Essentials in Metagenomics (Part II)

Authors: Angela Peña-González[1] and Alejandro Reyes Muñoz[2]

Affiliations:

1-Angela Peña-Gonzalez, PhD, MS
Postdoctoral Researcher
Group in Computational Biology and Microbial Ecology
Max Planck Tandem Group in Computational Biology
Department of Biological Sciences
Universidad de los Andes
Bogotá, Colombia.
https://orcid.org/0000-0003-0684-8179

2-Alejandro Reyes Muñoz, PhD (corresponding author)
Associate Professor
Director BCEM (bcem.uniandes.edu.co)
Group Leader Max Planck Tandem Group in Computational Biology
Department of Biological Sciences
Universidad de los Andes
Bogotá, Colombia
https://orcid.org/0000-0003-2907-3265
Max Planck Tandem Group in Computational Biology, Department of Biological Sciences, Universidad de los Andes, Cra. 1 #18a 12, Bogotá, Colombia, 111711
Center for Genome Sciences and Systems Biology, Department of Pathology and Immunology, Washington University School of Medicine, St Louis, MO, USA, 63110

Teaching goals and learning outcomes

This course is aimed at undergraduate students, graduate students and professionals at any career stage performing research in the fields of microbiology and bioinformatics. Also, molecular biologists and microbiologists (life scientists) interested in learning basics on metagenomics data analysis and bioinformatics will benefit from this course. An undergraduate knowledge of a subject related to the life sciences (molecular biology) would be an advantage.

By the end of this CABANA e-Learning tutorial, the user will be able to:

- Describe relevant bioinformatic analysis commonly included in a protocol for metagenomic data processing
- Recognize common tools and databases used in a typical read-based and assembly-based data analysis pipelines
- Be aware of the advantages and limitations of tools and methods frequently found in a computational experimental design

About this guide

This is a guide that you can read through at your own speed. The sections "***Material to study***" are additional material to expand your knowledge. Additionally, the sections "***Stop and Think***" include questions or situations for you to consider in more depth about the knowledge you are acquiring. For general definitions in the field of biological sciences, the user can query the following link: *https://www.ebi.ac.uk/training/online/glossary*.

Key terms

Metagenomic data analysis, bioinformatics, quality control, assembly, gene prediction and annotation, recovery of Metagenome-Assembled Genomes (MAGs), metabarcoding

1- Introduction to metagenomics data analysis

In metagenomic studies, once the sequences are obtained, the data is processed in a set of steps that can be performed with various free or licensed tools. The tools can be run from the command line or on platforms with graphical user interfaces (GUIs) designed to facilitate analysis by researchers without programming experience. However, it's important that the users understand the many parameters or options that each tool offers to make informed decisions depending on the aims and experimental design of each experiment. Adjusting parameters appropriately will avoid obtaining superficial or uninformative conclusions. In addition, it is advisable to repeat the analysis with at least two different programs as it is known that the choices at each step of the analysis can influence the final results.

1.1.Towards reproducible computational research

Reproducibility is of paramount importance in computational research. The arrival of new tools and sequencing technologies has resulted in a massive influx of data and an increase in the complexity of the questions now being asked. Thus, it is critical that both experimental and bioinformatic researchers provide clear and detailed protocols that allow successful repetition and extension of analyses conducted on the original data. We advise the user to develop a **"culture of reproducibility"** in which individual researchers and institutions establish routines and practices that increase transparency in their research. Keep in mind that good practices of reproducibility may actually turn out to be a time-saver for individuals who might need to apply previously developed methodologies on new data. This advantage also applies to present or future members of your own lab!

Sandve and collaborators (2013) [1] published a set of ten rules to promote reproducible computational research. We encourage users to apply these rules in their research area.

1. *For every result, keep track of how it was produced*
2. *Avoid manual data manipulation steps*
3. *Archive the exact versions of all external programs used*
4. *Version control all custom scripts*
5. *Record all intermediate results, when possible in standardized formats*
6. *For analyses that involve randomness, note underlying random seeds*
7. *Always store raw data behind plots*
8. *Generate hierarchical analysis outputs, allowing layers of increasing detail to be inspected*
9. *Connect textual statements to underlying results*
10. *Provide public access to scripts, runs and results.*

2- Overview of common analysis performed for metagenomic data

Many bioinformatic tools and strategies can be used for the analysis of metagenomic shotgun data (**Figure 1**). A common first step is to run computational tools for quality control, which identifies and removes low-quality sequences and contamination due to commonly used adapters or from impure nucleic acid preparations. The specific set of adapters to be removed, will depend on the sequencing strategy that was used (i.e. shotgun versus amplicon). After quality control, the reads can be either assembled into longer contiguous sequences called ‘contigs’ and/or passed directly to taxonomic and functional classifiers. Taxonomic and/or functional classifications could also be performed directly on assembled contigs. Another strategy is to sort the contigs into so-called "MAGs" (Metagenome-Assembled Genomes). Binning can be done using sequence properties such as composition, abundance profiles, or a combination of both. The choice of assembly versus direct taxonomic classification of reads depends on the research question being addressed and both strategies are often implemented together.

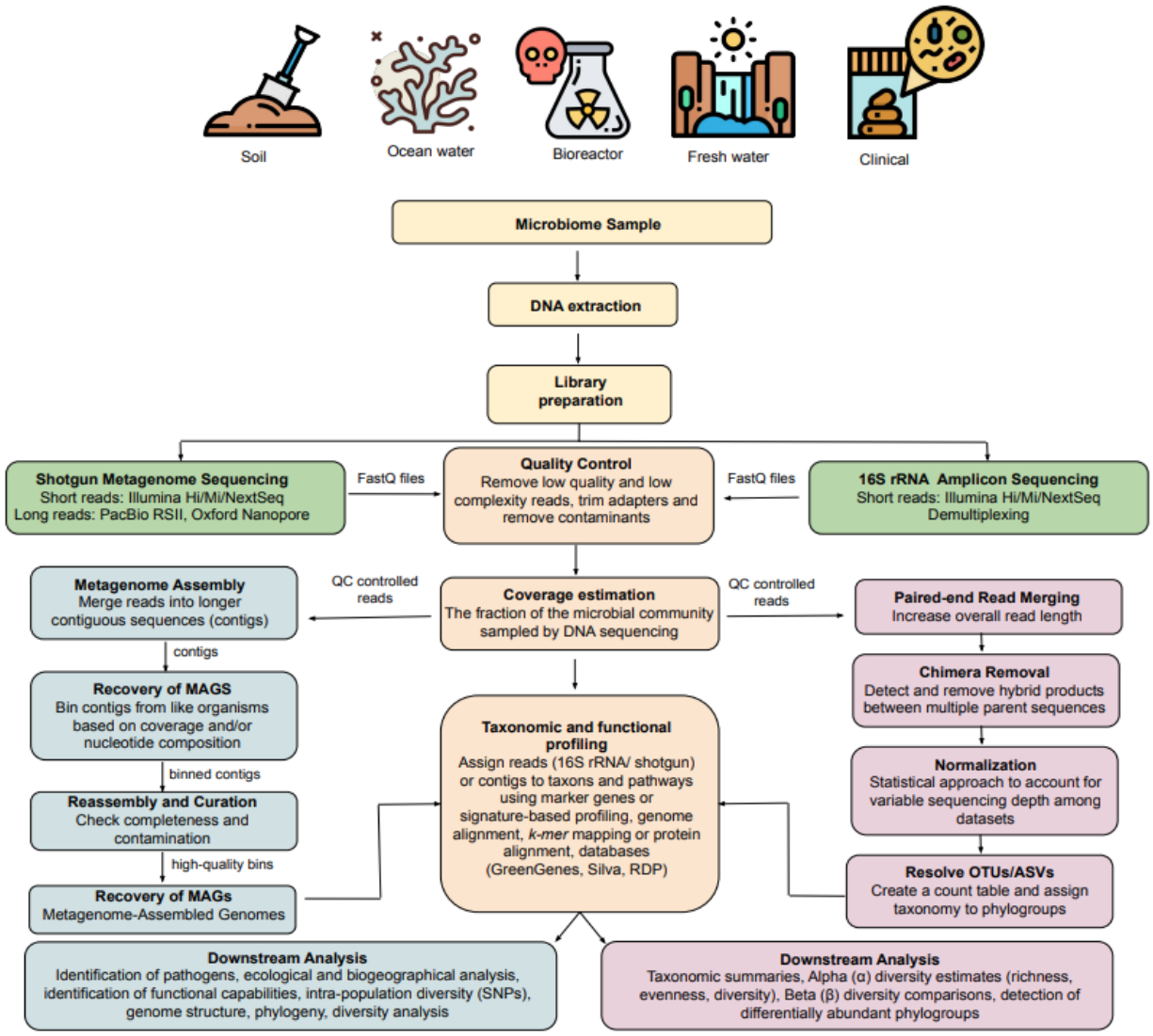

Figure 1. Overview of common bioinformatic data analysis performed in metagenomic and amplicon sequencing studies. Typically the first steps in both shotgun metagenomic and amplicon datasets are: 1) perform quality control (QC) to remove low quality reads, trim adapters and remove contaminants and 2) estimate the fraction of the microbial community that was sampled in the DNA sequencing event (coverage). Next, for metagenomic libraries (boxes in light blue), the reads can be either assembled into contigs and/or used directly with taxonomic and functional classifiers. After assembly, the retrieval of MAGs is performed based on the analysis of the sequence properties, such as composition and abundance profiles. Resulting MAGs, taxonomic and functional annotations are subsequently used in downstream analysis. For amplicon datasets (boxes in light pink), the paired-end reads are usually merged (when possible) to increase the overall read length and then screened to detect and remove chimeric sequences. Once this is completed, the next step is to perform a statistical treatment of the datasets (usually rarefaction) to normalize the number of reads per dataset (in order to make samples and sequencing efforts comparable) and generate Operational Taxonomic Units (OTUs) or Amplicon Sequence Variants (ASVs). Once a count table is generated, the reads are taxonomically annotated using reference databases as GreenGenes, Silva or RDP. Steps that are common to both strategies are color-labeled in yellow.

2.1. Data processing and quality control

Conducting quality control protocols at different points during data processing is crucial to ensure a successful and meaningful study [2]. Controlling the quality of the raw data helps to quickly identify poor-quality samples and flagging data issues. This often means saving a great amount of time in future analysis and identifying stages in the library preparation that could be improved or optimized.

In this initial step, the goal is to obtain basic statistics of the reads for each sample separately. Researchers often want to know the quality of the reads in each position, the percentage of GCs (guanines and cytosines), the average quality per library, the number of Ns (unidentified nucleotides), the presence of repeated and artificial sequences as adapters and primers. In addition to artificial sequences, contaminants can also be detected in the sequences of organisms foreign to the sample [3].

The most commonly used program for quality control is probably FastQC [58]. FastQC is a tool that provides a simple way to perform quality control checks on raw sequence data coming from high-throughput sequencing platforms. You can use this tool to get a quick impression of whether your data has any problems of which you should be aware of before doing any further analysis. This software was particularly designed for Illumina data, however, any second and third generation sequencing results can be used as long as the input is in FastQ format. However, if using a technology other than Illumina, the analysis and interpretation of the metrics may vary.

FastQC provides a report that can spot problems which originate in the sequencer or in the starting library material. In this way, you can get an idea of how good (or bad) the sequencing run performed.

Main features of FastQC include:

- Import data from FastQ files
- Provide a quick overview that identifies in which areas there may be problems
- Provide graphs and tables to quickly evaluate the data
- Export the results to an HTML file
- Support offline operation to allow automatic generation of reports without running the application interactively

A convenient feature of this tool is that rather than looking at quality scores for each individual read, FastQC looks at quality collectively across all reads within a library. Keep in mind that due to the chemistries used for second generation sequencing which are based on amplification, it is common to observe a decrease in quality at the 3 'end of the sequences (see **Figure 2**).

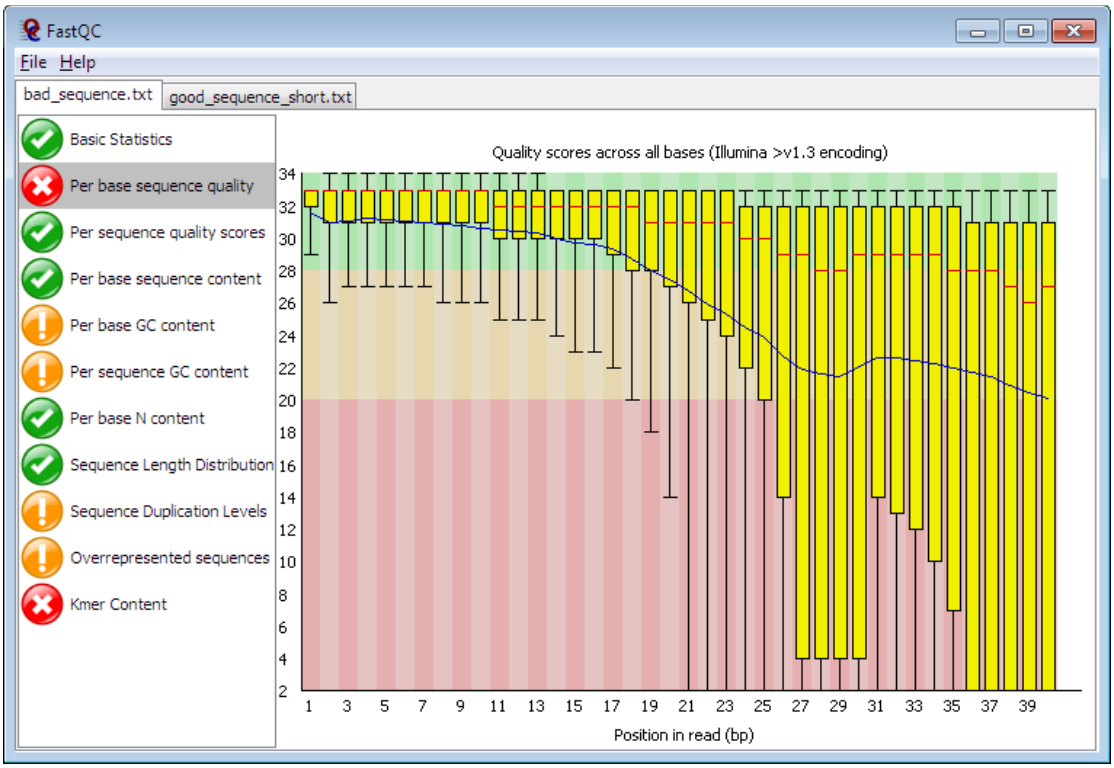

Figure 2. Example of a quality control report provided by FastQC in which a drastic reduction in the quality score (Q<20) is observed for nucleotides from the position 25 to 40 for this particular library. This is an example of a library with poor quality. Image taken from https://www.bioinformatics.babraham.ac.uk/projects/fastqc

Other tools that can be used for quality control are:

- MultiQC (https://multiqc.info/)
- NGS QC Toolkit (http://www.nipgr.ac.in/ngsqctoolkit.html)
- The Galaxy Project (https://usegalaxy.org/)
- Rqc (https://bioconductor.org/packages/release/bioc/html/Rqc.html)
- QuasR (https://bioconductor.org/packages/release/bioc/html/QuasR.html)

After the initial evaluation, sequences are polished (trimming) and sections with low quality are eliminated (Q<20 is usually recommended, implying a 1 in 100 chance of mistake in base calling).

Usually, quality control steps include:

1. Detection and removal of adapters (sequences used to make the libraries, which are different according to the sequencing technology employed)
2. Trimming of low-quality nucleotides (Q<20 is usually recommended)
3. Removal of leading and trailing N (unidentified bases) and large stretches of homopolymers

4. Removal of very short reads after trimming (less than 50 bps recommended)

At the end of this process a quality check run is done again to evaluate the improvement in the data.

Three widely used programs to perform the whole quality control flow outlined above are:

- Trimmomatic (http://www.usadellab.org/cms/?page=trimmomatic)
- FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/index.html)
- SolexaQA++ (http://solexaqa.sourceforge.net/)

Other trimming tools exist to carry out cleaning of sequencing libraries in FastQ format

- CutAdapt (https://cutadapt.readthedocs.io/en/stable/)
- PRINTSEQ (http://prinseq.sourceforge.net/)
- Scythe (https://github.com/vsbuffalo/scythe)
- Sickle (https://github.com/najoshi/sickle)

2.2 Decontamination

During quality control, it is important to assess whether there are sequences that do not come from an organism or community of interest. These are broadly known as contaminants. These contaminant sequences may arise from impure nucleic acid preparations that contain DNA from sources other than the sample [59]. Or, when sequencing host associated microbiomes, contamination from the host organism is common. Even commonly used DNA isolation kits are another potential source of contaminants. In any case, the source of contamination varies depending on the origin of the sample and the way in which it has been manipulated.

Sequence contamination is a serious concern for the quality of the data used for downstream analysis, possibly causing misassemblies of sequence contigs and erroneous conclusions. Furthermore, contaminants may reduce the number of targeted sequences and thus decrease the coverage and the power of the analysis; hence, it is very important to try to maximally reduce contaminants before sequencing. Since this is not always possible, the removal of sequence contaminants present in a library is a necessary step for all metagenomic projects. For instance, when your sequence data is obtained from clinical samples (say fecal samples), the removal of host genomic reads from the metagenomic datasets is critical to ensure that the study's conclusions are based only on the microbial fraction of the sample. The removal of host sequences also protects the subject's privacy, in the case of human specimens.

Useful tools to perform decontamination of sequence data are:

- DeconSeq (https://hpc.nih.gov/apps/DeconSeq.html)
- BMTagger (ftp://ftp.ncbi.nlm.nih.gov/pub/agarwala/bmtagger/)
- MicroDecon (https://github.com/donaldtmcknight/microDecon)

2.3. Coverage estimation

Following quality control and decontamination, the suggested first step of a culture-independent metagenomic study that aims to robustly assess the sequence diversity present in a sample is to estimate coverage, i.e., **the fraction of the microbial community that was sampled by DNA sequencing** [4]. It is important to keep in mind that the fraction of a metagenome that is captured in a metagenomic dataset is not always adequate to draw robust conclusions about the diversity observed in a given sample. This is particularly important when datasets with small depth of coverage are used to characterize highly complex communities such as soil environments.

As explained by Rodriguez and Konstantinidis [5], **a metagenome refers to the theoretical collection of genomes from all members of a given microbial community, while a metagenomic dataset is the subset captured in a given sequencing event**. Although these terms are often used interchangeably, their relationship is analogous to '*population*' and '*sample*' in statistics, respectively. The fraction of the metagenome represented in the metagenomic dataset, termed coverage (not to be confused with sequencing depth), is of key importance when assessing the statistical significance of features observed in a sample (taxa, genes, genomes, etc).

A useful tool to estimate metagenomic coverage is Nonpareil 3 [5]. It is a database-independent tool that allows estimation of coverage in metagenomic datasets by assessing the redundancy of reads. By examining the degree of overlap among individual sequence reads of a metagenomic dataset (by k-mers or alignment), this tool computes the fraction of unique reads, which is subsequently used to estimate the abundance-weighted average coverage. It also fits a projection line to the estimated values to determine the amount of sequencing required for almost complete coverage, for example 95% coverage (**Figure 3**). Nonpareil 3 can be accessed from this link: https://github.com/lmrodriguezr/nonpareil

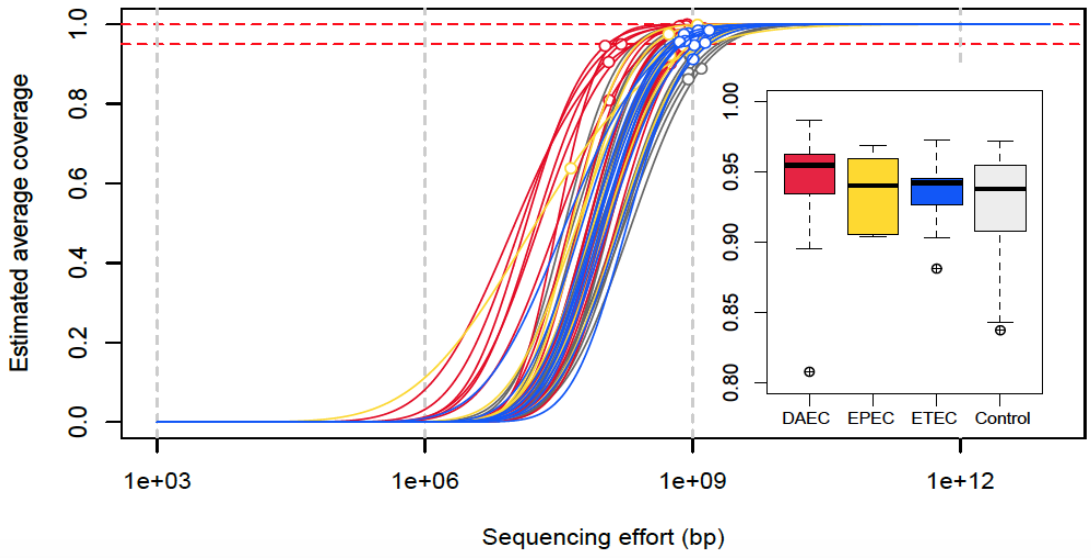

Figure 3. Nonpareil curves example estimated for metagenomic datasets obtained from stool samples collected from individuals infected with different *E. coli* pathotypes: Diffuse Adherent (DAEC), Enteropathogenic (EPEC) and Enterotoxigenic (ETEC) *E. coli* and healthy (control) individuals. The horizontal dashed lines indicate 100% (upper red line) and 95% (bottom red line) coverage. Empty circles indicate the size (x-axis) and estimated average coverage (y-axis) of the datasets, and the lines after that point are projections of the fitted model. Inset plot shows the distribution of estimated coverage values for each pathotype and control groups. This image was taken from [34]

Other existing tools available to estimate metagenomic coverage include:

- COVER (http://botero.cnb.csic.es/cover/)
- MetLab (https://github.com/norling/metlab)

Material to study

For an extended discussion on coverage estimation in metagenomic datasets and why it matters, please review the following articles: https://www.nature.com/articles/ismej201476 and https://academic.oup.com/bioinformatics/article/26/3/295/215491

2.4. Assembly

Assembly involves the merging of reads from the same genome into single contiguous sequences known as 'contigs' [6]. Most available tools build upon a De Bruijn graph approach (explained below) for genome assembly. However, genome assembly is still a very challenging

problem, even for single genomes. The assembly of a highly complex, mixed sample with many species in different abundances, as the case of soil samples for instance, is even more complicated. It requires special purpose assembly algorithms [7, 8].

There are at least six main issues to keep in mind when assembling metagenomic libraries:

1. The highly uneven sequencing depth of different organisms in a metagenomic sample which should be proportional to their relative abundance in the sample.
2. The non-clonal nature of the organisms within the sample (strain mixture, with varying abundances).
3. The depth of coverage of a particular species is rarely high, unless that species dominates the community sampled.
4. Various related species within a microbial community often share highly conserved genomic regions.
5. Requires a great amount of memory and computational processing capacity.
6. The existence of repeat regions within the same genome or across multiple genomes, which tend to break the assembly.

Nonetheless, assembly of metagenomic libraries often succeeds at merging a percentage of reads, resulting in contigs that are easier to align to a database or analyze without reference databases. Whether the percentage of reads successfully assembled is large or small depends on the complexity of the community studied and the sequencing effort.

2.4.1 Overview of metagenomic assembly strategies

In metagenomic studies, Illumina libraries are usually composed of a large number of short reads each with length ranging from 100-400 bp. Therefore, the data consist of a mixture of reads from different microbial organisms with different-sized genomes. However, remember that more recent technologies such as PacBio SMRT and Oxford Nanopore produce reads of much larger lengths.

Both genome and metagenome assembly algorithms can be grouped into three categories, based on their *de novo* assembly strategies [53]:

1. Greedy-extension
2. Overlap-Layout-Consensus (OLC)
3. De Bruijn graph

Greedy-extension

The greedy-extension algorithms start with some reads as seeds, followed by extension of the seeds using the reads with the highest-scoring overlap or the reads whose prefix or suffix have overlap length longer than a given threshold. The algorithms then take the extension of these sequences as new seed sequences and make the next extension until no more reads can be merged [54] (**Figure 4**) . The main limitation of this strategy is that the algorithm just makes the best choice in each step, potentially leading to local optimal solutions, which usually result in a relatively high number of miss-assemblies.

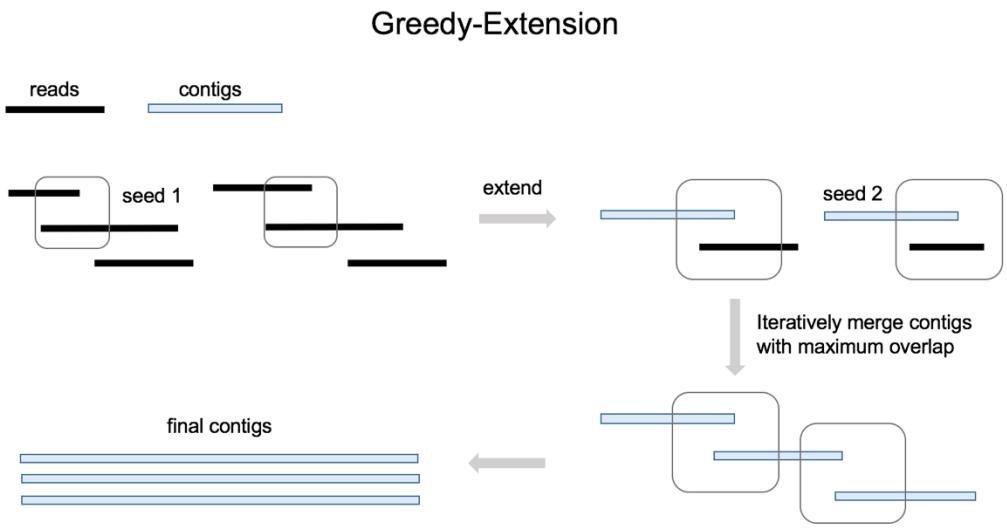

Figure 4. Cartoon showing the principle behind the greedy-extension algorithm for assembly and contig scaffolding.

Overlap-Layout-Consensus (OLC)

The OLC principle consists of three steps. First, **the overlap step** finds the overlaps among all the reads. Second, **the layout step** uses a graph to represent a layout of all the reads and the overlap information obtained in the first step. Finally, **the consensus step** infers the consensus sequence according to the layout (**Figure 5**). This algorithm is very efficient when using long sequences where there is not a high need of sequence coverage and the overlap between sequences is large, with limited false positive overlaps. The main limitation of the OLC algorithms is that the computational cost is very high, making it impractical for processing large metagenomic shotgun sequence libraries, in particular when processing large numbers of short reads.

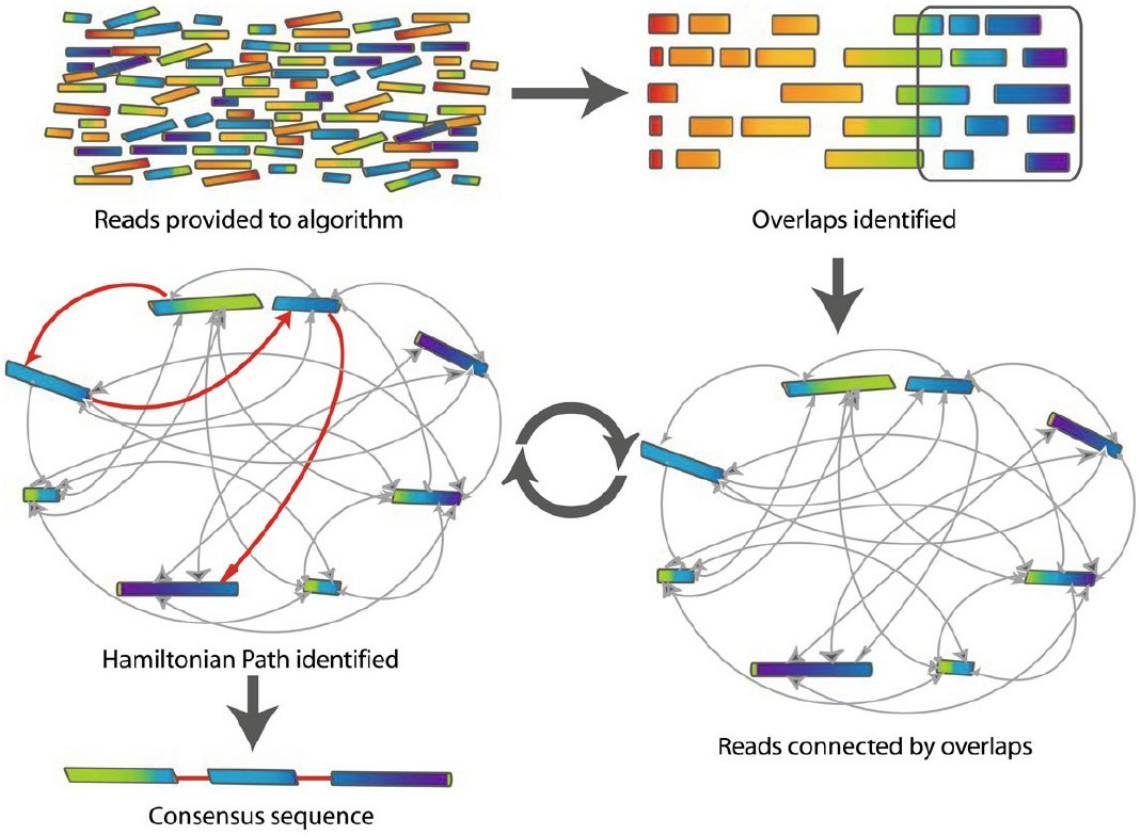

Figure 5. Cartoon showing the principle behind the OLC algorithm for assembly and contig scaffolding. Image taken from *http://pbil.univ-lyon1.fr/members/sagot/htdocs/coursesENS/Lecture5.pdf*

De Bruijn graph

The assembly algorithms based on the De Bruijn graph use the relationship between k-mers (**k-length substrings**) derived from the reads to construct the graph. First, the *k - 1* prefix and suffix of each *k*-mer are extracted and used as two nodes in the graph. If there are no corresponding identified nodes in the graph already, it creates a new node for the (*k - 1*)-mer, and then establishes a directional connection between the prefix and suffix. The resulting graph is called the de Bruijn graph. Next, sequence assembly might be achieved by finding a path that connects all edges from the de Bruijn graph, that is, identifying an Eulerian path in the graph (9) (**Figure 6**). This strategy is more memory efficient but is very dependent on the selection of the right k-mer. Long k-mers lead to longer assemblies but are very susceptible to sequencing errors. Shorter k-mers lead to fragmented assemblies and are more susceptible to smaller repeats, interfering with the assembly [10].

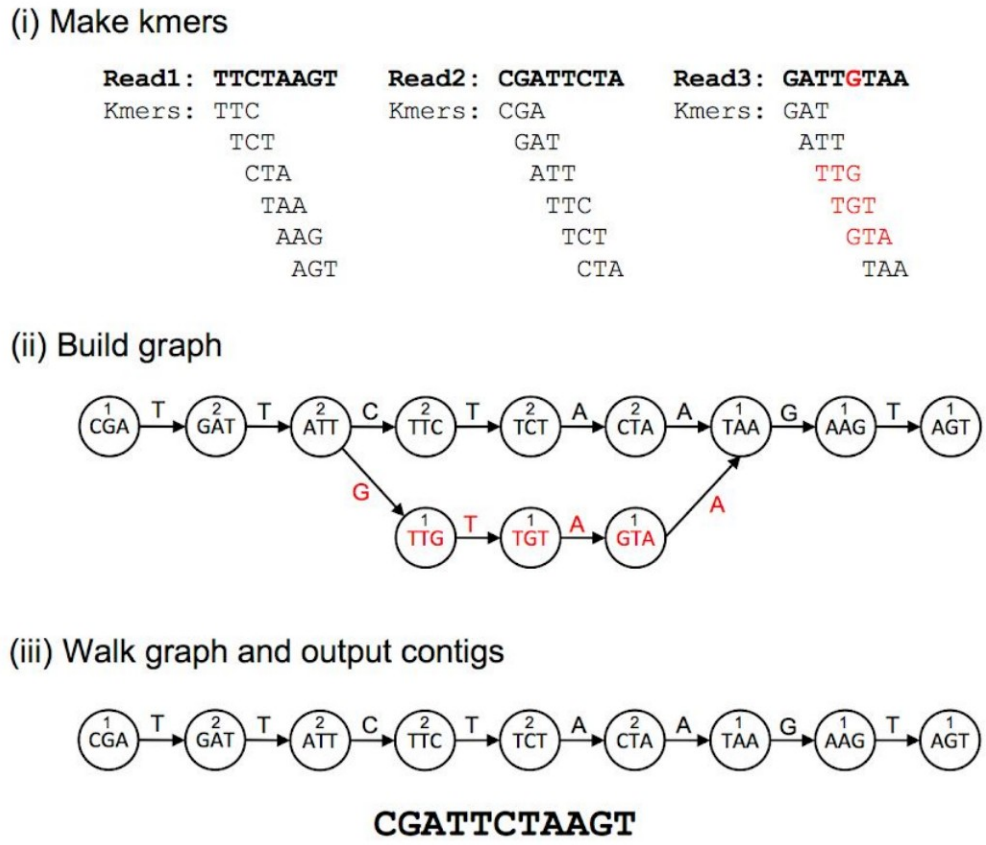

Figure 6. Cartoon showing the principle behind the de Bruijn graph algorithm for assembly and contig scaffolding. In (i) reads are decomposed into kmers by sliding a window of size k across the reads. (ii) The kmers become vertices in the Bruijn graph, with edges connecting overlapping kmers. Polymorphisms (red) form branches in the graph. A count is kept of how many times a kmer is seen, shown here as numbers above kmers. (iii) Contigs are built by walking the graph from edge nodes. Image taken from [11]

Common *de novo* metagenomic assemblers:

- IDBA-UD (https://github.com/loneknightpy/idba)
- MetaSPAdes (http://cab.spbu.ru/software/spades/)
- MEGAHIT (https://github.com/voutcn/megahit)
- Meta-Velvet (http://metavelvet.dna.bio.keio.ac.jp/MV.html)

Stop and think

The choice of k size is important for single k-mer de Bruijn graph assemblers. Small k's are more sensitive in making connections, but fail to resolve repetitive regions. Large k´s may miss connections and are more sensitive to sequencing errors, but usually create longer contigs. Most current metagenomic assemblers thus generate contigs from iteratively constructing and refining de Bruijn graphs using multiple k-mer lengths [11].

2.4.2 Assessing assembly quality

The N50 is a metric often used to assess the quality of an assembly. N50 is defined as the minimum contig length needed to cover 50% of the genome. It means that half of the assembly is contained in contigs of the N50 contig size or longer [12]. For instance, an N50 of 15,000 bp implies that 50% of the assembled bases are contained in contigs of at least 15,000 bp. Usually a larger N50 indicates a better assembly, since it suggests that a larger percentage of the assembly is contained in long contigs. However, keep in mind that this metric only indicates the contiguity of the assembled bases but does not provide a measure on the assembly accuracy. This metric can often be overestimated due to repeats or unremoved adapters that can create artificially long contigs. It must be used cautiously and its significance understood.

QUAST [12] and its metagenomic extension **MetaQUAST** [13] are tools that assess meta(genome) assembly quality. For example, MetaQUAST performs a BLAST search of contigs against a database of *16Sr RNA* genes and will automatically download the top 50 references. It then performs a reference-based quality assessment of contigs that align to these references. This approach is limited to bacterial sequences only. QUAST is accessible in the following link (http://bioinf.spbau.ru/quast) and MetaQuast can be accessed in this link (http://bioinf.spbau.ru/metaquast).

2.5 Gene prediction and annotation

Once reads are assembled, genes can be predicted and functionally annotated. **Gene prediction is the process of locating genes in genomic sequences** [14]. Computational approaches for finding genes are usually divided into two approaches:

1. Homology-based
2. *De novo* (content-based)

Homology-based methods detect genes by searching for similar existing sequences in databases. Tools such as BLAST (Basic Local Alignment Search Tool) [15] can be used to

search for similarities between a candidate gene and a reference database. However, this approach is computationally expensive and cannot be used to discover novel genes. *De novo* methods overcome these limitations using statistical approaches to detect variations between coding and non-coding regions [16].

Frequently used *de novo* tools for gene prediction in prokaryotic (meta)genomes include:

- MetaGeneMark (http://exon.gatech.edu/meta_gmhmmp.cgi)
- MetaProdigal (https://github.com/hyattpd/Prodigal)
- MOCAT2 (https://mocat.embl.de/)
- Orphelia (http://orphelia.gobics.de/)
- FragGeneScan (https://omics.informatics.indiana.edu/FragGeneScan/)

Functional annotation is performed by classifying predicted metagenomics proteins into protein families using sequence information or Hidden Markov Models (HMM) databases. Frequently used sequence databases for functional annotation include:

- SEED (http://pubseed.theseed.org/)
- KEGG (https://www.genome.jp/kegg/kegg1.html)
- MetaCyc (https://metacyc.org/)
- EggNOG (http://eggnogdb.embl.de/#/app/home)
- MG-RAST (https://www.mg-rast.org/)
- PATRIC (https://patricbrc.org/)

HMM databases for metagenomics analysis are usually limited to Pfam (https://pfam.xfam.org/) which uses HMM to model protein domains.

2.6. Metagenomic classification of reads and contigs

There are several tools for matching sequences (reads or contigs) against reference databases to identity taxa, pathways, viruses or genes of interest, as for instance virulence factors or antimicrobial resistance genes within metagenomic datasets. A variety of strategies have been implemented for matching sequences: ultra-fast alignments, mapping *k-mers*, the use of complete genomes as references, identifying marker genes or translating the DNA to protein sequences to find protein families. Below, we will introduce some of the tools and databases more commonly used to generate taxonomic and functional profiles from metagenomic datasets.

2.6.1 Kraken2

Kraken 2 is an extension of Kraken, which is a popular tool that provides ultra-fast classification of all reads in a metagenomic sample. This tool uses an algorithm that relies on exact k-mer

matches, replacing the need for alignment with a simple table lookup. Kraken constructs a database that stores every k-mer in every genome and includes the species identifier (taxonomy ID) for that k-mer. When a k-mer is found in two or more taxa, this tool stores the lowest-common ancestor (LCA) of those taxa with that k-mer. Database k-mers and their taxa are saved in a compressed lookup table that can be rapidly queried for exact matches to k-mers found in the reads or contigs of a metagenomic dataset. Kraken 2 is available at https://ccb.jhu.edu/software/kraken2/

2.6.2 MetaPhlAn2

MetaPhlAn2 is a computational tool for profiling the composition of microbial communities from metagenomic shotgun sequencing data. This tool relies on unique clade-specific marker genes identified from ~17,000 reference genomes (~13,500 bacterial and archaeal, ~3,500 viral, and ~110 eukaryotic) to assign taxonomy unambiguously and estimate their relative abundance. When possible, MetaPhlAn2 provides species-level resolution for bacteria, archaea, eukaryotes and viruses. The limitation of this method is that it requires enough sequenced genomes to derive the marker genes needed to identify species. MetaPhlAn2 is available at http://huttenhower.sph.harvard.edu/metaphlan2

2.6.3 HUMAnN2

HUMAnN2 is the next version of HUMAnN (HMP Unified Metabolic Analysis Network). HUMAnN2 is a complete pipeline for efficiently and accurately profiling the presence/absence and abundance of microbial pathways in a community from metagenomic sequencing data (functional profiling). Functional profiling aims to describe the metabolic potential of a microbial community. More generally, functional profiling answers the question "What are the microbes in my community-of-interest doing (or capable of doing)?". HUMAnN2 is available at http://huttenhower.sph.harvard.edu/humann2

2.6.4 DeepARG

DeepARG is a machine learning solution that uses deep learning to characterize and annotate antibiotic resistance genes in metagenomes. It is composed of two models for two types of input: short sequence reads and gene-like sequences. The main contribution of the deepARG models are their low false negative rates during predictions. Also, the gene-like sequence model is designed to find novel ARGs based on sequence homology. DeepARG is available at https://bench.cs.vt.edu/deeparg

2.6.5 VirFinder

VirFider is a *k-mer* based tool for identifying viral sequences from assembled metagenomic data. This tool uses a machine learning approach to identify viral contigs, avoiding gene-based similarity approaches. VirFinder instead identifies viral sequences based on the empirical

observation that viruses and hosts have discernibly different k-mer signatures. VirFinder is available at https://github.com/jessieren/VirFinder

2.6.6 PathoScope 2.0

PathoScope is a bioinformatics workflow for rapidly quantifying the proportion of reads from individual microbial strains present in metagenomic sequence datasets obtained from environmental or clinical samples. The pipeline performs several computational analysis steps that include reference genome library extraction and indexing, read quality control and alignment, pathogenic strain identification, and summarization and annotation of results. PathoScope can be accessed at https://github.com/PathoScope/PathoScope

2.7. Recovery of Metagenome-Assembled Genomes (MAGs)

This analysis **attempts to reconstruct genomes from metagenomic reads**. Recent advances in sequencing yield and computational techniques have allowed the recovery of draft genomes directly from metagenomic datasets, bypassing the need for culturing. These genomes are called **metagenome-assembled genomes or 'MAGs'**. It has been estimated that MAGs have increased the phylogenetic diversity of bacterial and archeal genome trees by >30% [17] and have provided the first representative strains of 17 bacterial and 3 archaeal candidate phyla [18]. The metagenomic recovery of complete or draft bacterial and archaeal genomes directly from metagenomes is important because it allows us to gain insights into the ecological adaptation, trophic interactions and metabolic versatility of uncultured and eco-genetically adapted microorganisms within their ecosystem context [19].

Main analyses involved in the recovery of MAGs from metagenomic libraries include (**Figure 7**):

1) Metagenomic assembly
2) Characterization of contigs and scaffolds
3) Binning
4) Quality checking

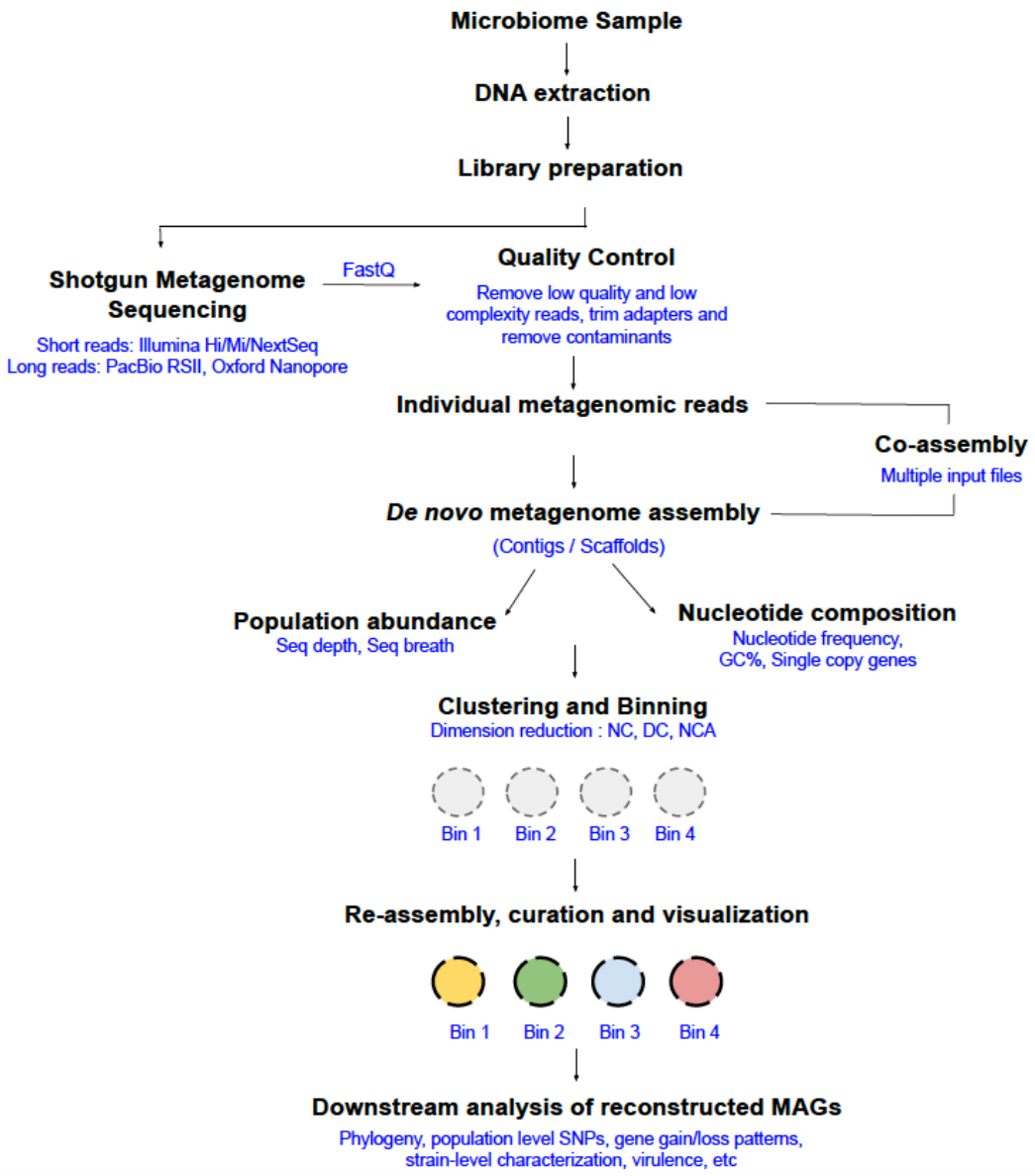

Figure 7. Overview of a bioinformatic strategy for recovering metagenome-assembled genomes (MAGs) from shotgun metagenome data. Figure modified from [22]

2.7.1 Metagenomic assembly

As we discussed previously, **the assembly step involves the merging of reads that originate from the same genome into a single contiguous sequence that we call 'contig'**. This step however is not always easy given that DNA fragments from many different microorganisms will be mixed in the dataset and quite often it is not possible to find and merge reads from the same

genome into a single continuous piece. Due to sequence gaps and repetitive sequences, an assembly is often fragmented into many contigs. The relative order and orientation of these contigs is quite often unknown, and the challenge of ordering and orienting these contigs is called scaffolding. **Scaffolds are composed of contigs and gaps**. Scaffolding is accomplished using paired-end sequencing, where both ends of an original large DNA fragment are sequenced. The success of scaffolding is limited by the size of the library fragment and the ability to span the largest repeat sequences of the genome.

2.7.1.1 Co-assembly

Co-assembly refers to performing an assembly where the input files would be the reads obtained from multiple samples instead of executing an independent assembly for each sample. With the goal of retrieving MAGs from complex metagenomic libraries, there are some benefits to performing co-assembly:

1. It provides a higher read depth, which can help to produce a more robust assembly that can sometimes improve capture of the diversity in the environment of interest.
2. It facilitates the comparison across samples by providing a reference assembly.
3. It can provide the power to recover genomes from complex metagenomic libraries due to differential coverage.

Though a co-assembly has benefits, it is not always ideal in all circumstances. There is no golden rule as to when it is better to perform co-assembly of multiple samples versus running individual assemblies on each sample. This depends on many factors: variation among samples, diversity scale of the microbial community, the software being used, and the research question, among others. In some cases a co-assembly can produce poorer results than individual assemblies. Co-assembly has shown to be particularly useful when using samples from the same population over time, thus, more likely recovering the same organisms but at different abundances. It is even better when a low diversity population (like the human gut) is used. These considerations should be taken into account for each particular dataset. We advise trying and assessing the outcome of both approaches.

Material to study

For more discussion on co-assembly, please visit the following link
https://astrobiomike.github.io/metagenomics/metagen_anvio

2.7.2 Characterization of contigs and scaffolds

Once the reads have been assembled into contigs and scaffolds, the next step is to characterize the assembled sequences in terms of depth of coverage, breadth of coverage, tetranucleotide frequency and GC% content.

- **Depth of coverage**. Per-base coverage is the average number of times a base of a contig (or genome) is sequenced. The coverage depth of a contig is calculated as the number of bases of all short reads that match a contig divided by the length of the contig. It is often expressed as 1X, 5X, 15X (1, 5, 15 times coverage).

- **Breadth of coverage**. Breadth of coverage is the percentage of bases of a contig (or reference genome) that are covered with a certain depth. For example, you can conclude that a given contig is 90% covered at 1X depth and 75% is covered at 5X depth.

- **Tetranucleotide frequency**. The occurrence of certain four-base sequences (ATAT, GGCA, TAAC) is not totally random within an organism, it is biased toward certain tetranucleotides. By calculating the frequency of every possible tetranucleotide, it is possible to group contigs that likely come from the same organism, even though the aligner tool failed to align them.

- **GC% content**. The GC content of a strand of nucleic acid (i.e. contig) is the percentage of nucleotides in the strand that corresponds to guanine (G) or cytosine (C) bases. Because the GC% of the genome differs among species, this metric can offer a rough preliminary test to group contigs that likely come from the same organism.

2.7.3 Binning: grouping assembled contigs into taxonomic bins

In metagenomics, **binning is the process of grouping contigs and scaffolds into discrete units that roughly represent operational taxonomic units (OTUs)** [20, 21]. The binning process typically includes two major components: clustering and data representation. Clustering involves grouping contigs, scaffolds, or genes based on their sequence characteristics (oligonucleotide frequency, GC% content or coverage), using a combination of different approaches as hierarchical clustering (HC) and neural networks. The generated clusters are then grouped into individual taxonomic bins using data representation methods.

According to [22], current methods for recovering MAGs from metagenomic assemblies can be divided in three categories:

- Nucleotide composition-based (NC)

- Differential abundance-based (DA)
- Nucleotide composition and abundance-based (NCA)

The main difference among the three categories resides in the starting point for the content binning process. NC methods rely on oligonucleotide frequency variations. DA methods rely on the coverage of contigs across multiple samples where the organism´s abundance changes. NCA-based methods focus on creating a composite distance matrix from a combination of NC and DA analysis. These approaches have the benefit of not relying on databases, instead they use the composition of each sequence and coverage profiles to cluster together sequences that might belong to the same microorganism.

Although most binning tools can work with single metagenomic samples, most can make use of the differential coverage of multiple samples to improve the binning process.

Three common tools used to perform binning are: Metabat2 [23], MaxBin2 [24] and CONCOCT [21], although several more are also available.

- METABAT2 (https://gitlab.com/jfroula/kbase-metabat)
- MaxBin2 (https://github.com/kbaseapps/kb_maxbin)
- CONCOCT (https://concoct.readthedocs.io/en/latest/)

2.7.4 Quality checking (post-binning analysis)

The rapidly increasing number of MAGs deposited in public databases is starting to compete with the total number of microbial isolate genomes. These MAGs have also raised concern regarding their potential chimeric nature, which sometimes does not meet the quality requirement suggested by the community [25]. MAGs could aggregate sequences originating from multiple distinct populations, which could potentially mislead biological insights when treated and reported as single genomes.

Three common tools used to perform quality check in MAGs are:

2.7.4.1 CheckM

In 2014, Parks et al. presented CheckM, a method for estimating completeness and contamination across population genomes [26]. CheckM provides robust estimates of genome completeness and contamination by using co-localized sets of genes that are ubiquitous and single-copy within a phylogenetic lineage. When marker genes are missing, the genome is probably not complete, and if the marker gene is present multiple times, it suggests contamination. MAGs with estimated completeness above 75% and contamination below 5% are considered of high-quality [27]. CheckM is available at https://ecogenomics.github.io/CheckM/

2.7.4.2 BUSCO V3

BUSCO is a tool to assess genome assembly and annotation completeness with Benchmarking Universal Single-Copy Orthologs [28]. BUSCO uses gene content to assess assembly quality and completeness. It uses a database of single-copy vertebrate, arthropod, metazoan, fungi and eukaryotic genes, as well as a smaller set of prokaryotic universal marker genes. BUSCO v3 is available at https://busco.ezlab.org/

2.7.4.3 Anvi'o

Anvi'o is an advanced analysis and visualization platform for metagenomic data [29]. Its interactive interface facilitates the management of metagenomic contigs and associated data for automatic or human-guided identification of genome bins, and their curation. The visualization approach distills multiple dimensions of information for each contig into a single, intuitive display, offering a dynamic and unified work environment for data exploration, manipulation and reporting. Anvi'o is available at http://merenlab.org/software/anvio/

Material to study

The following publication (link) presents the results of the CAMI challenge (The Critical Assessment of Metagenome Interpretation) in 2017, which was a large effort to engage the global developer community to benchmark their programs in assembly, taxonomic profiling and binning on highly complex and realistic data sets.

2.8. Read-based detection of genes and genomes

To avoid the limitations of the assembly process, such as gaps, truncated genes and misassemblies, an alternative practice to assess gene content variations of genomes and metagenomes is to recruit high-quality (trimmed) reads against a reference and compare these with the predicted genes. This will determine gene presence/absence by the number of reads recruited (or not) (i.e. depth of coverage) and the percentage of the gene length that is covered (i.e. breadth of coverage**)**.

The observed and estimated sequencing depths, as well as the number of reads mapping to each gene in a database, can be calculated using the script "*BlastTab.seqdepth_ZIP.pl*" from the Enveomics collection (http://enve-omics.ce.gatech.edu/enveomics/), assuming a zero-inflated Poisson distribution to correct for non-covered positions. Genes with zero inflation values of ≥0.3, which represents the fraction of the gene that is not covered, are usually excluded. Thus, only genes with ≥70% coverage can be considered to be present.

2.9 Strain-level comparative metagenomics

Strains within the same microbial species can produce substantially different phenotypes. Several studies of both opportunistic and pathogenic species have provided evidence that many microbial phenotypes are strain-specific [30]. For example, species such as *Escherichia coli* comprise strains that are gut commensals as well as highly pathogenic [31] or even carcinogenic [32]. At least six different pathotypes with variant pathogenicity mechanisms have been described for *E. coli* [33]. The massive heterogeneity at the strain level in human and environmental microbiomes still requires systematic characterization. These strain level variations can be metagenomically studied with respect to conditions of interest, including diseases, to generate hypotheses on the mechanistic host-microbiome interactions and to provide microbial targets for diagnosis and therapy [34].

What are microbial strains? Despite the current lack of consensus regarding the definition of a bacterial or archaeal strain, different approaches have been taken based on accumulating single-nucleotide variants (SNV) on core genes with the goal of extracting strain-level signatures from metagenomes.

Stop and think

Metagenomic data allows us to track and compare microbial strains across samples and subjects. However, it's important to realize that strains obtained from metagenomes are likely to have lower quality than genomes obtained from cultures, and it can thus be difficult to evaluate strain variability from low-quality reconstructions. In addition, researchers have observed a rather large inter-subject strain variation both in the gut [35, 36] and in the oral microbiome [37], which suggests that each one of us has a unique microbiome at the strain level. Therefore, it is imperative to apply strain profiling to thousands of metagenomes so we can model the strain diversity at the level of between-subject variability and intra-subject evolution.

There are several tools to characterize strain-level diversity in metagenomes:

- MetaMLST (http://segatalab.cibio.unitn.it/tools/metamlst/)
- StrainPhlAn (https://bitbucket.org/biobakery/biobakery/wiki/strainphlan)
- DESMAN pipeline (https://github.com/chrisquince/DESMAN)
- MetaSVN (http://metasnv.embl.de/)
- ConStrains (https://bitbucket.org/luo-chengwei/constrains/src/master/)

2.10. Beta-diversity

One of the most common analyses in metagenomic studies is the calculation of pairwise dissimilarity between the samples (beta-diversity). **Beta-diversity is a quantitative measure of the differences in composition between two communities** [38]. It's value can be calculated from features like taxonomic or functional composition, phylogenetic structure of the whole community, nucleotide composition, etc.

A dissimilarity matrix composed of pairwise distances between all samples is used for further clustering analysis, classification and study of the influence of the experimental factors. In large-scale studies involving tens and hundreds of metagenomic samples, critical requirements for beta-diversity analysis include high algorithm performance and low memory usage. **Clustering and comparison of massive metagenomic datasets based on *k-mers* is a useful analysis in metagenomic workflows**. Recently, researchers started to compare metagenomes using reference-free methods based on the analysis of oligonucleotide (*k*-mers) frequency previously applied to isolated genomes. The goal is to calculate a pairwise dissimilarity distance among metagenomes to identify clustering by composition similarity.

K-mer based distance estimation among metagenomic data sets

Mash [39] is an ultra-fast computational tool used to estimate distances among large metagenomic datasets based on k-mer profiles. For sequences to be compared with this tool, they must first be sketched (see **Figure 8**), which creates a reduced representation of them. It is recommended to use at least 1,000 reads for this step. It is also a good idea to test at least three different k-mer sizes and evaluate the results since it is known that larger k-mers will provide more specificity, while smaller kmers will provide more sensitivity.

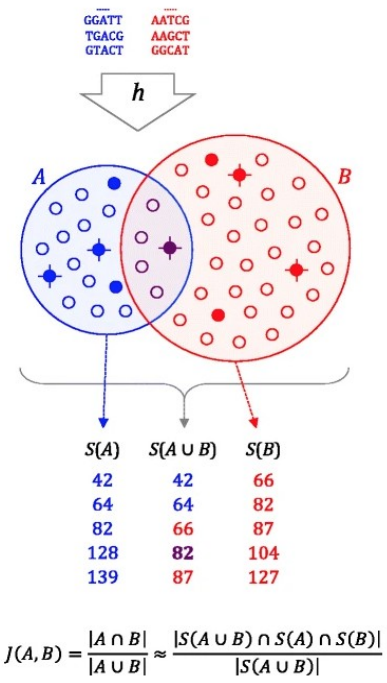

Figure 8. Overview of Mash strategy to estimate Jaccard index in metagenomic datasets. Initially two datasets are decomposed into their k-mers and each k-mer is passed through a hash function ***h*** to obtain a 32- or 64-bit hash, depending on the k-mer size. The resulting hash sets, A and B, contain |A| and |B| distinct hashes each (small circles). The Jaccard index is estimated, calculating the fraction of shared hashes (purple) out of all distinct hashes in A and B. This can be approximated by considering a much smaller random sample from the union of A and B. MinHash sketches S(A) and S(B) of size s = 5 are shown for A and B, comprising the five smallest hash values for each (filled circles). Merging S(A) and S(B) to recover the five smallest hash values overall for A∪B (crossed circles) yields S(A∪B). Because S(A∪B) is a random sample of A∪B, the fraction of elements in S(A∪B) that are shared by both S(A) and S(B) is an unbiased estimate of J(A,B). Figure taken from [39]

3- *16S rRNA* gene amplicon data (metabarcoding) analysis

Data analysis of bacterial and archaeal *16S rRNA* gene amplicon sequence data from complex microbial communities are bioinformatically and computational challenging. We will discuss some of the most widely used software packages for *16S rRNA* gene data analysis: QIIME 2 [40, 41], Mothur [42] and DADA2 [43]. All three packages are open source and have extensive online tutorials and forums.

- QIIME2 (https://qiime2.org/)
- Mothur (https://www.mothur.org/)
- DADA2 (https://benjjneb.github.io/dada2/index.html)

Here, our discussion will focus on data treatments that are common to all packages and will give you an overview of the common analysis that are performed when dealing with amplicon data. However, keep in mind that different workflows can vary on the order of the analysis performed and how they are done.

In general, amplicon data analysis can be divided into five main analyses (**Figure 9**):

1. Data filtering and normalization
2. OTU picking and read grouping
3. Taxonomy assignment
4. Alpha and beta -diversity analysis
5. Detection of differentially abundant OTUs

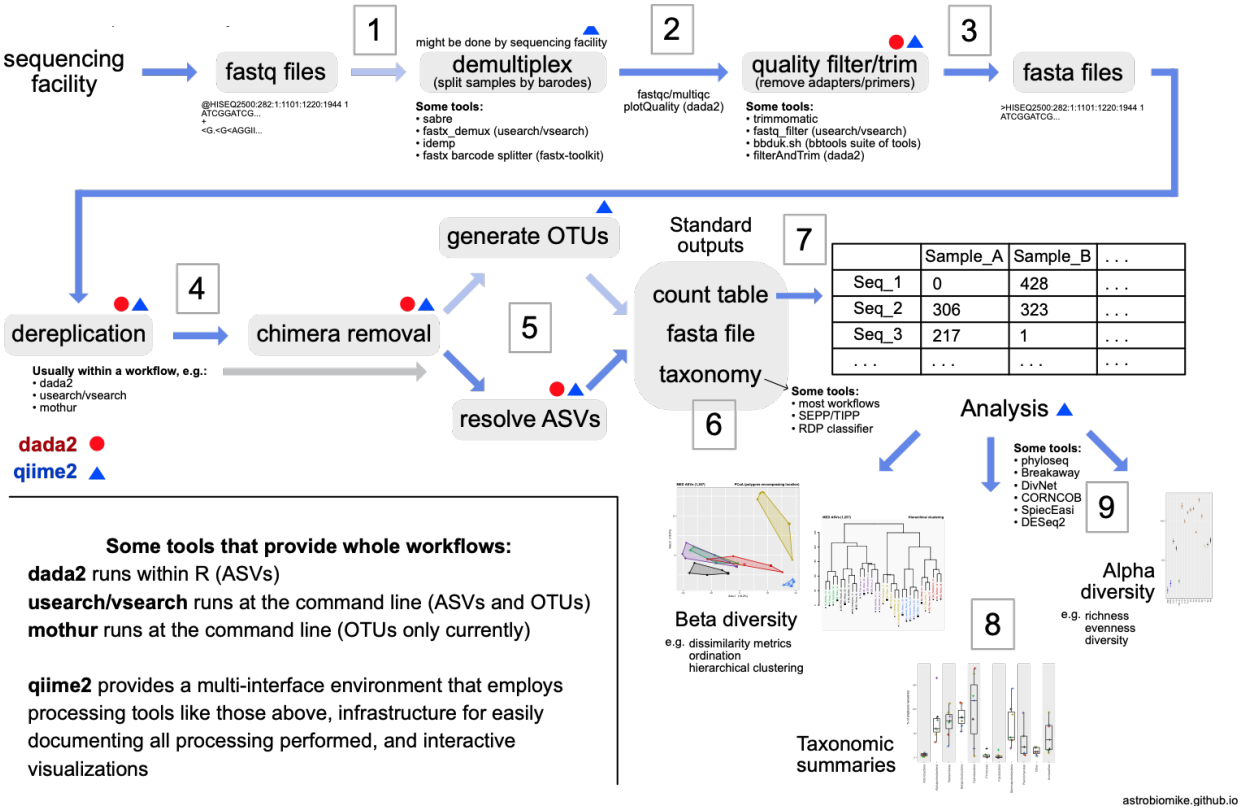

Figure 9. 16S rRNA gene data analysis workflow. In general, amplicon data analysis involves five main analyses: data filtering and normalization (steps 1-4), OTU picking or read clustering into ASVs (step 5), taxonomic assignment (step 6-7), taxonomic summaries and diversity analysis (step 8) and detection of differentially abundant OTUs (step 9). Red circles represent analysis that can be performed with dada2 and blue triangles represent analysis that can be performed with QIIME2. Figure obtained and modified from https://astrobiomike.github.io/amplicon/

Now let's review the different analyses represented in the workflow in more detail.

3.1 Data filtering and normalization (steps 1- 4 in Figure 9)

Analysis of *16S rRNA* gene amplicon sequence data generally starts with demultiplexing. **Demultiplexing is a process in which each read sequence is assigned to its sample of origin based on a specific sequence (barcode) added during library preparation**. Reads not matching any barcode are usually tagged, quantified and discarded. Following demultiplexing, a series of quality control steps are applied to the dataset based on features like quality scores provided by the instrument (Phred score Q), read length and the presence of ambiguous base calls.

An example of quality control processing in an Illumina dataset of the 16S rRNA gene hypervariable region V4 would be:

1. Raw paired-end sequencing reads are initially merged into single reads to increase the overall read length. This step can be accomplished with tools like PEAR (the Paired-End Read Merger) or PANDAseq (paired-end assembler for illumina sequences).

2. Reads are trimmed to the longest continuous segment for which the Phred quality score is greater or equal to 20 (Q≥20).

3. Sequences are additionally inspected to remove any adaptor sequences. This step can be performed with tools like Scythe (https://github.com/vsbuffalo/scythe), cutAdapt (https://cutadapt.readthedocs.io/en/stable/) or PRINSEQ (http://prinseq.sourceforge.net/).

4. Only amplicon sequences longer than 150 bp (~60% of expected length, 254 bp), with low N-content (0.3 is the maximum proportion of Ns allowed) and free of long homopolymers (15 is the maximum number of consecutive identical nucleotides allowed) are retained for further analysis.

5. Detection and removal of chimeric sequences. **Chimeric sequences are hybrid products between multiple parent sequences that can be falsely interpreted as novel OTUs**, thus inflating apparent diversity [44].

The number of sequences obtained in a sequencing run almost always varies across samples for technical rather than biological reasons, and these sequencing depth artifacts can affect diversity estimates. One approach to account for variable sequencing depth is to rarefy the datasets. This statistical approach is termed **rarefaction, where equal numbers of sequences are randomly selected from each sample.** The number of sequences drawn is usually the sequence count of the sample with the smallest acceptable number of sequences. This

sequence count number should reflect a balance between retaining as many sequences as possible without excluding too many low-sequence samples. A major disadvantage of rarefaction is that valuable data from highly sequenced samples are discarded. Thus, rarefaction can lead to a more conservative view of the abundances of rarer taxa across samples.

Stop and think

Alternative approaches to rarefaction for data normalization have been proposed, as it has been observed that this manipulation might introduce error in the analyses, however, this is still a topic of discussion [45]. For further details on the different techniques applied to 16S rRNA gene data normalization, please review the following paper from Weiss et al. [46]

3.2 OTU picking or read grouping (step 5 in Figure 9)

At this stage, sequences are clustered into **operational taxonomic units** OTUs (sometimes referred to as phylogroups), which is useful because it provides a working name for groups of related bacteria. OTUs are defined based on sequence identity (%ID), and various thresholds of sequence identity are used to represent different taxonomic levels (e.g., 97% ID for species level, 95% for genus level). These taxonomic thresholds are known to be **very rough estimates**: the degree of sequence variability depends on the region of the *16S rRNA* gene sequenced, the length of the amplicon, and the specific taxa in question. A sequence identity of 97% is most often used to denote bacterial "species" despite the fact that there is no rigorous species concept for bacteria. Nonetheless, these OTU naming conventions are useful because they have become the shared vocabulary used to discuss sequence-based observations.

There are at least three OTU clustering algorithms: *de novo*, closed-reference and open-reference. The specific OTU-picking algorithm used in your datasets can have a major impact on downstream findings and interpretation of the results.

1. In ***de novo* OTU picking**, sequences are clustered into OTUs, without any external reference sequences [47]. An advantage of this strategy is that all reads are clustered and novel diversity can be potentially detected. However, keep in mind that you cannot use this strategy if you are comparing non-overlapping amplicons, for instance the V2 and V4 regions of the *16S rRNA* gene.

2. **Closed-OTU picking** uses a reference sequence *16S rRNA* gene database as *Greengenes* , Ribosomal Database Project (RDP) or SILVA. Sample sequences that fail to match the reference sequence database are discarded. The advantage of this strategy is that it is fully parallelizable and therefore, large datasets can be run

simultaneously. The main disadvantage is that with the closed-OTU strategy you will not be able to detect novel diversity. Your analysis will only focus on the diversity that is already covered within the collection of sequences in the reference database.

3. **Open-reference OTU picking** is a two-step process consisting of first closed-reference OTU picking followed by *de novo* clustering of sequences that fail to match to the reference database. This is probably the preferred strategy among researchers given that you will cluster all reads at a faster pace than with *de novo* OTU strategy. Just remember that you cannot implement this strategy if you are processing non-overlapping amplicons or if you do not have a reference database to cluster against, for example when you're working with an infrequently used marker genes.

3.2.1 Amplicon sequence variants (ASVs)

In an attempt to advance the methodological aspect of the OTU picking, several approaches have been recently published which attempt to produce biologically meaningful phylogroups independently of a predefined level of similarity. The microbiome research community is currently moving away from the traditional OTU approach and into single-nucleotide-resolving methods that generate what are referred to as ASVs (amplicon sequence variants). Callahan and collaborators [43] developed a tool called **DADA2** (Divisive Amplicon Denoising Algorithm 2) which groups amplicons by considering their abundance distribution (since common reads are more likely to be true sequences) and sequence distance from other reads (since errors are expected to occur at most a few times per read). DADA2 uses the clusters generated and the quality scores of bases, produced by the sequencing platform, to calculate a substitution error model conditioned on quality scores for the sequencing run at hand. Finally, it uses this error model to "correct" reads, that is, assigning low frequency reads to higher frequency reads from which they could be derived by substitution, with high probability [48]. Instead of clustering the reads at some percent identity, all remaining sequences are considered as Amplicon Variants (AVs). **If you are new to *16S rRNA* gene amplicon data analysis, we advise you to follow the ASV approach [43]; however, it's important to be familiar with the traditional concept of OTU clustering by similarity that has been largely employed by the research community and is present in much of the current available literature.**

Other tools that implement single-nucleotide-resolving methods for detection of phylogroups are:

- Deblur (https://github.com/biocore/deblur)
- UNOISE3 (https://drive5.com/usearch/manual/cmd_unoise3.html)

Material to study

You can find a complete DADA2 pipeline for processing amplicon datasets in the following link: https://benjjneb.github.io/dada2/tutorial.html

3.3 Taxonomy assignment (steps 6-7 in Figure 9)

After OTU picking, the next step is to assign taxonomy to representative OTU sequences and generate a list of OTUs with taxonomic labels. Keep in mind that many OTUs will lack a complete taxonomy label; for example, the classification might include a family level categorization but might lack genus or species categorization.

Incomplete taxonomy can result from either a lack of confidence over where the OTU fits in the phylogeny (i.e., several matches are equally likely). For instance, there is a set of organisms so similar in that region of the *16S rRNA* gene that:

- it is not possible to discriminate between them
- or that it is equally distant from everything known and cannot be assigned to any one in particular,
- or from matching to a branch in the phylogeny that lacks taxonomic information.

This is very common, especially for poorly characterized environments [49]. OTUs with genus/species information are more likely to have a reference strain that was cultured, and it's well known that current cultivable organisms are not randomly distributed across the phylogeny. In addition, reference databases are incomplete and often biased toward organisms of medical importance.

After the representative sequences are taxonomically classified, the next step is to tabulate an OTU table (matrix) where each row represents a different observed phylogroup (OTU or ASV) and each column represents a different sample. The initial OTU table contains a direct count of the number of reads assigned to each phylogroup per sample.

Stop and think

It is important to know that most OTU tables are sparse, meaning that they contain a high proportion of zero counts (~90%) [50]. This sparsity implies that the counts of rare OTUs are uncertain, since they are at the limit of sequencing detection ability when there are many sequences per sample (i.e., large library size) and are undetectable when there are few sequences per sample. In addition, the total number of reads obtained for a sample does not

really reflect the absolute number of microbes present in a given environment, since the sample is just a fraction of the original environment [51].

3.4 Alpha (α) and beta (β)-diversity analysis (step 8 in Figure 9)

OTU diversity (a proxy to species diversity) is a valuable tool for describing the ecological complexity within a sample (alpha diversity) or between samples (beta diversity). However, the heterogeneity of an ecological community is not a physical quantity that can be measured directly, and many different metrics have been proposed to quantify the observed variability. Both the alpha and beta diversity analysis that we will describe below can also be applied to the functional and taxonomic data derived from shotgun metagenomes.

3.4.1 Alpha (α) diversity estimates

The description of within sample diversity consists of three components: richness, evenness and diversity.

1. **Richness is a metric describing the number of different species observed and/or estimated in an ecological community**. Species richness is basically a count of species and it does not take into account the abundances of the species or the relative distributions. However, the characterization of species diversity takes into account both richness and evenness. The Chao1 metric [55] estimates species richness based on the detection of species not occurring very often. In short, it uses the number of rare species found just once, and weights those against more common species, to measure how diverse a community is in terms of the number of different species. The higher the Chao1, the more rich the community.

2. **Evenness describes how equally abundant species are in a community relative to one another.** The closer the evenness to 1, the more equally the species are represented in a community (or, said another way, the more equal the probability of pulling an individual of any species in that community at random). Pielou's index (J) [56] can be used to estimate community evenness.

3. **Diversity accounts for both species richness and evenness combined**. Several metrics can be used to estimate diversity, e.g. Shannon and Simpson diversity, among others.

3.4.2 Beta (β) diversity estimates

Beta diversity metrics provide a measure of the degree to which samples differ from one another. Comparative compositional analysis is key in microbiome analysis because these can reveal aspects of microbial ecology that are not apparent from looking at the composition of individual samples [48]. Beta diversity metrics can be grouped in three different ways. First, they can be **quantitative** (using OTU abundance, e.g., Bray-Curtis or weighted UniFrac), **qualitative** (considering only presence-absence of OTUs, e.g., binary Jaccard or unweighted UniFrac) and they can also account for **phylogenetic relationships** (the UniFrac metrics).

Ordination techniques, such as multidimensional scaling plots (NMDS) and Principal Components (PCs), reduce the dimensionality of microbiome data sets so that a summary of the beta diversity relationships can be visualized in two- or three-dimensional scatterplots. The principal coordinates (PCOs), each of which explains a certain fraction of the variability observed in the data set, can be plotted to create a visual representation of the microbial community differences among samples. In ordination techniques such as PCA and PCoA each dimension represents a percentage of the variability seen among the distances between the samples, so it is important to take these numbers into account when interpreting the data.

Non-metric Multidimensional Scaling (NMDS) analysis is a visual representation of all distances or dissimilarities between sets of objects [52]. “Objects” can be colors, faces, map coordinates, etc. In this case, ‘objects’ represent a particular taxonomic configuration (arrangement) of the microbial community per sample. Therefore, microbiomes that are more similar are closer together (or separated by shorter distances) on a NMDS plot than less similar microbial communities (**Figure 10**). In NMDS plots, data is compressed into the number of dimensions that you select. If you select 2 dimensions, they represent 100% of the variation. However it is necessary to take into account the level of stress in the compression of the dimensions to determine if the data really fit the given dimensions or are being forced. A good rule of thumb to keep in mind with NMDS plots: stress <0.05 provides an excellent representation in reduced dimensions, stress <0.1 is great, stress <0.1-0.2 is good/ok and stress >0.3 provides a poor representation.

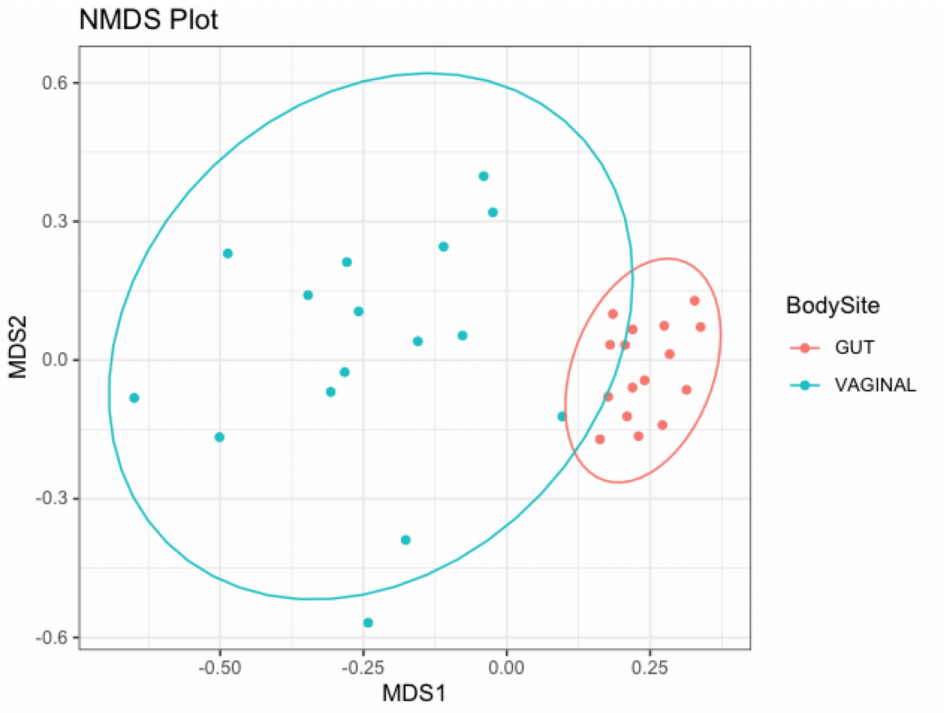

Figure 10. Example of a beta-diversity analysis (NMDS plot) based on *16S rRNA* microbiome data derived from human gut [34] and vaginal samples [60]. Ellipses represent 95% CI (confidence intervals) around the centroid. NMDS plot was generated using the abundance-weighted Jaccard distance metric among samples and the stress is 0.13. Overall, we can observe that there is very little overlap between gut and vaginal microbial communities. This means that the particular OTU composition in each community as well as their relative abundances are not very similar between the two body sites.

3.5 Detection of differentially abundant phylogroups (step 9 in Figure 9)

When comparing groups of samples based on their microbiome data structure, several statistical methods can be used to determine which OTUs (or ASVs) better discriminate between groups, (for example healthy and diseased groups). Graphic tools such as heat maps can be additionally used to visualize over/under representation of these OTUs in the groups. Keep in mind that the goal of this analysis is to perform differential abundance testing to **identify which representative sequences have significantly different copy-number counts between samples** [46]. Recovered *16S rRNA* gene copy numbers do not equal to organism abundance, since the ribosomal operon copy number varies in different species (**Figura 11)** [57]. This is one of the main limitations of amplicon sequencing data.

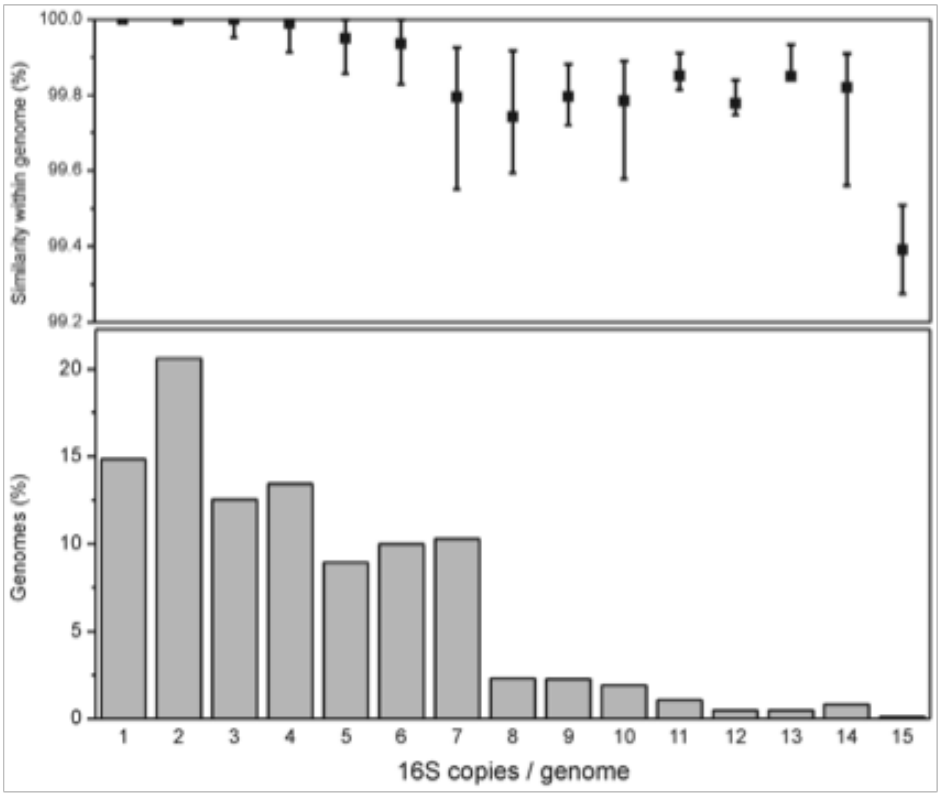

Figure 11. Similarity within the genome and copy number of *16S rRNA* gene in 1,690 sequenced bacterial genomes. Taken from [57]

Several tools with different statistical approaches have been developed for this task, these include:

- DESeq2 (https://bioconductor.org/packages/release/bioc/html/DESeq2.html)
- Metastats (http://clovr.org/docs/metastats/)
- LEfSe (https://bitbucket.org/biobakery/biobakery/wiki/lefse)

4- Final comments and future perspectives

In this tutorial, we presented an overview of the different bioinformatic analyses that are commonly performed in metagenomic studies, including the description and principles behind the different strategies and databases used in typical read-based and assembly-based data analysis pipelines. We also discussed the advantages and limitations of the different tools and methods frequently found in a computational experimental design and provided additional information and website links to download the software and access more information. We anticipate that in the coming years, third generation sequencing platforms will be more

accessible to researchers, producing a larger amount of data (longer reads and higher yields) which will bring new challenges for analyzing, storing and transferring data. This will require the development of new approaches and applications capable of processing and extracting biological information from this large amount of data. Genome-sequencing centers and laboratories performing metagenomic studies will become more dependent on information technologies and bioinformatics to mine data and extract useful information from microbial diversity. Metagenomics is and will continue to play an important role in the fields of biotechnology, clinical, and environmental sciences.

Acknowledgements and funding

We thank Drs Maria Mercedes Zambrano (CorpoGen, Colombia), Rebecca Campos (Universidad de Costa Rica, Costa Rica), and Anisha Thanki (University of Leicester, UK) for their valuable insights and edits on this document. We would also like to thank the members of the Group in Computational Biology and Microbial Ecology (BCEM) at Universidad de los Andes, Colombia for the evaluation and helpful discussions in the content of this tutorial. This tutorial was created for the CABANA eLearning portal with funding from the CABANA project, under Grant ID: BB/P027849/1 supported by the Global Challenges Research Fund (GCRF) through the UK Biosciences and Biotechnology Research Council.

Authors contributions

APG was involved in formal structuring and writing of the tutorial's content. ARM was involved in conceptualization, funding acquisition, supervision, review and editing of this document.

About the organizations and Grant Information

Universidad de los Andes (Uniandes) is a private, research-oriented university located in Bogotá, Colombia. Uniandes has been consistently ranked as one of the best universities in Colombia and is a leader and a reference in higher education in Latin America. Mainly guided by principles of excellence, inclusion, diversity, solidarity, innovation, internationalization and liaison with regions, Uniandes contributes to society with the quality and relevance of its teaching, research and innovation. EMBL-European Bioinformatics Institute makes the world’s public biological data freely available to the scientific community via a range of services and tools, performs basic research and provides professional training in bioinformatics. The CABANA Project aims to strengthen capacity for bioinformatics research and training in Latin America, with the goal of addressing three challenges - management of communicable disease,

protection of biodiversity, and improving food security. It is funded by the Global Challenges Research Fund, part of the UK AID budget.

Competing Interests

The authors declare no conflict of interest

References

1. Sandve GK, Nekrutenko A, Taylor J, Hovig E. Ten simple rules for reproducible computational research. PLoS Comput Biol. 2013 Oct 24;9(10):e1003285.
2. Guo Y, Ye F, Sheng Q, Clark T, Samuels DC. Three-stage quality control strategies for DNA re-sequencing data. Brief Bioinformatics. 2014 Nov;15(6):879–89.
3. Schmieder R, Edwards R. Quality control and preprocessing of metagenomic datasets. Bioinformatics. 2011 Mar 15;27(6):863–4.
4. Rodriguez-R LM, Gunturu S, Tiedje JM, Cole JR, Konstantinidis KT. Nonpareil 3: fast estimation of metagenomic coverage and sequence diversity. mSystems. 2018 Jun;3(3).
5. Rodriguez-R LM, Konstantinidis KT. Nonpareil: a redundancy-based approach to assess the level of coverage in metagenomic datasets. Bioinformatics. 2014 Mar 1;30(5):629–35.
6. Ghurye JS, Cepeda-Espinoza V, Pop M. Metagenomic assembly: overview, challenges and applications. Yale J Biol Med. 2016 Sep 30;89(3):353–62.
7. Lai B, Wang F, Wang X, Duan L, Zhu H. InteMAP: Integrated metagenomic assembly pipeline for NGS short reads. BMC Bioinformatics. 2015 Aug 7;16:244.
8. Wang Z, Wang Y, Fuhrman JA, Sun F, Zhu S. Assessment of metagenomic assemblers based on hybrid reads of real and simulated metagenomic sequences. Brief Bioinformatics. 2019 Mar 11
9. Crawford VG, Kuhnle A, Boucher C, Chikhi R, Gagie T. Practical dynamic de Bruijn graphs. Bioinformatics. 2018 Dec 15;34(24):4189–95.
10. Ye Y, Tang H. Utilizing de Bruijn graph of metagenome assembly for metatranscriptome analysis. Bioinformatics. 2016 Apr 1;32(7):1001–8.
11. Ayling M, Clark MD, Leggett RM. New approaches for metagenome assembly with short reads. Brief Bioinformatics. 2019 Feb 28;
12. Gurevich A, Saveliev V, Vyahhi N, Tesler G. QUAST: quality assessment tool for genome assemblies. Bioinformatics. 2013 Apr 15;29(8):1072–5.
13. Mikheenko A, Saveliev V, Gurevich A. MetaQUAST: evaluation of metagenome assemblies. Bioinformatics. 2016 Apr 1;32(7):1088–90.
14. McHardy AC, Kloetgen A. Finding genes in genome sequence. Methods Mol Biol.

2017;1525:271–91.

15. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990 Oct 5;215(3):403–10.

16. Borodovsky M, Lomsadze A. Eukaryotic gene prediction using GeneMark.hmm-E and GeneMark-ES. Curr Protoc Bioinformatics. 2011 Sep;Chapter 4:Unit 4.6.1-10.

17. Parks DH, Rinke C, Chuvochina M, Chaumeil P-A, Woodcroft BJ, Evans PN, et al. Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. Nat Microbiol. 2017 Nov;2(11):1533–42.

18. Hugerth LW, Larsson J, Alneberg J, Lindh MV, Legrand C, Pinhassi J, et al. Metagenome-assembled genomes uncover a global brackish microbiome. Genome Biol. 2015 Dec 14;16:279.

19. Nayfach S, Shi ZJ, Seshadri R, Pollard KS, Kyrpides NC. New insights from uncultivated genomes of the global human gut microbiome. Nature. 2019 Mar 13;568(7753):505–10.

20. Yu G, Jiang Y, Wang J, Zhang H, Luo H. BMC3C: binning metagenomic contigs using codon usage, sequence composition and read coverage. Bioinformatics. 2018 Dec 15;34(24):4172–9.

21. Alneberg J, Bjarnason BS, de Bruijn I, Schirmer M, Quick J, Ijaz UZ, et al. Binning metagenomic contigs by coverage and composition. Nat Methods. 2014 Nov;11(11):1144–6.

22. Sangwan N, Xia F, Gilbert JA. Recovering complete and draft population genomes from metagenome datasets. Microbiome. 2016 Mar 8;4:8.

23. Kang DD, Li F, Kirton E, Thomas A, Egan R, An H, et al. MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. PeerJ. 2019 Jul 26;7:e7359.

24. Wu Y-W, Simmons BA, Singer SW. MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. Bioinformatics. 2016 Feb 15;32(4):605–7.

25. Shaiber A, Eren AM. Composite Metagenome-Assembled Genomes Reduce the Quality of Public Genome Repositories. MBio. 2019 Jun 4;10(3).

26. Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res. 2015 Jul;25(7):1043–55.

27. Bowers RM, Kyrpides NC, Stepanauskas R, Harmon-Smith M, Doud D, Reddy TBK, et al. Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea. Nat Biotechnol. 2017 Aug 8;35(8):725–31.

28. Waterhouse RM, Seppey M, Simão FA, Manni M, Ioannidis P, Klioutchnikov G, et al. BUSCO Applications from Quality Assessments to Gene Prediction and Phylogenomics. Mol Biol Evol. 2018 Mar 1;35(3):543–8.

29. Eren AM, Esen ÖC, Quince C, Vineis JH, Morrison HG, Sogin ML, et al. Anvi'o: an advanced analysis and visualization platform for 'omics data. PeerJ. 2015 Oct 8;3:e1319.

30. Segata N. On the Road to Strain-Resolved Comparative Metagenomics. mSystems. 2018 Apr;3(2).

31. Croxen MA, Law RJ, Scholz R, Keeney KM, Wlodarska M, Finlay BB. Recent advances in understanding enteric pathogenic Escherichia coli. Clin Microbiol Rev. 2013 Oct;26(4):822–80.

32. Mähner B, Dulce HJ. [Excretory products of hydroxyanthraquinones in rat urine]. Z Klin Chem Klin Biochem. 1968 Mar;6(2):99–102.

33. Nash JH, Villegas A, Kropinski AM, Aguilar-Valenzuela R, Konczy P, Mascarenhas M, et al. Genome sequence of adherent-invasive Escherichia coli and comparative genomic analysis with other E. coli pathotypes. BMC Genomics. 2010 Nov 25;11:667.

34. Peña-Gonzalez A, Soto-Girón MJ, Smith S, Sistrunk J, Montero L, Páez M, et al. Metagenomic Signatures of Gut Infections Caused by Different Escherichia coli Pathotypes. Appl Environ Microbiol. 2019 Dec 15;85(24).

35. Greenblum S, Carr R, Borenstein E. Extensive strain-level copy-number variation across human gut microbiome species. Cell. 2015 Feb 12;160(4):583–94.

36. Schloissnig S, Arumugam M, Sunagawa S, Mitreva M, Tap J, Zhu A, et al. Genomic variation landscape of the human gut microbiome. Nature. 2013 Jan 3;493(7430):45–50.

37. Lloyd-Price J, Mahurkar A, Rahnavard G, Crabtree J, Orvis J, Hall AB, et al. Strains, functions and dynamics in the expanded Human Microbiome Project. Nature. 2017 Oct 5;550(7674):61–6.

38. Legendre P, De Cáceres M. Beta diversity as the variance of community data: dissimilarity coefficients and partitioning. Ecol Lett. 2013 Aug;16(8):951–63.

39. Ondov BD, Treangen TJ, Melsted P, Mallonee AB, Bergman NH, Koren S, et al. Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol. 2016 Jun 20;17(1):132.

40. Bolyen E, Rideout JR, Dillon MR, Bokulich NA, Abnet CC, Al-Ghalith GA, et al. Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. Nat Biotechnol. 2019;37(8):852–7.

41. Caporaso JG, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, Costello EK, et al. QIIME allows analysis of high-throughput community sequencing data. Nat Methods. 2010 May;7(5):335–6.

42. Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB, et al. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. Appl Environ Microbiol. 2009 Dec;75(23):7537–41.

43. Callahan BJ, McMurdie PJ, Rosen MJ, Han AW, Johnson AJA, Holmes SP. DADA2: High-resolution sample inference from Illumina amplicon data. Nat Methods. 2016 May 23;13(7):581–3.

44. Haas BJ, Gevers D, Earl AM, Feldgarden M, Ward DV, Giannoukos G, et al. Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequencing PCR amplicons. Genome Res. 2011 Mar;21(3):494–504.

45. McMurdie PJ, Holmes S. Waste not, want not: why rarefying microbiome data is inadmissible. PLoS Comput Biol. 2014 Apr 3;10(4):e1003531.

46. Weiss S, Xu ZZ, Peddada S, Amir A, Bittinger K, Gonzalez A, et al. Normalization and microbial differential abundance strategies depend upon data characteristics. Microbiome. 2017 Mar 3;5(1):27.

47. Schloss PD, Handelsman J. Introducing DOTUR, a computer program for defining operational taxonomic units and estimating species richness. Appl Environ Microbiol. 2005 Mar;71(3):1501–6.

48. Hugerth LW, Andersson AF. Analysing Microbial Community Composition through Amplicon Sequencing: From Sampling to Hypothesis Testing. Front Microbiol. 2017 Sep 4;8:1561.

49. Goodrich JK, Di Rienzi SC, Poole AC, Koren O, Walters WA, Caporaso JG, et al. Conducting a microbiome study. Cell. 2014 Jul 17;158(2):250–62.

50. Tsilimigras MCB, Fodor AA. Compositional data analysis of the microbiome: fundamentals, tools, and challenges. Ann Epidemiol. 2016 Mar 31;26(5):330–5.

51. Harrison JG, Calder WJ, Shastry V, Buerkle CA. Dirichlet-multinomial modelling outperforms alternatives for analysis of microbiome and other ecological count data. Mol Ecol Resour. 2019 Dec 24;

52. Hawinkel S, Kerckhof F-M, Bijnens L, Thas O. A unified framework for unconstrained and constrained ordination of microbiome read count data. PLoS ONE. 2019 Feb 13;14(2):e0205474.

53. Miller JR, Koren S, Sutton G. Assembly algorithms for next-generation sequencing data. Genomics. 2010 Jun 1;95(6):315-27.

54. Huson DH, Reinert K, Myers EW. The greedy path-merging algorithm for contig scaffolding. Journal of the ACM (JACM). 2002 Sep 1;49(5):603-15.

55. Chao A. Nonparametric estimation of the number of classes in a population. Scandinavian Journal of statistics. 1984 Jan 1:265-70.

56. Pielou EC. Ecological diversity. 1975. John Wiley and Sons, New York, NY

57. Větrovský T, Baldrian P. The variability of the 16S rRNA gene in bacterial genomes and its consequences for bacterial community analyses. PloS one. 2013;8(2).

58. FastQC: A Quality Control Tool for High Throughput Sequence Data [Online]. Available online at: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/ (2015), "FastQC," https://qubeshub.org/resources/fastq

59. Schmieder R, Edwards R. Fast identification and removal of sequence contamination from genomic and metagenomic datasets. PloS one. 2011 Mar 9;6(3):e17288.

60. Tsementzi D, Pena-Gonzalez A, Bai J, Hu YJ, Patel P, Shelton J, Dolan M, Arluck J,

Khanna N, Conrad L, Scott I. Comparison of vaginal microbiota in gynecologic cancer patients pre-and post-radiation therapy and healthy women. Cancer medicine. 2020 Apr 1.