UNIVERSIDAD
DE ANTIOQUIA

DATMA: Distributed AuTomatic Metagenomic Assembly and Annotation framework

Autor

Bernardo Andrés Benavides Arévalo

Universidad de Antioquia

Facultad de Ingeniería, Doctorado en Ingeniería Electrónica

Medellín, Colombia

2019

cc BY NC ND

DATMA: Distributed AuTomatic Metagenomic Assembly and Annotation framework

Bernardo Andrés Benavides Arévalo

Tesis Doctoral
como requisito para optar al título de:
Doctor en Ingeniería Electrónica

Asesor

Ph.D. Felipe Cabarcas Jaramillo

Universidad de Antioquia
Facultad de Ingeniería.
Medellín, Colombia
2019.

Declaration of Authorship

I, Bernardo Andrés BENAVIDES ARÉVALO, declare that this thesis titled "DATMA: Distributed AuTomatic Metagenomic Assembly and Annotation framework" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Abstract

Bacterial populations have colonized almost every possible niche on Earth, including those considered harsh for most organisms. These extreme physical conditions make it hard to get genetic information from the organism community. Next-generation sequencing has provided a large amount of DNA data that can be used by researchers to study environmental samples using culture-independent shotgun metagenomic experiments. Metagenomics has made it possible to explore the large variety of microorganisms present in many complex ecosystems, like soils, oceans, biosolids, hot springs, among others. Moreover, it has allowed the identification of novel bacterial and archaeal species, generating complete or near-complete genomes. It has helped filling blind spots into underrepresented or missed taxonomical clades.

One of the main challenges in the metagenomic analysis is the assembly process. Microbial communities are complex, bacteria have different genome size and abundances, some regions of their genome are very similar, and metagenomic sequencing results in a mixture of reads from the several microorganisms present in the community. Despite the development of dozens of implementations for *de novo* assembly for metagenomics, they have not eliminated the high risk of assembling reads from different organisms as a single chromosome, which creates chimeric molecules. One alternative to address this is to separate reads in groups (binning) before the assembly process. Given that most assemblers consider that the reads belong to a single species, by grouping highly similar reads in bins, the assembly complexity and the probability of creating chimeric contigs are significantly reduced.

In this dissertation, we introduce a binning strategy to group reads from the same molecule into the single bin. We named our method CLAME. We showed that CLAME decreases the complexity of metagenome, and allows recovering almost complete bacterial genomes. We also introduce DATMA, an integration of CLAME into a distributed workflow for metagenomics analysis. DATMA is a pipeline for fast metagenomic analysis that orchestrates the following: sequencing quality control, 16SrRNA-identification, reads binning, *de novo* assembly and evaluation, gene prediction, and taxonomic annotation.

We show CLAME and DATMA functionality analyzing complex metagenomes and recovered from them most of its species and, more important DATMA automatically extracted an almost complete genome from the predominant species.

Acknowledgements

First and foremost, I would like to thank my thesis director, Ph.D. Felipe Cabarcas Jaramillo, and to the professor Juan Fernando Alzate, for their continuous support and guidance during these last years. Without Felipe and Juan, this thesis would not have been possible. I also want to thank my family and friends for their accompany and support along this journey.

I want to express my profound gratitude to SISTEMIC members and CNSG researchers by their interest in my thesis work. I am grateful for their constructive suggestions and recommendations for this thesis.

This study was supported by Colombia government department COLCIENCIAS Scholarship Program No. 6172, and the research groups: SISTEMIC belongs the Universidad de Antioquia; Medellín-Colombia, and CNSG from the Universidad de Antioquia; Medellín-Colombia.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The world is dominated by microorganisms that, although we cannot see, are an essential part of all biomes on Earth. They contribute with the photosynthesis, help to produce nutrients for plants and animals. Many of them are used to create pharmaceutical drugs, enzymes, and other bioactive compounds. Moreover, the billions of microorganisms that live in the human gut help us to digest food, break down toxins and fight off disease-caused by others microbes. Unfortunately, it is hard to obtain information about their genetic composition because most of them cannot be cultivated [89], [102], [114].

One alternative to study these microorganisms is to use their deoxyribonucleic acid (DNA) to identify and classify them directly from an environmental sample. Next-Generation Sequencing (NGS) platforms can sequence DNA from environmental samples without the need for isolating the species. These experiments are called metagenomics, and they allow the study of microorganisms without the need for prior cultivation.

Thanks to metagenomics, complex ecosystems like soil, seawater, biosolids, etc., have been studied (e.g. [37], [66], [10]). It has been possible to report microorganisms genomes from these environments (e.g. [86], [38]). Since metagenomic NGS approaches generate millions of short DNA reads from large genomes, one of the main challenges is to reconstruct genome or genomes from these pieces. This thesis proposes a technique to address this challenge.

1.1 Computational Challenges in Metagenomics

The primary challenge in metagenomics is to characterize the taxonomic diversity of microbial communities. Several review papers (i.e., [121], [32]) describe all the challenges present in a complete metagenomic analysis. In this work, we address the problem of assembling metagenomic reads and describe the main computational challenges of metagenomics.

1.1.1 Metagenome Assembly

We focus this thesis on current DNA sequencing technology that reads short sequences of DNA bases (typically 150-1000 base-pairs.) It is the sequencing technology that dominates the market. Short-read sequencing means that genomes in the sample are highly fragmented, and the challenge is to recover them using these small fragments. Oxford Nanopore technologies [111] are developing strand sequencing, a method for DNA analysis that could potentially sequence completely intact DNA strands/polymers passed through a protein nanopore. To date, however, the use of such technologies in metagenomic settings has been limited because of the complex sample processing requirements, their error rate, and cost.

Metagenome assembly is complicated since the number of species and strains and their relative abundance is unknown. Furthermore, we are interested in cases in which mapping reads to a reference genome is not possible (because most species are still unknown) and metagenomic assembly is accomplished *de novo* by reconstructing genomes directly from the information of overlapping reads. Despite the development of dozens of implementations for *de novo* assembly for metagenomics (e.g., MetaVelvet [73] and metaSPAdes [78]), they have not eliminated the high risk of assembling reads from different organisms as a single chromosome, which creates chimeric molecules [105]. In our experiments, for example, their performance does not generate the expected results, probably because of the complexity of our samples.

1.1.2 Metagenomic Binning

Since most assemblers (i.e. [125], [77]) consider that the reads belong to a single species, the assembly complexity and the probability of creating chimeric contigs can be significantly reduced by grouping highly similar reads in bins. The problem is that the classification of sequences within a metagenomic dataset is very challenging, mainly when the experiment includes unknown microorganisms that lack genomic reference. Moreover, the shotgun process makes that the genomes present in the metagenome result fragmented in millions of short sequences, making it difficult to find a biological feature that allows binning them. While dozens of supervised and unsupervised binning methods (e.g. [122], [118], [123]) are available, there is still room for improvement.

1.1.3 Computational requirements in metagenomics

The vast amount of information of DNA provided by next-generation sequencing brings enormous challenges referred to data processing, storage, management, and interpretation. It may require distributed algorithms, new compressing methods, and sophisticated store strategies that allow processing, save and access to this information in reasonable time and memory. Therefore, an essential challenge in metagenomic studies is to build efficient and robust computational tools that can deal with the massive amount of sequence data and obtain accurate microbial identification of hundreds or thousands of species in a reasonable time and memory consuming.

1.2 Problem Statement

A primary objective in metagenomics is to classify DNA sequence fragments based on their DNA molecule precedence. This task, known as binning, is challenging for the following reasons. On the one hand, most organisms on an environmental sample lack taxonomically related sequences in existing reference databases, since around 99% of bacteria found in environmental samples have not been sequenced. Consequently, binning methods usually fail to align with confidence the metagenomics reads against a reference dataset. On the other hand, current sequencing technology generates reads whose average length vary between 100 to 1000 pair bases, depending on the sequencing platform used. Hence, binning methods suffer from a lack of resolution due to insufficient phylogenetic information in each read. Although several algorithms and tools perform binning, they are not accurate when the data size increases or the biodiversity of the sample is different of their assumed models, and therefore the challenge of binning metagenomic reads is still an open problem.

1.3 Key Contributions

In this section, we highlight the key contributions of this dissertation.

- A methodology for binning metagenomic reads: We developed a new binning method that groups metagenomic reads in bins using their biological and shotgun sequencing properties without the need of a reference genome. We implemented this methodology in a program named CLAME.
- CLAME software: CLAME, from the Spanish words "CLAsificador MEtagenomico," is a C++ program that bins DNA sequences using a graph representation of the metagenome dataset. We compared CLAMEs performance, and speed to bin metagenomic reads against different states of the art binning programs and demonstrated that CLAME can group most reads from the same molecule faster than the other tools and in many cases better.
- A flexible pipeline for metagenomic data analysis: We designed a Distributed AuTomatic Metagenomic Assembly and Annotation framework (DATMA). It

is a bioinformatics tool that can be used to study complex metagenome in an automated fashion using multiple computing resources. Using DATMA, we analyzed several metagenomes and proposed two novel draft genomes.

- Xanthomonadaceae_UdeA_SF1 draft genome: We recovered a high-quality draft genome reconstructed from a Colombian's Andes hot spring metagenome. The genome seems to be from a new lineage within the family Rhodanobacteraceae of the class Gammaproteobacteria, closely related to the genus Dokdonella. This draft genome is available on the NCBI project PRJNA431299.
- Anaerolineaceae_UdeA_SF1 draft genome: We used DATMA to study the San Fernando biosolid metagenome. DATMA allowed us to recover an Anaerolineaceae draft genome. Genome annotation shows that the draft genome seems to be close to the family Anaerolineaceae and it has a relation with the genus Pelolinea and Leptolinea. This low-quality draft genome is available on the NCBI project PRJNA529916.

1.4 Outline

The remainder of this thesis is organized as follows: Chapter 2 gives the theoretical background of DNA sequencing, metagenomics significance, and an overview of representative metagenomic projects and typical analysis pipelines. Then, in Chapter 3, we introduce CLAME, a new alignment-based binning algorithm. We show its computational performance, limitations, and its strategy to address computational challenges related to downstream analysis. Next, we introduce DATMA a Distributed Automatic Assembly and Annotation Tool for Metagenomics in Chapter 4. Then, in Chapter 5, we present our experimental setup and results. We conclude in Chapter 6 and outline future work in Chapter 7.

1.5 Related Publications and Software Development

The work in this dissertation has resulted in several publications and software tools.

Chapter 3

1. Benavides A, Alzate JF, Cabarcas F. Using graph theory for metagenomic binning. III Congreso Colombiano de Biología Computacional y Bioinformática (CCBCOL-2015), September 2015.
2. Benavides A, Isaza JP, Niño-garcía JP, Alzate JF, Cabarcas F. CLAME: a new alignment-based binning algorithm allows the genomic description of a novel Xanthomonadaceae from the Colombian Andes. BMC Genomics; 2018;122; doi:10.1186/s12864-018-5191-y.

3. Jaime Lotero, Andres Benavides, Anibal Guerra, Sebastian Isaza. UdeAlignC: Fast Alignment for the Compression of DNA Reads. IEEE COLOMBIAN CONFERENCE ON COMMUNICATIONS AND COMPUTING (COLCOM 2018). May 2018.

Chapter 4

1. Benavides A, Sanchez F, Alzate J.F and Cabarcas F. DATMA: Distributed AuTomatic Metagenomic Assembly and Annotation framework. PeerJ Journals. Submitted with Corrections May 2019 (Peer-reviewed, Corrections).

Chapter 5

1. Benavides A, Bedoya K, Alzate JF, Cabarcas F. ATMA: A data analysis tool for metagenomics allows recovering an Anaerolineaceae draft genome from the San Fernando biosolid. XIII Latin American Workshop and Symposium on Anaerobic Digestion (DAAL 2018). October 2018.

Chapter 2

Background

2.1 Deoxyribonucleic acid

Deoxyribonucleic acid (DNA) is the molecule that contains the instructions for the functions and development of all the cells of living organisms. It is formed by the union of nucleotides, which are composed of a monosaccharide sugar, a phosphate group and a nitrogen base that can be guanine (G), adenine (A), thymine (T), or cytosine (C). The number of nitrogen bases and their order is what differentiates each organism on earth. A string of these bases forms the complete chain of DNA (e.g., Human DNA consists of about 3×10^9 base pairs).

2.1.1 Sequencing

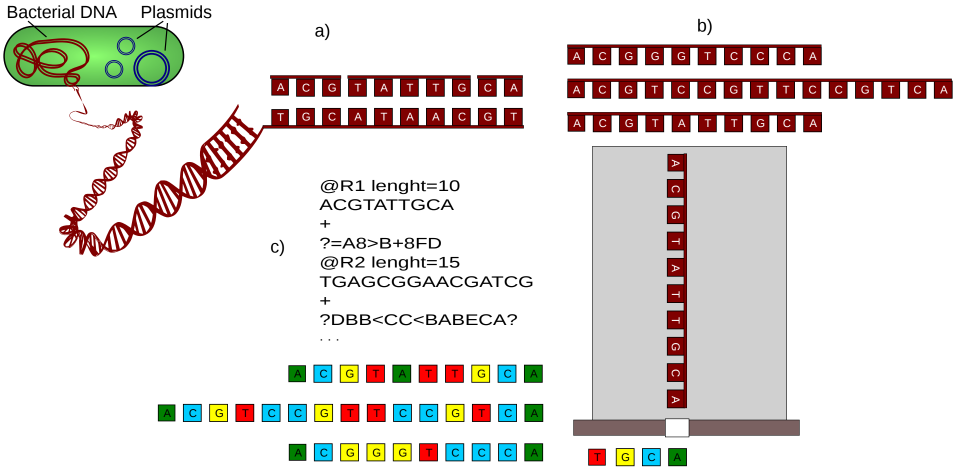
DNA sequencing is the process of determining the precise order of nucleotides within a DNA molecule. Sanger and Coulson described plus and minus the first sequencing method. Sangers approach [97] is considered the first-generation technology, and it is the base for the Next-Generation Sequencing (NGS) technologies such as Roche [94], Illumina [45], Pacific Bioscience (PacBio) [82], Ion Torrent [107], Oxford-Nanopore [81] among others. Table 2.1 summarizes the main features of these NGS platforms.

Current NGS technologies provide only fragmented DNA rather than the whole chain. Such fragments are called reads, and the FASTQ format [19] is a de facto standard for storing such DNA reads. FASTQ files store the DNA chains like strings of ASCII characters that represent the DNA bases. Each record contains the name the read, the bases sequence, and quality scores for each one of them. Quality scores tell us the level of confidence of every base identified by the sequencing machine. The $'N'$ character represents unidentified bases. Figure 2.1 illustrates the DNA structure, the sequencing process, and the FASTQ format.

TABLE 2.1: Features of some next-generation sequencing platforms

Platform	Model	Read Length (bp)	Run Time	Output per run	Costs per Mb (US$)	Reported Problems
Roche/454 GS-FLX Titanium	2008	∼400	1 day	∼400 Mb	12	Homopolymer stretches
Illumina Hiseq 3000/400	2015	150	4 days	750 1500 Gb	NA	GGCxG motifs
PacBio RS II	2013	∼15000	4 hours	1 Gb per SMRT cell*	0.6	Quite high random error rate
Ion Torrent PGM	2012	400	4 hours	>1Gb	1	Homopolymer stretches
Oxford nanopore MiniON	2018	>10000	NA	NA	NA	High random error rates

*Single-molecule real-time (SMRT)



```
@R1 lenght=10
ACGTATTGCA
+
?=A8>B+8FD
@R2 lenght=15
TGAGCGGAACGATCG
+
?DBB<CC<BABECA?
. . .
```

FIGURE 2.1: DNA and sequencing process. a)DNA structure, b)sequencing process, c)FASTQ representation

2.2 Metagenomics and Related Projects

In genomics, when the sequencing subject is not from an individual organism previously isolated, but from a microbial community, it is called a metagenomic experiment. Metagenomics allows the direct genetic analysis of genomes contained within environmental samples without the prior need for cultivating. The goal of a metagenomic project is usually to address the questions of who is present in an ecological community and what they are doing. According to the aims and the information to get two kinds of experiments can be conducted: target metagenomic and whole-genome projects.

2.2.1 Target metagenomic projects

Targeted metagenomic experiments are limited to sequencing a particular maker rather than the whole DNA chain. Most of these projects (e.g., [103], [100]), use the

16S ribosomal RNA gene marker, to obtain a community/taxonomic distribution profile. 16S rRNA gene is a well-conserved sequence that exists in most microbial genomes, specifically bacteria, and archaea, and allows identification of microbes within different taxonomic groups in a complex community.

2.2.2 Whole genome projects

Whole genome-based approaches are not limited by sequence conservation or primer binding and allow the sequencing most genomes within an environmental sample. Full shotgun metagenomics enables scientists to identify and annotate diverse arrays of microbial genes that encode many biochemical or metabolic functions.

This dissertation is about whole genome projects rather than target metagenomic projects. Therefore we will refer to the whole-metagenomic project as metagenomic analysis or simply metagenomics.

2.3 Overview of Metagenomic Analysis

Most next-generation sequencing metagenomic experiments (see Figure 2.2) require: remove low-quality bases, bin reads (optional), and assemble reads into molecules (contigs) to assign them a taxonomic classification using reference databases or predict Open reading frames (ORFs). Tools like Trimmomatic [12], SolexaQA [21] (quality control tools), Velvet [125], MetaVelvet [73], SPAdes [77], metaSPAdes [78], (assembly tools), CLARK [80], Kaiju [65] (annotation tools), Prodigal [44], GeneMark [9] (gene prediction tools), among others, can be used to address these tasks. Many of them have been integrated into full pipelines like MetAMOS [109], RAST server (MG-RAST) [120], IMG/M server [17], MetaWRAP [113], SqueezeMeta [105], MetaMeta [90], and MOCAT [54]. These pipelines allow processing a metagenomic dataset automatically. But currently, there is not a standard tool designed to study a metagenomics dataset and the design of accurate algorithms and tools is an open field of research.

2.4 Metagenomic Binning

Although each part of the analysis of metagenomic data is crucial and complex, characterizing the taxonomic diversity of microbial communities is one of the primary objectives of metagenomic studies. This objective, called binning, is essential because an accurate classification helps assemble, annotate the reads, and even better gene cluster. Binning methods can be categorized, based on the methodology and final objective, as taxonomy-dependent or taxonomy-independent (Figure 2.3).

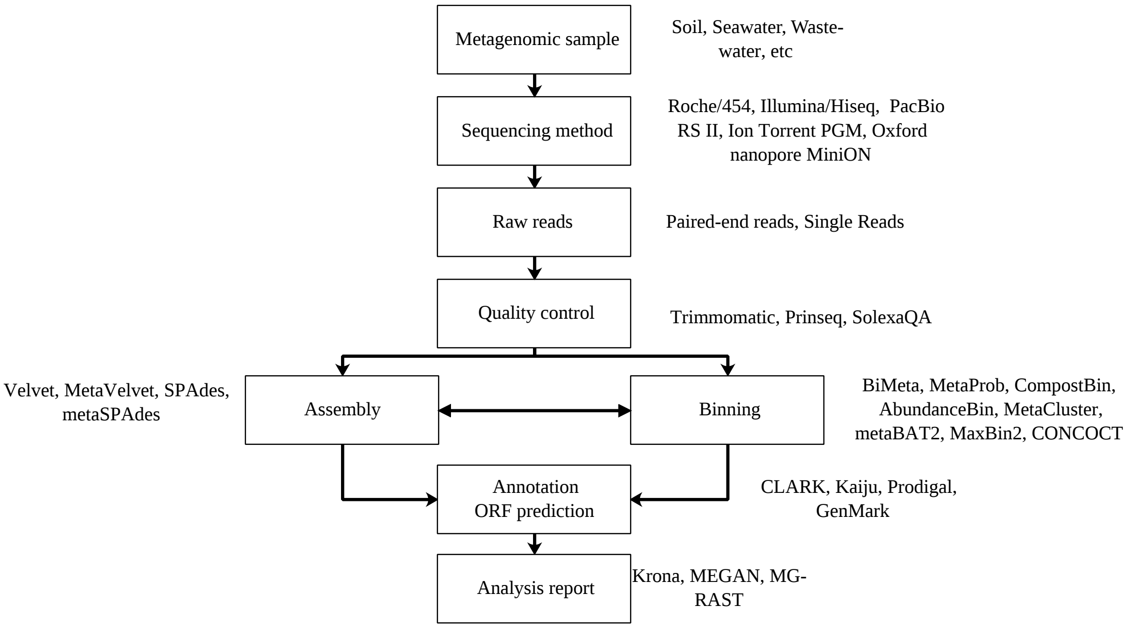

Metagenomic sample
Soil, Seawater, Waste-water, etc
Sequencing method
Roche/454, Illumina/Hiseq, PacBio RS II, Ion Torrent PGM, Oxford nanopore MiniON
Raw reads
Paired-end reads, Single Reads
Quality control
Trimmomatic, Prinseq, SolexaQA
Velvet, MetaVelvet, SPAdes, metaSPAdes
Assembly
Binning
BiMeta, MetaProb, CompostBin, AbundanceBin, MetaCluster, metaBAT2, MaxBin2, CONCOCT
Annotation
ORF prediction
CLARK, Kaiju, Prodigal, GenMark
Analysis report
Krona, MEGAN, MG-RAST

FIGURE 2.2: Typical workflow and tools for metagenomics

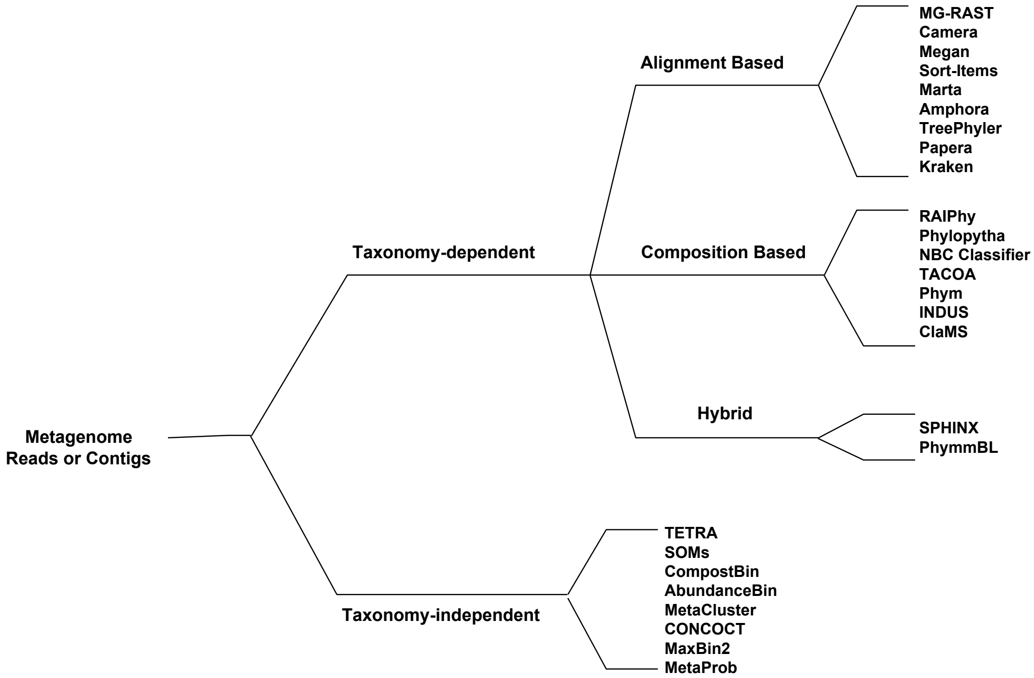


FIGURE 2.3: Taxonomy-dependent and Taxonomy-independent binning methods

2.4.1 Taxonomy-dependent binning methods

Taxonomy-dependent methods involve supervised learning procedures. They classify reads by comparing them against sequences in reference databases, or pre-computed models. Reads that classify under a similar taxonomic category conform a bin. The accuracy of the classification depends on obtaining enough levels of similarity, between reads and sequences/models in the reference databases. According to the methodology used, Taxonomy-dependent methods are subdivided into alignment-based and composition-based methods.

2.4.1.1 Alignment-based binning methods:

Alignment-based tools such as Megan [43], MG-RAST [120], Camera [99], MetaBinG [46], align reads to sequences from a database like NCBI [74], PFAM [25], UniProt [112], EMBL [27], or DDBJ [11]. Most of them use Basic Local Alignment Search Tool BLAST [2] to calculate a similarity metric, called bit-score, which is then used to assign reads into specific bins. Other tools like SOrt-ITEMS [69], MetaPhyler [61], MARTA [41], combine the bit-score with other alignment parameters, like the percentage of identities, positives, and gaps penalties, to improve the classification and avoid incorrect assignments. A limitation of the BLAST-based approaches is that they require a huge compute power for aligning millions of reads against a large number of sequences belong to reference databases.

To reduce the computation time, tools like AMPHORA2 [122] and WebCARMA [31], compare only regions of the genome against pre-built markers. AMPHORA2, for example, uses 31 bacterial protein-markers and 104 archaea genes, while WebCARMA uses protein conversation regions reported in the PFAM database. Both tools generate a phylogenetic tree based on Hidden Markov Models (HMM). These approaches are faster than BLAST strategies; however, they have problems when they classify reads from species far from the prebuild models.

2.4.1.2 Composition-based binning methods:

Tools like PhyloPythiaS+ [34], NBC-classifier [95], TACOA [24], and RAIphy [72] use reads properties (i.e., Guanine-Cytosine GC-percentage, codon usage, oligonucleotide usage) to classify them into a specific group. These tools use Support Vector Machines (SVMs), Naive-Bayesian models or Markovian properties, to store the compositional properties.

In an initial step, these methods build a specific model based on one or more compositional properties of known genomes. This phase is usually executed once, but comprising a high computational cost, which can increase quickly if the model needs to be retrained. Moreover, composition-based binning methods assume that a single compositional model can represent all the genomes complexity. However, specific genomes are characterized by distinct regions of heterogeneity as compared to the rest of the genome [115]. Therefore, these methods usually generate a high number of false positives.

2.4.1.3 Hybrid binning methods:

Tools like PhymmBL [14], and SPHINX [68] are Hybrid binning methods that use the advantages of alignment-based methods and composition-based methods to improve the classification. SPHINX, in its first phase, uses tetra-nucleotide frequency propriety to compare the structure of a given read and then uses SOrt-ITEMS strategy to align them to a reference sequence. PhymmBL combines the composition-based methodology of Phymm along with BLAST to improve the confidence of taxonomic assignments. However, the computational time and a large number of false positives are drawbacks for these methods.

2.4.2 Taxonomy-independent methods

Taxonomy-independent methods group reads in a given dataset based on their mutual genetic similarity and do not involve a database comparison step. They determine the distribution of each species in the sample by observing the frequency of k-bases in the query sequence. Methods under this category include BiMeta [116], MetaProb [33], TETRA [106], CompostBin [16], AbundanceBin [124], and MetaCluster 5.0 [118], MaxBin2 [123], CONCOCT [1].

BiMeta first bins DNA sequences according to the overlap information between them. Then it merges the groups by using an observation on the l-mer frequency distribution of the sets of non-overlapping reads. AbundanceBin models the number of reads of different species using Poisson distributions to avoid generating a high number of bins. Later it groups reads with similar abundance levels. This kind of methods works efficiently with samples having high abundance levels variation, but its binning efficiency decreases with metagenomes having species with similar abundance distribution.

To improve resolution in the dataset, some tools bin contigs rather the raw reads. For example, MetaProb uses assembled contigs to compute l-mer frequencies and generate probabilistic sequence signatures. Then it bins contigs with the same signature into the same group. CONCOCT applies Gaussian mixture models (GMMs) and Bayesian information criteria (BIC) to cluster contigs into groups based on sequence composition (kmer frequencies) and coverage across multiple samples. MaxBin2 employs coassembling sequencing reads of various metagenomic datasets. It first measures the tetranucleotide frequencies of the contigs and their coverages for all involved metagenomes. Then it classifies contigs into individual bins according to an ExpectationMaximization (EM) algorithm. Since contig-based binning methods require a previous assembly, they can propagate the error generated in this stage. Moreover, some metagenomes are too complicated that it is not possible to assemble all the reads without an initial binning step.

The focus of our work was to develop an unsupervised reads-base binning method for metagenomics that works accurately for shotgun DNA reads. In this study, we compared our approach against MetaProb [33], BiMeta [116], AbundanceBin [124] and MetaBinG [46] tools. We also analyzed the results generated by binning the reads and assembling every bin against the results generated by the contig-based binning methods. We used metaBAT2 [49], MaxBin2 [123], and CONCOCT [1] tools. We selected the reference programs from several factors: whether they are actively maintained, how recently they were published, and whether another program has superseded them.

Chapter 3

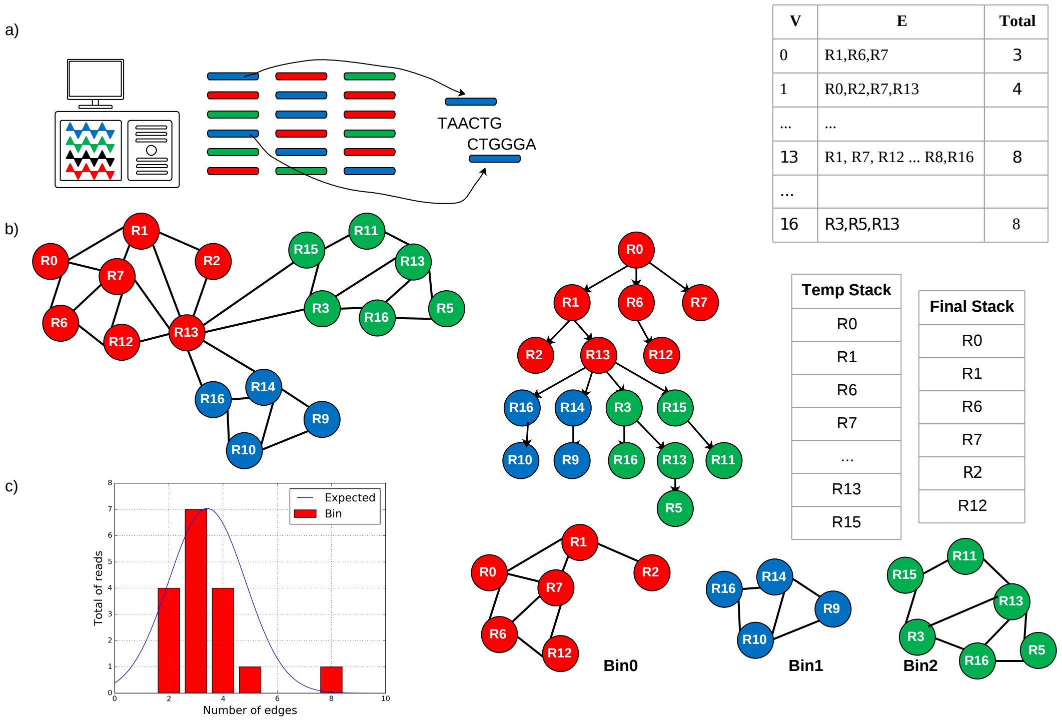
CLAME, A binning tool for metagenomics

In this Chapter, we introduce CLAME, (from the Spanish words: CLAsificador para MEtagenómica), a new binning method that groups metagenomic reads in bins using their biological and shotgun sequencing properties. The fundamental idea of CLAME is that exact matches, of a large number of bases, between reads, is very unlikely if they do not come from the same DNA molecule. Furthermore, assuming that in a metagenome there is at least one genome sufficiently covered, and given that the sequencing errors is low (on platforms like Illumina Miseq or Roches 454), most sequences from a DNA chromosome will have exact matches between them. Moreover, we have observed that reads from conserved regions tend to align several times with other sequences, and reads with sequencing errors or chimeras, tend to align few times with other reads. It allows discriminating them and avoids merging regions belong to different genomes.

We performed a set of experiments to evaluate the classification performance of CLAME on several datasets with different complexities. We also assessed state-of-art binning tools and compared our results against these tools. The results show that our approach consistently outperforms other binning tools like MetaProb [33], BiMeta [116], AbundanceBin [124], and MetaBinG [46]. We also compared their computational performance and showed that CLAME bins metagenomic reads faster than the other tools.

3.1 Methods

CLAME starts by aligning each sequence against all the other reads, looking for reads with similar DNA composition (Read alignment stage). It creates a graph representation, G= (V, E), of the metagenome in which reads are nodes (v ∈V), and the overlapping relation between each pair is an edge between them (e ∈E). Ideally, two reads from different DNA chromosomes will not align together, at least not in a considerable number of bases, and thus, the graph will represent the different organisms or chromosomes as organized subgraphs. The binning will, therefore, follow naturally by traversing the graph, creating a bin for each connected subgraph (Subgraph traversal and bin generation stage). However, conserved regions, such as the ribosomal RNA genes, may generate edges between reads with different species memberships. Using various experiments, we showed that these cases can be analyzed as outliers into a normal distribution. Consequently, CLAME uses the median absolute deviation (MAD) statistic metric to rate the sequence abundance, redefine the graph, and produce the final bins (Edge analysis stage). We show CLAME methodology in Figure 3.1 and explain each stage in the following subsections. Appendix A shows the pseudocode algorithms for each function of CLAME.



V	E	Total
0	R1,R6,R7	3
1	R0,R2,R7,R13	4
...	...	
13	R1, R7, R12 ... R8,R16	8
...		
16	R3,R5,R13	8

Temp Stack
R0
R1
R6
R7
...
R13
R15

Final Stack
R0
R1
R6
R7
R2
R12

FIGURE 3.1: CLAME methodology. a) Read alignment stage: The metagenome is composed of reads from different genomes (red, blue, and green blocks); each read, is aligned against all the reads. b) Subgraph traversal and bin generation: An adjacency list represents a graph G=(V, E), where each vertex v in V denotes a read and each edge e in E indicates that two reads align in at least b bases. The bins are generated by traversing the graph and reporting each subgraph into a temporal stack (e.g., R0, R1, R6, R13 ... R15). c) Edge analysis stage: Reads that belong to a shared region can connect the subgroups (i.e., R13 from red group aligns with the R3 and R15 from the green group). These connections usually make that the number-of-edges histogram departs from a normal like form. Edges analysis removes sequences with extreme values (i.e., R13), and report the final bin (e.g., R0, R1, R6, R7, R2, and R12). CLAME traverses the graph several times until grouping all the reads.

3.1.1 Read alignment stage

Read alignment stage computes the alignment of all versus all the reads and creates the edges of the graph (see Figure 3.1(a)). Algorithms like Needleman-Wush [75] and Smith-Waterman [101] were designed to find the optimal local alignment. These algorithms compute the best alignment by accepting insertions and deletions into each base pairs. It is an intense-computational task, which has been tackled by many researchers (e.g. [71], [104]). Since, these algorithms require $O(n^2)$ computational time, where n is the number of bases of the reads, they are very slow for big datasets.

Hatem et al. [40] performed a comprehensive review of the most relevant single-threaded tools for short sequence alignment. They focused on the analysis of the performance-sensitivity trade-off, (number of sequences rightly aligned VS speed). The study concludes that fast aligners follow the Seed and Extend strategy [2]. This methodology first produces a reference structure from the target dataset (e.g., the entire human genome). Then it extracts a small substring from the query sequence (e.g., the first b bases of a read) and searches it across the reference to find an exact coincidence (Seed stage). If a match exists, the whole read is then aligned (Extend stage) using an algorithm that supports mismatches, insertions, and deletions, and starting at the reported position in the reference. If the Seed stage found more than one coincidences, the Extend process is developed for all them, and the best matching is reported. If there is not any coincidence the read is not extended, and it is reported as unaligned.

Although an optimal solution is right for several genomic and metagenomic tasks, it is not necessary to identify if two reads have similar DNA composition. Note that if two sequences share enough region (we select a suitable seed size), the Seed-search stage is enough to detect some possible alignment. CLAME takes advantage of this fact and improves the execution time over the tools reviewed by only implementing Seed-search strategy. We summarize this methodology in the next paragraphs and invite the reader to consult our UdeAlignC tool [63]. It is a fast alignment algorithm that implements the complete version of Seed and Extended approach. We demonstrated that UdeAlignC algorithm is 2x faster than the state of the art tools while precision (measured as the number of sequences rightly aligned) is only reduced by 5.6%. We also showed that the GPU-accelerated version has a speedup of up to 12x compared with the sequential version.

3.1.1.1 Seed-search strategy

The seed-search strategy is traditionally developed using a suffix-trie, which represents all the substrings of a reference text *S* into a tree graph (e.g., *S*= ACAAACATAT in Figure 3.2). This tree allows that a query text or a substring *Q* from it can be searched by means a backward search (e.g., Q= ACAA in Fig 3.2). Backward search traverses the suffix-trie, starting from the root (indicated by the symbol \$), matching successive symbols of the query, with the leaves (nodes) on the tree. If the length of the path is equal to the size of Q ($|Q|$), it means that the substring Q occurs in the text S.

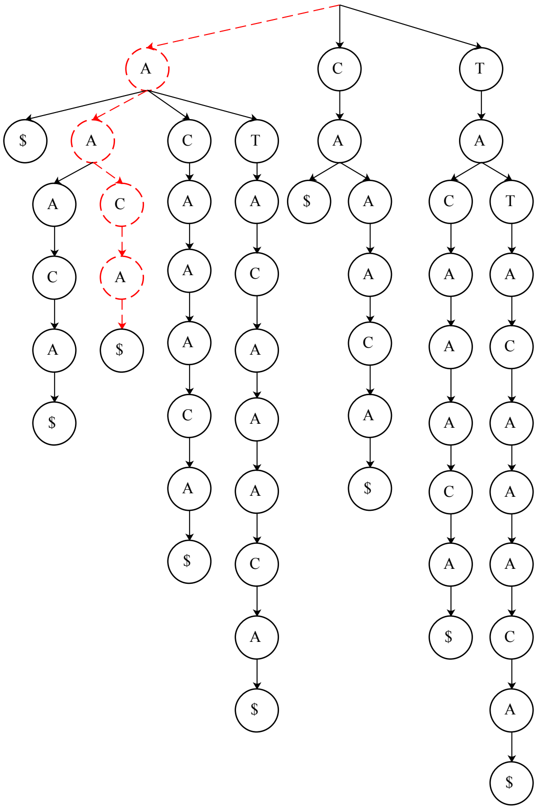


FIGURE 3.2: Suffix trie for the text S = ACAAACATAT and the Backward search for Q =ACAA. Read path indicates the exact overlap

3.1.1.2 FM-index

Backward search requires a Suffix-trie representation of the reference text. The most popular Suffix-trie based aligners (i.e., Bowtie 2 [57] and BWA [60] tools) use an FM-index tree [29]. Paolo Ferragina and Giovanni Manzini designed this data structure. They showed that this representation allows searching a query text of size Q in a reference text using $O(|Q|)$ time and considerable few memory. Central to the FM-index is the Burrows-Wheeler transform (BWT) generated from a Suffix array (SA). In Appendix C, we illustrate the complete construction of an FM-index and the formal backward search strategy to detect overlaps using metagenomic sequences. Next, we explain CLAME's Read-alignment stage utilizing a state of the art library that produces an FM-index and enables substring queries.

3.1.1.3 Succinct Data Structure Library (SDSL)

There is a set of open-source versions of the FM-index algorithm available in public repositories (e.g., [88], [70], [26]). CLAME uses the Succinct Data Structure Library (SDSL) [96]. The authors have demonstrated that, in contrast with proposed implementations in literature, SDSL Library provides high quality, efficient construction, and excellent run-time performance.

SDSL library provides more than 40 data-structures and algorithms implemented into flexible C++ templates that offer a set of efficient methods for storing, traversing, and seeking information inside such structures. We performed a benchmark measuring index size and search times over the set of data structures and algorithms offered by the library. We used a human genome as a reference to build several FM-index. Then we queried a set of reads, taken randomly from the genome, on each three. We summarized the result in Figure 3.3 and showed the complete description in the UdeAlignC report [63]. We found that a suffix array with a sample density of 8 bits, stored in a Huffman Wavelet Tree [36] (the yellow line in the figure), produced the best results. Consequently, we selected this structure in CLAME.

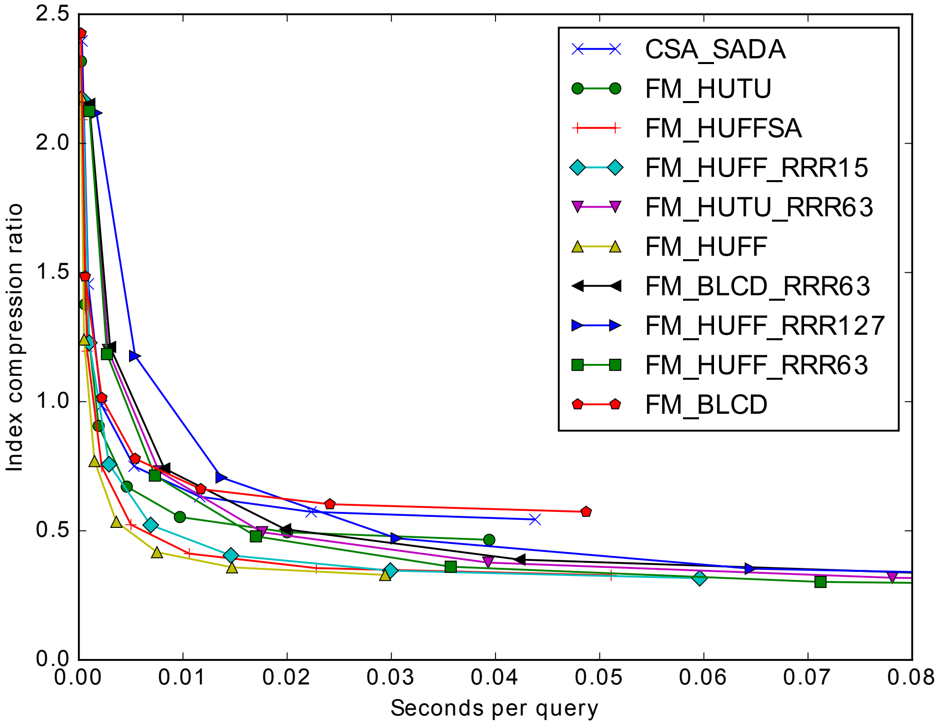


FIGURE 3.3: Results of different data-structures benchmarks from SDSL library applied to genomic information. The vertical axis represents the size of the original genome divided by the size of the index, and the horizontal axis shows query latencies on each data-structure. We varied the density of the saved index across the plot.

3.1.1.4 Seed-search strategy implementation

CLAME supports DNA-sequences files in FASTA and FASTQ formats. To build an FM-index using these metagenomic reads, CLAME produces a long text by concatenating the bases from the raw reads; it includes the symbol & to separate the bases from one sequence of another. It also avoids that a query search can be wrongly reported by the alignment between the beginning and end of two different reads. Read alignment stage reports only exact matches.

The concatenated sequences generate the text (S) that is the argument of the *genFM9* function of CLAME, which produces the FM-index representation of the raw reads. Later, CLAME calls the *map2FM9* function that implements the backward strategy, to align each sequence against the entire dataset. To reduce the computational time, CLAME uses the first, and last b bases of each read in forward and reverse complement. The parameter b is the number-of-bases threshold defined by the user and represents the seed size. Although CLAME only uses queries of b-size, since the FM-index contains all suffixes for each read, the alignment is checked on the entire length of the target sequence.

FM-index representation of all reads allows that each query sequence can be processed individually using the backward search process. CLAME uses the Open Multi-Processing Programming Model (OpenMP) [52] to distribute one each query search per thread and speedup the alignment process.

CLAME uses a Key-Mapped structure to save the reads alignments, where the number of the sequence corresponds the key-value, and a list with the overlaps is the map-value (see Table 3.1). It requires $O(np)$ space of memory, where n is the number of reads, and p is the maximal number of alignments per sequence. The worst-case occurs when p=n, for all the reads, it implies $O(n^2)$ space of memory, which is a constraint for large datasets or computers with low memory capacity. Subsequently, the number-of-bases threshold (b) plays an important role. In the experimental section, we show that a low b-value generates a high number of alignments and increase the memory consume. We recommended starting with a considerable b-value (70bp is the default) and then iterate with minor values. We illustrate this methodology in Chapter 4.

3.1.2 Subgraph traversal and bin generation

The MatrixQuery matrix, generated in the Read alignment stage, is a graph representation, in an adjacency list format, of the relation of the reads. For example, the MatrixQuery in Table 3.1 shows that the read *R0* aligns with the reads *R1*, *R6*, and *R7*. CLAME traverses the graph using a greedy breadth-first search strategy [87]. It employs two vectors: the query vector (Qv) and the Stack vector (Sv), both of size

TABLE 3.1: MatrixQuery container. Key-value represents the sequences, and Map-value represents the reads overlaps. It is a graph representation of the metagenome in adjacency list format.

Key (reads)	**Map (reads overlaps)**
0	[1,6,7]
1	[0,2,7,13]
2	[1,7,12, , 8, 16]
. . .	. . .
n	[1,5,13]

n (*n*=number of reads). The first saves the visited nodes and the second stores the temporal the bin (a set of reads before of Edge analysis stage). Two pointers, *put* and *get*, allow adding and removing nodes into *Sv*.

Subgraph traversal starts at first key-value into the MatrixQuery. It is added into the query vector *Qv* to be marked as visited. Then it and its alignments (the list in the map-value) are copied into the Stack vector *Sv* (*put*-pointer increases *e* times, where *e* is the number of edges). Further, each node in the stack is checked into the *Qv* vector to know if it was visited. If the node was visited, the next node from the stack is taken (*get*-pointer increases one position). Else, it is added to *Qv*, and its edges are passed to *Sv* (*put*-pointer increases *e* times). The process finishes when no more nodes can be inserted into *Sv* (*get*-pointer coincides with *put*-pointer). Finally, the Stack vector contains the temporal bin (a subgraph), and the Edge analysis starts to remove the outliers. The Graph traversal process finishes when all the nodes are into the query vector (*Qv*), which indicates that they were visited. Edges analysis stage removes some reads and generates the final bin. Once all nodes (reads) have been visited, the bins and their reads are saved on output FASTA or FASTQ files. The user can define a minimum bin size (number of reads into the bin) to avoid reporting small bins.

3.1.3 Edge analysis stage

The adjacency list, generated in the Read-alignment stage, allows reporting the reads number-of edges histogram (Figure 3.1(c)). It is computed by counting, for each key-value, the number of reads in the map-value field into MatrixQuery (e.g., Table 3.1 indicates that read *R0* has three edges that connected it with the reads *R1*, *R6*, and *R7*). We have observed that the number of edges distribution should be normal like, after separating repeat regions, and that it is similar for each molecule (each bin) into a metagenomic experiment.

Normal distribution for the number of edges can be explained using the central limit theorem (CLT) [6]. The different abundance level of each species in the metagenome, the shotgun sequencing process and the number of reads generated in it, make that read alignment of all reads versus all reads can be view as a process of random variables independently drawn from independent distributions. Under these conditions, the CLT establishes that the sum of these distributions must converge in a normal distribution. We have observed this behavior in our experiments (see experimental subsection) by plotting the number-of edges histogram.

The histogram enables identification of the following problems in the bins. i) nodes with a total of edges higher than the mean: they usually represent repeated regions in the same genome or zones that are common to several species. ii) nodes with a total of edges less than the mean: we have observed that they are produced mainly by chimeric reads or sequencing errors. Both of these problems make that reads from different DNA molecules end up being related. To separate the graph, we must keep only nodes such that the number of edges histogram follows a normal-like distribution. Therefore, we must detect extreme values, unusually large or small amounts when compared with others into the bin, and remove them. In the next sections, we demonstrate that we can process these nodes as outliers.

3.1.3.1 Outliers definition

An outlier is an observation that appears to deviate markedly from other members in the sample. The classical approach to screen outliers is to use the Standard Deviation (σ) method. It defines observation as an outlier if it is outside the intervals $\pm 3\sigma$, (other authors, i.e., [58] [67] use 2.5σ or even 2.0σ around the mean). However, the authors indicate two main problems when using the mean as the central tendency indicator. i) Outliers affect the mean and standard deviation. ii) Outliers cannot be detected for small samples. These problems can be resolved by substituting the mean by the median as follows.

3.1.3.2 The median absolute deviation (MAD) scale estimator

MAD is a robust nonparametric spread estimator. It uses the median instead of mean to estimate the amount of data dispersion. The median (*M*), like the mean, is a measure of the central tendency of a random variable, but, as opposed to the mean, it is very insensitive to outliers and the sample size. The MAD is defined as:

$$MAD = median(|x_i - median(x_i)|) \tag{Eq. 3.1}$$

For a normal distribution, the MAD can be used as a consistent estimator of the population standard deviation as:

$$\sigma^{'} = b.MAD \qquad \text{(Eq. 3.2)}$$

where b is a constant scale factor, for normally distributed data *b=1.4826*.

This reworked form of $\sigma^{'}$ allows flagging outliers by considering distances from the median (M). The decision criterion (for the value of 3) becomes:

$$M - 3.\sigma^{'} < x_i < M + 3.\sigma^{'} \qquad \text{(Eq. 3.3)}$$

3.1.3.3 Outliers relation with the maximal and minimal number of edges by node

Since the distribution on the number of edges per node departures from a normal, because of the noise produced by the similarity of regions of the genome with other genomes or repetitive zones, we can use MAD (according to the Eq. 3.2) to compute the population standard deviation of the number of edges per read in the bin and detect outliers. Consequently, we use the distances from the median (according to the Eq. 3.3), to mark sequences out of the three standard deviations as outliers, and separate them. After separating outliers, it is common that the number of edges into the bin becomes normally distributed.

The characteristics of a normal distribution (see Figure 3.4) and because it is not possible to have nodes with the number of edges less than zero, allow defining the parameter p (in Eq. 3.4) as the measure of normality for the bin. A p-value close to one indicates that 95% of edges per node are not more than three standard deviations from the mean; as a result, the bin must have a near-normal distribution. The p-parameter also allows iterating on the outlier process removing new reads until reaching a p-value close to 1.0; it can also stop when the bin is too small to be reported.

Table B.1, in Appendix B, illustrates the Edge analysis process for the example in Figure 3.1. The experimental section exhibits MAD convenience to remove outliers and produce "pure" bins (in which most of the reads are from the same molecule).

$$p = \frac{3.\sigma^{''}}{\mu^{''}} \qquad \text{(Eq. 3.4)}$$

where the μ'' and σ'' are the mean and standard deviation of the bin, after the outliers removing process.

CLAME implements the Edge-analysis stage in the binning function, which develops the MAD process. It removes outlier reads from the Stack vector Sv. The process starts assessing the normality of the bin using the p-parameter (according to the Eq. 3.4). If the p-value is higher than a fixed tolerance, (CLAME's tol-parameter with a default value of 0.5), means that distribution is not normal and the reads marked as outliers must be removed from *Sv*. If the *p*-value is less than the tolerance value, the Edge analysis process finishes, and the balanced nodes into of *Sv* are printed.

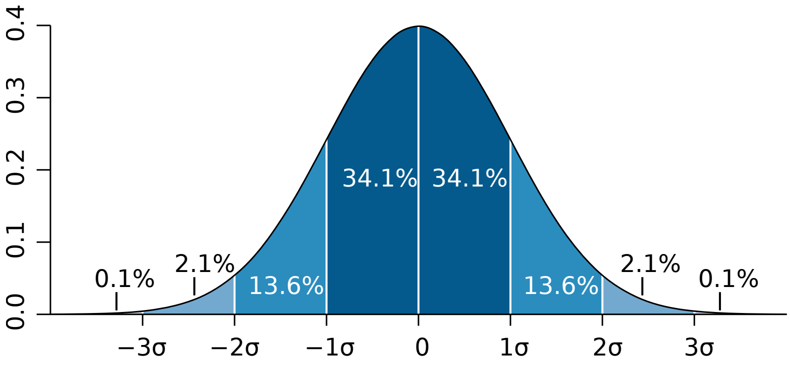


FIGURE 3.4: Probability density function for a normal distribution

3.2 Experimental Evaluation

In this section, we describe several controlled experiments that we used to validate and asses the performance of our method. We illustrate the application of CLAME on two real metagenomes in Chapter 5.

3.2.1 Simulated simple metagenome

We created a synthetic metagenome dataset using 289,917 reads of *Brucella canis* and 375,122 reads of *Mycobacterium tuberculosis* genomes, both generated with the ROCHEs 454 titanium platform and associated with the NCBIt's BioProjects PRJEB4803 and PRJEB8877, respectively. The reads were quality trimmed at Q30 using RAPIFILT, our custom tool to clean the reads; we introduce it in Chapter 4. The cleaned reads were concatenated on a simple multi-Fasta file to get a total of 665,039 mixed reads that formed the Brucella-Mycobacterium synthetic metagenome.

We started the analysis using different number-of-bases alignment parameter *b*, and showed its effect on the read alignment stage and bins production. Then, to clarify the profile of the number of edges, we set the number-of-bases alignment parameter to $b = 70bp$ and ran the edges analysis stage.

Finally, we compare our results against MetaProb [33], BiMeta [116], AbundanceBin [124], and MetaBinG [46] tools. We set up to two the number of bins or species for the tools that these numbers have to be specified. Quality control for each binning tool was again checked by matching the content (read codes) of each bin against the original raw files.

3.2.2 Simulated multi-species metagenome

We created a metagenomic dataset based on bacterial genomes of five species, which we selected to mimic the biological diversity found in the San Fernando hot spring metagenome (which we describe in Chapter 5). We downloaded the raw reads from the NCBI database and merged them to produce the final dataset with 601,628 reads (150.14 Mbp). Table B.2, in Appendix B, shows the number of raw reads, the NCBI reference, the taxonomy, and the total of reads used from each genome.

We used the 16S rRNA gene, which is a highly conserved zone between different species of bacteria, to illustrate how shared regions affect the bin generation by connecting two subgraphs from different species. We executed CLAME on two scenarios (i.e., with 16S rRNA sequences and without 16S rRNA sequences). In the first case, we used the value $b = 70$ bp as the number-of-bases alignment and binned the whole metagenome. For the second scenario, we first mapped the metagenome into the Rfam database [35]. We used *genFM9* fuction to build the FM-index of the Rfam sequences. Then we used the *map2FM9* fuction to align the sequences and manually remove them fromm the pool dataset. Finally, we used CLAME to bin the balance sequences.

We executed MetaProb [33], BiMeta [116], AbundanceBin [124], and MetaBinG [46] tools with this metagenome. For the binning tools in which the number of bins or species have to be specified, we set this parameter to five. Quality control for each tool was checked, by matching the content of each bin against the original raw file.

3.2.3 Mock-Even community metagenome

The Mock-Even sample makes part of the Human Microbiome Project (HMP) [110] and has been studied using MOCAT [54] and MetAMOS [109] frameworks. We downloaded the raw data (1,386,198 sequences) from NCBI, SRA accession number SRR072233. We also download the references sequences of the species that form this metagenome from MOCAT web page. To rate the contribution of each species in the sample, we used Bowtie 2 [57] to map the raw reads against the contigs reported by MOCAT. Table B.3, in Appendix B, summarizes the abundance of the five dominant organisms in the sample.

We removed low-quality reads ($Q < 30$ and $length < 70bp$) and sequences that align with the 16S-rRNA ribosomal Rfam database. Then the reads were binned with CLAME using $b = 40bp$, only bins with more of 2000 read were kept. Quality control for each tool was checked by matching the content of each bin against the original raw file codes using Bowtie 2. Moreover, we use CheckM tool [85] to estimate contamination of each bin by detecting the presence of single-copy of essential genes. It also reports the genome completeness according to the number of genes presents in the bin. Furthermore, the tool measures the Strain Heterogeneity by indicating which percent of the essential genes come from near species.

Finally, we used MetaProb [33], BiMeta [116], AbundanceBin [124], and MetaBinG [46] tools to bin this metagenome and compare all the results. For the binning tools in which the number of bins or species have to be specified, we configured this parameter to five.

3.2.4 Brocadia caroliniensis metagenome

We used a metagenome recovered from a full-scale glycerol-fed nitritation-denitritation separate centrate treatment process (NCBI project PRJNA228949). The original paper [84] reports that 2,448,982 reads were manually analyzed to generate 209 contigs (with size > 500 bp) which integrate the draft genome for *Brocadia caroliniensis* species.

We removed low-quality reads ($Q < 30$ and $length < 70bp$) and sequences that align with 16S-ribosomal Rfam database. We set $b = 70$ bp as the number-of-bases alignment parameter and ran the edges analysis stage. Then, we assembled the reads from the main bins, using SPAdes tool (default parameters). Then we used Quast tool [39] to assess the contigs quality, abundance, and coverage of the genome recover against the reported Brocadia genome. Additionally, we used CheckM [85] tool to estimate contamination of each bin by detecting the presence of single-copy of essential genes and measure the completeness according to the number of genes presents in the bin.

Finally, we compared our results and performance versus the report generated by MetaProb [33], BiMeta [116], AbundanceBin [124], and MetaBinG [46] tools.

3.2.5 Computational performance

We started assessing the computational requirements and precision of our alignment strategy. We aligned the reads of each experiment against Rfam 16S rRNA ribosomal database. We downloaded the Rfam sequences (1,690,540 reads, 6.3 Mbp) and used the *genFM9* function of CLAME to produce the FM-index of the database. Then we used the function *map2FM9* to map the sequences of each experiment against this structure. We reported the number of reads that aligned with the reference and compared these results against the report generated by BLAST [2] BWA [60] and Bowtie 2 [57] tools.

Later, we evaluated the computational time and memory consumption of CLAME to study the Brocadia metagenome. We have observed similar results with the other datasets. We analyzed two scenarios i) when the FM-index needs to be build and ii) using an FM-index from previous construction. Then, to illustrate CLAME scalability, we executed all the experiments using several CPUs. We used OpenMP version 4.4 [52] with 1, 2, 4, 8, 16, 32, and 64 threads on each dataset. Finally, we compare CLAME computational requirements against other states of the art binning tools.

We executed all the experiments on a computer equipped with 64 Intel(R) Xeon(R) CPU X7560 @ 2.27GHz and 500 GB of RAM, and Linux-Centos-7.2 OS. We used Extrae 3.7 [28] and Paraver 4.8 [83] tools to measure computational performance. We measured the computation time using PAPI instrumentation tool [62]; we inform the average of five execution for each experiment. We employed the Valgrind [76] tool to measure CLAME memory usage; we report the maximal memory consumption of each dataset.

3.3 Results

3.3.1 Binning performance

3.3.1.1 Simulated-simple metagenome

Table 3.2 illustrates the relation among the number-of-bases alignment parameter b, the bins size, and the species contribution for the Brucella-Mycobacterium metagenome. It shows that, a reduced number of bases ($b \leq 35bp$) groups all the reads into the same bin. When this value increases the bin size decreases but the "bins quality," referred to as the number of reads from different species into the same bin, improves. Finally, a significant value for this parameter ($b > 100bp$)) makes that the metagenome results fragmented into too many small bins.

TABLE 3.2: CLAME report for the simulated-simple metagenome

Number-of-bases alignment (bp)	Total Bins	Bin Size (Number of reads)	B. canis contribution (Number of reads into the biggest bin)	M. tuberculosis contribution (Number of reads into the biggest bin)
20	1	645434	282666	362768
35	1	642867	282921	359946
70	2	625946 (bin0+bin1)	279362 (bin0)	346584 (bin1)
100	2	607212 (bin0+bin1)	271173 (bin0)	336039 (bin1)
150	3	559068 (bin0+bin1+bin2)	245171 (bin0)	311866 (bin2)
200	13	300714 (bin0++bin12)	6720 (bin1)	265792 (bin9)

Since the high-quality sequencing process, and the taxonomic distance of the two species (phylum level, which suggests few shared regions), the graph result into two subgraphs (bins) after of the Read alignment stage (for $b \geq 70bp$). We have observed that this value produces suitable results in most of our experiments. We set $b = 70bp$ as the default value for the number-of-bases alignment parameter in CLAME; however, it can vary according to the metagenome complexity.

Figure 3.5 illustrates the number of edges histogram, using $b = 70bp$, for the simulated metagenome (red line), we manually highlighted the reads from *M. tuberculosis* (blue line), and *B. Canis reads* (blue line). It shows that the distribution of the metagenome results of the contribution of each species distribution. It also indicates normal-like distribution (if we exclude the nodes with very few connections), that follows the number of edges for each one of the species.

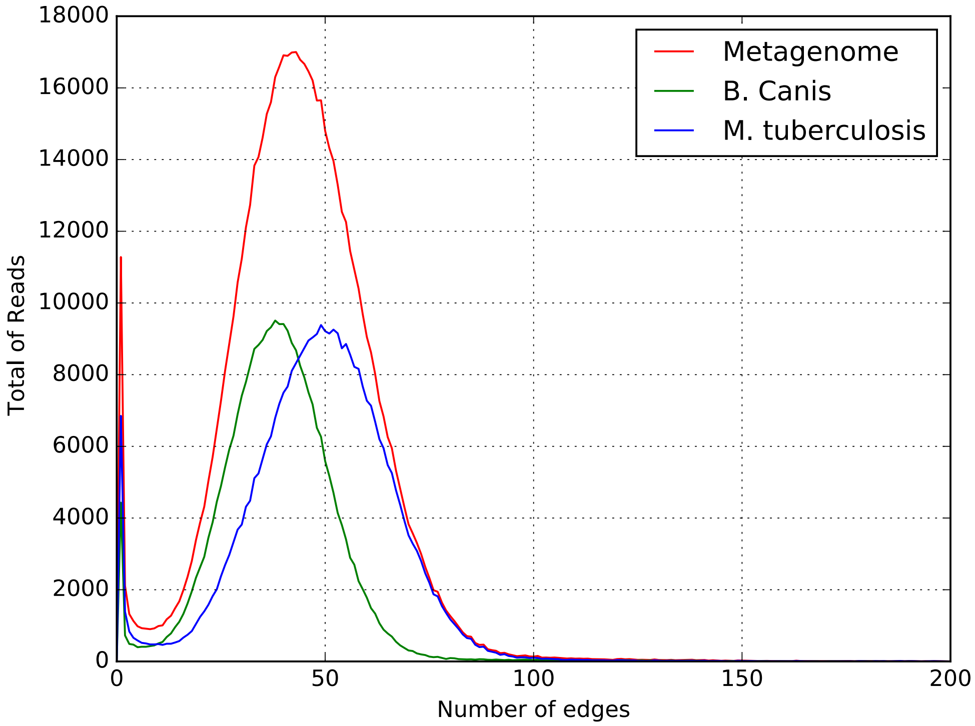


FIGURE 3.5: Number-of-edges histogram for the simulated-simple metagenome

Table 3.3 shows the total of bins generated and the Edges Analysis report (with and without the MAD process) produced by CLAME ($b = 70bp$). When the MAD analysis is disabled all the reads no matter the number of edges, are reported. When it is enabled only reads in the range, 6 to 94 for the *bin0* and 3 to 194 for the *bin1* are considered. This process reduces the bin size but improves the statistical values of each bin (p-value is close to 1.0).

Table 3.4 compares CLAME's results against MetaProb [33], BiMeta [116], AbundanceBin [124], and MetaBinG [46] tools. It shows that although most of them produced individual bins for *B. canis* and *M. tuberculosis* species, only our strategy created bins that contained reads from only one species. Moreover, it was the fastest tool.

TABLE 3.3: Edges analysis report for the simulated-simple metagenome

MAD	Bin	Bin Size (Number of reads)	mean	std	p=3std/mean	Outlier boundaries
OFF	Bin0: M. tuberculosis	353876	51.89	35.07	2	0 to inf
(tol=inf)	Bin1: B. canis	280014	14.33	13.02	2.7	0 to inf
ON	Bin0: M. tuberculosis	346584	49.63	14.95	0.9	6 to 94
(tol=0.5)	Bin1: B. canis	279392	39.16	14.23	1.1	3 to 194

TABLE 3.4: Bins reported by each tool on the simulated metagenome

Tool	Bins	Total reads by bin	B. Cannis	M. Tuberculosis	Time (m)
CLAME (b=70)	**2**	346584 279392	**0** **279392**	**346584** **0**	**8**
BiMeta	2	8990 656049	8683 366439	307 289610	49
MetaProb	2	368642 296397	2901 287062	365787 9335	12
AbundanceBin	2	659892 5142	288233 1684	371659 3458	85
MetaBinG*	2	300615 338650	5215 267794	295400 70856	97

*We used the CPU version

3.3.1.2 Simulated multispecies metagenome

Table 3.5 illustrates the bins generated, using $b = 70bp$, for the total of reads previous to remove the 16S rRNA sequences. It also shows the contribution of each species within the bins. Given the taxonomic distance of the species (class level) of this experiment, some bins contain sequences from different species.

Table 3.6 shows the total of bins generated by CLAME, using $b = 70bp$, after removing the 8900 sequences that aligned with the 16S-ribosomal Rfam database. It shows that, in this case, CLAME did not mix reads from different species.

Figure 3.6 shows the number of edges histogram. We manually underline the contribution of the five species in the histogram. It shows a normal distribution for the Dokdonella, Synechocystis, Hymnobacter, and Rhizobium species in the range 0

TABLE 3.5: Binning report for the raw reds that compose the multi-species metagenome

Bin	Bin Size (Number of reads)	Synecho-cystis	Dokdo-nella	Hymnobacter	Micro-bacteriaceae	Rhizobium
0	366818	**59650**	**306927**	**187**	0	54
1	24339	0	0	0	24339	0
2	6939	0	6939	0	0	0

TABLE 3.6: Bins composition for the simulated multispecies metagenome

Tool	Bins	Total reads by bin	Synecho-cystis	Dokdo-nella	Hymno-bacter	Micro-bacteriaceae	Rhizo-bium	Time (m)
CLAME (b=70bp)	7	21182	21182	**0**	**0**	**0**	**0**	3
		18054	18054	**0**	**0**	**0**	**0**	
		209642	**0**	209642	**0**	**0**	**0**	
		12152	**0**	12152	**0**	**0**	**0**	
		13927	**0**	13927	**0**	**0**	**0**	
		10405	**0**	10405	**0**	**0**	**0**	
		24315	**0**	**0**	**0**	24315	**0**	

to 100 edges; Microbacteriaceae edge-distribution exceed the 100 edges.

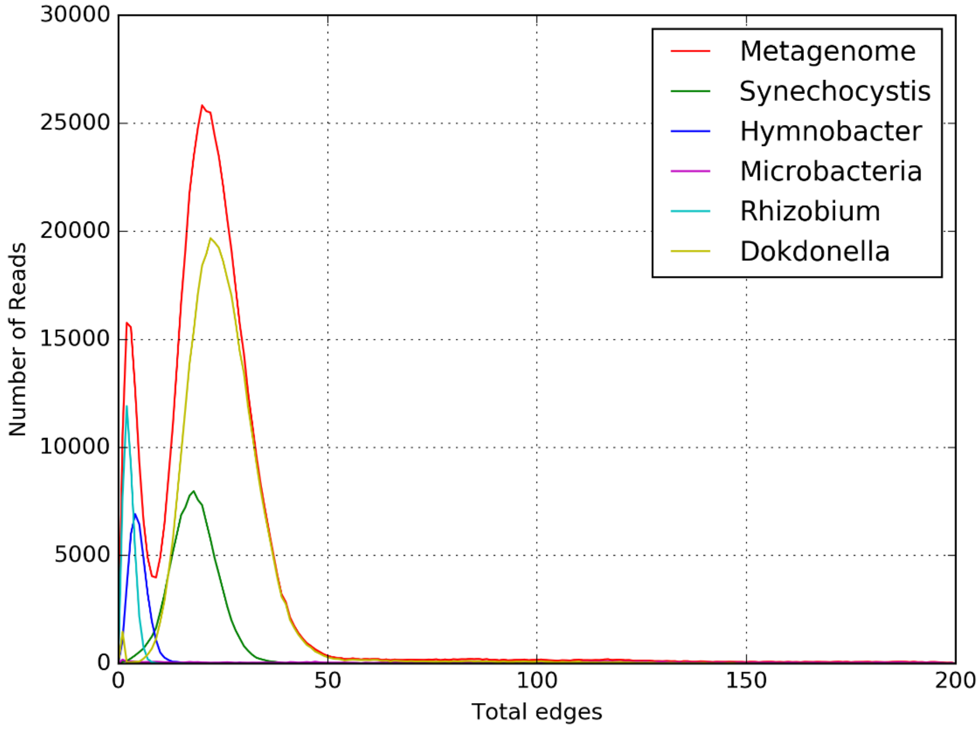


FIGURE 3.6: Number-of-edges histogram for the simulated multi-species metagenome

Table 3.7 shows the statistics values for the simulated multi-species metagenome. MAD statistic analysis shows that most of the bins are in the range 0 to 100 number of edges, except the *Bin2*, which contain the Microbacteriaceae species. The p-value indicates a normal distribution in each bin. Since the few species in the genome, the removing 16S rRNA sequences process was enough to get "pure" bins and not MAD analysis was necessary.

Table 3.8 compares CLAMEs results against MetaProb [33], BiMeta [116], AbundanceBin [124], and MetaBinG [46] tools. It shows that our method was the fastest tool, and the only that does not combine reads from more than one species into the same bin. However, CLAME could not recover the Hymnobacter species and

TABLE 3.7: Statistics values for the multispecies metagenome

Bin	Bin Size (Number of reads)	mean	std	Median	MAD	p=3std/mean	Outlier boundaries
0	21182	18.76	5.44	28.5	21.12	0.87	3 to 36
1	18054	18.83	6.03	28.5	21.13	0.96	3 to 44
2	209642	88.72	20.49	135	100.08	0.69	28 to 152
3	12152	25.90	7.27	25	7.413	0.84	4 to 47
4	13927	23.66	5.81	34.5	25.57	0.73	4 to 46
5	10405	23.56	7.83	23	5.93	0.99	6 to 40
6	24315	1462.94	610.98	1518	634.55	1.25	3 to 2859

the Rhizobium species. In Chapter 4, we show that an iterative process, removing binned reads and reducing the bases parameter, can recover some part of them.

3.3.1.3 Mock-Even community metagenome

Table 3.9 illustrates the total of bins and the MAD statistics values generated by CLAME $(b = 40bp)$ for the leftover reads after removing low-quality bases and 16S rRNA ribosomal sequences. It also shows the iterative process, developed by the Edges analysis stage, to redefine the bin by removing outliers until to get a near-normal distribution (using CLAME's $tol = 0.5$).

Figure 3.7 shows the abundance level of each species reported by CLAME. We manually highlighted the contribution of the five main species in the histogram. It shows a normal distribution, in the range 0 to 60 edges, for the sequences belong to Acinetobacte, Bacteroidetes, Staphylococcus, and Propionibacterium species. Deinococcus species indicates a scattered distribution. It agrees with the outliers boundaries reported by CLAME.

Table 3.10 reports the species contribution into each bin. It indicates that CLAME recovered most of the reads belong to predominant species (Deinococcus-Deinococcus, and Proteobacteria-Acinetobacter) into two main groups (the first with 409,719 reads and the second with 58,301). However, the bins show sequences from different species into the same bin. It is essential to mention that this is not a controlled metagenome, and therefore we cannot be sure of the origin of each read.

To improve the annotation, we used CheckM [85] tool to assess the bin contamination in terms of single-copy of essential genes. Table 3.11 summarizes these results. It illustrates that all the bins show near-zero contamination level. It also shows that the most significant bin contains some 50% of the genome of the Deinococcus bacteria. Bin0 and Bin2 comprise less than 10% of the Proteobacteria-Acinetobacter and Bacteroidetes-Bacteroides genomes. Bin3 and Bin4 are too small to provide some gene. CLAME could not bin the other species. These results confirm the ability of our method to discriminate the most relevant reads from the predominant species

TABLE 3.8: Report for multispecies metagenome using several binning methods

Tool	Bins	Total reads by bin	Synechocystis	Dokdonella	Hymnobacter	Microbacteriaceae	Rhizobium	Time (m)
CLAME (b=70bp)	7	21182	21182	**0**	**0**	**0**	**0**	3
		18054	18054	**0**	**0**	**0**	**0**	
		209642	**0**	209642	**0**	**0**	**0**	
		12152	**0**	12152	**0**	**0**	**0**	
		13927	**0**	13927	**0**	**0**	**0**	
		10405	**0**	10405	**0**	**0**	**0**	
		24315	**0**	**0**	**0**	24315	**0**	
BiMeta	1	601624	112805	376022	37599	37599	37599	32
MetaProb	5	361966	1	341866	108	7236	12755	11
		27977	508	12139	1707	214	13409	
		113349	111889	695	641	6	118	
		38400	294	729	34383	2446	548	
		59932	113	20593	760	27697	10769	
AbundanceBin	5	41326	32546	8780	0	0	0	68
		512104	502795	9309	0	0	0	
		86501	42296	44205	0	0	0	
		324135	11975	312160	0	0	0	
		1645	77	0	0	0	0	
MetaBinG*	5	410033	30727	302805	23480	19944	33081	120
		73263	799	57637	3915	9490	1423	
		61401	56764	2344	772	1211	310	
		24966	18955	3042	1079	870	1021	
		10826	12	3800	6444	436	134	

*We used the CPU version

TABLE 3.9: Total of bins and statistic values for the Mock-Even metagenome

Bin	Bin Size (Number of reads)	mean	std	Median	MAD	p=3std/mean	Outlier boundaries
0	1070791	186.75	797.68	36	41.5128	12.8141	3 to 119
	732305	33.611	30.6895	21	14.826	2.73924	3 to 50
	275187	21.3269	10.9219 20	20	11.8608	1.53636	3 to 40
	58301	**20.7973**	**9.10806**	**20**	**10.3782**	**1.31383**	**5 to 35**
1	472795	395.609	1182.59	158	75.6126	8.9679	41 to 309
	409719	**150.683**	**58.9758**	**145**	**60.7866**	**1.17417**	**41 to 266**
2	63057	1987.07	2745.61	747	607.866	4.14522	310 to 1962
	46508	665.996	420.036	678	700.529	1.89206	310 to 1956
	29959	810.747	450.652	709	481.845	1.66754	310 to 1552
	13747	**792.279**	**336.759**	**791**	**382.511**	**1.27515**	**310 to 1364**
3	18705	5243.36	3168.59	3010	2007.44	1.81291	1553 to 6803
	10334	**2431.09**	**666.083**	**3550.5**	**2740.96**	**0.821954**	**1553 to 6603**
4	**8371**	**8715.06**	**265.39**	**8769**	**203.116**	**0.0913558**	**8505 to 9175**

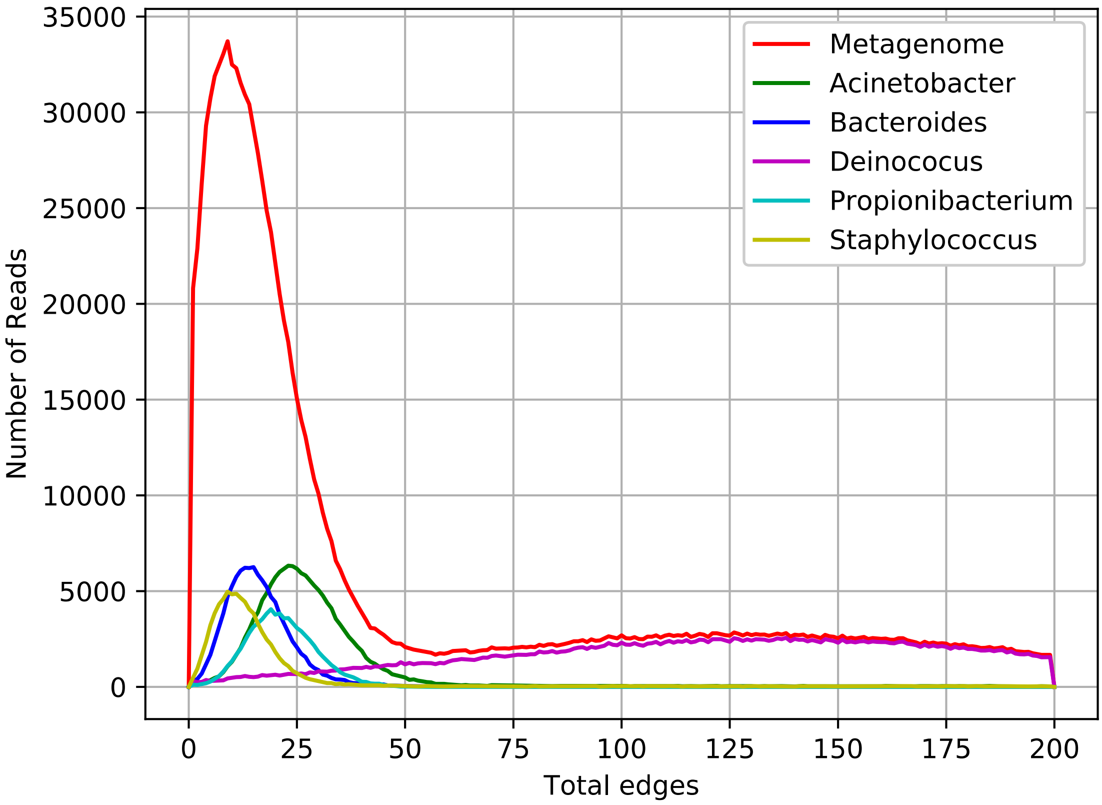


FIGURE 3.7: Number-of-edges histogram for the Mock-Even metagenome

and show the limitation of our approach to detect species in minor abundance. We show in Chapter 4 how we improve the study of this metagenome by using an iterative process removing the studied reads and binning the balance sequences, reducing the b parameter.

Table 3.12 compares CLAME results and performance versus MetaProb [33], BiMeta [116], AbundanceBin [124], and MetaBinG [46]. It shows that all the tools binned Deinococcus species, but they failed with the species in minor abundance. MetaProb and BiMeta recovered the most predominant species, but both tools show contamination in the bins and required more time than our method. These results

TABLE 3.10: Bins composition for the Mock-Even community metagenome

Tool	Bins	Total reads by bin	Deinococcus	Proteobacteria	Bacteroidetes	Firmicutes	Actinobateria	Time (m)
			Deinococcus	Acinetobacter	Bacteroides	Staphylococcus	Propionibacterium	
CLAME	5	58301	1986	18304	8791	5726	6661	23
		409719	361271	9918	2271	302	5473	
		13747	6345	526	334	3914	107	
		10334	9165	244	127	141	73	
		8371	650	125	34	1234	7328	

TABLE 3.11: Completeness and Contamination levels for Mock-Even metagenome

Bin	Size	Completeness	Contamination	Strain Heterogeneity	Lineage
0	58301	7.84	0.47	0.00	Proteobacteria
1	**409719**	**54.68**	**0.00**	**0.00**	Deinococcus
2	13747	4.17	0.00	0.00	Bacteroidetes
3	10334	0.00	0.00	0.00	Deinococcus
4	8371	0.00	0.00	0.00	Actinobateria

TABLE 3.12: Binning report using different binning tools on the Mock-Even metagenome

Tool	Bins	Total reads by bin	Deinococcus Deinococcus	Proteobacteria Acinetobacter	Bacteroidetes Bacteroides	Firmicutes Staphylococcus	Actinobateria Propionibacterium	Time (m)
CLAME	5	58301	1986	18304	8791	5726	6661	23
		409719	361271	9918	2271	302	5473	
		13747	6345	526	334	3914	107	
		10334	9165	244	127	141	73	
		8371	650	125	34	1234	7328	
BiMeta	5	70135	1	1540	1467	21235	0	235
		323284	14162	94544	93404	11038	1258	
		319872	154135	2621	7499	798	73163	
		250189	522	62440	11048	60594	38	
		408053	317452	217	5	3	3717	
MetaProb	3	269431	235623	23451	1267	6745	2345	34
		148152	124167	15668	346	2348	5623	
		70774	56489	12589	1245	126	325	
AbundanceBin	1	1386198	486683	161464	113507	93820	78223	2600
MetaBinG*	5	72216	45	66221	463	1033	5	205
		46016	125	162	43514	22	11	
		405800	384768	18	132	0	1020	
		52101	1716	2	8	13	45107	
		58216	3	3256	75	45820	0	

*We used the CPU version

show the complexity of this metagenome. We study this metagenome with more detail in Chapter 4.

3.3.1.4 Brocadia caroliniensis metagenome

Figure 3.8 shows the number of edges histogram generated by CLAME $(b = 70bp)$. It indicates that although most sequences are singletons (reads that do not align with any other), there are a secondary concentration in the range 20 to 60 edges.

Table 3.13 illustrates the total of bins and the MAD statistics values generated by CLAME $(b = 70bp)$ for the leftover reads after removing low-quality bases and 16S rRNA ribosomal sequences. We also show the iterative process of removing outliers, developing by the Edges-Analysis stage, to get bins with a near-normal distribution (we used $tol = 0.5$). Note that the outlier boundaries for the bin 0 agree with the limits observed in the edges histogram plot.

Table 3.14 shows the assembly metrics for the contigs generated from each bin.

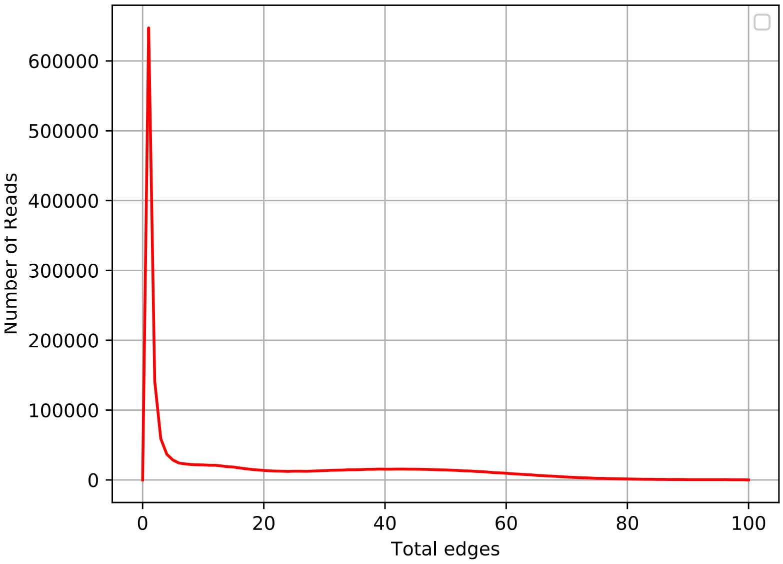


FIGURE 3.8: Number-of-edges histogram for the Brocadia metagenome

TABLE 3.13: Binning report for the Brocadia metagenome

Bin	Bin Size (Number of reads)	mean	std	Median	MAD	p=3std/mean	Outlier boundaries
0	663229	73.15	212.32	30	14.83	8.71	11 to 74
	607483	**29.78**	**12.20**	**28**	**13.34**	**1.22**	**11 to 68**
1	54879	552.83	540.38	347	342.48	2.93	75 to 1374
	45017	452.428	334.882	347	317.28	2.22	75 to 1219
	4918	**630.26**	**232.96**	**970.5**	**719.43**	**1.11**	**431 to 1095**
2	**7788**	**1546.71**	**634.93**	**2271**	**1694.61**	**1.23**	**75 to 3430**

We have included Quast report about genome coverage and CheckM report referent to bin completeness and contamination level. The results show that the contigs from the principal bin cover, at least one time, some 97% of the Brocadia genome. Moreover, these contigs contain more than 90% of the universal genes with a contamination level of less than 11%. The other bins represent a small fraction of the genome, but they are too short to detect any gene. These results show the effectivity of CLAME to recover the dominant genome from a metagenome.

We compare our results against MetaProb [33], BiMeta [116], AbundanceBin [124], and MetaBinG [46] tools. Table 3.15 shows the results of each tool, the number of bins, and the number of reads that map to Brocadia genome. The table also shows the time required for the tools to generate the bins. It indicates that our method was the fastest of all and produced the bin with most of the genome into a single bin.

TABLE 3.14: Assembly metrics and genome coverage for the Brocadia metagenome

Size Bin	Total Contigs per bin	Contigs metrics				Mapping Report		Recovered genome		
		Largest (bp)	N50 (bp)	Genome (Mbp)	ORFS	Genome Fraction (%)	Duplication ratio	Completeness (%)	Contamination (%)	Lineage
0	677	88819	18421	3.96	4330	97.21	1.02	93.96	10.05	Brocadiaceae
1	80	1613	252	0.02	76	0.03	1.55	0.00	0.00	Brocadiaceae
2	12	1347	945	0.06	16	0.07	1.13	0.00	0.00	Brocadiaceae

TABLE 3.15: Analysis report using different binning tools on the Brocadia metagenome

Tool	Bins	Total reads by bin	Brocadia reads	Time (m)
CLAME	3	607483 4918 7788	590534 4918 7788	10
BiMeta	2	229783 2219199	346 589520	11734
MetaProb	886000	5x1 4x1 3x1 2x287 1x1934327	0 1 1 18 589520	41
AbundanceBin	1	1934912	589520	2295
MetaBinG*	4	780749 454970 130449 16158	78758 12764 11508 16158	278

3.3.2 Computational performance

3.3.2.1 Read-Alignment stage accuracy and performance

Table 3.16 compares the construction time, the RAM required and the data compression ratio of our *genFm9* program against representation generated by BWA (index option), Bowtie2 (bowtie2-build function), and BLAST (makeblastdb command) tools for the Rfam 16S rRNA ribosomal database. It shows that although *genFM9* strategy required more RAM than the other programs, it had the best data compression ratio. It is an essential feature because of the vast number of reads in metagenomic experiments. Moreover, it was close to 2x faster than Bowtie2 and BWA, which implement the same strategy.

Figure 3.9 shows the computation statistics and the number of the alignments, against the Rfam database, reported by *map2FM9* function ($b = 20bp$), BLAST ($PI = 70$), BWA (default parameters), and Bowtie 2 (with default parameters). It illustrates that *map2FM9* function is the one that uses more memory, but it is close to 2x faster while having similar results than the other tools.

TABLE 3.16: Compressed representation for 16S-risbomosal Rfam database

	Time (m)	RAM (MB)	Ratio= (Uncompressed/Compressed)
genFM9	2.39	3153.92	3.04
Bowtie 2	18.42	1038.80	0.54
BLAST	3.57	59	1.10
BWA	5.52	630.85	0.68

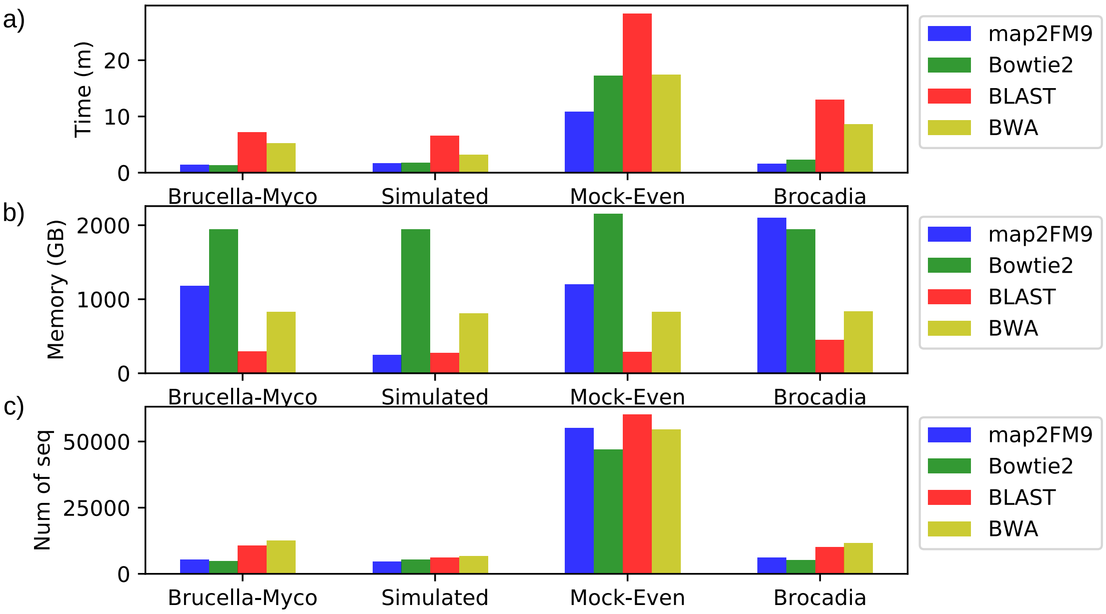


FIGURE 3.9: map2Fm9, Bowtie2 and BLAST performance about a)computation time, b)memory usage, and c)number of alignments

3.3.2.2 CLAME: Computation time

Table 3.17 shows CLAME execution time using b=40bp for the Brocadia metagenome. We illustrate two scenarios: i) CLAME generates the FM-index, ii) CLAME loads an FM-index. The table shows that the alignment stage (composed by the *genFM9* and *map2FM9* functions) requires some 90% of the time in both cases. However, the total time decreases near 25% when an FM-index is loaded. It also shows that the map2Fm9 function is the most demanding task.

Figure 3.10 compares CLAME sequential execution against the parallel implementation of the alignment stage. It shows that the computational time decreases when we use several threads to execute the map2FM9 function. We achieve the maximal speedup to eight threads. When the number of threads increases, the FM-index construction, that is a sequential process, becomes the stage that takes more time, some 64% of the execution. It also shows that when the FM-Index is loaded, total time decreases near 40%.

TABLE 3.17: Global performance of CLAME

CLAME Function		**Time (s)**	
		Built FM9	**Load FM9**
readDNA_sequencesFile		6.96	7.12
alignment	genFM9	147.36	13.32
	map2Fm9	372.73	372.51
binning		8.05	7.41
TOTAL		535.1	400.36

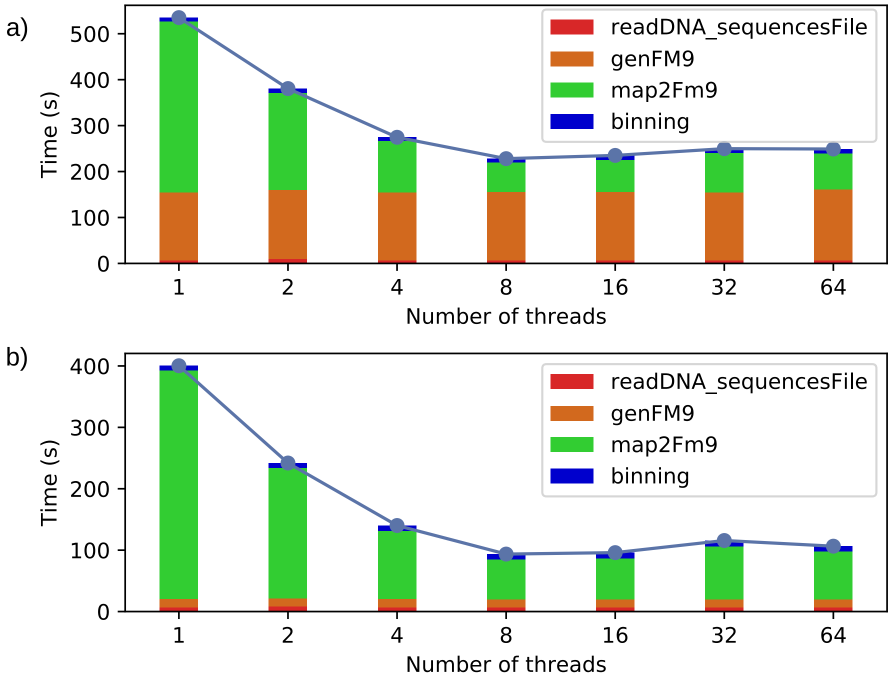


FIGURE 3.10: Sequential versus OpenMP execution of CLAME. a)multithreading execution, with FM-index generation, b)multithreading execution, loading an FM-index

3.3.2.3 CLAME: Memory performance

Figure 3.11 shows the memory consumption for the main stages of CLAME. It shows that the alignment stage (*genFM9* and *map2FM9* functions) requires the most percentage of memory. It also indicates that the RAM consumption reduces when CLAME uses an existing FM-index.

Figure 3.12 shows the memory necessary to load an FM-index and save the containers: Bases, MatrixQuery, Query vector *Qv*, and Stack vector *Sv*. It shows that the FM-index structure requires most of the memory. We also illustrate the memory behavior for several values of the number of bases parameter. Since Bases, *Sv*, and *Qv* arrays depend only on the number of sequences in the metagenome, the size of memory changes only due to the MatrixQuery requirements. It decreases as the number-of-bases parameter increases.

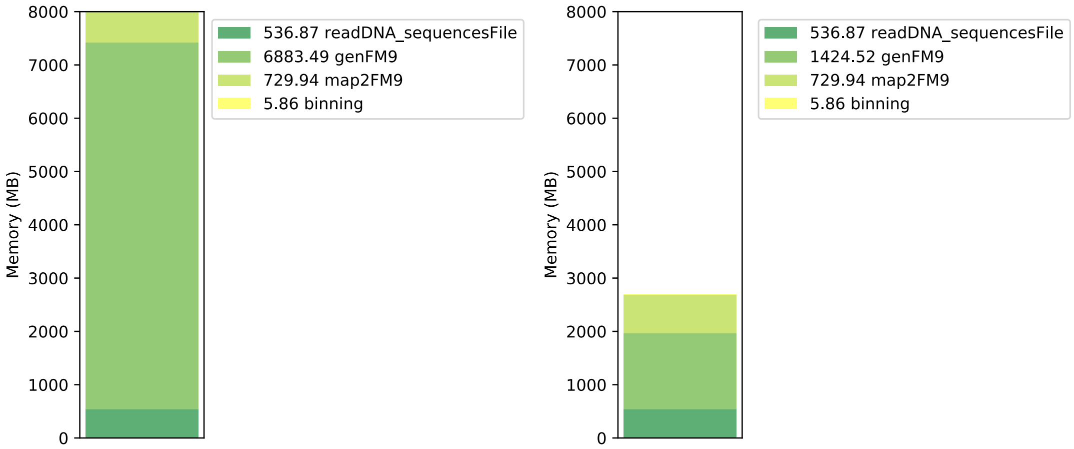


FIGURE 3.11: Memory performance of CLAME for a)multithreading execution, with FM-index generation, and b)multithreading execution, loading an FM-index

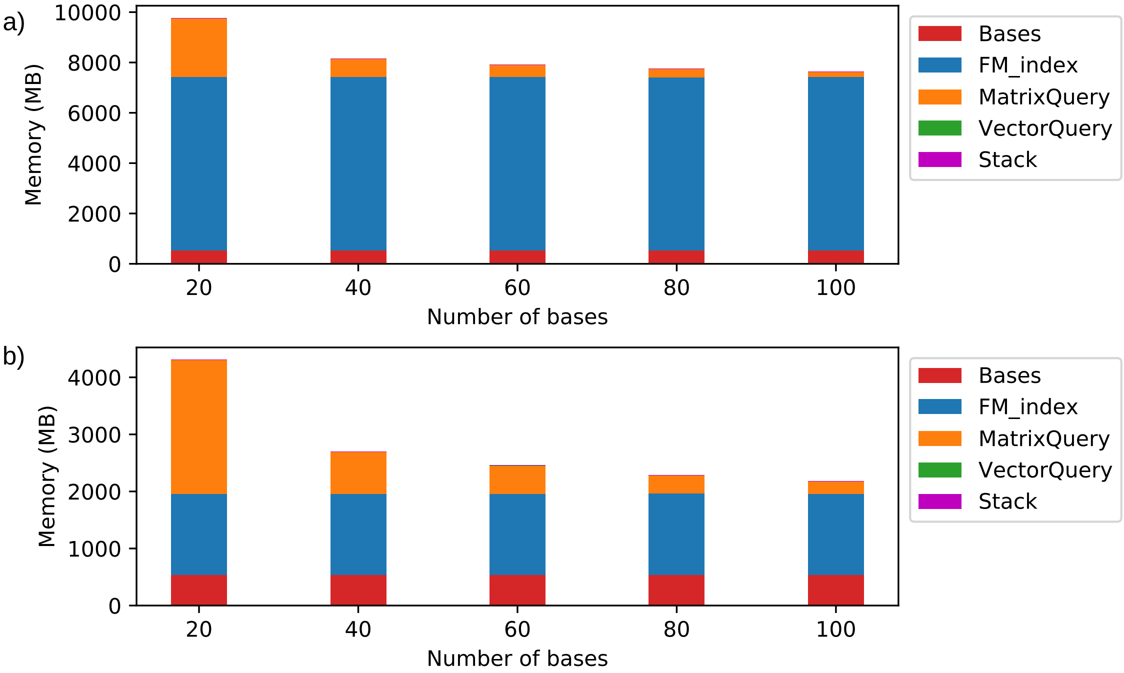


FIGURE 3.12: Memory performance for the CLAME read alignment stage using a different number of bases parameter. a)with FM-index generation, b)loading an FM-index

3.3.2.4 CLAME: Speedup performance

Figure 3.13 shows CLAMEs speedup for the different datasets. It shows scalability up to eight threads. After eight threads, the scalability is not linear because the size of the problem since Brocadia dataset is the biggest it stays linear the longest.

3.3.2.5 CLAME vs other state of the art binning tools

Figure 3.14 shows the computational time and memory consumption required by CLAME, MetaProb [33], BiMeta [116], AbundanceBin [124], and MetaBinG [46]. It

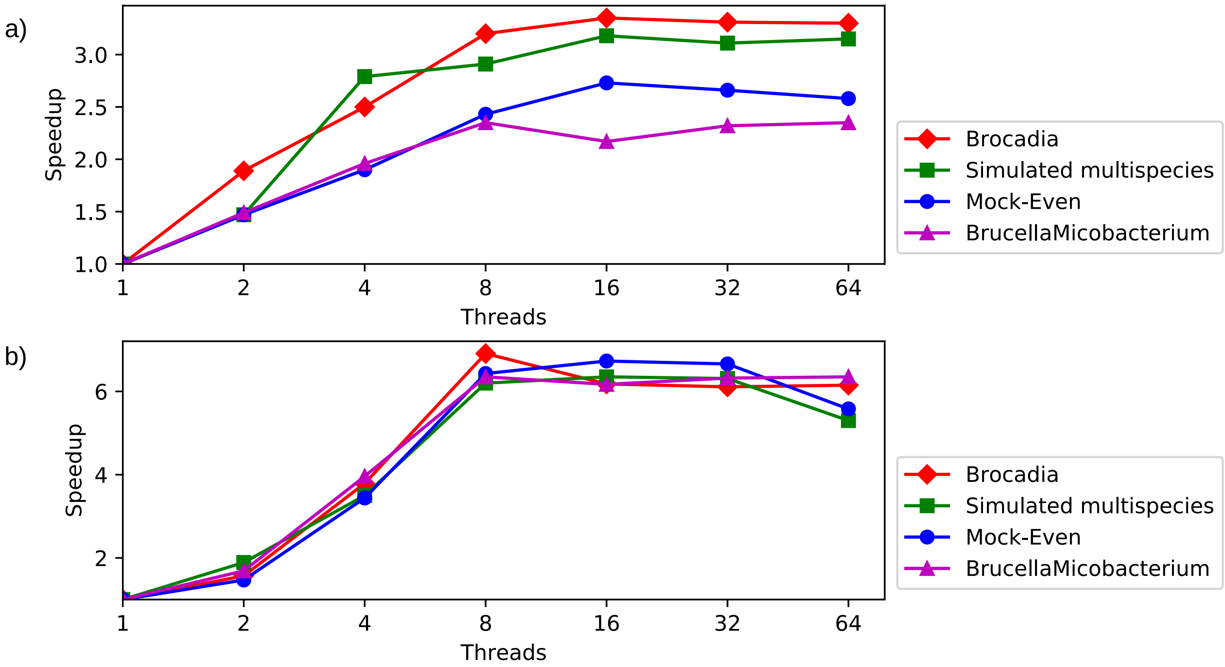


FIGURE 3.13: CLAME speedup using different experiments. a) withFM-index generation, b)loading an FM-index

shows that CLAME faster than the other taxonomy-independent tools (MetaProb, BiMeta, and AbundanceBin) and have similar behavior than MetaBinG, which is a Taxonomy-dependent binning tool. It also shows the high level of memory required by CLAME.

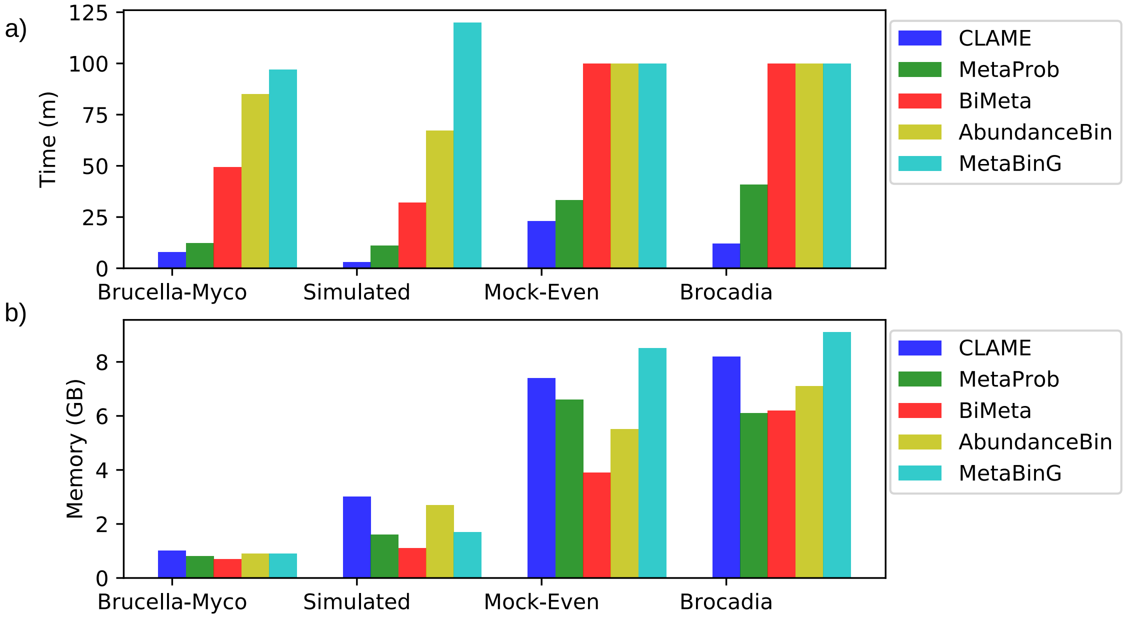


FIGURE 3.14: CLAME performance against other states of the art binning tools. a) computational time, b) Memory consume

3.4 Conclusions

Even though metagenomics allows studying a community without the need of cultivating the species, these datasets contain a mix of the sequences from all organisms in the sample, and it is very challenging to know the origin of each read. We showed that using a very restricted alignment most reads, from a single DNA molecule, could be assigned to the single bin. Moreover, for closely related species in a metagenome, with a significant difference in concentration, the Edges analysis stage can bin them in different groups.

Since NGS technologies generate small DNA fragments, sequence alignment is a fundamental task to reconstruct long DNA sequences. CLAME uses read alignment to produce the relationship graph. A naive implementation of this task compares all the reads versus all the reads, which has $O(n^2)$ complexity. We show that using an FM-index structure, to represent the dataset, it is possible to reduce the computational complexity by allowing to match each read against the whole structure without the need of comparing all possible read pairs. Moreover, a multithread implementation allows distributing the tasks to increase the speed.

While several metagenomic binning tools were unable to separate the synthetic and real problems that we tested, we show that CLAME was faster and most cases better on these problems. However, the binning performance look reduces when the abundance of species is low, or there is some previous knowledge about the species into the metagenome. In this scenery, reference-based methods look more appropriate. CLAME also shows that it is faster than another state of the art binning tools, there is still work needed in all the components of CLAME to reach satisfactory speedup and low memory consuming. It is clear that a more efficient construction of the FM-index and a compact representation of the MatrixQuery container are necessary to reduce memory requirements. Moreover, although Edges-Analysis and Bins generation functions are not computational time demanding tasks, in contrast with the read alignment stage, they limit the global speedup and need to be improved in a future version.

In Chapter 3, we introduce CLAME implementation, the different approaches designed to reduce the analysis time and its computational restrictions. Chapter 4 shows that by grouping sequences with similar DNA composition, CLAME reduces the dataset complexity and improves the assembly and annotation. In Chapter 4, we also present the integration of CLAME into a full framework for metagenomics and compare our methodology against another state of the art pipelines. We show the utility of CLAME to recover novel species from real datasets in Chapter 5.

Chapter 4

DATMA, A Distributed AuTomatic Metagenomic Assembly and Annotation framework

In Chapter 3, we introduced CLAME and showed as it displays promising results reducing the complexity of metagenomes and helping researchers to study metagenomic datasets. However, CLAME requires many manual steps, making it hard to use, especially with large projects. In this Chapter, we introduce DATMA, an integration of CLAME with other omics tools into a distributed workflow for complete metagenomic analysis. We used different experiments to illustrate DATMA's performance and compare its results against other metagenomics frameworks.

4.1 DATMA stages

Figure 4.1 illustrates DATMA's structure and the tools available in each stage of the process. In the following subsections, we will describe each one of these stages.

4.1.1 Reads Quality Trimming and Filtering

DATMA receives FASTQ, FASTA, or Standard Flowgram Files (SFF). For reads' quality control, it uses Trimmomatic [12] or RAPIFILT, which is a custom tool. This stage trims low-quality bases at both ends of the reads and removes the ones that are too short from the dataset. Afterwards, it uses FastQC [3] to plot the quality statistics.

For pair-end reads, DATMA uses FLASH2 [64] to extend the reads and merge them into a single (FASTA or FASTQ) file, before passing them to the next stage. If the fragment length is too large to be combined, we force the merging, only for binning purposes, by adding three extra **N** characters between the end of the first read and the beginning of the second one, which is in reverse-complement (e.g., ATCGT**NNN**TTATC).

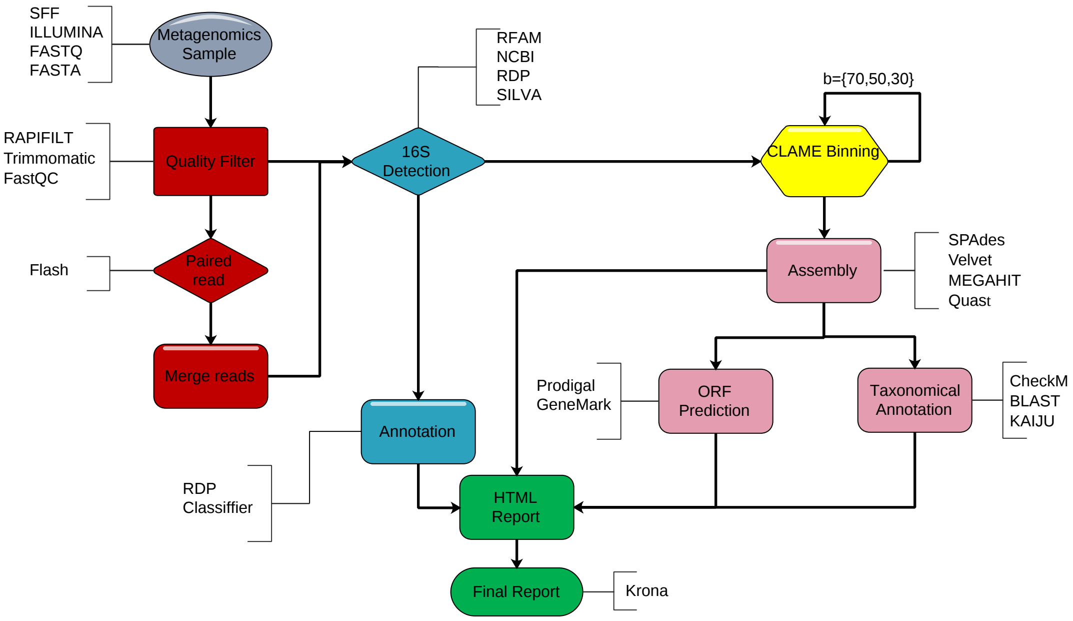


FIGURE 4.1: DATMA automatically executes. (i) sequencing quality control (red blocks) (ii) 16S rRNA genes sequences detection (blue blocks), (iii) CLAME binning (yellow blocks), (iv) de novo assembly, ORF detection, taxonomic analysis (violet blocks) and (vi) data management report (green blocks)

4.1.2 16S rRNA genes sequences detection

In a metagenomic dataset, ribosomal sequences can be used to profile the bacteria species in the sample and estimate their abundance. DATMA uses the BWA tool [60] to map the raw reads against a ribosomal database and remove ribosomal sequences from the pool of reads to improve the binning. This process reduces the probability that these conserved regions connect reads from different species on the same bin. DATMA aligns the reads to a reference 16S rRNA gene-database, the user can select any of NCBI-16S rRNA database [74], RDP [20], Greengenes [22], Rfam [35], RNAmmer [56] or SILVA [92] (Table B.4, in Appendix B, details each one of them). Finally, the detected sequences are classified using the RPD-tool classifier [117].

4.1.3 CLAME binning

DATMA uses CLAME tool to bin DNA sequences. DATMA, by default, starts with 70 (bp) as CLAME's b-parameter. Then, it iterates with other values (e.g., using 50 bp or 30 bp) to explore the metagenome in detail. It is important to highlight that lowering the b-value increases the probability of reads from different molecules reported on the same bin. The user can modify the b-parameter using the configuration file (see DATMA's user manual available in DATMA's GitHub).

4.1.4 Assembly and contigs' evaluation

DATMA assembles (de novo) all bins produced by CLAME. The user can select among different assembly tools: Velvet[125], SPAdes [77] or MegaHit [59]. After assembly Quast tool [39] evaluates the contigs and report their metrics. Finally, DATMA uses CheckM program [85] to assess the quality and contamination of the bins.

4.1.5 ORF detection and taxonomic analysis

DATMA uses the assembled contigs to predict protein-coding-genes; the user can select between Prodigal [44] or GeneMark [9] for this task. Next, the contigs are annotated using BLAST [2] and a local NT-database. DATMA also provides the Kaiju tool [65] for sensitive taxonomic classification.

4.1.6 Final report

DATMA reports the statistics of each workflow stage into an HTML file. It uses Krona [79] to represent the taxonomical classification into an interactive plot. Using the Krona report, the user can explore each bin classification at different taxonomic ranks and select between individual annotation of each bin or combine data from all bins. Figure D.1, in Appendix D, shows an example of the output file generated by DATMA.

4.2 Workflow design

DATMA is a command line application written in Python and tested in Linux. We provide an installation script in our GitHub to automatically install DATMA source codes and the tools that make up part of the workflow. We tested it on Ubuntu 16.04 and included a user manual for custom compilation and installation of source codes on other Linux distributions. By default, DATMA configures all tools called in the workflow according to the authors recommended parameters, but these values can be modified using a configuration file. In this file, the user specifies the input sequence file, the output directory, the workflow stages, the database directories, the number of threads to use, CLAMEs parameters, etc. The minimum configuration file should contain the input-sequence file, the sequence type (i.e., FASTA, FASTQ, or SFF) and the output directory. We show a complete configuration file in DATMA's user manual.

Although there are several workflow engines (e.g., Snakemake [51], Nextflow [108], Ibis [5], and Swift [119]) that we could have used to create DATMA, most of them require that the user learns a set of rules, rewrites the code to include additional

API functions, or specifies the parallel sections. We selected COMPSs [4] framework for its simplicity and because of the parallel distributed execution of the workflow stages. COMPSs offers a simple programming model, that does not require the use of APIs to modify the original user applications, and enables the execution of the same code on different back-ends. It uses a sequential description of the work, and it identifies and launches asynchronous parallel tasks automatically. A complete description of COMPSs and its performance is in [4].

COMPSs allows DATMA to be executed in single or distributed mode. In single mode, the framework executes all the stages into the same computer. In distributed mode, DATMA uses a master-worker execution strategy, to distribute application tasks across the different computer nodes available. It executes the quality control, 16S rRNA identification, and CLAME binning stages in the master node (these stages can be multi-threaded). Once the bins are generated, DATMA assembles and annotates them using the available nodes. It requires two configuration files (resources.xml and project.xml) within the execution environment. The first file contains the information of the available computing resources, and the second file has information about the computing resources to be used for a specific execution. The user manual has an example of each file.

4.3 Experimental evaluation

4.3.1 Metagenomic experiments

We used the experimental dataset explained in Chapter 3 to illustrate DATMA performance and functionality. Since the simplicity of the simulate simple (Brucella-Mycocobacterium) metagenome, we did not study it in this chapter. We included a second controlled experiment that helps us to understand the San Fernando biosolid metagenome (we describe it, in Chapter 5). We created it based on bacterial genomes of five species, which were selected to mimic the biological diversity found in the biosolid metagenome. We downloaded the raw reads, for each species, from the NCBI database. To simulate different abundance levels, similar to the real biosolid metagenome, we randomly took varying amounts of sequences from each dataset. The final dataset (with 1,600,000 reads and 239.5 Mbp) was produced by concatenating the selected sequences into a single multi-FASTA file. Table B.5, in Appendix B, shows the number of raw reads, the NCBI reference, the taxonomy, and the total of reads used from each genome.

We compared DATMA's results and performance against MetaWRAP [113] and SqueezeMeta [105] frameworks. For the experiments, we set the number of threads to four for all the datasets and pipelines. Similar to MetaWRAP, we configured DATMA to use SPAdes [77] as the assembly tool; however, the user can select a different assembler using the DATMA's configuration file. Since most of our experiments are from one sequencing run per sample, we use SqueezeMeta in a sequential mode. This framework illustrates the execution of a pipeline without a binning stage.

4.3.2 Computational performance evaluation

To illustrate the computational performance of DATMA we executed the experiments within two different scenarios: i) single mode, using only the Master machine, and ii) distributed mode, using the Master machine with multiple workers like a grid of computers. We simulated the grid of computers using tree servers (Master, Worker1, and Worker2) connected via a secure shell connection. Table B.6, in Appendix B, illustrates the computer specifications of each server. To simulate a more significant number of workers, like a bigger grid of computing, we allow for several tasks to run on the same computer. Applications were configured to use four threads on all the experiments.

4.4 Results

4.4.1 Simulated multi-species metagenome

We configured DATMA to remove low-quality reads ($Q < 30$, $size < 70bp$). Since it is an Illumina dataset, DATMA automatically merges the reads. We configured DATMA to use: Rfam as 16S rRNA database, CLAME (with $b = 60$,40, and $20bp$), SPAdes, Prodigal, and a local NT to annotate the contigs using BLAST. We provide the complete configuration file in DATMA GitHub.

Table 4.1 shows the assembly metrics and CheckM report for the first five bins. It indicates that our DATMA framework recovered some 60% of the Dokdonella and Synechocystis genomes, which are the predominant species (see Table B.3 in Appendix B). The contamination level, close to 0% for the bins with the strains, shows the outstanding performance of CLAME to bin sequences from the same DNA molecule. It was corroborated by Quast report which suggests that CLAME organized the proposed strains into a reduced number of contigs. DATMA also recovered some of 60% of the Hymenobacter genome. However, it was produced using a reduced number of bases ($b = 20bp$), which increased the contamination level. We found that CLAME binned the species in minor abundance into short contigs too small to detect any gene. It is a consequence of our binning approach.

We studied the metagenome with the alternative frameworks.Table 4.1 illustrates that MetaWRAP [113] shows better completeness than DATMA for the Dokdonella and Synechocystis genomes, but the contamination levels are higher than the

TABLE 4.1: DATMA report for the simulated multispecies metagenome

	Bins	Total Contigs per bin	Contigs metrics				Recovered genome			Time (m)
			Largest (bp)	N50 (bp)	Genome (Mbp)	ORFs	Complete -ness (%)	Contami -nation (%)	Lineage	
DATMA	5	217	228117	56638	3.22	2768	52.71	0.58	Rhodanobacteraceae-Dokdonella	45
		2	101157	89021	0.1	86	10.34	0.00	Rhodanobacteraceae-Dokdonella	
		90	66750	29632	1.24	1215	32.90	0.00	Cyanobacteria-Synechocystis	
		57	18084	31413	1.03	932	24.48	0.00	Cyanobacteria-Synechocystis	
		454	51141	4440	1.33	1517	59.01	10.34*	Cytophagales-Hymenobacter	
MetaWRAP	3	163	119264	41053	3.65	3472	99.67	0.22	Cyanobacteria-Synechocystis	110
		47	335615	183659	4.54	3792	99.19	1.05	Xanthomonadaceae	
		1568	7776	1870	2.84	3641	61.67	1.00	Cytophagales-Hymenobacter	
SqueezeMeta	NA (†)	3735	7656	792	2.47	3713	76.53	3.82	Bacteroidetes	44
		2845	12600	1194	2.59	2822	82.09	0.44	Cyanobacteria	
		740	5937	1197	0.69	737	12.93	0.00	Firmicutes	
		9303	9072	804	5.62	9134	100.00	46.93	Proteobacteria	

*It correspond a bin generated using $b = 20bp$

†We manually selected the contigs from the annotation report

reported by our tool. Moreover, it only can annotate the Dokdonella reads until family level while DATMA could assign them in species level. MetaWRAP overcame the other tested tools for the Hymenobacter genome. SqueezeMeta [105] shows a large number of contigs annotated into the Proteobacteria phylum, but it could not classify any of them into a family clade. In this experiment, DATMA was the fastest tool.

4.4.2 Controlled-Biosolid experiment

We set DATMA with default parameters. It removed low-quality ($Q < 35$) bases at both ends, and the reads with less than 70 bases were discarded. The remaining 1,590,225 sequences were merged using the FLASH2 tool [64]. Then, the 16S rRNA ribosomal sequences were separated using BWA to map the reads against the Rfam database [35]. DATMA reported that a total of 25,629 sequences aligned to the database. Then, CLAME binned the 1,564,596 leftover reads. We configured it with 60bp as the initial alignment threshold and set DATMA to iterate using 40bp and 20bp. We reported the bins with more than 20,000 reads.

Table 4.2 shows the assembly metrics, reported by Quast tool [39], and contigs' quality in terms of the universal single-copy genes using CheckM [85]. It also compares DATMA results against the report generated by MetaWRAP [113], and SqueezeMeta [105] frameworks. MetaWRAP presented higher completeness for Actinobacteria Streptomyces, Chloroflexi Pelolinea, and Proteobacteria Pseudomonas, while DATMA was better for Firmicutes Aneurinibacillus. In the case of Cyanobacteria-Prochlorococcus, CheckM does not have results, which explains why DATMA was better than MetaWRAP which relies on CheckM to create the bins. Because DATMA employs a rigorous binning process, it has the lowest contamination, except for the Firmicutes-Aneurinibacillus, but this could be an annotation problem of CheckM

TABLE 4.2: DATMA report for Controlled-Biosolid experiment

	Total Bins	**Total Contigs per bin**	**Contigs metrics**			**Recovered genome**				**Time (m)**
			Largest (bp)	N50 (bp)	Genome (Mbp)	ORFs	Complete -ness (%)	Contami -nation (%)	Lineage	
DATMA	5	123	398431	58415	3.99	4008	97.03	3.14	Firmicutes-Aneurinibacillus	41
		719	115473	15215	6.59	6147	88.52	4.35	Actinobacteria-Streptomyces	
		278	52700	14522	2.78	2519	37.59	0.74	Actinobacteria-Streptomyces	
		185	213722	66050	4.14	3667	43.86	0.00	Proteobacteria-Pseudomonas	
		5	161499	161499	0.25	550	20.61	0.00	Chloroflexi-Anaerolinea	
MetaWRAP	4	46	955344	232540	6.45	6111	100.00	0.10	Proteobacteria-Pseudomonadaceae	153
		1049	30625	7890	6.18	5681	96.28	2.18	Actinobacteria-Streptomyces	
		66	187207	56363	2.89	2896	91.29	1.61	Firmicutes-Aneurinibacillus	
		7	1422612	567500	3.51	3078	91.81	7.45	Proteobacteria	
SqueezeMeta	NA (†)	8870	6129	768	5.23	8784	88.64	5.23	Actinobacteria	57
		6041	31923	1209	6.08	5949	99.82	1.82	Proteobacteria	
		4044	5211	1074	3.46	4029	95.16	9.48	Firmicutes	
		684	5298	1308	0.81	683	40.36	0.91	Chloroflexi	

†We manually selected the contigs from the annotation report

since all the reads are from a single genome. SqueezeMeta, which does not include a binning process, has higher contamination than the other frameworks; even though its assembly had higher completeness than DATMA for Actinobacteria-Streptomyces and higher completeness than MetaWRAP for Firmicutes-Aneurinibacillus. Finally, DATMA was the fastest tool.

4.4.3 Mock-Even community metagenome

We configured DATMA with default parameters to remove low-quality reads ($Q < 30$ and $length < 70$ bp), leaving a total of 1,371,533 reads after this stage. Rfam database [35] was used as a reference database to identify 16S rRNA ribosomal sequences. DATMA separated 67,600 reads that aligned to the 16S rRNA regions. The 1,303,933 leftover reads were aligned with CLAME starting with b=40bp and iterating with $b = 30bp$ and $b = 20bp$. We set DATMA to report only bins with more than 2000 reads.

Table 4.3 summarizes the number of bins generated, the assembly metrics, the total ORF detected, and the completeness-contamination level of the bins. It also compares DATMA results and performance versus MetaWRAP [113] and SqueezeMeta [105] frameworks. The results show that DATMA can obtain more than 60% of three predominant genomes (Deinococcus -Deinococcus, Proteobacteria-Acinetobacter, and Bacteroidetes-Bacteroides) with a contamination level less than 7%. MetaWRAP can recover most of the predominant genomes in the sample, all of them with completeness higher than 80% and contamination less than 1%, except for Firmicutes bacteria. SquuezeMeta, executed in sequential mode, shows a lower performance than the other tools. Because all the genomes are well-referenced, MetaWRAP overcomes the other used frameworks. This experiment shows the ability of DATMA to distinguish the reads from the predominant species in a short time (most of Deinococcus genome was recovered with 0.0% contamination level), but indicates the limitation

TABLE 4.3: DATMA report for the Mock-Even experiment

	Total Bins	Total Contigs per bin	Contigs metrics				Recovered genome			Time (m)
			Largest (bp)	N50 (bp)	Genome (Mbp)	ORFs	Complete -Ness (%)	Contami -nation (%)	Lineage	
DATMA	3	119	58301	4463	2.30	2277	54.68	0.00	Deinococcus-Deinococcus	97
		240	13747	3526	2.26	8612	89.08	6.98	Proteobacteria-Acinetobacter	
		84	10336	1109	3.16	5417	66.66	3.47	Bacteroidetes-Bacteroides	
MetaWRAP	5	126	122483	43957	3.95	4246	98.17	1.21	Proteobacteria-Acinetobacter	236
		90	110942	38746	2.54	2611	93.53	0.65	Actinobacteria-Cutibacterium	
		489	66633	11865	4.44	5095	89.31	0.90	Bacteroidetes-Bacteroides	
		655	31456	5870	2.75	3363	85.03	0.21	Deinococcus-Deinococcaceae	
		647	8351	2403	1.41	2589	56.85	6.22	Firmicutes-Streptococcus	
SqueezeMeta	NA (†)	36164	389	483	14.09	35713	100	505.16	Firmicutes	105
		26711	3894	558	11.69	26262	100	289.42	Proteobacteria	
		7245	4194	786	4.22	7170	88.87	10.99	Bacteroidetes	
		6579	4605	804	3.98	6420	100.00	70.01	Actinobacteria	
		5267	3504	630	2.32	5123	77.31	18.38	Deinococcus	

†We manually selected the contigs from the annotation report

of our tool to recover species in lessor abundance into the metagenome (only the most abundant were reported).

4.4.4 Brocadia caroliniensis metagenome

DATMA was executed with default parameters to remove low-quality bases and reads that were too short ($Q < 30$ and $length < 70$ bp). The 1,860,653 leftover reads were aligned against the Rfam database [35] to remove 16S rRNA gene sequences. After removing 12,754 reads, DATMA called CLAME with 1,847,899 sequences using $b = 70bp$, as the number of bases alignment parameter. The bins with more than 2000 reads were assembled with SPAdes [77].

Table 4.4 summarizes the number of bins generated, the assembly metrics, the total ORFs detected, the completeness-contamination of the bins, and the computational time used by DATMA. It also contrasts these results against the report produced by MetaWRAP [113] and SqueezeMeta [105] frameworks. MetaWRAP completeness of the Brocadia genome is higher than the obtained by DATMA; but, DATMA obtains a better N50. SqueezeMeta annotated most reads as Brocadicae family, but it generated a larger number of contigs than the other frameworks. DATMA was the fastest tool.

4.4.5 Computational performance

Figure 4.2 shows the execution time for all the datasets using several scenarios. It shows that computational time decreases as the number of workers increase. Fig 4.2 also illustrates the memory performance of DATMA. It reports a peak in the

TABLE 4.4: DATMA report for the *Brocadia caroliniensis* experiment

Tool	**Total Bins**	**Total Contigs per bin**	**Contigs metrics**				**Recovered genome**			**Time (m)**
			Largest (bp)	N50 (bp)	Genome (Mbp)	ORFs	Complete -ness (%)	Contami -nation (%)	Lineage	
DATMA	2	677	88819	18421	3.96	4330	93.96	10.05	Brocadiaceae	60
		1382	13527	2456	2.37	3656	47.95	1.78	Brocadiaceae	
MetaWRAP	2	607	58497	9402	3.67	4273	96.08	5.00	Brocadiaceae	135
		374	29910	10268	2.81	4015	77.30	1.75	Brocadiaceae	
Squeeze-Meta	NA(†)	10345	3264	519	4.13	10283	89.47	111.28	Brocadiaceae	85
		12753	3420	360	4.21	12607	74.76	100.00	Bacteroidetes	
		12698	4314	342	4.14	11916	65.33	84.78	Proteobacteria	

†We manually selected the contigs from the annotation report

binning stage, but it then decreases when DATMA distributes the next tasks into the available computing resources.

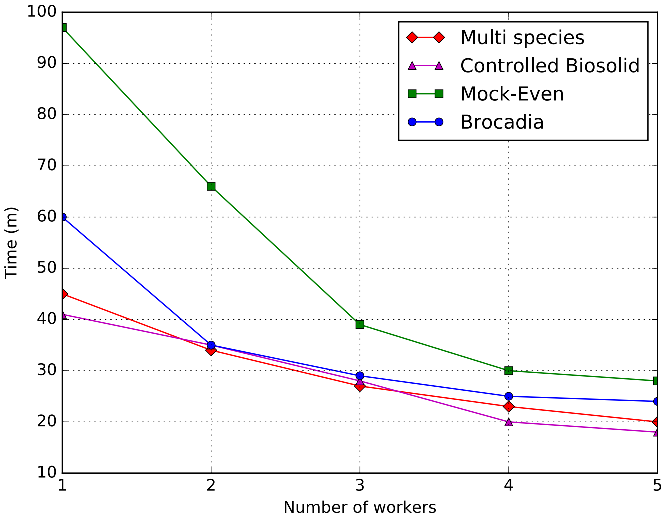


FIGURE 4.2: It illustrates the computational time of DATMA for all datasets using several workers

4.5 Conclusions

Distributed AuTomatic Metagenomic Assembly and Annotation framework (DATMA) is designed to address two typical challenges of metagenomic projects: i) metagenomics assembly, a complex task due to the mix of reads from several species, and ii) the computational time required to analyze the massive amount of data recovered with NGS technologies.

We showed DATMAs functionality using metagenomic samples with known species composition. It showed that DATMA automatically, using CLAME, effectively groups reads without mixing from different species. The controlled experiments also illustrated that in contrast with the other frameworks without the binning stage, the inclusion of a CLAME improves the assembly. We also show that

DATMA automatically detected the number of assembly-annotation tasks and distributed them into the computational resources, decreasing the time to analyze a complete dataset. We reported similar performance with the Mock-Even and Brucella metagenomes, in which DATMA produced comparable results than MetaWRAP and SqueezeMeta frameworks, but faster.

Even though exploiting parallelism from a problem is a complex task, we show that by using COMPSs, DATMA can run in parallel on several threads or better on different computing infrastructures. It is an essential feature of DATMA that difference our framework from traditional pipelines, which are typically built as standalone applications or bash scripts, and enable future studies of huge metagenomes. However, additional work needs to be done to get a versatile pipeline. Memory usage stays to be the primary constraint for our framework. Moreover, current DATMAs version includes the stages that we consider are the main into full metagenomics, but new tools will be included in next versions.

Chapter 5

Experimental Setup

In this Chapter, we show how CLAME and DATMA are used to study real metagenomes and extract the predominant species from them. First, we introduce the San Vicente hot spring metagenome, from which, we obtained a novel Xanthomonadaceae draft genome. Then we present the San Fernando wastewater biosolid metagenome, in which we extracted a novel Anareolinacea draft genome.

5.1 San Vicente hot spring metagenome

San Vicente is a hot spring within the Cerro-Machin-Cerro-Bravo volcanic complex in Colombian Andes, located at N4°50.25′ W75°32.35′ at an altitude of 1,715 masl. Waters with discharge temperatures above 60°C (max. 91°C), pH of 6.7 and high concentrations of chlorides characterize hot springs.

Hot spring bacteria have unique biological adaptations to survive the extreme conditions of these environments; these bacteria produce thermostable enzymes that traditionally are used in biotechnological and industrial applications. However, sequencing those bacteria is complicated, since it is not possible to culture them. As an alternative, genome shotgun sequencing of whole microbial communities can be used. The problem is that the classification of sequences within a metagenomic dataset is very challenging, mainly when they include unknown microorganisms since they lack genomic reference.

In this section, we show that CLAME allowed us to recover a high-quality draft genome of a Gammaproteobacteria closely related to Dokdonella genus, which seems to represent a new lineage within the family Rhodanobacteraceae. This draft genome was validated using several genomic strategies and summited on the NCBI's project PRJNA431299.

5.1.1 Methods

To reduce the complexity of the community, we incubated a sample of the San Vicente hot spring (discharge temperature 64°C) in a non-selective mineral medium, maintained at $45°$C with white light during 15 days. Then, we extracted the DNA community using PowerMax Soil DNA Isolation Kit supplied by MOBIO Corporation [23], following the instructions of the manufacturer. The sample was sequenced using ROCHEs 454 Titanium technology in 3/4 PTP at the Centro Nacional de Secuenciación Genómica - CNSG, Universidad de Antioquia, Medellin, Colombia. We recollected a total of 926,130 reads, with a 300bp average length.

We set DATMA to trim low-quality ($Q < 35$) reads and keep sequences with at least 70 bases long. CLAME was configured to start with 70 bp and iterate with 50, 40 and 20 base pairs. We set the bin size to 10,000 reads and selected SPAdes [77] tool to assembly the bins. Putative open reading frames (ORFs) were detected using Prodigal [44] tool. Taxonomic annotation for the contigs was developed using BLAST [2] and Kaiju [65] tools (both against a local NT database). Bin contamination was checked by detecting the presence of single-copy of essential genes using CheckM [85] tool. The complete configuration file for this metagenome is available in the DATMA GitHub.

The phylogenetic tree, built for the main bin reported by CLAME, was inferred by using the Maximum Likelihood method with the Jukes-Cantor model [48] and the process described by Brumm et al. [15]. We replied the Brumm et al., strategy to obtain the first tree(s), but our analysis involved 29 nucleotide sequences, instead of 26 samples. The ribosomal-sequences were manually curated, annotated and used to build an evolutionary tree. We conducted our study on MEGA 7.0 [55].

Finally, we used MG-RAST [120], MetaWRAP [113], and SqueezeMeta [105] frameworks to study the hot spring metagenome and compare our results. All the tools, except MG-Rast, were executed on a computer equipped with 64 Intel(R) Xeon(R) CPU X7560 @ 2.27GHz and 500 GB of RAM, and Linux-Centos OS. We set the number of threads to four for all the datasets and pipelines. Although SqueezeMeta includes a binning stage, it requires several metagenomics samples. Because we have only one DNA sample, we executed this tool in a sequential mode, which does not include the binning stage. We decided to use this framework to illustrate the execution of a pipeline without a binning phase.

5.1.2 Results

Figure 5.1 shows the Microscopic photograph of the water sample from the San Vicente hot spring. It shows that a filamentous Cyanobacterium dominated the sample, and several small cells suggest that a reduction in the complexity of the community was achieved after the enrichment of the sample at $45°$C for 15 days.

Table 5.1 shows the metrics of bins generated by CLAME, using $b = [70, 50, 40,$ and $20]$ bp, we reported only those with at least 10,000 reads. It shows that some 60% of the raw reads were binned into four main bins.

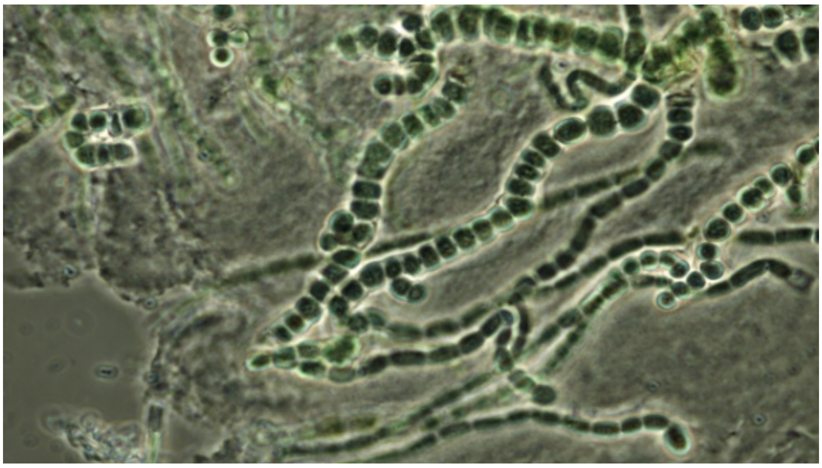


FIGURE 5.1: Microscopic photograph of cultured water from San Vicente hot spring

TABLE 5.1: DATMA report for the hot spring metagenome

b	Bin	Bin Size (Number of reads)	mean	std	Median	MAD	p=3std/mean	Outlier boundaries
70	0	361175	80.02	22.43	81	25.2	0.84	29-127
50	1	41177	12.70	4.63	12	4.45	1.09	3-25
40	2	30471	13.43	4.66	13	4.45	1.04	3-25
20	3	48317	97.85	31.26	99	32.61	0.95	16-188

Figure 5.2 shows the edge histogram, produced by CLAME, considering 70 bases alignment. It shows a normal-like distribution in the range of 30 to 130 edges. This range agrees with the DATMA reports using MAD statistics, which report a normal distribution in the field 29 to 127 edges for the large bin.

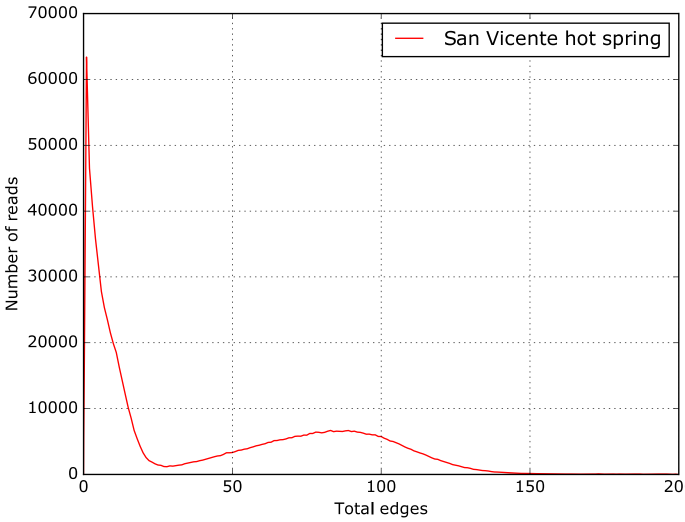


FIGURE 5.2: Number of edges histogram reported by CLAME for the hot spring metagenome

Table 5.2 summarizes the assembly metrics for the contigs generated from each

TABLE 5.2: Assembly metrics for the hot spring metagenome

CLAME (bp)	Bin	Size (reads)	bp	Contigs	Expected genome size (Mbp)	N50 (bp)	ORFS	Contamination	Completeness	Lineage
70	0	361175	142309134	251	2.95	24167	2675	1.2	83.69	Proteobacteria
50	1	41177	15008981	444	2.027	6770	2249	0.0	22.41	Cyanobacteria
40	2	30471	11091561	300	1.49	7861	1583	0.0	26.21	Cyanobacteria
20	3	48317	18655081	219	0.478	6040	465	0.0	17.24	Cyanobacteria
Total		486307	189189763	1245	2027	45846	7027	NA	NA	NA

bin using SPAdes assembler tool. It also shows the total of open reading frames (ORFs) detected by Prodigal from these contigs. We have included the CheckM report; it indicates the contamination level and genome completeness of each bin according to single-copy of universal genes.

We have included in the Table 5.2 the annotation report generated by BLAST (against a local NT) for the contigs produced using the reads from each bin. It indicates that most of the contigs from bin 0 belong to Proteobacteria phylum and Xanthomonadaceae family. CheckM report shows contamination of less than 2% for this bin. The number of contigs, the contamination level, the expected genome size (>2.0 Mbp) and completeness ration (>80%), show that the *Bin 0* is an excellent candidate to describe a Xanthomonadaceae genome. Although most of the contigs from *Bin 1*, *Bin 2*, and *Bin 3* belong to the Cyanobacteria phylum, they present a completeness ration less than 20% which is not enough to report a draft genome.

We focus our study on *Bin 0*. Figure 5.3 details the BLASTn report for this bin. It indicates that BLAST classified most of the contigs into the Xanthamonadaceae family belonging to the Proteobacteria phylum. Using the set of standards for the minimum information regarding a metagenomeassembled genome (MIMAG) proposed by Bowers et al. [13] and the results in Table 5.2 and Figure 5.3, we can use the contigs from the Bin0 to introduce a High-quality draft genome. We named our sequences as Colombian thermophile Xanthomonadaceae_UdeA_SF1 draft genome, and it was made public by submitting it to the NCBIs project PRJNA431299.

Figure 5.4 illustrates the phylogeny tree building from the 16S rRNA sequence of our Xanthomonadaceae_UdeA_SF1 genome and several families of Proteobacteria phylum. It confirms that our strain is closely related to several uncultured bacteria within the family Xanthomonadaceae of the Gammaproteobacteria. Besides, the phylogeny reconstructed only based on culture-type strains showed that the obtained 16S rRNA ribosomal sequence is consistently within Order Xanthomonadales, separated from the outgroup Alkanibacter difficilis Order Sinobacteriales and apart from the cluster composed by the Genus Dokdonella and other Xanthomonadales such as Rhodanobacter, Dyella, Aquimonas, and Pseudoxanthomonas.

Table 5.3 compares DATMA results against the report generated by MetaWRAP

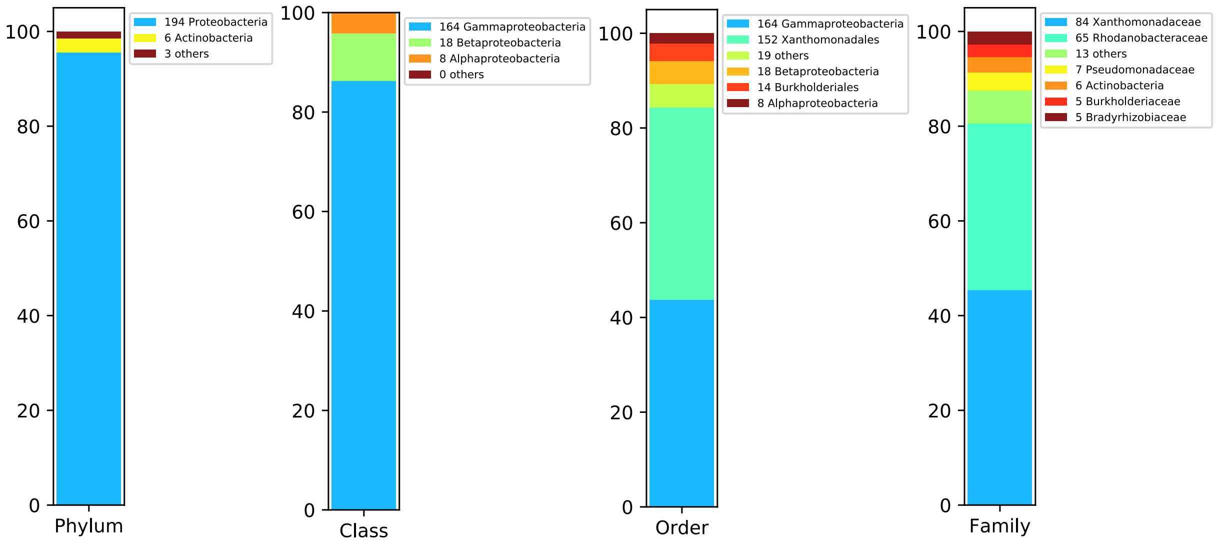


FIGURE 5.3: Taxonomic report for the contigs from the Bin0

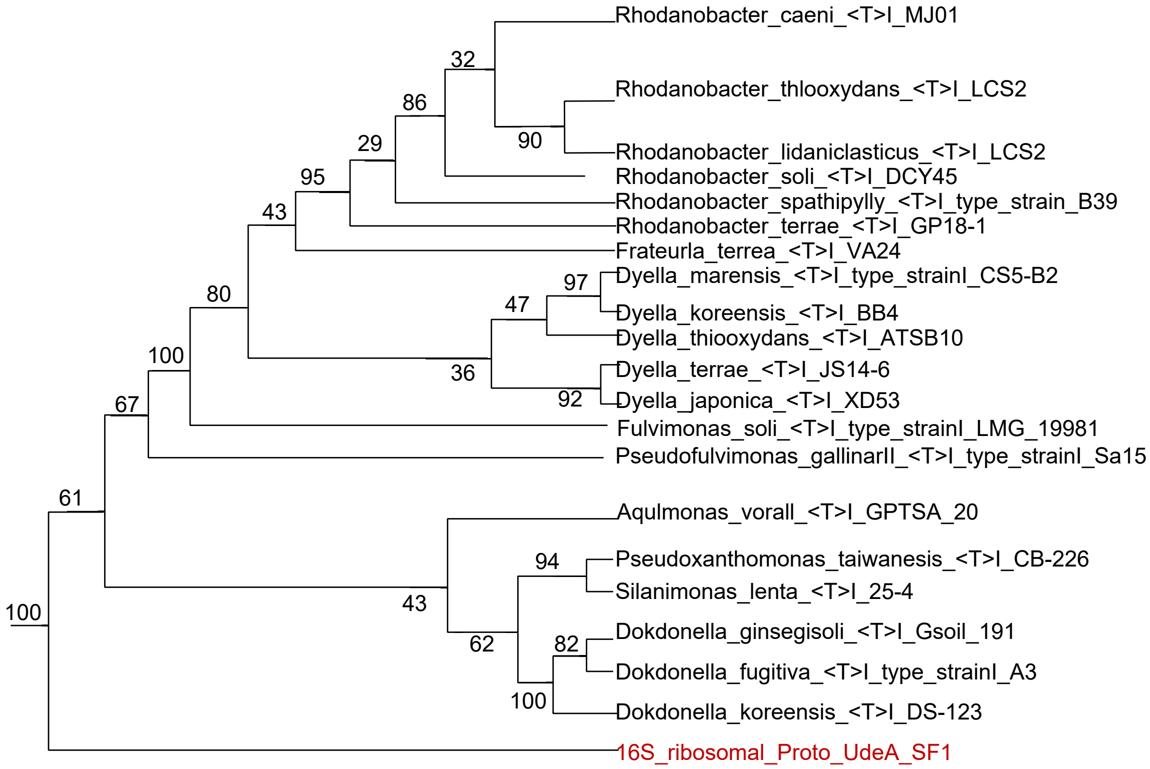

Rhodanobacter_caeni_<T>l_MJ01
Rhodanobacter_thlooxydans_<T>l_LCS2
Rhodanobacter_lidaniclasticus_<T>l_LCS2
Rhodanobacter_soli_<T>l_DCY45
Rhodanobacter_spathipylly_<T>l_type_strain_B39
Rhodanobacter_terrae_<T>l_GP18-1
Frateurla_terrea_<T>l_VA24
Dyella_marensis_<T>l_type_strainl_CS5-B2
Dyella_koreensis_<T>l_BB4
Dyella_thiooxydans_<T>l_ATSB10
Dyella_terrae_<T>l_JS14-6
Dyella_japonica_<T>l_XD53
Fulvimonas_soli_<T>l_type_strainl_LMG_19981
Pseudofulvimonas_gallinarIl_<T>l_type_strainl_Sa15
Aqulmonas_vorall_<T>l_GPTSA_20
Pseudoxanthomonas_taiwanesis_<T>l_CB-226
Silanimonas_lenta_<T>l_25-4
Dokdonella_ginsegisoli_<T>l_Gsoil_191
Dokdonella_fugitiva_<T>l_type_strainl_A3
Dokdonella_koreensis_<T>l_DS-123
16S_ribosomal_Proto_UdeA_SF1
32
86
90
29
95
43
97
47
80
36
92
100
67
61
100
94
43
62
82
100

FIGURE 5.4: Phylogenetic tree for the 16S-ribosomal assembled gene (16SProto marks whit red). The values in the branches indicate the percentage of replicate trees in which the associated taxa clustered together in the bootstrap test

[113], SqueezeMeta [105], and MG-RAST [120] frameworks. It shows that Xanthomonadeceae is the predominant family for all the tools, but only MetaWRAP and DATMA can recovery more than 80% of this genome. Although MetaWRAP shows superior completeness ration than DATMA, it also has a contamination level gather than our tool. We executed SqueezeMeta in sequential mode, which disables

TABLE 5.3: Analysis report for the hot spring metagenome using different metagenomic frameworks

	Total Bins	Total Contigs per bin	Contigs metrics				Recovered genome			Time (m)
			Largest (bp)	N50 (bp)	Genome (Mbp)	ORFs	Complete-Ness (%)	Contami-nation (%)	Lineage	
DATMA	4	251	99677	24167	2.95	2675	83.69	1.2	Proteobacteria-Xanthomonadales	27
		444	28143	6770	2.03	2249	22.41	0.0	Cyanobacteria	
		300	27804	7861	1.50	1583	26.21	0.0	Cyanobacteria	
		219	13966	6040	0.48	465	17.24	0.0	Cyanobacteria	
MetaWRAP	4	282	268735	97835	3.42	2975	96.35	3.04	Proteobacteria-Xanthomonadaceae	151
		466	138319	23444	7.22	8100	94.96	2.32	Cyanobacteria	
		370	42515	9098	2.44	2932	87.82	1.41	Actinobacteria-Microbacteriaceae	
		1371	15051	2664	3.23	5237	69.78	2.45	Cyanobacteria	
Squeeze-Meta	NA (†)	25218	6600	609	12.92	24771	100	324.09	Proteobacteria	79
		12449	6117	849	8.31	12404	98.12	85.02	Cyanobacteria	
		3750	4515	801	2.41	3692	88.55	7.13	Actinobacteria	
MG-Rast	NA (†)	73100(*)	NA	NA	NA	NA	NA	NA	Proteobacteria	1 week
		6748(*)							Actinobacteria	

*The values correspond to reads
†We manually selected the contigs from the annotation report

the binning stage and can explain its low performance. MG-Rast classified most of the reads into Proteobacteria and Actinobacteria phyla. However, DATMA reports a more significant number of sequences into the Proteobacteria species. Moreover, because we submitted the raw dataset as private, we only have access to the basic report of MG-Rast. We decided to conserve these results to evaluate the annotation report and explain the limitation of a web framework. The table also indicates that MetaWRAP overcomes the other tools to study the Cyanobacteria and Actinobacteria species, which is in minor abundance for this sample. It is a current limitation of our tool, which splinted the Cyanobacteria genome into three regions. However, our tool is who report 0% of contamination. Moreover, DATMA was the fastest tool.

5.2 San Fernando biosolid metagenome

Waste Water Treatment Plant (WWTP) San Fernando is located in Itagüí-Colombia and operated by the company Empresas Publicas de Medellín (EPM). This WWTP services a population of approximately 500,000 people and receives an influent flow of $1.8m^3/s$ of residential houses, hospital and industrial wastewater. Municipal wastewater treatment plant produces large amounts of sludge as a byproduct (Biosolid). The vast diversity of bacteria present in a biosolid makes that traditional biological methodologies are unsuitable for their identification and characterization.

Analysis of microbial communities in anaerobic reactors traditionally has been based on molecular tools such as denaturing gradient gel electrophoresis (DGGE), fluorescent in situ hybridization (FISH), and 16S rRNA clone libraries in bacterial plasmids [30]. However, these approaches cannot elucidate the whole complexity of the genetic and functional diversity in microbial structure [47]. Notwithstanding high-throughput sequencing technologies offer an effective method to characterize the phylogenetic composition and metabolic profiling in environmental samples, few studies have been made in activated sludge and biosolid samples using this sequencing method (i.e., [126], [53], [93]).

Below we show how DATMA allowed us studying the biosolid metagenome and recover a Low-quality draft genome that belongs the family Anaerolineaceae closely related to the genus Anaerolinea. A study of the microbial diversity, as well as the methanogenesis pathway of this metagenome, is presented in [7].

5.2.1 Methods

We collected two biosolid samples from municipal (WWTP) San Fernando, one of them in the rainy season ($9.1mm/h$ precipitation, average maximum temperature 27.8°C, average minimum temperature 17.1°C, August 2013) and the other in the dry season ($1.9mm/h$ precipitation, average maximum temperature 28°C, average minimum temperature 17.4°C, February 2012). Dewatered biosolids (about 500 g) were collected and transferred to the laboratory in refrigeration. The DNA extraction was done using PowerMax Soil DNA Isolation Kit supplied by MOBIO Corporation [23]. Then, the samples were sequenced using ROCHEs 454 Titanium technology in $3/4$ PTP at the Centro Nacional de Secuenciación Genómica-CNSG, Universidad de Antioquia, Medellin, Colombia. A total of 6,206,317 reads were analyzed.

We set DATMA with default parameters to remove low-quality sequences ($Q < 30$ and $length < 70$ bp). These resultant reads were aligned against the Rfam database [35] to identify 16S rRNA ribosomal sequences. The leftover sequences were binned with CLAME using default parameters but reporting bins with more than 5000 reads. We selected SPAdes [77] as the assembler tool. The whole configuration file for this dataset is available into the DATMA GitHub.

We focus the study on the main bin generated by CLAME. We assessed the assembly completeness of the contigs generated from this bin using CheckM [85] tool to detect the presence of single-copy essential genes. We built an evolutionary tree to complement the annotation report of BLAST. It was constructed using the ribosomal sequences for the bin. The tree was inferred by using the Maximum Likelihood method with the Jukes-Cantor model [48] and the process described by Brumm et al. [15]. We conserved the same number of replicates (500) and bootstrapped tree topology to represent the evolutionary history of the taxa analyzed. We used Brumm et al., strategy to obtain the first tree(s) but our analysis involved 29 nucleotide sequences, instead of 26 samples. We conducted our study on MEGA 7.0 tool [55].

Finally, we used MG-RAST [120], MetaWRAP [113], and SqueezeMeta [105] frameworks to study the biosolid metagenome and compare our results. All the metagenomic pipelines, except MG-RAST, were executed on a computer equipped with 64 Intel(R) Xeon(R) CPU X7560 @ 2.27GHz, 500 GB of RAM, and Linux-Centos OS. We

TABLE 5.4: DATMA report for the biosolid metagenome

CLAME (bp)	Bin	Size (reads)	bp	Contigs	Expected genome size (Mbp)	N50 (bp)	ORFS	Completeness	Contamination	Strain heterogeneity
70	0	115754	43499518	2337	3.58	2097	5332	58.83	148.56	79.16
	1	84476	35342229	329	1.71	7897	1929	67.24	32.49	98.08
50	2	396449	138490533	4077	13.81	5346	16607	95.45	235.28	14.26
	3	14735	5528036	9	0.060	10963	112	0.00	0.00	0.00
25	4	10740	3853093	196	0.37	2155	570	0.00	0.00	0.00
	5	12621	4282265	7	15967	3014	18	0.00	0.00	0.00
Total		634775	230995674	6955	15984.82	31472	24568	NA	NA	NA

set the number of threads to four for all the datasets and pipelines. We configured SqueezeMeta in merge mode using the two metagenomic samples (rainy and dry) by separated; it enables the binning stage. However, it generated a No-consensus output in the merge stage. We reconfigured it in sequential mode and executed on all the datasets.

5.2.2 Results

DATMA left 5,668,260 reads after the quality control stage. A total of 54,931 sequences were automatically identified as 16S rRNA reads and separated from the dataset. The 5,613,329 leftover sequences were binned with CLAME. Table 5.5 shows the number of bins, with at least 30,000 reads, and the results after assembling those using SPAdes [77]. We have included the CheckM [85] report, to indicate the contamination level and completeness ration of each bin. According to MIMAG standards [13] to report a genome, only Bin0 and Bin1 have suitable results to propose a draft genome. We focus or study over these two predominant bins.

Figure 5.5 shows the BLASTn report for the assembled contigs from the Bin0. It indicates that some 43% of the contigs were classified into Chloroflexi phylum, but only close to 38% of them were annotated into a single phylum-family clade. In this case, the contamination level is too high to propose a draft genome.

Figure 5.6 shows DATMAs annotation report using the BLAST [2] tool for the second bin (Bin 1). It indicates that BLAST annotated most of the contigs into the Chloroflexi phylum and Anaerolineaceae family. Moreover, the relation between the number of ORFs and the genome estimation (1 ORF per Kbp) agrees with the relationship reported for this kind of species (i.e., Pelolinea submarina with 3131 ORFs, 3.5 Mbp and a relation of 0.89 ORFs/Kbp and Leptolinea tardivitalis with 3301 ORFs, 3.69 Mbp and a relation of 0.90 ORFs/Kbp).

CheckM [85] report for this bin indicated that 60% of Universal Single-Copy Orthologs are in the contigs. It also shows a contamination level of the 32%, but the strain-heterogeneity index (92%) indicates that most markers present appear to be from closely related organisms. We highlight that our observation suggests that it

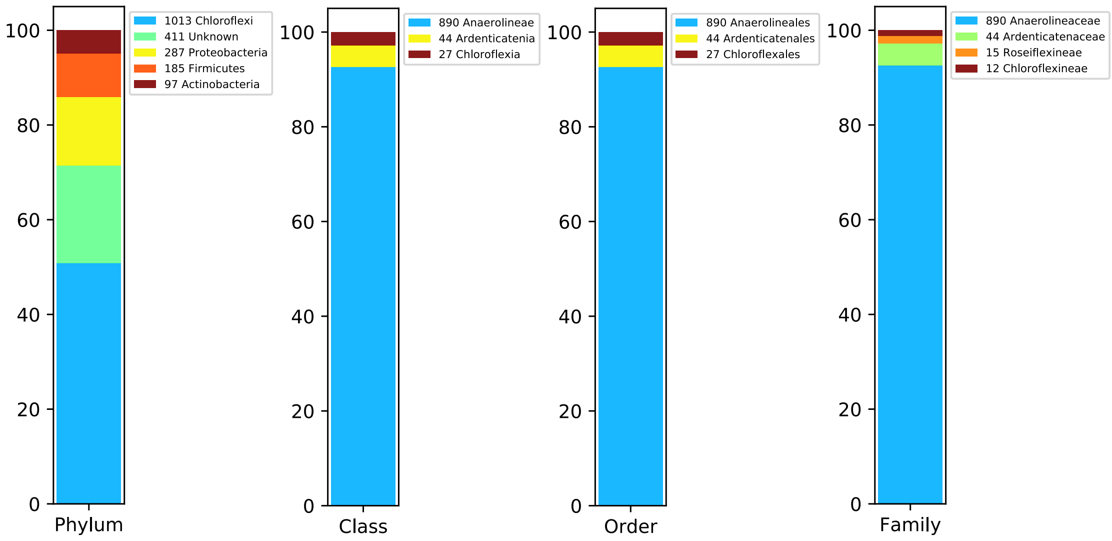


FIGURE 5.5: Taxonomic report for the contigs from the Bin0 for the biosolid metagenome

is a novel genome without a near reference. According to the set of standards for the minimum information regarding a metagenome-assembled genome (MIMAG) proposed by Bowers et al. [13], the contigs and their metrics are enough to describe a Low-quality draft genome belongs to the Anaerolineaceae family. We called this draft genome Anaerolineaceae_UdeA_SF1 and submitted it into the NCBIs project PRJNA529916.

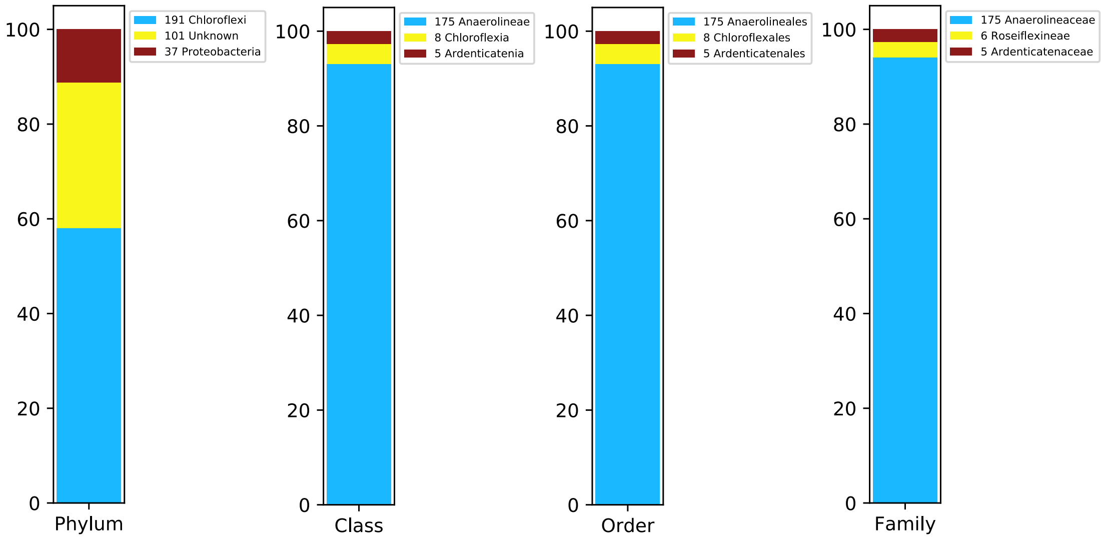


FIGURE 5.6: Taxonomic report for the contigs from the Bin1 for the Biosolid metagenome

To improve the taxonomic annotation, we used MEGA 7.0 [39] to build a phylogenetic tree using the 16S rRNA sequences for this bin and the Ribosomal data project database [18]. The evolutionary tree, in Figure 5.7, indicates that the recovered reads are close to the family Anaerolineaceae, and it has a relation with the genus Pelolinea and Leptolinea.

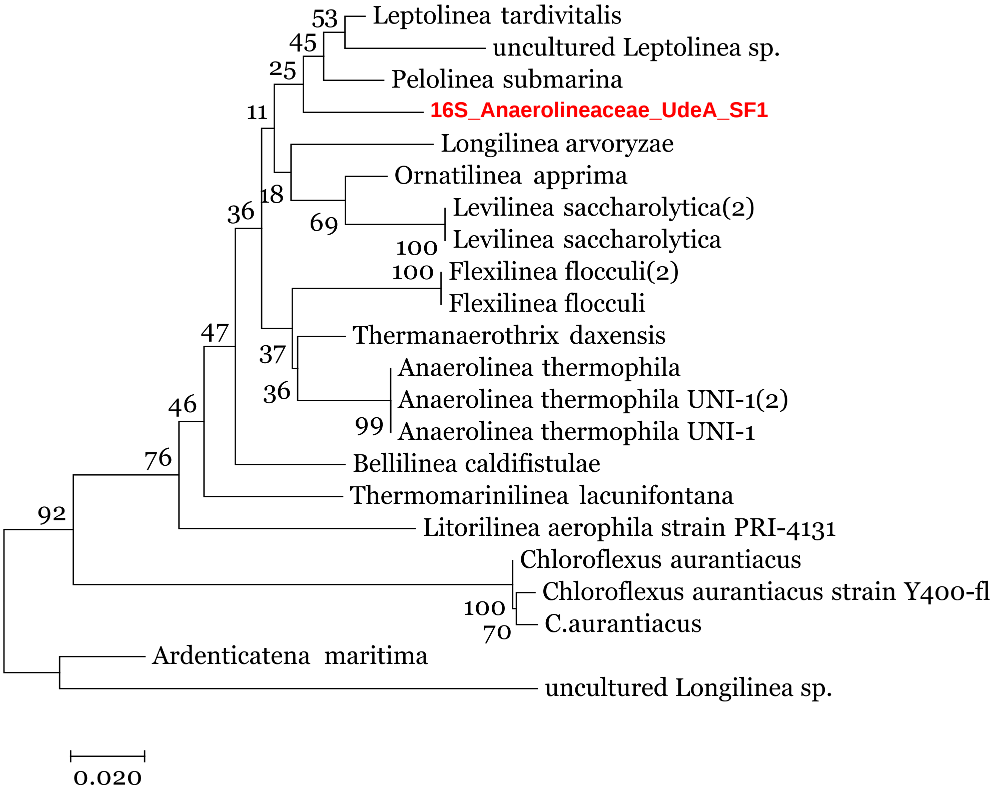

53
45
25
11
18
36
69
100
100
47
37
36
46
99
76
92
100
70
Leptolinea tardivitalis
uncultured Leptolinea sp.
Pelolinea submarina
16S_Anaerolineaceae_UdeA_SF1
Longilinea arvoryzae
Ornatilinea apprima
Levilinea saccharolytica(2)
Levilinea saccharolytica
Flexilinea flocculi(2)
Flexilinea flocculi
Thermanaerothrix daxensis
Anaerolinea thermophila
Anaerolinea thermophila UNI-1(2)
Anaerolinea thermophila UNI-1
Bellilinea caldifistulae
Thermomarinilinea lacunifontana
Litorilinea aerophila strain PRI-4131
Chloroflexus aurantiacus
Chloroflexus aurantiacus strain Y400-fl
C.aurantiacus
Ardenticatena maritima
uncultured Longilinea sp.
0.020

FIGURE 5.7: Phylogenetic tree for the 16S-ribosomal gene (16S_-Anaerolineaceae_UdeA_SF1). The values in the branches indicate the percentage of replicate trees in which the associated taxa clustered together in the bootstrap test

Table 5.5 compares the assembly results of DATMA, MG-RAST [120], MetaWRAP [113] and SqueezeMeta [105] frameworks. It shows that MetaWRAP reports eight genomes with some 80% completeness ration and contamination level less than 7%. However, most of them cannot be assigned with precision into a family clade, and most important any bin belongs to Anaerolineaceae family. SqueezeMeta shows that Proteobacteria is the dominant phylum, but any bin belongs Chlorofexi. MG-RAST indicates that most of the reads classify into of Pseudomonadaceae and Anaerolineaceae families, but because we submitted the data as a private project, any additional information could be recollected. For this experiment, DATMA was the fastest tool and the only tool which can recover a draft genome.

5.3 Conclusions

In this chapter, we show that using DATMA, the reads belong to the predominant species from two real metagenomes can be binned and the respective draft genomes

TABLE 5.5: Analysis report for the Biosolid metagenome using different metagenomic frameworks

	Total Bins	Total Contigs per bin	Contigs metrics				Recovered genome			Time (m)
			Largest (bp)	N50 (bp)	Genome (Mbp)	ORFs	Complete -ness (%)	Contami -nation (%)	Lineage	
DATMA	2	2337	11791	2097	3.58	5332	58.83	148.56 (79.16%)||	Chloroflexi-Anaerolineaceae	125
		329	43084	7897	1.71	1929	67.24	32.49 (98.08%)||	Chloroflexi-Anaerolineaceae	
MetaWRAP	8	495	87496	17399	4.92	4266	94.59	4.73	Bacteria	485
		157	82800	18931	2.10	2110	89.03	1.69	Bacteria	
		218	60788	20930	2.72	2954	88.70	3.22	Proteobacteria	
		463	34164	7341	2.57	3031	87.16	1.32	Bacteria	
		731	22922	4571	2.78	3293	85.49	3.01	Actinobacteria	
		994	23123	4103	3.58	4481	84.98	6.70	Proteobacteria-Pseudomonas	
		420	26523	6363	2.17	2969	83.09	1.11	Gammaproteobacteria	
		754	19037	5384	3.33	3944	82.64	3.61	Proteobacteria-Pseudomonadaceae	
Squeeze-Meta	NA †	204323	11961	528	93.69	46730	100	2844	Proteobacteria	626
		49288	5376	579	24.30	49227	95.83	1258	Firmicutes	
		47728	5055	519	21.66	46730	100	675	Actinobacteria	
		41526	9084	585	20.69	41342	100	659	Bacteroidetes	
MG-Rast	NA †	114806* 95148*	NA	NA	NA	NA	NA	NA	Pseudomonadaceae Anaerolineaceae	1 week

|| Strain-heterogeneity index

*The values correspond to reads

†We manually selected the contigs from the annotation report

obtained. They were validated further studying the assembly, and we proposed Xanthomonadaceae_UdeA_SF1 and Anaerolineaceae_UdeA_SF1 draft genomes.

Xanthomonadaceae_UdeA_SF1 genome is around 3 Mbp, with 2,726 predicted ORFs; it is a small genome size, compared to Dokdonella and Dyella species, both with genomes around 4.5 Mbp and 3,519 and 3,966 annotated proteins, respectively. CheckM results showed that although the genome is not complete, it has an estimation of 80% completeness, which is adequate to present a high-quality genome according to MIMGAG parameters. BLAST annotation indicated that there are not a very close species to Xanthomonadaceae_UdeA_SF1. It means that our genome is candidatus for a new species. The evolutionary tree confirmed that the genome seems to be from a novel lineage within the family Rhodanobacteraceae of the class Gammaproteobacteria, closely related to the genus Dokdonella.

DATMA also showed a suitable performance to study complex metagenomes as the San Fernando biosolid dataset. It indicates that Proteobacteria is the dominant phylum; however, there are several families into it. Chloroflexy is not the dominant phylum, but it contains the predominant genome. It was detected by CLAME that grouped most of the reads of this genome into a bin, then DATMA used SPAdes and Kaiju tools to assemble and annotate it as an Anaerolineaceae family. This annotation was corroborated using the 16S rRNA gene phylogenetic analysis. It showed that DATMA extracted most reads of a novel taxon of the family Anaerolineaceae of the class Anaerolineae, closely related to the genus Pelolinea and Leptolinea.

We observed that other metagenomic frameworks show similar results than our

DATMA pipeline. In particular, for the two presented experiments, the tools that include a binning stage showed better performance than those without this phase. MetaWRAP and DATMA showed suitable performance to recover the abundant species, but DATMA presented the bin with minor contamination. MetaWRAP is better than the other frameworks for studying the species in low abundance concerning the hot spring metagenome. DATMA is better than the used tools to analyze the biosolid metagenome and was the fastest tool in all the cases.

MG-RAST requires zero computing power, but it needs to submit the data as public to access advanced studies, which can be forbidden for some projects. MetaWRAP, SqueezeMeta, and DATMA can run on a local computer, but since DATMA split the data into consistent bins and enables the parallel study of the dataset, it results in a reduced time of analysis. We discuss the advantage of our methodology in Chapter 7 and describe several future studies to improve the current limitation in Chapter 8.

Chapter 6

Conclusions

In this dissertation, we have presented an algorithm and a framework to analyze metagenomic datasets. Our main contribution is the design of an efficient method, called CLAME, that groups metagenome reads from the same molecule into bins. We have also integrated CLAME into a full pipeline, DATMA, which allows studying complex metagenomes using multi-core processors and several computers(when available). Our binning approach and complete framework are publically available and have been assessed using controlled and real metagenomes.

CLAME creates a graph representation of the metagenome, where reads are the nodes, and the connections represent the reads with highly similar DNA composition. Later a statistical analysis separates the graph and produces bins. We show that this methodology bins metagenomic reads without the need of a reference genome. This feature is essential since most of the unculturable microorganisms do not have reference genomes. A central limitation in this kind of binning methods is the time necessary to align the reads. We showed that CLAME, using an FM-index representation of the metagenome and proper a multi-threaded search algorithm, produces bins with similar precision that other state-of-the-art alignment tools but faster.

DATMA integrates CLAME binning tool with other state-of-the-art omic's tools and enables full analysis of metagenomic datasets. It analyzes CLAME's bins using several instantiations into a single computer or distributing them into the different computing resources. We showed that based on this strategy, DATMA pipeline provides assembly and annotation faster, and in many cases, better than similar metagenomic frameworks.

We showed DATMA functionality analyzing complex metagenomes and recovered from them most of their species and, more importantly, automatically extracted an almost complete genome from the predominant species. Therefore, DATMA can be used to improve the metagenomic analysis by grouping reads from DNA fragments of novel species, such as the Xanthomonadal genome presented in the hot sprint metagenome and the Anaerolineacea genome present in the biosolid metagenome. These draft genomes are one of the first species members of their families, and it was only possible to obtain them thanks to CLAME and DATMA.

Chapter 7

Future Work

Although CLAME and DATMA show a proper performance, in contrast with the other state of the art tool in most experiments, much remains to be done.

One of the main limitations of CLAME is memory occupancy. FM-index structure and the format used to represent the resulting overlaps are memory consume. Therefore, it is necessary to study alternative approaches to reduce the amount of RAM needed, like those used by BWA and Bowtie, which uses a similar structure, but that requires less memory. MatrixQuery container used to save the overlaps is a sparse matrix; hence, it can be stored using a compressed format designed for this kind of matrices. However, because it is a dynamic matrix, its size is only computed during execution, it requires a suitable strategy to insert elements. All these modifications are necessary to use CLAME in a computer with RAM constraints and bigger datasets.

Besides, CLAME methodology, based on the sequences abundance, can be unsuitable for experiments in which the species contribution is equality distributed. We have observed that the main effect of our binning approach on these experiments is splinted the raw reads into several bins. We have perceived that contig-binning methods perform better in these datasets. However, they require a metagenomic assembly, which is challenging. Future strategies can be oriented to bin the raw reads with CLAME, assemble the bins individually, (it reduces the assembly requirements), and cluster the contigs using contig-base tools.

On the other hand, since DATMA distributes the bins to be assembled and annotated, two consume time tasks, into several computers; it showed the best time to study datasets. However, its performance is limited to the number of bins generated by CLAME. This restriction makes DATMA unsuitable for experiments in which the species abundance or sequencing depth are not enough to create enough groups to require all the computational resources. Moreover, because DATMA only distributed the tasks after the binning stage, the parallelism is confined to the last steps of the framework. Future versions of DATMA can be adapted to use the complete computing structure to develop all the stages within the pipeline.

Finally, we will continue improving our tools to accommodate fast-growing technologies, including new stages in our pipeline and studying complex dataset. We have started to explore the Critical Assessment of Metagenome Interpretation (CAMI) dataset [98]. A challenging dataset that evaluates methods in metagenomics independently, comprehensively, and without bias. We hope that the result of this dissertation help researchers to study complex metagenomes and discover novel species from them.

Appendix A

Pseudocode algorithms of CLAME

Algorithm A.1 Main functions of CLAME

```
Example of DNA-sequences file in Fasta format
>R1
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTC
>R2
CTCCTGACTTTCCTCG

procedure Main
    Input: sequencesFile
    Input: CLAME_parameters
    Let: bases, QV, MatrixQuery: Matrices of size n, with n number of reads
    goto: readDNA_sequencesFile(sequencesFile, bases)
    goto: alignment(bases, parameters, MatrixList)
    goto: binning (parameters,queryList,MatrixList)
end procedure
```

Algorithm A.2 Read DNA-sequences

```
procedure readDNA_sequencesFile(sequencesFile,bases)
    String Line
    Open(sequencesFile)
    while Not EOF(sequencesFile) do
        Input Line
        if line Not Startwith '>' then
            for all bp in line do
                bases[i]=bp
                i=i+1
            end for
        else
            bases[i]='&'
            i=i+1
        end if
    end while
    Close(sequencesFile)
end procedure
```

Algorithm A.3 Read alignment stage

```
procedure ALIGNMENT(BASES,PARAMETERS,MATRIXQUERY)
    Declare n= parameters.totalReads
    Declare t= parameters.cpus
    Declare seedSize=parameters.b
▷
▷ FMindex generation
    Declare String S
    S=&bases[0]
    FM_index=genFM9(S)
▷
▷ Multithread Backward search
    Declare String Q
    for i=0; i<n; i=i+t do
        for all tread in t do
            Q=substring(bases[i+ThreadID],seedSize)
            MatrixQuery[i+ThreadID]=map2FM9(Q,FM_index)
        end for
    end for
end procedure
```

Algorithm A.4 Subgraph traversal and bin generation

```
procedure BINNING(PARAMETERS, MATRIXQUERY)
    Declare n= parameters.totalReads
    Declare tol=paremeters.tolerance
    Declare Qv, Sv
    Declare get=&Sv[0], put=&Sv[0]
▷
▷ Graph traversal
    for i=0; i<n; i=i+1 do
        if NOt i IN Qv then
            Qv.append(i)
            *put++=i
            while get < put do
                edges=(MatrixQuery[*get++])
                for e in edges do
                    if NOt e IN Qv then
                        Qv.append(e)
                        *put++=(e)
                    end if
                end for
            end while

            do                                          ▷ Edge analysis stage
                p=MAD(Sv)
            while abs(p)>tol
            delete get, put
        end if
    end for
end procedure
```

Appendix B

Auxiliar Tables

TABLE B.1: Edge analysis example using MAD to detect outliers. Let us consider the adjacency list X_i, which indicates the number of edges per node. It has an original $mean = 3.4$, a standard deviation $std = 1.46$, and a median $M_j = 3$. The $p - value = 1.28$ indicates a non-normal distribution. Edge analysis stage subtracts the median from each observation to get the new median $M_i = 1$. It will be multiplied by 1.4826 to find a $MAD = 1.48$ (Eq. 3.1 and Eq. 3.2). MAD reports that the read R13, with total edges, equal 8, is an outlier (according to the Eq. 3.3) and removes it. Removing this point the new statistic parameters are: $mean = 3.3$, $std = 1.03$ and $p-value = 0.93$. The new $p - value$ is close to one, which indicates a near-normal distribution and stops the Edge analysis process

Read	x_i	M_j	$abs(x_i - M_j)$	M_i	**MAD**	$(xi - Mj)/MAD > \|\pm 3\|$	**outlier**
R0	3	3	0	1	1.4826	0	NO
R1	4		1			0.67	NO
R2	2		1			0.67	NO
R6	3		0			0	NO
R7	5		2			1.35	NO
R12	3		0			0	NO
R13	**8**		**5**			**3.37**	**YES**
R9	2		1			0.67	NO
R10	3		0			0	NO
R14	4		1			0.67	NO
R16	3		0			0	NO
R3	4		1			0.67	NO
R5	2		1			0.67	NO
R11	2		1			0.67	NO
R13	4		1			0.67	NO
R15	3		0			0	NO
R17	3		0			0	NO
mean	3.41	3.33					
std	1.46	1.03					
p=3std/mean	1.28	0.92					

TABLE B.2: Species and total reads used to create the simulated multispecies metagenome

Species	NCBI reference	Phylum/Class	Total reads	Total bases (Mbp)	Used reads	Used bases (Mpb)	Genome size (Mpb)	Depth(x)
Synechocystis	DRR 106442	Cyanobacteria Cyanobacteria	589,689	21.9	112,805	41.5	3.5	11.7
Dokdonella	SRR 4217676	Proteobacteria Gammaproteobacteria	376,022	80.5	376,022	80.5	4.6	17.41
Hymnobacter	SRR 1334914	Bacteroidetes Cytophagia	2,917,298	958.5	37,599	12.3	5.0	2.4
Microbacteriaceae	SRR 5493999	Actinobacteria Actinobacteria	1,815,433	382.4	37,599	7.9	3.2	2.4
Rhizobium	SRR 5165471	Proteobacteria Alphaproteobacteria	1,152,754	242.2	37,599	7.9	4.5	1.7
	TOTAL		965,711	1685.5	601,624	150.1	20.8	NA

TABLE B.3: Taxonomic composition for the Mock-Even metagenome

Organism	Rank	Total reads	Total bases (Mbp)	Percentage
Deinococcus-Deinococcus	species	486683	249.1	35%
Proteobacteria-Acinetobacter	species	161464	84.9	12%
Bacteroides	species	113507	59.4	8%
Firmicutes-Staphylococcus	species	93820	49.9	7%
Actinobateria-Propionibacterium	genus	78223	39.9	6%
Other organisms	NA	452501	252	33%
Total		1386198	734.9	100%

TABLE B.4: List of available 16S rRNA databases for the 16S-identification stage

Database	Lab	Version	Num Seq	Size
NCBI	NCBI, USA	2018	19757	30 MB
RDP	Mothur, USA	2016	13212	20 MB
Greengenes	Greengenes Database Consortium	2013	1262986	1740MB
Rfam	EMBL-EBI, UK	2017	2319743	527 MB
RNAmmer	DTU, Denmark	2007	12260	19 MB
SILVA	Microbial Genomics and Bioinformatics Research Group, Germany	2017	1861373*	2764MB

* We conserved only bacteria sequences

TABLE B.5: Taxonomic composition for the simulated biosolid metagenome

Species	NCBI reference	Phylum/Class	Total reads	Total bases (Mbp)	Used reads	Used bases (Mpb)	Genome size (Mpb)
Streptomyces-albus	SRR-7080885	Actinobacteria Actinobacteria	2136790	2000	200000	30.2	7.63
Pelolinea-submarina	SRR-7174333	Chloroflexi Anaerolineae	2313660	1200	400000	60.4	3.52
Prochlorococcus.sp	SRR-7041236	Cyanobacteria Cyanobacteria	1863742	2000	200000	28.1	1.18 *
Aneurinibacillus-soli	SRR-7178569	Firmicutes Bacilli	3135417	1700	200000	30.2	4.12
Pseudomonas-fluorescens	SRR-7168455	Proteobacteria Gammaproteobacteria	1615297	1000	600000	90.6	6.85
	Total		11064906	7900	1600000	239.5	23.3

* We used a draft genome

TABLE B.6: Computer specifications for the servers used in our grid computing

	CPUs	CPU model name	RAM (GB)
Master	14	Intel(R) Xeon(R) CPU E52620 @ 2.0 GHz	99
Worker1	64	Intel(R) Xeon(R) CPU X7560@ 2.27 GHz	500
Worker2	80	Intel(R) Xeon(R) CPU E7- 4870@ 2.4 GHz	69

Appendix C

FM-index Construction

The next section illustrates the construction of an FM-Index structure, using metagenomic sequences.

C.1 Suffix array

If S' is a string of length $|S'|$, over the alphabet $\sum$, (i.e., $\sum = A, C, G, T$ for DNA sequences), \$ a character that not is in $\sum$, and $S = S'\$$ the string resulting from appending \$ to S'. The suffix array of a string S denoted SA_S, is a permutation of the integers $1, 2, ...|S|$ such that $SA_S[i] = j$ iff $S[j, |S|]$ is the i_{th} lexicographically lowest suffix of S.

A suffix array is constructed for a string $S[1 :: |S|]$, by building an array of pointers to all suffixes $suff[1 :: |S|]$, $suff[2 :: |S|]$, ..., $suff[|S| :: |S|]$, and sorting these pointers by the lexicographical (i.e., alphabetical) ordering of their associated suffixes. Table C.1 shows the corresponding suffix array and its construction for the sequence "AGGAATGGCC." A formal definition and creation of suffix arrays can be found in [50].

TABLE C.1: Suffix array SA_S, for the sequence AGGAATGGCC

Index	Suffixes	SAs	Suffixes alphabetic order
1	AGGAATGGCC$	11	$AGGAATGGCC
2	GGAATGGCC$A	4	AATGGCC$AGG
3	GAATGGCC$AG	1	AGGAATGGCC$
4	AATGGCC$AGG	5	ATGGCC$AGGA
5	ATGGCC$AGGA	10	C$AGGAATGGC
6	TGGCC$AGGAA	9	CC$AGGAATGG
7	GGCC$AGGAAT	3	GAATGGCC$AG
8	GCC$AGGAATG	8	GCC$AGGAATG
9	CC$AGGAATGG	2	GGAATGGCC$A
10	C$AGGAATGGC	7	GGCC$AGGAAT
11	$AGGAATGGCC	6	TGGCC$AGGAA
SAs= [11, 4, 1, 5, 10, 9, 3, 8, 2, 7, 6]			

C.2 Burrows-Wheeler Transform (BWT)

Burrows-Wheeler transform (BWT) of a string S, denoted B_S, is a permutation of the symbols of S such that: $B_S[i] = S[SA_S[i] - 1]$, and $B_S[1] = \$$, that is, the ith symbol of the BWT is the symbol prior to the ith suffix in the SA_S. It is similar to take the last column from the sorting pointers in the SA_S construction. Table C.2 shows the corresponding BWT B_S for the sequence AGGAATGGCC and its suffix array SA_S.

TABLE C.2: Burrows-Wheeler Transform (BWT) for the sequence AGGAATGGCC

Index	SAs	Suffixes alphabetic order	Bs
1	11	$AGGAATGGCC	C
2	4	AATGGCC$AGG	G
3	1	AGGAATGGCC$	$
4	5	ATGGCC$AGGA	A
5	10	C$AGGAATGGC	C
6	9	CC$AGGAATGG	G
7	3	GAATGGCC$AG	G
8	8	GCC$AGGAATG	G
9	2	GGAATGGCC$A	A
10	7	GGCC$AGGAAT	T
11	6	TGGCC$AGGAA	A
Bs=[C,G, $, A, C, G, G, G, A, T, A]			

C.3 FM-index structure

An FM-index, in Table C.3, is a data structure representation to fast substring queries. An FM-index is created by computing the BWT and adding two additional data structures:

- $C[c]$ a table that, for each character c in the alphabet, contains the number of occurrences of lexically smaller characters in the text. From the suffix alphabetic order, it corresponds to the index menus one, in which the first character c occurs (see the Index and the Suffixes alphabetic order rows in Table C.2).
- $O_{cc}(c, k)$ a table that contains the number of times symbol c appears in the range $B_S[1; i]$.

C.4 Text reconstruction from the FM-index structure

Using the FM-index is possible to produce the original string by sorting the BS representation, tracing a path from the last prefix to the first prefix and conserving the corresponding Bs symbol. Figure C.1 illustrates this process.

TABLE C.3: FM-Index representation for the sequence AGGAATGGCC. Upper) B_S representation, center) $C[c]$ occurrence table, and lower) frequency table for each character

Bs=[C, G, $, A, C, G, G, G, A, T, A]

c	**$**	**A**	**C**	**G**	**T**
C(c)	0	1	4	6	10

Occ(c,k)		**Bs[1;i]**											
			C	**G**	**$**	**A**	**C**	**G**	**G**	**G**	**A**	**T**	**A**
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
c	**$**	0	0	0	1	1	1	1	1	1	1	1	1
	A	0	0	0	0	1	1	1	1	1	2	2	3
	C	0	1	1	1	1	2	2	2	2	2	2	2
	G	0	0	1	1	1	1	2	3	4	4	4	4
	T	0	0	0	0	0	0	0	0	0	0	1	1

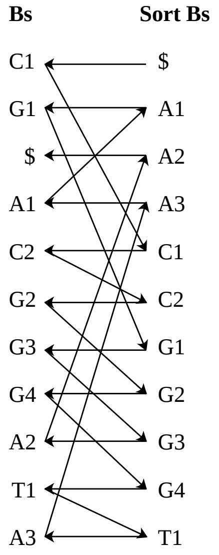


FIGURE C.1: Reversible process to get the original string from its BWT. It shows the path $,C1,C2,G2,G4,T1,A3,A1,G1,G3,A2

C.5 Backward search

Backward search, in Eq. C.1, allows mapping any substring P into the original string S using an FM-index structure.

$$\begin{bmatrix} top = C[P[i]] + O_{cc}(top-1; P|i|) + 1 \\ bottom = C[P[i]] + O_{cc}(bottom; P|i|) \end{bmatrix} \qquad \text{(Eq. C.1)}$$

top and *bottom* indicate the starting and ending point in the suffix array; i is a counter from the last to the first character of P. For the first iteration *top* and *bottom* correspond to the index for the entire structure. Final value for each pointer indicate the range in which the pattern P is a prefix of S.

The range size shows the number of times that P pattern is a prefix of S. If the range becomes empty or the range boundaries cross each other means that the pattern does not occur on S. The corresponding suffix array indicates that P is the i_{th} prefix of size $|P|$ for S.

Table C.4 illustrates the Backward search for the sequence $P = AGG$ and the FM-index $B_S = [C, G, \$, A, C, G, G, G, A, T, A]$. It shows that the substring P is a prefix in the range [3:3] of S. The size range indicates that P occurs once into the string S. From Table C.2 the corresponding suffix array shows that P is the first prefix of size three for S.

TABLE C.4: Backward search for the pattern AGG

i	P	Backward search		Range	
3	G	*top*	= C[G] + Occ(0;G) + 1	= 6+0+1	=7
		bottom	= C[G] + Occ(11; G)	= 6+4	=10
2	G	*top*	= C[G] + Occ(6;G) + 1	= 6+2+1	=9
		bottom	= C[G] + Occ(10; G)	= 6+4	=10
1	A	*top*	= C[A] + Occ(8;A) + 1	= 1+1+1	**=3**
		bottom	= C[A] + Occ(10; A)	= 1+2	**=3**

C.6 Wavelet Tree

Wavelet Tree (WT) is a data-structure that converts strings into balanced binary-trees to offer reduced select and rank times, primary operations for querying sequences inside the FM-index. WT is formed by recurrent binary assignation for each middle of the text. Left branches contain cero symbols, and right leaves carry the one symbols. Figure C.2 illustrates the Wavelet Tree construction for the BWT showed in Table C.2. Figure C.3 demonstrates the rank query $O_{cc}(c)$ calculation from the Wavelet Tree.

C.7 Huffman Wavelet Tree (HWT)

Mĺakinen and Navarro describe a Huffman Shaped Wavelet Tree based on the frequency of symbols. Characters with higher rates are placed in the tree in such a way that the path from the root to a leaf corresponds to the binary Huffman Code of the symbol of that leaf. It decreases query time massively for symbols with high frequency, which for uniform data would result in higher average query time. A complete description of Huffman Shaped Wavelet Tree process is described in [18].

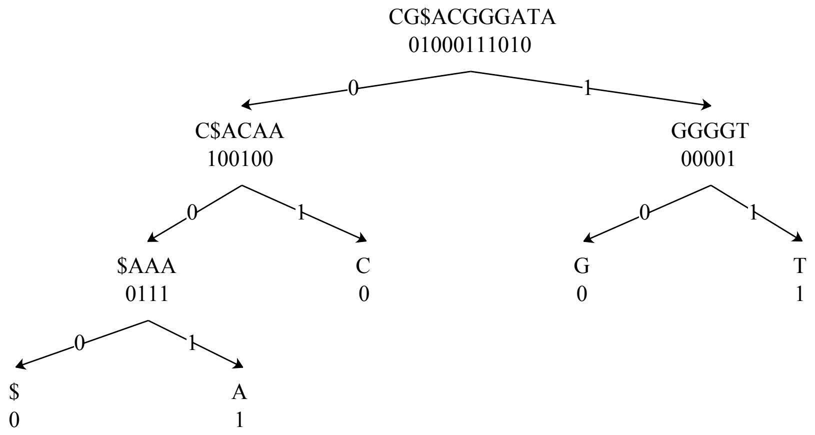


FIGURE C.2: Wavelet Tree for $B_S = [C, G, \$, A, C, G, G, G, A, T, A]$; using $=000, A=100, C=010, G=01, T=11

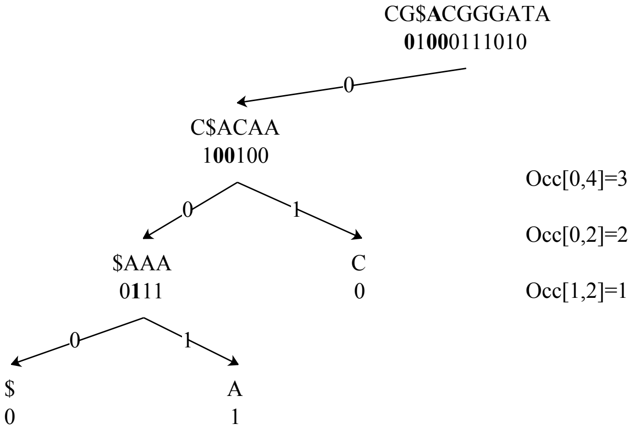


FIGURE C.3: Example of rank query $O_{cc}['A', 4]$ computes using the Wavelet Tree

C.8 RRR representation for a HWT

Generally, Wavelet Tree nodes are stored as RRR sequences [18] for fast binary rank queries and compression. It uses a global table of pre-calculated ranks, which offers $O(1)$ rank queries and zeroth-order entropy compression for binary strings. Depth analysis of this structure and function are distant of our objectives. We invited the reader to consult the references [42], [8], and [91] to study a detail description about this process.

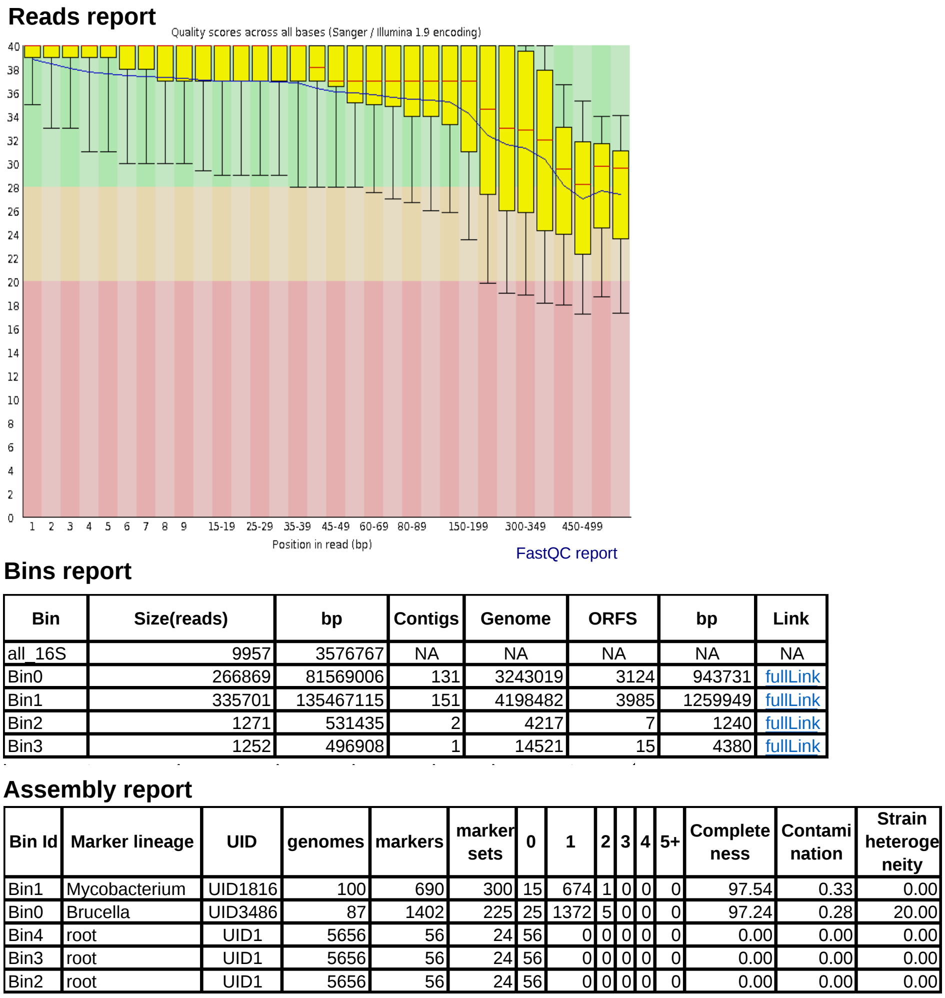
Since BWT is more accessible to compress than the original text by applying Wavelet Tree process and RRR representation, the final structure results in a compact form of the document. Some authors (i.e., [96] and [88]) refer the resulting structure as compress suffix array CSA-WT instead of FM-index.

Appendix D

Auxiliar Figures

It shows an example of the output file generated by DATMA for the Simulated simple metagenome (see Chapter 3 and Chapter 4)

D.1 DATMA Output



Reads report

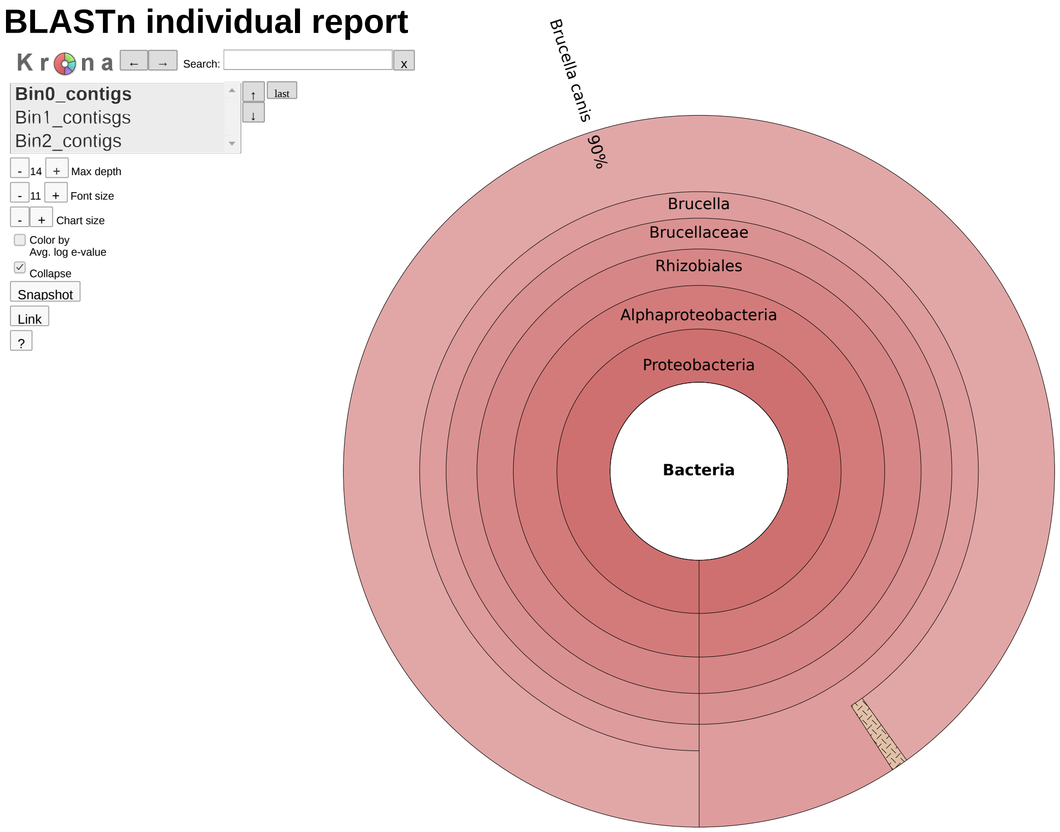
Bins report

Bin	Size(reads)	bp	Contigs	Genome	ORFS	bp	Link
all_16S	9957	3576767	NA	NA	NA	NA	NA
Bin0	266869	81569006	131	3243019	3124	943731	fullLink
Bin1	335701	135467115	151	4198482	3985	1259949	fullLink
Bin2	1271	531435	2	4217	7	1240	fullLink
Bin3	1252	496908	1	14521	15	4380	fullLink

Assembly report

Bin Id	Marker lineage	UID	genomes	markers	marker sets	0	1	2	3	4	5+	Completeness	Contamination	Strain heterogeneity
Bin1	Mycobacterium	UID1816	100	690	300	15	674	1	0	0	0	97.54	0.33	0.00
Bin0	Brucella	UID3486	87	1402	225	25	1372	5	0	0	0	97.24	0.28	20.00
Bin4	root	UID1	5656	56	24	56	0	0	0	0	0	0.00	0.00	0.00
Bin3	root	UID1	5656	56	24	56	0	0	0	0	0	0.00	0.00	0.00
Bin2	root	UID1	5656	56	24	56	0	0	0	0	0	0.00	0.00	0.00

FIGURE D.1: Reads Quality, CLAME report and Assembly metrics for every bin



BLASTn individual report

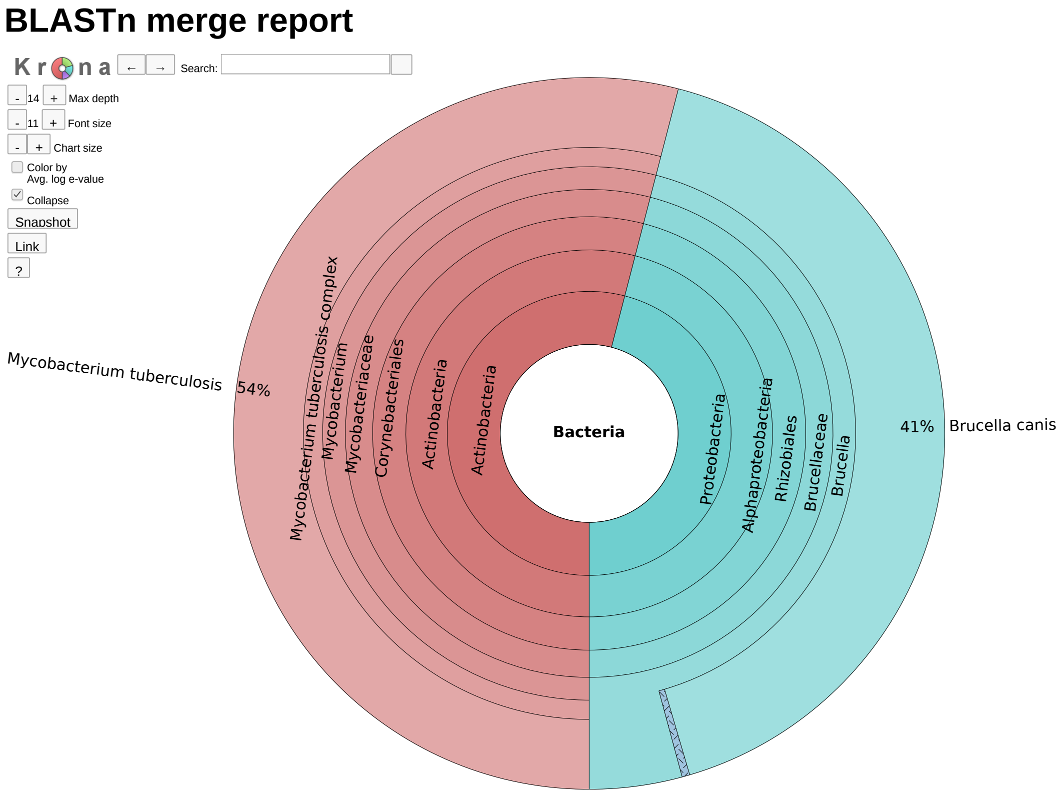
Krona ← → Search: x

Bin0_contigs
Bin1_contisgs
Bin2_contigs

↑ ↓ last

- 14 + Max depth
- 11 + Font size
- + Chart size
□ Color by Avg. log e-value
☑ Collapse
Snapshot
Link
?

FIGURE D.2: Taxonomic annotation for each bin



BLASTn merge report

Krona ← → Search:

- 14 + Max depth
- 11 + Font size
- + Chart size
□ Color by Avg. log e-value
☑ Collapse
Snapshot
Link
?

FIGURE D.3: Taxonomic annotation for all bins

Bibliography

[1] J. Alneberg, B. S. Bjarnason, I. de Bruijn, M. Schirmer, J. Quick, U. Z. Ijaz, N. J. Loman, A. F. Andersson, and C. Quince. "CONCOCT: Clustering cONtigs on COverage and ComposiTion". In: *Arxiv preprint arXiv:1312.4038v1* (2013), p. 28. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0005299. arXiv: 1312.4038.

[2] S. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. *Basic Local Aligment Search Tool*. Vol. 215. 1990, pp. 403–410. DOI: 10.1016/S0022-2836(05)80360-2.

[3] S. Andrews. *FastQC: A quality control tool for high throughput sequence data*. Accessed: 2019-05-06. URL: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/.

[4] R. M. Badia, J. Conejero, C. Diaz, J. Ejarque, D. Lezzi, F. Lordan, C. Ramon-Cortes, and R. Sirvent. "COMP Superscalar, an interoperable programming framework". In: *SoftwareX* 3-4 (2015), pp. 32–36. ISSN: 23527110. DOI: 10.1016/j.softx.2015.10.004.

[5] H. E. Bal, J. Maassen, R. V. van Nieuwpoort, N. Drost, R. Kemp, T. van Kessel, N. Palmer, G. Wrzesinska, T. Kielmann, K. van Reeuwijk, F. J. Seinstra, C. J. H. Jacobs, and K. Verstoep. "Real-World Distributed Computer with Ibis". In: *Computer* 43.8 (Aug. 2010), pp. 54–62. DOI: 10.1109/mc.2010.184. URL: https://doi.org/10.1109/mc.2010.184.

[6] I. Bárány and V. Vu. "Central limit theorems for Gaussian polytopes". In: *The Annals of Probability* 35.4 (July 2007), pp. 1593–1621. DOI: 10.1214/009117906000000791.

[7] K. Bedoya, O. Coltell, F. Cabarcas, and J. F. Alzate. "Metagenomic assessment of the microbial community and methanogenic pathways in biosolids from a municipal wastewater treatment plant in Medellín, Colombia". In: *Science of The Total Environment* 648 (Jan. 2019), pp. 572–581. DOI: 10.1016/j.scitotenv.2018.08.119.

[8] D. Belazzougui, F. Cunial, J. Karkkainen, and V. Makinen. "Versatile Succinct Representations of the Bidirectional Burrows-Wheeler Transform". In: *Lecture*

Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 133–144. DOI: 10.1007/978-3-642-40450-4_12.

[9] J. Besemer and M. Borodovsky. "GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses". In: *Nucleic Acids Research* 33.Web Server (July 2005), W451–W454. DOI: 10.1093/nar/gki487.

[10] S. J. Biller, P. M. Berube, K. Dooley, M. Williams, B. M. Satinsky, T. Hackl, S. L. Hogle, A. Coe, K. Bergauer, H. A. Bouman, T. J. Browning, D. De Corte, C. Hassler, D. Hulston, J. E. Jacquot, E. W. Maas, T. Reinthaler, E. Sintes, T. Yokokawa, and S. W. Chisholm. "Marine microbial metagenomes sampled across space and time". In: *Scientific Data* 5 (2018). Data Descriptor, 180176 EP –. DOI: 10.1038/sdata.2018.176.

[11] *Bioinformation and DDBJ Center*. Accessed: 2019-05-06. URL: https://www.ddbj.nig.ac.jp/.

[12] A. M. Bolger, M. Lohse, and B. Usadel. "Trimmomatic: a flexible trimmer for Illumina sequence data". In: *Bioinformatics* 30.15 (Apr. 2014), pp. 2114–2120. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btu170.

[13] R. M. Bowers, N. C. Kyrpides, R. Stepanauskas, M. Harmon-Smith, D. Doud, T. B. K. Reddy, F. Schulz, J. Jarett, A. R. Rivers, E. A. Eloe-Fadrosh, S. G. Tringe, N. N. Ivanova, A. Copeland, A. Clum, E. D. Becraft, R. R. Malmstrom, B. Birren, M. Podar, P. Bork, G. M. Weinstock, G. M. Garrity, J. A. Dodsworth, S. Yooseph, G. Sutton, F. O. Glöckner, J. A. Gilbert, W. C. Nelson, S. J. Hallam, S. P. Jungbluth, T. J. G. Ettema, S. Tighe, K. T. Konstantinidis, W.-T. Liu, B. J. Baker, T. Rattei, J. A. Eisen, B. Hedlund, K. D. McMahon, N. Fierer, R. Knight, R. Finn, G. Cochrane, I. Karsch-Mizrachi, G. W. Tyson, C. Rinke, N. C. Kyrpides, L. Schriml, G. M. Garrity, P. Hugenholtz, G. Sutton, P. Yilmaz, F. Meyer, F. O. Glöckner, J. A. Gilbert, R. Knight, R. Finn, G. Cochrane, I. Karsch-Mizrachi, A. Lapidus, F. Meyer, P. Yilmaz, D. H. Parks, A. M. Eren, L. Schriml, J. F. Banfield, P. Hugenholtz, and T. Woyke. "Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea". In: *Nature Biotechnology* 35.8 (Aug. 2017), pp. 725–731. DOI: 10.1038/nbt.3893.

[14] A. Brady and S. L. Salzberg. "Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models". In: *Nat Methods* 6.9 (2009), pp. 673–676. ISSN: 1548-7105. DOI: 10.1038/nmeth.1358.

[15] P. Brumm, M. L. Land, L. J. Hauser, C. D. Jeffries, Y. J. Chang, and D. A. Mead. "Complete genome sequences of Geobacillus sp. Y412MC52, a xylan-degrading strain isolated from obsidian hot spring in Yellowstone National

Park". In: *Standards in Genomic Sciences* 10.1 (2015), pp. 1–9. ISSN: 19443277. DOI: 10.1186/s40793-015-0075-0.

[16] S. Chatterji, I. Yamazaki, Z. Bai, and J. A. Eisen. "CompostBin: A DNA Composition-Based Algorithm for Binning Environmental Shotgun Reads". In: (2008). Ed. by M. Vingron and L. Wong, pp. 17–28.

[17] I. M. A. Chen, V. M. Markowitz, K. Chu, K. Palaniappan, E. Szeto, M. Pillay, A. Ratner, J. Huang, E. Andersen, M. Huntemann, N. Varghese, M. Hadjithomas, K. Tennessen, T. Nielsen, N. N. Ivanova, and N. C. Kyrpides. "IMG/M: Integrated genome and metagenome comparative data analysis system". In: *Nucleic Acids Research* 45.D1 (2017), pp. D507–D516. ISSN: 13624962. DOI: 10.1093/nar/gkw929.

[18] F. Claude, G. Navarro, and A. Ordnez. "The wavelet matrix: An efficient wavelet tree for large alphabets". In: *Information Systems* 47 (Jan. 2015), pp. 15–32. DOI: 10.1016/j.is.2014.06.002.

[19] P. J. A. Cock, C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice. "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants". In: *Nucleic Acids Research* 38.6 (Dec. 2009), pp. 1767–1771. DOI: 10.1093/nar/gkp1137.

[20] J. R. Cole, Q. Wang, J. A. Fish, B. Chai, D. M. McGarrell, Y. Sun, C. T. Brown, A. Porras-Alfaro, C. R. Kuske, and J. M. Tiedje. "Ribosomal Database Project: Data and tools for high throughput rRNA analysis". In: *Nucleic Acids Research* 42.D1 (2014), pp. 633–642. ISSN: 03051048. DOI: 10.1093/nar/gkt1244.

[21] M. P. Cox, D. A. Peterson, and P. J. Biggs. "SolexaQA: At-a-glance quality assessment of Illumina second-generation sequencing data". In: *BMC Bioinformatics* 11 (2010). ISSN: 14712105. DOI: 10.1186/1471-2105-11-485.

[22] T. Z. DeSantis, P. Hugenholtz, N. Larsen, M. Rojas, E. L. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, and G. L. Andersen. "Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB". In: *Applied and Environmental Microbiology* 72.7 (2006), pp. 5069–5072. ISSN: 00992240. DOI: 10.1128/AEM.03006-05.

[23] *Diagnostics Products - MP Biomedicals*. Accessed: 2019-05-06. URL: https://www.mpbio.com.

[24] N. N. Diaz, L. Krause, A. Goesmann, K. Niehaus, and T. W. Nattkemper. "TACOA – Taxonomic classification of environmental genomic fragments using a kernelized nearest neighbor approach". In: *BMC Bioinformatics* 10.1 (2009), p. 56. DOI: 10.1186/1471-2105-10-56.

[25] S. El-Gebali, J. Mistry, A. Bateman, S. R. Eddy, A. Luciani, S. C. Potter, M. Qureshi, L. J. Richardson, G. A. Salazar, A. Smart, E. L. L. Sonnhammer, L. Hirsh, L. Paladin, D. Piovesan, S. C. E. Tosatto, and R. D. Finn. "The Pfam protein families database in 2019". In: *Nucleic Acids Research* 47.D1 (Oct. 2018), pp. D427–D432. ISSN: 0305-1048. DOI: 10.1093/nar/gky995.

[26] E. Elbre. *Educational implementations of burrows-wheeler tranformation and ferragina-manzini index*. Accessed: 2019-05-06. URL: https://github.com/egonelbre/fm-index.

[27] *EMBL Heidelberg - The European Molecular Biology Laboratory*. Accessed: 2019-05-06. URL: https://www.embl.de/.

[28] *Extrae 3.7.0 - BSC-Tools*. Accessed: 2019-05-06. URL: https://tools.bsc.es/sites/default/files/.../html/extrae/index.html.

[29] P Ferragina and G Manzini. "Opportunistic data structures with applications". In: *Proceedings 41st Annual Symposium on Foundations of Computer Science*. 2000, pp. 390–398. ISBN: 0272-5428 VO -. DOI: 10.1109/SFCS.2000.892127.

[30] I. Ferrera and O. Sánchez. "Insights into microbial diversity in wastewater treatment systems: How far have we come?" In: *Biotechnology Advances* 34.5 (Sept. 2016), pp. 790–802. DOI: 10.1016/j.biotechadv.2016.04.003.

[31] W. Gerlach, S. Jünemann, F. Tille, A. Goesmann, and J. Stoye. "WebCARMA: a web application for the functional and taxonomic classification of unassembled metagenomic reads". In: *BMC Bioinformatics* 10.1 (2009), p. 430. DOI: 10.1186/1471-2105-10-430.

[32] J. S. Ghurye, V. Cepeda-Espinoza, and M. Pop. "Metagenomic Assembly: Overview, Challenges and Applications". In: *Yale J Biol Med* 89.3 (2016), pp. 353–362. ISSN: 1551-4056.

[33] S. Girotto, C. Pizzi, and M. Comin. "MetaProb: accurate metagenomic reads binning based on probabilistic sequence signatures". In: *Bioinformatics* 32.17 (2016), pp. i567–i575. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btw466.

[34] I. Gregor, J. Dröge, M. Schirmer, C. Quince, and A. C. McHardy. "PhyloPythiaS+: a self-training method for the rapid reconstruction of low-ranking taxonomic bins from metagenomes". In: *PeerJ* 4 (2016), e1603–e1603. ISSN: 2167-8359. DOI: 10.7717/peerj.1603.

[35] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S. R. Eddy. "Rfam: An RNA family database". In: *Nucleic Acids Research* 31.1 (2003), pp. 439–441. ISSN: 03051048. DOI: 10.1093/nar/gkg006.

[36] R. Grossi, J. S. Vitter, and B. Xu. "Wavelet Trees: From Theory to Practice". In: *2011 First International Conference on Data Compression, Communications and Processing*. IEEE, June 2011. DOI: 10.1109/ccp.2011.16.

[37] J. Guo, J. Li, H. Chen, P. L. Bond, and Z. Yuan. "Metagenomic analysis reveals wastewater treatment plants as hotspots of antibiotic resistance genes and mobile genetic elements". In: *Water Research* 123 (2017), pp. 468 –478. ISSN: 0043-1354. DOI: https://doi.org/10.1016/j.watres.2017.07.002.

[38] A. Gupta, S. Kumar, V. P. K. Prasoodanan, K. Harish, A. K. Sharma, and V. K. Sharma. "Reconstruction of Bacterial and Viral Genomes from Multiple Metagenomes". In: *Frontiers in Microbiology* 7 (2016), p. 469. ISSN: 1664-302X. DOI: 10.3389/fmicb.2016.00469.

[39] A. Gurevich, V. Saveliev, N. Vyahhi, and G. Tesler. "QUAST: quality assessment tool for genome assemblies". In: *Bioinformatics* 29.8 (Feb. 2013), pp. 1072–1075. DOI: 10.1093/bioinformatics/btt086.

[40] A. Hatem, D. Bozdag, and U. V. Catalyurek. "Benchmarking Short Sequence Mapping Tools". In: *2011 IEEE International Conference on Bioinformatics and Biomedicine*. IEEE, Nov. 2011. DOI: 10.1109/bibm.2011.83.

[41] M. Horton, N. Bodenhausen, and J. Bergelson. "MARTA: a suite of Java-based tools for assigning taxonomic status to DNA sequences". In: *Bioinformatics* 26.4 (Dec. 2009), pp. 568–569. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btp682.

[42] H. Huo, L. Chen, H. Zhao, J. S. Vitter, Y. Nekrich, and Q. Yu. "A Data-Aware FM-index". In: *2015 Proceedings of the Seventeenth Workshop on Algorithm Engineering and Experiments*. Society for Industrial and Applied Mathematics, Dec. 2014, pp. 10–23. DOI: 10.1137/1.9781611973754.2.

[43] D Huson, A Auch, J Qi, and S Schuster. "MEGAN analysis of metagenome data". In: *Gennome Res.* 17 (2007), pp. 377–386. ISSN: 10889051. DOI: 10.1101/gr.5969107..

[44] D. Hyatt, G.-L. Chen, P. F. LoCascio, M. L. Land, F. W. Larimer, and L. J. Hauser. "Prodigal: prokaryotic gene recognition and translation initiation site identification". In: *BMC Bioinformatics* 11.1 (Mar. 2010). DOI: 10.1186/1471-2105-11-119.

[45] *Illumina*. Accessed: 2019-05-06. URL: http://www.illumina.com.

[46] P. Jia, L. Xuan, L. Liu, and C. Wei. "MetaBinG: using GPUs to accelerate metagenomic sequence classification". In: *PLoS One* 6.11 (2011), e25353–e25353. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0025353.

[47] F. Ju, F. Guo, L. Ye, Y. Xia, and T. Zhang. "Metagenomic analysis on seasonal microbial variations of activated sludge from a full-scale wastewater treatment plant over 4 years". In: *Environmental Microbiology Reports* 6.1 (Oct. 2013), pp. 80–89. DOI: 10.1111/1758-2229.12110.

[48] T. H. Jukes and C. R. Cantor. *Evolution of protein molecules.* ACADEMIC PRESS, INC., 1969, pp. 21–132. ISBN: 978-1-4832-3211-9. DOI: citeulike-article-id:768582.

[49] D. Kang, F. Li, E. S. Kirton, A. Thomas, R. S. Egan, H. An, and Z. Wang. "MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies". In: (Feb. 2019). DOI: 10.7287/peerj.preprints.27522v1.

[50] J. Karkkainen. "Suffix Array Construction". In: *Encyclopedia of Algorithms*. Springer US, 2015, pp. 1–6. DOI: 10.1007/978-3-642-27848-8_412-2.

[51] J. Koster and S. Rahmann. "Snakemake–a scalable bioinformatics workflow engine". In: *Bioinformatics* 28.19 (Aug. 2012), pp. 2520–2522. DOI: 10.1093/bioinformatics/bts480. URL: https://doi.org/10.1093/bioinformatics/bts480.

[52] L. Kriaa, A. Bouchhima, M. Gligor, A.-M. Fouillart, F. Pétrot, and A.-A. Jerraya. "Parallel Programming of Multi-processor SoC: A HW–SW Interface Perspective". In: *International Journal of Parallel Programming* 36.1 (Apr. 2007), pp. 68–92. DOI: 10.1007/s10766-007-0042-5.

[53] M. Kröber, T. Bekel, N. N. Diaz, A. Goesmann, S. Jaenicke, L. Krause, D. Miller, K. J. Runte, P. Viehöver, A. Pühler, and A. Schlüter. "Phylogenetic characterization of a biogas plant microbial community integrating clone library 16S-rDNA sequences and metagenome sequence data obtained by 454-pyrosequencing". In: *Journal of Biotechnology* 142.1 (June 2009), pp. 38–49. DOI: 10.1016/j.jbiotec.2009.02.010.

[54] J. R. Kultima, S. Sunagawa, J. Li, W. Chen, H. Chen, D. R. Mende, M. Arumugam, Q. Pan, B. Liu, J. Qin, J. Wang, and P. Bork. "MOCAT: A Metagenomics Assembly and Gene Prediction Toolkit". In: *PLoS ONE* 7.10 (Oct. 2012). Ed. by J. A. Gilbert, e47656. DOI: 10.1371/journal.pone.0047656. URL: https://doi.org/10.1371/journal.pone.0047656.

[55] S. Kumar, G. Stecher, and K. Tamura. "MEGA7: Molecular Evolutionary Genetics Analysis Version 7.0 for Bigger Datasets". In: *Molecular biology and evolution* 33.7 (2016), pp. 1870–1874. ISSN: 15371719. DOI: 10.1093/molbev/msw054.

[56] K. Lagesen, P. Hallin, E. A. Rødland, H. H. Stærfeldt, T. Rognes, and D. W. Ussery. "RNAmmer: Consistent and rapid annotation of ribosomal RNA genes". In: *Nucleic Acids Research* 35.9 (2007), pp. 3100–3108. ISSN: 03051048. DOI: 10.1093/nar/gkm160.

[57] B. Langmead and S. L. Salzberg. "Fast gapped-read alignment with Bowtie 2". In: *Nature Methods* 9.4 (Mar. 2012), pp. 357–359. DOI: 10.1038/nmeth.1923.

[58] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata. "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median". In: *Journal of Experimental Social Psychology* 49.4 (July 2013), pp. 764–766. DOI: 10.1016/j.jesp.2013.03.013.

[59] D. Li, C. M. Liu, R. Luo, K. Sadakane, and T. W. Lam. "MEGAHIT: An ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph". In: *Bioinformatics* 31.10 (2015), pp. 1674–1676. ISSN: 14602059. DOI: 10.1093/bioinformatics/btv033.

[60] H. Li and R. Durbin. "Fast and accurate short read alignment with Burrows-Wheeler transform". In: *Bioinformatics* 25.14 (May 2009), pp. 1754–1760. DOI: 10.1093/bioinformatics/btp324.

[61] B. Liu, T. Gibbons, M. Ghodsi, and M. Pop. "MetaPhyler: Taxonomic profiling for metagenomic sequences". In: (2010), pp. 95–100. DOI: 10.1109/BIBM.2010.5706544.

[62] K. London, S. Moore, P. Mucci, K. Seymour, and R. Luczak. "The PAPI Cross-Platform Interface to Hardware Performance Counters". In: Biloxi, Mississippi, 2001.

[63] J. Lotero, A. Benavides, A. Guerra, and S. Isaza. "UdeAlignC: Fast Alignment for the Compression of DNA Reads". In: *2018 IEEE Colombian Conference on Communications and Computing (COLCOM)*. IEEE, May 2018. DOI: 10.1109/colcomcon.2018.8466336.

[64] T. Magoč and S. L. Salzberg. "FLASH: Fast length adjustment of short reads to improve genome assemblies". In: *Bioinformatics* 27.21 (2011), pp. 2957–2963. ISSN: 13674803. DOI: 10.1093/bioinformatics/btr507.

[65] P. Menzel, K. L. Ng, and A. Krogh. "Fast and sensitive taxonomic classification for metagenomics with Kaiju". In: *Nature Communications* 7.1 (Apr. 2016). DOI: 10.1038/ncomms11257.

[66] K. M. Meyer, A. M. Klein, J. L. M. Rodrigues, K. Nüsslein, S. G. Tringe, B. S. Mirza, J. M. Tiedje, and B. J. M. Bohannan. “Conversion of Amazon rainforest to agriculture alters community traits of methane-cycling organisms”. In: *Molecular Ecology* 26.6 (2017), pp. 1547–1556. DOI: 10.1111/mec.14011.

[67] J. Miller. “Short Report: Reaction Time Analysis with Outlier Exclusion: Bias Varies with Sample Size”. In: *The Quarterly Journal of Experimental Psychology Section A* 43.4 (Nov. 1991), pp. 907–912. DOI: 10.1080/14640749108400962.

[68] M. H. Mohammed, T. S. Ghosh, N. K. Singh, and S. S. Mande. “SPHINXan algorithm for taxonomic binning of metagenomic sequences”. In: *Bioinformatics* 27.1 (Oct. 2010), pp. 22–30. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btq608.

[69] M. Monzoorul Haque, T. S. Ghosh, D. Komanduri, and S. S. Mande. “SOrt-ITEMS: Sequence orthology based approach for improved taxonomic estimation of metagenomic sequences”. In: *Bioinformatics* 25.14 (May 2009), pp. 1722–1730. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btp317.

[70] Y. Mori. *libdivsufsort*. Accessed: 2019-05-06. URL: https://github.com/y-256/libdivsufsort.

[71] F. Muhamad, R. Ahmad, S. Asi, and M. Murad. “Performance Analysis Of Needleman-Wunsch Algorithm (Global) And Smith-Waterman Algorithm (Local) In Reducing Search Space And Time For Dna Sequence Alignment”. In: *Journal of Physics: Conference Series* 1019 (June 2018), p. 012085. DOI: 10.1088/1742-6596/1019/1/012085.

[72] O. U. Nalbantoglu, S. F. Way, S. H. Hinrichs, and K. Sayood. “RAIphy: Phylogenetic classification of metagenomics samples using iterative refinement of relative abundance index profiles”. In: *BMC Bioinformatics* 12.1 (Jan. 2011). DOI: 10.1186/1471-2105-12-41.

[73] T. Namiki, T. Hachiya, H. Tanaka, and Y. Sakakibara. “MetaVelvet: an extension of Velvet assembler to de novo metagenome assembly from short sequence reads”. In: *Nucleic Acids Research* 40.20 (July 2012), e155–e155. DOI: 10.1093/nar/gks678.

[74] *National Center for Biotechnology Information (NCBI)*. Accessed: 2019-05-06. URL: https://www.ncbi.nlm.nih.gov.

[75] S. B. Needleman and C. D. Wunsch. “A general method applicable to the search for similarities in the amino acid sequence of two proteins”. In: *Journal of Molecular Biology* 48.3 (Mar. 1970), pp. 443–453. DOI: 10.1016/0022-2836(70)90057-4.

[76] N. Nethercote and J. Seward. “Valgrind”. In: *Proceedings of the 2007 ACM SIGPLAN conference on Programming language design and implementation*. ACM Press, 2007. DOI: 10.1145/1250734.1250746.

[77] S. Nurk, A. Bankevich, D. Antipov, A. A. Gurevich, A. Korobeynikov, A. Lapidus, A. D. Prjibelski, A. Pyshkin, A. Sirotkin, Y. Sirotkin, R. Stepanauskas, S. R. Clingenpeel, T. Woyke, J. S. Mclean, R. Lasken, G. Tesler, M. A. Alekseyev, and P. A. Pevzner. “Assembling Single-Cell Genomes and Mini-Metagenomes From Chimeric MDA Products”. In: *Journal of Computational Biology* 20.10 (2013), pp. 714–737. ISSN: 1066-5277. DOI: 10.1089/cmb.2013.0084.

[78] P. P. Nurk S, Meleshko D, Korobeynikov A. “metaSPAdes: A New Versatile Metagenomic Assembler”. In: *Genome Res.* 1.27 (2017), pp. 30–47. DOI: 10.1101/gr.213959.116.4.

[79] B. D. Ondov, N. H. Bergman, and A. M. Phillippy. “Interactive metagenomic visualization in a Web browser”. In: *BMC Bioinformatics* 12.1 (Sept. 2011). DOI: 10.1186/1471-2105-12-385.

[80] R. Ounit, S. Wanamaker, T. J. Close, and S. Lonardi. “CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers”. In: *BMC Genomics* 16.1 (2015), p. 236. ISSN: 1471-2164. DOI: 10.1186/s12864-015-1419-2.

[81] *Oxford Nanopore Technologies*. Accessed: 2019-05-06. URL: https://www.nanoporetech.com.

[82] *Pacific Biosciences*. Accessed: 2019-05-06. URL: http://www.pacb.com.

[83] *Paraver: a flexible performance analysis tool - BSC-Tools*. Accessed: 2019-05-06. URL: https://tools.bsc.es/paraver.

[84] H. Park, A. C. Brotto, M. C. M. van Loosdrecht, and K. Chandran. “Discovery and metagenomic analysis of an anammox bacterial enrichment related to Candidatus Brocadia caroliniensis in a full-scale glycerol-fed nitritation-denitritation separate centrate treatment process”. In: *Water Research* 111 (2017), pp. 265–273. ISSN: 0043-1354. DOI: https://doi.org/10.1016/j.watres.2017.01.011.

[85] D. H. Parks, M. Imelfort, C. T. Skennerton, P. Hugenholtz, and G. W. Tyson. “CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes.” In: *Genome research* 25.7 (2015), pp. 1043–55. ISSN: 1549-5469. DOI: 10.1101/gr.186072.114.

[86] D. H. Parks, C. Rinke, M. Chuvochina, P.-A. Chaumeil, B. J. Woodcroft, P. N. Evans, P. Hugenholtz, and G. W. Tyson. "Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life". In: *Nature Microbiology* 2.11 (2017), pp. 1533–1542. ISSN: 2058-5276. DOI: 10.1038/s41564-017-0012-7.

[87] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1984. ISBN: 0-201-05594-5.

[88] M. Petri. *Fm-index - compressed full-text index*. Accessed: 2019-05-06. URL: https://github.com/mpetri/FM-Index.

[89] V. H. T. Pham and J. Kim. "Cultivation of unculturable soil bacteria". In: *Trends in Biotechnology* 30.9 (2012), pp. 475–484. ISSN: 0167-7799. DOI: 10.1016/j.tibtech.2012.05.007.

[90] V. C. Piro, M. Matschkowski, and B. Y. Renard. "MetaMeta: integrating metagenome analysis tools to improve taxonomic profiling". In: *Microbiome* 5.1 (2017), p. 101. ISSN: 20492618. DOI: 10.1186/s40168-017-0318-y.

[91] P. Prochazka and J. Holub. "Compressing Similar Biological Sequences Using FM-index". In: *2014 Data Compression Conference*. IEEE, Mar. 2014. DOI: 10.1109/dcc.2014.47.

[92] C. Quast, E. Pruesse, P. Yilmaz, J. Gerken, T. Schweer, P. Yarza, J. Peplies, and F. O. Glöckner. "The SILVA ribosomal RNA gene database project: improved data processing and web-based tools". In: *Nucleic acids research* 41.Database issue (2013), pp. D590–D596. ISSN: 1362-4962. DOI: 10.1093/nar/gks1219.

[93] D. Rivière, V. Desvignes, E. Pelletier, S. Chaussonnerie, S. Guermazi, J. Weissenbach, T. Li, P. Camacho, and A. Sghir. "Towards the definition of a core of microorganisms involved in anaerobic digestion of sludge". In: *The Isme Journal* 3 (2009). Original Article, 700 EP –.

[94] *Roche Sequencing*. Accessed: 2019-05-06. URL: https://sequencing.roche.com/en.html.

[95] G. L. Rosen, E. R. Reichenberger, and A. M. Rosenfeld. "NBC: the Naïve Bayes Classification tool webserver for taxonomic classification of metagenomic reads". In: *Bioinformatics* 27.1 (Nov. 2010), pp. 127–129. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btq619.

[96] S. Gog, J. Bader, T. Beller, and M. Petri. *Sdsl - succinct data structure library*. Accessed: 2019-05-06. URL: https://github.com/simongog/sdsl-lite.

[97] F. Sanger and A. Coulson. "A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase". In: *Journal of Molecular Biology* 94.3 (1975), pp. 441 –448. ISSN: 0022-2836. DOI: https://doi.org/10.1016/0022-2836(75)90213-2.

[98] A. Sczyrba, P. Hofmann, P. Belmann, D. Koslicki, S. Janssen, J. Dröge, I. Gregor, S. Majda, J. Fiedler, E. Dahms, A. Bremges, A. Fritz, R. Garrido-Oter, T. S. Jørgensen, N. Shapiro, P. D. Blood, A. Gurevich, Y. Bai, D. Turaev, M. Z. DeMaere, R. Chikhi, N. Nagarajan, C. Quince, F. Meyer, M. Balvočiūtė, L. H. Hansen, S. J. Sørensen, B. K. H. Chia, B. Denis, J. L. Froula, Z. Wang, R. Egan, D. D. Kang, J. J. Cook, C. Deltel, M. Beckstette, C. Lemaitre, P. Peterlongo, G. Rizk, D. Lavenier, Y.-W. Wu, S. W. Singer, C. Jain, M. Strous, H. Klingenberg, P. Meinicke, M. D. Barton, T. Lingner, H.-H. Lin, Y.-C. Liao, G. G. Z. Silva, D. A. Cuevas, R. A. Edwards, S. Saha, V. C. Piro, B. Y. Renard, M. Pop, H.-P. Klenk, M. Göker, N. C. Kyrpides, T. Woyke, J. A. Vorholt, P. Schulze-Lefert, E. M. Rubin, A. E. Darling, T. Rattei, and A. C. McHardy. "Critical Assessment of Metagenome Interpretation—a benchmark of metagenomics software". In: *Nature Methods* 14.11 (Oct. 2017), pp. 1063–1071. DOI: 10.1038/nmeth.4458.

[99] R. Seshadri, S. A. Kravitz, L. Smarr, P. Gilna, and M. Frazier. "CAMERA: A community resource for metagenomics". In: *PLoS Biology* 5.3 (2007), pp. 0394–0397. ISSN: 15449173. DOI: 10.1371/journal.pbio.0050075. eprint: NIHMS150003.

[100] L. Siegwald, C. Audebert, G. Even, E. Viscogliosi, S. Caboche, and M. Chabé. "Targeted metagenomic sequencing data of human gut microbiota associated with Blastocystis colonization". In: *Scientific Data* 4 (2017), 170081 EP –.

[101] T. F. Smith and M. S. Waterman. "Identification of common molecular subsequences". In: *Journal of Molecular Biology* 147.1 (1981), pp. 195–197. ISSN: 00222836. DOI: 10.1016/0022-2836(81)90087-5.

[102] J. T. Staley and A. Konopka. "MEASUREMENT OF IN SITU ACTIVITIES OF NONPHOTOSYNTHETIC MICROORGANISMS IN AQUATIC AND TERRESTRIAL HABITATS". In: *Annual Review of Microbiology* 39.1 (1985). PMID: 3904603, pp. 321–346. DOI: 10.1146/annurev.mi.39.100185.001541.

[103] H. Suenaga. "Targeted metagenomics: a high-resolution metagenomics approach for specific gene clusters in complex microbial communities". In: *Environmental Microbiology* 14.1 (2012), pp. 13–22. DOI: 10.1111/j.1462-2920.2011.02438.x.

[104] J. Sun, K. Chen, and Z. Hao. "Pairwise alignment for very long nucleic acid sequences". In: *Biochemical and Biophysical Research Communications* 502.3 (July 2018), pp. 313–317. DOI: 10.1016/j.bbrc.2018.05.134.

[105] J. Tamames and F. Puente-Sánchez. "SqueezeMeta, A Highly Portable, Fully Automatic Metagenomic Analysis Pipeline". In: *Frontiers in Microbiology* 9 (2019), p. 3349. ISSN: 1664-302X. DOI: 10.3389/fmicb.2018.03349.

[106] H. Teeling, J. Waldmann, T. Lombardot, M. Bauer, and F. O. Glöckner. "TETRA: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in DNA sequences". In: *BMC Bioinformatics* 5 (2004), pp. 163–163. ISSN: 1471-2105. DOI: 10.1186/1471-2105-5-163.

[107] *Thermo Fisher Scientific*. Accessed: 2019-05-06. URL: https://www.thermofisher.com.

[108] P. D. Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame. "Nextflow enables reproducible computational workflows". In: *Nature Biotechnology* 35.4 (Apr. 2017), pp. 316–319. DOI: 10.1038/nbt.3820. URL: https://doi.org/10.1038/nbt.3820.

[109] T. J. Treangen, S. Koren, D. D. Sommer, B. Liu, I. Astrovskaya, B. Ondov, A. E. Darling, A. M. Phillippy, and M. Pop. "MetAMOS: a modular and open source metagenomic assembly and analysis pipeline." In: *Genome biology* 14.1 (2013), R2. ISSN: 1474-760X. DOI: 10.1186/gb-2013-14-1-r2.

[110] P. J. Turnbaugh, R. E. Ley, M. Hamady, C. Fraser-Liggett, R. Knight, and J. I. Gordon. "The human microbiome project: exploring the microbial part of ourselves in a changing world". In: *Nature* 449.7164 (2007), pp. 804–810. ISSN: 1476-4687. DOI: 10.1038/nature06244.The.

[111] A. D. Tyler, L. Mataseje, C. J. Urfano, L. Schmidt, K. S. Antonation, M. R. Mulvey, and C. R. Corbett. "Evaluation of Oxford Nanopore's MinION Sequencing Device for Microbial Whole Genome Sequencing Applications". In: *Scientific Reports* 8.1 (2018), p. 10931. ISSN: 2045-2322. DOI: 10.1038/s41598-018-29334-5.

[112] *UniProt Consortium*. Accessed: 2019-05-06. URL: https://www.uniprot.org/.

[113] G. V. Uritskiy, J. DiRuggiero, and J. Taylor. "MetaWRAP—a flexible pipeline for genome-resolved metagenomic data analysis". In: *Microbiome* 6.1 (2018), p. 158. ISSN: 2049-2618. DOI: 10.1186/s40168-018-0541-1.

[114] S. R. Vartoukian. "Cultivation strategies for growth of uncultivated bacteria". In: *Journal of Oral Biosciences* 58.4 (2016), pp. 143 –149. ISSN: 1349-0079. DOI: https://doi.org/10.1016/j.job.2016.08.001.

[115] N. Vijay, C. M. Bossu, J. W. Poelstra, M. H. Weissensteiner, A. Suh, A. P. Kryukov, and J. B. W. Wolf. "Evolution of heterogeneous genome differentiation across multiple contact zones in a crow species complex". In: *Nature Communications* 7 (2016), 13195 EP –.

[116] L. Vinh, T. Lang, L. Binh, and T. Hoai. "A two-phase binning algorithm using l-mer frequency on groups of non-overlapping reads". In: *Algorithms for Molecular Biology* 10.1 (2015), p. 2. DOI: 10.1186/s13015-014-0030-4.

[117] Q. Wang, G. M. Garrity, J. M. Tiedje, and J. R. Cole. "Naive Bayesian Classifier for Rapid Assignment of rRNA Sequences into the New Bacterial Taxonomy". In: *Applied and Environmental Microbiology* 73.16 (June 2007), pp. 5261–5267. DOI: 10.1128/aem.00062-07. URL: https://doi.org/10.1128/aem.00062-07.

[118] Y. Wang, H. C. M. Leung, S. M. Yiu, and F. Y. L. Chin. "Metacluster 5.0: A two-round binning approach for metagenomic data for low-abundance species in a noisy sample". In: *Bioinformatics* 28.18 (2012), pp. 356–362. ISSN: 13674803. DOI: 10.1093/bioinformatics/bts397.

[119] M. Wilde, M. Hategan, J. M. Wozniak, B. Clifford, D. S. Katz, and I. Foster. "Swift: A language for distributed parallel scripting". In: *Parallel Computing* 37.9 (Sept. 2011), pp. 633–652. DOI: 10.1016/j.parco.2011.05.005. URL: https://doi.org/10.1016/j.parco.2011.05.005.

[120] A. Wilke, J. Bischof, W. Gerlach, E. Glass, T. Harrison, K. P. Keegan, T. Paczian, W. L. Trimble, S. Bagchi, A. Grama, S. Chaterji, and F. Meyer. "The MG-RAST metagenomics database and portal in 2015". In: *Nucleic Acids Research* 44.D1 (2016), pp. D590–D594. ISSN: 13624962. DOI: 10.1093/nar/gkv1322.

[121] J. C. Wooley and Y. Ye. "Metagenomics: Facts and Artifacts, and Computational Challenges". In: *Journal of Computer Science and Technology* 25 (2009), pp. 71–81.

[122] M. Wu and A. J. Scott. "Phylogenomic analysis of bacterial and archaeal sequences with AMPHORA2". In: *Bioinformatics* 28.7 (2012), pp. 1033–1034. ISSN: 13674803. DOI: 10.1093/bioinformatics/bts079.

[123] Y.-W. Wu, B. A. Simmons, and S. W. Singer. "MaxBin2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets". In: *Bioinformatics* 32.4 (Oct. 2015), pp. 605–607. DOI: 10.1093/bioinformatics/btv638.

[124] Y.-W. Wu and Y. Ye. "A novel abundance-based algorithm for binning metagenomic sequences using l-tuples". In: *J Comput Biol* 18.3 (2011), pp. 523–534. ISSN: 1557-8666. DOI: 10.1089/cmb.2010.0245.

[125] D. R. Zerbino and E. Birney. "Velvet: Algorithms for de novo short read assembly using de Bruijn graphs". In: *Genome Research* 18.5 (2008), pp. 821–829. ISSN: 10889051. DOI: 10.1101/gr.074492.107. arXiv: 0209100 [arXiv:quant-ph].

[126] T. Zhang, M.-F. Shao, and L. Ye. "454 Pyrosequencing reveals bacterial diversity of activated sludge from 14 sewage treatment plants". In: *The Isme Journal* 6 (2011). Original Article, 1137 EP –.