuie sur les méthodologies fondamentales établies par l'outil de pointe Celebrimbor, ouvrant la voie à une recherche métagénomique plus précise et évolutive.

Mots-clés : *MAGs, Pangénomique bactérienne, Flux de travail, Benchmarking, Méthodes d'imputation, Gene-Graph caller, FAIR.*