UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Agronòmica i del Medi Natural

BACHELOR THESIS

LONG-READ, ERROR PRONE METAGENOMICS: SYSTEMATIC EVALUATION OF ASSEMBLY TOOLS FOR NANOPORE SEQUENCING.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA AGRONÒMICA I DEL MEDI NATURAL (ETSEAMN)

València, July 2020

Biotechnology Bachelor's Degree

Academic year 2019/2020

AUTHOR: Morgane Blanot

ACADEMIC SUPERVISOR: Prof. José Gadea Vacas

EXTERNAL SUPERVISOR: PhD. Cristina Vilanova Serrador

EXTERNAL COLLABORATING SUPERVISOR: MSc. Adriel Latorre Pérez

LONG-READ, ERROR PRONE METAGENOMICS:
SYSTEMATIC EVALUATION OF ASSEMBLY TOOLS FOR NANOPORE SEQUENCING.

València, July 2020

ABSTRACT:

Microorganisms can produce a wide variety of compounds and are key factors for understanding the behavior of ecological systems. Metagenomics is the tool for achieving this knowledge, studying microorganisms directly from their source through different approaches: focusing on sequencing marker genes (metataxonomics) or puzzling up the whole genetic material into separate genomes (metagenomics). In the recent past, Illumina has been the most sequencing technology used. However, short reads generated by Illumina are hard to assemble, and they produce very fragmented metagenomes. Despite their high intrinsic error, third generation sequencing platforms harbor the potential to overcome this issue thanks to their ability to generate longer reads. In this regard, MinION (Oxford Nanopore Technologies) is very advantageous for metagenomics applications, since it is cheap, portable and provides information in real-time.

Many assemblers have been designed for dealing with error-prone data, but there is not a clear consensus about the tools to use to achieve the best results when handling MinION data. It is thus necessary to benchmark these tools and state what, why and when to use them. For that, using the best performing assemblers we know in the present, the aim of the present work is to analyze sequencing data from microbial communities of different complexities and systematically compare the metagenomes retrieved. The final goal of the study is to provide guidance for other scientists to choose the proper software, and to stimulate the rational development of tools and methodologies for this field.

KEY WORDS:

MinIOn; Nanopore Sequencing; Metagenomics; Assembly; Bioinformatics; Benchmark

Author: Morgane Blanot

Academic Tutor: Prof. José Gadea Vacas

External Cotutor: PhD. Cristina Vilanova Serrador

External Collaborating Cotutor: MSc. Adriel Latorre Pérez

METAGENÓMICA DE LECTURAS LARGAS: EVALUACIÓN SISTEMÁTICA DE HERRAMIENTAS DE ENSAMBLAJE PARA SECUENCIACIÓN NANOPORE.

València, Juliol 2020

ABSTRACT:

Los microorganismos producen compuestos muy variados y son esenciales para entender el comportamiento de sistemas ambientales. La metagenómica es la vía para obtener este conocimiento, estudiando los microorganismos directamente desde su hábitat usando diferentes enfoques: centrarse en secuenciar genes marcadores (metataxonómica), o montar todo el material genético en genomas separados (metagenómica). En los últimos años, Illumina ha sido la tecnología más usada para secuenciar; sin embargo, las lecturas cortas que genera Illumina son difíciles de ensamblar, y producen metagenomas muy fragmentados. A pesar de su alto error intrínseco, las plataformas de tercera generación tienen potencial para superar este inconveniente gracias a que pueden generar lecturas más largas. En este contexto, MinION (Oxford Nanopore Technologies) es muy conveniente para aplicaciones metagenómicas; ya que es barato, portable y proporciona información en tiempo real.

Se han diseñado muchos ensambladores para tratar con datos de mayor error intrínseco, pero no hay un consenso en cuanto a las herramientas que se deben usar para obtener los mejores resultados cuando se usa MinION. Es por tanto necesario evaluar estas herramientas y definir qué, por qué y cuándo usarlas. Para este propósito, y usando los mejores ensambladores que se conocen actualmente, la finalidad de este proyecto es analizar datos secuenciados de comunidades microbianas de varias complejidades y comparar sistemáticamente los metagenomas obtenidos, con el objetivo de orientar a otros científicos para elegir el software adecuado y estimular el desarrollo racional de nuevas herramientas y estrategias para este campo.

PALABRAS CLAVE:

MinION; Secuenciación Nanopore; Metagenómica; Ensamblaje; Bioinformática; Evaluación sistemática

Autora: Morgane Blanot

Tutor académico: Prof. José Gadea Vacas

Cotutora externa: Dr. Cristina Vilanova Serrador

Cotutor colaborador externo: Adriel Latorre Pérez

AKNOWLEDGEMENTS / AGRAÏMENTS

En el transcurs de l'orientació professional, i en la consecució de projectes, n'hi ha moments decissius que definixen la direcció d'una carrera. Moments decissius que requerixen molta gent decissiva, que t'espenten cap a una o altra direcció.

Voldria donar les gràcies a Pepe per animar-me, recolzar-me en les meues caramboles amb empreses, creure en el meu potencial i recomanar-me. A Manel, per donar-me varios tours pels laboratoris en què treballa, aceptar-me i obrir-me la porta al seu món. Al personal dels laboratoris de Darwin i del I2SysBio per acollir-me i tractar-me tan bé el poc temps que els vaig acompañar, i especialment a Marta i Javi, per instruïr-me. A Cristina, per confiar en mi i no donar-me'n uno, sinó dos TFGs quan el confinament va cancel·lar el primer projecte, i per avaluar-me, aconsellar-me i ajudar-me a millorar. A Adriel, per la seua infinita paciència, ànims, coaching, consells, temps, coneixement, guia i amabilitat. A Pascual, per obrir-me el camí amb el seu treball.

Y de quienes influyen de casa, quería agradecer a José Antonio, Paco e Inma, por introducirme en el mundo de las ciencias biológicas; y a mi hermana, por haber sido mi modelo, inspirarme y hacerme ayudarla a estudiar Ciencias Naturales cuando aún no sabía multiplicar. A mi padre, por darme todo lo que sabe y puede, haberme hecho trabajadora y desearme lo mejor. A Aitor, por mudarse tres veces en un año, aguantarme todos los días con mis inestables estados anímicos y todos mis desastres, hacerme café y seguir ahí. A la gente que me acompaña, por darme paz y alegría, juegos, memes y cerveza.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SUPPLEMENTARY FIGURES

1. Glossary

Metagenomics: The study of the composition and functionalities of the microbial communities, from a shotgun sequencing-based approach.

Metataxonomics: The study of the composition of the microbial communities through marker genes sequencing.

Shotgun sequencing: Sequencing approach that focuses on targeting the whole (meta)genome instead of discrete regions, to obtain the whole sequence.

NGS: Next Generation Sequencing, a group of sequencing technologies that sequence in a much faster and cheaper way than the first sequencing methods, namely Sanger, producing high throughput data.

Read: Piece of information obtained through sequencing that corresponds to a fragment of the DNA of interest in an assembly experiment. This information is integrated to conform the genome of interest.

Contig: Fragment of DNA sequence originated as a result of an assembly of sequenced reads.

Assembly: Integration of sequencing data to form the genome of interest.

Coverage: Parameter for describing the sequencing depth, it is the amount of times a genome has been sequenced and therefore the number of fragments each spot in the genome has in the reads pool. The higher, the better the final genome accuracy.

Draft sequence: Resulting preliminar sequence of the assembly and each round of a polisher.

Base pairs (bp): Length unity used in genetics and genomics, that includes one pair of complementary bases of the DNA.

Kilobase (kpb): A thousand base pairs.

Megabase (Mb): A million base pairs.

N50: Minimum contig length needed to cover 50% of the genome (or metagenome). If contigs sizes of size N50 or greater are added, the result is, at least, the size of half the metagenome. It is used for quality assessment as an indicator assembly contiguity, considered generally the highest the better.

L50: Total number of contigs that have a length equal or superior to the N50 value; that is, they contain at least half the bases of the metagenome. This is also a quality assessment parameter that provides

contiguity information. Generally, the lowest its value, the better.

ANI: Average Nucleotide Identity, sequence coincidence between two microorganisms. The more related they are, the higher the ANI is. If ANI is very high, the assembly cannot differentiate the microorganisms involved.

Indel: Short for insertions and deletions of genomic bases in a sequence. In the context of assembly, these are mutations originated during the assembly, dependent on the assembly algorithm, that can cause truncated interpretation of protein regions through frameshifts or premature stop codons.

Mismatch: SNP (single nucleotide polymorphism), variants in the identity of a base in a genomic sequence. In the context of assembly, it consists on a wrong base assignment.

Polishing: Correction step for improving the accuracy of an assembly draft. Consists on using the input reads to obtain, through iterations, a consensus sequence that better applies to the sample.

Annotation: Functional analysis and assignation of the obtained sequencing data, to obtain the functional, expressing profile of the organism.

Biosynthetic gene cluster (BCG): Group of genes regulated together that are involved in a certain process that belongs to the secondary metabolism. They are repetitive and sensitive to frameshift mutations.

2. Introduction

2.1-Why should this topic matter in the current world?

Humankind has been using technology to increase its quality of life for centuries. From fire, to the wheel, to electricity, medicine and machines of every kind, constant efforts for increasing our well-being are constantly made. However, in the recent past those efforts have come with great environmental damage, depending mostly on fossil fuels and processes that, even though they have helped to achieve a great deal of products and possibilities not even imagined by our ancestors, they jeopardize the surroundings integrity and the possibility of maintaining our activities for a long time (Cavicchioli *et al.*, 2019).

Now, we live in a very dynamic world in which science evolves fast. Industry evolves fast. We search for advances to improve our performance or being able to develop new products, and we look for solutions to diminish our damage to the planet. There is a clear need for change in the way industry is built, how we obtain primary resources and energy; and a need to develop fast ways of obtaining relevant information, to obtain the most from the least (Lorenz and Eck, 2005).

Microorganisms are extraordinary creatures, that shape our environment and drive energy and matter flowing in collaboration with other organisms and geology. They live in virtually any environment and their presence affect the capability of an ecosystem to respond to a certain situation (Cavicchioli *et al.*, 2019; Ratzke and Gore, 2018). They possess an enormous variety of functionalities, which makes them treasures of industrial interest. Transformations not possible in other conditions and products incredibly difficult to imagine and obtain synthetically are made by microorganisms, and they can be modified to produce a wide range of molecules in an efficient way (Priscu and Christner., 2003; Correa and Abreu, 2020). Industrial usage of microorganisms is a very old practice, and now we are switching to a model in which we control the conditions and the products of this usage more precisely, from a proof-based to a rational-based methodology (Demain *et al.*, 2016; Cassidy *et al.*, 1996), for which metagenomics is proving to have a lot to contribute.

2.2-Glasses for reading: DNA sequencing

In the past few decades, we have been living in the "sequencing era". Sequencing is a technique by which a genome is "read", meaning, the sequence –order and identity– of its bases is determined. Many techniques have been developed for this aim, in what are known as

sequencing technologies (McCombie *et al.*, 2019). The beginning of the sequencing era culminated with the Human Genome Project, the sequencing of the human genome (Venter *et al.*, 2001).

According to the technicalities of the process and the timeline in which they have been developed, sequencing techniques are divided in three generations. The **first generation** started with Sanger sequencing, which was based on chain-termination method, that is very slow but accurate. Applications of Sanger sequencing have been centered on small genomes and target genes obtention (Metzker, 2010), but as omic sciences started to gain importance, there was a clear limitation in both speed and cost (McCombie *et al.*, 2019). **Second generation** platforms produced high throughput data of short reads by reading thousands of templates at the same time. It has been mostly represented by Illumina in the last years, which became the most used technology for '**omic'** analyses (Kim *et al.*, 2013; Metzker, 2010). The **third generation**, represented initially by Pacific Biosciences (Pacbio) and now by Oxford Nanopore Technologies (**ONT**), kept the high throughput concept and is based on outputting long reads—in contraposition to the short reads produced by Illumina that increase computational demands—obtained through single-molecule sequencing. The main drawback of this group has been its high error rate (Kim *et al.*, 2013; McCombie *et al.*, 2019). On the whole, DNA sequencing is a very versatile technique and its utility has only been increasing since '**omic'** studies arose and became a regular research approach.

2.3-Reading the microenvironment: metagenomics

Once we acknowledged the opportunities existing in the microbial world and we aim to understand those organisms, the arising problem is how to do so. Classical microbiology is focused on the isolation and growth of the microorganism. However, this approach is limited since most microorganisms on the planet cannot be cultured with traditional techniques –according to most estimations 99% are not culturable, and the culturable 1% is not representative of the whole bacterial diversity (Arya, 2020)–. In this context, metagenomics has received more and more attention in the recent years (Schloss and Handelsman, 2005; Handelsman, 2004). '**Omic'** studies are based on integrating data to obtain a descriptive, functional or overall picture. Typically, an omic study will employ lots of experimentally obtained data and will aim to obtain a general overview of what is being studied (Evans, 2000). They allow for understanding nature and systems in a holistic way. The term 'metagenomics' was born in 1998 when Handelsman *et al.* used it when describing the "collective genomes of soil microflora" (Highlander, 2014) (Figure 1). However, it was not until 2004 when the first two metagenomic

studies were published: one describing sequencing and analysis of a metagenome representing a handful of organisms forming an artificially simple community of a biofilm growing on the surface of an acid mine drainage (Tyson et al., 2004); and the other describing a metagenome of a much more complex community of the Sargasso Sea microbiome (Venter *et al.,* 2004). It has since increasingly acquired popularity, becoming a common research tool nowadays (Hugenholtz and Tyson, 2008; Jansson and Baker, 2016).

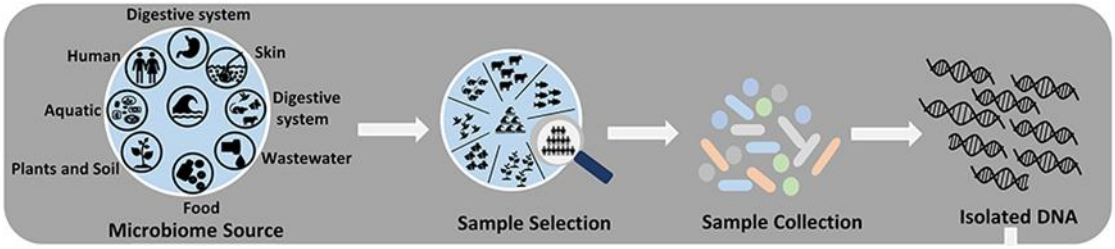


Figure 1 **The concept of metagenome.** Adapted from Bharti and Grimm, (2019).

Strictly speaking, '**metagenomics'** does not include every culture-independent, sequence-based approach for microbiology research, but only the 'shotgun' sequencing of the DNA from the whole community, and further assembly –puzzling up that sequencing data— or taxonomic/functional analysis. When marker genes (16S, 18S, ITS…) are amplified and sequenced in order to obtain taxonomic information of the sample, the approach used is '**metataxonomics'** or amplicon sequencing (Marchesi and Ravel, 2015); and when the focus is on sequencing the total RNA of the microbial community to see what is the expression profile of that community, the approach used is called '**metatranscriptomics'** (Breitwieser and Salzberg, 2019). Each strategy has its own advantages and disadvantages and is more suitable depending on the objective of the study (Tringe *et al.*, 2005; Breitwieser and Salzberg, 2019; Morgan and Huttenhower, 2012) (Table 1, Figure 2).

Metataxonomics is cheaper, faster and requires less computational resources than metagenomics, but the information that can be obtained from this approach is limited since it only focuses on one or few genes, and those genes are compared to databases that include already established knowledge (Breitwieser and Salzberg, 2019). Furthermore, for targeting only those few genes, the PCR primers used must be designed. This leads to biases towards species that are known in detriment of new species that might not be amplified because they contain genetic sequences different from the standards, thus failing to obtain the complete picture of the sample in some cases. The same problem applies when estimating sample abundances (Bharti and Grimm, 2019). Despite the bias, metataxonomics achieves better screening of less abundant microorganisms (Breitwieser and Salzberg, 2019) because the PCR amplification allows for targeting a taxonomic group, thus increasing screening capacity.

Using metagenomics is advantageous for estimating abundances more accurately, obtaining draft and reference genomes and reconstructing functional and ecological profiles (Tringe *et al.*, 2005), which is of most interest when dealing with unknow microbiomes, specially from the environment, that potentially contain new species and unique physiological functions. In those cases, shotgun metagenomics and assembly allow for a more complete qualitative analysis of the sample and obtention of new information in terms of new microbial species, combinations, and synergies; as well as detecting other features that affect the community, such as plasmids and viruses (Breitwieser and Salzberg, 2019). Technical limitations of this approach are related to both the experimental data obtention and analysis, but as sequencing and bioinformatic tools develop, the cost both monetary and computational is progressively decreasing, making it more affordable for regular scientists (Breitwieser and Salzberg, 2019; Bharti and Grimm, 2019; Tringe *et al.*, 2005).

Table 1. **Main differences between metataxonomics and metagenomics**. Adapted from Breitwieser and Salzberg, (2019).

Strategy	Concept	Advantages and challenges	Main applications
Meta taxon omics	Using amplicon sequencing of the 16S or 18S rRNA gene or ITS	+ Fast and cost-effective identification of a wide variety of bacteria and eukaryotes – Does not capture gene content other than the targeted genes – Amplification bias – Viruses cannot be captured	* Profiling of what is present * Microbial ecology * rRNA-based phylogeny
Meta gen omics	Using random shotgun sequencing of DNA or RNA	+ No amplification bias + Detects bacteria, archaea, viruses and eukaryotes + Enables de novo assembly of genomes – Requires high read count – Many reads may be from host – Requires reference genomes for classification –More expensive and machine demanding	* Profiling of what is present across all domains * Functional genome analyses * Phylogeny * Detection of pathogens * New microorganisms obtention

Finally, metatranscriptomics is the approach that actually confirms any information that either metataxonomics or metagenomcis can provide, because it allows for screening the real activity of the sample: which microorganisms are alive and which functionalities are being expressed (Breitwieser and Salzberg, 2019). The main limitation of this approach –although also bioinformatically challenging–is rather experimental, as it is based on mRNA obtention and

RNA is very fragile –samples are unstable and degrade fast–, which makes this analysis a difficult practice (Bharti and Grimm, 2019).

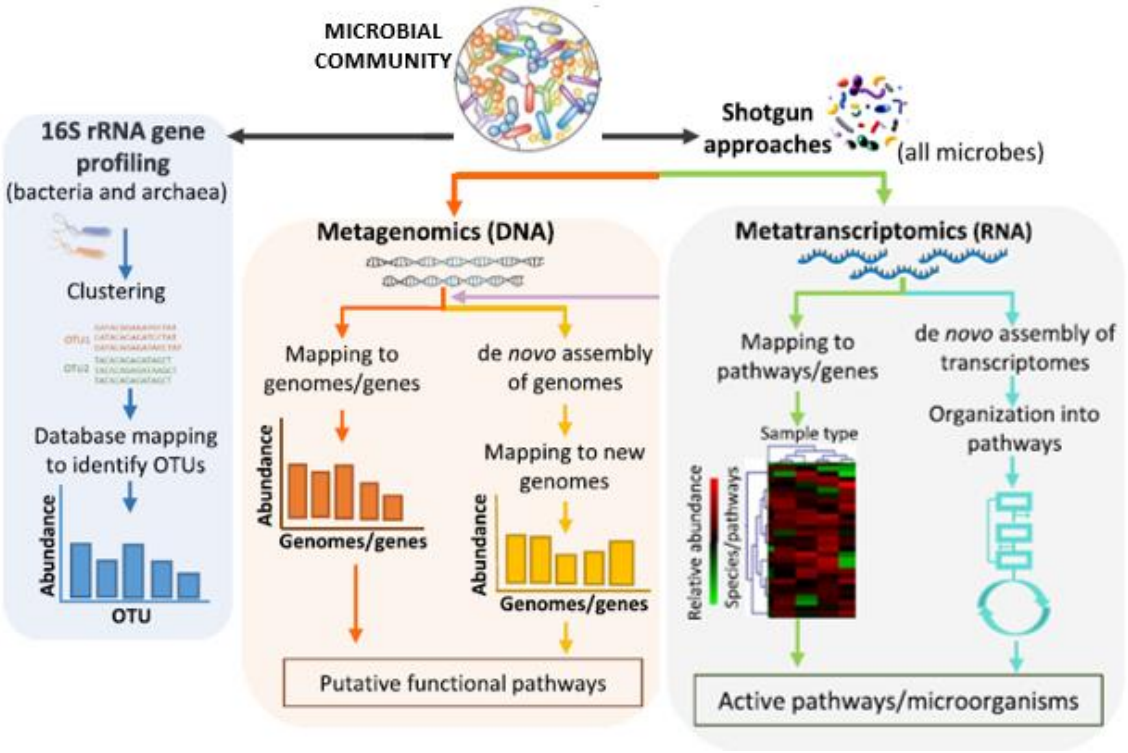


Figure 2. **Metataxonomics, metagenomcis and metatranscriptomics approaches overview and most important application.** Adapted from Bikel *et al.*, (2015).

Metagenomics therefore appears as the most convenient strategy for industrial development, I+D, product discovery and environmental assessment, that is, for bioprospecting, through sequencing and assembling the microbial DNA into genomic drafts (Priscu and Christner, 2004; Breitwieser and Salzberg, 2019); and improvement of this discovery approach needs for sequencing and computational development and optimization. **Assembling** –constructing draft genomes computationally from the sequencing data–properly the metagenome is crucial for good results obtention, and it is the first big computational step of the analysis. However, community complexity –which makes it difficult to discern whether a read is part of one microorganism or the other–and performance of the assembly programs in terms of screening capacity, metagenomic datasets, speed and computing resources usage, are variables that have to be considered when choosing an assembler or a sequencing technique (Teeling and Glöckner, 2012). In this regard, Illumina sequencing shows limitations.

2.4- Oxford Nanopore technologies: the revolution of long-read sequencing

Importantly, when dealing with metagenomic data, short reads increase drastically the computing power required for data analysis and assembly, and the resulting metagenomes are highly fragmented due to the difficulty on creating scaffolds –bridges between reads that serve to put them together and order them in the overall sequence— when the genomes have similar sequences, repetitive or complex genomic regions (Zavodna *et al.*, 2014; Kim *et al.*, 2013). As mentioned above, the major drawback in third generation sequencing is the high error rate they carry (McCombie *et al.*, 2019). Consequently, combining second and third generation sequencing has arisen as a solution for the drawbacks that both technologies carry, and has drastically increased assembly contiguity while maintaining accuracy (Bertrand *et al.*, 2019; Giguere *et al.*, 2020; Moss *et al.*, 2020), although at least two technologies and several processing steps need to be combined thus making the overall process more complex.

Among the long-read technologies, using Oxford Nanopore Technologies (ONT) produces longer fragments, increases assembly **contiguity**, and decreases false redundancies when comparing it to Pacific Biosciences (PacBio) (Lang *et al.*, 2020; Moss *et al.*, 2020). Although better accuracy is achieved by PacBio (Lang *et al.*, 2020), ONT solves computational processing by aiming for very long read size; and with new advances on the technology, it has become very suitable for metagenomic analysis. Firstly, its **chemistry** –conformation of the pore that reads the DNA base passing through it–and **basecalling** –obtaining the DNA base from the sequencing signal—are increasing accuracy thus diminishing error rates (OXFORD NANOPORE TECHNOLOGIES, 2020b; OXFORD NANOPORE TECHNOLOGIES, 2020c). Secondly, format variation of devices, namely the development of **MinION** devices, is making *in situ* applicability available, which allows for analyses to be made easily in many environments and situations –from forensics, clinical, environmental and in-space (Burton *et al.*, 2020; Burgess, 2020; Hall *et al.*, 2020; Chan *et al.*, 2020; Carradec *et al.*, 2020)–, as well as reducing the **cost** and **time** needed for sequencing, making it a cheap, portable, accessible and highly versatile technology with much potential to obtain highly complete microbial genomes from metagenomic samples (Deamer *et al.* 2016; Hasnain *et al.* 2020).

With the fast development of MinION and in general Oxford Nanopore Technologies, new assemblers are being developed to deal with their data. As there is not a standard guideline for best results when assembling metagenomes with ONT, doubts arise when scientists not familiarized with all tools attempt to assemble a metagenome and do not know which one to choose.

2.5-Evaluating what works best: benchmarking

A benchmark, or evaluation, is a type of study in which the objective is to measure the yield of one or more tools and use this knowledge to apply it to similar systems (Anand and Kodali, 2008). In bioinformatics, this approach is used to validate workflows, pipelines, strategies, and to test improvements of existing tools (Agers *et al.*, 2018; Goderis *et al.*, 2009; Angers-Loustau *et al.*, 2018).

For these types of studies, it is necessary to use reliable and known data, from which the interpretation of the results can be more accurate; and for that, it is useful to employ mock communities as microbiome genetic material source (Sun *et al.*, 2012; Leidenfrost *et al.*, 2020; Singer *et al.*, 2016). **Mock communities** are defined microbial communities, composed by known microorganism with an available reference genome (Highlander, 2014). Using a community whose composition you already know is most useful to asset how well a pipeline has worked. Moreover, the proportion of each organism in the mixture can be modified to simulate different complexities, harder to analyze (Bokulich *et al.*, 2016; Leidenfrost *et al.*, 2020). However, this type of approach has the risk of overfitting since it is based on an oversimplification of real microbial communities. For that reason, it is of most importance to use variated and more representative data to conduct benchmarking studies (Hawkins, 2004), and to keep validating the results said studies obtain.

There is a need for developing tools for other scientists to use and improve the research field, but there is also a need to validate these improvements and state how well they work by comparing them to other alternatives available, and provide less-experienced scientists on the field of guidelines they can follow, if we want to make science, and concretely bioinformatic analysis, more available (Aniba *et al.,* 2010). The world is digitalizing and biological sciences should not fall behind.

3. Objectives

Keeping in mind the discussed context and considerations, the **main objectives** of this project are to:

1. Delve into benchmarking ONT assembly tools, discerning which provide the best metagenomes from mock communities when dealing with ONT Nanopore data.

2. Assess Nanopore data capacity of reconstructing different microbial communities on its own.

And the **secondary objectives** are to:

3. Contribute to tools criteria homogeneity and provide guidance to other scientists, extending assembly analysis accessibility.

4. Stimulate rational development of tools and strategies for metagenomic assembly and ONT data analysis for better results obtention.

4. Materials and methods

4.1-Environment and background

This project was developed in coordination and under the supervision of Darwin Bioprospecting Excellence –referred as Darwin throughout the text–, that provided the *in silico* equipment for analysis, and guidance. In the bioinformatics department of Darwin, research regarding benchmark of metagenomic assemblers using MinION reads was already being made. In that regard, this project was originated as a continuation and deepening of the previous work done by Latorre-Pérez *et al.* (2019), in which they assembled data of mock communities generated by Nicholls *et al.* (2019). *Due to the Covid19 lockdown situation, the non-essential access to the Universitat de València Parc Científic -in which Darwin is located- was forbidden and I did all of my activities from my home. To access the computer from Darwin I used the remote PC controller TeamViewer.*

4.2-Overall project workflow

Briefly, sequencing data from metagenomic mock communities was obtained, assembled, polished, and results were evaluated in function of contiguity and accuracy parameters (Figure 3).

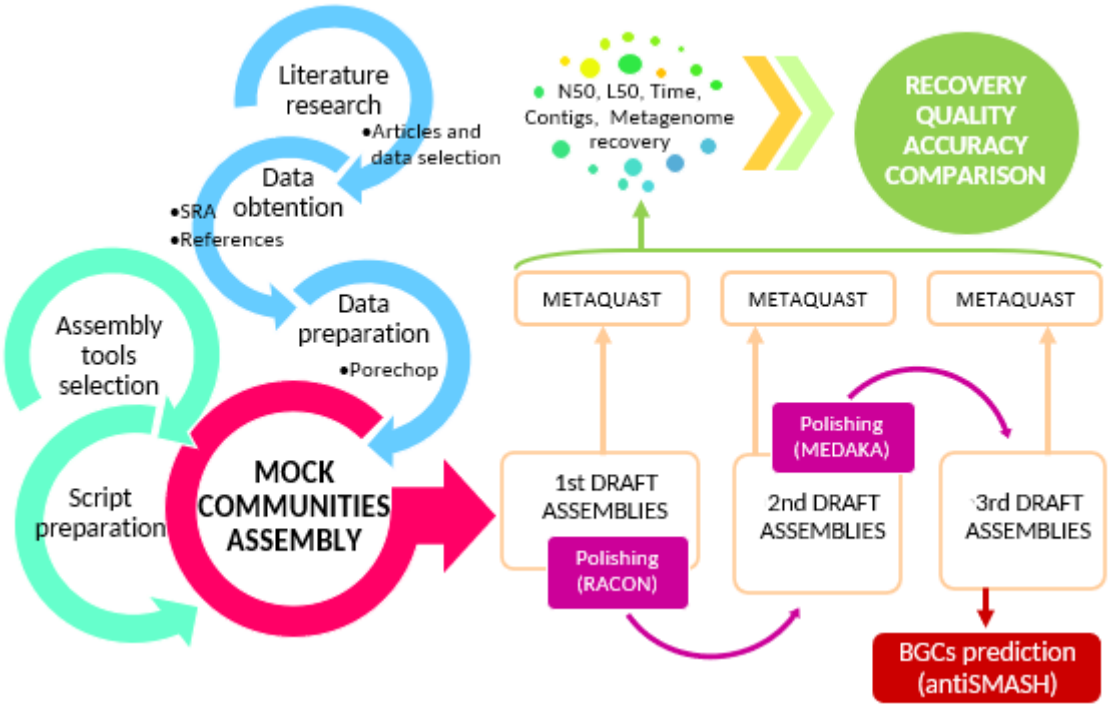


Figure 3. **General overview of the Project methodology.**

4.3-Nanopore data obtention.

When this project was started, the ongoing situation —the COVID-19 pandemic crisis and lockdown restrictions in Spain— made it not possible to work on the laboratory and obtain data for the study. Despite these initial constraints, this issue could be solved thanks to nowadays scientific policies on OpenAccess and public data availability, as researchers are required to make the data they have used for their study accessible on public repositories such as SRA (under BioProject accessions) or ENA (in EMBL). This cloud system allows for available data to be reused using different study approaches. Data of sequenced mock communities that had not been analyzed using the same strategies or in the same depth as herein arose as the most convenient option for performing the whole project *in silico,* and were employed instead. Therefore, metagenomic Nanopore sequencing data was searched in the literature to find studies providing sequencing data. The complete list of results can be consulted in Supplementary Figure 1.

The first criteria for studies selection included:

- Metagenomic studies.
- Data generated from real communities using Oxford Nanopore MinION, PromethION or GridION -preferably MinION-, using flowcells with R9.4 pore.
- Aim for data of mock communities (defined and validated, with reference genomes available).

The search was made through the official Oxford Nanopore Technologies site -in which every study that is related to or uses their technologies is published- using the 'metagenomic' filter and the 'most recent' order, and through Google Scholar using key words such as 'ONT', 'Nanopore', 'MinION', 'Metagenomics', 'Mock community', and filtering by most recent publications -1 and 2 years-. Pubmed and PLOS were searched but they included the results already found in either Google Scholar or Oxford Nanopore. Elsevier provided only general titles, mainly not relevant. The title and abstract of the retrieved results were revised, and if relevant information was found, the full text was accessed and read. A list containing study details of the results was elaborated, and the 9 most promising studies were selected for further filtering.

The list was then filtered according to more restrictive criteria:

- Data had to be available to download.
- Sequencing depth had to be as high as possible, ideally greater or equal to 4Gbp.

- The basecaller had to be preferably Albacore if raw signal files were available, or Guppy if no signal files were available.

Finally, three studies generating Nanopore sequencing data were selected (Leidenfrost *et al.*, 2020; Sevim *et al.*, 2019; Moss *et al.*, 2020). Fastq files (Table 2) were downloaded from the publicly available SRA repositories, that were searched using the BioProject IDs provided by the researchers. Direct download was not available for the data generated by Sevim *et al.* (2019). In that case, the fastq reads were only available as an SRA file, thus the SRA toolkit was used to obtain the data and transform the file from SRA to fastq format using the commands "prefetch" and "fastq-dump".

Table 2. **Summary of data size and obtention.**

Study	fastq obtention	fastq size	Dataset details
Leidenfrost *et al.*, (2020)	Direct download from SRA	4,6 GB and 7,9 GB	Both datasets
Sevim *et al.*, (2019)	Direct download from SRA	7,5 GB	10kb size-selected
Moss *et al.*, (2020)	Obtained using SRA toolkit	62,3 GB	atcc dataset

4.4-Mock communities and reference genomes obtention

Leidenfrost *et al.* (2020) generated a community that is a mixture of 12 type strains of gram positive and gram negative bacteria with varying GC content available in either the German Collection of Microorganisms and Cell Cultures GmbH (DSMZ), the National Collection of Type Cultures (NCTC) or the American Type Culture Collection (ATCC), with published reference genomes at the National Center for Biological Information (NCBI), that was blindly sequenced by a different sequencing laboratory. Two samples were generated prior to sequencing this community: one sample with equimolar microorganism composition –referred in the text as **BenchEV**–, and one sample with logarithmic scale-like heterogeneous microorganism composition –referred in the text as **BenchHE**– (Table 3).

The community generated by Sevim *et al.* (2019) is also a mixture of 12 bacterial strains, but since the aim of this work was to generate sequencing data for different technologies, it was designed to be especially difficult to sequence. It contains variations in terms of genome size, GC content, repeat content; and taxonomic complexity factors such as several strains belonging to the same genus of *Actinobacteria* with high average nucleotide identity (ANI), two different proteobacterial classes –alpha and gamma–, and two different phyla –*Flavobacteria* and *Actinobacteria*. This community is here referred with the same name as the authors used in their study, **BMock12** (Table 3).

Table 3. **Mock communities compostition and IDs.** (1) All are IDs from Refseq. (2) All are taxomnomic IDs from IGM. (3) All are IDs from Refseq, except from that of *Yersinia* enterocolitica, that was only available at the ATCC website.

COMPOSITION OF COMMUNITIES					
BenchEV and BenchHE		**BMock12**		**MSA2006**	
Size (bp)	56430306	Size (bp)	59520276	Size (bp)	46750506
Estimation	56.4 MB	Estimation	59 MB	Estimation	47 MB
NAME	**ID (1)**	**NAME**	**ID (2)**	**NAME**	**ID (3)**
Xanthomonas campestris	NC_003902.1	*Muricauda ES.050*	2615840527	*Bacteroides fragilis*	NC_003228.3
Chromobacterium violaceum	NC_005085.1	*Thioclava ES.032*	2615840533	*Bacteroides fragilis pBF9343*	NC_006873.1
Corynebacterium glutamicum	NC_003450.3	*Cohaesibacter ES.047*	2615840601	*Bacteroides vulgatus*	NC_009614.1
Staphylococcus saprophyticus	NC_007350.1	*Propionibacteriaceae bacterium*	2615840646	*Fusobacterium nucleatum*	NZ_CP028101.1
Staphylococcus saprophyticus pSSP1	NC_007351.1	*Marinobacter LV10R510-8*	2615840697	*Salmonella enterica*	NC_006511.1
Staphylococcus saprophyticus pSSP2	NC_007352.1	*Marinobacter LV10MA510-1*	2616644829	*Helicobacter pylori*	NC_000915.1
Bacillus licheniformis	NC_006270.3	*Psychrobacter LV10R520-6*	2617270709	*Escherichia coli*	NZ_CP009685.1
Micrococcus luteus	NC_012803.1	*Micromonospora echinaurantiaca*	2623620557	*Enterobacter cloacae*	NC_014121.1
Paenibacillus odorifer	NZ_CP009428.1	*Micromonospora echinofusca*	2623620567	*Enterobacter cloacae pECL_A*	NC_014107.1
Serratia fonticola	NZ_CP011254.1	*Micromonospora coxensis*	2623620609	*Enterobacter cloacae pECL_B*	NC_014108.1
Achromobacter xylosoxidans	NZ_LN831029.1	*Halomonas HL-4*	2623620617	*Lactobacillus plantarum*	NC_004567.2
Dickeya solani	NZ_CP015137.1	*Halomonas HL-93*	2623620618	*Enterococcus faecalis*	NC_004668.1
Enterobacter hormaechei subsp. Steigerwaltii	NZ_CP017179.1			*Enterococcus faecalis pTEF1*	NC_004669.1
Cronobacter sakazakii	NZ_CP011047.1			*Enterococcus faecalis pTEF2*	NC_004671.1
Cronobacter sakazakii CSK29544_1p	NZ_CP011048.1			*Enterococcus faecalis pTEF3*	NC_004670.1
Cronobacter sakazakii CSK29544_2p	NZ_CP011049.1			*Clostridioides difficile*	NZ_LN614756.1
Cronobacter sakazakii CSK29544_3p	NZ_CP011050.1			*Clostridioides difficile pCD630*	NC_008226.2
				Bifidobacterium adolescentis	NC_008618.1
				Yersinia enterocolitica	ATCC 27729

Lastly, the community used by Moss *et al.* (2020) is also a mixture of 12 bacteria, of equal composition (8.3%), that was obtained from the ATCC –distributed under the item name "MSA-2006" –. This community was designed for human gut microbiota experiments; therefore, it contains bacteria typically present in human gut microbiota, among which there are bacteria from the same genus. Here it is referred as **MSA2006** (Table 3).

All reference genome sequences were obtained from either the National Center for Biotechnology Information (NCBI) Refseq or the Integrated Microbial Genomes and Microbiomes (IGM) databases. As a rule of thumb, NCBI reference sequences were chosen over other platforms, unless the sequence was not available. In the case of the BMock12 community, all sequences were obtained from IGM (as specified by the researchers in their work). Reference metagenomes were independently constructed by concatenating all the genomes and plasmids comprising each mock community. After creating the reference file, the total metagenome size was calculated and used as input for the genome size parameter of the assemblers, when necessary (Table 3).

4.5-Data preprocessing

Before proceeding with the assembly, the adapters added to the sample DNA for sequencing must be removed to avoid including them in the metagenome. Porechop (https://github.com/rmcolq/Porechop) is the most established tool for adapter trimming and removal of Oxford Nanopore reads.

Therefore, all fastq files were processed using the following command line:

```
porechop -i reads.fastq -o reads_after.fastq -t 16
```

Where the parameter **-i** is for the input file, **-o** is for the name of the processed file and **-t** is for thread number.

The MSA2006 sample file was too big to be run in one single round because there was not enough RAM in the computer. In that case, reads were split in two files and each file was processed through Porechop and then concatenated to create one single file.

4.6-Metagenome assembly

Although many assembly programs exist, most of them are designed for short reads, and there are not many that are suitable for third generation sequencing data. With recent improving of the technology, more have been developed. Among those, a selection was made for this study.

Briefly, criteria for selection of assembler programs were:

- Support ONT long-reads.
- Reported to give good results from both the literature and the previous project of assemblers' evaluation conducted in Darwin (Latorre-Pérez *et al.*, 2019).
- Free to download and use, and public documentation available.

Assemblers meeting these criteria were downloaded or updated to the newest available release. The complete list of assemblers was: Flye (Kolmogorov *et al.*, 2019a; Kolmogorov *et al.*, 2019b), Canu (Koren *et al.*, 2017), Pomoxis (OXFORD NANOPORE TECHNOLOGIES, 2018), Raven (Vaser and Šikić, 2019), Redbean (Ruan and Li, 2020) and Necat (Chen *et al.*, 2020). Among the selected assemblers, Necat was the only not being included in Latorre-Pérez *et al.* (2019) since it was released in February 2020. Assembler details are collected in Table 4.

Documentation and usage for each of the assemblers was consulted in their GitHub repositories and official sites, and tools were run using the recommended parameters for metagenomic assembly. When authors did not mention any specific configuration for metagenomics, default parameters were used (Table 4 and Table 5). These criterion was established to use fair base conditions for all the assemblers, and under the reasoning that assembler developers are responsible for the development and optimization of these tools for other scientists that might, or not, be advanced in bioinformatics and need to use them.

The order in which the communities were assembled –and the rest of the analyses were performed–was the following: BenchEV, BenchHE, BMock12, and lastly MSA2006. In all cases, equivalent parameters were filled with the same values, and all assemblers were run at the maximum computation power that the computer was capable of (16 threads).

***Table 4.* Assembly tools characteristics.** Metagenomic settings availability was screened for best performance configuration. Metagenome size estimation requirement was noted.

Assembler	Version	Metagenomic settings in documentation?	Recommended settings for metagenomics	Required size estimation?
Flye	2.7	Yes	–plasmids –meta	Yes
Canu	2.0	Yes	corOutCoverage=10000 corMhapSensitivity=high corMinCoverage=0 redMemory=32 oeaMemory=32 batMemory=200	Yes
Pomoxis	0.3.2	No	Not specified	Yes
Raven	1.1.5	No	Not specified	No
Redbean	2.5	No	Not specified	Yes
Necat	First available	No	Not specified	Yes

***Table 5.* Complete commands for each assembler used**. $filepath: absolute path of the input reads; $resultdir: absolute path of the results directory (folder);$size: estimate genome size of the community (an approximate value).

Assembler	Complete command
Flye	flye –nano-raw $filepath –out-dir $resultdir/Flye –genome-size $size –threads 16 –meta –plasmids
Canu	canu -p assembly -d $resultdir/Canu genomeSize=$size corOutCoverage=10000 corMhapSensitivity=high corMinCoverage=0 redMemory=32 oeaMemory=32 batThreads=16 batMemory=60 -nanopore $filepath
Pomoxis	mini_assemble -i $filepath -o $resultdir/Pomoxis -p assembly -l $size -t 16
Raven	raven –threads 16 $filepath > assembly.fa
Redbean	wtdbg2 -x ont -t 16 -g $size -i $filepath -fo step1 ; wtpoa-cns -t 16 -i step1.ctg.lay.gz -fo assembly.ctg.fa
Necat	necat.pl bridge config.txt

Not all assemblers employed required a size estimation of the genomes, which can be useful in studies in which the sample is completely unknown and size estimation is tricky and can introduce bias in the assembly. Coverage parameters were not changed unless specified

otherwise in the recommendations, because coverage is also a rough estimate when dealing with unknown data, as it depends on genome size.

Once settings were defined, a bash script was prepared to run all assemblers independently but automatically; count and store run times, store log data –program status data displayed otherwise on the terminal –and avoid time losses between runs. The complete bash script is included as Supplementary code 1.

4.7-Assessing assembly quality

The quality statistics of the draft assemblies –before and after polishing –were assessed using MetaQUAST (Mikheenko *et al.*, 2016; Gurevich *et al.*, 2013). Assemblies judgement and comparison was made in terms of performance and accuracy, using statistics provided by MetaQUAST. Regarding performance, parameters considered were: clock **running time**, percentage of **recovered genome** –total amount of the metagenome that was assembled–, and contiguity parameters such as total generated **contigs** –number of assembly fragments–, **N50** –size such as half of the metagenome is in contigs that are larger or equal than this size–, and **L50** –number of contigs whose added size is equal or larger to the N50. With respect to accuracy, statistics of **Indels** –insertion or deletion of bases, that can cause frameshift mutations and premature stop codons–and **mismatches** –SNPs, that can cause point mutations–per 100kbp of assembly were used.

Even though MetaQUAST requires reference genomes as input, it also provides statistics calculated without those reference genomes (i.e. N50, L50, number of contigs…).

MetaQUAST commands were directly run in the shell as follows:

```
metaquast.py -o $outdir -r $refsfolder -L –unique-mapping $file1 $file2…
```

Where the parameter **-o** is for the output directory path; **-r** is for a folder including all individual reference genomes; **-L** is for using folder name in the report graphics; **–unique-mapping** is for forcing **–ambiguity-usage** 'one' in metaquast, which uses only best score alignments; **$file1, $file2**… are the location paths for each assembly draft.

4.8-Polishing

After first draft assemblies were obtained, a polishing step was performed to assess the impact of polishing in draft metagenomes obtention and whether the improvement of assembly

accuracies improves overall accuracy results more in certain assemblers. The central pipeline followed was according to Oxford Nanopore Technologies recommendations of using one round of Racon (Vaser *et al.*, 2017) and one of Medaka (OXFORD NANOPORE TECHNOLOGIES, 2020a; https://github.com/nanoporetech/medaka). A comparison was then made between draft assemblies, drafts resulting from using one round of Racon, and drafts resulting from one round of Medaka after Racon. Complete bash scripts for Racon and Medaka polishing can be found as Supplementary code 2 (Racon) and Supplementary code 3 (Medaka).

4.9-BCGs prediction.

Biosynthetic gene clusters (**BCGs**) are genomic regions which encode biosynthetic pathways for the production of specialized metabolites (Medema *et al.*, 2015). BCGs often consist of several kilobases and contain repetitive material, thus complicating the assembly process when using short reads. Despite increasing assembly contiguity, which is advantageous for BGC obtention, ONT reads are prone to errors -especially Indels- which truncate BGC prediction (Miller *et al.*, 2017). Therefore, overall assembly performance and polishing success can be qualitatively evaluated through BGCs prediction (Watson and Warr, 2019). For assessing how well each assembly tool achieved functional depiction of the communities, an annotation analysis was made using **antiSMASH** bacterial version (Blin *et al.*, 2019; Medema *et al.*, 2011). Results obtained for each draft were compared to the BCG profile obtained for each reference metagenome. Restrictions were set as 'relaxed', which allows for the detection of well-defined and partial clusters, with all or missing a few functional parts.

4.10-Equipment details

All analyses performed throughout this work were run in a computer from Darwin, whose characteristics are: CPU: AMD RYZEN 7 1700X 3.4GHZ; Cores: 8; Threads: 16; RAM: Corsair Vengeance 64 GB; SSD: Samsung 860 EVO Basic SSD 500GB; HDD: x2 Toshiba Canvio Basics 2Tb, with Ubuntu 18.04 operating system.

These working conditions are not the most powerful but belong to an average computing power that many research teams can achieve. They were considered the most appropriate for running the analysis because the MinION technology is meant to be widely spread among regular scientists for fast analysis in different laboratory environments, and computational resources for downstream analysis should be designed accordingly.

5. Results and discussion.

Throughout this work, the performance of six assembly tools designed to handle ONT sequences (Canu, Flye, Pomoxis, Raven, Redbean and Necat) was evaluated for metagenomic assembly. This benchmark was carried out using data generated from four different mock communities: BenchEV, BenchHE (Leidenfrost *et al.*, 2020), BMock12 (Sevim *et al.*, 2019) and MSA2006 (Moss *et al.*, 2020). In total, 21 first-draft metagenome assemblies were obtained during this work, that were evaluated in terms of contiguity and accuracy. Note that in each section the main remarks are highlighted in a dotted square.

5.1-General performance

5.1.1-Recovered metagenome fraction.

The first evaluation was in terms of overall (Figure 4) and individual (Figure 5) genome fraction recovery, which is the amount of DNA of the metagenome reference that is also present in the assembly.

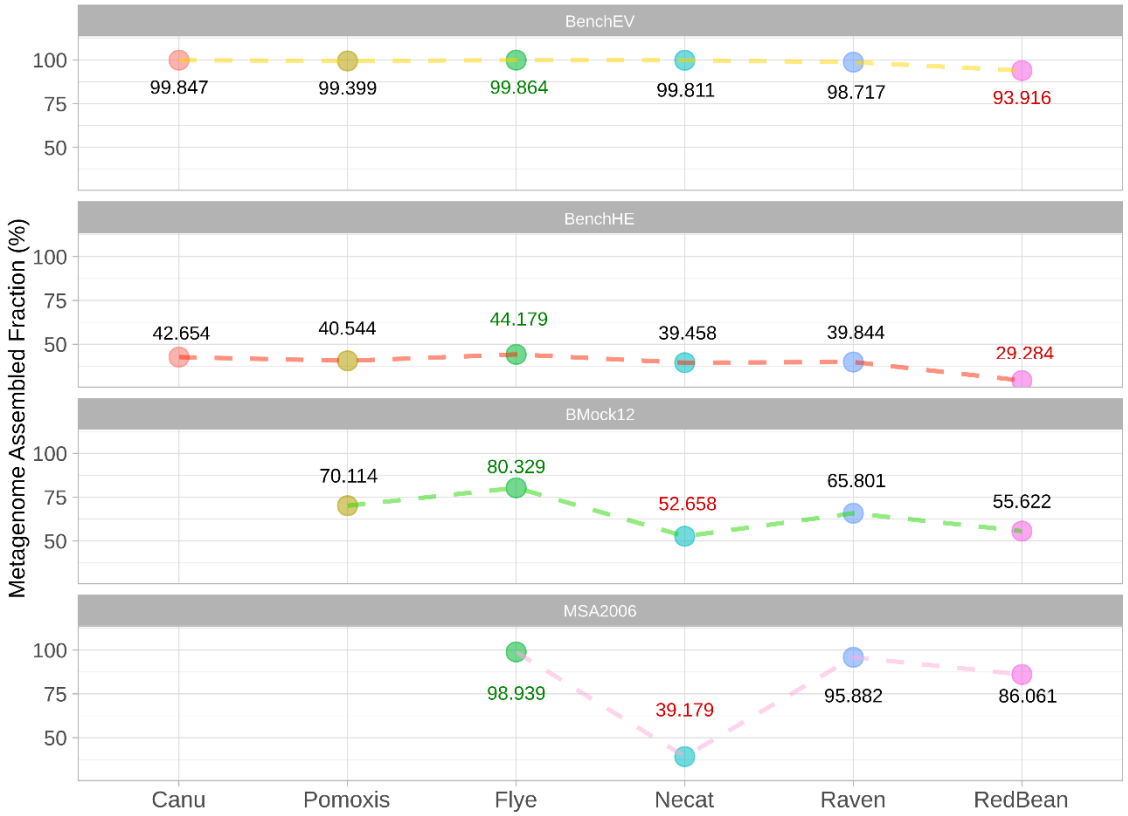


Figure 4. **Total metagenome assembled fraction per assembly and community**. Best values are highlighted in green and worst values are highlighted in red. Missing points are due to assembly failures and/or stopping (data after Polishing).

Microorganisms

	BenchEV						BenchHE					
	Canu	Flye	Necat	Pomoxis	Raven	Redbean	Canu	Flye	Necat	Pomoxis	Raven	Redbean
C. glutamicum	99.05	99.05	99.04	99.05	99.05	99.00	99.03	99.05	99.05	99.05	99.05	0.27
B. licheniformis	100.00	100.00	99.89	99.82	100.00	99.83	99.99	100.00	99.93	100.00	99.99	100.00
X. campestris	99.91	99.93	99.89	99.93	99.89	99.73	7.86	11.90	-	0.56	-	3.64
E. hormaechei	99.99	100.00	99.93	99.58	99.31	75.88	99.85	99.33	86.38	94.43	90.41	98.67
S. fonticola	100.00	100.00	100.00	99.71	99.51	93.59	100.00	99.95	100.00	100.00	99.55	99.81
A. xylosoxidans	99.83	99.99	99.88	99.78	99.90	99.00	0.25	-	-	-	-	0.01
M. luteus	98.08	98.08	98.08	98.08	98.08	97.50	28.63	55.31	-	5.85	0.73	7.73
Cr. sakazakii	99.95	99.44	99.50	96.15	87.84	71.04	99.97	99.61	99.73	99.99	99.78	23.88
S. saprophyticus	100.00	100.00	100.00	100.00	99.99	99.75	-	-	-	-	-	-
Ch. violaceum	99.92	99.92	99.91	99.76	99.75	99.47	-	-	-	-	-	-
P. odorifer	100.00	100.00	99.99	99.91	100.00	99.66	-	-	-	-	-	-
D. solani	100	100	100	100	100	94.54	-	-	-	-	-	-

	BMock12				
	Flye	Necat	Pomoxis	Raven	Redbean
Cohaes. ES047	99.65	100.00	99.85	99.85	99.59
Halomonas HL4	99.48	94.75	99.98	97.74	21.18
Halomonas HL93	88.99	95.68	86.40	98.19	16.69
Marin. LV10M	99.96	100.00	100.00	100.00	97.77
Marin. LV10R	100.00	99.98	100.00	99.97	98.55
Micr. DSM43904	78.27	-	29.65	24.98	20.53
Micr. DSM43913	90.89	-	45.99	36.39	22.87
Micr. DSM45161	-	-	-	-	-
Muric. ES050	100.00	100.00	99.99	100.00	100.00
Propion. ES041	100.00	1.31	92.29	81.41	70.08
Psychr. LV10R	100.00	99.88	100.00	100.00	99.24
Thioclava ES032	99.76	99.57	99.64	99.64	99.76

	MSA2006			
	Flye	Necat	Raven	Redbean
Enterobacter	99.56	9.07	95.64	83.90
Bacter. 9343	99.99	99.94	99.93	95.72
Bifidobacterium	100.00	-	99.70	57.55
Clostridioides	90.33	90.37	90.34	85.60
E. coli K12	99.82	-	82.06	75.13
Bacter. 8482	99.66	-	99.06	95.26
Salmonella	99.90	3.26	98.22	85.26
Fusobacterium	99.99	-	99.77	29.31
Helicobacter	100.00	96.17	100.00	99.42
Lactobacillus	100.00	100.00	99.99	99.78
Enterococcus	100.00	100.00	99.48	99.56
Yersinia	100.00	1.45	98.39	97.28

Plasmids

	BenchEV						BenchHE					
	Canu	Flye	Necat	Pomoxis	Raven	Redbean	Canu	Flye	Necat	Pomoxis	Raven	Redbean
Cr. sakaz. CSK1	100.00	99.99	100.00	100.00	99.90	98.73	100.00	99.99	100.00	99.72	99.96	9.10
Cr. sakaz. CSK2	99.98	99.94	-	98.95	-	-	100.00	99.96	-	-	-	-
Cr. sakaz. CSK3	100.00	100.00	100.00	99.68	99.96	100.00	100.00	100.00	68.95	99.71	99.56	100.00
S. sapr. pSSP1	99.99	99.99	99.99	99.65	99.75	58.02	-	-	-	-	-	-
S. sapr. pSSP2	99.92	99.94	99.92	99.97	99.68	23.42	-	-	-	-	-	-

	MSA2006			
	Flye	Necat	Raven	Redbean
Enter. pECLA	99.47	61.95	98.20	84.15
Enter. pECLB	100.00	83.13	-	99.42
Enteroc. pTEF1	100.00	100.00	38.14	-
Enteroc. pTEF2	99.99	100.00	86.85	-
Enteroc. pTEF3	100.00	99.99	98.75	-
Bacter. pBF9343	100.00	100.00	99.78	5.20

Figure 5. **Genome fractions (%) recovered per microorganism (up) and plasmids (down) for each mock community.** Each box is filled proportionally with its recovery fraction, and blank spaces represent lack of recovery. Assembly success is proportional to fullness ratio. Note that BenchEv and BenchHE share the same microorganisms. Complete microbial names were not included for space issues, but can be consulted in Table 3. (Data after Polishing).

The **BenchEV** mock community contained a total of 17 genomes from 12 bacteria and 5 plasmids. Most assemblers performed well, being the worst recovery percentage of 93.916%, by Redbean. The highest genome fraction was recovered by Flye (99.864%), followed by Canu (99.847%) and Necat (99.811%) (Figure 4). All bacteria and plasmids were successfully assembled except from the *Cronobacter* plasmid CSK2, that was only recovered by Flye, Canu and Pomoxis (Figure 5). The worstly recovered organism by Flye and Canu was *Micrococcus luteus* (99.08%), as well as by Necat (98.077%). For Pomoxis and Raven, it was *Cronobacter sakazakii* (96.153% and 88.199%, respectively); and for Redbean, it was the plasmid pSSP2 from *Staphylococcus saprophyticus* (23.424%). Except from Redbean and Raven, all assemblers recovered more than 90% of all microorganisms.

Due to the heterogeneous log-scale like composition of the **BenchHE** mock community (Leidenfrost *et al.*, 2020), not all microorganisms had enough coverage to be assembled, and many were very little or not recovered. The highest overall recovery was achieved by Flye (44.179%) and Canu (42.654%), followed by Pomoxis (40.544%), Raven (39.844%), and finally Redbean (29.284%) (Figure 4). *Paenibacillus odorifer*, *Corynebacterium glutamicum*, *Dickeya solani* and *Spathylococcus saprophyticus* could not be recovered, which is of note because the least recovered bacteria (*Chromobacterium violaceum*, *Paenibacillus odorifer*, *Acromobacter xylosoxidans*, *Dickeya solani*, *Staphylococcus saprophyticus*, *Xantomonas campestris* and *Micrococcus luteus*) were coincident with the bacteria described by Leidenfrost *et al.* (2020) to be on the lowest concentration in the sequenced samples. When not enough coverage is achieved, assemblers do not have enough sequences to form an assembly for that microorganism and it ends scarcely assembled or undetected. Again, the CSK2 plasmid from *Cronobacter sakazakii* was not recovered by all assemblers, but only by Flye and Canu (Figure 5). In this case, however, the CSK3 plasmid was also roughly obtained by Necat. Although Flye and Canu had the best overall recovery rates, Flye recovered better the two partially recovered microorganisms *Xantomonas campestris* and *Micrococcus luteus* (Figure 5).

The **BMock12** community was composed of 12 bacterial genomes, with many similarities between them. Given the high assembly times and the community complexity described by Sevim *et al.* (2019) the expectations were that the results would be worse or similar to the BenchHE assembly, but the overall recovery percentage was ranged from ~51%, (Necat) to ~79% (Flye) (Figure 4). However, there were originally three strains of *Micromonospora* in the community, and one of them, DSM45161, was not assembled by any tool (Figure 5). *Micromonospora* strains were purposely put in the community due to their high ANI (Average Nucleotide Identity, the fraction of coincident genomic nucleotides between two

microorganisms), anticipating that they could lead to assembly inaccuracies. The two remaining strains of *Micromonospora* were the least recovered microorganisms, being only recovered in a range of 20 to 46% by Pomoxis, Redbean and Raven, not discriminated by Necat, and only recovered in a range of 78 to 91% by Flye (Figure 5). These results are tricky because contigs belonging to DSM45161 were most probably mapped into the other two *Micromonospora* strains. Different strains are very difficult to distinguish by assemblers because high ANI can make differences between strains seem as polymorphisms or be as low as the sequencing basal error rate. Necat failed to obtain all *Micromonospora* strains and *Popionibacteriaceae ES.041*, while the rest of assemblers obtained more than 70% of this last microorganism, and at least part of two of the *Micromonospora* strains. Nevertheless, Raven and Necat were better screening the *Halomonas HL93* than the rest, having more than 98% recovered in the case of Raven and 95% in the case of Necat, while Flye and Pomoxis only achieved 86 to 89% of recovery and Redbean roughly achieved 16% (Figure 5). Flye, was the only tool having more than 78% in all microorganisms except for *Micromonospora DSM45161*.

A total of 18 genetic sequences (12 bacterial genomes and 6 plasmids) composed the **MSA2006** mock community. Since this dataset was obtained from a defined ATCC community and had a defined even distribution –8.3% for each microorganism— (Moss *et* al., 2020; ATCC, 2019) the complexity of this assembly relied more on the high amount of data that was used as input (Table 2). The original dataset consisted of ~60 Gb and failed to be assembled (Supplementary data S2). The reduced ~30Gb dataset was successfully assembled by all assemblers but Pomoxis. Despite the high coverage of the data, only Flye and Raven recovered a significant fraction of the metagenome (98.888% and 95.776%, respectively) (Figure 4). Necat performance was dramatically diminished for this dataset (only 39.119% of the metagenome was recovered): it retrieved neither *Bacteroides 8482, Bifidobacterium, Escherichia coli,* or *Fusobacterium nucleatum*; and less than 10% of *Enterobacter* and *Salmonella*, while all *Enterococcus* plasmids were fully or almost fully recovered (Figure 5). Redbean failed to assemble all *Enterococcus* and *Bacteroides* plasmids and recovered *Fusobacterium nucleatum* in less than 30%. Raven, on the other hand, failed to assemble one of *Enterobacter*'s plasmids and roughly assembled one of *Enterococus*'s. Finally, Flye had the best overall results, with a genome fraction recovery over 90% in all genomes, and over 99% in all microorganisms but *Clostridioides*.

In overall, the most completely recovered community was BenchEV, then MSA2006, then BMock12 and finally BenchHE, in agreement with their taxonomical and ANI complexities. Flye had the highest or among the highest recovery percentages in all datasets, and the higher recovery percentages for microorganisms that were more roughly obtained by the other assemblers (Figure 4 and Figure 5). Redbean, on the contrary, had the worst recovery percentages. Necat and Canu had good recovery in the datasets they could function, but their performance was truncated due to dataset size or complexity in the case of Necat; and due to dataset complexity in the case of Canu, as it will be further explained in the next section. Raven and Pomoxis remained constant and gave overally good results although Pomoxis failed to run the MSA2006 dataset. From these data, the most reliable assembler for diversity screening appears to be Flye, followed by Raven, Necat and Pomoxis.

5.1.2.-Contiguity and computational efficiency

The second evaluation of the assembly tools was in terms of computational resources and contiguity (Figure 6). Running time was used for computational resources, and contiguity metrics such as the length containing at least half of the assembled metagenome, N50, the number of contigs containing at least half of the assembled metagenome, L50, and total obtained contigs were used for evaluating whether the assembly tools had been able to recover contiguous genomes or not. Although N50 and L50 statistics are good parameters for continuity assessment, there has to be kept in mind that no straightforward conclusion can be made only taking into account these statistics, as they do not consider the amount of total metagenome assembled but only length and contig number. Furthermore, they depend on the total assembled metagenome size, which can lead to have good parameters –long N50 and low L50–in a very incomplete assembly.

With regards to the computational resources, Canu was, by far, the slowest and least versatile assembler: it took nearly 12 and half hours to assemble the **BenchEV** community while Flye, Pomoxis, Raven and Redbean took less than an hour; and more than 5 whole days to assemble the **BenchHE** community, while the rest of the assemblers took less than 3 hours (Figure 6). After the BenchHE assembly, it was decided that Canu would be stopped every time the assembly was more than 6 times slower than the second slowest assembler, and therefore it was stopped for both the **BMock12** and the **MSA2006** assemblies. On the contrary, Redbean was the fastest assembler in all cases: it took 16 minutes for the BenchEV dataset and the highest amount of time it needed was 30 minutes for the MSA2006 dataset. In accordance with dataset

complexity and size, the BenchHE mock community assembly times were importantly higher with respect to those of the BenchEV community (Figure 6), BMock12 and BenchHE times were similar, and the higher assembly times were those of the MSA2006 assembly. After Redbean, Raven, and then Flye, were the fastest assemblers.

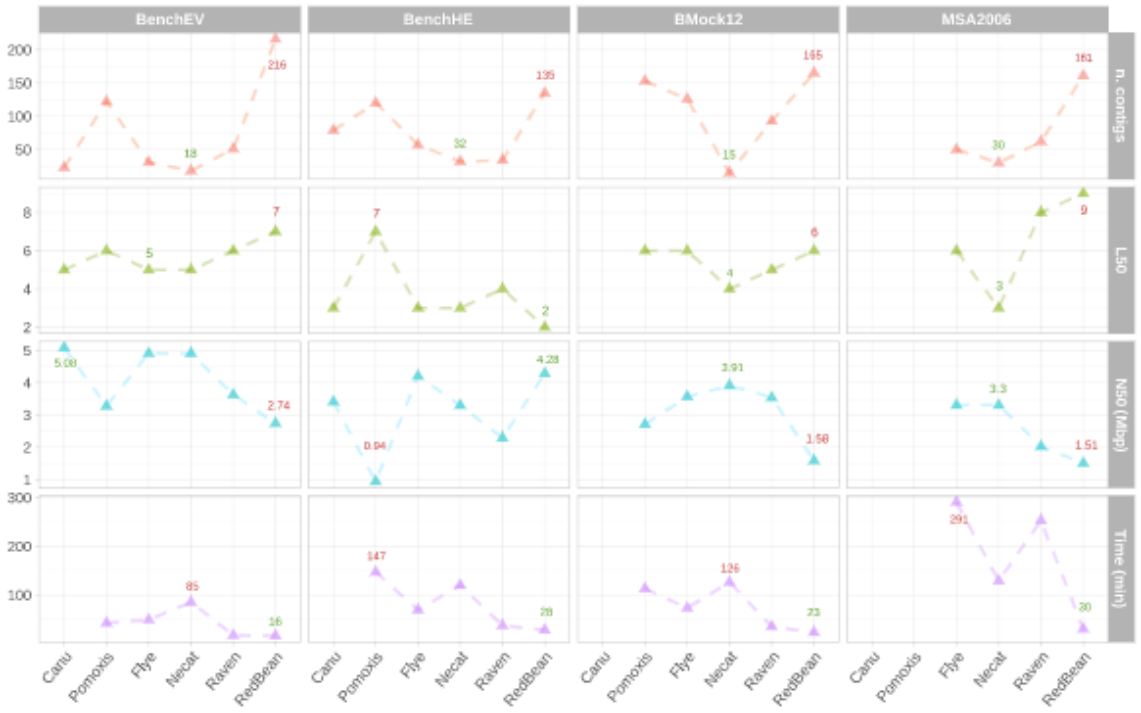


Figure 6. **General performance of assemblers.** Contigs (pink), N50 (blue), L50 (green) and time (purple) are displayed. Running times of Canu were too high and are not included for easing visualization. Best values are highlighted in green and worst values are highlighted in red (data after Polishing). (This Figure can be found in higher size as Supplementary Figure 2).

In terms of contiguity, in the **BenchEV** mock community it was Necat that produced the lowest amount of contigs (17), followed by Canu (23) and Flye (31). Canu obtained the highest N50, followed by Flye and Necat, while Redbean obtained the lowest N50 (Figure 6). L50 values for the entire metagenome ranged from 5 (Canu, Flye and Necat) to 7 (Redbean). All tools were able to recover highly contiguous genomes, only in a few contigs. This scenario changed in the **BenchHE** mock community, since the assemblies were more fragmented, thus contig number increased and N50 decreased. Necat was also the assembler that made the lowest amount of contigs (32), followed this time by Raven (34), then Flye, Canu and Pomoxis, and finally Redbean, that decreased its total contigs (135) since it recovered less metagenome. Redbean had the highest N50 and the lowest L50 (2), followed by Flye and Canu in both parameters and Necat in the L50 (3); while the highest L50 was obtained by Pomoxis (7). For

this case, it should be noted that Redbean N50 metrics were probably artificially high due to its lower metagenome recovery in comparison to other tools (Figure 4). Lower recovery fraction can increase N50 since there is less total metagenome to compare the longest contig to. The **BMock12** metagenome assembly was more fragmented than the BenchHE assembly, as it was overally more recovered but had a complex composition that challenged the assembly. Necat was the assembler that obtained the lowest amount of contigs (15) and the highest N50, followed by Raven and Flye, in both cases. Necat and Raven had the lowest L50 (4 and 5, respectively) and the rest had an L50 of 6. Nevertheless, Necat results could be considered artificially high due to its low metagenome recovery fraction (Figure 4). Lastly, in the **MSA2006** assembly Necat was the assembler with the lowest contig production (30) once again, followed by Flye (50), Raven (62) and Redbean (161). The highest N50 was obtained by Flye and Necat. In agreement with its low recovery ratio in in this community, the lowest L50 was obtained by Necat (3).

To sum up, except Canu (and Pomoxis in the MSA2006 assembly), all assemblers were very efficient computationally, being able to obtain assemblies in less than 3 hours in all cases. In general, the metagenomes here obtained were very continuous, especially when comparing the herein obtained metrics with those usually obtained in Illumina and even hybrid assemblies (Koren and Phillippy, 2015; Frank *et al.*, 2016). Necat was very efficient in contig obtention, while Redbean obtained the highest number. As community complexity increased, two general tendencies were observed depending on the assembler: assemblers with higher microbial screening capacity (Canu, Flye, Raven and Pomoxis) produced more complete but more fragmented metagenomes, with higher number of contigs and L50 as well as lower N50. On the contrary, assemblers with lower microbial screening capacity (Redbean and Necat) obtained less complete but more contiguous metagenomes, with higher N50 and lower L50 and total contig number.

5.2-Accuracy and polishing

The third evaluation of the assemblies was in terms of accuracy, which was measured through the number of Indels and SNPs per 100kbp of metagenome obtained. As discussed previously, long read sequencing technologies have high error rates (McCombie *et al.*, 2019). When dealing with Nanopore data, this issue is addressed using coverage for correction, that is, generating a consensus sequence with the highest possible number of reads. Furthermore, assembly algorithms also influence error rates, and not all assemblers are equally accurate

(Wick and Holt, 2019). Once the draft assemblies are generated, for best result obtention those drafts have to be polished, which allows for Indels and SNPs correction (Chin *et al.,* 2013). The errors are detected through aligning the draft assemblies to the input reads, measuring similarity between the reads and the assembly draft, and creating a more accurate consensus draft accordingly. In this work, the most recently recommended polisher pipeline was followed; that is, one round of Racon followed by one round of Medaka (OXFORD NANOPORE TECHNOLOGIES, 2020a), which improved the general assembly accuracy statistics. Accuracy metrics before and after polishing are summarized in Figure 7.

After the polishing pipeline, **Indels** were reduced in all assemblies (Figure 7). The reduction after the polishing step with Medaka was, in all cases, of at least 100 Indels per 100kbp, and in most cases it was halved. The response of each assembler upon polishing—the shape of the curve seen in Figure 7— was similar among communities, which suggests that the effect of polishing on draft total Indels is constant and is dependent on the assembler. **Mismatches** were, however, not as clearly improved as Indels after polishing (Figure 7), and were in general minimally varied after polishing. In fact, mismatches rates were only clearly improved in Flye and Necat assemblies for all the mock communities.

More in detail, and starting with the **BenchEV** mock community, in the **1st draft** assemblies Pomoxis and Redbean had the highest mismatches values per 100kbp (149 and 148 per 100kbp, respectively), while Canu had the lowest values of mismatches (98 per 100kbp) (Figure 7). In contrast, Flye was the tool that introduced the least Indels (305 per 100kpb), followed by Raven and Pomoxis. **After polishing**, the Flye assembly kept having the lowest Indels (194 per 100kbp), while the lowest mismatches were obtained by Necat (97 per 100kbp). The most Indel-containing assembly was that of Pomoxis and the most mismatches-containing assembly was that of Redbean. Indels reduction after polishing was more acute than mismatches reduction for all assemblers. After the Medaka round all Indels were reduced in at least 100 per 100kbp. Differences between assemblers were residual in this community.

In the **BenchHE** community **1st draft** assemblies, Canu obtained the lowest number of mismatches again (144 per 100 Kbp), while Flye retrieved the highest (309 per Kbp). It is important to highlight that most of the detected mismatches for Flye came from barely recovered genomes (Figure 8). Due to the high species screening capacity of Flye, contigs for the least abundant microorganisms are recovered despite having low coverage. The lower coverage also reduces mismatches correction with respect to the more abundant species, thus increasing error ratios (Figure 8). Raven retrieved the least Indels per 100kbp (420), while

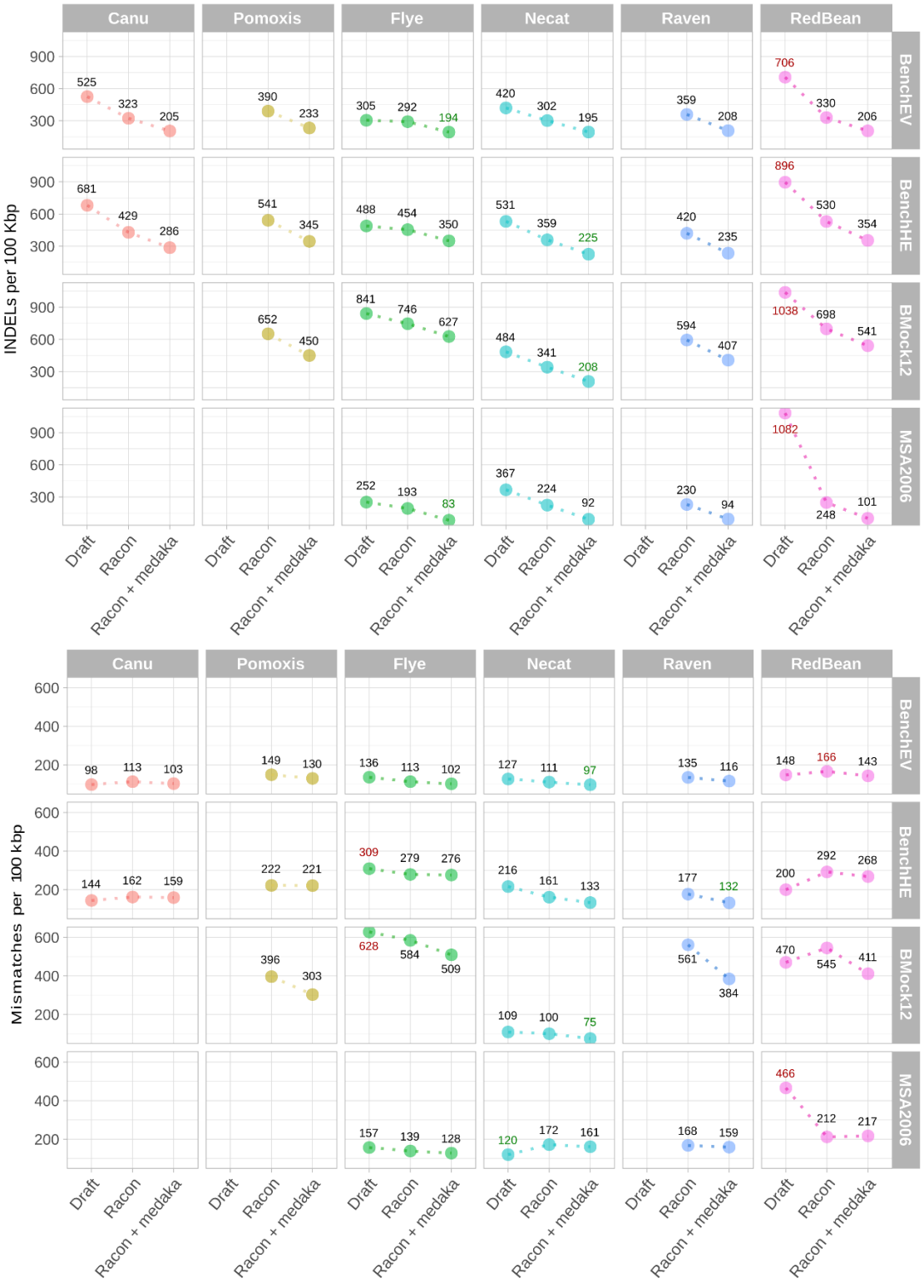


Figure 7. **Accuracy statistics for assemblies after first obtention (draft), Racon polishing and Medaka polishing.** Accuracy statistics include Indels (up) and mismatches or SNPs (down). Raven and Pomoxis have default Racon rounds included, and therefore their draft and Racon data are equal and only one point is included. Best values for each assembly are highlighted in green and worst values are highlighted in red. Indels for each microorganism after polishing can be consulted in Supplementary Figure 3.

Redbean (896) and Canu (681) retrieved the most. **After polishing**, the least Indel- and mismatches-containing assembly was that of Necat, followed by Raven's. Indels were most abundant in the Redbean assembly, while mismatches were higher for Flye.

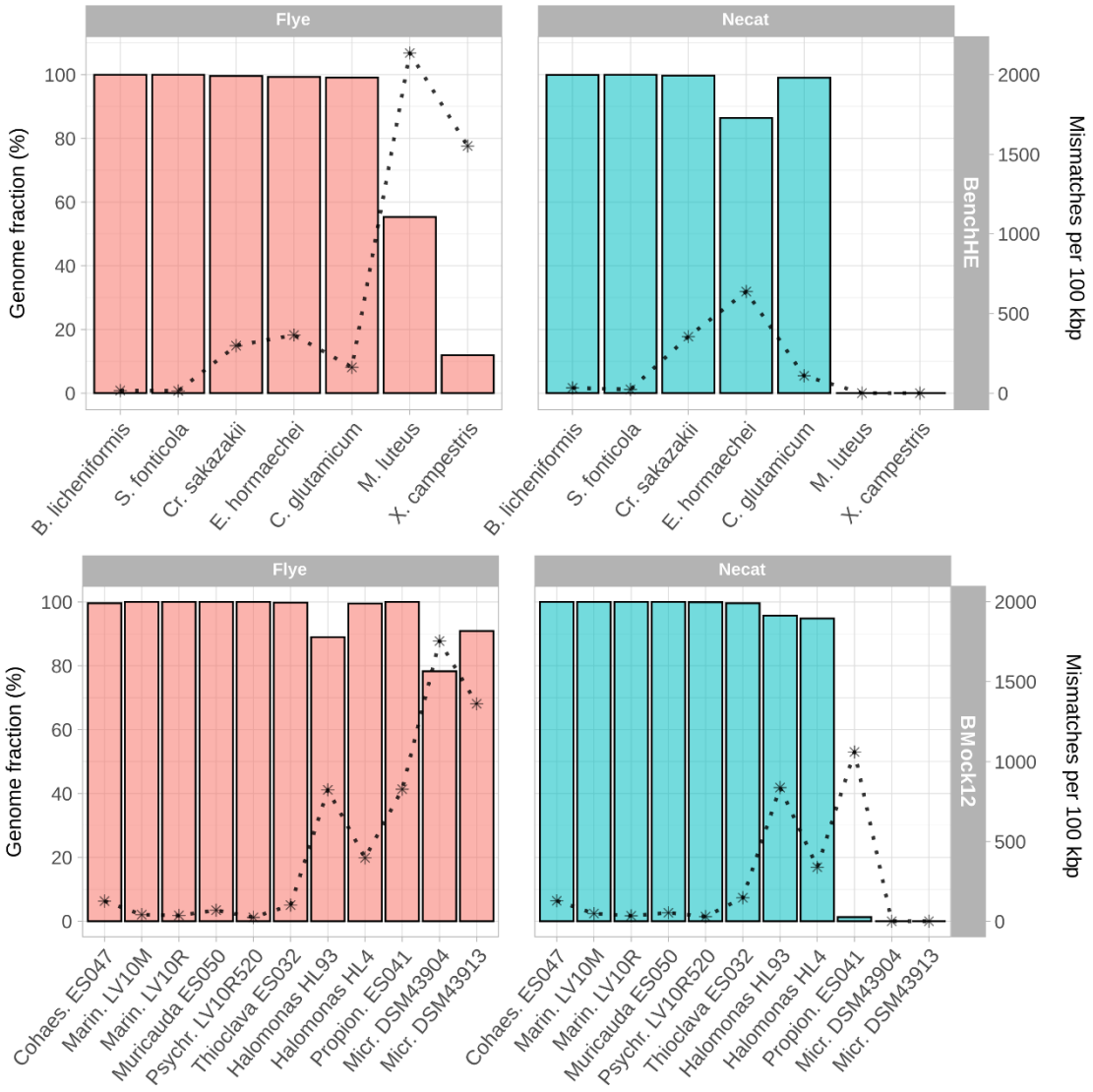


Figure 8. **Mismatches per 100kpb compared to genome recovery for each microorganism in the BenchHE and BMock12 mock communities, after the polishing pipeline.** Only Flye and Necat results are pictured here. Mismatches (dots line) are similar between the two assemblers for the microorganisms recovered in a similar amount. This behavior changes for microorganisms recovered by Flye but not by Necat, as Flye has a bias towards higher mismatches ratios in these microorganisms, evidencing that it also recovers incomplete genomes.

For the **BMock12** mock community **1st draft** assembly, Necat obtained the lowest mismatches (109) and Indels per 100kbp (484). Conversely, Flye obtained the highest number of mismatches (628 per 100kpb), and the second highest number of Indels per 100kbp (841) behind Redbean. **After polishing** with Racon and Medaka, the most accurate assembler for Indels and

mismatches was Necat, while the least was Flye. These results are also biased due to Flye's high recovery ratio. In fact, Flye was in the same range of accuracy as the rest of the assembly tools when considering the recovered genome of each microorganism (Figure 8) and demonstrated to be the most accurate assembler for some completely recovered genomes (Figure 7 and Figure 8).

In the **MSA2006** community **1st drafts**, Redbean obtained both the highest mismatches (466) and Indels (1082) per 100kbp, while Necat obtained the lowest mismatches (119) and Raven obtained the lowest Indels (230). This time, Flye's accuracy was the best, proving that this tool has fine accuracy when enough coverage of microorganisms is achieved. **After polishing** with Racon and Medaka, Flye had both the lowest Indels (82.91) and mismatches (128.28) per 100kpb, while Redbean still had the highest (100.53 and 217.38).

In purely accuracy parameters, both prior and after polishing **Necat** provided best results in the BenchEv, BenchHE and BMock12 communities, and **Flye** obtained best results for the most homogeneous BenchEV and MSA2006 communities. Flye, however, obtained more genomes from each community than Necat, which decreased accuracy in the heterogeneous communities BenchHE and BMock12, as less coverage provided less material for consensus sequence generation. Therefore, **Flye increases diversity screening at the expense of accuracy in less abundant genomes** (Figure 8). Despite including default polishing rounds with Racon, Raven and Pomoxis did not specifically outstand in accuracy. However, Raven did perform well in the MSA2006 and BenchHE assemblies, although not as well as Necat or Flye. Canu did not outperform any other assembler in accuracy after polishing, which further discards its suitability for metagenome assembly.

5.3- BGCs prediction from complete datasets (BenchEV and MSA2006)

After polishing the assembly drafts, an annotation analysis using antiSMASH bacterial version (Blin *et al.*, 2019; Medema *et al.*, 2011) was made for further assembly quality assessment. BGCs –biosynthetic gene clusters, operons involved in secondary metabolites production that are grouped together in the prokaryote genome—prediction analysis was selected as being of most interest for addressing accuracy in functionality terms. BGCs are complex regions under tight regulation, containing repetitive sequences and very sensitive to frameshift mutations (Watson and Warr, 2019; Millet et al., 2017). High contiguity obtained with ONT long reads should theoretically be advantageous for resolving repetitive regions present in BGCs better than short-read technologies, but since Nanopore reads are prone to SNPs and specially Indels

errors (Amarasinghe *et al.*, 2020), the prediction can be truncated thus affecting the functional analysis of the community (Millet et al., 2017). The most accurately recovered mock communities —the BenchEV and MSA2006 mock communities—were selected for conducting this analysis. Both their references and the drafts resulting from the polishing pipeline using Racon and Medaka were analyzed using the online antiSMASH platform, and restrictions were set as relaxed, that allow for well-defined and partial clusters, with all or missing a few functional parts (Figure 9).

The **MSA2006** community reference was predicted to contain 26 BGC regions, and none had a matching known cluster with more than 88% of similarity. Flye, Raven and Redbean predicted 24 BGCs, and Necat predicted 12. This is consistent with the recovery and accuracy results explained in the previous sections. Both Raven and Flye obtained a very similar profile than that of the reference, while Redbean obtained a less accurate profile. Necat gave the worst prediction (Figure 9). The **BenchEV** community reference, on the other hand, was predicted to contain 75 BGC regions in the reference metagenome, from which 8 had been assigned to a known cluster with 100% similarity. Canu predicted 70, with 7 having 100% similarity to known clusters, Flye and Necat predicted 68 and had 6 regions with 100% similarity to known clusters. Redbean and Pomoxis predicted 67, and Raven predicted 66, all having 5 regions with 100% of similarity to known clusters. All profiles were overally similar to that of the reference, being the most complete those of Flye, Pomoxis and Canu (Figure 9).

BGCs prediction gave more complete results in the **BenchEV** community than in the MSA2006, but predicted groups were more accurate in the **MSA2006** community than in the BenchEV community (Figure BCG). Reasons for this, besides assemblers' performance, might be just due to best knowledge of functional groups of the microorganisms included in the BenchEV community, since the reference of the MSA2006 also gave an overall lower number of predicted BGCs. The fact that for most assemblers the predicted BGCs were similar in both quantity and identity to those obtained using the reference suggests that assemblies were overally contiguous and inaccuracies remaining after polishing did not substantially change the functional profile of the community. Flye and Raven obtained the most similar profile to the reference in the MSA2006 community, while Flye, Pomoxis and Canu obtained the most complete profiles in the BenchEV community. Flye was, overall, the most robust among communities.

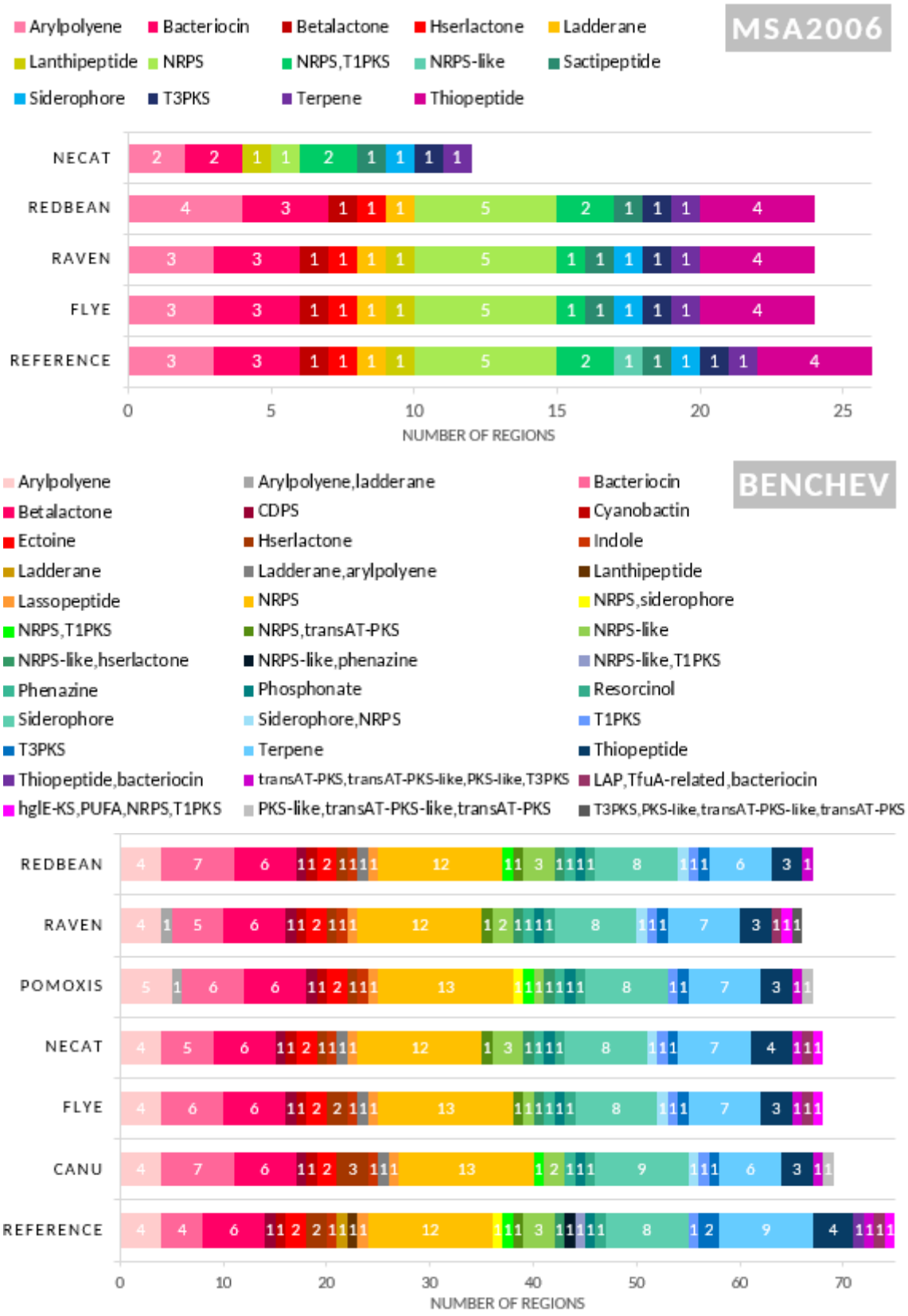


Figure 9. **Predicted BCGs for the reference metagenome and for each polished assembly of the MSA2006 (up) and BenchEV (down) mock communities.** Note that in the BenchEV prediction all asemblers introduced at least one new region (grey shades), and several regions were only predicted for the reference.

6. Concluding remarks.

In the course of this work data from four mock communities has been assembled and polished with the aim of benchmarking the assembly tools most suitable for metagenomics assembly of ONT-obtained reads. A total of six assemblers were tested in communities of different complexities and data sizes, and the results were evaluated in terms of contiguity, accuracy, and computer resources consumption.

Redbean was the fastest assembler, which is its main advantage when comparing it with other assembly tools, but also obtained the lowest genome fractions for all the datasets. **Raven** was also fast and had relatively good results even though it used no genome size estimation parameter, which can be of advantage when dealing with unknown communities. **Pomoxis** in general achieved better genome fractions and accuracy than Redbean and similar to Raven, but was slower, and had memory issues when dealing with the deep-sequenced dataset (MSA2006). It had a general good performance but even when it recovered more genome fraction than Necat it had much lower accuracy. **Necat** was inconsistent, since it had very good contig formation, accuracy and speed, as well as genome recovery fraction for most datasets, but it made an incomplete assembly for MSA2006 and BMock12. Therefore, higher accuracy, higher N50, lower L50 and lower total contig formation seen in Necat were most probably at the expense of recovered genomes and species. **Canu** is known for its high accuracy, but despite its accuracy, the amount of time that Canu needed for performing the assemblies in relatively small datasets –specially for the BenchHE and BMock12 communities—when comparing it to the other assembly tools was disproportionate. Its performance is seriously compromised by community complexity, which is an important drawback for metagenomic analysis, where complex communities are the most common case scenario. High computational requirements and time consumption are too unpractical for Canu to be the general usage assembler. Not only that, results obtained with Canu were similar, but generally not better, than those obtained with Flye, which makes Canu's drawbacks more relevant. Finally, **Flye** attempted to assemble genomes in lower abundance than other assemblers, obtaining the highest microbial diversity, but at the cost of accuracy. This is a behavior that can be both an advantage and a drawback. In certain conditions, Canu or Necat could outperform Flye in terms of contig formation or accuracy. However, the gap in those cases is very small, and Flye outperforms all other assemblers in communities more heterogeneous, which are more similar to real communities. Once accuracy issues are solved—which is fastly evolving with new pore chemistries and basecalling algorithms–, low-accuracy assemblies of least abundant microorganisms will have an increased quality confidence. Furthermore, imperfect assemblies of least abundant

microorganisms can be discarded further in the analysis process, and for that, having at least signs of their presence is of interest to understand the community.

Table 6. **Best and worst of each assembler herein tested.**

Assembler	BEST	WORST
Redbean	Speed	Metagenome recovery
Raven	No need of genome size estimation, overally good	Did not excel in any parameter
Pomoxis	Overally good	Memory issues
Necat	Contigs, accuracy, speed	Inconsistent in heterogeneous communities
Canu	Accuracy and metagenome recovery	Speed
Flye	Microbial screening	Accuracy

The assembly tools herein tested performed heterogeneously on the datasets used. Despite the variability, all the tools retrieved highly contiguous metagenomes. The number of contigs was way smaller and contigs were longer than in traditional Illumina metagenome sequencing, or hybrid sequencing (Koren and Phillippy, 2015; Frank *et al.*, 2016). Total contig formation was dependent on community complexity –being higher for the communities harder to assemble—and assembler –being Necat, Canu, and Flye the assemblers obtaining the least contigs–. Draft assembly polishing and annotation were only briefly treated here, as assembly is upstream on the analysis process and is therefore more urgent to treat; but also because there is less variability in either pipelines and more standardized protocols exist. Polishing was, however, proved to be a very useful and necessary step when analyzing ONT data, as it improved accuracy and Indels, that are the main errors obtained through ONT sequencing.

In overall, the results obtained in this study agree with and validate the results dilucidated by Latorre-Pérez *et al.* (2019), and ease the path for Nanopore metagenomic assembly standardization by supporting the usage of Flye as the by-defect metagenomic assembly tool. In this work, the current panorama of the best assembly tools for Nanopore data is depicted and compared, and through different mock communities' comparison, said assembly tools have been evaluated and discussed. For maximized versatility, **Flye** arises as the most convenient choice. However, it is not to be ignored its drawbacks and flaws, that should be kept present when making any analysis with this tool. Hopefully, benchmarking studies and rational development of metagenomic analysis, as well as validation of pipelines through real communities-analysis will end uncertainty in this field.

7. Conclusions

Summarizing, the overall conclusions of this research project are:

1. Assembly tools performance for metagenomic data is variable and depends on the dataset, hence benchmarking is necessary.

2. In this work, and in accordance to other evaluations, Flye was the most robust assembler: it provided the most complete results among datasets and conditions.

3. Pomoxis and Raven also have a good performance among datasets, in spite of being less powerful for screening. Raven was more efficient and balanced.

4. Canu and Necat are very accurate assemblers and provide very good results in some cases, but their performance, especially for Canu, is easily jeopardized by dataset complexity.

5. The polishing step in ONT data is necessary and very useful for accuracy improvement, especially for Indels correction.

6. In spite of the high error rate associated to date to nanopore sequencing, Oxford Nanopore MinION is a versatile technology that allows for continuous and highly complete genomes obtention from metagenomes.

7.1-Future work

This study opens other questions to conduct further research:

- Comparing assembly and BCG obtention results with the same analyses but using the high accuracy basecaller instead of the older and less accurate versions used by the researchers that sequenced the communities.
- Using only one round of Medaka for polishing, to assess if Racon is necessary for best accuracy results.
- Expanding the datasets to include data from real communities.

8. Bibliography

AMARASINGHE, S. L., SU, S., DONG, X., ZAPPIA, L., RITCHIE, M. E., & GOUIL, Q. (2020). Opportunities and challenges in long-read sequencing data analysis. *Genome biology*, 21(1), 30.

ANAND, G. AND KODALI, R. (2008). Benchmarking the benchmarking models. *Benchmarking: An International Journal*, pp. 257-291.

ANGERS-LOUSTAU, A., PETRILLO, M., BENGTSSON-PALME, J., BERENDONK, T., BLAIS, B., CHAN, K. G., ... & KRUMBIEGEL, C. (2018). The challenges of designing a benchmark strategy for bioinformatics pipelines in the identification of antimicrobial resistance determinants using next generation sequencing technologies. *F1000Research*, 7.

ANIBA, M. R., POCH, O., & THOMPSON, J. D. (2010). Issues in bioinformatics benchmarking: the case study of multiple sequence alignment. *Nucleic Acids Research*, 38(21), 7353-7363.

ARYA, P. (2020). Metagenomics based approach to reveal the secrets of unculturable microbial diversity from aquatic environment. In *Recent Advancements in Microbial Diversity* (pp. 537-559). Elsevier (ed).

ATCC (2019). Gut Microbiome Whole cell Mix (ATCC® MSA2006™) – Product sheet. URL: https://www.lgcstandards-atcc.org/~/ps/MSA-2006.ashx ; last consulted on July 3rd, 2020.

BERTRAND, D., SHAW, J., KALATHIYAPPAN, M., NG, A. H. Q., KUMAR, M. S., LI, C., ... & NG, O. T. (2019). Hybrid metagenomic assembly enables high-resolution analysis of resistance determinants and mobile elements in human microbiomes. *Nature biotechnology*, 37(8), 937-944.

BHARTI, R., & GRIMM, D. G. (2019). Current challenges and best-practice protocols for microbiome analysis. *Briefings in Bioinformatics*, bbz155.

BIKEL, S., VALDEZ-LARA, A., CORNEJO-GRANADOS, F., RICO, K., CANIZALES-QUINTEROS, S., SOBERÓN, X., ... & OCHOA-LEYVA, A. (2015). Combining metagenomics, metatranscriptomics and viromics to explore novel microbial interactions: towards a systems-level understanding of human microbiome. *Computational and structural biotechnology journal*, 13, 390-401.

BLIN, K., SHAW, S., STEINKE, K., VILLEBRO, R., ZIEMERT, N., LEE, S. Y., ... & WEBER, T. (2019). antiSMASH 5.0: updates to the secondary metabolite genome mining pipeline. *Nucleic acids research*, 47(W1), W81-W87.

BOKULICH, N. A., RIDEOUT, J. R., MERCURIO, W. G., SHIFFER, A., WOLFE, B., MAURICE, C. F., ... & CAPORASO, J. G. (2016). mockrobiota: a public resource for microbiome bioinformatics benchmarking. *MSystems*, 1(5), e00062-16.

BREITWIESER, F. P., LU, J., & SALZBERG, S. L. (2019). A review of methods and databases for metagenomic classification and assembly. *Briefings in bioinformatics*, 20(4), 1125-1136.

BURGESS, D. J. (2020). Expanding applications for nanopore sequencing. *Nature Reviews Genetics*, 21(2), 67-67.

BURTON, A. S., STAHL, S. E., JOHN, K. K., JAIN, M., JUUL, S., TURNER, D. J., ... & CASTRO-WALLACE, S. L. (2020). Off Earth Identification of Bacterial Populations Using 16S rDNA Nanopore Sequencing. *Genes*, 11(1), 76.

CARRADEC, Q., POULAIN, J., BOISSIN, E., HUME, B. C., VOOLSTRA, C. R., ZIEGLER, M., ... & WINCKER, P. (2020). A framework for in situ molecular characterization of coral holobionts using nanopore sequencing. *bioRxiv*.

CASSIDY, M. B., LEE, H., & TREVORS, J. T. (1996). Environmental applications of immobilized microbial cells: a review. *Journal of Industrial Microbiology*, 16(2), 79-101.

CAVICCHIOLI, R., RIPPLE, W. J., TIMMIS, K. N., AZAM, F., BAKKEN, L. R., BAYLIS, M., ... & CROWTHER, T. W. (2019). Scientists' warning to humanity: microorganisms and climate change. *Nature Reviews Microbiology*, 17(9), 569-586.

CHAN, W. S., AU, C. H., LEUNG, S. M., HO, D. N., WONG, E. Y. L., TO, M. Y., ... & TANG, B. S. F. (2020). Potential utility of targeted Nanopore sequencing for improving etiologic diagnosis of bacterial and fungal respiratory infection. *Diagnostic Pathology*, 15, 1-7.

CHEN, Y., NIE, F., XIE, S. Q., ZHENG, Y. F., BRAY, T., DAI, Q., ... & HE, L. J. (2020). Fast and accurate assembly of Nanopore reads via progressive error correction and adaptive read selection. *bioRxiv*.

CHIN, C. S., ALEXANDER, D. H., MARKS, P., KLAMMER, A. A., DRAKE, J., HEINER, C., ... & TURNER, S. W. (2013). Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nature methods*, 10(6), 563-569.

CORREA, T., & ABREU, F. (2020). Antarctic microorganisms as sources of biotechnological products. In *Physiological and Biotechnological Aspects of Extremophiles,* (pp269-284). Elsevier (ed).

DEAMER, D., AKESON, M., & BRANTON, D. (2016). Three decades of nanopore sequencing. *Nature biotechnology*, 34(5), 518-524.

DEMAIN, A. L., VANDAMME, E. J., COLLINS, J., & BUCHHOLZ, K. (2017). History of industrial biotechnology. *Industrial biotechnology: microorganisms*, 1, 1-84.

EVANS, G. A. (2000). Designer science and the "omic" revolution. *Nature Biotechnology*, 18(2), 127-127.

FRANK, J. A., PAN, Y., TOOMING-KLUNDERUD, A., EIJSINK, V. G., MCHARDY, A. C., NEDERBRAGT, A. J., & POPE, P. B. (2016). Improved metagenome assemblies and taxonomic binning using long-read circular consensus sequence data. *Scientific reports*, 6(1), 1-10.

GIGUERE, D. J., BAHCHELI, A. T., JORIS, B. R., PAULSSEN, J. M., GIEG, L. M., FLATLEY, M. W., & GLOOR, G. B. (2020). Complete and validated genomes from a metagenome. *bioRxiv*.

GODERIS, A., FISHER, P., GIBSON, A., TANOH, F., WOLSTENCROFT, K., DE ROURE, D., & GOBLE, C. (2009). Benchmarking workflow discovery: a case study from bioinformatics. *Concurrency and Computation: Practice and Experience*, 21(16), 2052-2069.

GUREVICH, A., SAVELIEV, V., VYAHHI, N., & TESLER, G. (2013). QUAST: quality assessment tool for genome assemblies. *Bioinformatics,* 29(8), 1072-1075.

HALL, C. L., ZASCAVAGE, R. R., SEDLAZECK, F. J., & PLANZ, J. V. (2020). Potential applications of nanopore sequencing for forensic analysis. *Forensic science review*, 32(1), 23-54.

HANDELSMAN, J., RONDON, M. R., BRADY, S. F., CLARDY, J., & GOODMAN, R. M. (1998). Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products. *Chemistry & biology*, 5(10), R245-R249.

HANDELSMAN, J. (2004). Metagenomics: application of genomics to uncultured microorganisms. *Microbiology and molecular biology reviews*, 68(4), 669-685.

HASNAIN, M., AFZAL, B., MUHAMMAD TARIQ PERVEZ, T., & HUSSAIN, T. (2020). 2. A review on nanopore sequencing technology, its applications and challenges. *Pure And Applied Biology (PAB)*, 9(1), 154-161.

HAWKINS, D. M. (2004). The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1), 1-12.

HIGHLANDER, S. (2014). Mock community analysis, p 1–7. *Encyclopedia of metagenomics. Springer, New York, NY.*

HUGENHOLTZ, P., & TYSON, G. W. (2008). Metagenomics. *Nature*, 455(7212), 481-483.

JANSSON, J. K., & BAKER, E. S. (2016). A multi-omic future for microbiome studies. *Nature microbiology*, 1(5), 1-3.

KIM, M., LEE, K. H., YOON, S. W., KIM, B. S., CHUN, J., & YI, H. (2013). Analytical tools and databases for metagenomics in the next-generation sequencing era. *Genomics & informatics*, 11(3), 102–113.

KOLMOGOROV, M., YUAN, J., LIN, Y., & PEVZNER, P. A. (2019a). Assembly of long, error-prone reads using repeat graphs. *Nature biotechnology*, 37(5), 540-546.

KOLMOGOROV, M., RAYKO, M., YUAN, J., POLEVIKOV, E., & PEVZNER, P. (2019b). metaFlye: scalable long-read metagenome assembly using repeat graphs. *bioRxiv*, 637637.

KOREN, S., & PHILLIPPY, A. M. (2015). One chromosome, one contig: complete microbial genomes from long-read sequencing and assembly. *Current opinion in microbiology*, 23, 110-120.

KOREN, S., WALENZ, B. P., BERLIN, K., MILLER, J. R., BERGMAN, N. H., & PHILLIPPY, A. M. (2017). Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome research*, 27(5), 722-736.

LANG, D., ZHANG, S., REN, P., LIANG, F., SUN, Z., MENG, G., ... & HAN, L. (2020). Comparison of the two up-to-date sequencing technologies for genome assembly: HiFi reads of Pacbio Sequel II system and ultralong reads of Oxford Nanopore. *bioRxiv*.

LATORRE-PÉREZ, A., VILLALBA-BERMELL, P., PASCUAL, J., PORCAR, M., & VILANOVA, C. (2019). Assembly methods for nanopore-based metagenomic sequencing: a comparative study. *bioRxiv*, 722405.

LEIDENFROST, R. M., PÖTHER, D. C., JÄCKEL, U., & WÜNSCHIERS, R. (2020). Benchmarking the Minion: evaluating long reads for microbial profiling. *Scientific reports*, 10(1), 1-10.

LORENZ, P., & ECK, J. (2005). Metagenomics and industrial applications. *Nature Reviews Microbiology*, 3(6), 510-516.

MARCHESI, J. R., & RAVEL, J. (2015). The vocabulary of microbiome research: a proposal. Microbiome 3, 31.

MCCOMBIE, W. R., MCPHERSON, J. D., & MARDIS, E. R. (2019). Next-generation sequencing technologies. *Cold Spring Harbor Perspectives in Medicine*, 9(11), a036798.

MEDEMA, M. H., BLIN, K., CIMERMANCIC, P., DE JAGER, V., ZAKRZEWSKI, P., FISCHBACH, M. A., ... & BREITLING, R. (2011). antiSMASH: rapid identification, annotation and analysis of secondary metabolite biosynthesis gene clusters in bacterial and fungal genome sequences. *Nucleic acids research*, 39(suppl_2), W339-W346.

MEDEMA, M. H., KOTTMANN, R., YILMAZ, P., CUMMINGS, M., BIGGINS, J. B., BLIN, K., DE BRUIJN, I., CHOOI, Y. H., CLAESEN, J., COATES, R. C., CRUZ-MORALES, P., DUDDELA, S., DÜSTERHUS, S., EDWARDS, D. J., FEWER, D. P., GARG, N., GEIGER, C., GOMEZ-ESCRIBANO, J. P., GREULE, A., HADJITHOMAS, M., … GLÖCKNER, F. O. (2015). Minimum Information about a Biosynthetic Gene cluster. *Nature chemical biology*, 11(9), 625–631.

METZKER, M. L. (2010). Sequencing technologies—the next generation. *Nature reviews genetics*, 11(1), 31-46.

MIKHEENKO, A., SAVELIEV, V., & GUREVICH, A. (2016). MetaQUAST: evaluation of metagenome assemblies. *Bioinformatics*, 32(7), 1088-1090.

MILLER, I. J., CHEVRETTE, M. G., & KWAN, J. C. (2017). Interpreting Microbial Biosynthesis in the Genomic Age: Biological and Practical Considerations. *Marine drugs*, 15(6), 165.

MORGAN XC, HUTTENHOWER C. (2012). Chapter 12: Human microbiome analysis. *PLoS Comput Biol.*;8(12):e1002808.

MOSS, E.L., MAGHINI, D.G. & BHATT, A.S. (2020). Complete, closed bacterial genomes from microbiomes using nanopore sequencing. *Nat Biotechnol* 38, 701–707.

NICHOLLS, S. M., QUICK, J. C., TANG, S., & LOMAN, N. J. (2019). Ultra-deep, long-read nanopore sequencing of mock microbial community standards. *Gigascience*, 8(5), giz043.,

OXFORD NANOPORE TECHNOLOGIES. (2018). Pomoxis - bioinformatics tools for nanopore research. *Available at GitHub*: https://github.com/nanoporetech/pomoxis

OXFORD NANOPORE TECHNOLOGIES. (2020a). Assembling microbial genomes using long nanopore sequencing reads. Published online in *Nanopore Resource Center*, URL: https://nanoporetech.com/resource-centre/microbial-genome-assembly-workflow?utm_content=130359910&utm_medium=social&utm_source=twitter&hss_channel=tw-37732219; last consulted on June 13th, 2020.

OXFORD NANOPORE TECHNOLOGIES. (2020b). New research algorithms yield accuracy gains for nanopore sequencing. Published online in *Oxford Nanopore Technologies News*, URL: https://nanoporetech.com/about-us/news/new-research-algorithms-yield-accuracy-gains-nanopore-sequencing; last consulted on June 15th, 2020.

OXFORD NANOPORE TECHNOLOGIES. (2020c). R10.3: the newest nanopore for high accuracy nanopore sequencing –now available in store. Published online in *Oxford Nanopore Technologies News*;

URL: https://nanoporetech.com/about-us/news/r103-newest-nanopore-high-accuracy-nanopore-sequencing-now-available-store ; last consulted on June 15th, 2020.

RATZKE, C., & GORE, J. (2018). Modifying and reacting to the environmental pH can drive bacterial interactions. *PLoS biology*, 16(3), e2004248.

PRISCU, J. C., & CHRISTNER, B. C. (2003). Earth's icy biosphere. *Microbial diversity and bioprospecting*, 130-145.

RUAN, J., & LI, H. (2020). Fast and accurate long-read assembly with wtdbg2. *Nature methods*, *17*(2), 155-158.

SCHLOSS, P.D., HANDELSMAN, J. (2005). Metagenomics for studying unculturable microorganisms: cutting the Gordian knot. *Genome Biol* 6, 229.

SEVIM, V., LEE, J., EGAN, R., CLUM, A., HUNDLEY, H., LEE, J., ... & GÖKER, M. (2019). Shotgun metagenome data of a defined mock community using Oxford Nanopore, PacBio and Illumina technologies. *Scientific data*, 6(1), 1-9.

SINGER, E., ANDREOPOULOS, B., BOWERS, R. M., LEE, J., DESHPANDE, S., CHINIQUY, J., ... & CLUM, A. (2016). Next generation sequencing data of a defined microbial mock community. *Scientific data*, 3(1), 1-8.

SUN, Y., CAI, Y., HUSE, S. M., KNIGHT, R., FARMERIE, W. G., WANG, X., & MAI, V. (2012). A large-scale benchmark study of existing algorithms for taxonomy-independent microbial community analysis. *Briefings in bioinformatics*, 13(1), 107-121.

TEELING, H., & GLÖCKNER, F. O. (2012). Current opportunities and challenges in microbial metagenome analysis—a bioinformatic perspective. *Briefings in bioinformatics*, 13(6), 728-742.

TRINGE, S. G., VON MERING, C., KOBAYASHI, A., SALAMOV, A. A., CHEN, K., CHANG, H. W., ... & BORK, P. (2005). Comparative metagenomics of microbial communities. *Science*, 308(5721), 554-557.

VASER, R., & ŠIKIĆ, M. (2019). Yet another de novo genome assembler. In 2019 *11th International Symposium on Image and Signal Processing and Analysis (ISPA)* (pp. 147-151). IEEE.

VASER, R., SOVIĆ, I., NAGARAJAN, N., & ŠIKIĆ, M. (2017). Fast and accurate de novo genome assembly from long uncorrected reads. *Genome research*, 27(5), 737–746.

VENTER, J. C., ADAMS, M. D., MYERS, E. W., LI, P. W., MURAL, R. J., SUTTON, G. G., ... & GOCAYNE, J. D. (2001). The sequence of the human genome. *Science*, 291(5507), 1304-1351.

WATSON, M., WARR, A. (2019). Errors in long-read assemblies can critically affect protein prediction. *Nat Biotechnol* 37, 124–126.

WICK, R. R., & HOLT, K. E. (2019). Benchmarking of long-read assemblers for prokaryote whole genome sequencing. *F1000Research*, 8. doi: 10.12688/f1000research.21782.2

ZAVODNA, M., BAGSHAW, A., BRAUNING, R., & GEMMELL, N. J. (2014). The accuracy, feasibility and challenges of sequencing short tandem repeats using next-generation sequencing platforms. *PloS one*, 9(12), e113862.

9. Supplementary data

S.1-Supplementary codes:

Supplementary code 1. **Bash script for assemblies**. Each assembler command line is inserted into a time command for measuring the total assembly time. $filepath is the variable including the absolute path of the input reads. $resultdir is the variable including the absolute path of the results directory (folder) and $size is the variable including the estimate genome size of the community.

```
#!/bin/bash

#Ask for fastq file path

echo Please enter absolute path of file:

read filepath

echo Now please enter absolute path of results directory:

read resultdir

cd $resultdir

#All results are in $resultdir

#_____________________________________________________________________________

echo ~~~ Calling assemblers ~~~

#Create txt for time recording

touch assembly_tpo.txt

#NECAT_____________________________________________________________________________

echo ~~~ Starting NECAT ~~~

#NECAT in time file

echo Time NECAT: >>assembly_tpo.txt

cd NECAT/

#CORRECTION + ASSEMBLY + BRIDGING

{ time Necat.pl bridge config.txt 2>> ../assembly.stderr ; } 2>> ../assembly_tpo.txt

#Separation in time file

echo    >> ../assembly_tpo.txt

echo ~~~ NECAT finished ~~~
```

```
cd $resultdir

#RedBean (wdtgb2)_____________________________________________________________

echo ~~~ Running RedBean ~~~

#Redbean in time txt

echo Time Redbean: >>assembly_tpo.txt

echo    >>assembly_tpo.txt

#Folder for RedBean results

mkdir Redbean

cd Redbean/

echo ~~~ Step1. wdtgb2 starting ~~~

echo Step1: >>../assembly_tpo.txt

#Assembler

{ time /home/darwin/Descargas/Programas/Redbean/wtdbg2/wtdbg2 -x ont -t 16 -g 47m -i
$filepath -fo step1 2>>../assembly.stderr ; } 2>> ../assembly_tpo.txt

#Space

echo   >>../assembly_tpo.txt

echo ~~~ Step2. wtpoa-cns starting ~~~

echo Step 2: >>../assembly_tpo.txt

#Consenser

{ time /home/darwin/Descargas/Programas/Redbean/wtdbg2/wtpoa-cns -t 16 -i
step1.ctg.lay.gz -fo assembly.ctg.fa 2>>../assembly.stderr ; } 2>>
../assembly_tpo.txt

#Space

echo   >>../assembly_tpo.txt

echo ~~~ RedBean finished ~~~

#Return to directory of results

cd $resultdir

#Raven_________________________________________________________________________

echo ~~~ Running Raven ~~~
```

```
echo Raven: >>assembly_tpo.txt

mkdir Raven

cd Raven/

#Running Raven and storing the time

{ time raven –threads 16 $filepath > assembly.fa 2>>../assembly.stderr ; }
2>>../assembly_tpo.txt

echo   >>../assembly_tpo.txt

echo ~~~ Raven finished ~~~

cd $resultdir

#Flye_____________________________________________________________________________

echo ~~~ Running metaFlye ~~~

#Flye in txt

echo Time Flye: >>assembly_tpo.txt

#Running Flye and measuring the time

{ time flye –nano-raw $filepath –out-dir $resultdir/Flye –genome-size 47m –threads 16
–meta –plasmids 2>>assembly.stderr ; } 2>>assembly_tpo.txt

#space on file

echo    >>assembly_tpo.txt

echo ~~~ metaFlye finished ~~~

#Pomoxis__________________________________________________________________________

#The environment for Pomoxis must be previously initialized, the command can be found
in activate.txt, among Pomoxis program files

echo ~~~ Running Pomoxis ~~~

echo Time Pomoxis: >>assembly_tpo.txt

#Running Pomoxis and measuring time

{ time /home/darwin/Descargas/Programas/pomoxis/scripts/mini_assemble -i $filepath -o
$resultdir/Pomoxis -p assembly -l 47mb -t 16 2>>assembly.stderr ; }
2>>assembly_tpo.txt

echo    >>assembly_tpo.txt

echo ~~~ Pomoxis finished ~~~
```

```
#Canu_____________________________________________________________________________

echo ~~~ Running Canu 2.0 ~~~

echo Canu: >>assembly_tpo.txt

#Running Canu and measuring time

{ time /home/darwin/Descargas/Programas/canu-2.0/Linux-amd64/bin/canu -p assembly -d
$resultdir/Canu genomeSize=47m corOutCoverage=10000 corMhapSensitivity=high
corMinCoverage=0 redMemory=32 oeaMemory=32 batThreads=16 batMemory=60 -nanopore
$filepath 2>>assembly.stderr ; } 2>>assembly_tpo.txt

#batThreads was set to 16 and the recommended batMemory=200 was changed to 60 due to
canu failure and warnings of CPU resources when 1st running it

echo ~~~ Canu finished ~~~

echo ~~~ All assemblies finished. You can now check your results ~~~
```

Supplementary code 2. **Bash script for Racon polishing**. $reads is the variable including the path to the fastq file containing the sequenced reads. The file containing each assembly draft is assembly.fasta, and assembly_racon.fasta is the file containing the new polished draft.

```
#!/bin/bash

#This script is for running minimap2 + racon. Run the script on the target folder
Correction

#Raven and Pomoxis already have run racon as part of their pipeline (2 and 4 times
respectively)

# First, we create the alignment using minimap2, then we use that alignment for
running Racon once, then we erase sam files

echo Please enter fastq reads file absolute path

read reads

echo Starting correction with Racon

echo Redbean

mkdir Redbean

cd Redbean

echo Indexing draft assembly

minimap2 -x map-ont -d indexed_draft.mmi assembly.fasta

echo Aligning

minimap2 -ax map-ont assembly.fasta $reads > aln.sam

echo Polishing with racon

racon -t16 $reads aln.sam assembly.fasta > assembly_racon.fasta

rm aln.sam

cd ../

echo NECAT

mkdir NECAT

cd NECAT

echo Indexing

minimap2 -x map-ont -d indexed_draft.mmi assembly.fasta

echo Aligning
```

```
minimap2 -ax map-ont assembly.fasta $reads > aln.sam

echo Polishing

racon -t16 $reads aln.sam assembly.fasta > assembly_racon.fasta

rm aln.sam

cd ../

echo Flye

mkdir Flye

cd Flye

echo Indexing

minimap2 -x map-ont -d indexed_draft.mmi assembly.fasta

echo Aligning

minimap2 -ax map-ont assembly.fasta $reads > aln.sam

echo Polishing

racon -t16 $reads aln.sam assembly.fasta > assembly_racon.fasta

rm aln.sam

cd ../

echo Canu

mkdir Canu

cd Canu

echo Indexing

minimap2 -x map-ont -d indexed_draft.mmi assembly.fasta

echo Aligning

minimap2 -ax map-ont assembly.fasta $reads > aln.sam

echo Polishing

racon

rm aln.sam

cd ../

echo Racon polishing finished
```

Supplementary code 3. **Bash script for running medaka after Racon polishing.** Valid for running Medaka without Racon if the input files are substituted by the draft assemblies obtained after first assembly.

```
#!/bin/bash

#Run from Toshiba/Morgane_TFG cd Benchmarking_SRA/Assembly_even

#Bench data: for Albacore v2.0.2, recommended model is r941_trans, but it is not
supported in new versions of medaka

#The data will be run in Mekaka v 0.11.5 with its default model.

echo Starting correction with Medaka

echo Bench data polishing

echo Starting with BenchEv dataset

echo Redbean

medaka_consensus -i equimolar_all.fastq -d ./Correction/Redbean/assembly_racon.fasta
-t 16 -o ./Correction/Redbean

echo Raven

medaka_consensus -i equimolar_all.fastq -d ./Assembly/Raven/assembly.fa -t 16 -o
./Correction/Raven

echo Pomoxis

medaka_consensus -i equimolar_all.fastq -d ./Assembly/Pomox_2/assembly_final.fa -t 16
-o ./Correction/Pomoxis

echo NECAT

medaka_consensus -i equimolar_all.fastq -d ./Correction/NECAT/assembly_racon.fasta -t
16 -o ./Correction/NECAT

echo Flye

medaka_consensus -i equimolar_all.fastq -d ./Correction/Flye/assembly_racon.fasta -t
16 -o ./Correction/Flye

echo Canu

medaka_consensus -i equimolar_all.fastq -d ./Correction/Canu/assembly_racon.fasta -t
16 -o ./Correction/Canu

cd ../Assembly_uneven

echo Starting with BenchHE dataset

echo Redbean
```

```
medaka_consensus -i heterogeneous_all.fastq -d
./Correction/Redbean/assembly_racon.fasta -t 16 -o ./Correction/Redbean

echo Raven

medaka_consensus -i heterogeneous_all.fastq -d ./Assembly/Raven/assembly.fa -t 16 -o
./Correction/Raven

echo Pomoxis

medaka_consensus -i heterogeneous_all.fastq -d ./Assembly/Pomox_2/assembly_final.fa -
t 16 -o ./Correction/Pomoxis

echo NECAT

medaka_consensus -i heterogeneous_all.fastq -d
./Correction/NECAT/assembly_racon.fasta -t 16 -o ./Correction/NECAT

echo Flye

medaka_consensus -i heterogeneous_all.fastq -d ./Correction/Flye/assembly_racon.fasta
-t 16 -o ./Correction/Flye

echo Canu

medaka_consensus -i heterogeneous_all.fastq -d ./Correction/Canu/assembly_racon.fasta
-t 16 -o ./Correction/Canu

cd ../../BMock12_SRA

#BMock12 dataset, obtained with Albacore v2.3.1, recommended model r941_trans

echo Starting with BMock12 dataset

echo Redbean

medaka_consensus -i BMock_por.fastq -d ./Correction/Redbean/assembly_racon.fasta -t
16 -o ./Correction/Redbean

echo Raven

medaka_consensus -i BMock_por.fastq -d ./Assembly/Raven/assembly.fa -t 16 -o
./Correction/Raven

echo Pomoxis

medaka_consensus -i BMock_por.fastq -d ./Assembly/Pomoxis/assembly_final.fa -t 16 -o
./Correction/Pomoxis

echo NECAT

medaka_consensus -i BMock_por.fastq -d ./Correction/NECAT/assembly_racon.fasta -t 16
-o ./Correction/NECAT

echo Flye
```

```
medaka_consensus -i BMock_por.fastq -d ./Correction/Flye/assembly_racon.fasta -t 16 -
o ./Correction/Flye

cd ../atcc_subsample

#MSA2006 dataset, obtained with Guppy v2.3.5

echo Starting with MSA2006 dataset

echo Redbean

medaka_consensus -i first_half_pore.fastq -d
./Correction/Redbean/assembly_racon.fasta -t 16 -o ./Correction/Redbean

echo Raven

medaka_consensus -i first_half_pore.fastq -d ./Assembly/Raven/assembly.fa -t 16 -o
./Correction/Raven

echo NECAT

medaka_consensus -i first_half_pore.fastq -d ./Correction/NECAT/assembly_racon.fasta
-t 16 -o ./Correction/NECAT

echo Flye

medaka_consensus -i first_half_pore.fastq -d ./Correction/Flye/assembly_racon.fasta -
t 16 -o ./Correction/Flye

echo Polishing with Medaka finished.
```

S.2-Assembly complications and solutions

These events helped tunning assembly parameters and input data.

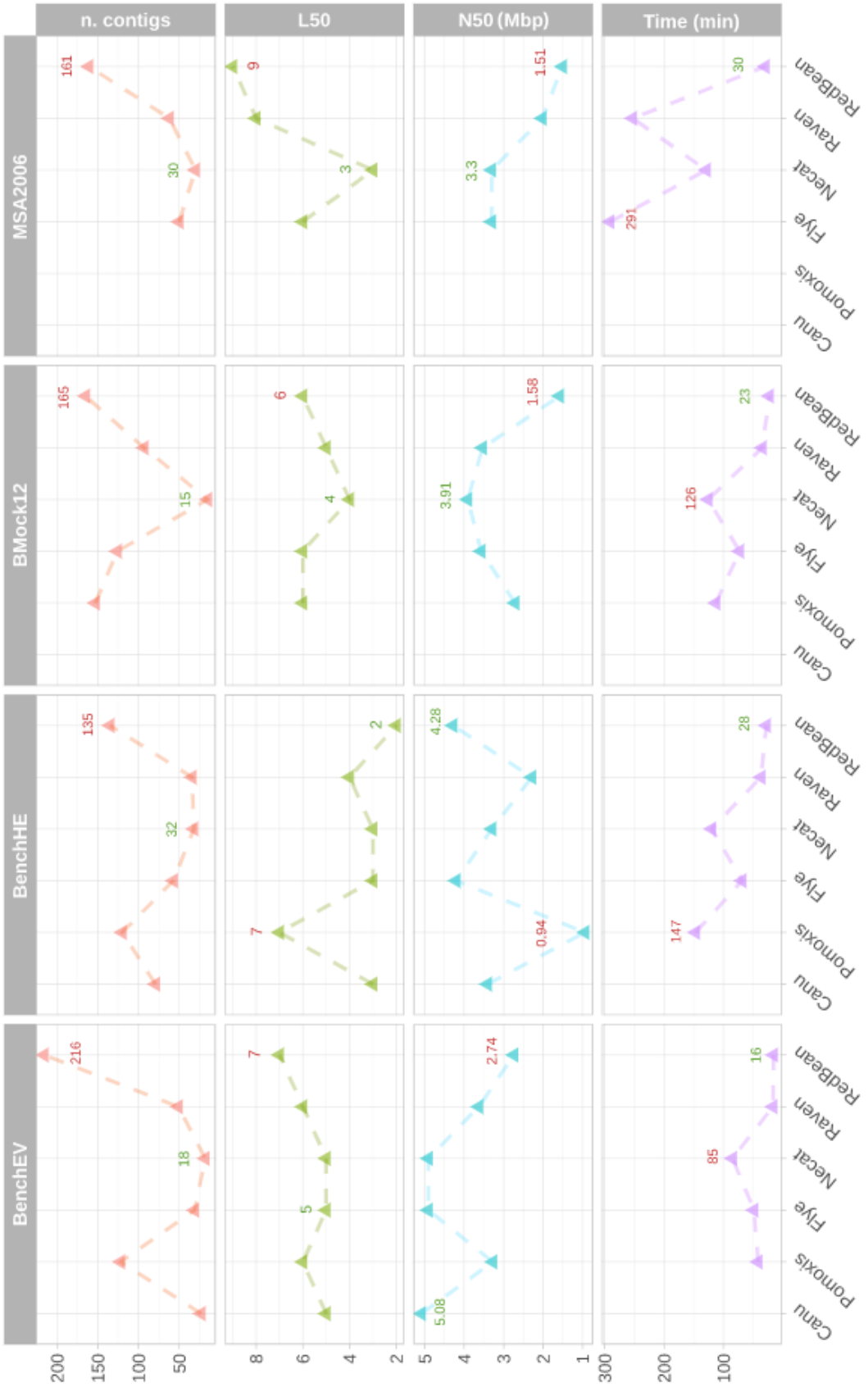
When trying to run Canu for the first sample, an error message was displayed with respect to the machine configuration memory. It advised to change batMemory and batThreads parameters: "16 CPUs and 63 GB detected, cannot run task, change batMemory and/or batThreads. The values were therefore changed to 60 and 16, respectively.

Furthermore, the MSA2006 dataset had to be subsampled because it was too large (60.2Gb) to be porechoped or assembled in the Darwin computer. Flye and Pomoxis and Porechop failed, while Necat gave a very low assembled fraction. For this reason, the datased was split in two and the assembly was run again, using only the first half of the reads.

S.3-Supplementary figures

Title
Benchmarking the MinION : Evaluating long reads for microbial profiling
Shotgun metagenome data of a defined mock community using Oxford Nanopore, PacBio and Illumina technologies
Complete, closed bacterial genomes from microbiomes using nanopore sequencing
Long-read based de novo assembly of low-complexity metagenome samples results in finished genomes and reveals insights into strain
Synthetic microbe communities provide internal reference standards for metagenome sequencing and analysis
De novo Nanopore read quality improvement using deep learning
Implications of error-prone long-read whole-genome shotgun sequencing on characterizing reference microbiomes
Discovering and exploiting multiple types of DNA methylation from individual bacteria and microbiome using nanopore sequencing
Preprint: Analysis procedures for assessing recovery of high quality, complete, closed genomes from Nanopore long read metagenome
Pathogen Detection and Microbiome Analysis of Infected Wheat Using a Portable DNA Sequencer
Metagenomic Profiling of Microbial Pathogens in the Little Bighorn River, Montana
Near-complete Lokiarchaeota genomes from complex environmental samples using long and short read metagenomic analyses
Novel prosthecate bacteria from the candidate phylum Acetothermia
Generating closed bacterial genomes from long-read nanopore sequencing of microbiomes
New tools for diet analysis: nanopore sequencing of metagenomic DNA from rat stomach contents to quantify diet
Deciphering taxonomic and functional diversity of fungi as potential bioindicators within confluence stretch of Ganges and Yamuna Rivers,
Improving recovery of member genomes from enrichment reactor microbial communities using MinION–based long read metagenomics
Rapid MinION profiling of preterm microbiota and antimicrobial-resistant pathogens
Genetic repertoires of anaerobic microbiomes driving generation of biogas
Long-read based de novo assembly of low-complexity metagenome samples results in finished genomes and reveals insights into strain
Stationary and portable sequencing-based approaches for tracing wastewater contamination in urban stormwater systems
Ultra-deep, long-read nanopore sequencing of mock microbial community standards

Supplementary figure 1. **List of studies containing ONT metagenomic data considered for this project.** Studies were retrieved during the first two weeks of April 2020. Green = selected studies. Red = discarded after first filtration. Yellow = discarded after second filtration. Grey = Not evaluable (data employed by Latorre-Pérez *et al.* (2019)).



Supplementary figure 2. **General performance of assemblers (higher size).**

Microorganisms

	BenchEV						BenchHE					
	Canu	Flye	Necat	Pomoxis	Raven	Redbean	Canu	Flye	Necat	Pomoxis	Raven	Redbean
C. glutamicum	202.08	188.29	193.50	192.38	203.14	201.90	179.41	148.69	155.77	233.13	158.24	501.80
B. licheniformis	357.43	341.88	344.34	351.86	346.47	346.69	263.63	249.63	254.50	431.01	254.66	252.36
X. campestris	84.14	71.74	72.57	81.65	72.62	78.96	1,070.57	1,555.09	-	1,263.61	-	1,195.75
E. hormaechei	262.75	250.53	248.83	365.97	328.64	303.40	339.82	319.80	365.48	421.13	368.12	368.02
S. fonticola	158.70	155.45	156.92	210.50	160.90	172.84	157.55	154.01	156.89	177.28	166.42	171.53
A. xylosoxidans	106.29	91.24	92.30	98.42	96.69	99.71	1,343.65	-	-	-	-	1,353.09
M. luteus	205.70	186.38	198.40	214.10	220.09	203.73	1,396.93	1,793.14	-	1,272.67	843.27	1,437.65
Cr. sakazakii	297.34	291.56	286.36	461.85	339.79	337.85	236.78	205.64	204.57	449.77	230.17	260.19
S. saprophyticus	249.59	239.61	240.84	245.81	243.21	242.04	-	-	-	-	-	-
Ch. violaceum	97.84	91.54	92.18	104.75	97.19	96.29	0.00	-	-	-	-	-
P. odorifer	316.46	313.22	314.64	323.56	318.55	316.12	-	-	-	-	-	-
D. solani	147.00	146.85	145.70	185.74	155.43	163.31	-	-	-	-	-	-

	BMock12				
	Flye	Necat	Pomoxis	Raven	Redbean
Cohaes. ES047	182.89	183.65	189.98	188.99	186.36
Halomonas HL4	205.26	197.20	399.01	228.29	243.59
Halomonas HL93	248.50	228.04	383.95	245.84	241.91
Marin. LV10M	203.71	206.36	223.19	216.64	215.84
Marin. LV10R	162.99	163.59	173.84	173.20	171.63
Micr. DSM43904	1,543.59	-	1,151.22	1,289.75	962.53
Micr. DSM43913	1,546.92	-	1,162.35	1,280.16	1,182.66
Micr. DSM45161	-	-	-	-	-
Muric. ES050	310.71	311.04	333.82	314.38	319.37
Propion. ES041	860.16	712.89	924.70	909.67	935.06
Psychr. LV10R	222.83	222.81	279.50	226.65	228.92
Thioclava ES032	185.64	190.57	199.33	195.31	196.22

	MSA2006			
	Flye	Necat	Raven	Redbean
Enterobacter	86.69	335.09	112.29	107.61
Bacter. 9343	30.21	33.49	29.86	35.83
Bifidobacterium	128.41	-	145.81	149.55
Clostridioides	94.85	94.56	79.37	97.83
E. coli K12	110.06	92.59	164.03	172.38
Bacter. 8482	35.69	-	32.90	42.17
Salmonella	111.44	222.83	166.34	157.75
Fusobacterium	105.45	-	98.25	121.83
Helicobacter	364.30	386.17	360.65	378.02
Lactobacillus	20.46	21.07	20.86	22.11
Enterococcus	50.91	52.92	50.61	51.98
Yersinia	60.37	170.26	67.01	70.14

Plasmids

	BenchEV						BenchHE					
	Canu	Flye	Necat	Pomoxis	Raven	Redbean	Canu	Flye	Necat	Pomoxis	Raven	Redbean
Cr. sakaz. CSK1	607.00	285.41	275.81	304.57	285.68	289.05	298.17	175.72	176.77	185.82	190.70	196.70
Cr. sakaz. CSK2	1,964.36	587.64	-	429.80	-	-	567.03	202.55	-	-	-	-
Cr. sakaz. CSK3	853.02	359.17	370.40	360.34	366.80	381.62	851.15	267.51	252.33	249.52	242.39	248.80
S. sapr. pSSP1	629.32	213.24	221.04	232.23	260.72	332.82	-	-	-	-	-	-
S. sapr. pSSP2	2,789.68	315.02	385.07	1,351.11	381.51	486.77	-	-	-	-	-	-

	MSA2006			
	Flye	Necat	Raven	Redbean
Enter. pECLA	76.58	101.82	87.70	130.24
Enter. pECLB	60.25	73.89	-	92.84
Enteroc. pTEF1	70.87	63.33	241.90	501.20
Enteroc. pTEF2	57.24	60.70	292.92	622.85
Enteroc. pTEF3	183.71	27.84	180.03	-
Bacter. pBF9343	60.18	51.97	46.60	44.58

Supplementary figure 3. **Indels per assembler and microorganism, after Polishing**. Squares are filled proportionally to the number of total Indels.