

Computational microbial genomics

Giacomo Fantoni

telegram: @GiacomoFantoni

Github: <https://github.com/giacThePhantom/computationl-microbial-genomics>

April 8, 2022

Contents

1	Introduction	4
1.1	Microbes	4
1.1.1	Prevalence of microbes	4
1.1.2	Difficulties in studying them	4
1.2	Genomics	4
1.2.1	History of sequencing	4
1.2.2	Comparative genomics	5
1.2.3	Metagenomics	5
1.3	Leveraging computational power	5
1.3.1	Comparing low-throughput and high-throughput pipeline	5
2	<i>Escherichia Coli</i> general informations	6
2.1	<i>E. coli</i> genomics	6
2.1.1	<i>E. coli</i> long-term evolution experiment	6
2.1.2	<i>E. coli</i> strains	6
2.1.3	Stc-EAEC outbreak	7
2.1.4	Shigella	7
2.1.5	PanPhlAn - strain detection and characterization	7
2.2	Genomes of <i>E. coli</i>	7
2.2.1	Core and accessory genome	7
2.2.2	Pangenome	8
3	NGS principles (second gen. sequencing) - From Sanger to third gen sequencing	10
3.1	History of Sequencing	10
3.1.1	Progresses of sequencing	10
3.1.2	The Chain Terminators	11
3.1.3	Sanger method: the first one	12
3.2	Development of Sequencing Machines	14
3.3	Next Generation Sequencing NGS	16
3.3.1	Fragments/Library preparation	17
3.3.2	Clonal amplification and ILLUMINA sequencing procedure	17
3.3.3	Pacific Bioscience (PacBio)	20
3.3.4	Nanopore sequencing	20

CONTENTS

4 Sequencing data	22
4.1 Intro to sequencing raw data	23
4.1.1 Base caller	23
4.1.2 FASTQ format	25
4.1.3 Duplication artifacts	27
4.1.4 FASTQ quality control (QC)	28
5 Mapping	29
5.1 Mapping	29
5.2 Mapping algorithm	30
5.2.1 Local vs Global alignment	31
5.2.2 Smith-Waterman algorithm (local alignment) - 1981	31
5.2.3 Needleman-Wunsch algorithm (global alignment)	34
5.2.4 BLAST (Basic Local Alignment Search Tool)	35
5.2.5 Speed seed alignment	37
5.2.6 Burrow-Wheeler alignment	37
5.2.7 LF (Last-First) property	38
5.2.8 Exact mapping using LF property	39
6 Assembly	41
6.1 Feasibility of sequence assembly	43
6.1.1 Last year exercise	43
6.1.2 Merging overlapping reads	44
6.1.3 Overlap graphs	44
6.2 How to solve an overlapping graph?	45
6.3 Graph simplification operations	46
6.3.1 De Bruijn graph assembly	47
6.3.2 Scaffolding	48
6.3.3 Evaluating assemblies	48
7 16S-rRNA sequencing	50
7.1 Introduction to metagenomics	50
7.1.1 Definition of metagenomics	50
7.1.2 Why studying the metagenome	50
7.1.3 Differences with older microbiome studies	50
7.1.4 Example: skin microbiome	51
7.2 16S rRNA sequencing	51
7.2.1 Simplified 16S rRNA analysis workflow	51
7.2.2 16S rRNA gene	51
7.2.3 Primer and high-throughput machine choice	52
7.2.4 In depth 16S rRNA analysis workflow	54
7.2.5 OTU clustering	55
7.2.6 OTU taxonomic annotation	57
7.2.7 RDP classifier (Naive Bayes Model)	58
7.3 Diversity analysis	59
7.3.1 Alpha diversity analysis	59
7.3.2 Beta diversity analysis	59
7.4 Principal Coordinate Analysis (PCoA)	60

CONTENTS

7.5 Study: Enhanced microbial diversity in the saliva microbiome induced by short-term probiotic intake revealed by 16S rRNA sequencing on the IonTorrent PGM platform	60
8 Shotgun Metagenomics	63
8.1 Workflow	63
8.2 Comparison with the 16s sequencing	64
8.3 Latest technology	64
8.4 Identification of microbes from Shotgun Metagenomics data: do we really need something fancy?	65
8.4.1 MetaPhlAn: unique marker genes for taxonomic profiling	66
8.4.2 Other approaches	67
8.4.3 The curatedMetagenomicData resource	68
8.4.4 The link between the gut microbiome and colorectal cancer	68
8.5 PanPhlAn: strain-level profiling	69
8.5.1 Investigating population genomics thanks to PanPhlAn	71
8.5.2 StrainPhlAn: a complementary approach	73
8.5.3 StrainPhlAn applications	74
8.5.4 Uncharacterized species	75
8.5.5 Applications of strain-level metagenomic profiling	76
8.5.6 HUMAnN2: Functional profiling	80
9 Staphylococcus aureus	82
9.1 Immune evasion strategies	82
9.2 Antibiotic resistance in <i>S. aureus</i>	83
9.2.1 Methicillin-resistant <i>S. aureus</i> (MRSA)	83
9.2.2 Methicillin resistance: where is it encoded?	83
9.2.3 Methicillin-resistant <i>S. aureus</i> (MRSA)	84
9.3 Whole genome epidemiology, characterization, and phylogenetic reconstruction of <i>S. aureus</i> strains in a pediatric hospitals	85
9.3.1 Methods	85
9.3.2 Typing methods	86
9.3.3 MLST	86
9.3.4 Spa-typing	86
9.3.5 SCCmec	87
9.3.6 PVL	87
9.3.7 The cohort	88
9.3.8 Typing highlights common clones and newly sequenced ones	88
9.3.9 Co-presence of local, global, animal-associated and hypervirulent clones	88
9.3.10 Genomic signature of chronic versus acute <i>S. aureus</i> infections	89
9.3.11 Variability in SCCmec cassettes	89
9.3.12 Diversity of virulence factors and antigens	89
9.3.13 Virulence factors with available vaccines targets	90
9.3.14 Phylogenetic of specific STs highlights the aggressive spread of a novel independently acquired ST1 clone	90
9.3.15 Conclusions	91

Chapter 1

Introduction

1.1 Microbes

Microbes are defined as whatever is not visible at the human eye: bacteria cells' dimensions are in the order of micrometre, while viruses in the order of nanometre. It is obvious how there is no visible part by eye. This is particularly true for viruses: their dimension make them almost impossible to perceive by any other method than genomics.

1.1.1 Prevalence of microbes

We are living in a microbial world: more than 90% of the biomass is composed of them and they are responsible for a great part of the biochemical cycle. Microbes can thrive in a variety of environment and according to some estimates they compose $10^{17} g$ of biomass. To put that in context the overall weight of the human species is three or four order of magnitude less. They also form the human microbiome, with important medical implication.

1.1.2 Difficulties in studying them

Of the predicted 30 million species to exist only thousands can be cultured in isolation in the lab. There is a need to create a way to directly study and characterized samples taken from the environment.

1.2 Genomics

Once the genetic material is isolated and sequenced a huge amount of information needs to be interpreted.

1.2.1 History of sequencing

- The first sequenced gene was one of a bacteriophage.
- Also the first complete genome was one of a bacteriophage and is used by ILLUMINA as a control.
- The first bacterial genome was pub-

1.3. LEVERAGING COMPUTATIONAL POWER

lished in 1995 and was that of *Haemophilus influenzae*.

- It has a dimension of $1.8Mb$ and sequencing took a year to complete.

- The first archaea was sequenced in 1996.
- In 1996 the genome of *S. cerevisiae* has been sequenced and it was noticed that the genome shows a considerable amount of genetic redundancy.

After having sequenced the genomes there is a need to elucidate the biological functions of the genes contained in them.

1.2.2 Comparative genomics

Studying two different strains of the same organism allows to link difference in the genome to difference in phenotype.

1.2.3 Metagenomics

Metagenomics is the study of the DNA from all the genomes in an environment. By sampling all of the DNA from a given environment, it is possible to study the presence of bacterial ecosystems, independent of the ability to culture each bacterial strain in the laboratory. Large evolutionary radiation of bacterial lineages whose members are mostly uncultivated and only known through metagenomics and single cell sequencing have been described as nanobacterial. They have small genomes and lack several biosynthetic pathways and ribosomal proteins.

1.3 Leveraging computational power

Despite the advantages in DNA sequencing technology the sequencing of genomes has not progressed beyond clones on the order of the size of λ because of the lack of sufficient computational approaches that would enable the efficient assembly of a large number of independent, random sequences into a single assembly. When moving from low-throughput to high-throughput biology statistical power is needed: the genome of a bacterium must comprise all the DNA coding molecules present in the cell. With millions of reads from NGS of an environmental sample, it is possible to get a complete overview of any complex micro biome with thousands of species.

1.3.1 Comparing low-throughput and high-throughput pipeline

Let's consider the pipeline to find the pathogenic agent for a novel outbreak. In a low-throughput one a panel of reasonable putative causative agents is identified. Then one-by-one cultivation protocols to grow the agents from the infected tissue are performed. The pipeline ends when a pathogen grows in a sample. This is very time consuming. High-throughput instead sequences the full DNA repertoire of the sample and try to identify the pathogen by its genomic signature.

Chapter 2

***Escherichia Coli* general informations**

2.1 *E. coli* genomics

Escherichia Coli is a Gram-negative, facultative anaerobic, rodshaped, coliform bacterium, it pertains to the phylum of proteobacteria and to the family of Enterobacteriaceae. It can be grown easily and inexpensively. Its genome has a length between 4.5-4.7Mb, including about 4000-5000 genes, and about seven ribosomal RNA operons. Only the 38% of the genes of K-12 (one of the most studied bacterial strains of *E. coli*) were experimentally identified, overall 40-50% of the genes are to date without a known function. The original *E. coli* strain K-12 was obtained from a stool sample of a diphtheria patient in Palo Alto, California in 1922.

2.1.1 *E. coli* long-term evolution experiment

The *E. coli* long-term evolution experiment led by Richard Lenski is one of the longest evolutionary experiments ever made. Starting from the 24th of February 1988 12 population of *E. coli* have been cultivated in parallel. After each day a portion of the population was introduced in a new flask, where it proliferated. Every 500 generations a sample from each flask is saved, so to track the evolutionary changes made. Today the 66000th generation have been reached. Long-term adaptation to a fixed environment can be characterized by a rich and dynamic set of population genetic processes, instead of the evolutionary desert expected near a fitness optima. In particular some bacteria developed the capacity to aerobically grow on citrate.

2.1.2 *E. coli* strains

E. coli could be found as commensal strains, pathogenic strains, or environmental strains. The pathogenic strains could pertain to these categories (which are not exclusive):

2.2. GENOMES OF *E. COLI*

- enteropathogenic (EPEC),
- enteroinvasive (EIEC),
- enterotoxigenic (ETEC),
- diffusely adherent (DAEC),
- adherent invasive (AIEC),
- shiga-toxin producing (STEC),
- enteroaggregative(EAEC),
- extraintestinal pathogenic (ExPEC).

Resistances to antibiotics adds another layer of complexity in the categorization categorization of *E. coli*. Most of the genes are on plasmids, circular, additional to chromosome, and can be moved easily horizontally. Plasmids between different strains can be moved in enterobacteriaceae, this doesn't happen normally in other families. Some *E. coli* strains are even capable of causing cancer in humans: for example, colibactin-positive *E. coli* can cause colon and rectal cancer, by creating mutations which are responsible of the onset of the cancer.

2.1.3 Stc-EAEC outbreak

In 2011 in Germany there was an outbreak of Stx-EAEC. An efficient counter-measure was found by sequencing the genome of those bacteria.

2.1.4 Shigella

Shigella is a strain of *E. coli* capable of producing the shiga toxin. It has been difficult to categorize for taxonomists. Several antigens can be used by taxonomists to categorize *E. coli* strains. In particular the O (171), H (56), K (80) antigens, respectively related to the somatic, the flagella and the capsule are widely used.

2.1.5 PanPhlAn - strain detection and characterization

Pangenome-based Phylogenomic Analysis (PanPhlAn) is a strain-level metagenomic profiling tool for identifying the gene composition and in-vivo transcriptional activity of individual strains in metagenomic samples. PanPhlAn's ability for strain-tracking and functional analysis of unknown pathogens makes it an efficient tool for culture-free infectious outbreak epidemiology and microbial population studies. This tool was for example used to study the strain responsible of an outbreak in Germany in 2011 and found a strain of shiga-toxigenic Escherichia coli (STEC). This method, due to its greater efficiency and accuracy has been used since instead of low-throughput, traditional pipelines.

2.2 Genomes of *E. coli*

2.2.1 Core and accessory genome

In the genome of *E. coli* strains, it is possible to distinguish between:

- Core genome: the set of all genes shared by all members of a bacterial species, it includes 1000 up to 3000 genes.
- Accessory or dispensable genome: the

2.2. GENOMES OF *E. COLI*

set of genes present in some but not all genomes within the same bacterial

species. It is found on a single strain or in a subset of strains.

2.2.2 Pangenome

The pangenome is the union of the core genome and the accessory genome. It is the set of all the set of all the genes that can be found in the species strains. The pangenome can be characterized regarding its size with respect to the number of genomes:

- Closed: the pangenome size tends to a maximum as number of genomes increases.
- Open: the pangenome size keeps increasing as you add new genomes.

Typically sequencing more organisms of the same species tends to lower the amount of genes in the core genome and increase the number of those in the pangenome, 2.1. Due to technical errors, the core genome should tend to a size of 0, but a more reasonable plateau can be predicted with a more accurate mathematical formulation.

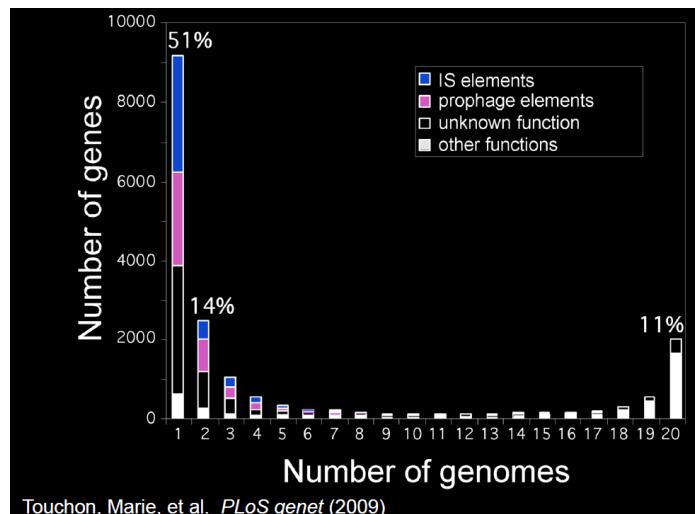


Figure 2.1: It can be seen that 51% of the genes are strain specific, and the other are shared between 2 to 20 strains of *E. coli*

Each *E. Coli* genome contains a balance genes of the core genome and of the pangenome, for a total amount of 4700 genes 2.2). Core genomes' genes are responsible of some basic cellular functionalities and utilities to survive the environment, while instead elements of the pangenome are quite usually specific to a single strain, like for example antibiotic resistance, and they are often not functionally well characterized.

2.2. GENOMES OF *E. COLI*

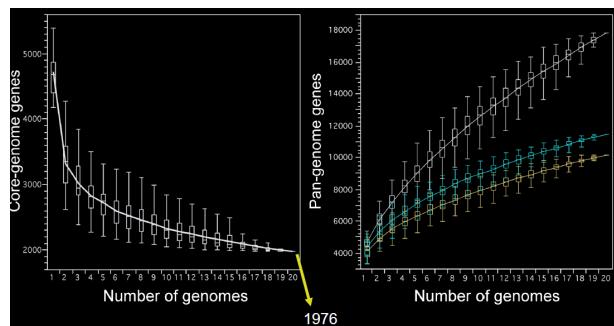


Figure 2.2: Balance between genes of the core- and of the pan-genome

Chapter 3

NGS principles (second gen. sequencing) - From Sanger to third gen sequencing

Figure 3.1

- 1959 – First homeogenous DNA purified
- 1970 – First discovery of type II restriction enzymes
- 1972 – First RNA gene sequence published (lac operon)
- 1975 – Sanger first publishes his plus/minus method of sequencing (unable to distinguish homopolymers)
- 1977 – Maxam & Gilbert publish their method (could distinguish homopolymers)
- 1977 – Sanger publishes Dideoxy sequencing method

NGS stands for **Next Generation Sequencing**, and it represents the method of sequencing most used nowadays. Before that, a series of other discoveries were done, elen-cated in the figure 3.1.

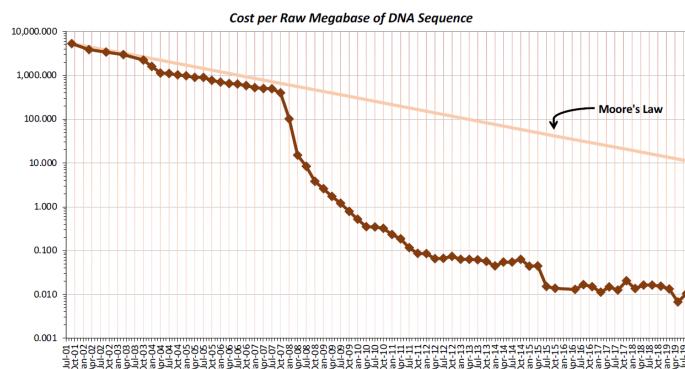
3.1 History of Sequencing

3.1.1 Progresses of sequencing

As the time passes, the cost of sequencing the DNA is diminishing. The rate of decrease actually is higher than the one predicted by the **Moore's Law**, "*Democratization of sequencing*" is seen nowadays, because of the costs lowering. After the Human genome project, several other animals and plants' genomes were sequenced.

3.1. HISTORY OF SEQUENCING

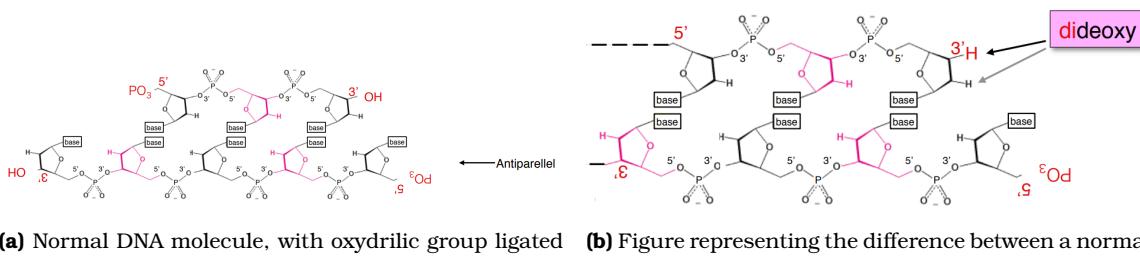
Figure 3.2



The methods of sequencing actually can be grouped in three **groups**, that are:

- **Chemical degradation** of DNA: it includes the method of Maxam-Gilbert
- **Sequencing by synthesis ("SBS")** which is the most common approach and the first to be developed. It uses DNA polymerases in primer extension reactions Illumina, Pacific Biosciences, Ion Torren and 454
- **Ligation-based**: sequencing using short probes that hybridize to the template, the technologies pertaining to this class are SOLiD, Complete Genomics
- **Other**: Nanopores

3.1.2 The Chain Terminators



(a) Normal DNA molecule, with oxydrilic group ligated to 3'-ends

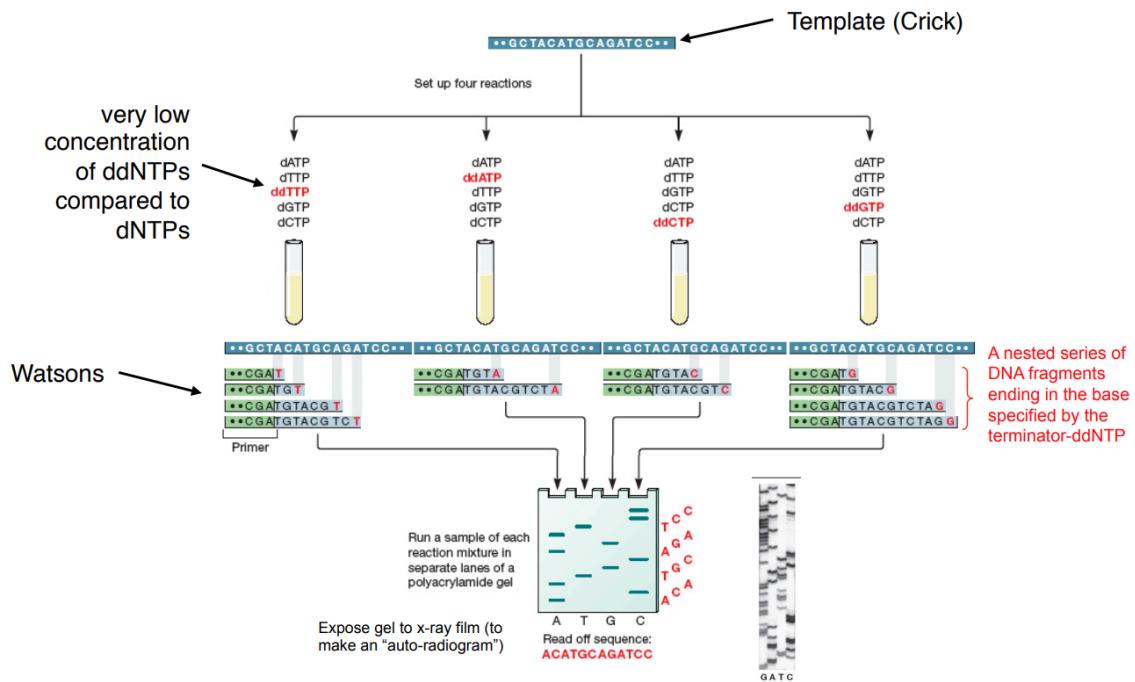
(b) Figure representing the difference between a normal DNA chain and one with chain terminators

Figure 3.3: Normal DNA synthesys vs Chain terminators

Normally, the addition of new nucleotides to a generated molecule of DNA happens with the 3'-end of the nucleotide chain (figure 3.3a). Chain terminators are dideoxy nucleotides, ddNTPs, that cannot be further extended. These nucleotides aren't able to add a new nucleotide on the 3'-end, as they have not the needed oxydrilic group (figure 3.3b).

3.1. HISTORY OF SEQUENCING

Figure 3.4: The Sanger's method process



3.1.3 Sanger method: the first one

The first method ever used to sequence DNA was designed by Frederick Sanger. The Sanger manual sequencing system consists in an *in vitro* process, which is described in figure 3.4, also named as "*primer extension*" method. It is performed over a single-filament DNA sample, and it uses the chain terminators nucleotides, a type for each nucleobase: ddATP, ddGTP, ddTTP, ddCTP.

The reaction is done inside four different reactions tubes, each containing the sample DNA to be reproduced, a DNA polymerase, the normal nucleotides and one of the four possible chain terminator. The chain terminators are marked with sulfur-35, a radioactive atom. In each tube, the corresponding dideoxy-nucleotide was used with a concentration 10 times lower than the other "normal" nucleotides.

From the polymerization reactions, several molecules of DNA were produced, with different length: each replicative cycle is in fact terminated after the addition of a chain terminator nucleotide.

To reconstruct the initial DNA sequence, a long PAGE gel was prepared, with high concentration of urea ($6 - 7M$) to avoid the coiling of the DNA single-filaments. To run the gel, high voltages were required and it had to be highly resolute, as DNA's fragments are different only for a nucleotide. It was needed to do an auto-radiography of the gel in order to see the bands, in order to differentiate the fluorescent signals.

To read the sequence you have to start from the shortest fragments, at the end of the gel, and carefully go up along the gel, looking for the first presence of a band in one of the

3.1. HISTORY OF SEQUENCING

four runs.

Past procedures: In the past, the Klenow's fragment (part of the 1st DNA polymerase) was used to perform the Sanger method, and the DNA to be sequenced was inserted inside the genome of an M13 phage.

3.1.3.1 Automatic sequencing

It does not use the radioactive signals, instead it uses fluorescent proteins. Several versions were developed after modifications of the Sanger method, in this order:

1. fluorescent primers marked with a single fluorochrome.
2. four aliquotes of the same primer were used marked with four different fluorochromes, able to emit different fluorescences.
3. four different fluorochromes were used to mark the single ddNTPs

Thanks to the use of 4 different fluorochromes, it was possible to use a single electrophoretic lane to carry the sequencing reaction. For this type of sequencing also, a cyclic replicative reaction was performed, with this procedure, made possible by using a thermal cycler:

1. **Denaturation at 95°C** of the DNA to be sequences
2. **Annealing at 50 – 70°C** of the primer specific to one of the two filaments
3. **Extension at 72°C** by using a *Taq*-polymerase. The use of the *Taq*-polymerase makes it possible to avoid the formation of coiled structures in the DNA molecule to be sequenced.

Traditionally, also in this type of sequencing it is used a long PAGE gel, as the Sanger's one, but with a great difference: all the ddNTPs are inserted in the same electrophoretic run, and after it, it is not needed an auto-radiography. Fluorescence, instead, can be triggered simply by irradiating the DNA molecules synthesized, which produce different fluorescences with different wavelengths.

More usefully, this sequencing method is performed by using a capillary, filled with a synthetic polymer, with the same function of polyacrylamide.

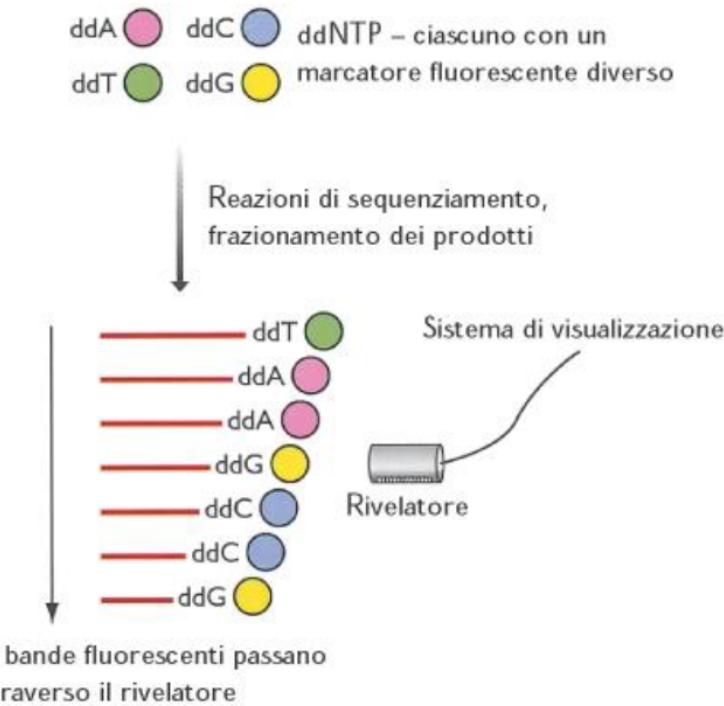
At the end of the analysis, it is produced an electropherogram, with a color depicting the probability of each nitrogen base in each position. The production of the electropherogram is made better thanks to algorithms to boost signal/noise ratio, to correct the dye-effects, and other effects that generate systemic errors.

Regarding the Sanger machines in use, the upgrades viewable in figure ???. Based on the same technology, new machines were developed and it was obtained a ~1000 fold improvement.

When talking about the **human project**, it is important to specify that most of the job was done by using the Sanger automatic sequencing method.

3.2. DEVELOPMENT OF SEQUENCING MACHINES

Figure 3.5



3.2 Development of Sequencing Machines

The way you can get to the point could be different based on technology and the wanted output, comprised quality.

SOLID gave only 35-75 sequenced bases, and it is not used anymore; Sanger sequencing, the capillary, can give up to 1000 read length, with a low output; on MINION it was possible to sequence an entire genome of *E. coli*. A plethora of sequence machines are available today. None of the machines are able to sequence DNA from a sample of blood, some things have to be done. Nowadays machines producing bigger outputs of short reads are preferred.

3.2. DEVELOPMENT OF SEQUENCING MACHINES

Figure 3.6

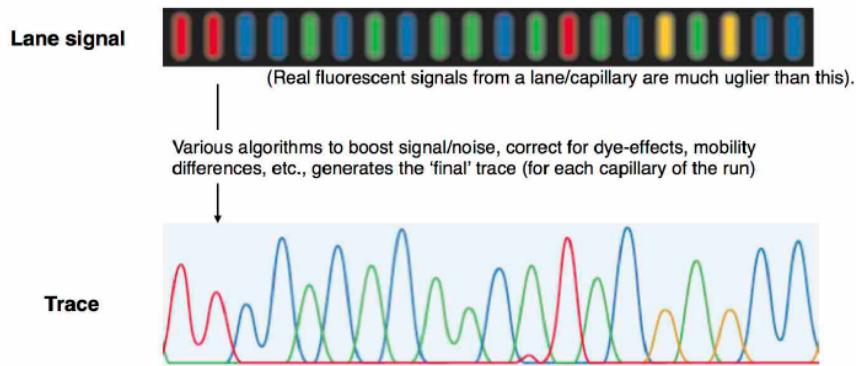
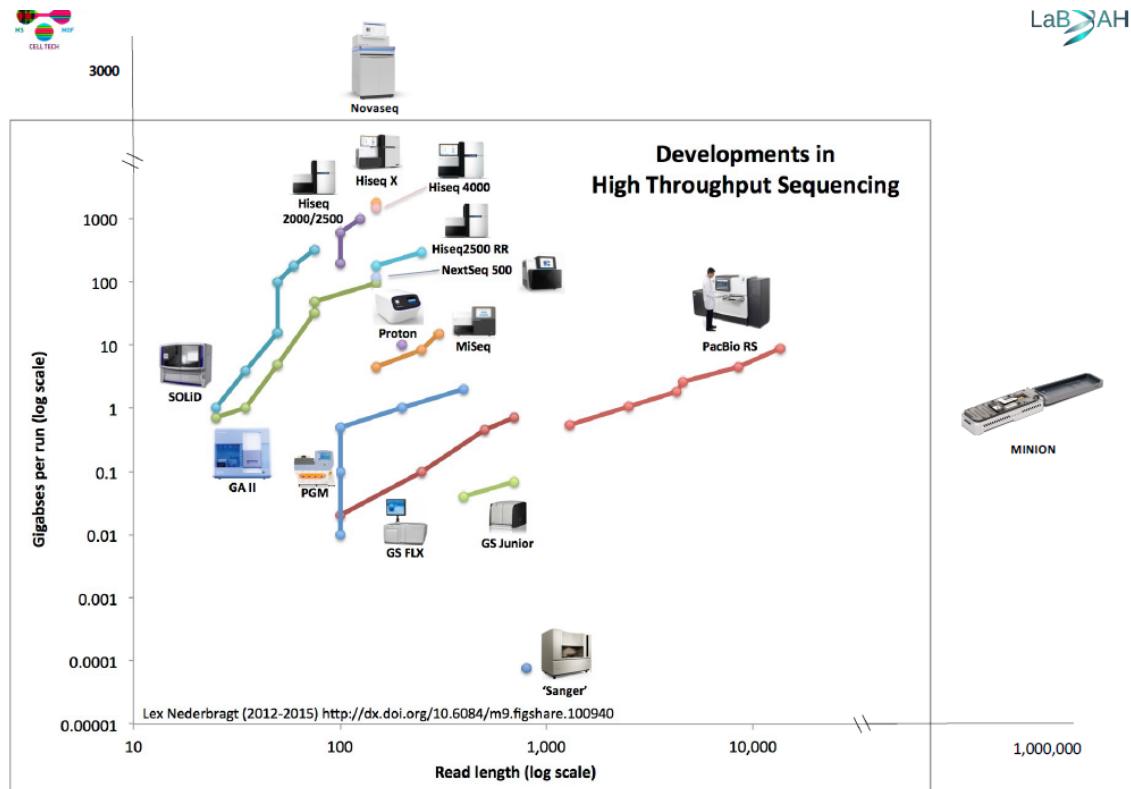


Figure 3.8: It can be noticed how recent developments had the scope of increasing the output data



At the top of the market nowadays there are ILLUMINA machines, that use sequencing by synthesis method, NovaSeq is the biggest one. They need to amplify the signal through clusters formation.

3.3. NEXT GENERATION SEQUENCING NGS

Figure 3.7: The implementation of capillary sequencing machines gave the possibility to make more runs than with the others. A ~1000 fold productivity increase was allowed



		
Radioactive polyacrylamide slab gel Low throughput, labor intensive	AB slab gel sequencers (370, 373, 377) Fluorescent sequencing 1990-1999 6 runs/day 96 reads/run 500 bp/read 288,000 bp/day	AB capillary sequencers (3700, 3730) 1998-now 24 runs/day 96 reads/run 550 – 1,000 bp/read 1-2 million bp/day

Figure 3.9

ILLUMINA PRODUCTION-SCALE Sequencers



	NextSeq Series	HiSeq Series	HiSeq X Series [†]	NovaSeq Series
Popular Applications & Methods	Key Application	Key Application	Key Application	Key Application
Large Whole-Genome Sequencing (human, plant, animal)	●	●	●	●
Small Whole-Genome Sequencing (microbe, virus)	●	●		●
Exome Sequencing	●	●		●
Targeted Gene Sequencing (amplicon, gene panel)	●	●		●
Whole-Transcriptome Sequencing	●	●		●
Gene Expression Profiling with mRNA-Seq	●	●		●
miRNA & Small RNA Analysis	●	●		●
DNA-Protein Interaction Analysis	●	●		●
Methylation Sequencing	●	●		●
Shotgun Metagenomics	●	●		●

3.3 Next Generation Sequencing NGS

The NGS protocol requires 3 steps, that are:

1. **Sample preparation:** series of fragments added
2. **Clonal amplification:** which is needed to replicate fragments attached to the solid surfaces, since machines are not sensible to single molecules

3.3. NEXT GENERATION SEQUENCING NGS

3. **Sequencing:** ILLUMINA sequencing is one of the techniques used to obtain sequence data nowadays

Tools pertaining to the 3rd generation are those that permit to read a molecule without replicating it.

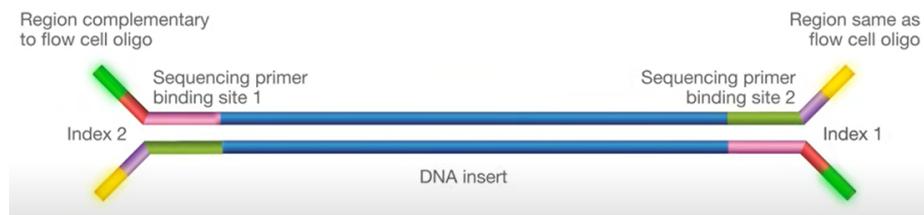
3.3.1 Fragments/Library preparation

Most of the sequences sequenced are fragmented in short read sequences, since most of the machines today used aren't able to sequence reads longer than some hundreds of nucleotides.

The fragments obtained have to be prepared for the sequencing process, through a process called **tagmentation**. The obtained fragments are shown in the figure. Those fragments are provided with one or two indexes, called also barcodes, two sequencing primer binding sites and regions complementary to the oligonucleotides present in the chamber (see in Clonal amplification chapter). The fragments' length has to be checked, depending on the scope of the process. Indexes are needed to run sequencing process on multiple samples, they are needed to distinguish those; when they are two, they permit to distinguish also the 2 types of sequencing that are performed, forward and reverse.

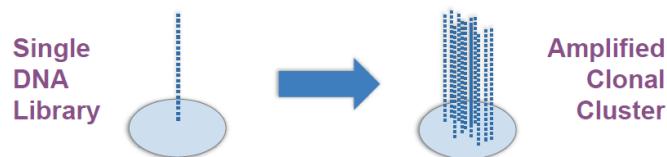
P5 and P7 are the oligos needed to attach fragments to ILLUMINA sequencing machines.

Figure 3.10: Figure representing the a good prepared fragment, it has two indexes, two sequencing primer binding sites and regions complementary to the oligonucleotides present in the chamber



3.3.2 Clonal amplification and ILLUMINA sequencing procedure

Figure 3.11



Clonal amplification are necessary to amplify the signal from each single fragment. ILLUMINA machines make use of clusters to sequence DNA. Clusters are a group of DNA strands positioned closely together and generated from a single DNA filament. Generally,

3.3. NEXT GENERATION SEQUENCING NGS

Each cluster represents thousands of copies of the same DNA strand in a 1–2 micron spot (figure 3.11).

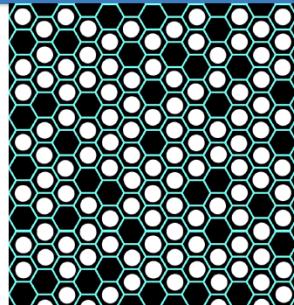
Figure 3.12

Identify the location of every cluster and create a map



www.lookandlearn.com/blog/5793/constellations-mapping-stars/

Start with a map of where clusters could be and overlay on images



[/black-light-studio.deviantart.com/art/Free-Hexagon-pattern-02-371945610](http://black-light-studio.deviantart.com/art/Free-Hexagon-pattern-02-371945610)

On patterned flow cells, the clusters' formation location could be known or not, in the first case it is said that there is a "Rigid registration" (figure 3.12).

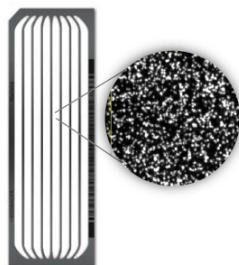
ILLUMINA sequencers normally use slides of glass, named flow cells, and the fragments to be sequenced are able to flow over the channels. The temperature inside the cells can be changed to produce ligation or separations. The surface of the cells is functionalized with a series of oligos complementary to library adapters.

Two kinds of flow cells, the *patterned* flow cell permits to create clusters in specific positions, inside nanowalls, contrarily to *Random Flow Cell* which instead have randomly positioned clusters.

Figure 3.13

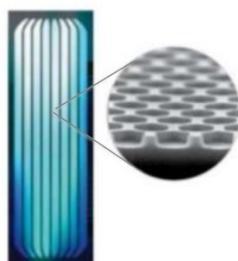
Random Flow Cell

- HiSeq™ 2500, MiSeq™, NextSeq™, MiniSeq™
- Randomly spaced clusters
- Variable Insert Sizes
- Lower Duplication Rates



Patterned Flow Cell

- HiSeq 3K/4K/X, NovaSeq™ 6000, iSeq™ 100
- Defined size and spacing
- Increased Cluster density
- Simplified imaging



3.3. NEXT GENERATION SEQUENCING NGS

Once the fragments are made flow over the chambers, they can bind only to p5 or p7 (ILLUMINA oligos), the two oligos functionalizing the plate. Once the fragments are attached to the surface, using temperature and solvents flows you can control the sequencing process. To see the entire procedure: (procedure on Youtube: video about ILLUMINA sequencing). In the video, it is shown the two index sequencing process.

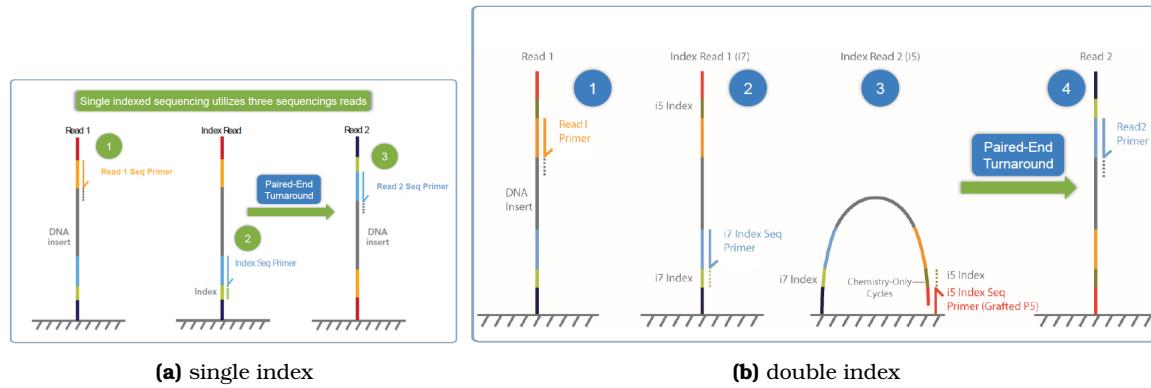
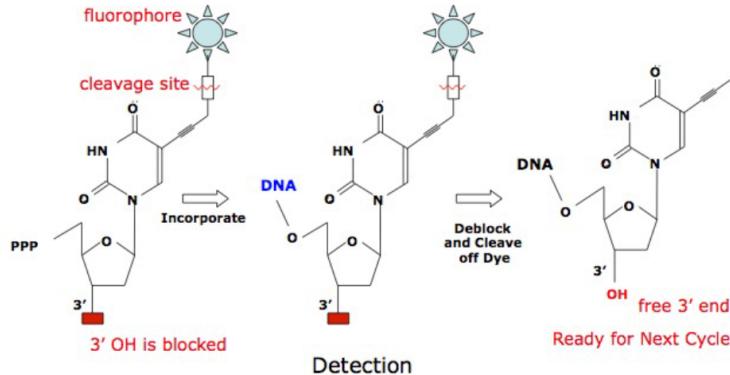


Figure 3.14: Single/double index for ILLUMINA sequencing

With 1 and 2 indexes the sequencing process actually differs, as shown in figure 3.14

Figure 3.15

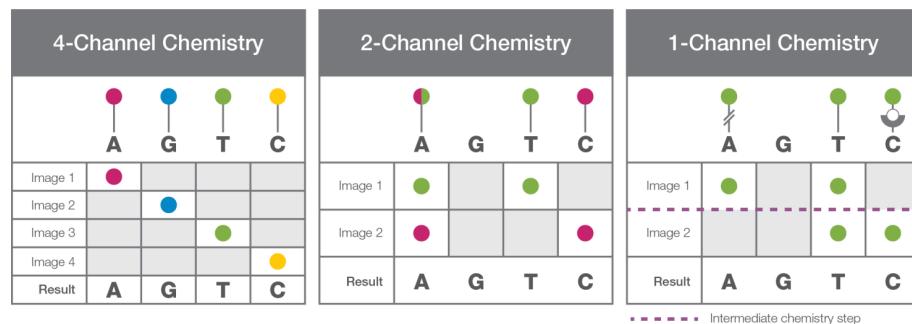


To perform the sequencing process, ILLUMINA machines utilize *reversible terminators* (figure 3.15). They permit a real time analysis of the sequencing by synthesis reaction, and because of this they are different from the Sanger method. Fluorophores are reversible, they can be cleaved to eliminate the light signal.

To perform their activity, ILLUMINA sequencers could be of 3 different types: 4-Channel, 2-Channel or 1-Channel, depending on the number of fluorescent molecules used. In the case of the 4-Channel technology, 4 images are taken in each cycle, and each cluster appears in only one of four images 3.16. 2-Channel technique is used by some sequencing

3.3. NEXT GENERATION SEQUENCING NGS

Figure 3.16



machines, like NextSeq 550, MiniSeq, NovaSeq 6000

4-Colors base calling is needed to make the true signal the purest, and after, The base with the highest intensity becomes the called base for that cluster. In the case no base is clearly related to a position, N is the result.

The reading process could be done in two ways: through single reads, on a single extreme of the fragments, or paired-end, on both the extremes. The second one in particular gives structural and 2 sequence informations.

3.3.3 Pacific Bioscience (PacBio)

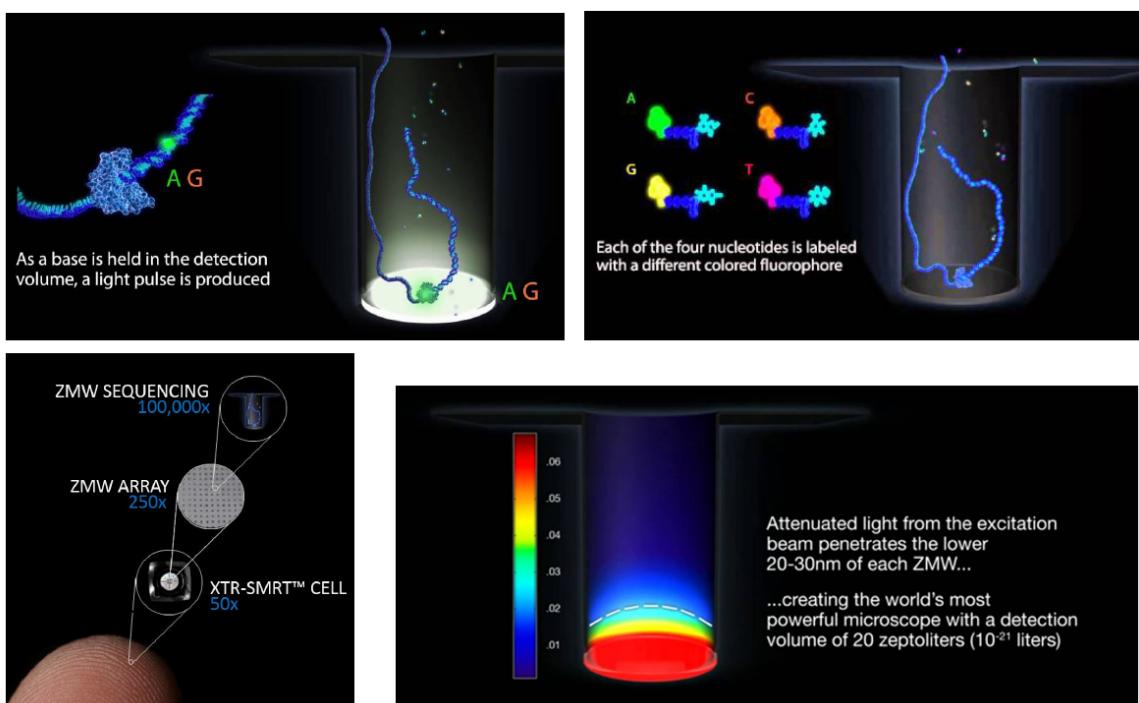
the long DNA filament to be sequenced is attached to a polymerase, over the surface of a SMRT (Single Molecule Real Time) cell. This cell is really small, and at each nucleation process a light signal is emitted. The produced light is not able to get out of the walls, and its duration is extremely restricted. Sequencing, also in this case, is made by sequencing, and the main advantage consists in the possibility of sequencing really big DNA molecules. In the video PacBio procedure it is briefly shown how the process works and what are the strategies used to reduce errors.

3.3.4 Nanopore sequencing

Intramembrane proteins are used, sequence detected through the passage of DNA nucleotides, which produce different voltage changes. During the first periods, this type of instruments gave great amount of errors, nowadays the technique is improving. Nanopore Sequencing

3.3. NEXT GENERATION SEQUENCING NGS

Figure 3.17



Chapter 4

Sequencing data

After all, genetics/genomics studies a code of a digital information (4bases/2bits). We should try to be as hypothesis driven as possible and use the already available and processed data to guide new data analysis. Be aware that, in genomics, data generation is the starting point of the study (the ratio of wet experiments vs computational effort is 1:10)! Given the biological problem at hand, we need to choose the optimal sequencing machine. To achieve this, we need to consider:

- Throughput;
- Cost;
- Read lengths,
- Data output (reads per run);
- Coverage;
- Sequencing errors (indel, substitution, CG deletion, AT bias). Although the error rate is decreasing with new technologies;
- Library preparation compatibility;
- Speed (run time)

Suppose that we want to sequence a genome of a bacterium: which is the best machine that we can use?

Illumina NovaSeq : if one wants to sequence a lot of DNA molecules at the same time, genomes, metagenomes. It can't go over the 300 bp readlengths runned, but it has the highest throughput so far (3TB of output). It is capable of multiplexing, so we have a unique barcode for any input sample.

Illumina iSeq : If you need to sequence shorter genomes. From iSeq to NextSeq (increasing the reads lengths).

NanoPore (minion) : pocket-sized wet-lab free sequencer for DNA, RNA and (possibly) proteins, but the read lengths is smaller than Illumina's. The machine is cheap; the running flow is more expensive (going down by time). It's a real-time sequencer.

4.1. INTRO TO SEQUENCING RAW DATA

PacBio : carries a high amount of error. A solution to this is using more than one technology for the same project. For example, if the goal is to put together a genome of a bacterium, the challenge of using PacBio is errors in the sequence, while Illumina is unable to reconstruct the sample because there are ambiguities. By using both technologies, PacBio will construct the genome and Illumina will correct the sequence. Anyway in the case of bacteria, you can use only PacBio because you have to sequence more and more, and since the PacBio error is random (not systematic), you will be able to overlap the reads and solve the errors. PacBio has problem with homopolymers (same base or same set of bases repeated multiple times), which is a systematic error, more difficult to overcome just by sequencing more. If one wants to sequence a complex sample - with more than one genome - we cannot do the trick of overlapping, because we don't know a priori which reads are coming from which organism. Another trick to work on this problem (complex samples), adopted by minION and also PacBio, is to sequence the same molecule multiple times. By doing this you will have a molecule sequenced multiple times, and then we can go on with the other reads, and in conclusion do the consensus on that (sequencing error is usually declared by the machine).

4.1 Intro to sequencing raw data

All sequencing platforms can produce FASTQ output files or they can at least convert their format to FASTQ. FASTQ contains the sequencing reads and the quality of each base and needs to be stored in a compressed format.

The first output is an internal output, not directly accessible by the user. You will see only the translation of the raw signal (FASTQ format). The internal raw signal can be completely different (ionTorrent measures pH, Illumina takes photos of light signals, nanoPore looks at a signal coming from a little hole, ...). However, all machines, internally, interpret this physical signal into a file which is in FASTQ format (the standard format).

Sequencing might lead to short (100nt) strings of few digital values (A T G C), puzzles of 3,000 M letters (solved then by assembling and mapping) or thousands of puzzles of 5 M up to 5,000 M letters, meaning 3 TB of letters that need to be made sense of. Furthermore, many pieces do not fit due to sequencing error/SNP/structural variant or low complexity region/microsatellite/repeat.

More on the FASTQ format will be covered in section 4.1.2.

4.1.1 Base caller

A base caller is an instrument (algorithm) inside the machine needed to understand how to work with the sequence, in particular, it translates the analogical signal of the reading into numbers/nucleotides. The most popular algorithm is **Phred**.

Phred is an algorithm based on ideas about what might go wrong in a sequencing reaction and in electrophoresis. It was tested on a huge dataset of 'gold standard' sequences (finished human and *C. elegans* sequences generated by highly-redundant sequencing). Its results were compared with the ABI base caller and Phred was considerably more accurate (40-50% fewer error).

4.1. INTRO TO SEQUENCING RAW DATA

NB: When things are getting too messy, the algorithm needs to understand that it is impossible to recover an high quality sequencing and so the base caller just needs to be able to give up for some reads (the ones without a high quality). The confidence that the base caller has to call a certain nucleotide ATCG is annotated in the FASTQ file, so one can later decide what to do with the bases that are not supported or clear.

4.1.1.1 Errors solved by Illumina's base caller

Phasing noise ϕ : when a certain base is not seen frequently, the first time in which you will see this specific base (or pair of bases) there will be a peak (a spike) in the graph, increasing the signal of the nearby bases. There are going to be some errors in estimating the real nucleotide that is occurring in the site. This problem can be solved by waiting more time between a reading and another, but the sequencing will be less efficient (slower) and the throughput will be lower. So the machine needs to find a trade-off between the efficiency and the clear reading.

Signal decay δ : after a while one has read the same base, or the same repeated couple of bases, the signal will go down and at some point will be indistinct (so at some point you will need to cut the read since it will be not trustable after a while).

Mixed cluster μ : when you have a cluster mixed in two sequences, the machine will read the signal resulting by this two strings simultaneously.

Boundary effects ω : in this case the machine needs to interpret an image (a digital object.. and a blue point will appear if you magnify it a little bit). To estimate the intensity of the blue charter here you should probably look at the internal part here and average out all the pixels. Putting the edge of the internal part is tricky, since it is a gradient and not clear. So this error occurs when you can't figure out what is the signal and what is the around background.

Cross-talk χ

Fluophore accumulation τ

4.1.1.2 Density on the flow cell

There is an optimal clustering in the flow cell and the presence of readings that go one through the other represent a problem. In the case of the dark yellow spots, the one that are too closer are going to be considered as a single peak. It is even more a problem if the spots are of different colours, and the signal will be a result of different bases.

In the case of under clustered the machine doesn't read simultaneously as many reads as it should do (and it is not a problem of the sequencing technology, it is a problem for the researcher who did not optimize the method). By increasing the flow' density you will have an optimal clustering. If you increase the density too much you will have and over clustered sequencing.

4.1.1.3 An ecology of base callers

With base callers the matter is finding a satisfying trade-off between accuracy and computational efficiency.

The quality associated to a single nucleotide is internally estimated by the machine. We need to check later if the quality of the read is right. This quality is the estimation of the probability that that nucleotide in that position is wrong.

There's a difference between PHRED score reported by base caller vs. PHRED score when mapped to a reference sequence. There are some base callers more calibrated than the others (like the Ibis). In general base callers need to be calibrated with a standard to make sure that the estimations of the quality are accurate enough.

4.1.2 FASTQ format

1. @SEQ_ID
 2. GATTGGGGTTCAAAGCACTATCGATCAAATAGTAAATCCATTGTTCAACTCACAGTT
x N 3.
 3. +
 4. !”*((((***+))%%%++)(%%%%).1***-+*)***55CCF»»CCCCCCC65
1. '@' followed by a sequence identifier. The sequence identifier contains the unique instrument name, the flowcell and tile number, the x and y coordinates of the cluster within the tile, the index number for multiplexing and the pair number of paired-end sequencing. In Illumina MiSeq, each flowcell has 8 microfluidic channels (lanes), each lane contains three columns with 96 tiles, i.e. 96 multiplexed samples.
 2. The sequence. It could be mate paired for paired end sequencing.
 3. '+', optionally followed by a sequence Identifier.
 4. The quality scores. Quality is a number based on the estimated probability of error. $p=\text{probability of error}$, $Q = -10\log_{10} p$. We need a base quality of at least 20 to reach a small probability (1%) [bq=40, p=0.01%]. The FASTQ quality score is the phred score +33, converted in CHAR code.

The FASTA file format is a FASTQ fomrat without the quality score reported and the seq ID is preceded by '>'.

4.1.2.1 Quality control: read length distribution

Quality scores are used typically for the operations of quality control and/or quality cleaning of the reads, which are then saved in FASTA files and used for other operations. However, since there exist algorithms, e.g. mappers, that take the reads and map them against a reference genome that can take into account the quality into a FASTQ directly.

Looking at the qualities will allow you to judge how well the sequencing run went. In this case, for example, the read length is used as quality control. In the second

4.1. INTRO TO SEQUENCING RAW DATA

graph we can see that something went wrong, there are too many short reads. In this case you can use the sequencing results, but make sure to get rid of short reads when you do another sequencing, to have an optimal use of the machine.

Another way to look at your sequencing output is to look at the quality score directly. Here the quality goes down as the length of the reads increases. This is typical, since the machines have more and more problems in estimating the nucleotides when in continuum. In this case you can only cut the last results since the quality is too low. Another problem that can occur is the adapter content. Sometimes it happens that you see the adapter, which should not be part of the sequencing reads, since the machine should be able to identify and remove it. This happens frequently when you have very short fragments in your output. After the machine has sequenced the sample, it starts to sequence also the adapter (if you don't identify the end of the sequencing you will have the adapter in the sequencing output). The main solution is to remove the adapter and what follows it (but be aware that you will remove even the part of the sequence in the middle).

We can use FastQC to plot the quality distribution of our data. FastQC on Nanopore data: the reads are really long, but the quality is low (expected result). On PacBio data we also see the distribution falling entirely in the red zone, but here we have the box plot in the central part.

We can also plot differently the PacBio data. Here we still have PacBio, but in a different view. Instead of looking at each position separately, here we look at the average quality for the whole read. Probably you will discard the low-quality reads, like 1-2 score. When thinking at what technology use, think also at the length of the genome you want to sequence!

4.1.2.2 FastQC on PacBio vs ONT

The analyses were performed using long-read and short-read data from *E. coli* strain K12 sub-strains DH5 α and MG16. Substitutions were more frequent in ONT, while insertions were more frequent in PacBio. On nanopore A>G, T>C, C>T and G>A are 3 times more frequent than the other, while in PacBio the frequencies are more alike.

k-mer frequencies: we know the frequency in the reference, so we assume that if the sequencing is systematic, without biases and without errors we should have the same frequency. In this example, it's clear that nanopore has done a better job.

(By looking at 6-mer you have a larger potential combination, but it is easier to find errors in it). Moreover, one can look at which are the over-represented and the under-represented k-mers in the reads compared to the genome, and this may give you an idea about where the errors are. Both in nanopore and PacBio, the most troubling k-mers are those containing the same repeated nucleotide (this is a frequent sequencing error).

This is interesting because the low entropy of the repeated sequences can be detected by the quality control software and are usually discarded. We will see that sometimes it is not the right thing to do, since in non bacterial genomes there are repeated sequences (consideration to take when you decide which technology use for your experiment).

4.1.3 Duplication artifacts

It is not frequent to see duplication, but it can be a problem. If you want to reconstruct a genome from a scratch of a bacterium you don't really care whether if there are duplicates or not. But if you want to quantify the gene expression or the copy number of genes in a bacterial genome, this can really skew your analysis. The distribution of the duplicates should be very similar to the distribution of the reads. There shouldn't be any bias along the length of the reads. If you see something like this, you probably sequenced the same thing over and over again, and so something went wrong. This happens when you sequence the adapter or the primer (they are always the same sequence).

4.1.3.1 GC content analysis

We know that each single organism has a signature GC content. We expect to see a peak and a normal (gaussian) distribution. Therefor, if two peaks are seen, it means that we have sequenced two different organisms (maybe, a contamination). The problem is that these two organisms may share some sequences.

4.1.3.2 K-mers frequency plot

Frequent k-mers can be a signature, as the GC content. For example: 15-mer coverage model fit to 76x coverage of 36bp reads from *E. coli*. The expected coverage of a k-mer with reads of length L:

$$L_{cov} = \frac{L - k + 1}{L} \times Cov \quad (4.1)$$

Than, given a k-mer, we can see in how many reads that k-mer was present. And in this case the coverage is around 40%. However, there is a huge amount of k-mers present with coverage 1 (i.e. that you have seen only once in your reads). So, there are errors in that specific k-mers (remove these results). Instead, coverage higher than 80% would mean that the k-mers are located in more than one position.

4.1.3.3 Low-complexity artifacts

Same nucleotide repeats (especially A) are in a lot of cases artifacts. It's an error of the machine and the quality control looking at the quality scores, that in this case are very high.

How to measure their 'artificiality'? Some parameters to take into consideration are:

- Low complexity
- Low entropy
- High compression (the artifacts increase the information inside the file)

But some low-complexity sequences are not artifacts!

- Hydrophobic transmembrane alpha-helical sequences in membrane proteins

4.1. INTRO TO SEQUENCING RAW DATA

- CAG repeats in genes causing Huntington disease, spinal and bulbar muscular atrophy, dentatorubropallidoluysian atrophy.
- Proline-rich regions in proteins
- Poly-A tails in nucleotide sequence
- Micro-satellites

4.1.4 FASTQ quality control (QC)

This is the first step (or sub-pipeline) in any NGS pipeline

Clipping/trimming : removing (low quality) parts of reads

Masking : avoiding to consider (e.g. low entropy) parts of reads

Read removal : discard low quality reads or reads that are too short after clipping

Additional features that can be exploited for QC are the GC content, clustering for contamination detection, TAG identification, ambiguous bases.

Chapter 5

Mapping

Mapping (or aligning) and assembly are the operations that allow us to make some sense out of fragments (input data) deriving from sequencing, produce assemblies.

- **Mapping** is a key step in a modern genomic analysis and consists in the process of aligning the reads on a reference genome, in order to assign them to a specific location. With mapping, insights like the expression level of genes can be gained.
- **Assembly** by contrast, is the process of aligning and merging overlapping sequences in longer consensus sequences in order to reconstruct the original sequence/genome.

A consensus sequence is the calculated order of most frequent residues, either nucleotide or amino acid, found at each position in a sequence alignment. In many cases, someone may have already assembled the genome or part of the genome (available reference sequences), so we don't need to do sequence assembly, only mapping (like for the human genome). Assembly will be needed however when studying new organisms.

Problems related to mapping and assembly:

- absence of DNA fragments covering the gaps, makes it difficult to order the contigs (since there is no connection)
- presence of DNA artefacts (those must be discriminated with Quality Control)
- repeated sequences

5.1 Mapping

The coverage, or read depth, is the average number of reads representing a given nucleotide in the reconstructed sequence (Some points of the reference sequence will be aligned with more reads, some with less. The average number of reads for each nucleotide is the coverage). The coverage of a genome is defined as the average coverage of each single nucleotide across all nucleotides of the genome. A coverage of 1x does not mean that all reads are read once. This would be true if sampling were systematic, but sampling is not systematic, it is random and is biased. Hence, a coverage of 1x means that on average each nucleotide is covered once, but there will be some nucleotides covered more and some not covered (0

5.2. MAPPING ALGORITHM

coverage). The coverage can be represented with a coverage map and can also be defined theoretically as:

$$P_{\text{val}} = K e^{-\lambda S} = \text{Evalue}/mn$$

where G is the length of the genome; N is the number of reads; L is the average read length. Knowing the wanted coverage is important to set up the machine in order to obtain the right number of reads (to achieve a certain coverage). A higher coverage means indeed more reads to be obtained.

5.1.0.1 Exercise on coverage

How many reads do I need to cover the entire genome? Define the probability to cover the full genome (with a certain epsilon subtracted - so not 100%) or: define the number of reads N to cover the whole genome with a certain probability $P = 1 - \epsilon$.

A bit more on the coverage

Suppose:

- I want to sequence a genome which I know is ~3M bases
- I can use Illumina single-end sequencing with read length of 100 nt

How many reads do I need to cover the entire genome (i.e. no holes) ?

Average coverage needed to cover the entire genome with probability $1 - \epsilon$

$$\text{Cov} = \frac{N \cdot L}{G} \approx \ln \frac{G}{L \cdot \epsilon}$$

With $\epsilon = 0.05$ the needed coverage is ~13

Number N of reads needed to cover the entire genome with probability $1 - \epsilon$

$$N \approx \frac{G}{L} \ln \frac{G}{L \cdot \epsilon}$$

G: length of the genome
L: read length
→ ~400k reads

Figure 5.1

In general, the sequence mapping process consists in performing comparisons between experimental sequence data (reads obtained with sequencing) with some reference information, like reference genomes and known genes (eg. the human genome). The comparison can lead to obtaining new information, such as the presence of SNPs which can be linked to pathological conditions or whatsoever, based on the starting point.

5.2 Mapping algorithm

Over time many different mapping algorithms were implemented; some of them are not used anymore, while others are the base of current mapping and aligning genomic tools. Ideally, the **simplest aligning algorithm** could consist in: We have a smaller sequence that we want to align to a longer one. Start from the first position, align the query sequence against the subject, and look at how many nucleotides are correct (score: 4/10 = 40%),

5.2. MAPPING ALGORITHM

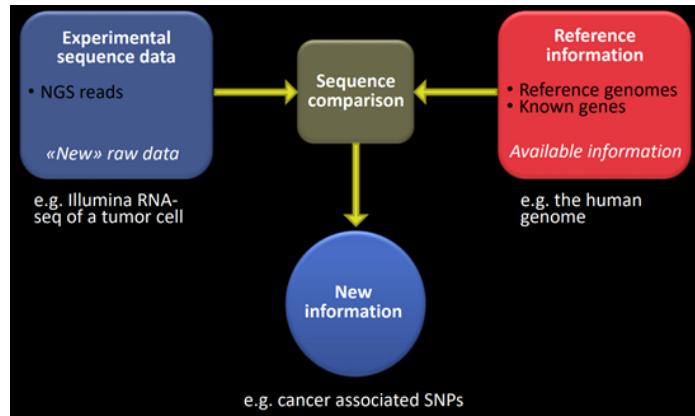


Figure 5.2

then repeat for all positions until a perfect match is found (if found). The problem with this algorithm is that it doesn't consider insertions and deletions.

5.2.1 Local vs Global alignment

Sequence alignment can follow two different approaches.

1. In **global alignment** an attempt is made to align completely the 2 sequences (end to end alignment). So global alignment finds the best alignment across the whole two sequences. This approach is suitable for comparing closely related sequences like homologous genes.
2. **Local alignment**, on the other hand, focuses on finding regions of similarity in parts of the sequences (so it aligns subsequences of the query sequence to a subsequence of the target sequence). This approach is suitable for aligning more divergent sequences or distantly related sequences. Used for example for finding out conserved patterns in DNA sequences of motifs in two proteins.

Sequence similarity is connected with **evolutionary distance** (also for cell populations in a tumour). Very high similarity implies a very low distance. But when the similarity goes down and reaches the twilight zone, it is more difficult to define the evolutionary distance and to give meaning to results (the zone depends on the experiments).

5.2.2 Smith-Waterman algorithm (local alignment) - 1981

Take the algorithm for string recognition and apply it to find common molecular sequences. Not good for indels and substitutions. The S-W Algorithm is a local-alignment algorithm based on dynamic programming, whose aim is to find the best match among all possible (optimal local alignment) with respect to the scoring system used. Firstly, we need to define a **score** for matches and penalties for mismatches and gaps (defined by the formula). The algorithm starts by putting zeros in the first row/column (**INITIALIZATION**). Then, starting from the first position H(1,1), at each iteration a number, based on the formula,

5.2. MAPPING ALGORITHM

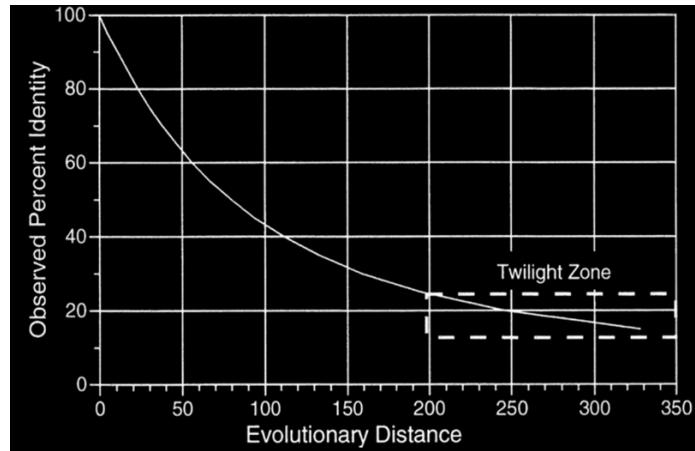


Figure 5.3

is added (**ITERATION**). The number to be added is the greater between the 4 numbers defined by the formula, where d represents the penalty for gaps, and $s(x,y) = \text{score}$ when 2 NTs are the same (This gives a score of 5 if the 2 bases are equal, -3 if they are different) (**PARAMETER SETTING**).

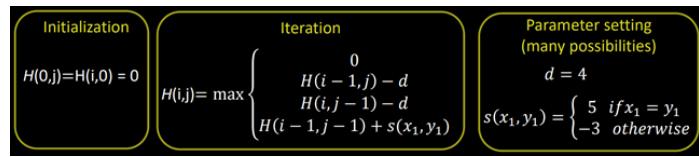


Figure 5.4

Example

First position $H(1, 1) = (\text{G}, \text{C})$, max between:

- 0
- $H(i-1, j) = 0 - 4 = -4$
- $H(i, j-1) = 0 - 4 = -4$
- $H(\text{diag}) + s(x, y) = 0 - 3 = -3$
- \rightarrow We put 0

Second position $H(2, 1) = (\text{A}, \text{G})$:

- -0
- -4
- -4
- -3

5.2. MAPPING ALGORITHM

- → We put 0

Third position H(3,1) = (C e C):

- - 0
- -4
- -4
- 0 + 5 (since C=C)
- → We put 5

Here we also put an arrow indicating the nucleotide in the diagonal, to indicate that the 5 score derived from that point (hence from that path). The arrows are needed to reconstruct the alignment at the end of the process. The iteration is performed for each cell of the table (column wise). The parameters chosen for d (gap penalty) and s(x,y) (match or mismatch) can change the results. Final Solution:

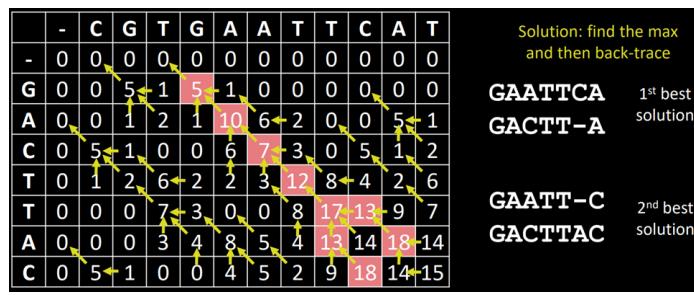


Figure 5.5: Aligning CGTGAAATTCA and GACTTAC

Find the **maximum number** (18, here we have 2 of them) and go back following the arrows. In case of multiple highest scores, traceback should be done starting with each highest score. From this path, the sequence is constructed by these rules:

- A diagonal arrow represents a match or mismatch, so the letter of the column and the letter of the row of the origin cell will align.
- A horizontal or vertical arrow represents an indel. Vertical arrows will align a gap ("") to the letter of the row (the "side" sequence), horizontal arrows will align a gap to the letter of the column (the "top" sequence).
- If there are multiple arrows to choose from, they represent a branching of the alignments. If two or more branches all belong to paths from the bottom right to the top left cell, they are equally viable alignments. In this case, note the paths as separate alignment candidates.

This algorithm is not used anymore due to a too high computational expense (huge number of comparisons for real genomes) and storage (memory and speed). In fact comparing two Eukaryotic genomes will result in a excessively great aligning matrix ($\sim 3\text{Gb} \times \sim 3\text{Gb}$), and also using small reads against the human genome is unfeasible: 100 $\times \sim 3\text{Gb}$ for each read!

5.2.3 Needleman-Wunsch algorithm (global alignment)

This global alignment algorithm was developed in 1970 and is also based on dynamic programming. The purpose of the algorithm is to find all possible alignments having the highest score. Again, a scoring system must be defined, then the algorithm proceeds in a similar way as the SW.

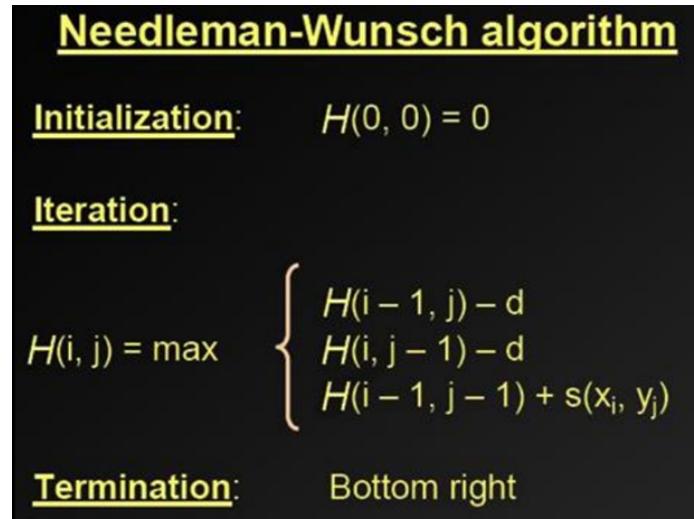


Figure 5.6

The differences are:

- Initialization: we start from only one value, one zero, putted in $H(0,0)$ (need to start from the first nucleotide - global).
- The formula contains no 0 anymore, it adds the maximum score between 3 possible scores, we are always comparing with the first nucleotide of the sequence.

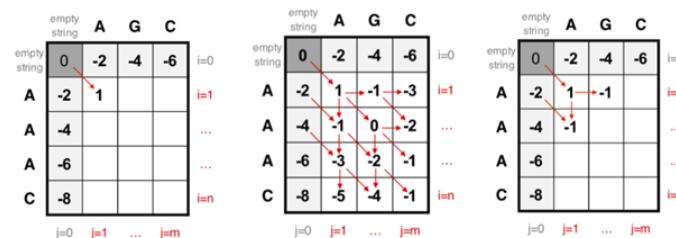


Figure 5.7

New algorithms were implemented later. BWA and BowTie2 are the best ones available right now for short reads. Blast could go too but it would take a lot of time.

The methods going from FastA to BowTie2 are **heuristic methods**: the solution found does not need to be the best one, but the best approximation given certain parameters

5.2. MAPPING ALGORITHM

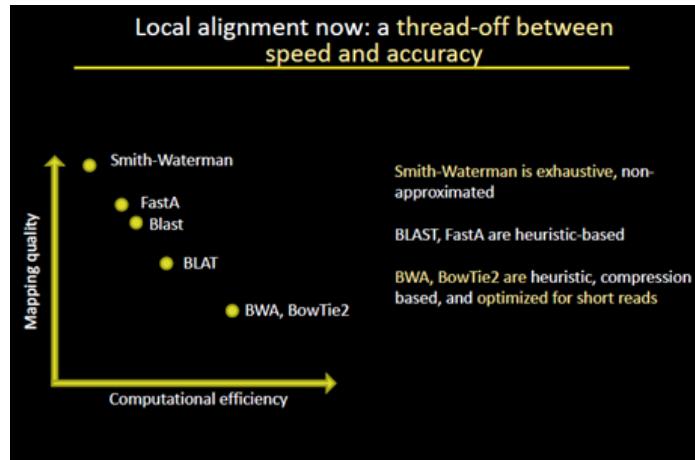


Figure 5.8

(like time). Also, Blast gives an answer based on a fixed level of sequence similarity (in a reasonable amount of time) and gives no result under that level. So Blast online is not very sensitive, it is trained to provide only very good results to not overload the servers. A **heuristic**, or heuristic technique, is any approach to problem solving or self-discovery that employs a practical method that is not guaranteed to be optimal, perfect, or rational, but is nevertheless sufficient for reaching an immediate, short-term goal or approximation. Where finding an optimal solution is impossible or impractical, heuristic methods can be used to speed up the process of finding a satisfactory solution. Heuristics can be mental shortcuts that ease the cognitive load of making a decision. BWA and BowTie2 are also ‘compression based’ algorithms, meaning that they exploit data compression techniques in order to compress reference genome files to reduce the number of bits used to encode the document (better explained below).

5.2.4 BLAST (Basic Local Alignment Search Tool)

Despite its limitations, Blast is still used a lot and when it was first released it revolutionised the field making sequence alignment available for anyone and for any computer (it can be run online). How does it work? Steps:

1. **Seeding:** find perfect or almost exact k-mer matches (series of sequences with defined length). The idea is to look for identical short matches and try to expand from that.
2. **Extension:** extend the seeds at point one 1 with possibly some non-exact but high-score matches, that permit to obtain better alignments.
3. **Evaluation:** create alignments for the regions of high-scoring extended seeds. Every time the statistical significance of the match is evaluated with methods inspired on the NW and SW approaches.

In online Blast tools the seed is set to 25/29. If a sequence of that length does not match perfectly another sequence we will have no results. One solution would be to lower

5.2. MAPPING ALGORITHM

the seed, but this cannot be done online because it would take too much time. Another option is to reduce the dataset in which we search. There are many types of blast available, shown in figure:

Blast comes with many flavours	
Program	Description
blastp	Compares an amino acid query sequence against a protein sequence database.
blastn	Compares a nucleotide query sequence against a nucleotide sequence database.
blastx	Compares a nucleotide query sequence translated in all reading frames against a protein sequence database. You could use this option to find potential translation products of an unknown nucleotide sequence.
tblastn	Compares a protein query sequence against a nucleotide sequence database dynamically translated in all reading frames.
tblastx	Compares the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database.

Program	Query	Database (Subject)
BLASTN	Nucleotide	Nucleotide
BLASTP	Protein	Protein
BLASTX	Nucleotide -> Protein	Protein
TBLASTN	Protein	Nucleotide -> Protein
TBLASTX	Nucleotide -> Protein	Nucleotide -> Protein

Figure 5.9

Scoring matrices are very important especially for amino acids, they are used to score alignments between protein sequences. Blosum 62 (BLOcks SUbstitution Matrix) is one of the most used. They put non simple penalties in substitutions of different amino acids, for example based on the functional properties of the substitutions. A scoring matrix contains values proportional to the probability that amino acid i mutates into amino acid j for all pairs of amino acids. Such matrices are constructed by assembling a large and diverse sample of verified pairwise alignments of protein sequences. Other **parameters** can be set in online BLAST:

- Max target sequences → number of reported sequences
- Expected threshold → e-value
- Words size → seed
- Gap cost → cost for adding multiple gaps. Linear = twice the gap penalty (if I have to add a second gap after another one).
- Filter low complexity regions to avoid getting stuck

5.2.4.1 BLAST E-value

If we have a match and the database is random, how many other matches I will likely find?

The *e-value* represents the number of distinct alignments with a score equivalent to or higher than S, that are expected to occur in a database search by chance. An e-value of 10 means that up to 10 alignments can be expected to be found just by chance, given the same size of a random database. E-value can be used as a first quality filter for the BLAST

5.2. MAPPING ALGORITHM

$$Evalue = Kmn e^{-lambda S}$$

- Parameters K and lambda depend on the substitution matrix and on the gap penalties
- n is the query lenght
- m is the lenght of the sequences of the database
- S is the matching score

Figure 5.10

search result, to obtain only results equal to or better than the number given by the e-value option. Blast results are sorted by E-value by default (best hit in first line). The smaller the e-value, the better is the match. A small e-value means a low number of hits, but of high quality, whereas a high e-value indicates many hits, partly of low quality (if the number of better possible alignments is high, it means that it is particularly probable to find by chance an alignment which is better than the one found, and so it is more probable that those alignments are bad). There is a relationship between the E-value and the p-value.

- The E-value is the number of sequences that we would find by chance;
- The p-value measures the probability of finding by chance another sequence with an equal or better score.

$$Pval = K e^{-lambda S} = Evalue/mn$$

5.2.5 Speed seed alignment

There are algorithms implemented 10/15 years ago that focus on seeds -> not used anymore. Those methods cut both the reads and the reference sequence (read-sized pieces of reference sequence) into small seeds. The reference seeds are then stored in an index (hash look-up table). The idea is to do that for all possible k-mers. Eg. we choose a seed of 10: the first starts at position 1, the second at position 2, ecc. And then to look up seeds for each read and identify the positions in genome where spaced seed pair is found.

- 1 SNPs means that at most 1 seed is not-matching.
- X SNPs means that at most X seeds are not-matching.

Algorithms based on this approach are: Maq, SOAP, MOSAIK. Problem: the lookup table (the reference) for the human genome is too big: 50Gb → problem in RAM.

5.2.6 Burrow-Wheeler alignment

This algorithm is based on a very efficient way to store the reference genome, based on the BW transformation, that allows to have a reference index which is way lighter (with respect to the spaced seed approach); in fact, at the end of the transformations, equal letters are next to each other. It is successful because the database is compressed and does not need to be decompressed. However, when we compress a file, the compression must be reversible. There must be a way to go from the BW compressed/transformed index to the original one, to find reads in the genome. The search is based on finding suffixes of the reads in the BW structure. With this approach the index of the human genome is around

5.2. MAPPING ALGORITHM

2GB. The algorithms BW-based are the fastest currently available. BWT is also used in text compression. Example: We want to compress the word BANANA

- - we take all the rotations of the word
- we sort them based on lexicographic order (Lex order - lexicographical order is a generalization of the alphabetical order of the dictionaries to sequences of ordered symbols)
- we take the last column

We end up with a string which is more compressible, there are sequences of the same letters that can be described at a higher level (Eg. there is 1 B, 2 NN, ecc) In the method shown, in order to **reverse the transformation** and go back to the original word, the input sequence (output of the compression) is added as a column and then sorted (based in lex order), and this two passages are repeated as many times as are the characters of the sequence (8 in this case), obtaining at the end a matrix with equal number of columns and rows. As a result we go back to the ordered sequence of the word's rotations, in which the last one (last line) is the original input string. It takes a lot of time but it makes it reversible; plus, there are actually smarter ways to go back, such as ones base on the LF property.

5.2.7 LF (Last-First) property

As in the previous method, the first step consists in sorting all word's rotations and taking the last column which will be the compressible one. The LF property is then applied to the list of all rotations, to get back to the input sequence, by following this principle: the *i*th occurrence of character X in the last column corresponds to the same text character as the *i*th occurrence of X in the last column. Example: We want to reverse the transformation of the string "gc\$aaac".



Figure 5.11

- *g* is the first occurrence of *g* in the last column and corresponds to (arrow) the first occurrence of *g* in the first column. Tracing a horizontal arrow gives the next nucleotide.
- *c* is the second occurrence in the second column and corresponds to (arrow) the second occurrence of *c* in the first column. Tracing an horizontal arrow gives the next nucleotide.

5.2. MAPPING ALGORITHM

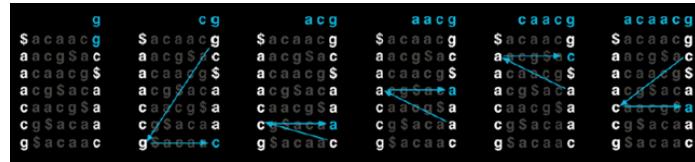


Figure 5.12

- a is the third occurrence of a in the second column and corresponds (arrow) to the third occurrence of a in the first column. Tracing an horizontal arrow gives the next nucleotide.

And so on. Each time we keep track of the character that we are looking at and we end up by constructing the original word. With this method we reconstruct the word from its end to its beginning (backward).

5.2.8 Exact mapping using LF property

In mapping, we can look for sequence matches using the compressed database. Backward exact mapping works by calculating the range of matrix rows beginning with successively longer suffixes of the query. Example: we want to match the string 'aac' with the database.



Figure 5.13

- we take the 'first suffix' **c** of the query string and we look for it in the first column. We find 2 matches in lines 5 and 6, so in the range 5-7.
- we pass to a 'second longer suffix' **ac** and see if we can find it in the matrix → lines 3 and 4, range 3-5.
- we look at the suffix **aac** and find it in the 2-3 range.

Doing the backward matching we exploit the characteristics of the BWT transform. This same approach could be used to find a match between a query read and a reference genome which has been compressed using the BW alignment. The problem is that it does not take into accounts indels, mismatches. A possible solution to this problem could be the **inexact mapping**. Perform exact mapping and, if the query is not found, go back and perform backtracking by hypothesising mismatches. For example we want to find a

5.2. MAPPING ALGORITHM

match for the query GGTA. As always, we start looking for a match starting from the last character.

- we find a match for a, t, g but not for the last g.
- so we go back and hypothesize a mismatch for the first letter a. We then look for all the matches that we could find if we replace a with another nucleotide (we do the same for the other NTs in the query?).
- we find than by replacing a with a g, we find a match (ggtg) (not exact match).

The Burrows-Wheeler algorithm is used in BowTie and Bowtie2. Right now there are many algorithms to choose between and many review papers compare their characteristics and performance. No tool outperforms the others in all the test, hence the decision of which algorithm to choose must depends on the specific needs.

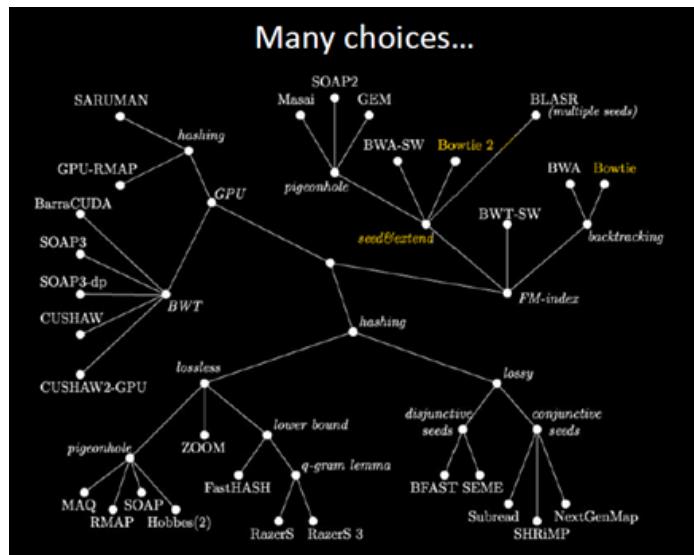


Figure 5.14

Chapter 6

Assembly

Assembly is the process of **aligning and merging overlapping sequences** in longer consensus sequences to reconstruct an original sequence or genome. No reference databases are used to perform assembly, unlike in mapping. Assembly is the process of aligning and merging overlapping sequences in longer consensus sequences to reconstruct an original sequence or genome. No reference databases are used to perform assembly, unlike in mapping.

- sequencing a genome for the first time (eg. the genome of a new bacterium);
- when the reference genome is not complete or very distant phylogenetically and hence not usable as a reference;
- in case of new genes, which cannot be discovered just by mapping.

With human genetics assembly is rarely needed, since the human genome is already available. By contrast, in microbial genomics it is particularly important. Different strains of *E. coli* for example could have new genes, that cannot be discovered by mapping against a reference genome. Applied also for viruses or yeasts. The mechanism of assembly is quite complicated. In theory, all assembly algorithms could be based on this basic framework

1. They start from the reads obtained through sequencing and identify overlaps between them (this is performed by all algorithms).
2. Once found, the order of the reads is defined by connecting reads by overlaps.

In practice this is not always possible, due to many issues that occur, like:

- multiple overlapping: there will be multiple overlapping of the same reads. Sometimes this is just due to a high coverage, sometimes (frequently) there may be different locations in the genomes that are matching, at least partially, our reads.
- partial matches, ecc.

One possible solution would be to **keep track of all possible matches** (overlaps) and then try to reconstruct the sequence. One way to represent the possible overlaps between reads is by using a **graph** in which the nodes represent the reads and the edges

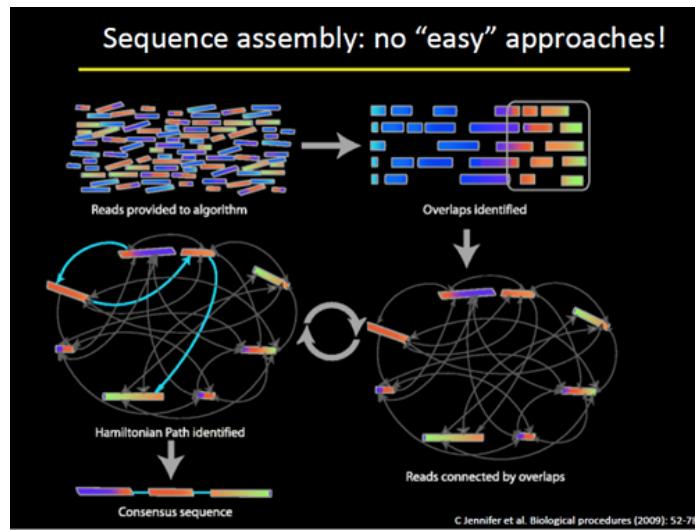


Figure 6.1

represent the connections between the reads that are overlapping in some way. Then, a **mathematical algorithm** is needed to find a solution in this very intricate network. Many assembly algorithms are available, and they are all based on this idea. For this reason, we always end up with multiple copies of genomes, not just one. Also, the output is almost never the full genome, but pieces of the genome, **multiple contigs**. Ideally, we could close the genome, but this is not possible just with short-read sequencing; other experiments are needed. By default, assembly is done in this way and gives a good representation of the genome, but not complete. Some quality control is also needed, especially to detect contamination.

The **general pipeline** for sequence assembly consists in a few steps:

1. Find overlapping reads.
2. Merge the 'good pairs' of reads into longer **contigs**. A contig is a contiguous sequence formed by several overlapping reads with no gaps.
3. Link contigs to form **scaffolds** (or supercontigs), which are ordered and oriented sets of contigs. Scaffold assembly usually exploits mate pairs (which allows to link different contigs together, even if the information in the middle is not completely known). Mate pairs, also called long-insert paired-end reads (LIPERs), are a kind of read obtained from paired-end sequencing which can pair reads across greater distances. Scaffolds do contain gaps, but at least the order of the contigs is known and helps to reconstruct the whole genome.
4. Derive consensus sequence. Sometimes we have multiple scaffolds and the consensus sequence is the set of all scaffolds representing the genome or the chromosome that is being reconstructed.

Finding the order of contigs is not easy and different approaches are used, hence scaffolding is usually a quite time-consuming operation. The aim would be to try to close the

6.1. FEASIBILITY OF SEQUENCE ASSEMBLY

genome; for this purpose, some additional experiments could be needed. For example, one could use PCR to try to clone pieces between two contigs. In microbial genomics, assembly usually finishes at the contig step. Contigs contained in the output files allow to define the genes present in the genome (for this goal, the order is not fundamental), whereas the position of genes in the genome is more difficult to study. **What are mate pairs?** - Mate pair library preparation process Following DNA fragmentation, the DNA fragments are end-repaired with labeled dNTPs. The DNA fragments are **circularized**, and non-circularized DNA is removed by digestion. Circular DNA is fragmented, and the labeled fragments (corresponding to the ends of the original DNA ligated together) are affinity-purified. Purified fragments are end-repaired and ligated to Illumina paired-end sequencing adapters. Additional sequences complementary to the flow cell oligonucleotides are added to the adapter sequence with tailed PCR primers. The final prepared libraries consist of short fragments made up of two DNA segments that were originally separated by several kilobases. These libraries are ready for paired-end cluster generation, followed by sequencing utilizing an Illumina next-generation sequencing (NGS) system. Combining data generated from mate pair library sequencing with that from short-insert paired-end reads provides a powerful combination of read lengths for maximal sequencing coverage across the genome.

6.1 Feasibility of sequence assembly

Doing a full assembly is considered to be impossible from a computational point of view. It is completely infeasible (NP hard), cannot be finished in a reasonable time, regardless the computational resources available. So technically assembly is impossible, differently from mapping, which is feasible. Read length → important for reconstructing the whole sequence Coverage → many reads but impossible to assemble.

6.1.1 Last year exercise

As an example of the assembly process and of the aspects that might be important, students were asked to reconstruct a piece of English text which was previously fragmented.

	Group 1	Group 2	Group 3
Coverage	5	5	4
Read length	10 PE	10	14
Mut. Rate (every 1k)	20	20	50
Number of corr. words	20/120	21/120	73/120
Number of sentences	2	3	7
Reconstruction perc.	16.6	17.5	60.8
Time (mins)	45	45	45

Figure 6.2

Group 3 was the winner (highest percentage of reconstruction), despite having a lower coverage and a higher error rate, indicating that the length of the reads was very important. Without long-enough reads it was impossible to assemble sentences, going out of the repetitive regions of the text. Knowing the characteristics of the sequence reads ('the language') might also help assembly. Eg. knowing that some sequences are just artifacts.

6.1.2 Merging overlapping reads

Basic principle: the stronger the similarity between the end of one read and the beginning of another the highest the likelihood the reads are coming from the same overlapping sequence region. When overlapping reads, we must maximize the length of the overlap but also accuracy. Of course, since sequencing is not perfect, errors will happen. So again, we must find a score for the overlap. The fact that two reads are not perfectly overlapping can be due to different reasons:

- they are not really contiguous (just similar sequences inside the genome). We should be able to distinguish between the two (not easy).
- Sequencing error/noise
- Diploid genome or multiple copy genomes: sometimes sequences coming from 2 different chromosomes; they are similar but not contiguous and this increases the difficulty.

6.1.3 Overlap graphs

As said, one way to represent reads overlaps is by using graphs. In an overlap graph, each read is a node, each overlap is an edge between two nodes, representing the two overlapping reads. Edges are directed: indicating that the final part of the first sequence matches the initial part of the second sequence. Directed graphs are very common in biology, for example to describe binding between proteins, or other biological functions. Different overlaps can have different "strengths", based on the score of the overlap, so to each node we can also associate a weight. Graphs are ideally cyclic, since in theory most organisms' DNAs are circular. So we could have the last read matching the first one. However, sometimes cycles are just due to random overlapping or to the presence of repeats. Those problems must be resolved in the mathematical object. **The problem of the repeats** is a key aspect in bacterial genomics since bacterial genomes have many regions with repeats. Also, as we previously saw, short reads are the default sequencing technology for bacteria, and this increases the difficulty in assembly.

If the read length is not long at least the length of the repeat, we will never be able to resolve this. The read that matches a part of the repeat and a part of the non-repeated region will be known, but for all reads that are inside the repeated region it will be impossible to define how long these regions are. There will be a lot of cycles and connecting the 2 non repeated parts won't be possible. By knowing the **coverage** we can try to solve at least partially the problem. If we know that all the genes have a coverage of 10x for example, we could assume that the repeated part has a 10x coverage too and we could define a length for this region proportional to the expected coverage. This is however a bit risky: repeated regions are also problematic for the sequencing machine, so there could be a skew in coverage for these regions. The **safest idea** would be to **stop the contig** before the repetitive region (A) and start it again after it. Some assemblers will try to solve these repetitive regions and might produce a wrong contig. Hence, if we try to minimize the number of contigs produced (to obtain longer genome sequences) we might end up having wrong contigs (without even knowing it). **Long reads** are the ultimate solution for this problem (they have many errors but represent a backbone on which to do the assembly). With overlap graphs, which are not what assemblers are using right now, we can

6.2. HOW TO SOLVE AN OVERLAPPING GRAPH?

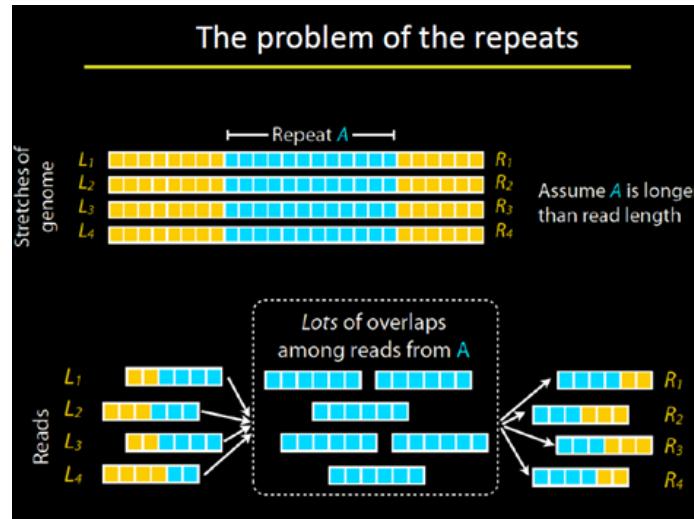


Figure 6.3: Example of genome with repeats

lead to having wrong contigs. The sides of regions containing repeats (mix of yellow and light-blue seen before) will overlap pretty good and will be considered as consequential in the sequence, discarding the repeated sequence in the middle. This will produce a wrong contig, in which fragments are considered to be **closer** than they are in reality. A solution could be to put many Ns in between and to keep track of the fact that the sequence is not reliable. Some hints of the presence of repeated regions can be given by:

- Coverage: we could see a region with a much higher coverage, which might contain actually repeats that were wrongly assembled.
- Paired-end: we could notice that the pairs over repeated regions have a shorter insert size than they should have. With long paired end sequencing we could also try to estimate the length of the regions with repeats.

6.2 How to solve an overlapping graph?

In this graph we have five known reads. Overlaps of length 1 are also present, usually not considered. A possible ideal solution would be to use **Hamiltonian paths**. Hamiltonian paths are the paths touching all connected nodes once, so they provide all possible solutions. Then, how do we define which is the best one? We choose the one that maximizes the length of the overlaps. This problem however will explode, it implies too many possibilities. It is equivalent to the shortest common superstring (SCS) problem, and it is NP-hard = not feasible. Another solution what be to use a **greedy approach**, which maximizes each choice, by following these steps:

1. Randomly select a starting point/node.
2. Select the connected node with maximum overlap as the next visitor.

6.3. GRAPH SIMPLIFICATION OPERATIONS

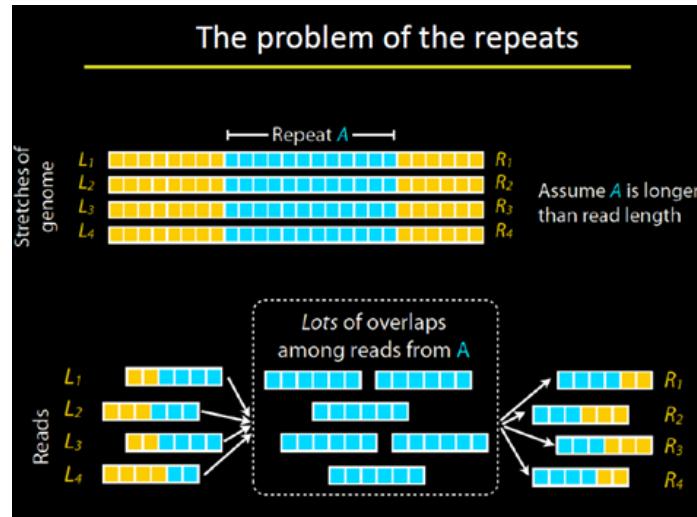


Figure 6.4

As always, the greedy approach will provide the best local solution, which might not be the best global solution.

6.3 Graph simplification operations

A practical trick that could be used to solve the graph, and to minimize the number of operations which must be performed (in an attempt to make it feasible), is to simplify it. Simplification operations include:

- Merging consecutive nodes: if among more reads there is only one possible path, we can just merge those reads, without losing information. So, nodes that are sequentially connected only with each other are linked together.
- Remove dead ends: if some reads are branching from a path toward a dead-end, we can remove them.

Here we may lose some information since the removed path could be the right one. Assembly by overlap graphs has several problems:

- It is problematic when dealing with repeats: maximizing the overall weight will produce wrong assemblies.
- It is not tractable: finding the optimal solution is not computationally feasible.

Overlap graphs can work to some extent, but now other algorithms are used in assemblers, like the de Bruijn Graphs (DBG) and the Overlap Layout Consensus (OLC).

As a simple example, in the case of the DBG, this problem would be splitted into 2 contigs, which is actually the best solution: better to have 2 right contigs, instead of 1 wrong contig.

6.3. GRAPH SIMPLIFICATION OPERATIONS

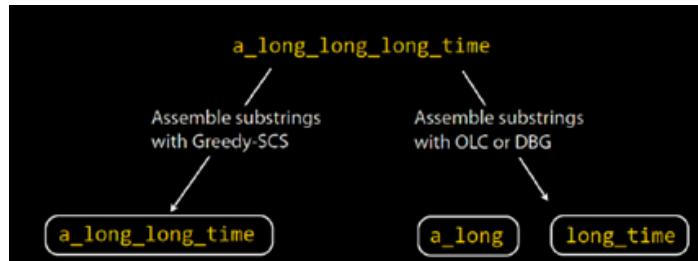


Figure 6.5

6.3.1 De Bruijn graph assembly

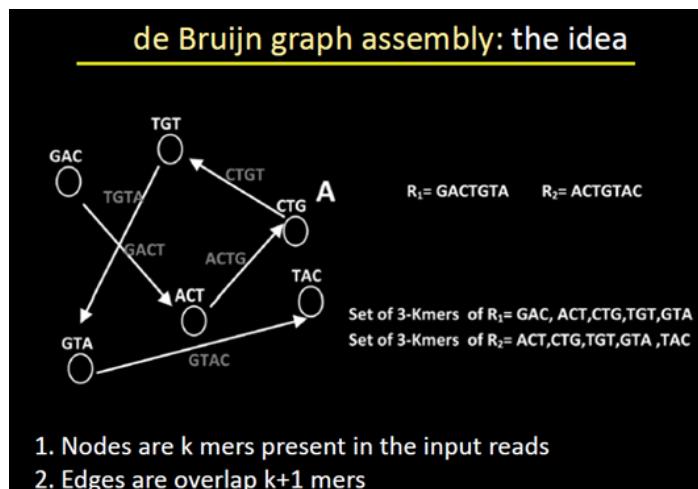


Figure 6.6

On this graph the nodes are not the reads but the k-mers present in the input reads (with k smaller than the read-length). We define the length of k-mers length in advance and create all combinations of k-mers for each read, for example for reads R1 and R2. Clearly there will be some overlap, some k-mers will be present in both sequences, but in the graph there will be only one copy of them. Hence, the nodes will be the **unique k-mers** present in the reads, the edges are the **overlaps** of the k-mers of length k+1 from the 2 sequences. Then, instead of using Hamiltonian paths touching the nodes, we use **Eulerian paths**. Those are paths in which all edges are visited, each edge exactly once. From a computationally point of view, they can be found much more efficiently than Hamiltonian paths (no explosion of search space). Other advantages:

- if we have a k-mer of length 3, we cannot have a huge amount of nodes, but 4^3 possibilities. So the set of nodes will not be too big compared to the number of reads. With $k = 3 \rightarrow$ huge amount of overlaps hence edges. But with longer k-mers (eg. 15) there will be a good trade off between the number of nodes and the number of edges.
- Cleaning reads: we can avoid putting in the k-mers that appear only once in a genome

6.3. GRAPH SIMPLIFICATION OPERATIONS

with a lot of coverage and are therefore the result of sequencing errors (otherwise the number of nodes would increase a lot).

- The de Bruijn graphs seem to be bigger sometimes, but they are not. With the de Bruijn graph the paths are more linear, there are less possibilities of wrong cycles.

6.3.2 Scaffolding

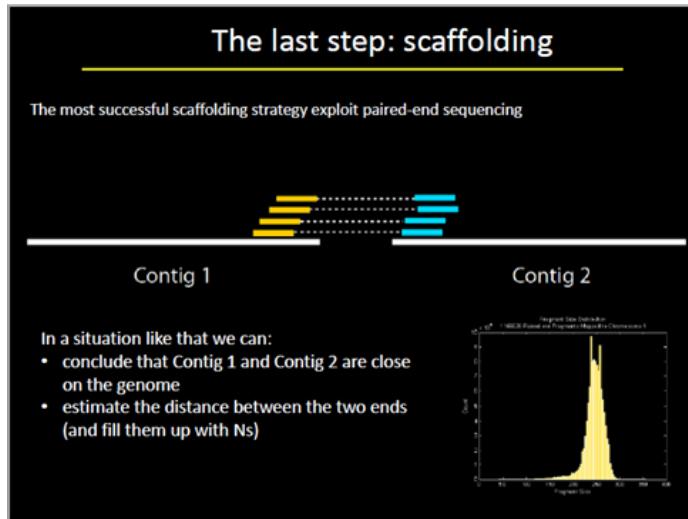


Figure 6.7

Scaffolding can be done with paired-end sequencing or with the estimation of the coverage. The fragment size in this case was 250 bp on average and with a Gaussian distribution like that one we can put an optimized number of Ns between the 2 contigs (not perfect, but connected). Doing it for all contigs leads again to a very complex situation. Problems: If the end of a contig matches the middle of another one in paired-end sequencing, there might be a mistake. Either the contig must be splitted because the overlaps are wrong, or the two matches are wrong.

6.3.3 Evaluating assemblies

The **N50** is the size of a set of entities (e.g., contigs or scaffolds) which represents the largest entity E such that at least half of the total size of the entities is contained in entities larger than E. Kind of a weighted median. For example, if we have a collection of contigs with sizes 7, 4, 3, 2, 2, the N50 is 4 because. (4 or more must be equal to what is 4 or less: $7+4 = 11$, $4+2+2+3 = 11$. Value in the middle for which the total length above and below is equal). Long contigs help to have a higher N50. Other definition: N50 length is the length 'x' such that 50% of the sequence is contained in contigs of length x or greater. X must be the length of a real contigs, not just a number. One long read = perfect N50 but might be wrong, we might check other characteristics. Eg. we can look at what the genes are coding for (is there at least a gene that codes for ribosomes? if not, wrong).

6.3. GRAPH SIMPLIFICATION OPERATIONS

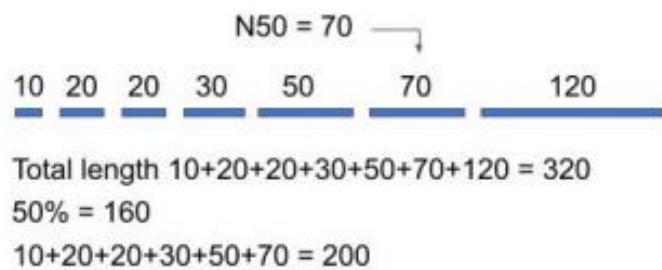


Figure 6.8

Chapter 7

16S-rRNA sequencing

7.1 Introduction to metagenomics

7.1.1 Definition of metagenomics

The term **metagenomics** refers to the "*study of uncultured microorganisms from the environment, which can include humans or other living hosts*" which "focus on taxonomic and functional characteristics of the **total collection of microorganisms** within a community". The main way to analyze the entire microbial population of an environment is through **high-throughput sequencing** of nucleic acids isolated from the sample; we can further distinguish two approaches, namely **16S rRNA gene sequencing** and **shotgun metagenomics**.

7.1.2 Why studying the metagenome

Microbes are basically everywhere, in and outside of our bodies, in oceans, glaciers, hot springs and rocks. Given how widespread and abundant microbes are, studying the metagenome provides us plenty of information (both on human and non-human microbiome and environment). For instance, it has been shown that the microbiome correlates to several diseases, therefore it can be used as a non-invasive **biomarker** (colorectal cancer, immunotherapy efficacy, autoimmune diseases...); the list of activities microbes are involved in is evergrowing.

7.1.3 Differences with older microbiome studies

The microbiome was discovered many years ago but there were no tools to analyze it properly; the only way was to culture and isolate each bacterium, which is an unfeasible approach to study the entire community, since only some bacteria can be grown in lab and it still would take an unreasonably long amount of time. The advent of high-throughput technologies is what made possible to study the microbiome of a sample, reducing significantly times, costs and increasing substantially the fraction of the microbiome that can be known.

7.1.4 Example: skin microbiome

Some studies were performed on skin microbiome (Segata et al, Nature Methods 2012, Truong et al, Nature Methods, 2015); only about 60% of the contigs (of various size) were mapped to known microbes while 40% belonged to unknown species. When separating these sequences based on GC content and abundance, many clusters formed, some with higher abundance while others with lower abundance, probably due to the low GC content that makes more difficult for the machine to sequence them, therefore causing them to be underestimated. Studying this 40% of unknown sequences is one of the main tasks of metagenomics.

7.2 16S rRNA sequencing

16S rRNA sequencing is one of the first techniques developed to study the microbiome, since it does not require a huge amount of sequences nor excessive costs; for these reason the technique became popular.

7.2.1 Simplified 16S rRNA analysis workflow

The general workflow for a 16S rRNA analysis is the following:

- **DNA extraction** from the entire community present in the sample; some bacteria will be over-represented while other will be under-represented.
- **Selective PCR amplification of 16S rRNA gene** (due to the characteristics explained below)
- **High-throughput sequencing**
- **Sequence mapping** against genomes in databases; this allows to define which bacteria (and which variants of those) are present in the sample and to find new and unknown bacteria.

7.2.2 16S rRNA gene

The ribosome is one of the most conserved, if not the most conserved, structure in all living organisms, making it one of the best phylogenetic markers. In prokaryotes, the ribosome is composed of several elements, both proteic and RNA based. Of the RNA based ones, 3 of them are **ribosomal RNAs** (rRNAs), namely 5S, 16S, 23S. Since these components are fundamental for any bacterium, all bacteria present the genes codifying for these rRNAs; most of the sequences are highly conserved but some regions have some variability which, since this variability is species-specific, can be used as a *barcode* to find and classify species (it also allows to distinguish between Archea and Bacteria). The most conserved of the rRNAs is 23S but the one used for microbiome analysis is 16S (which corresponds to the human 18S). The 16S rRNA gene is a few thousands nucleotides long, most of which are highly conserved; the bulk of the differences among species is in the hypervariable regions (named V1 to V9), which are terminal loops of the structure (basically regions far away from the catalytic site, therefore more free to mutate). Despite the high degree

7.2. 16S RRNA SEQUENCING

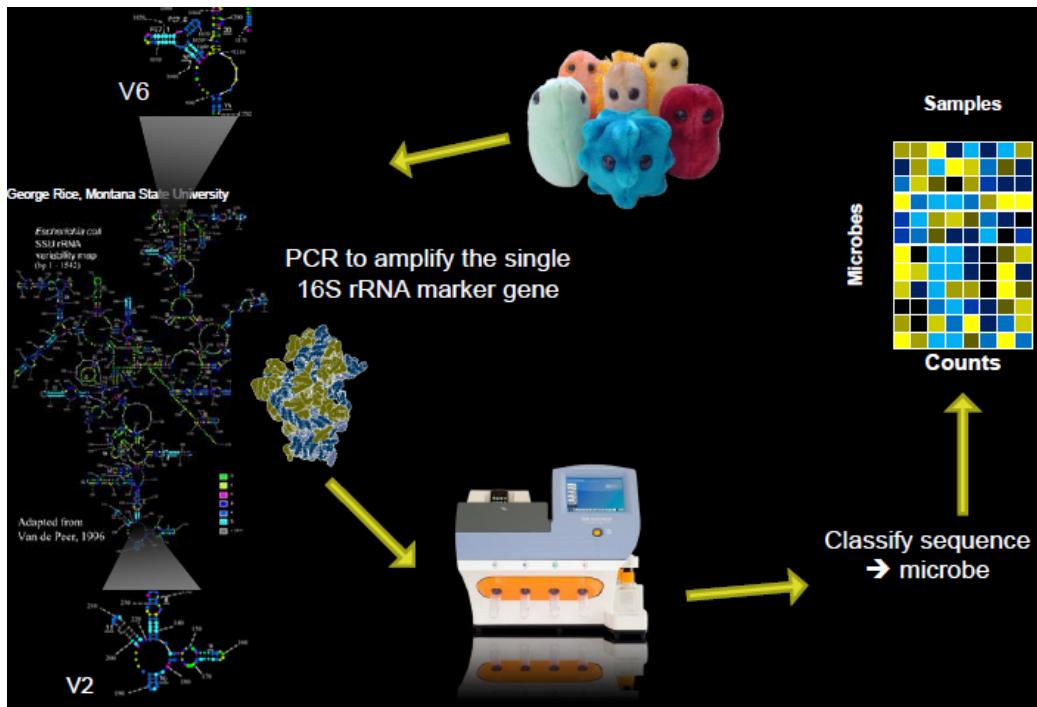


Figure 7.1: General 16S gene analysis workflow

of conservation, some variability can be found outside the hypervariable regions too (eg. 530 loop structure). The annotation of which portions of the 16S rRNA gene are conserved has been performed using *E. coli* as a reference; for a few hundred organisms the gene has been compared to the reference one to define the degree of conservation of each stretch of nucleotides. Some totally conserved regions (meaning pretty much identical in all species) are present but they are not very big.

7.2.3 Primer and high-throughput machine choice

One could sequence the entirety of the 16S rRNA gene, for example using NanoPore seq, but this would introduce many errors that could lead to mapping the sequence to the wrong organism; for this reason it is preferred to amplify only certain specific regions of the gene. To study the microbiome in a high-throughput way you need primers which can bind to all species, but since the sequences conserved in all species are too short, you use primers that bind highly conserved regions; for this reason, regardless of which primers you choose there will be bias in your results (some species will not be identifiable using those primers). This bias can be somewhat minimized using *in silico* primer validation, which means testing your primers against databases of 16S tRNA genes (silva and green genes), to test and decide the best pair of primers for your experiment.

Still, two experiments conducted with different primers will always have some differences. Moreover the binding regions must flank some variable region, in order to include it in the amplicon; finally you need pair end amplification (both primer back and forward)

7.2. 16S RRNA SEQUENCING

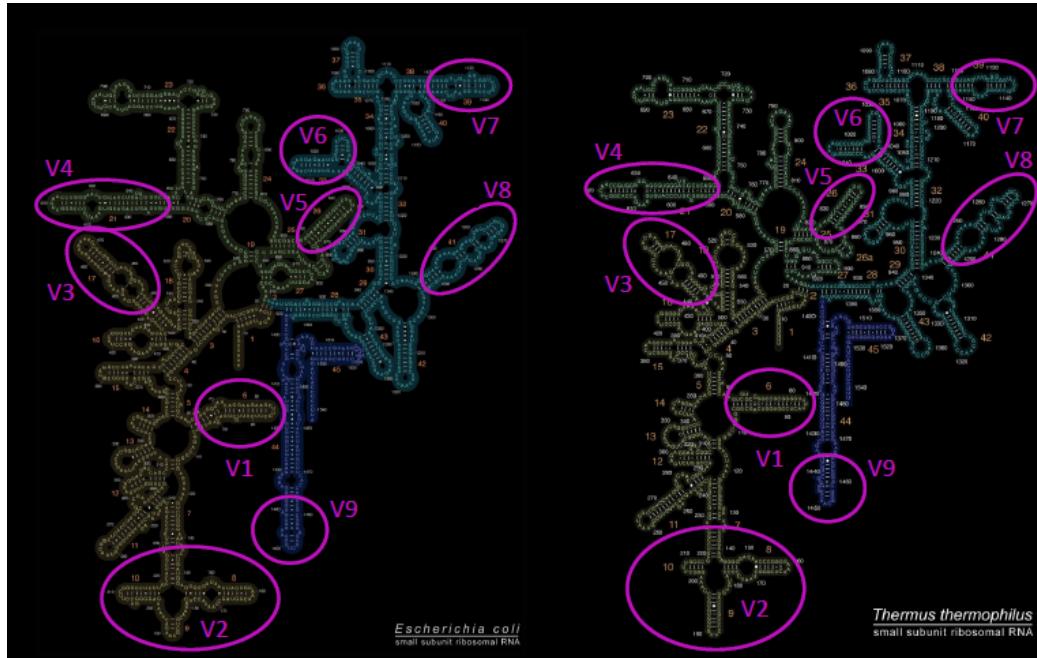


Figure 7.2: Structure of the 16S rRNA in *E. coli* and *T. thermophilus*

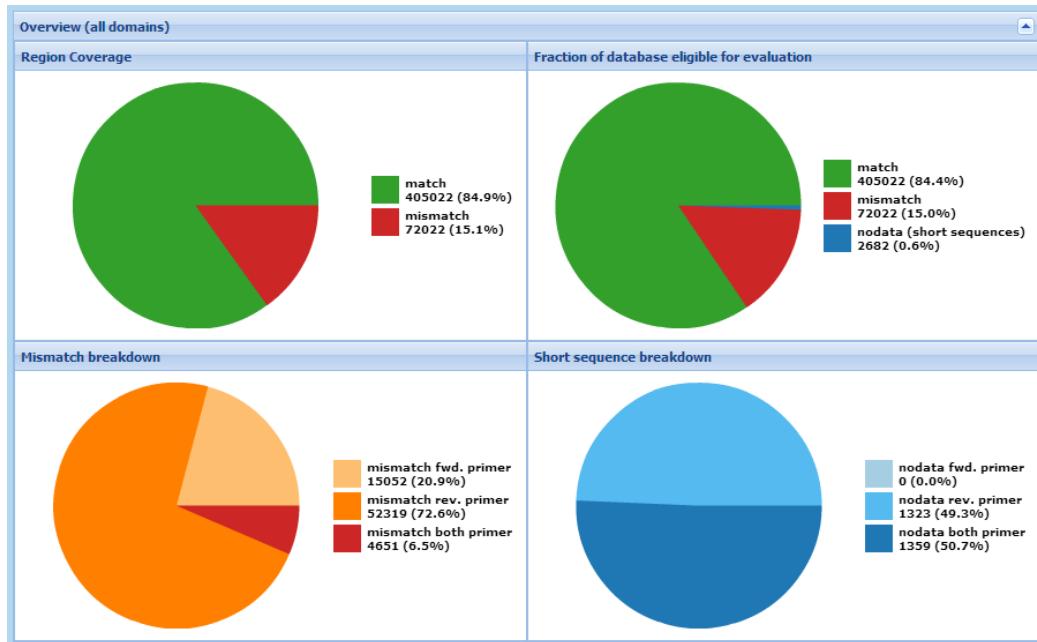


Figure 7.3: Example of *in silico* primer validation using silva; you can notice the different efficiency of the primers relative to different parameters.

7.2. 16S RRNA SEQUENCING

in order to have the complete amplicon (to make comparison easier). Given these characteristics there are multiple possible priming sites, based on the sequence and on chemical properties of the primers. Moreover, primers can be used as forward or reverse to obtain different combinations and sequences.

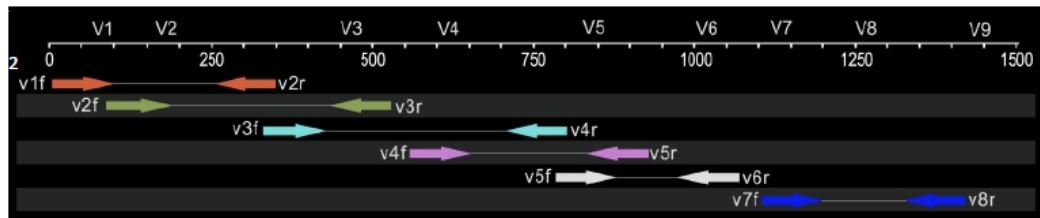


Figure 7.4: Examples of common primer placements relative to hypervariable regions

As an example of the importance of the choice of primers, in some skin microbiome analyses, researchers could not find two bacteria always present on human skin due to the choice of primers. Moreover *S. aureus* seemed over-represented due to the non-amplification of other important species. Despite the biases, this technique is still extremely useful even today.

There are different protocols to target conserved regions also based on the machine used. In general:

- Sanger machines are not very good for this application since they have low throughput and they are more suited for longer sequencing tasks (full genomes for instance)
- Roche 454 machines have historically been well suited, since it was possible to sequence three hypervariable regions together using 400 nucleotides reads, providing a good cost/throughput trade-off.
- Illumina HiSeq is not the optimal choice since it has shorter reads and unnecessarily high throughput; Illumina MiSeq and IonTorrent can be a decent compromise.

7.2.4 In depth 16S rRNA analysis workflow

A more in depth 16S rRNA analysis workflow is the following:

- **DNA extraction** from each of your samples
- **Selective PCR amplification of 16S rRNA gene**, introducing a barcode in the sequences using tagged primers.
- **High-throughput sequencing** of all the samples in a single run (to reduce costs); the result is a set of amplicons belonging to different samples and with a barcode attached.
- **Demultiplexing**, which means removing the barcodes and assigning each sequence to the corresponding sample. Sequencing noise must be taken into account, therefore low quality reads must be removed.
- **Multiple sequence alignment** against reference sequences. Some reads will probably not map to any reference sequence.

7.2. 16S RRNA SEQUENCING

- **Group related sequences into OTUs** (operational taxonomic units), which means grouping sequences that share some common variants; since there are some SNPs in the microbial genome, the similarity threshold between sequences cannot be too restrictive. OTUs can be used to define the relative abundance of each species in the sample, but in order to do so it is necessary to normalize for the copy number of the 16S gene sequence; this is very difficult since an accurate estimate can be made only if long reach sequencing has been performed on the organism, which is almost never the case since for microbes that basically corresponds for full genome mapping (needless to say it is therefore non applicable on unknown microbes).
- **Build phylogenetic tree** using one representative for each OTU.
- **Annotate** the OTUs using 16S gene databases.
- **Downstream analysis** is performed, such as clustering to visualize similarities among samples.

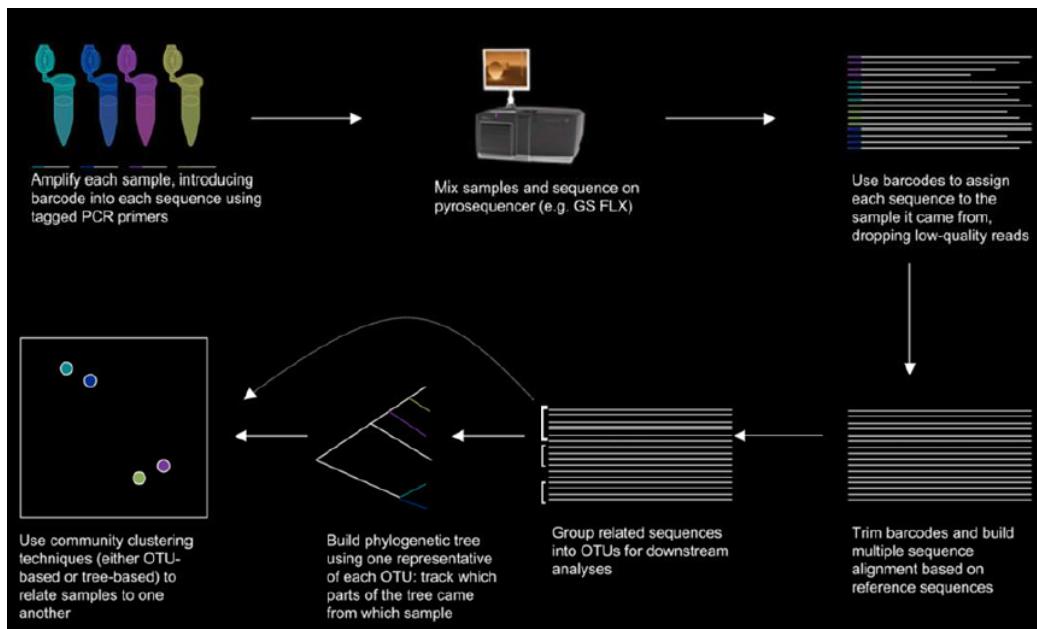


Figure 7.5: Expanded 16S gene analysis workflow

7.2.5 OTU clustering

Defining OTUs requires using multiple sequence alignment; since this approach is a generalization of the mapping algorithm (meaning you have to compare every sequence with every sequence) it is quite complex in terms of speed, but still feasible. Generally, greedy algorithms, which add the lowest possible amount of gaps, are used to perform multiple sequence alignment. After the alignment, sequences are split into OTUs (operational taxonomic units), which are basically groups of 16S sequences very similar to each other.

7.2. 16S RRNA SEQUENCING

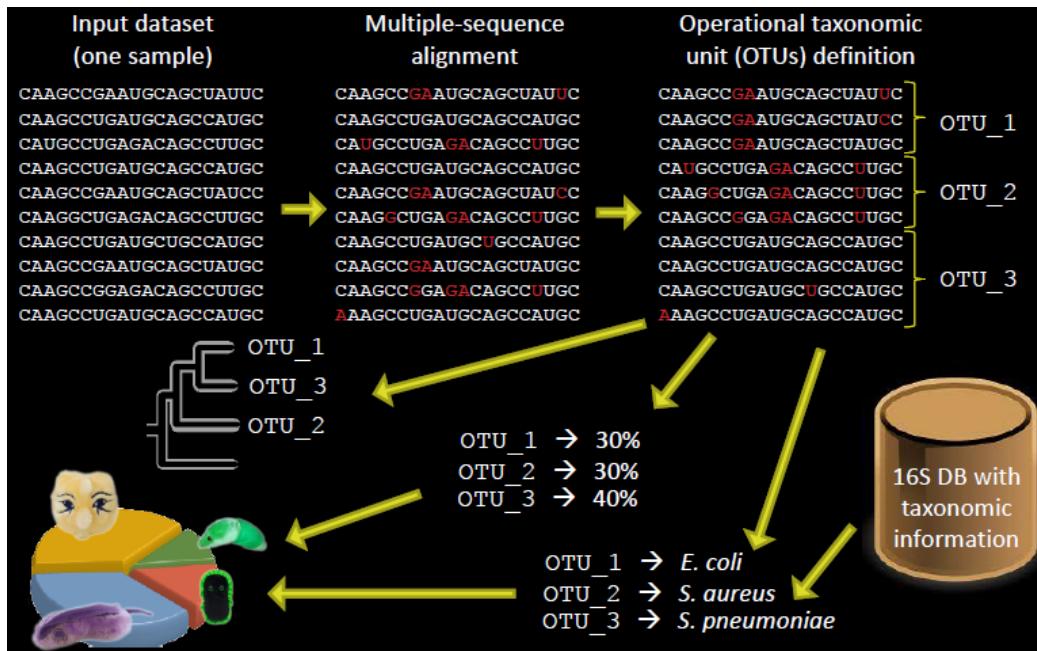


Figure 7.6: Zoom in on 16S gene analysis workflow

Generally a sequence is defined as the representative of the OTU, meaning that it has a certain threshold of identity with all other sequences in the OTU (usually 97% when considering species) and that minimizes the differences of all other sequences of the OTU with itself. Some OTUs can be assigned univocally to a species, some others may be associated to more species, some others cannot be mapped to known species. The fact that a species may map to multiple OTUs is often a negative factor (confusion in the analysis) but it may sometimes allow to find subspecies.

After sequence alignment, OTU clustering (= splitting the sequences into OTUs), can be done through several supervised or unsupervised learning methods. Each method has pros and cons, therefore there is not an always optimal method. The most common unsupervised clustering methods are:

- **Single linkage clustering** (nearest neighbour): assign the sequence to a cluster if that OTU already contains **at least a sequence** similar enough (97%). However two distant sequences in the OTU network could share a similarity which is way lower than 97% (because of a series of connections above 97%); this could result in **underclustering** (defining too few clusters).
- **Complete linkage clustering** (furthest neighbour): assign the sequence to a cluster only if **all the sequences** of the OTU are similar enough (97%). However two sequences may be similar enough (97%), yet belong to different OTUs, because the overall cluster width, or **divergence**, is at most 3%; this approach could then generate different solutions, based on the order the points are added in. Moreover, if the clustering conditions are too stringent, sequencing errors and SNPs in the microbial genome may result in **overclustering** (defining too many clusters).

7.2. 16S RRNA SEQUENCING



Figure 7.7: Example of multiple sequence alignment for OTUs

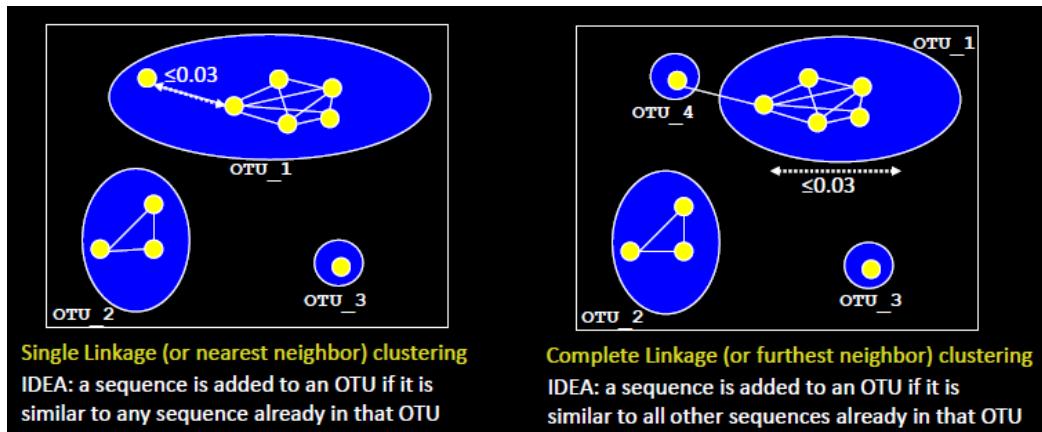


Figure 7.8: Visualization of single linkage analysis and complete linkage analysis

7.2.6 OTU taxonomic annotation

NOTE: this topic continues in the next lecture; when done merge together Assigning a taxonomic annotation to an OTU cannot be done simply using BLAST to get the best matching sequence; this is because there is too much noise in the sequences and because it is difficult to classify new strains. A better way is using some other algorithm that assigns the terms of the taxonomic notation (since it is more than just one label) and provides some degree of confidence in the prediction. For instance the algorithm may be able to correctly assign the first taxonomical terms, up until *Enterobacteriaceae*, but then it provides a prediction of the OTU belonging to *E. coli* with some confidence interval, say 85%, and some alternative like *S. dysenteriae*, say with 15% confidence.

7.2. 16S RRNA SEQUENCING

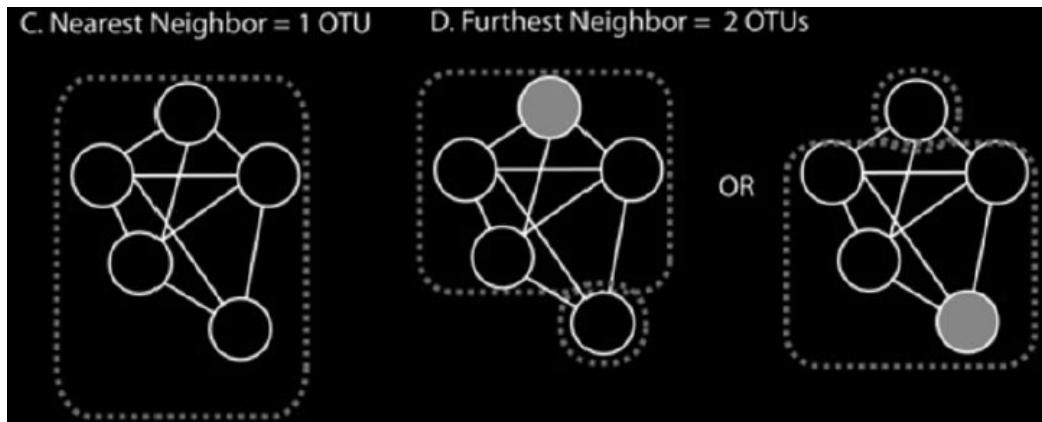


Figure 7.9: Example of overclustering and result multiplicity due to complete linkage analysis

7.2.7 RDP classifier (Naive Bayes Model)

$P(S|G)$ is computed by RDP using a 8-mer strategy basically comparing the 8-mers in S with the 8-mers in all the training sequences available for genus G . The confidence of each prediction is computed by bootstrapping:

- Select a (random) subsequence of S , Say S'
- Compute the G that maximizes $P(S'|G)$
- Repeat the procedure 100 times
- The number of time G is selected by the bootstrapping procedure is the confidence

Leave one-out-cross validation:

- Take out one data point from the training set
- Apply the classifier on the left-out point (without using it in the training set)
- Check the accuracy of the prediction
- Repeat the procedure for each training data point

The RDP classification accuracy is evaluated with the leave-one-out cross validation on the training set of hundreds thousands of 16S references:

- Accuracy → percentage of correct classification (over the all leave-one-out runs)
- Varying levels of the taxonomic resolution (from phylum to genus)
- Varying sequence lengths
- In real applications accuracies are probably smaller:
 - Sequencing errors
 - "unknown bacteria"

7.3 Diversity analysis

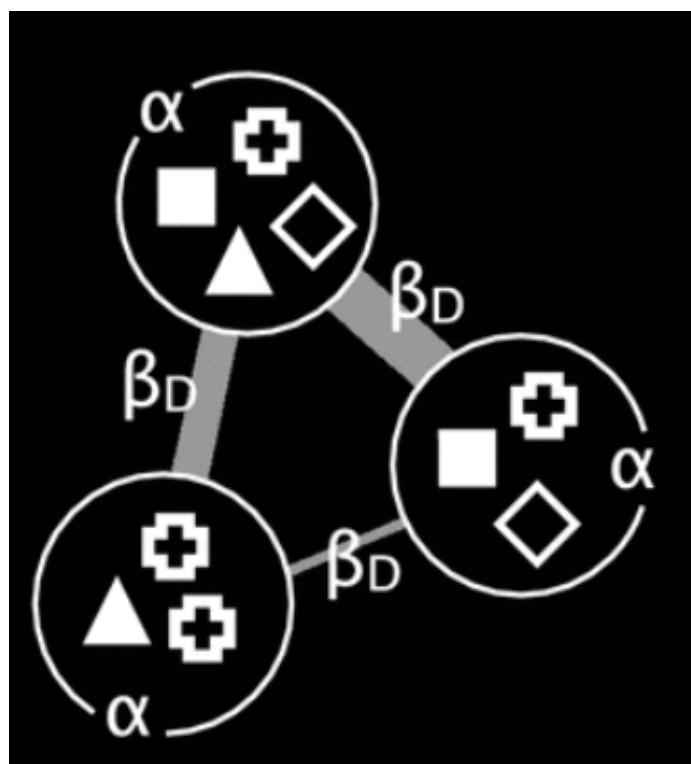


Figure 7.10

7.3.1 Alpha diversity analysis

Alpha diversity analysis is a measure of how diverse, so complex, is a microbial community. "Within sample" diversity. Species richness is a widely used alpha diversity index. All individuals considered have non-zero abundance, some will have high abundance (~ 99%) or low abundance (1%). High alpha diversity are usually associated with populations that are more robust and resilient to changes. For example gut microbiome with a high richness is usually associated with healthy state, instead of disease.

7.3.2 Beta diversity analysis

Beta diversity analysis is a measure of how different two microbial communities are. "Between sample" diversity. It is possible to measure the beta-diversity using the inverse of number of shared species. An example of beta-diversity is **UniFrac**. In UniFrac the distance is equal to the fraction of the total branch length that is unique to any particular environment. UniFrac can be also weighted in order to include abundances for each OTU.

7.4. PRINCIPAL COORDINATE ANALYSIS (PCOA)

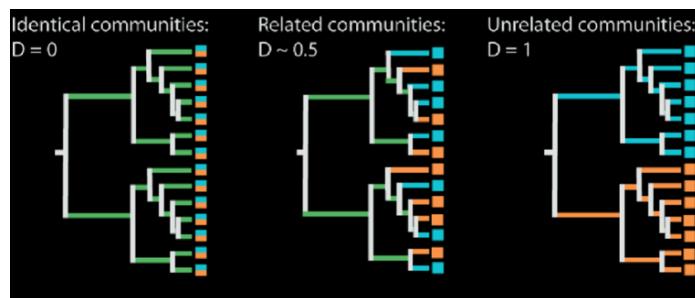


Figure 7.11: D=0. Blue and orange samples always have the same OTUs. Each 16S (each ramo) is present in both samples. **D=~0.5.** In reality we usually have a mix of the 2 situations. Some OTUs are present only in one of the samples and are either quite distant from the others or close (based on upstream the branch goes). **D=1.** Completely distinct OTUs . The difference is also in the upstream branches, which have different colors.

7.4 Principal Coordinate Analysis (PCoA)

PCoA is also known as multidimensional scaling. It is one of the most powerful approaches for exploratory analysis. The idea is to represent the multidimensional relationship between samples in a two or three dimensional space. It is possible to use any similarity function as Euclidean distance, UniFrac, bray-Curtis distance. We find frequently hierarchical clustering plots.

7.5 Study: *Enhanced microbial diversity in the saliva microbiome induced by short-term probiotic intake revealed by 16S rRNA sequencing on the IonTorrent PGM platform*

The pool was made of 14 samples and some of them with probiotic administration, so a very small number of high-quality reads. The samples number 1 and 2 are of the same student that repeat the sampling twice in order to see the reproducibility of the technique. All the students had firmicutes, proteobacteria and Bacteroidetes, plus a little bit of actinobacteria.

You cannot compare alpha-diversity of different student at different sequencing depth, but at the same. When you do PCR, multiplexing and you put multiple samples on the same sequencing run the sequencing machine will not get the exact n° of reads for each sample, you have a quite high variability. Since you cannot compare alpha-diversities at multiple sequencing depths, usually you have to decide a sequencing depth cut-off at least for alpha-diversity analysis. The greater the sequencing depth, the greater the alpha-diversity.

7.5. STUDY: ENHANCED MICROBIAL DIVERSITY IN THE SALIVA MICROBIOME INDUCED BY SHORT-TERM PROBIOTIC INTAKE REVEALED BY 16S RRNA SEQUENCING ON THE IONTORRENT PGM PLATFORM

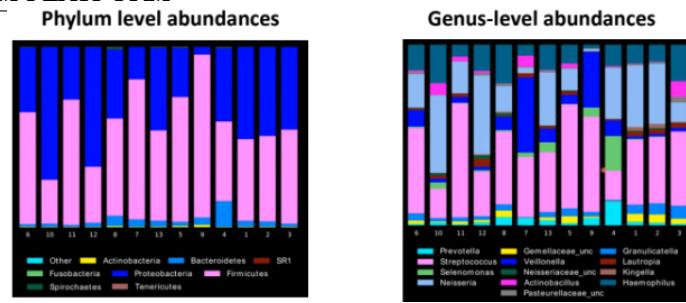


Figure 7.12

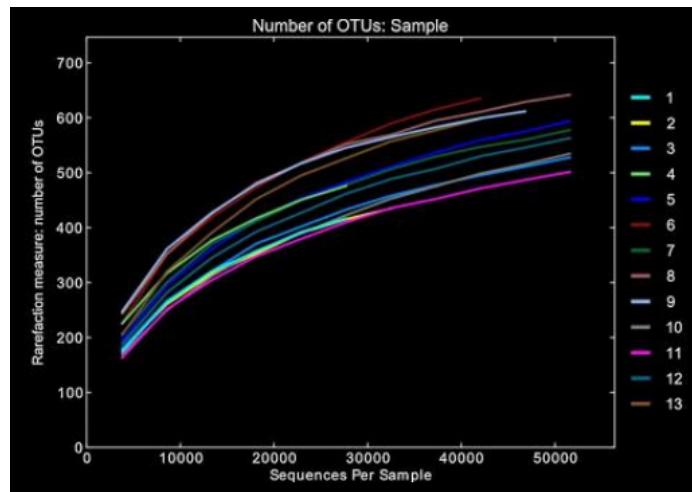


Figure 7.13: Alpha diversity and rarefaction plots. The cut-off should not be too high, otherwise some samples will not be included in the further analysis. (If they do not reach depth, like the student 4 in light green.)

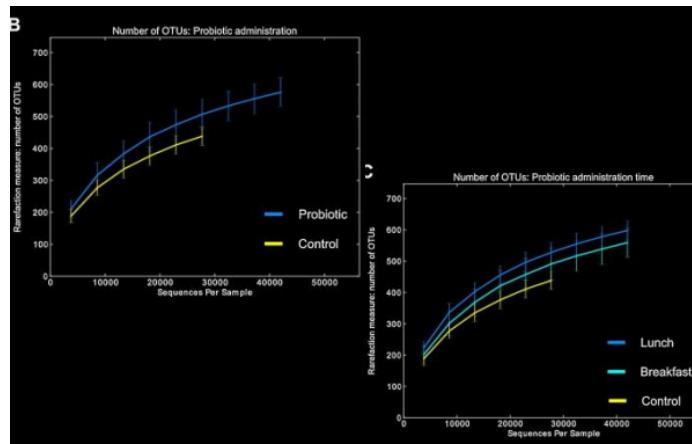


Figure 7.14: Probiotic effect on saliva microbiome? Students that took the probiotic had a more diverse microbiome, and the maximum diversity was reached by taking probiotics at lunch.

7.5. STUDY: ENHANCED MICROBIAL DIVERSITY IN THE SALIVA MICROBIOME INDUCED BY SHORT-TERM PROBIOTIC INTAKE REVEALED BY 16S RRNA SEQUENCING ON THE IONTORRENT PGM PLATFORM

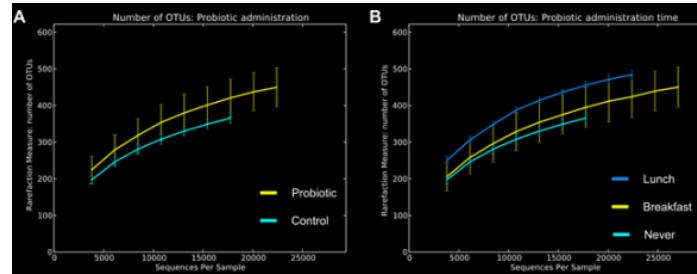


Figure 7.15: Is the effect due to the probiotic Streptococcus? The probiotic product contained *Streptococcus thermophilus*, *Lactobacillus delbrueckii* subsp. *bulgaricus* and *Lactobacillus paracasei*. What if we remove OTUs from these genera from the dataset? The analysis is still significant. This means that the probiotics yogurt induces a variation in the microbiome that make it more diverse.

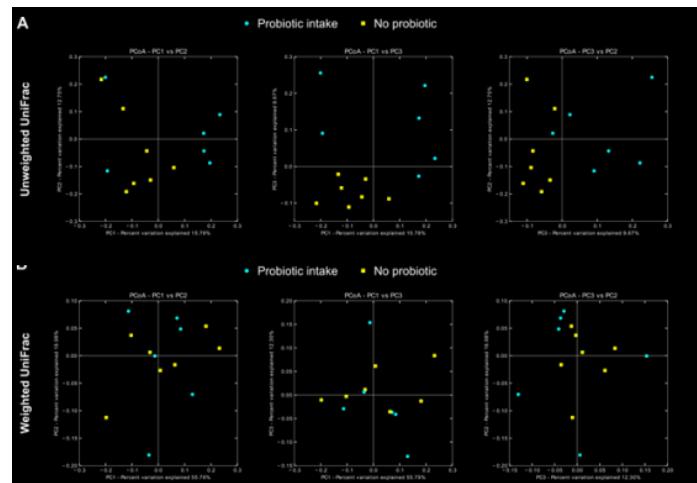


Figure 7.16: Looking at beta diversity There is no clusters.

Chapter 8

Shotgun Metagenomics

8.1 Workflow

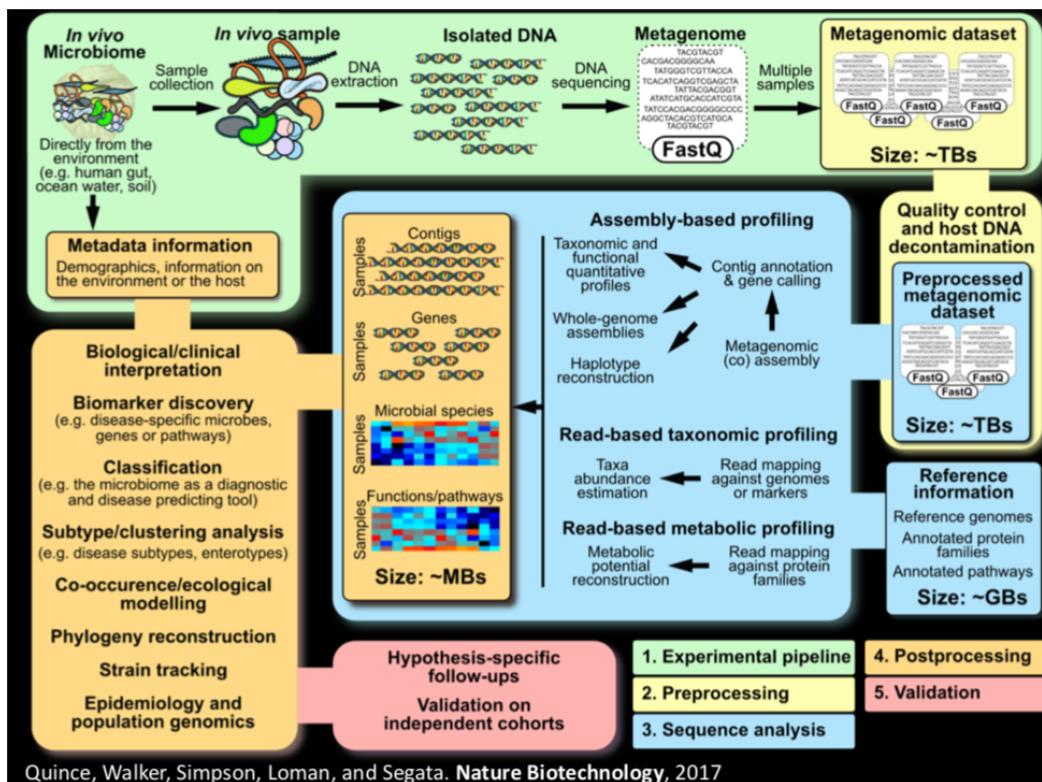


Figure 8.1: Shotgun metagenomics workflow

- **Experimental pipeline** → from sample collection to DNA sequencing.
- **Preprocessing** → decontamination + quality control

8.2. COMPARISON WITH THE 16S SEQUENCING

- **Mapping** or **assembling** (or both)
- **Sequence analysis** → identification of microbial species + identification of present pathways/functions
- **Post Processing** → integrate data with other information (where do the sample come from, healthy vs disease and other metadata)
- **Validation** → follow-up experiments + independent replicates

8.2 Comparison with the 16s sequencing

16S sequencing	
Pros	Cons
Cost-effective Avoids non-bacterial contamination Can catch low-abundance bacteria The output has reasonable size and complexity Mature software available to perform the computations	Non genome-wide Limited taxonomic resolution Not useful for pathogens profiling Does not detect viruses or eukaryotes Several biases Cross studies are difficult: the comparison is not possible due to biases

Shotgun sequencing	
Pros	Cons
Genome-wide: it is possible to retrieve information about all the genes present in the metagenome High taxonomic resolution Easy cross-study comparison thanks to the lack of biases All domains of life can potentially be observed in the same study	Expensive (but costs are decreasing: right now the cost for the sequencing of one sample is 100\$) DNA contamination are hard to remove Low-abundance bacteria could be missed Large dataset as output, that can be difficult to process (TBs of data)

8.3 Latest technology

The cost of 100\$ for the sequencing of one sample refers to a sample size of 5Gb, that should be enough for shotgun metagenomics. More sequencing depth could be needed if you want to detect microbes with very low abundances, if you want to assemble as many genomes as possible and if there is a lot of contamination.

Shotgun metagenomics is possible with **Illumina Hiseq** technology, but the latest and most used technology nowadays is **Illumina NovaSeq**.

8.4. IDENTIFICATION OF MICROBES FROM SHOTGUN METAGENOMICS DATA: DO WE REALLY NEED SOMETHING FANCY?

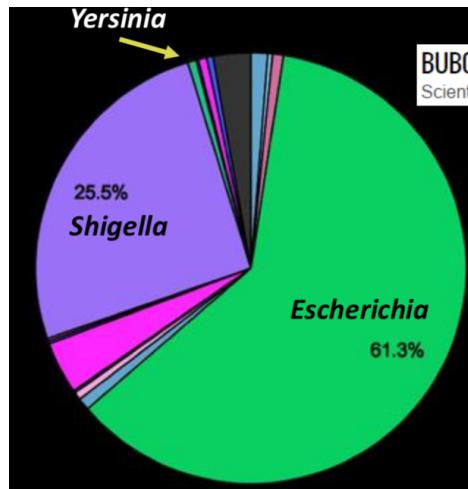


Figure 8.2

8.4 Identification of microbes from Shotgun Metagenomics data: do we really need something fancy?

N.Loman studied why a simple alignment (with BowTie because with BLAST it is not feasible) is not enough to determine which microbial species are present in our samples. He took one *E. coli* strain (K12), shredded its genome into 100 nts reads and classified these “reads” based on the best hit given by an alignment tool. Only 61% of the reads were assigned to *E. coli*. Even some *Y. pestis* resulted, probably because of some plasmid it has in common with *E. coli* (8.2).

With a similar approach, a study identified genes belonging to all sorts of organisms in samples taken in the New York City subway.

So yes, we need some fancy technique to solve this problem. The main **challenges** are:

- How to obtain specie-specific resolution
- Computational feasibility
- Being able to detect both bacteria and archaea. Phages are also a problem since there is little reference
- Obtain relative abundances of organisms with different genome sizes: the problem is to obtain the organism abundance and not the DNA abundance
- Consistent detection confidence for all clades
- How to handle reads as short as 50 nts
- Detect organisms without a sequenced genome or still unknown species.

8.4. IDENTIFICATION OF MICROBES FROM SHOTGUN METAGENOMICS DATA: DO WE REALLY NEED SOMETHING FANCY?

8.4.1 MetaPhlAn: unique marker genes for taxonomic profiling

Solution brought to you by Segata with C. Huttenhower (reference)

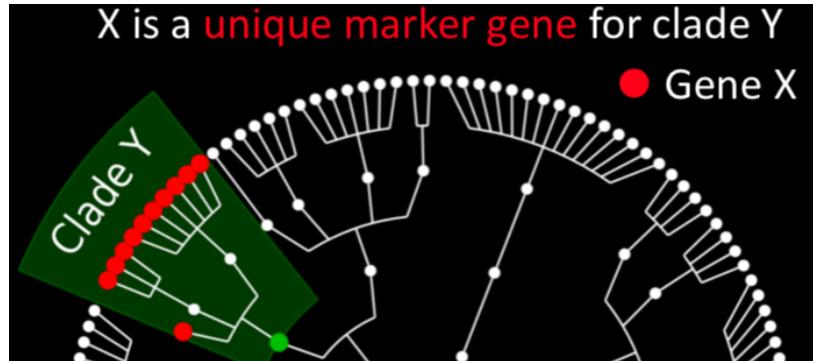


Figure 8.3

The idea is to find a marker gene that uniquely characterizes a species(8.3): it has to be present in all strains of a species and in no other species. These markers are then used to form taxonomic clades. The tool that generates the database is named **ChocoPhlAn**.

From a number of genomes, with ChocoPhlAn they created a database of marker genes of about 2 millions (400k are considered “most representative”) markers: the **MetaPhlAn** database. It contains about 200 markers per species. With the reduced size of this database compared with the whole genomes, it is possible to align the reads (first with BLAST, now they do it with BowTie2) directly on the markers database.

When they first created the marker genes database, they used 2887 genomes (and half of them were drafts) that belonged to 1222 species. The number of markers per species decreases with more sequenced genomes: the core genome tends to become smaller in size and noise is eliminated as more information is available. In the current version of BioBakery (the suite containing these tools) 1 million genomes are considered, of which about 200 000 are genome isolates and 800 000 are **MAGs** (Metagenomics-Assembled Genomes → putative genomes obtained from metagenomes).

MetaPhlAn computational performance can deal with about 200 000 reads per second and can profile thousands of microbiomes in some hours.

Validation with synthetic metagenomes: 10 well-known metagenomes were created and evaluated with MetaPhlAn to check if the results correspond to the actual species of microbes used to assemble the synthetic metagenome. Errors were also added in order to simulate sequencing noise.

Validation with biological methods: comparison of the results with the ones obtained with 16S-based abundance estimation.

8.4.1.1 The problem of the unknowns

Microbial species that were never observed and do not have a marker in the database cannot be detected with this method.

The solution is to cluster together contigs obtained from the metagenome based on coverage, GC content, codon bias, and other possible features. Strict quality controls are then performed on these putative genomes: on the number of genes and the number of

8.4. IDENTIFICATION OF MICROBES FROM SHOTGUN METAGENOMICS DATA: DO WE REALLY NEED SOMETHING FANCY?

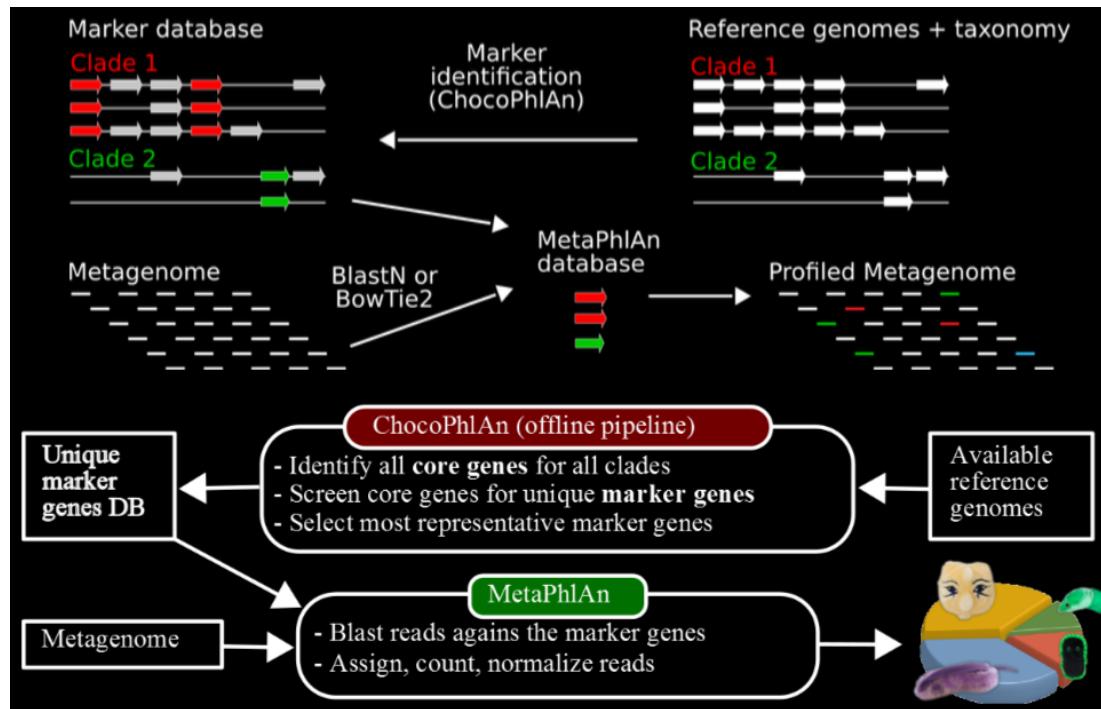


Figure 8.4: MetaPhlAn overview

known single-copy genes in order to be sure that what has been found is not a mixture of genomes. High quality putative genomes are considered MAGs and version 4 of MetaPhlAn can include them in the creation of the marker database. This way, these species can be detected in metagenomes even though they do not have a name yet.

8.4.2 Other approaches

Reference-free approaches:

- **Sequence-based clustering** of contigs to create putative genomes. The output are unlabelled bins with relative abundances. The main problem is that high coverage is needed to obtain valid results.
- **Machine learning algorithms** that exploit GC content (and possibly other features) to give as output clades with relative abundances. This method is not completely reference-free as it uses reference genomes to extract the features used for the classification. The main problem is that many species could have really similar features.

On the opposite side: **read-to-genome sequence mapping**. There is no processing of reference genome. The problems were discussed before: many reads map more than one genome but only one species is detected for that read.

8.4. IDENTIFICATION OF MICROBES FROM SHOTGUN METAGENOMICS DATA: DO WE REALLY NEED SOMETHING FANCY?

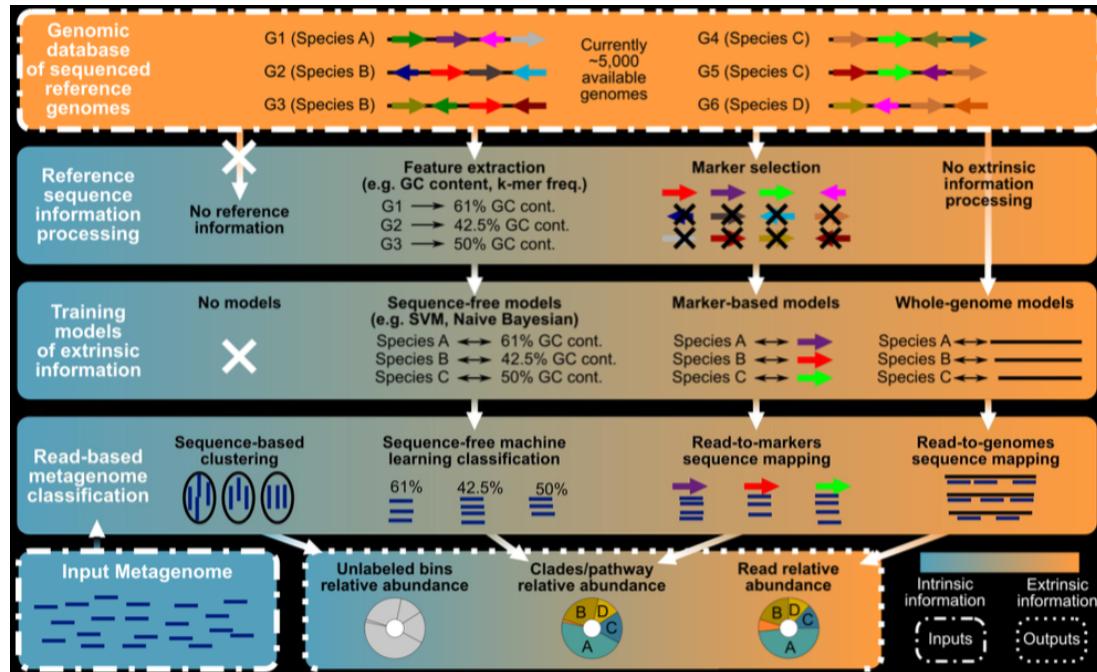


Figure 8.5: An overview of taxonomic profiling approaches

8.4.3 The curatedMetagenomicData resource

Since raw metagenomic sequencing data can be quite difficult to deal with computationally, this database stores features obtained from raw metagenomic datasets uniformly processed (MetaPhlAn or HUMAnN2) and integrated with associated metadata obtained from NCBI, papers and authors. This database is accessible and can be exploited to perform various types of analysis. (reference)

8.4.4 The link between the gut microbiome and colorectal cancer

Colibactin is a genotoxic metabolite produced by *E. coli*: it causes damages to the DNA, possibly causing cancer onset. A fraction (?) of human **colorectal cancer** (CRC) cases are caused by colibactin.

A study published by Segata's group collected stool samples from people having a colonoscopy in Milan (8.7a) and Turin (8.7b). The samples were then categorized after the diagnosis provided by the colonoscopy. The aim was the identification of **biomarkers** associated with the CRC phenotype. Different results were obtained in the two cities.

Comparing this study with similar ones performed around the world (France, China, Austria, USA, Germany and Japan) some biomarkers appeared to be reproducible (8.8a). Moreover, an accuracy around 80% was observed when a machine learning approach (random forest) was applied on all the datasets combined and then the model was applied on a brand new one(8.8b). On the other hand, when each group tried the same approach on their data separately, completely different results were found for each dataset, showing

8.5. PANPHLAN: STRAIN-LEVEL PROFILING

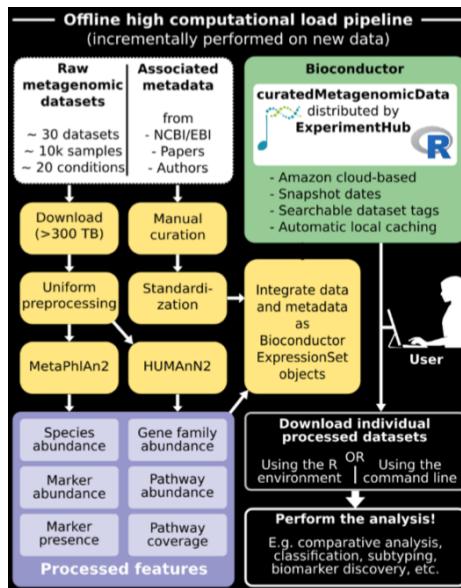
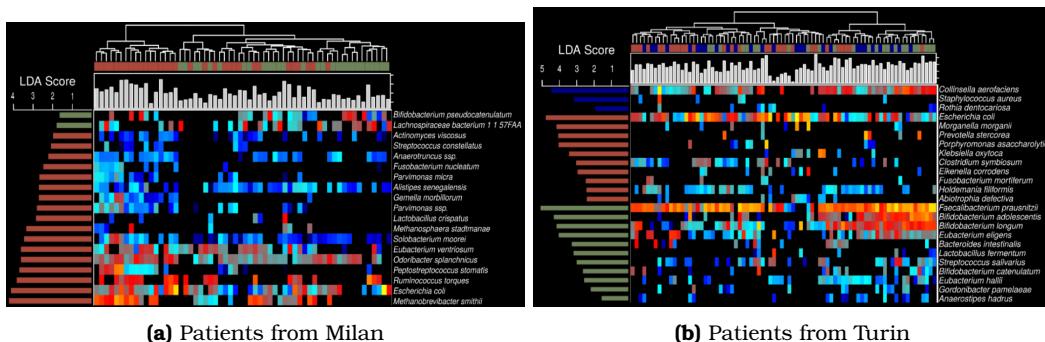


Figure 8.6: CuratedMetagenomicData pipeline



that such a technique could be valid for some cases and completely useful for others.

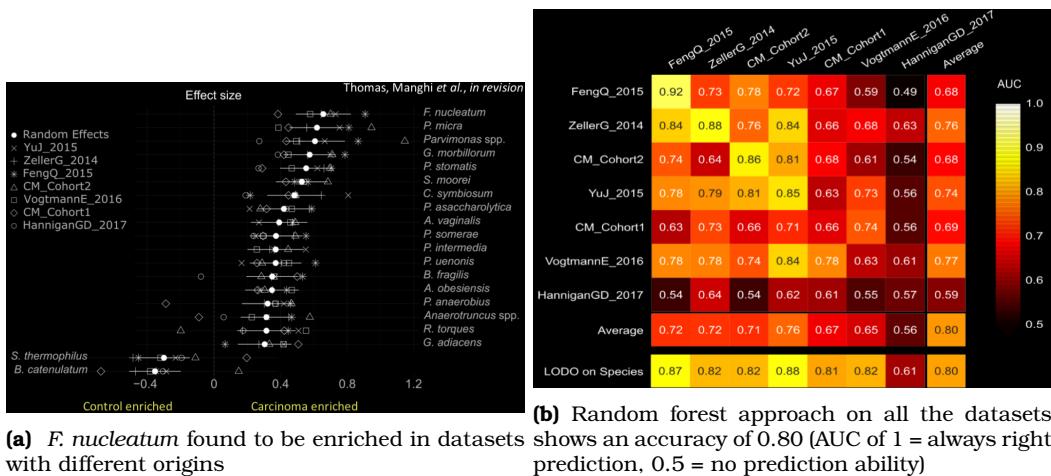
8.4.4.1 Hypothesis-driven analysis

The **cutC** gene appears to be associated with the CRC phenotype (8.9). The problem is that it is present in several microbial species and it is unknown whether the function and the efficiency are maintained.

8.5 PanPhlAn: strain-level profiling

Another tool brought to you by Segata's crew (reference).

8.5. PANPHLAN: STRAIN-LEVEL PROFILING



(a) *F. nucleatum* found to be enriched in datasets with different origins

(b) Random forest approach on all the datasets shows an accuracy of 0.80 (AUC of 1 = always right prediction, 0.5 = no prediction ability)

Figure 8.8

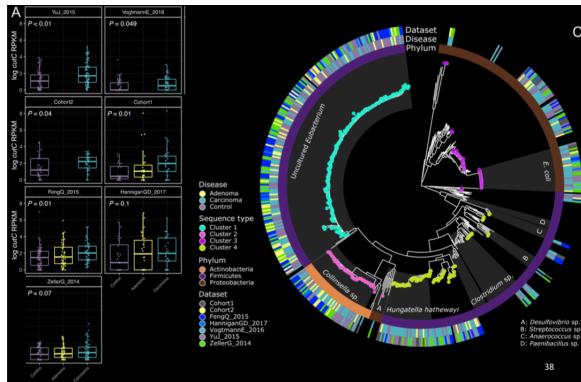


Figure 8.9

PanPhlAn is a tool for strain-level metagenomic profiling that allows to identify gene composition of individual strains in metagenomic samples.

Challenges:

- Identify the microbial strains present in the metagenome. One could also check whether a specific strain is present but the chances of finding one specific strain in a random sample are low.
- Discover new strains and species.
- Characterize the metagenome genomically.
- Track across samples to find the same strain and eventually prove that transmission of bacterial strains occurred among them.

The *E. coli* pan genome contains about 20.000 gene-families. The goal is to find what strains are present in the metagenome and their abundances.

8.5. PANPHLAN: STRAIN-LEVEL PROFILING

First all genes are grouped in **functional gene-families** (8.10). Then genes from *E. coli* found in the metagenome are mapped on *E. coli* reference genomes with BowTie2. The coverage is computed for each gene and then they are grouped into gene-families (8.11). Then gene-families are ranked based on their coverage. Multi-copy gene families have really high coverage, while the plot shows a plateau of single-copy gene-families. These families correspond to the strains present in the metagenome and their abundance can be deducted thanks to the base coverage of single-copy genes.

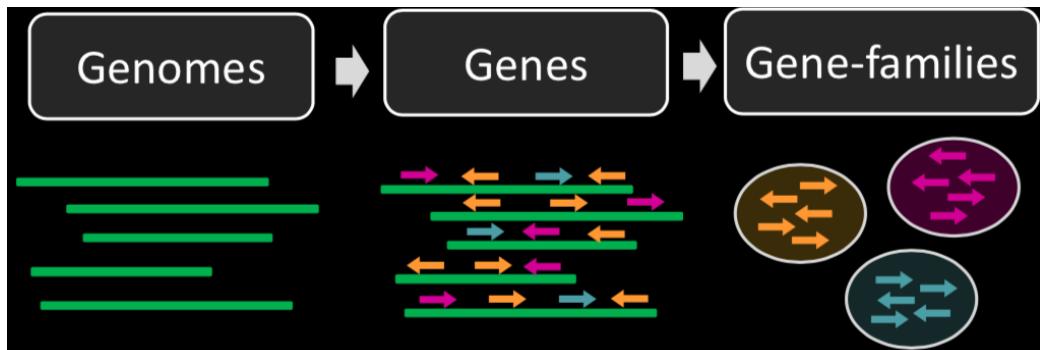


Figure 8.10: Genes are grouped in functional gene-families

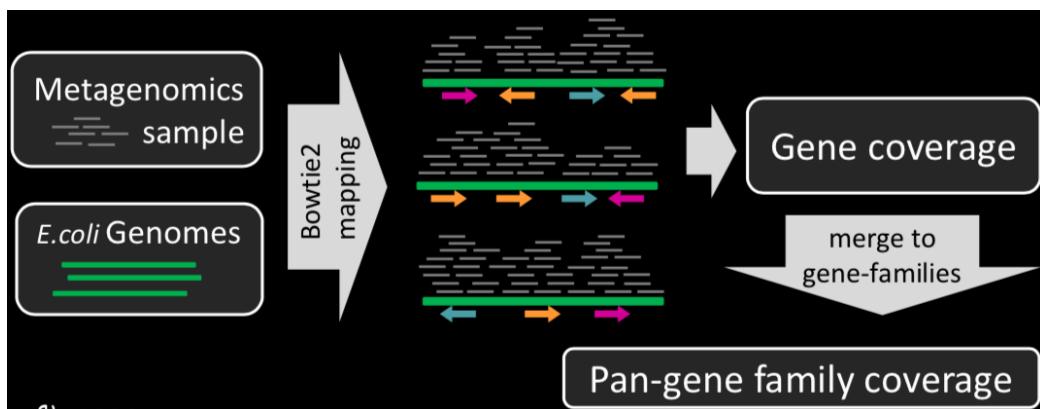


Figure 8.11: Mapping and subsequent coverage of gene-families

8.5.1 Investigating population genomics thanks to PanPhlAn

8.5.1.1 *E. coli* population genomics with PanPhlAn

Scholz, M., Ward, D., Pasolli, E. et al. Strain-level microbial epidemiology and population genomics from shotgun metagenomics. *Nat Methods* 13, 435–438 (2016).

Figure 8.13a shows the *E. coli* profiling of 1478 shotgun metagenomes carried out with PanPhlAn. Each column is either an *E. coli* strain obtained via shotgun metagenomics or a reference strain and the columns correspond to the gene-families that can be absent

8.5. PANPHLAN: STRAIN-LEVEL PROFILING

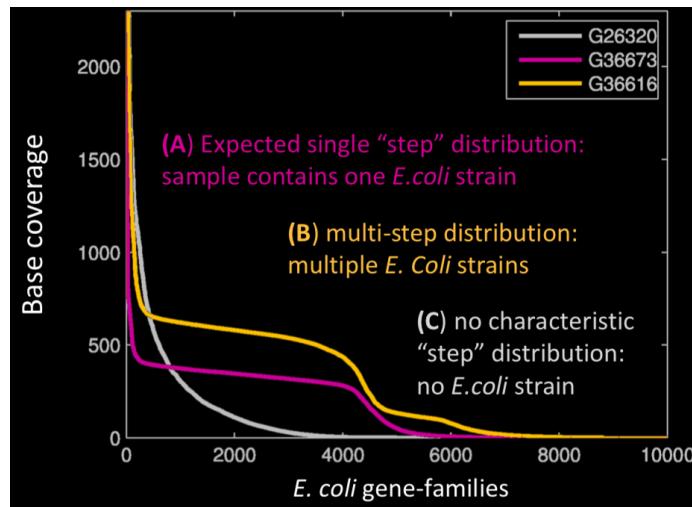


Figure 8.12: *E. coli* gene-family distribution: curve (A) shows the typical gene-families distribution: multi-copy genes with extremely high coverage, a plateau of single-copy genes and a tail of non-present gene-families. Curves like (C) should be discarded from the analysis because they indicate that no *E. coli* strain is detected in that sample.

or present in each strain. The strains are then clustered (8.13b) based on which gene-families are present in order to study the population genomics: in this case we can see that the strains isolated from the German *E. coli* outbreak cluster together, while other strains are present in several different areas of the world.

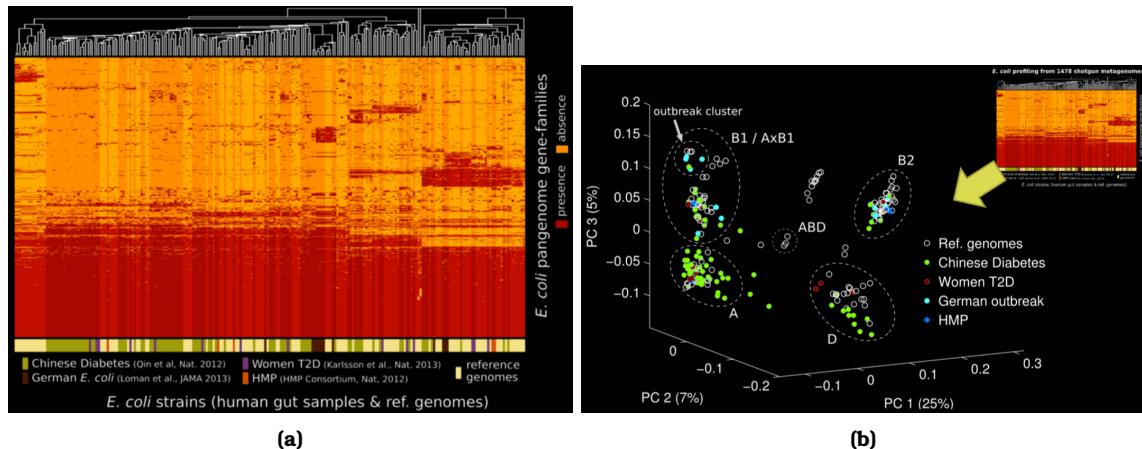


Figure 8.13: *E.coli* population genomics with PanPhlAn

8.5. PANPHLAN: STRAIN-LEVEL PROFILING

8.5.1.2 PanPhlAn on *Eubacterium rectale*

Thanks to PanPhlAn, it was possible to identify many subtypes of *E.rectale* even though only one reference genome was available at the time (8.14).

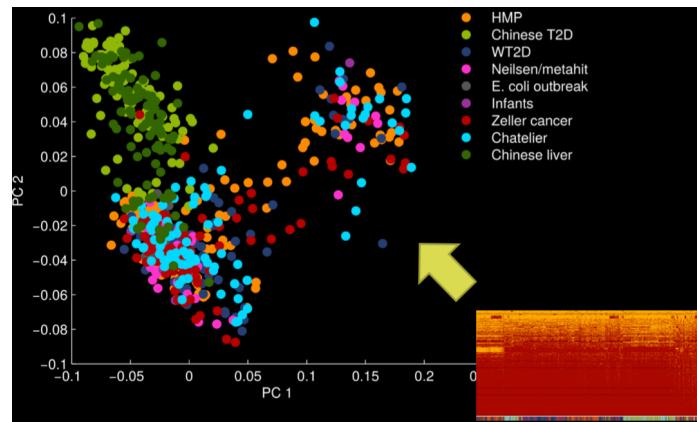


Figure 8.14: PanPhlAn on *Eubacterium rectale*

8.5.1.3 The infant gut microbiome in disease

Ward et al. *Cell Reports* 2016

Necrotizing Enterocolitis (NEC) is a devastating disease that affects mostly the intestines of premature infants. This study took samples from a cohort of 173 infants, 151 of them were preterm and 30 of them had NEC. They obtained 460 shotgun metagenomic samples and 284 shotgun metatranscriptomic samples, all of which were coupled with clinical data of the patients. The heatmap shows MetaPhlAn profiling of different bacterial species for all the samples: about 20% of the patients have a high *E. coli* predominance (8.15a: yellow circle). PanPhlAn was employed to investigate the *E. coli* strains found in these patients: of the 4 identified clades, only 2 are associated with NEC (8.15b).

8.5.2 StrainPhlAn: a complementary approach

As usual, tool by Segata and friends(reference)

The idea is to base the classification on the **genetic variance** of core genomes: look for unique combinations of **SNPs** in genes that are always present and analyze their variance to find some SNPs with a variance different from the others that could characterize a new strain.

StrainPhlAn exploits MetaPhlAn to compute species-level abundances thanks to species-specific markers and then aligns the marker genes present in the samples to find the SNPs. The SNPs are then analyzed to build a phylogenetic tree.

This approach can be applied to many different species. or *E. rectale* it seems to have a higher resolution compared to the previous approach.

8.5. PANPHLAN: STRAIN-LEVEL PROFILING

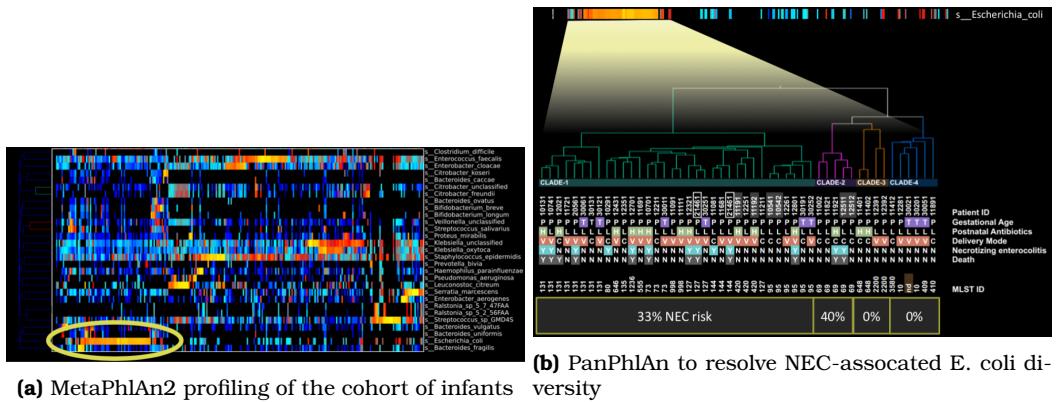


Figure 8.15: The infant gut microbiome in disease

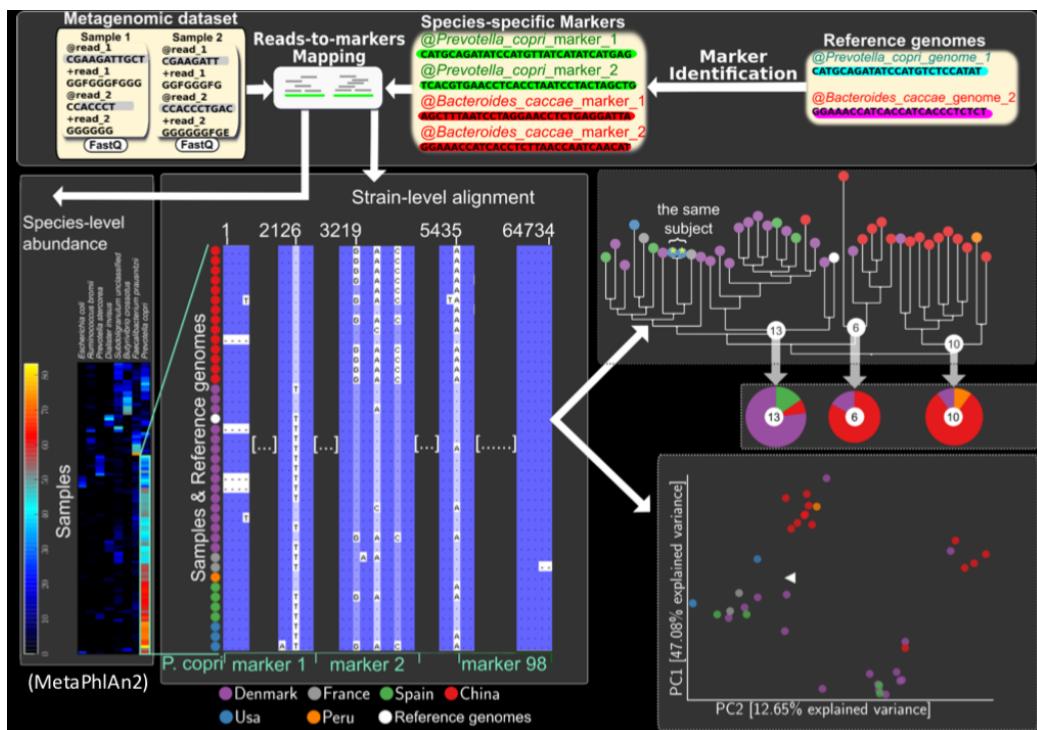


Figure 8.16: StrainPhlAn pipeline

8.5.3 StrainPhlAn applications

8.5.3.1 The stability of strains in the human gut

This study analyzes samples of the human gut microbiome obtained from different continents. The barplots (8.17) show the distances measured between results of SNPs analysis in different regions around the world. There are almost no pairs with zero or low distance

8.5. PANPHLAN: STRAIN-LEVEL PROFILING

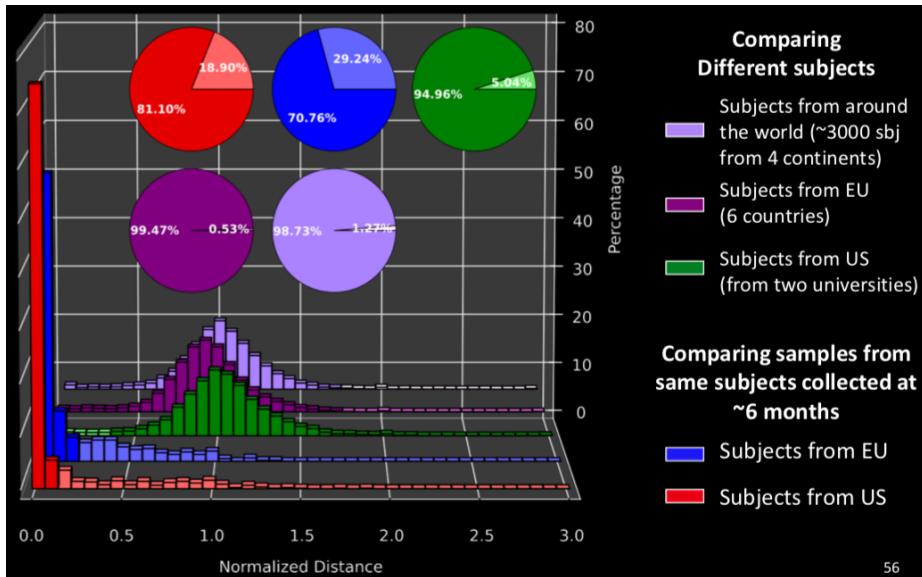


Figure 8.17: The stability of strains in the human gut

and the results do not change if only subjects from the EU or US are considered. On the other hand, the red and blue barplots (8.17) show that comparing the SNPs of samples coming from the same individual but collected 6 months apart they show great similarity. This indicates that there is some stability in the human gut microbiome: usually the strains present in one's gut microbiome tend to remain almost the same. Nevertheless, changes of diet or other habits can bring variations in their abundances.

8.5.3.2 Identification of subspecies

Some bacteria are strongly represented in the human population's microbiomes and their presence is detected in all continents. Everything changes when looking at **subspecies**: some of them are highly region-specific. In Figure 8.18, each color indicates a different country: it is clear that when looking at subspecies of these common microbes, they are mostly associated with only one country/continent.

8.5.4 Uncharacterized species

A great amount of information about the human is still unknown:

- **Functional unknowns:** genes for which we still do not know the function because they do not match any functional database.
- **Unknown species/strains:** genes not matching any of the known reference genomes.
- **Undetected unknowns:** things we do not know and were not even observed yet.

Westernized metagenomes are more characterized than non-westernized ones:

8.5. PANPHLAN: STRAIN-LEVEL PROFILING

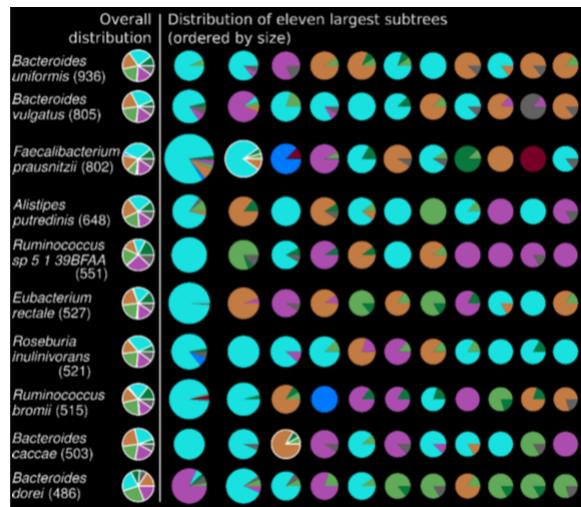


Figure 8.18: Association of sub-species structures with geography

Pasolli, Edoardo et al. "Extensive Unexplored Human Microbiome Diversity Revealed by Over 150,000 Genomes from Metagenomes Spanning Age, Geography, and Lifestyle." Cell vol. 176,3 (2019)

This study performed a large-scale metagenomic assembly on data from uncharacterized regions of the world. They were able to reconstruct 70.000 high quality genomes and 85.000 medium-quality ones (with 50% completeness)

Workflow (8.19):

- Assembly of the reads into contigs
- Binning of contigs into putative genomes (MAGs = Metagenome-Assembled Genomes)
- Quality control

The MAGs are then clustered (also thanks to reference genomes) in **Species-level Genome Bins** (SGBs) by measuring the distance scores between couples of putative genomes. SGBs are divided into known **kSGBs** (if they contain at least one reference genome), unknown **uSGBs** (they are not identified but can be used to detect novel marker genes) and non-human SGBs.

In this study they also characterized a new bacterium: the **Cibiobacter** (this is how they named it but sadly they had to change). Interesting fact: Madagascar-associated strains of Cibiobacter uniquely possess the trp operon for tryptophan metabolism.

8.5.5 Applications of strain-level metagenomic profiling

8.5.5.1 *E.rectale* refined population genomics

*Karcher, N., Pasolli, E., Asnicar, F. et al. Analysis of 1321 *Eubacterium rectale* genomes*

8.5. PANPHLAN: STRAIN-LEVEL PROFILING

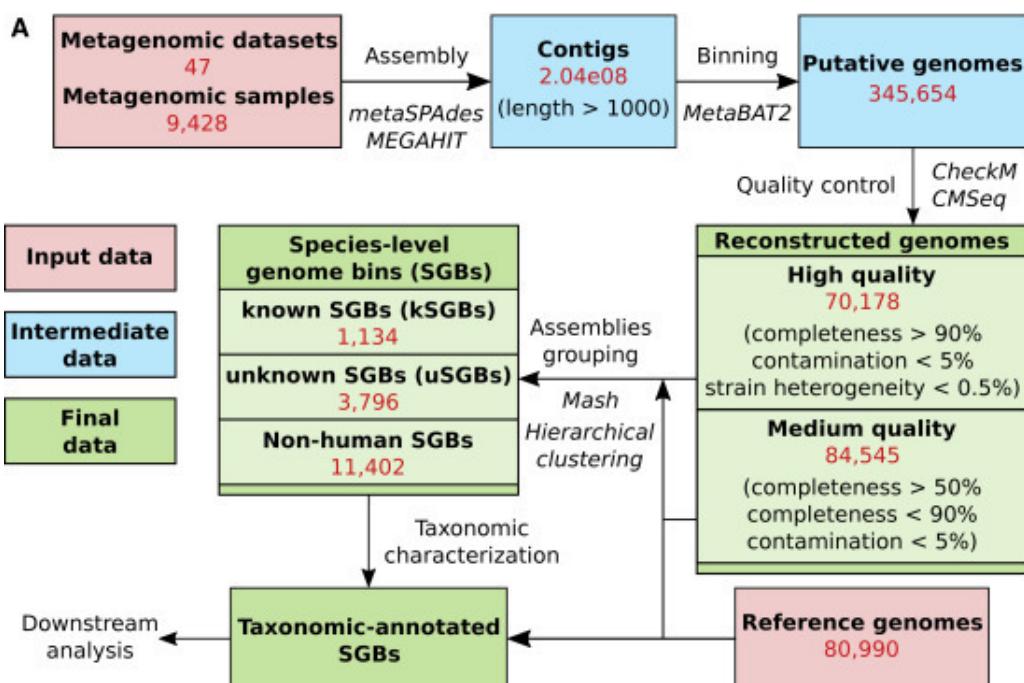


Figure 8.19: Workflow of large-scale metagenomic assembly

from metagenomes uncovers complex phylogeographic population structure and subspecies functional adaptations. *Genome Biol* 21, 138 (2020)

Thanks to the advancements brought by strain-level metagenomic profiling, a new subtype of *E. rectale* was discovered. Subtype 3 lacks the operon coding for motility and other genes that become useless for the bacteria if it is not motile. On the other hand, they present more copies of the **CAZy** genes: these are involved in metabolism and are required by non-motile bacteria in order to be more efficient in the exploitation of carbon sources since they cannot move to reach them.

8.5.5.2 *Prevotella copri* is strongly lifestyle-associated

Tett, Adrian et al. “The *Prevotella copri* Complex Comprises Four Distinct Clades Underrepresented in Westernized Populations.” *Cell host & microbe* vol. 26,5 (2019)

P. copri is a frequent bacterium in the gut microbiome and it tends to be an on/off one: if it is present, it is often dominant. 4 *P. copri* clades with the ability of degrading complex fibers were found. They are called clades just because it is not yet confirmed that they can be considered as new strains. The interesting part is that westernized populations are associated with a lower prevalence of these clades and with a lower probability of presenting all the 4 clades together (8.21). This is probably caused by our diet: westernized populations tend to eat less complex fibers than non-westernized ones. This was confirmed by the analysis of the microbiome found in Otzi (3.300 BC) and some ancient Mexican

8.5. PANPHLAN: STRAIN-LEVEL PROFILING

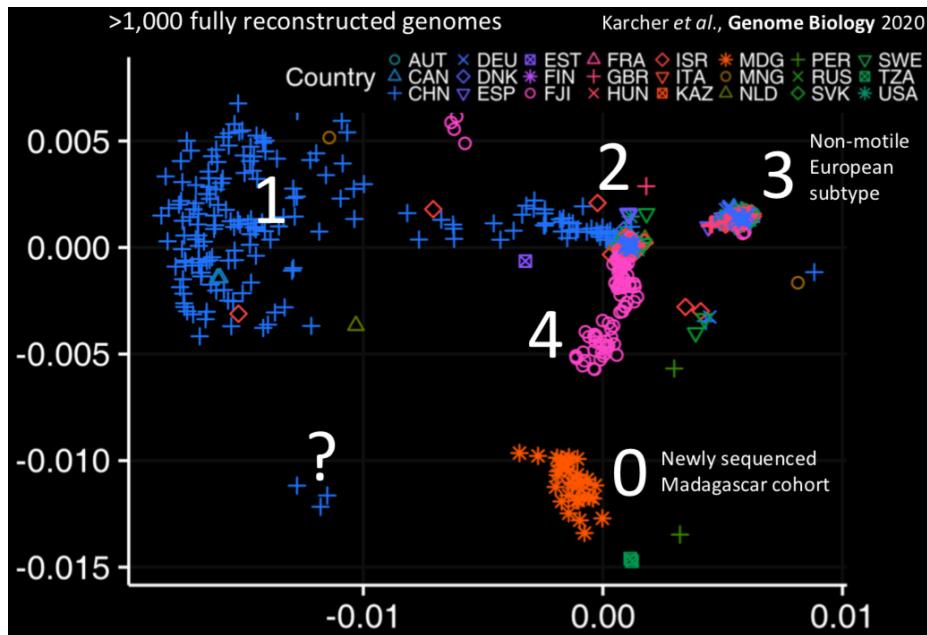


Figure 8.20: *E. rectale* refined population genomics

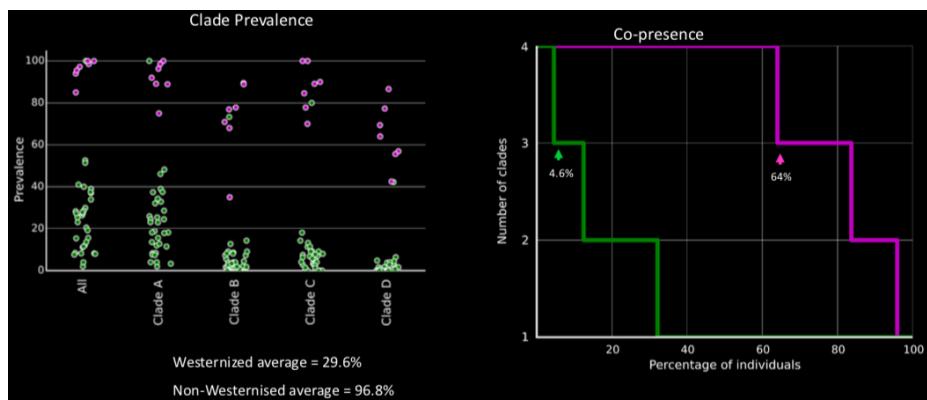


Figure 8.21: *Prevotella copri* is strongly (Western) life-style associated

coprolites (nice term to say cacca mummificata) (600-1300 AD). The *P. copri* clades were found in these samples, indicating that we are possibly losing *P. copri* in the westernized populations.

8.5.5.3 Identification of *Akkermansia* candidate subspecies

Karcher, N., Nigro, E., Punčochář, M. et al. Genomic diversity and ecology of human-associated Akkermansia species in the gut microbiome revealed by extensive metagenomic assembly. *Genome Biol* 22, 209 (2021)

8.5. PANPHLAN: STRAIN-LEVEL PROFILING

Only two subspecies of *Akkermansia* have been described and characterized so far: *A. muciniphila* and *A. glycaniphila*. In this paper, they used PhyloPhlAn3 (Segata's tool for phylogenetic analysis) to identify 4 MAGs that are candidate *Akkermansia* species. Moreover, they observed that these candidate *Akkermansia* species are mutually exclusive for what concerns hosts: they were rarely found coexisting in the same sample. They also appear to have different associations in respect to *A. muciniphila*: one is associated with decreased host body mass index (BMI) but the others are not (8.22).

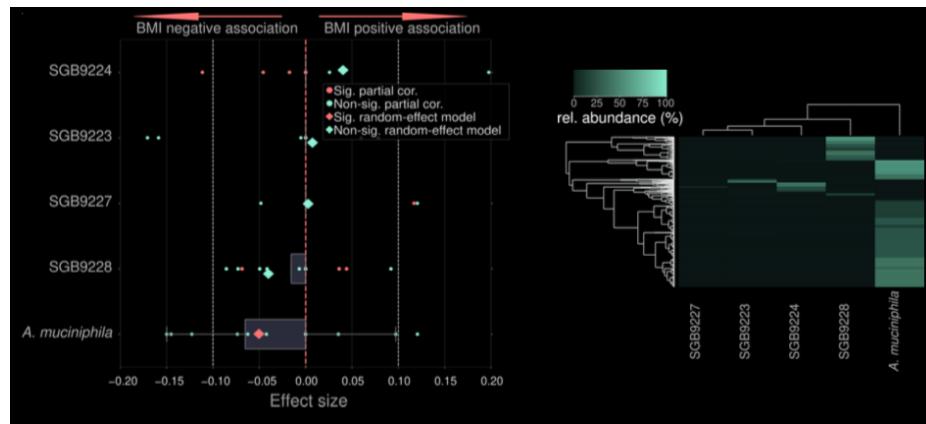


Figure 8.22: Distinct associations and co-exclusion for the *Akkermansia* (candidate) species

They also identified putative bacteriophages with spacer hits from *Akkermansia* candidate species and found that viral detectability correlates strongly with the relative abundance of the *Akkermansia* candidate species, suggesting an intimate ecological interplay.

Analyzing *A. muciniphila* subspecies, they determined that these are host-specific: some are found only in mice and some only in humans.

8.5.5.4 An example of eukaryotic microorganism: *Blastocystis*

Beghini, Francesco et al. "Large-scale comparative metagenomics of *Blastocystis*, a common member of the human gut microbiome." *The ISME journal* vol. 11,12 (2017)

They developed a pipeline to detect *Blastocystis* subtypes and applied it on 12 large datasets composed of 1689 subjects of different geographic origin, disease status and lifestyle.

They confirmed that *Blastocystis* is a component of the healthy gut microbiome and found a higher prevalence in non-westernized individuals. Moreover, they were able to construct and functionally profile 43 new *Blastocystis* genomes.

A strong association of specific microbial communities with *Blastocystis* was confirmed by the high predictability of the microorganism colonization based on the species-level composition of the microbiome.

8.5.5.5 Bacteriophages profiling

Just know that it is possible.

8.5. PANPHLAN: STRAIN-LEVEL PROFILING

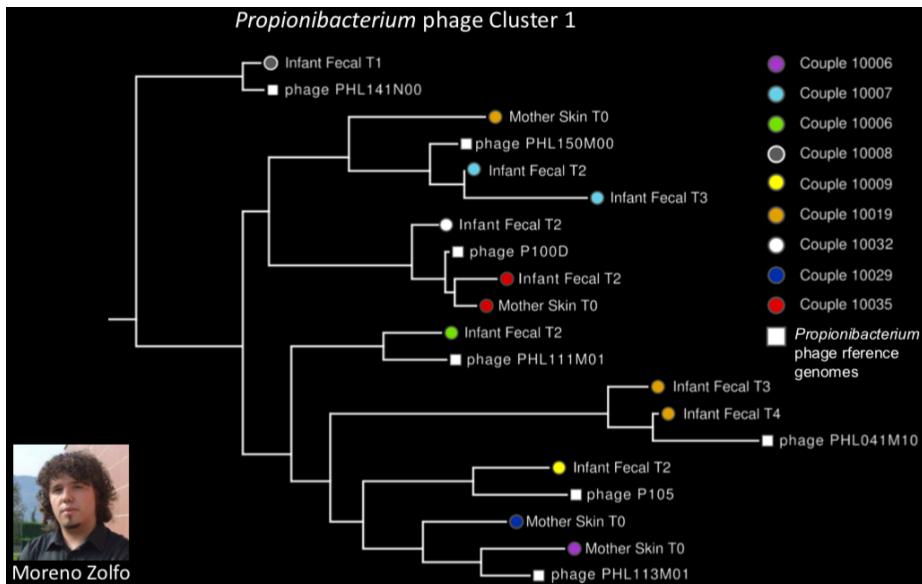


Figure 8.23: *Propionibacterium* phage Cluster 1

8.5.6 HUMAnN2: Functional profiling

The task of mapping nucleotide sequences to proteins (that can be done by blastX on a smaller scale) is important but computationally challenging. Multiple bacteria can be responsible for the same function in the microbiome. Thanks to this redundancy, functions are more conserved than bacterial prevalence: the abundance of a bacteria can vary but its function remains efficient because another microbe supplies it. For functional profiling, the idea is to reduce the reference dataset by mapping the genes only across proteins that we know are present in the bacteria found in the sample. Then the remaining unclassified reads can be mapped to a comprehensive protein database 8.24. HUMAnN2 performs species-level functional profiling of metagenomes and metatranscriptomes.

8.5. PANPHLAN: STRAIN-LEVEL PROFILING

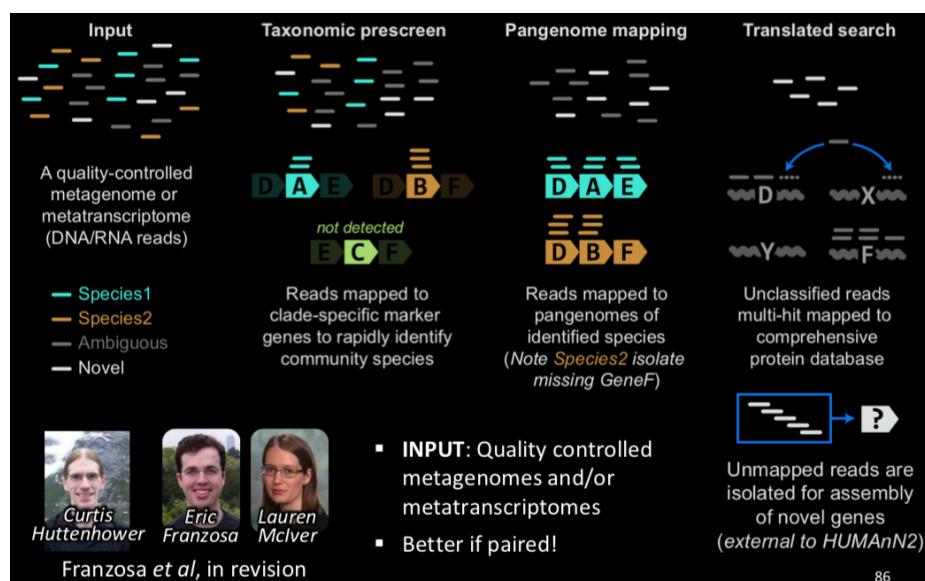


Figure 8.24: HUMAnN2 workflow

Chapter 9

Staphylococcus aureus

Staphylococcus aureus is a gram positive (which means that it has a peptidoglycan layer into the cell wall) and it is a facultative anaerobe bacterium. This is very crucial for *S. aureus* epidemiology because it is able to colonize nostrils where there is oxygen, but it is also able to colonize organs that are inside the body. It is one of the main players in common food poisoning. It is also involved in the menstrual toxic shock syndrome. It plays a key role, also, in other serious disease, like osteomyelitis which is an infection of the bones or sepsis that is a systemic infections. It is a common skin colonizer and for this reason 25% of the people probably have *S. aureus* in their body, but it is also the cause of very bad skin infection. The main reason for *S. aureus* to be tricky to treat because of its immune evasion strategies.

9.1 Immune evasion strategies

S. aureus has two different main strategies that are used in order to stop the immune system of the host from getting rid of it.

1. It prevents the engagement of the host immune system, so it is not recognized by the host immune system. That is done by different proteins that are present on the surface of *S. aureus*.
 - Adhesins bind complement factors to inhibit complement activation cascade.
 - Leukocidins are instead a number of toxins that are selectively killing the adapting immune cells, so killing those immune cells that would be able to kill *S. aureus*.
 - Immunoglobulin binding proteins bind and immobilize IgGs, so they cannot start the cascade of activation of the immune systems.
 - Proteases that cleave the immunoglobulin that are responsible for the activation.

There are four different kinds of classes of proteins that are all trying to hide the *S. aureus* to the host immune system.

2. Overactivation of the non-specific immune system. It is able to trigger a lot of inflammation to cytokine release, that is a non-specific reaction of the body, and also

9.2. ANTIBIOTIC RESISTANCE IN *S. AUREUS*

to facilitate invasion of the so-called non-professional phagocytes, the neutrophiles. There is the production of autholysins, that facilitate invasion of non-professional phagocytes, and super-antigens, that activate T cells and trigger the cytokine release. This could be an advantage to *S. aureus* because when a neutrophile phagocytizes a bacterium or any pathogen can happen that the microbiome is uptaken by the neutrophile, is killed through degranulation and ROS production. The neutrophile then undergoes apoptosis and it is removed by macrophages and so we have a resolution of the infection. If *S. aureus* is present, we have the neutrophile that uptake him. However, *S. aureus* stops the apoptosis of the neutrophile and is able to divide inside it and be screened by the immune system of the host and then, with the leukocidins, it can cause some holes in the neutrophile and also the release of its content outside causing an extra inflammation.

9.2 Antibiotic resistance in *S. aureus*

In the 1940s we have the introduction of penicillin G that is quickly followed by emergence of resistance, the penicillinase. In 1959 was found the methicillin, a semisynthetic penicillinase-resistant β -lactam antibiotics, but in 1960 we have already some cases of resistance to this antibiotic from an hospital in UK (MRSA). In 1960 MRSA emerged in many countries.

9.2.1 Methicillin-resistant *S. aureus* (MRSA)

S. aureus is resistant to β -lactam and is also able to acquire other resistances, even to last resource antibiotic, like vancomycin, linezolid, daptomycin.

Because *S. aureus* is a gram-positive bacterium, the peptidoglycan layer of the cell wall is extremely important for the correct assembly of the cell membrane and the β -lactam antibiotics have the ability to act as substrate analog causing an impaired transpeptidation of the peptidoglycan and the creation of a defective cell wall during cell division.

1. In absence of β -lactam antibiotics, we have the normal cell-wall biosynthesis.
2. In presence of β -lactam antibiotics, we have the binding of the antibiotic to the PBP active site and therefore the peptidoglycan cannot be transpeptidated and the peptidoglycan layer of the cell wall cannot be produced and so we have the cell death during division.
3. In presence of the β -lactam antibiotics, but with mutated PBP (PBP2a, aka MecA), β -lactam is not able to bind the modified PBP. So, the peptidoglycan can be normally transpeptidated and *S. aureus* can produce the cell wall and proliferate.

9.2.2 Methicillin resistance: where is it encoded?

The resistance to methicillin, but more in general to β -lactam, is not encoded on a plasmid. It is encoded on a mobile genetic element that is called staphylococcal chromosome cassette *mec* (SCC*mec*). SCC*mec* are mobile genetic elements that are wide spread across staphylococci genome. They commonly carries genes that might confers some increased

9.2. ANTIBIOTIC RESISTANCE IN *S. AUREUS*

fitness for specific environments. This type of mobile genetic elements can be easily integrated into the genome and also can easily excise from the genome. That means that it is really easy for a MSSA to integrate the mobile genetic element in case of a strong selective pressure that might be given by the presence of antibiotic. When the antibiotic is not more present, it is easy for MSSA to excise the mobile genetic element and return to the basic state of methicillin asset of the *S. aureus*. The mobile genetic element is not maintained inside the cell. It is integrated into the cell or it is excised and released outside.

9.2.3 Methicillin-resistant *S. aureus* (MRSA)

S. aureus is not well recognized for the problems that it causes. There are 80 thousands new patients per year in US that have invasive infections (not colonization), so that means that are people that are sick. The mortality rate is 20%. Hospitalized patients, immune compromise patients or patients with conditions like cystic fibrosis are very exposed to *S. aureus*. This is why, in 2017, the World Health Organization listed *S. aureus* resistant to methicillin and partially resistant to vancomycin or completely as a high priority bug for the research and the development of new antibiotics. It is the fifth one in the list. An article of three years ago estimated about 5 millions deaths associated with bacterial antibiotic resistances (not *S. aureus* only) in 2019. The World Health Organization has estimated that by 2025 there will be 10 millions deaths per year because of antibiotic resistance.

9.2.3.1 *S. aureus* worldwide

There are lots of studies that focused on MRSA or *S. aureus* infections, but the problem is that there are quite biases toward specifically lineages. With the term lineages, we refer to a specific strain or a group of strains that are known to be particularly hyper-virulent or resistant or affecting a specific population (e.g. cystic fibrosis patients). So, only a part of the pool of infections that *S. aureus* can cause. There is a greater underestimation of the strains that are sensitive to methicillin for no reason because if *S. aureus* can be sensitive to methicillin, so it can be resistant to all the other antibiotics. There is a great variability in MRSA that can change epidemiology:

- 60s-70s. There were lots of hospitals associated infections, so people go to the hospital, get a surgery and get *S. aureus*. Also, nowadays *S. aureus* is a key player in post-surgery infections.
- 80s-90s. The community started talking about community-associated *S. aureus* infections and methicillin resistance of *S. aureus* because the study started focusing on the dissemination, also in healthy people.
- 200s. There was a great study on livestock-associated MRSA. Zoonotic infections are pretty relevant, but treatment with antibiotics cause resistances in that community. Livestock diseases and resistances are serious consequences.

9.3. WHOLE GENOME EPIDEMIOLOGY, CHARACTERIZATION, AND PHYLOGENETIC RECONSTRUCTION OF *S. AUREUS* STRAINS IN A PEDIATRIC HOSPITALS

9.3 Whole genome epidemiology, characterization, and phylogenetic reconstruction of *S. aureus* strains in a pediatric hospitals

This work is a full pipeline of what you do if you want to do a survey of the general populations of *S. aureus* in a specific place.

Figure 9.1

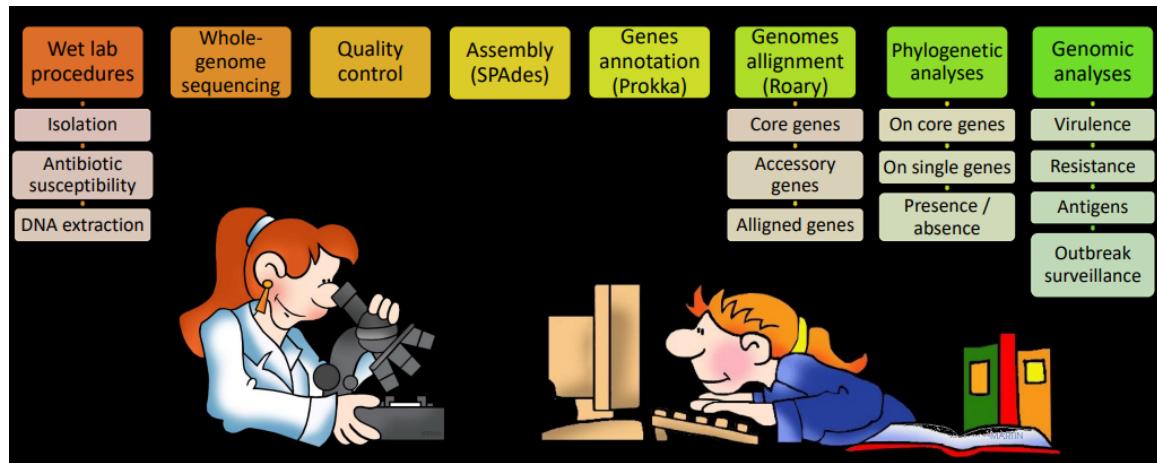


Figure 9.2: Wet and dry lab workflow

9.3.1 Methods

They had 11 operative units that were quite separated.

They started with 234 *S. aureus* isolates:

- antibiotic susceptibility test in vitro
- whole-genome sequencing

After, they selected 184 ($N_{50} > 50k$) so they discarded 50 of the isolates because they want high-quality genome to do a number of analysis. They did not select patients on the base of their infections or the status, because some of these people were not there because of *S. aureus*. People come from different departments such as cystic fibrosis , day hospital, first aid, immunology, infections diseases, intensive care, oncology, pediatrics, respiratory physio-pathology, surgery or week hospital. 135 single patients were selected because them wanted all single patients isolates.

- Average nr of contigs = 51 (12-138)
- Average N_{50} = 206k (50k-970k)

9.3. WHOLE GENOME EPIDEMIOLOGY, CHARACTERIZATION, AND PHYLOGENETIC RECONSTRUCTION OF *S. AUREUS* STRAINS IN A PEDIATRIC HOSPITALS

Also the materials of the samples were different. They had some bronchoaspiration material, sputum, nasal swab, pharyngeal swab, lesion swab and other materials, which included blood and similar.

9.3.2 Typing methods

Usually the typing of *S. aureus* is based on four different typing methods. All these methods have been created for wet-lab work. They are:

1. Multilocus sequence typing (MLST).
2. *S. aureus* protein A (spa). It is one of the major determinant of virulence on *S. aureus*.
3. Staphylococcal cassette chromosome *mec* (SSC*mec*). Looked at the presence or absence.
4. Proton-Valentine Leukocidin (PVL). Looked at the presence or absence.

9.3.3 MLST

1. Characterizing isolates by sequencing fragments of house-keeping genes.
2. Each isolate is characterized by its allelic profile at the house keeping loci → sequence type (ST).
3. Based on multilocus enzyme electrophoresis (MLEE). We should do that for all the genes and because the large amount of PCRs (945) it is not really fast. The allelic profile is much faster since you need only to sequence and then compare.

MLST take fragments of house-keeping genes and look at the sequence of them to assign an allelic profile. That means that if we have 3 strains A, B and C, if we look at the first gene (*abcZ*) and at a some position we have G it will be allele of type 1, while if we have a C it will be of type 2. We do this kind of analysis with all the genes. Then we put the allelic profiles of all the 7 house-keeping genes that are important for *S. aureus* and we have a sequence of numbers that we can compare to a database of sequence types to know of lineage we are looking at.

9.3.4 Spa-typing

1. Single locus DNA-sequencing of the repeat region of the *Staphylococcus* protein gene (spa).
2. Can be used to further discriminate STs.
3. Repeats are assigned a numerical code and the spa-type is deduced from the order of repeats.

The *Staphylococcus* protein A typing looks at the differences in the repeat sequence that is internal in the *Staphylococcus* protein A gene. This gene has a part of repeats that can be in different positions. The order of the repeats is the determinant of the spa-type. They have done 135 PCRs.

9.3. WHOLE GENOME EPIDEMIOLOGY, CHARACTERIZATION, AND PHYLOGENETIC RECONSTRUCTION OF *S. AUREUS* STRAINS IN A PEDIATRIC HOSPITALS

9.3.5 SCCmec

During the sequencing of the region containing *mecA* it was find a distinct mobile genetic element named the staphylococcal chromosome cassette *mec* (SCCmec).

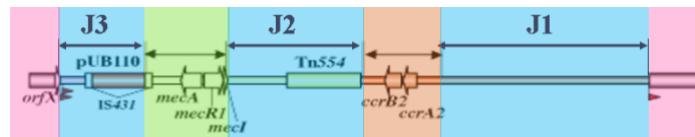


Figure 9.3

1. *mec* gene complex (*mecA* + *mecI* and *mecR*). *mecA* is responsible for the resistance, *mecI* is the inhibitor of *mecA*, *mecR* (1 or 2) is the inhibitor of the inhibitor (*mecI*). *mecA* is always present, the other two might be present or not and that will give a typing of the cassette.
2. Cassette chromosome recombinase (*ccr*) gene complex helps making the typing of the cassette.
3. 3 Joining (J) regions are responsible for other resistances. And can be also used for subtyping regions.
4. Specific inverted and directed repeats containing the insertion site recognized by the *ccr*-encoded recombinase.
5. 11 (from I to XI) types of cassette, different in size and gene content
 - SCCmec typing on *mec* and *ccr* gene complexes
 - Subtyping on J regions

There were made 675 PCRs.

9.3.6 PVL

PVL is a regarded as an important indicator of *S. aureus* virulence. The PVL factor is encoded in a prophage that secretes two toxins LukS-PV and LukF-PV. It is a good indicator of how invasive an infections can be.

- assemble in the membrane of host white blood cells, monocytes, and macrophages.
- form a ring with a central pore through which immune-cell contents leak
- the ring also acts as a superantigen and we have the suppression of adaptive immune response.

There were made 135 PCRs. In total were made 1890 PCRs to get the typing of the community.

9.3. WHOLE GENOME EPIDEMIOLOGY, CHARACTERIZATION, AND PHYLOGENETIC RECONSTRUCTION OF *S. AUREUS* STRAINS IN A PEDIATRIC HOSPITALS

9.3.7 The cohort

There are 1464 core genes that are present in at least 99% of the strains and there are some trees that are quite specific and they matched very well with the MLST typing. We have lot of information about the sample type, the operative unit, the PVL presence, the SCCmec type and also the presence of virulence genes. For virulence genes they made a list of all the very well known genes that are thought to be virulence genes in *S. aureus* and they just reported the presence or absence of them. In this little cohort there were:

- 28 STs (14 CCs)
- 41 *spa*-types
- 4 SCCmec types
- 27.4% PVL+

There are 8373 core genes in total that are present in at least 1 isolates.

9.3.8 Typing highlights common clones and newly sequenced ones

9.3.9 Co-presence of local, global, animal-associated and hypervirulent clones

Lineage	Epidemic clone	# of isolates	Notes
ST228-I	South German / Italian	16	Regional
ST22-IV	E-MRSA-15	13	
ST1-IV	USA400 / CA-MRSA-7	11 (5 var)	
ST5-IV	USA800	10	Paediatric
ST8-IV (PVL-)	USA500 / E-MRSA-2/6	4	
ST8-IV (PVL+)	USA300	2	Highly virulent
ST45-IV	Berlin / USA600	2	Highly virulent (bacteremia)
ST152-V	Balkan clone	2	Highly virulent
ST5-II	USA100 / New York / Japan	1	
ST247-I	Iberian / E-MRSA-5 clone	1	
ST5-I	E-MRSA-3	1	

Figure 9.4

1. Highly virulent STs

- USA300 ST8-SCCmecIV PVL+
- ST239 HA-MRSA → high transmissibility and quickly develops into bacteria
- ST45-SCCmecIV → bacteremia, MSSA isolated from infectious diseases
- ST121 MSSA → from lesion swabs
- ST152-SCCmecV → severe infections

2. Livestock-associated MRSA (LA-MRSA), like ST398 and ST97. Increasing in non at-risk individuals

9.3. WHOLE GENOME EPIDEMIOLOGY, CHARACTERIZATION, AND PHYLOGENETIC RECONSTRUCTION OF *S. AUREUS* STRAINS IN A PEDIATRIC HOSPITALS

3. They found also ST395 lineage that is very peculiar and it is usually not found in humans. It was found in a child that was not at risk. It is particularly interesting because it can exchange DNA with the coagulase negative *S. aureus* (CoNS) because it has modified wall teichoic (WTA).

Livestock-associated MRSA (LA-MRSA) cause mastitis. They found also in children that were not exposed.

9.3.10 Genomic signature of chronic versus acute *S. aureus* infections

Correlation of specific departments with virulent STs and PVL+:

- CF + intensive care → PVL + ST121
- first aid + infectious diseases → PVL
- infectious diseases → ST45

Sample types:

- lesion swabs → MSSA, ST121 and PVL. Here they found virulent and not resistant infections, and that make sense because it is an acute infection.
- Lung isolates (bronchoaspiration material + sputum) → ST128, PVL and MSSA.

They observed that chronic infections are usually less virulent, while normally acute infections are more virulent.

9.3.11 Variability in SCCmec cassettes

They took cassettes and they did genomics analysis to specifically check the genes that are present. When you focused on a specific part of the genome, you can look in depth which genes are present or absent and also the functions of each genes. They found two cassettes harboring extra genes that were resistant to antibiotics (kanamycin, bleomycin and trimethoprim).

9.3.12 Diversity of virulence factors and antigens

You can look also at specific class of genes, like immune evasion genes. Some immune invasion genes are present in almost all of the isolates. The resistant to vancomycin is never present. But it is present the resistant to penicillin. There is only one sample (first aid) positive for Edin (epidermal cell differentiation inhibitor) → translocation into the bloodstream. One USA300 isolate positive for the arginine catabolic mobile element (ACME) and that increased the pathogenicity. Higher prevalence of:

- resistance genes in chronic infections
- virulence genes in acute infections

Toxins primarily responsible for *S. aureus* skin manifestations (Eta and Etb) were strongly associated with ST121 and lesions. Staphylococcal enterotoxins are present in infections department, but are not also present in CF and intensive care departments.

9.3. WHOLE GENOME EPIDEMIOLOGY, CHARACTERIZATION, AND PHYLOGENETIC RECONSTRUCTION OF *S. AUREUS* STRAINS IN A PEDIATRIC HOSPITALS

9.3.13 Virulence factors with available vaccines targets

There are no vaccine approved now for *S. aureus*. They took a list of genes that encode for the target of these vaccines and they checked for the prevalence in the community of isolates and also the presence of SNPs or deletions. There are different strategies for the development of vaccine:

1. highly prevalent genes, but with an high degree of variability/indels
2. most virulent or lethal infections
3. non-virulence genes, more prevalent and conserved than virulence ones

The mentioned antigens are part of the formulation of putative vaccines tested in published clinical trial, with only a few of them getting favorable results and no approved vaccine to date.

9.3.14 Phylogenetic of specific STs highlights the aggressive spread of a novel independently acquired ST1 clone

They investigated the hypothesis that some of the prevalent STs could be hospital-associated clones:

1. isolates sharing the same ST, SCCmec, and spa types, were not monophyletic subtrees when considering external genomes for the same STs. There is an independent acquisition of the clones and there is no evidence of transmission in the hospital.
2. two ST121 MSSA isolates from two patients in the same time window were found to be almost identical.
3. all but two isolates belonged to the same sub-lineage, typed as SCCmecIV t127 PVL-.

It was used a Bayesian phylogenetic modelling approach integrating in the analysis all the ST1 reference genomes publicly available and the two ST1 SCCmecV:

1. Meyer's clone emerged 6 to 28 years ago as a specific branch of the ST1 tree (26-160 years old)
2. An isolate obtained in a recent study investigating the spread of a ST1 SCCmecIV t127 clone in Irish hospitals and carrying a virulence and resistance profile very close to the one of our cohort (differences in gene presence: 2/79 and 0/18 respectively) is phylogenetically rooted inside the Meyer's cluster

ST1 SCCmecIV t127 is not specific of the Meyer's hospital, but might represent a newly arising community clone that is now spreading in the nosocomial environment of different countries.

9.3.15 Conclusions

With a whole genome sequencing approach we can:

1. type and phylogenetically reconstruct a large *S. aureus* cohort
 - observe emerging / unexpected clones
 - spot potential outbreaks
2. test for antibiotic resistances and virulence factors
3. discovery of variants in genes of interest or of unknown relevant genes
4. track strain transmission among patients