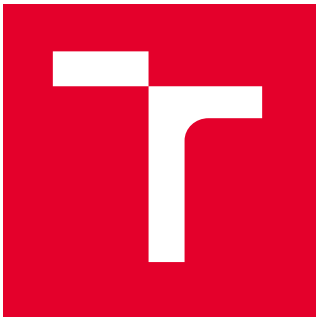

BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

TYPING OF BACTERIAL POPULATIONS BASED ON METHYLATION SITE DETECTION

TYPIZACE BAKTERIÁLNÍCH POPULACÍ NA ZÁKLADĚ DETEKCE METYLAČNÍCH MÍST

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR — **Kristína Hlavatá**
AUTOR PRÁCE

SUPERVISOR — **Ing. Helena Škutková, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2023

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Bachelor's Thesis

Bachelor's study program **Biomedical Technology and Bioinformatics**

Department of Biomedical Engineering

Student: Kristína Hlavatá ***ID:*** 227225

Year of study: 3 ***Academic year:*** 2022/23

TITLE OF THESIS:

Typing of bacterial populations based on methylation site detection

INSTRUCTION:

1) Learn about the occurrence and significance of DNA sequence methylation. Conduct a review of the literature on methods and tools freely available for the detection of methylated sites in DNA sequence. 2) Learn about methods and freely available tools for assembling sequencing data from third-generation sequencers. 3) Perform assembly of provided bacterial genomes. 4) Propose a methodological approach for classifying bacterial strains based on the specificity of occurrence of detected methylation sites in the genomes. 5) Perform detection of methylation sites in bacterial genomes using freely available tools. 6) Evaluate the similarity of bacterial strains based on the profile of detected methylated positions in bacterial genomes. Discuss the results.

RECOMMENDED LITERATURE:

[1] LIU, Y., W. ROSIKIEWICZ, Z. PAN, et al. DNA methylation-calling tools for Oxford Nanopore sequencing: a survey and human epigenome-wide evaluation. Genome Biology. 2021, 22(1), 295. ISSN 1474-760X.

[2] RAND, A. C., M. JAIN AND J. M. EIZENGA. Mapping DNA methylation with high-throughput nanopore sequencing 2017, 14(4), 411-413. ISSN 1548-7091

Date of project specification: 6.2.2023 ***Deadline for submission:*** 14.8.2023

Supervisor: Ing. Helena Škutková, Ph.D.

doc. Ing. Jana Kolářová, Ph.D.
Chair of study program board

ABSTRACT

This bachelor's thesis focuses on the detection of DNA methylations and the development of a methodology for typing bacterial strains. DNA methylations play a crucial role as a regulatory mechanism in the genome, influencing the final characteristics of organisms. We employed DeepSignal2 to detect methylation patterns in 10 strains of *Klebsiella pneumoniae*. Furthermore, we designed a typing method based on the identified methylations to categorize the bacterial strains. This thesis contributes to the improvement of our understanding of regulatory mechanisms in bacterial genomes and presents a novel approach for typing strains using DNA methylation patterns. It provides valuable insights into the characterization and classification of bacterial strains based on their methylomes.

KEYWORDS

methylation, DNA, genome, sequencing, detection, genome assembly

ABSTRAKT

Táto bakalárska práca sa zameriava na detekciu metylácií DNA a vývoj metodiky typizácie bakteriálnych kmeňov. DNA metylácie hrajú kľúčovú úlohu ako regulačný mechanizmus v genóme, ktorý ovplyvňuje konečné vlastnosti organizmov. Použili sme DeepSignal2 na detekciu metylačných vzorov v 10 kmeňoch *Klebsielly pneumoniae*. Okrem toho sme navrhli metódu typizácie na základe identifikovaných metylácií pre kategorizáciu bakteriálnych kmeňov. Táto práca prispieva k zlepšeniu nášeho chápania regulačných mechanizmov v bakteriálnych genómoch a predstavuje nový prístup k typizácii kmeňov pomocou vzorov metylácie DNA. Poskytuje cenné poznatky o charakterizácii a klasifikácii bakteriálnych kmeňov na základe ich metylómov.

KĽÚČOVÉ SLOVÁ

metylácia, DNA, genóm, sekvenovanie, detekcia, skladanie genómu

ROZŠÍRENÝ ABSTRAKT

Táto bakalárska práca sa zaoberá detekciou metylácií DNA a vývojom metodiky typizácie bakteriálnych kmeňov *Klebsielly pneumoniae*. DNA metylácie hrajú kľúčovú úlohu ako regulačný mechanizmus v genóme a majú potenciál dopomôcť k porozumeniu expresie génov. Ide o techniku známu niekoľko desaťročí, no až v poslednom období došlo k jej významnejšiemu skúmaniu a aplikácii. Téma ma zaujala jej potenciálom využitia v nemocničnom prostredí, kde by mohla prispieť k efektívnejšej diagnostike pacientov a následnému nastaveniu liečby, čo ma viedlo k hlbšiemu skúmaniu tejto oblasti. Cieľom je vytvorenie práce, ktorá by vystihovala podstatu problematiky spolu s praktickými ukážkami pre lepšie pochopenie praktického využitia detekcie metylácií.

Práca pozostáva z dvoch hlavných častí. Prvá časť predstavuje teoretické pozadie, nevyhnutné pre pochopenie témy. Postupne prechádza k praktickej časti, kde sú opísané použité nástroje a ich porovnanie. Posledná časť popisuje samotný postup, návrh riešenia a dôvod výberu nástrojov použitých pre účely tejto práce za účelom vyhodnotiť podobnosti použitých bakteriálnych kmeňov.

Na extrakciu barcodeov bol použitý nástroj `guppy_barcoder`. Primárnym cieľom tu bolo vygenerovať súbor `barcodes_summary.txt`, ktorý obsahuje dôležité metadáta o barcodeoch pre každý kmeň. Ďalším krokom bolo basecallovanie, kde sa použil nástroj Guppy basecaller. Kombinácia Guppy a GPU priniesla vynikajúcu rýchlosť a efektivitu v porovnaní s inými nástrojmi, ktoré boli zvažované. Pre resquiggling bol zvolený nástroj Tombo. Výber Tomba bol podmienený skutočnosťou, že pre detekciu metylácií mal byť použitý nástroj DeepSignal2, ktorý vyžaduje predspracovanie údajov pomocou nástroja Tombo. Toto predspracovanie je nevyhnutné na prípravu údajov pre presnú a spoľahlivú analýzu pomocou DeepSignal2. Posledným krokom bola samotná detekcia metylácií pre ktorú bol použitý už spomínaný DeepSignal2. Na základe porovnania dostupných nástrojov bolo vidieť, že DeepSignal2 poskytuje najlepšie výsledky v rámci kompatibilných nástrojov.

Získané dáta bolo pre hlbšiu analýzu potrebné odfiltrovať. Ako kritériá pri predspracovaní bolo použité pokrytie, frekvencia výskytu a pravdepodobnosť výskytu metylácie pre konkrétnu pozíciu.

Odfiltrované dáta mohli byť ďalej použité pre analýzu. V tejto časti bola vytvorená matica pozostávajúca z jednotiek (prítomnosť metylácie) a núl (neprítomnosť metylácie) pre každú pozíciu a kmeň. Táto matica vytvorila pozičné zarovnanie detekovaných pozícií a pre presnejšie výsledky bola doplnená o hodnoty "None" na miesta, kde boli prítomné delécie.

Posledným krokom analýzy bol výpočet distančnej matice a vykreslenie dendroframu. Metrika pre výpočet distančnej matice, bola na základe charakteru dát zvolená ako Hamminogva vzdialenosť. Takto pripravená matica bola následne po-

mocou zhlukovacej metódy UPGMA vykreslená do dendrogramu.

Zo získaných údajov je zjavné, že naprieč všetkými kmeňmi existuje trend medzi zvyšujúcou sa pravdepodobnosťou výskytu metylácie a ich počtom. Z tohto porovnania je navyše možné pozorovať, že počet detekovaných pozícií súvisí so spôsobom sekvenovania. U kmeňov ktoré boli sekvenované samostatne je vidieť vyšší počet detekovaných pozícií ako u kmeňov ktoré boli multiplexované. Kedže ide o experimentálnu tému, výsledky neboli vopred známe. Predpokladali sme však, že kmene s rovnakým sekvenčným typom by mali tvoriť jeden zhluk. Tento predpoklad sa potvrdil priradením kmeňov KP1179, KP1193 a KP1228, majúcich rovnaký sekvenčný typ, do jedného zhluku. Na základe toho možno predpokladať, že metóda detekcie metylácií by sa mohla použiť na typizáciu. Priradenie kmeňov KP687 a KP387 do toho istého zhluku napriek rozdielnym sekvenčným typom zasa naznačuje, že typizácia na základe metylácií by mohla priniesť k ešte presnejšiemu porovnaniu podobnosti medzi kmeňmi.

Snažila som sa túto prácu spracovať tak, aby bola zrozumiteľná a poskytla praktické riešenie spôsobom, ktorý je čitateľ schopný zreprodukovať a zároveň porozumieť jednotlivým krokom. Ide o experimentálnu oblasť, ktorej potenciál je značný a preto verím, že táto práca bude inšpiráciou pre ďalších.

KRISTÍNA, Hlavatá. *Typing of bacterial populations based on methylation site detection*. Brno: Brno University of Technology, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2023, 54 p. Bachelor's Thesis. Advised by Ing. Helena Škutková, Ph.D.

Author's Declaration

Author:	Hlavatá Kristína
Author's ID:	227225
Paper type:	Bachelor's Thesis
Academic year:	2022/23
Topic:	Typing of bacterial populations based on methylation site detection

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno ....

author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

I would like to express my sincerest thanks to the supervisor of my bachelor's thesis, Ing. Helena Škutková, Ph.D. for her valuable advice, useful comments, and the time she spent helping me.

Contents

List of Figures

List of Tables

Introduction

Methylation detection is a technique known for decades, but its contribution began to be more significantly investigated only recently. The topic intrigued me with its potential, for example, in a hospital setting where it could contribute to more effective diagnosis of patients and subsequent treatment settings, which led me to explore this area further. The aim is to produce a thesis that captures the subject's essence and practical demonstrations for a better understanding of the practical application of methylation detection.

DNA methylations have a great potential to help us to understand the level of expression and, consequently, their function. Based on that, we could predict the behaviour of different bacterial strains, which is particularly important in today's rapidly mutating bacterial populations, especially in a hospital environment. Standard laboratory methods were often time-consuming and not consistently effective. This is where typing based on methylation site detection would help. This approach shows promise for faster and more accurate typing of bacterial populations. It serves as an example of how the combination of biology and informatics can replace time-consuming methods and eliminate errors caused by human factors.

This thesis aims to present the DNA methylation topic and everything important to understand it. It consists of two main parts. First, the theoretical background has the task of summarizing the necessary knowledge for understanding the topic. In this section are also presented the different tools that can be used for each step. The central part is the practical part, which describes the chosen procedure. It consists of a methodological approach describing the individual detection steps, preprocessing procedure and data analysis.

1 Bacterial DNA

Deoxyribonucleic acid (DNA) is present in almost all living organisms and serves as the carrier of genetic information. It is the sole molecule responsible for storing and transmitting genetic information, making it an essential heredity component. [1]

1.1 Structure

Just like eucaryotic organisms, bacteria have encoded their genetic information in DNA. DNA represents a double helix with nucleotide bases. These bases are complementary and pair conventionally: Adenine (A) pairs with Thymine (T) and Cytosine (C) with Guanine (G). The bacterial chromosome consists of a single circular molecule which is part of the nucleoid. Nucleoid represents an irregularly shaped structure consisting of a chromosome, several proteins and RNA molecules placed in the cytoplasm of bacteria. [1] [2]

Moreover, plasmids can be found within the chromosome called extrachromosomal genetic elements. Plasmids are small circular DNA molecules picked from another bacterial cell or the environment. They often determine bacteria functions such as antibiotic resistance. Only a few genes can be found within the plasmid, but they are not commonly used. However, plasmids play an important role in survival in stressful situations, and bacteria with those plasmids are more likely to survive. [2] [3]

1.2 Replication

Replication is the copying of DNA strands. This function is present in all the domains of life, including bacteria. Despite many differences, we can find not a few similarities. For instance, in every organism, replication starts from a specific location called the origin. It is the location with unique proteins that initiate the replication process. The difference is in the number of these origins. While eucaryotes have many origins, in most bacteria, we meet only one origin. [4]

Bacteria have well-defined origins with AT-rich areas. Along with the bound of initiator proteins (also called origin-binding proteins), the replication can begin. These proteins are composed of AAA+ subunits. Specifically, it is DnaA, DnaB, DnaC and DnaG. DnaA binds with ATP and enables the opening of the double helix. The result of this opening is a single-stranded DNA (ssDNA) bubble, as shown in Figure 1.1. This bubble enables helicases to load onto DNA. DnaB represents the helicase but can be loaded only with the presence of DnaC. DnaC is a loader that opens the helicase ring, enables binding, and then is released. The replication can

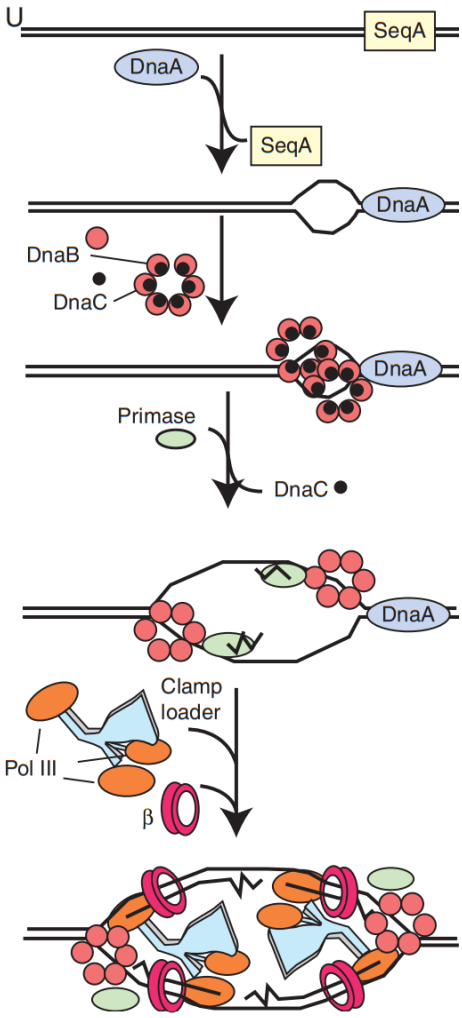

Fig. 1.1: The scheme of replication in bacteria: The replication process goes from the top to the bottom. The top shows the DNA strand followed by origin activation using each initiator protein. [4]

continue in two directions thanks to the replication fork resulting in two daughter strands. These strands are different because of the replication direction. The first strand is called a leading strand, which synthesises at high speed, always in a 5´ to 3´ direction. Unfortunately, the second strand, also called a lagging strand, replicates in the opposite direction as numerous short Okazaki fragments. Here comes the DnaG. The DnaG carries the RNA polymerase necessary to initiate the synthesis on the lagging strand. After replication, these fragments rejoin with ligases. After successful synthesis, there are two DNA strands, each with one original and one daughter strand. [4] [5]

1.3 Genome

The genome represents all the DNA in an organism. Understand the genes but also non-coding regions. It contains all the information we need for building and maintaining organisms. [6]

The bacterial genome is small, supercoiled and tightly packed. Typically it is classified as a circular chromosome which contains nearly the entire genome. In contrast with the eucaryotic genome, there are no introns, and its organisation reflects functional or regulatory purposes. The next difference is in the coding regions. Bacterial chromosomes use 80-90% to encode proteins in contrast with the human genome, where 98% constitute non-coding regions.[7] Based on this fact, we assume a strong connection between the size of the genome and the number of genes. In addition, by knowing the size of the genome, we can predict the lifestyle of bacteria. Bacteria with small genomes are symbionts dependent on the host, while bacteria with large genomes are free-living or environmental isolates. A closer look at this evidence also explains why the bacterial genome is so compact. Based on the search, host-associated bacteria descend from free-living forms. Through the host, they get most of the nutrients they need, which leads to the reduction of functional genes. Over time these genes mutate and get removed. That is why bacteria have the genome such as they have. [1] [7]

1.4 DNA methylations

DNA methylation represents a process of adding methyl groups to DNA molecules. As it turned out, these methylations are crucial in the epigenetic field which deals with DNA changes without changes in the DNA sequence itself. They are essential not only in the human genome but the bacterial genome. They influence gene activities and affect many biological processes, such as transcription, regulation of gene expression (Fig.1.2) or the interactions between DNA-binding proteins, and many more. [8] [9]

1.4.1 Types of methylations and their function

In the bacterial genome, three types of methylations occur. The 5-methylcytosine (5mC) and N6-methyladenine (6mA) are also in the eucaryotic genome. The third one, N4-methylcytosine (4mC), can be found only in bacteria. In contrast with eucaryotes, bacteria use mainly adenine methylation for signalisation. Based on the methyltransferase (MTase) type, adenine methylations regulate the cell cycle or transcription. Currently, there are two known MTases. DNA adenine methylase

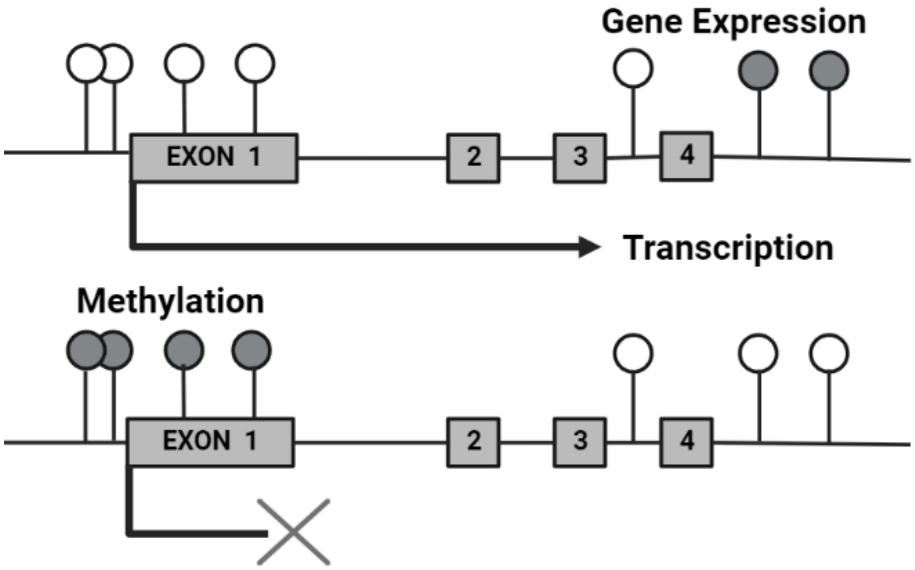

Fig. 1.2: Example of methylation effect on transcription: Squares represent exons, grey circles methylated sites and white circles unmethylated sites. [10]

(Dam) and cell cycle-regulated methylase (CcrM). Specifically, they manage gene regulation and cellular defence through the timing of DNA replication, distribution of newly created daughter chromosomes, repair of DNA. [9] [11] [12]

Dam methylases N-6 of adenine position in GATC sequence. This methylation can specifically alter interactions of regulatory proteins with DNA according to the affinity. CcrM represents a global expression regulator. This MTase methylase N-6 adenine in GANTC sequences. The N stands for any nucleotide. Significant differences between these two MTases are their presence in cell and substrate preference. The difference in preferences is that the CcrM prefers hemimethylated DNA while Dam does not. That is why the CcrM is not processive. The second difference is that the CcrM can be found only in a specific period while Dam is always present. Except for specific effects, they participate in the regulation of virulence together. [9] [12]

1.4.2 Formation of methylations

DNA methylation means adding a methyl group to the DNA molecule. In the 5mC, for example, the methyl group is added to the 5´ position of a cytosine residue (Fig.1.3). The additions are possible only with specific DNA MTases, which catalyse the formation. All of the MTases use S-adenosyl methionine as a source of the methyl group, but there is more than one type of methylation based on the MTases, place of action and effect of the methylation. [9] [12]

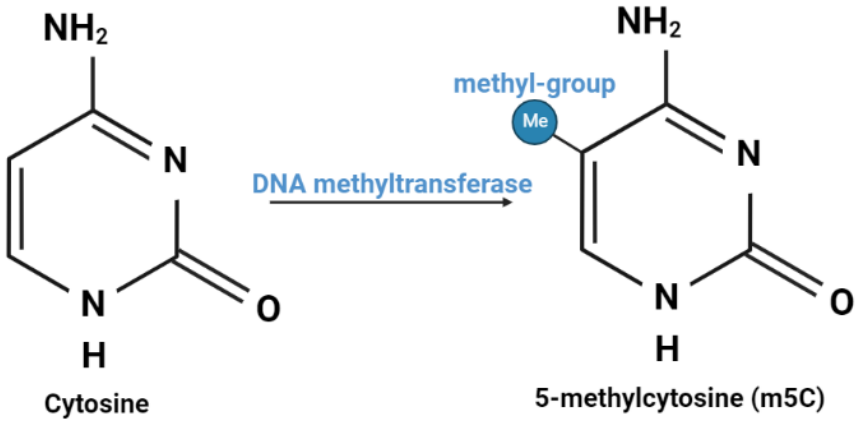

Fig. 1.3: Formation of m5C. [13]

All of the mentioned functions above use the hemimethylated state as a signal. A hemimethylated state means that the DNA strand consists of a parental methylated and a daughter nonmethylated strand. After passing through the DNA replication fork, methylated GATC sites are converted into two hemimethylated DNA duplexes. These duplexes are the opposite. The methylated strand of one is on top and nonmethylated on the bottom and the second duplex is the opposite. Most GATC sites exist in a hemimethylated state only for a short fragment of the cell cycle. They remethylate by Dam. An important note is that these hemimethylations can not be inherited. They occur only transiently in newly replicated DNA. [11]

An example of methylation is the inhibition of DNA replication. SeqA protein, one of the regulatory proteins, takes place here. It binds to the hemimethylated DNA in the GATC sequence, clustered in the origin of replication. Thus the DnaA necessary for replication initiation is blocked. Since methylation is only transient, it is repaired over a certain period. Here comes the methyl-directed mismatch repair protein MutH. This protein can recognise the hemimethylated sites. To ensure the repair, it cuts the nonmethylated daughter DNA strand, so the methylated parental strand serves as a template for repair-associated DNA synthesis. This shows the importance of regulatory proteins, as they can control the onset of methylations. [11] [14]

2 DNA sequencing

Obtaining information about our genome is essential, and sequencing plays a vital role in achieving this goal. By reading individual bases and determining the DNA sequence, we can gather valuable data that can be analyzed in various areas of life, including heredity information and biochemical properties. DNA sequencing has become more accessible in both the scientific and medical fields, being utilized for diagnostic and therapeutic purposes.[15]

2.1 Shotgun vs. amplicon

Before sequencing itself, it is necessary to create libraries because sequencers are not able to process the whole length of a DNA strand. This means creating a lot of DNA fragments which can be sequenced. There are two main principles based on the data. It is possible to create a library from the whole strand or from specific regions of interest. [16]

The shotgun method is also known as shotgun sequencing. However, the shotgun method does not represent sequencing per se. This method is used in order to create a library of fragments which are subsequently sequenced. The first step involves replication of DNA strands or even the entire genome using PCR (Polymerase Chain Reaction). PCR is a widely used molecular biology technique that allows researchers to amplify DNA strands. Subsequently, these copies are randomly broken up into small fragments, which are then sequenced using high-throughput sequencing technologies. The sequencing results in many reads, which are then assembled into longer contigs using computational algorithms. [17] [18]

The shotgun method allows sequencing the entire genome without requiring prior knowledge of the genome's structure or organization. However, sequencing such large areas comes with limitations, particularly when dealing with genomes containing repeated sequences or regions of low complexity. These regions can be difficult to assemble accurately and may result in gaps or errors in the final genome assembly. [17]

Another method to create a library is through an amplicon. This method also refers to the amplification of specific regions by a PCR. Since amplicon primarily works with specific regions, it is necessary to mark them. For this purpose, short DNA primers that are complementary to the sequence are used. These primers enable selective amplification of the desired regions. The resulting amplicons typically range from a few hundred to a few thousand base pairs in length, which fall within the range of read length that sequencers are able to process. The length depends on the size of the target region and the number of amplification cycles. [19]

Amplicons are widely used in various applications, including genome sequencing, genotyping, and mutation analysis. For example, researchers may use amplicons to sequence a specific gene or region of interest in a genome, to genotype individuals for genetic markers, or to detect mutations associated with the disease. The use of amplicons in these applications is facilitated by the ability of PCR to selectively amplify specific regions of DNA with high accuracy and reproducibility. [19]

2.2 Sequencer generations

In recent years, advancements in sequencing technologies have led to the development of three distinct generations of sequencers. The emergence of these sequencers was driven by a realization among scientists that older techniques were no longer sufficient for their research needs. With the ever-increasing importance of sequence knowledge in various fields, researchers needed more precise techniques to unlock the full potential of genomics research. [20]

2.2.1 First-generation sequencing

The first-generation DNA sequencing, which laid the foundations for upcoming platforms, comes with the Sanger dideoxy and the Maxam-Gilbert method. Booth techniques use electrophoresis on a polyacrylamide gel which enables establishing the DNA sequence. Based on the negative charge, according to the fragment length, samples divide in parallel lanes into band patterns. We read these bands from the bottom to the top because the shortest fragments are on the bottom (they are faster), followed by longer ones. The difference is in the fragment preparation. [21]

The Maxam-Gilbert method is more difficult since its technical implementation is more demanding and works with hazardous substances. In contrast with Sanger, the Maxam-Gilbert technique works directly with purified DNA without the previous requirement of ssDNA preparation. Maxam-Gilbert works with chemicals that cleavage the chain - the chemical cleavage technique. The basis of this method is a radioactively labelled chain. This chain is later aliquot into four samples, each with a different chemical that cleaves the chain into smaller fragments. Followed by electrophoresis, the DNA sequence is determined. Although this method was more popular than Sanger's at the time, over the years, Sanger replaced it. [20] [22]

Sanger is nowadays used only for the sequencing of short genome regions we are interested in. These regions are denatured, amplified and labelled. Initially, radioactive labels were used, which were later replaced by fluorescent ones. This eliminated the need for 4 separate samples. The mixture containing a primer, DNA

polymerase, ddNTPs with a specific base (ddATP, ddCTP, ddGTP, ddTTP) fluorescently labelled and natural deoxyribonucleotides (dNTPs). The primer binds first, followed by dNTPs and ddNTPs using DNA polymerase. The ddNTP terminate the elongations and the fluorescent label enables us to evaluate what nucleotide was bound. The binding process is random, resulting in ssDNA strands of different lengths. Electrophoresis then separates these strands, similar to the Maxam-Gilbert method. [22] [23]

2.2.2 Second-generation sequencing

The second-generation, also called the next-generation, came with five new platforms, four commercially available: 454, Illumina, SOLID and Ion Torrent. A huge difference is in the massive parallelisation, which resulted in a reduction in the price and less time-consuming methods. These four platforms differ in chemistries, capabilities and specifications. Although each has its benefits, Illumina has been the most successful and represents the most significant contribution to this generation. [20] [22]

Illumina enables high genome coverage making it suitable even for de novo assembly (assembling without reference). The first step consists of DNA preparation - adding adapters to the DNA fragments, followed by denaturation. Those prepared strands are attached to the surface of a flow cell so the bridge amplification can begin. This amplification is a simultaneous process followed by sequencing. During sequencing, primers, DNA polymerase and four labelled reversible terminators incorporate. Labelling of terminators enables the capture of emitted fluorescence. By repeating this step, we can determine the nucleotide basis and eventually the sequence itself. [22] [24]

2.2.3 Third-generation sequencing

The most recent generation, known as the third-generation, offers superior results compared to previous generations. This generation encompasses several platforms, with the Single Molecule Real-Time (SMRT) platform from Pacific Biosiences and nanopore sequencing from Oxford Nanopore Technologies (ONT) being the most prominent. The main advantage of these platforms lies in their ability to perform single-molecule sequencing, which eliminates the need for DNA amplification that was required by previous generations. Additionally, the third generation surpasses the read length of next-generation sequencing and offers faster and more affordable sequencing. [20] [25]

Oxford Nanopore Technologies brings a low-cost way to approach longer reads in less time with fever difficulties. The most significant advantage over other platforms

is real-time analysis in fully scalable formats and ultra-long reads. ONT even came with quite a large amount of nanopore sequencing devices. This variety enables us to choose the most suitable one for the projects. [26]

This method's key is nanopores since the DNA strands have to come through them. ONT works specifically with protein nanopores because they can be found even in nature in cell membranes, making them more suitable. However, ONT works on solid-state nanopores fabricated from synthetic materials. These nanopores promise improvement in cost and, more importantly, the scale of nanopore analyses. [27]

As mentioned, nanopores are the basis. They include absensors detecting changes in current as the DNA strand comes through. ONT consists of many nanopores embedded in a membrane (Fig. 2.1). The membrane has to be electro-resistant and nanopore electrically connected to a channel and sensor chip. This way, the current can be measured, and so the changes in it. Disruptions in current produce squiggles, characteristic for each base. Squiggles enable us to identify the nucleotide bases and determine the final sequence. [28]

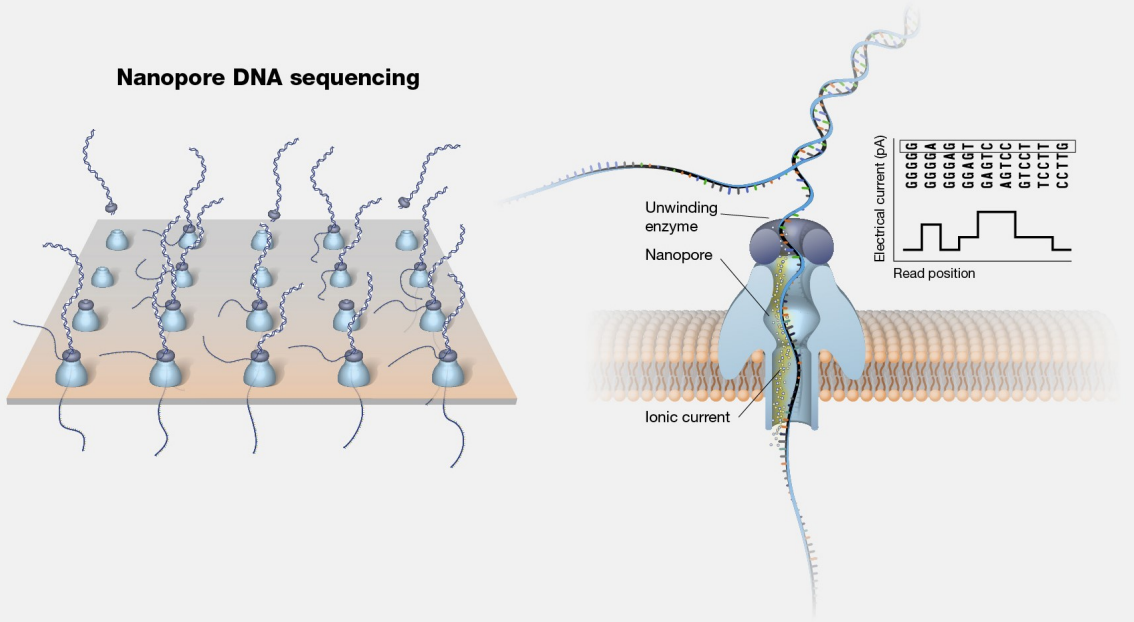

Fig. 2.1: Nanopore DNA sequencing: There are embedded nanopores on the left side. The right side shows a cross-section of nanopores with bonded DNA. The graph above shows squiggles with determined bases. [26]

2.3 Genome assembly

Genome assembly (Fig. 2.2) is the process of piecing together the DNA sequences from a genome, which is the complete set of an organism's genetic material. This process involves taking millions to billions of short DNA fragments and aligning them to create contiguous stretches of DNA that represent the original genome. Genome assembly is a crucial step in genomics research as it provides a complete representation of an organism's genetic material, which can be used for a variety of applications, including understanding genetic variation and evolution, identifying disease-causing mutations, and designing new therapies or treatments. [29]

The assembly process can be challenging because the genome is often very large, with many repeated sequences, and the sequencing data may contain errors or gaps. Various computational algorithms and tools have been developed to aid in the assembly process, including de novo assembly, which involves assembling the genome without using a reference sequence, and reference-based assembly, which involves aligning the sequencing data to a known reference genome. If we are working with large genomes without deeper knowledge or when studying genetic variations within a species, it might be more appropriate to use de novo assembly. On the contrary, if working with a well-known genome, using reference-based assembly would be more accurate. A more detailed comparison is in the Table 2.1 below.[29] [30] [31]

Tab. 2.1: De novo vs. reference-based assembly.

Method	Advantages	Disadvantages
De novo	- no reference - variation in species	- requires high-quality data - time-consuming
Reference-based	- deals with gasps and repetitions -fast assembly	- requires reference - read length limitation

2.3.1 De novo assembly

De novo assembly is a computational process used to reconstruct a complete genome sequence from short DNA fragments without the need for a reference genome. It is a crucial step in genome sequencing projects, especially when dealing with organisms whose reference genomes are not available or when studying genetic variations within a species. [30] [32]

The de novo assembly involves several steps. First, the short DNA fragments are generated from the organism's DNA. These fragments may overlap with each

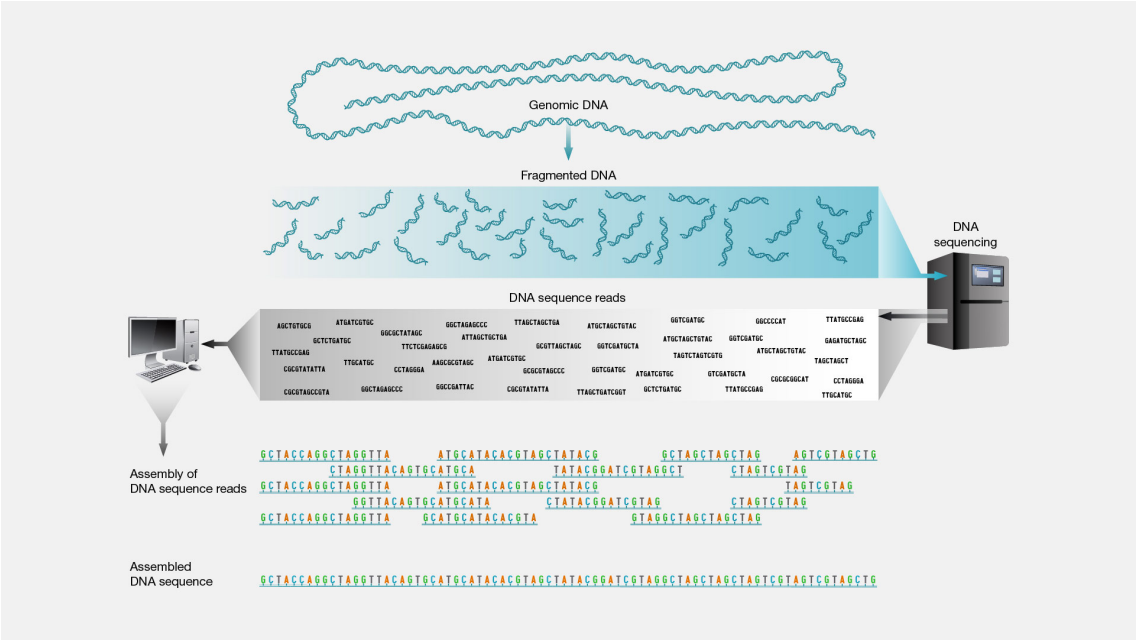

Fig. 2.2: Process of genome assembly: The process of genome assembly from DNA strands, through creating libraries to the final assembled genome. [18]

other, although in varying lengths and coverage depths. After this, the bioinformatics algorithms and software are utilized to analyze the overlapping regions and assemble the fragments into contiguous sequences called contigs. By identifying regions of overlap and utilizing the consensus between overlapping fragments, the assembly algorithm attempts to reconstruct the original genomic sequence. The final output of a de novo assembly is a set of contiguous sequences, representing the reconstructed genome. Although these can be enough, the contigs can be further refined by scaffolding, which involves ordering and orienting the contigs based on additional information. This process helps bridge gaps between contigs and provides insights into the relative positions and orientations of the contigs. [30] [32]

However, due to various challenges like repetitive regions, sequencing errors, and variations in coverage depth, the assembly process can result in misassemblies. To manage this, additional computational tools and experimental techniques can be used to resolve ambiguities and improve assembly accuracy. [33]

There are several tools for de novo assembly based on different algorithms. For example, the De Bruijn graph works with k-mers by splitting sequences. It takes sequences and divides them into smaller k-mers. Another graph-based algorithm is the overlap layout consensus, where similar sequences are overlapped, and the fragments of the graph are packed into contigs. In addition to these, we can also find string graphs and greedy or hybrid algorithms. The application of these algorithms has resulted in several tools such as Flye, Minimap2, Miniasm Velvet, Edena, etc.

[33] [34]

Minimap2 and Miniasm are tools that are commonly used together. While Minimap2 functions as an aligner, Miniasm is a regular de novo assembler. Initially, Minimap2 aligns the reads, enabling subsequent assembly by Miniasm. Miniasm serves as the assembler and employs graph traversal algorithms to identify paths through the overlap graph, which represent potential contigs. The output comprises contigs consisting of continuous and non-repetitive sequences in the genome. [35] [36]

Flye assembler is suitable for ONT long-reads and belongs to one of the most popular de novo assemblers. Flye utilizes a combination of overlap layout consensus and a repeat graph. Unlike other assemblers, Flye also provides polishing of the final consensus. As a result, the error rate has significantly decreased, making this assembler even more popular. Flye requires only raw base-called data, resulting in a consensus sequence in FASTA format. To achieve this, Flye employs multiple steps, starting with the detection of overlaps. Subsequently, the repeat graph is constructed, followed by error correction. The repetitive graph is then analyzed, enabling the resolution of repetitive regions and the generation of the final contigs. [37] [38]

2.3.2 Reference-based assembly

Reference-based assembly, also known as mapping-based assembly or alignment-based assembly, is a computational approach used to reconstruct a genome sequence by mapping and aligning short DNA reads or fragments to a reference genome. [31]

The reference genome is used as a guide to align the sequencing reads or fragments meaning the assembly heavily relies on the quality and accuracy of the reference. In order to get the best results the reference has to be obtained from a closely related species or a well-characterized individual within the same species. Unless a sufficiently high-quality reference is used, the result might contain errors and mistakes. [31] [39]

After the reads are aligned, the resulting alignments are analyzed to identify regions of the reference genome where the reads overlap. These overlapping regions can be used to assemble the reads into contiguous sequences, called contigs, representing the target genome. [39]

2.3.3 Practical implementation of genome assembly

For a genome assembly the Flye (2.8.1, `https://github.com/fenderglass/Flye/blob/flye/docs/USAGE.md`) was used. Flye is a de novo assembler working without a reference sequence plus polishing the final genome to reduce the number of errors.

It needs only basecalled fastq files. According to the documentation, we adapted the code to be as efficient as possible. We specify the coverage length and size of the genome. The command for the coverage length setting is `-asm-coverage`. We set it to 40, which is typically enough for good results. The genome size use `-genome-size` command, and we set it up to 3 million. These settings reduce memory consumption which is excellent for large genomes.

The output provided several folders and files. Most importantly the final assembly. Assembled genomes were stored in fasta format, consisting of several contigs. Their number varies from strain to strain and so do their length and coverage. More detailed information about each contig was in their statistic file. Overall assembly statistics could be found within logs:

```
[2022-12-04 04:36:02] root:  INFO: Assembly statistics:
Total length:  5513573
Fragments:  3
Fragments N50:  5232065
Largest frg:  5232065
Scaffolds:  0
Mean coverage:  982
[2022-12-04 04:36:02] root:  INFO: Final assembly:
path/assembly.fasta
```

The final step involves polishing the assembled sequence. Although Flye performs some level of polishing on the final sequence, it may still contain rough areas which are necessary to be refined to reduce the error rate. For this purpose, we use Medaka (1.7.2), which represents the most recent and updated version. By using Medaka, we eliminate the need for Racon and enhance the overall effectiveness of the polishing process. Medaka requires only basecalled data to carry out the polishing step. The result of Medaka can be found in a file called `consensus.fasta`.

The assembled sequences were not utilized for further analysis. Using the NTUHK2044 strain as a reference for all the strains was deemed to be a more suitable and efficient option. This well-defined reference genome is close to the used strains enabling direct alignment of the strains during re-squiggling. It removes the need to assemble the genomes in order to create references, align the data to its own reference and then align all the genomes together. This process would be very difficult since a de novo assembly was used. It may result in errors caused by individual aligning, which would affect the quality of the resulting alignment.

3 Methodology for methylation detection

Finding methylation in genomes might be essential for further epigenetic analysis. However, to find these methylations it is necessary to follow a certain procedure. On the Internet today we can find many tools for individual steps, but the overall procedure is the same as shown in the Figure 3.1 (B) below. Most available tools are designed to operate with FAST5 files, which consist of raw electronic signals known as squiggles. However, squiggles cannot be directly processed and must first be converted into a text-based file format that represents nucleotides. This process is called base calling. Once the data has been base called, it must then be aligned to a reference sequence and re-squiggled before a detection tool can be applied. It's worth noting that multiple tools and versions are available for each step of the process, and the specific choices will depend on the data and the desired outcomes. [8] [40] [41]

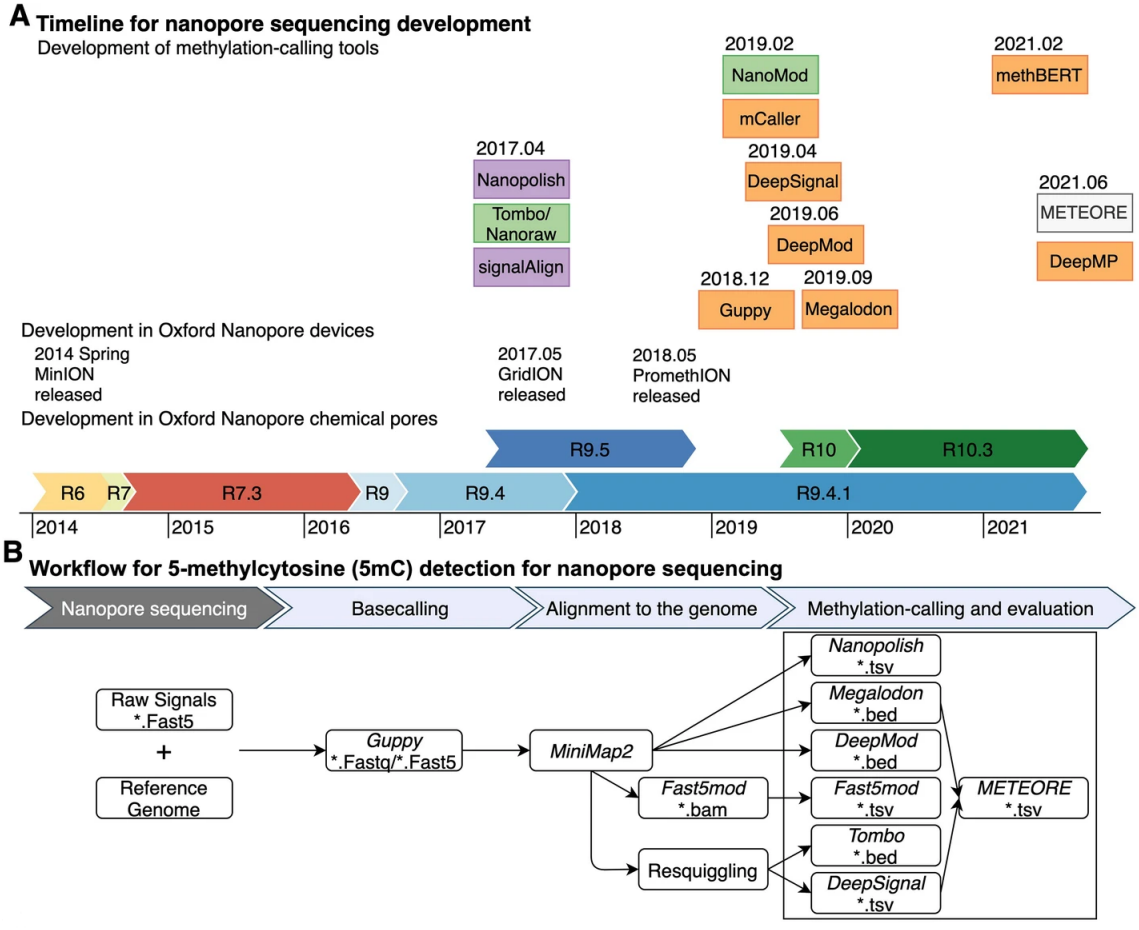

Fig. 3.1: Detection throughout the time and detection workflow. (A): Timeline of detection tools for nanopore. (B): Workflow for 5mC detection consisting of three steps. [41]

3.1 Demultiplexing

In some cases, it is desired to sequence multiple samples in a single run to save time and money. This type of sequencing is known as multiplex sequencing. To do so, it is necessary to tag each sequence to differentiate the sequences that belong together. For this purpose, we use special barcode sequences. The barcodes enable us to assign which sequences belong together and sort them before any further steps. However, the number of barcodes used is limited. For example ONT provides 12 barcodes. They are ligated on both ends of DNA strands, allowing simultaneous sequencing of 12 genomes. [42] [43]

Demultiplexing means sorting reads into files based on barcodes. For this purpose, the internet provides several tools. ONT comes with its own tools, which are the most suitable for data obtained from their sequencers. Some of them are preceded by base calling, another can be run right after sequencing. Besides the process, every tool should assign a sequencing read to the corresponding sample, resulting in files, each containing reads of one sample. After successful demultiplexing, we can select a file with the desired sample based on the list of barcodes to which it should correspond. As mentioned before, ONT provides 12 barcodes, the number of files found at the end of demultiplexing. [43]

Guppy

Guppy is the official basecalling software provided by ONT. Despite basecalling, Guppy provides also demultiplexing with its built-in `guppy_barcoder` module. This module assigns reads to specific samples based on barcode information during the basecalling process resulting in folders separated by barcodes and `barcodes_summary.txt`. Simultaneous basecalling and demultiplexing using `guppy_barcoder` is commonly employed for ONT sequencing data, providing convenience, but it also has its drawbacks. The simultaneous process can increase computational requirements, and it relies on accurate barcode information for successful demultiplexing. [44]

Ont_fast5_api

The `ont_fast5_api` is a Python package from ONT. It provides several scripts that enable the manipulation of fast5 files. For the demultiplexing, there is `demux_fast5` command. It operates on the raw fast5 data in multi-fast5 format generated by the nanopore sequencer. The required input is raw data and barcoding information. During demultiplexing, barcode information is extracted from the

fast5 files and assigned the reads to specific samples based on the barcodes. The process results in raw split data in the original multi-fast5 format. [45]

Deepbinner

Deepbinner is another ONT tool. It is a deep learning-based demultiplexing tool, which utilizes neural networks to predict sample assignments directly from the raw signal data. Thanks to the neural network, Deepbinner can handle demultiplexing tasks efficiently and accurately, even in cases where the barcodes might be challenging to differentiate due to sequencing errors or low-quality data. [43]

3.2 Basecalling

The third-generation sequencers result in electrical signals that we cannot process. It's necessary to convert these electrical signals into a text-based format with which we are able to work. This process is called basecalling and consists of translating the raw electrical signal into a nucleotide basis. [46]

There are several tools available to perform this task based on different algorithms. The oldest one works with Hidden Markov Models including Nanocall or Metrichor. Newer tools came up with the use of deep learning models such as neural networks. Currently, all modern tools utilize neural networks. The ones we are interested in are compatible with ONT sequencers output such as Guppy, Albacore, Scrappie or Flappie. All of these were developed by ONT, specifically for R9.4.1 flowcells. [46] [47]

Scrappie

Scrappie is considered the first modern ONT basecaller and is often referred to as a "technology demonstrator." That is because it served as a platform for testing new approaches before they were incorporated into subsequent tools like Guppy and Albacore. Scrappie comprises two basecallers: Scrappie events and Scrappie raw. Initially, Scrappie events perform event segmentation, followed by Scrappie raw, which conducts basecalling on the raw data. However, Flappie eventually replaced this basecaller with improved read accuracy but not consensus accuracy. [46]

Guppy

In addition to the already mentioned demultiplexing, Guppy also supports basecaling. Guppy basecaller relies on a neural network architecture to perform various

tasks such as filtering reads based on quality, clipping adapter sequences, and estimating the probability of methylations. It is a bi-directional recurrent-based neural network, allowing data to flow back and forth between the network nodes. To use Guppy as a basecaller, fast5 files need to be provided as input. The output of the basecalling process consists of two folders, log files and a sequencing summary text file. Within these folders, the reads are segregated into two groups: high-quality reads are placed in the "pass" folder, while lower-quality reads are placed in the "fail" folder. In our specific case, only the files in the "pass" folder were considered for obtaining the most accurate results. To confirm the successful completion of the process, a similar text as shown below should be obtained either in the command line interface or in a log file: [40]

```
0% 10 20 30 40 50 60 70 80 90 100%
|--|--|--|--|--|--|--|--|--|--|
**************************************************
Finishing up any open output files.
Basecalling completed successfully.
```

Albacore

Albacore is similar to Guppy in many aspects. However, the main difference lies in the utilization of the Graphics Processing Unit (GPU). Unlike Guppy, which leverages GPU acceleration for faster processing, Albacore relies on a Central Processing Unit (CPU), making it comparatively slower in performance. In terms of accuracy, both tools were initially quite similar. However, the development of Albacore took a backseat in favour of Guppy, resulting in Guppy receiving more updates and improvements, leading to better accuracy over time. [46]

3.3 Re-squiggling

After basecalling, there are raw data and basecalled nucleotide bases with sufficient quality. However, these data do not have a connection. To do so they have to be re-squiggled to reconstruct the raw electrical signals with the corresponding base sequence. [48]

Tombo

Tombo re-squiggling is a process in nanopore sequencing data analysis that involves improving the accuracy of basecalling by re-estimating the electrical signal levels associated with each base in the sequence. Tombo relies on fast5 files that

have been previously annotated with fastq files obtained through the base calling process. Fortunately, Tombo provides a convenient `tombo preprocess annotate_raw_with_fastqs` command that can be used to perform this step. Resquiggling can help refine the basecall assignments made by the initial basecaller and improve the overall quality of the sequencing data. [48]

3.4 Methylation detection

Final step to detect methylation is the methylation detection itself. There have been multiple tools for their detection, and new ones are still developing. Older ones used to combine next-generation sequencing and bisulfite conversion, but this combination has many disadvantages. DNA can damage due to bisulfite conversion, and next-generation sequencing results in short-range patterns. Third-generation sequencing overcomes this problem and enables direct methylation detection. Over time, several tools able to call these modifications were developed. Since the data used in this thesis were also obtained from the third-generation sequencer, tools compatible with this data are discussed in more detail. An overview of these tools based on release time and flow cells they are compatible with can be seen in Figure 3.1 (A). A more detailed overview of some of the tools is in the Table 3.1. [8] [41]

These tools employ various methods for predicting methylation states. NanoMod and nanoraw belong to statistics-based methods that analyze two types of reads: native reads and reads from matched amplified DNA. By comparing these two groups, we can predict the methylated state using statistical tests such as the Mann-Whitney U test or Kolmogorov-Smirnov test. Although these methods do not require training, their accuracy is significantly lower compared to model-based methods. Model-based methods, such as Nanopolish, signalAlign, or mCaller, take a different approach. They initially predict the methylation state of individual reads and then aggregate the information to determine the methylation state at a genome level. These methods utilize the Hidden Markov model or its alternative extended by the hierarchical Dirichlet process. However, the most successful approach is the third method, which relies on deep learning. This method is widely used in current tools like DeepSignal, Megalodon, Guppy, DeepMP, and others. With appropriate training, these deep learning-based tools can achieve high accuracy in predicting methylation states. [41] [8]

Nanopolish

Nanopolish is based on Hidden Markov Model, working with signal-level data. With other tools, Nanopolish belongs to one of the most used tools for methylation de-

Tab. 3.1: Overview of methylation tools using neural network. [41]

Tools	DeepSignal	Megalodon	mCaller	Guppy
DNA modifications	5mC, 6mA	5mC, 6mA	6mA	5mC, 6mA
Support multi-read fast5 format	NO	YES	NO	YES
Compatible flow cells	R9, R9.4, R9.4.1	R9.4.1,R10.3	R9, R9.4, R9.5	R7.3, R9, R9.4, R9.4.1, R9.5, R10, R10.3
Required input	Basecalled fast5 processed by Tombo resquiggle module	Raw fast5	Basecaller fast5	Raw fast5
Accuracy	0.9 - 0.92	N/A	0.954	N/A

tection. It has more modules, but only the call-methylation module can detect methylations. The whole detection consists of four steps. First of all, we have to detect events. Events detections use raw signals and create segments which represent the events. This step is based on changes in the signal, followed by the alignment of these segments. We align them to a generic k-mer model signal. Alignment continues with the final calibration. Performing the calibration enables getting the best set of scaling parameters. As the last, we use Hidden Markov Model for the calculation of methylation scores. After all these steps, we get the likelihood of the unmethylated and methylated sequence so the methylations can be detected. [49]

Megalodon

Megalodon is a software tool used for analyzing nanopore sequencing data. It is specifically designed for basecalling and detecting DNA modification in data obtained with ONT sequencing. Megalodon can detect DNA modifications, including DNA methylation, by analyzing the raw nanopore signal data. It utilizes machine learning algorithms to identify modified bases and generate methylation profiles along the DNA sequence. [41] [50]

DeepSignal2

DeepSignal2 is one of the up-to-date tools for methylation detection. In contrast with previous methods, DeepSignal2 can achieve higher performance detecting 6mA and 5mC methylation, achieving accuracy rates of over 90% even with lower coverage. In contrast with signalAlign, DeepSginal2 requires only two sampled reads. The tool operates through two modules: a convolutional neural network and a bidirectional recurrent neural network. The convolutional neural network creates a signal feature module by working with a raw electrical signal, while the bidirectional recurrent neural network creates a sequence feature module based on the sequence of signal information. The features produced by both modules are then sent into a fully connected neural network, which can predict the methylation state. [8]

4 Practical implementation of DNA methylations detection

4.1 Input genome data and sequencing parameters

We are working with 10 different strains of *Klebsiella pneumoniae*. We decided to work with those considering the wide variability of sequencing types (4.1) and sequencing platform settings. *Klebsiella pneumoniae* is a bacteria, common in hospital environments and belongs to one of the most frequently sequenced bacteria, which is another reason to work with these strains. These data were sequenced with Oxford Nanopore Technologie (ONT) which is commonly used for direct methylation detection. As a reference for every strain, we use NTUH-K2044 [51], a reference sequence which is well-annotated and the most similar to used sequencing types. Based on similarity it provided a suitable template to which the methylation positions of other strains were assigned, which made it unnecessary to align individual strains.

The reference genome of NTUH-K2044 was downloaded from the RefSeq database provided by National Center for Biotechnology Information known as NCBI (NC_012731.1, `https://www.ncbi.nlm.nih.gov/refseq/`). These data are in fasta format which is text-based, representing nucleotides or amino acid sequences. The genome size is 5,472,672 bases and consists of 5,293 genes. Other stains were sequenced in collaboration with the Department of Internal Medicine, Hematology and Oncology at the University Hospital Brno. These data were sequenced with MinION with R9.4.1 pores type from ONT which results in raw data in fast5 format. These files consist of a raw electrical signal, which had to be transformed for further processing.

Tab. 4.1: Strains sequencing types.

Strain	Sequencing type	Strain	Sequencing type
EB362	ST321	KP1236	ST397
KP387	ST433	KP1228	ST551
KP1179	ST551	KP1272	ST14
KP1193	ST551	KP1209	ST70
KP1231	ST405	KP687	ST11

4.2 Design of algorithmic workflow

In order to detect methylations, the steps to modify the input data had to be done so that the methylations could be extracted. A description of these individual steps is in the following subsections. The code used for each step is included in the electronic attachments as a `MethylationDetection.sh` script.

The initial stage of the workflow focused on demultiplexing the raw data. Given that the barcodes for the multiplexed data were known, the `demux_fast5` tool was the choice. This tool allowed us to segregate the different sequences based on their barcodes, which is essential for subsequent analysis.

In order to extract the barcodes, the `guppy_barcoder` tool was employed. The primary goal here was to generate the `barcodes_summary.txt` file, which contains crucial metadata about the barcodes associated with each sequence.

The next pivotal step was basecalling, where the Guppy basecaller saw used. A noteworthy aspect of this phase was the utilization of a GPU, which significantly boosted the performance of the Guppy tool. This combination of Guppy and GPU yielded superior speed and efficiency compared to other tools in our consideration set.

For the re-squiggling step, we decided for the Tombo tool. The choice of Tombo was carefully conditioned by the fact that we were using DeepSignal2 for the detection phase. DeepSignal2, our tool of choice, requires data preprocessing performed by Tombo. This preprocessing is crucial to prepare the data for accurate and reliable analysis using DeepSignal2.

Based on Table 3.1, unequivocally highlighted DeepSignal2 as the most appropriate option among the deep learning-based methods that were considered. Despite mCaller demonstrating the highest accuracy, it was regrettably deemed incompatible with the specific flow cell that was used for sequencing. As a result, mCaller cant be utilized for this particular study.

4.3 Demultiplexing using ont_fast5_api

All the strains except EB362 and KP387 were sequenced in multiplexed form so they need to be demultiplexed first. To do so, it is necessary to obtain the information about barcodes, that were added to samples before sequencing. For this purpose, the Guppy(6.4.2, (`https://help.nanoporetech.com/en/articles/6628059-how-do-i-use-guppy-to-demultiplex-my-barcoded-reads`)) was used. Initially, all multiplexed samples were basecalled with a Guppy basecaller. After that, the Guppy barcoder was used to get the `barcode_summary.txt`, which is necessary for the subsequent step.

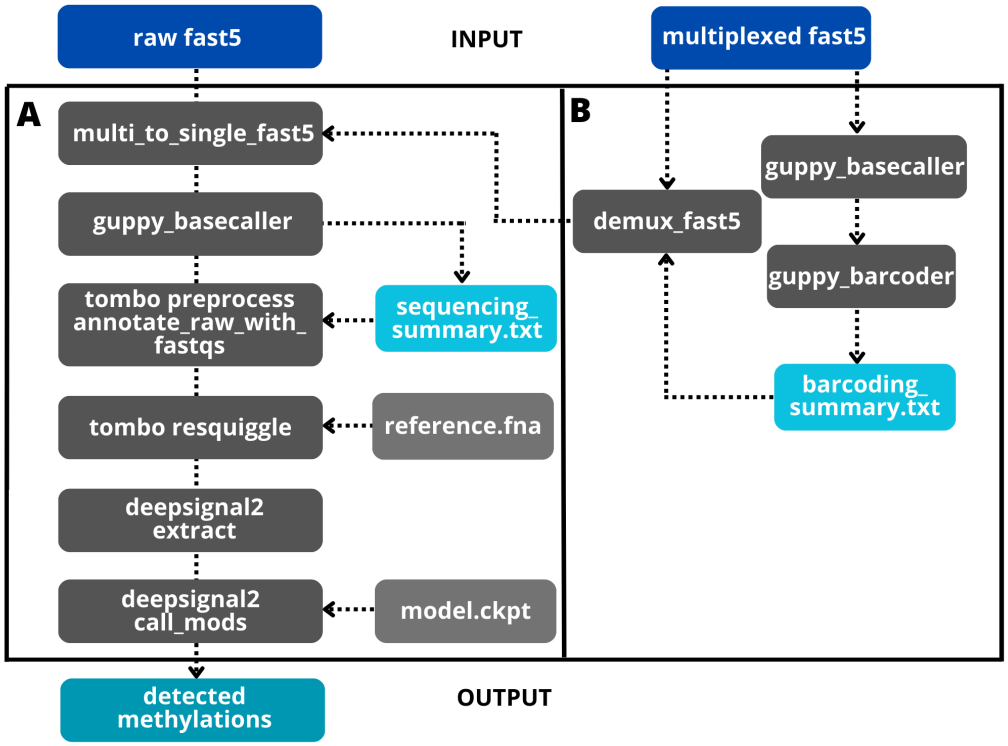

Fig. 4.1: Scheme of methylation detection workflow: (A): Workflow for the raw fast5. (B): Workflow for multiplexed data.

After acquiring the barcodes, the `demux_fast5` tool can be utilized. This script is integrated into the ont_fast5_api (4.1.1, `https://github.com/nanoporetech/ont_fast5_api`) interface, which facilitates the extraction of sorted reads. Considering that ONT offers 12 barcodes, this process results in receiving 12 corresponding files labelled as `barcode01` to `barcode12`. To determine the appropriate barcode for each genome, we referred to a predefined list that assigned specific barcodes to specific genomes. Each barcode file contained multiple fastq files, which were merged together to create a consolidated dataset ready for basecalling. This additional workflow for multiplexed samples is in Figure 4.1 (B).

4.4 Basecalling using Guppy

Prior to proceeding with any other task, the priority was to perform basecalling. Since the fast5 files were in multi-read format, it was essential to convert them into single-fast5 files first. To do so, the `multi_to_single_fast5` command was used. After that, the Guppy basecaller was used. It transformed the raw signal into a text-based format. In addition, this toolkit divided the basecalled

data according to quality. Except for folders containing data of various quality, also `sequencing_summary.txt` and logs were received. Basecalling is a time-intensive process that can take up to several hours. Therefore, utilizing a GPU was beneficial.

To ensure the highest level of precision, the data stored in the "pass" folder were selected. This file contains only reads of adequate quality. These reads were then concentrated into a single fastq file using `cat` command. Furthermore, the text file generated from this process was utilized in conjunction with the pycoQC (2.2.3, `https://a-slide.github.io/pycoQC/`) toolkit. Results are visible at table 4.2 shows the quantity of all reads, as well as reads that met the necessary quality standards. All other data in the table pertains solely to the high-quality reads prom "pass" folder.

Tab. 4.2: Strains information after base calling.

Strain	All reads	Pass reads	Output	Median read length	Median read quality
EB362	1 560 612	1 274 650	4 337 Mbp	1 094	11.56
KP387	734 097	629 295	9 615 Mbp	4 882	12.38
KP1179	346 899	346 878	3 177 Mbp	5 230	12.507
KP1193	555 212	555 180	3 370 Mbp	3 400	12.492
KP1231	161 325	161 308	1 154 Mbp	3 600	12.422
KP1236	497 324	497 308	2 394 Mbp	2 740	12.484
KP1228	545 789	545 771	2 256 Mbp	2 270	12.468
KP1272	486 849	486 832	2 055 Mbp	2 490	12.505
KP1209	60 477	60 474	232 Mbp	1 300	12.044
KP687	69 576	69 572	651 Mbp	1 520	12.042

4.5 Preprocessing and re-squiggling using Tombo

Preprocessing was performed with Tombo (1.5.1, `https://nanoporetech.github.io/tombo/resquiggle.html`), which required three inputs: raw single fast5 files, a fastq file containing all of the "pass" reads, and a `sequencing_summary.txt`. During preprocessing, the raw signals are annotated with all of the basecalled data. This process did not produce any output files. The only way to confirm that preprocessing had occurred is to check the number of reads that had been assigned sequences in a log file. Following preprocessing, the re-squiggling process was initiated using the annotated reads and a reference genome from NTUH-K2044. This

represents the final step before methylation detection. although once again no output files were generated, and we only received information regarding the percentage of unsuccessfully processed reads. It is crucial to set the same filenames at `-basecall-group`. Otherwise, the re-squiggling will result in:

```
FloatingPointError:  underflow encountered in exp.
```

After re-squiggling was done, there were no files as output, but the successful execution of the program could be verified based on the logs. Logs provide information about the percentage of unsuccessfully processed reads and the reasons why they weren't processed. If everything went right, there should be a text similar to this:

```
******************** WARNING ********************
Unexpected errors occured.  See full error stack traces for first
(up to) 50 errors in "unexpected_tombo_errors.7453.err"
[11:13:34] Final unsuccessful reads summary (13.8% reads unsuccess-
fully processed; 75522 total reads):
   8.7% (47524 reads):  Alignment not produced
   2.9% (15641 reads):  Poor raw to expected signal matching
   2.9% (15641 reads):  Poor raw to expected signal matching
   1.6% (8609 reads):  Read event to sequence alignment extends
   beyond bandwidth
   0.7% (3721 reads) :  Base calls not found in FAST5
   0.0% (23 reads):  Fewer changepoints found than requested
   0.0% (3 reads):  Unexpected error
   0.0% (1 reads):  Not enough raw signal around potential genomic
   deletion(s)
[11:13:34] Saving Tombo reads index to file.
```

4.6 Methylation detection using DeepSignal2

DeepSignal2 (0.1.3, `https://github.com/PengNi/deepsignal2`) was utilized to detect methylations. The initial step involved extracting the methylation information from fast5 files that had been modified in previous steps by `deepsignal2 extract`. This command results in a Tab-separated values (TSV) file which was used in a subsequent calling of modifications. This final step was performed using `deepsignal2 call_mods`. This command utilizes the previous TSV file and

trained model in order to call desired methylations. DeepSignal2 provides a trained model `model.dp2.CG.R9.4_1D.human_hx1_t2t.both_bilstm.b17_s16_epoch7.ckpt`, which can be used instead of training a new model. It is a model, that is available online without any restrictions. Using this model, the methylations could be obtained. Within the final `modification_call.tsv` file, several pieces of information can be found: chromosome name, name of the read and information if it is template or complement, probability of the predicted methylated or unmethylated state, label of methylation itself and others. These files were huge, some were up to hundreds of gigabytes. To be able to work with them, it was crucial to filter the data first.

Despite the modifications-calling command, DeepSignal2 comes with a script `call_modification_frequency.py` calculating the frequency of occurrence. This script counts the overall coverage for each position and the amount of methylated and unmethylated sites. Based on these data the frequency of occurrence for each position is calculated which can be utilized in the process of data optimization. These files are marked as `call_mods_frequency.tsv` in the following sections.

4.7 Optimization of the resulting data

As already mentioned the resulting data are huge and needed to be optimized before the analysis. The first optimization step used throughout the whole process was saving all the files in TSV format. The main optimization in order to reduce the amount of data was filtering all the `modification_call.tsv` files based on the probability of occurrence of methylation. For the most accurate results, the threshold was set to 90%. For this step, the `awk` command was used.

The commands utilized thus far have been executed through the utilization of PuTTY, the terminal interface. The establishment of a secure connection with the school server was achieved via the implementation of SSH (Secure Shell). The subsequent procedures were performed on a personal computer employing the Python programming language.

Based on `call_mods_frequency.tsv` files, data were filtered according to coverage and frequency of occurrence for each position. However, coverage varied considerably and therefore it was not possible to use a fixed threshold. In addition, the coverage did not have a symmetric distribution and therefore the use of the median proved to be the best choice. The median was calculated using the `MedianCounter.py` script and then used in line 9 of the `Preprocessing.py` to filter out positions from the `call_mods_frequency.tsv` files with too low coverage. However, these positions had varying frequencies of methylation occurrence. Positions for which more reads without methylation than with methylation were

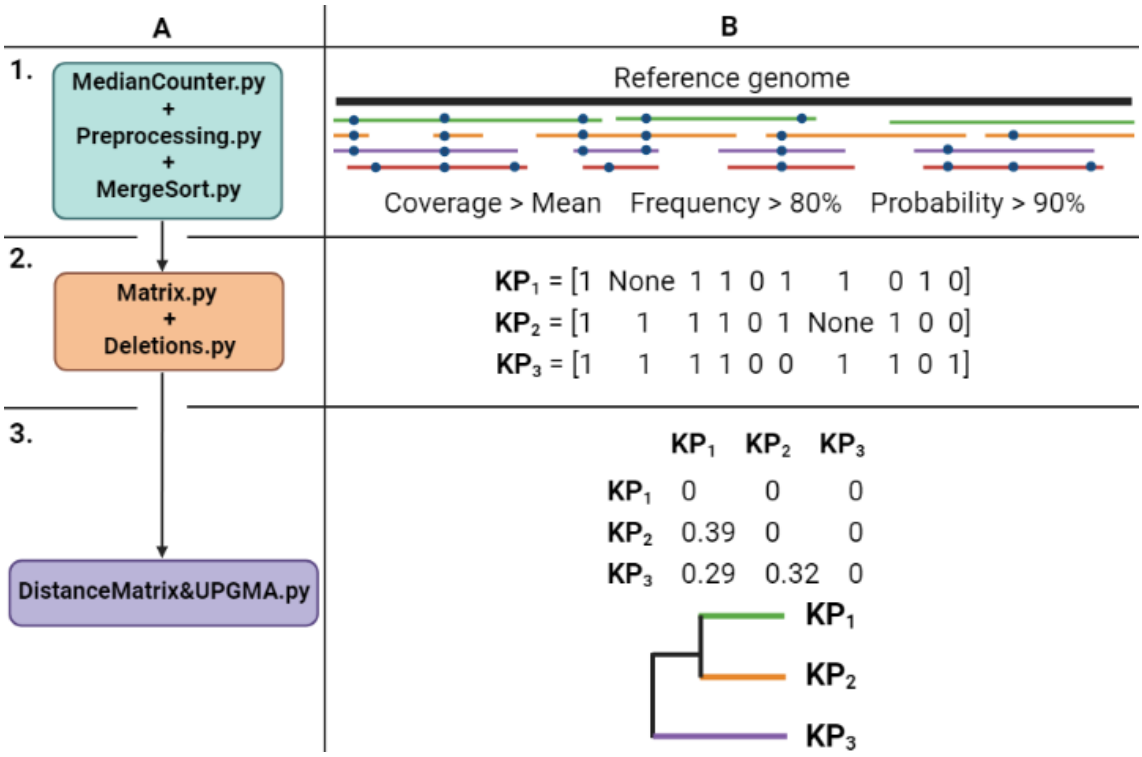

Fig. 4.2: Process of optimization and data analysis: (A): Scripts used for each step. (B): Schematic workflow of optimization and data analysis.

detected had a low frequency of occurrence and had to be removed. This threshold was set to 80%.

The last data preprocessing step was to link the positions in the `modification_call.tsv` file, filtered by the probability of methylation occurrence, and the positions from the `call_mods_frequency.tsv` file, filtered by coverage and frequency of occurrence. The output of this step was 10 files, one for each strain, containing positions meeting the above criteria. For ease of comparison, these positions were further sorted in ascending order using the MergeSort algorithm. A schematic drawing of the optimization procedure is shown in Figure 4.2 (B.1). The scripts used for this step can also be seen on the left side of Figure 4.2 (A.1).

Tab. 4.3: Maximal strains coverage and its median value.

Strain	Max. coverage	Median value
EB362	1 542	425
KP387	1 441	906
KP1179	540	253
KP1193	821	271
KP1231	192	85
KP1236	654	194
KP1228	503	191
KP1272	302	175
KP1209	49	15
KP687	52	10

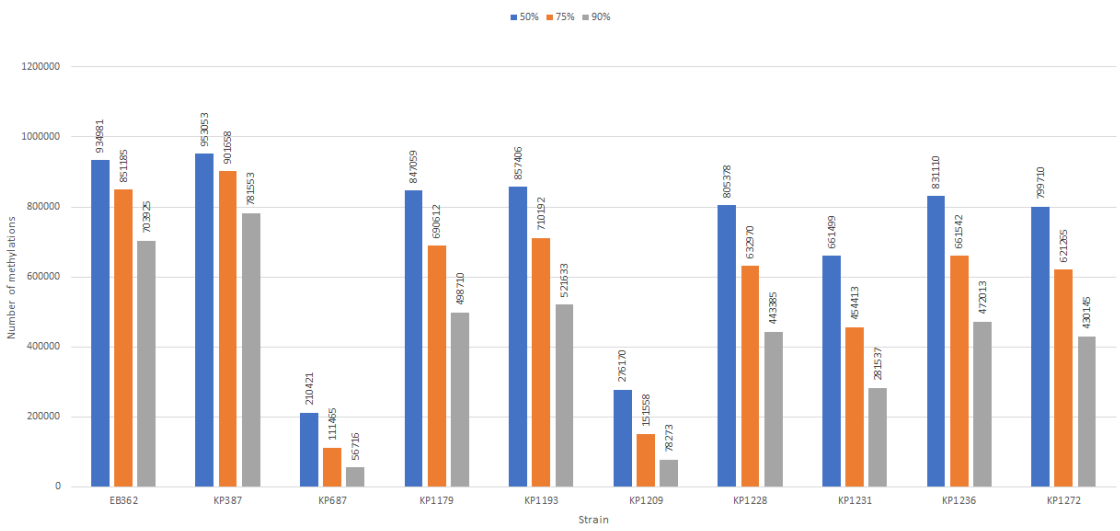

Fig. 5.1: Number of methylation based on probability of occurrence

5 Results

5.1 Analysis of the resulting data

Before any further analysis, the data were filtered based on different probabilities. The data obtained are shown in Figure 5.1. It is evident that there is a correlation between the probability of methylation occurrence and the number of methylations. As the probability that a position is methylated increases, the number of methylations decreases for all strains with approximately the same trend.

The data obtained from the procedure described in Section 4.7 were processed into a matrix for more detailed analysis. For this purpose, all positions from the 10 files were extracted and saved, creating a single vector of positions. This vector was then compared with the detected positions for each strain. If a position was present in a strain, a 1 was written to that position and vice versa. The result was an 11x531 matrix containing a vector of all positions and the presence or absence of methylation at that position for each strain (Fig.4.2(A.2, B.2)).

For more objective results, deletions were also detected. For their detection, `call_mods_frequency.tsv` files were used, which contain a list of all positions for each strain. These files were compared with the positions from the `AllPositions_sorted.tsv` file. If a position from `AllPositions_sorted.tsv` was not present in `call_mods_frequency.tsv`, the position was written as a deletion for the corresponding strain. These array positions were added to the matrix as the "None" value. Visualization of this matrix is in Figure 5.2 where only the zoomed area is shown for better readability. More detailed information on the number of

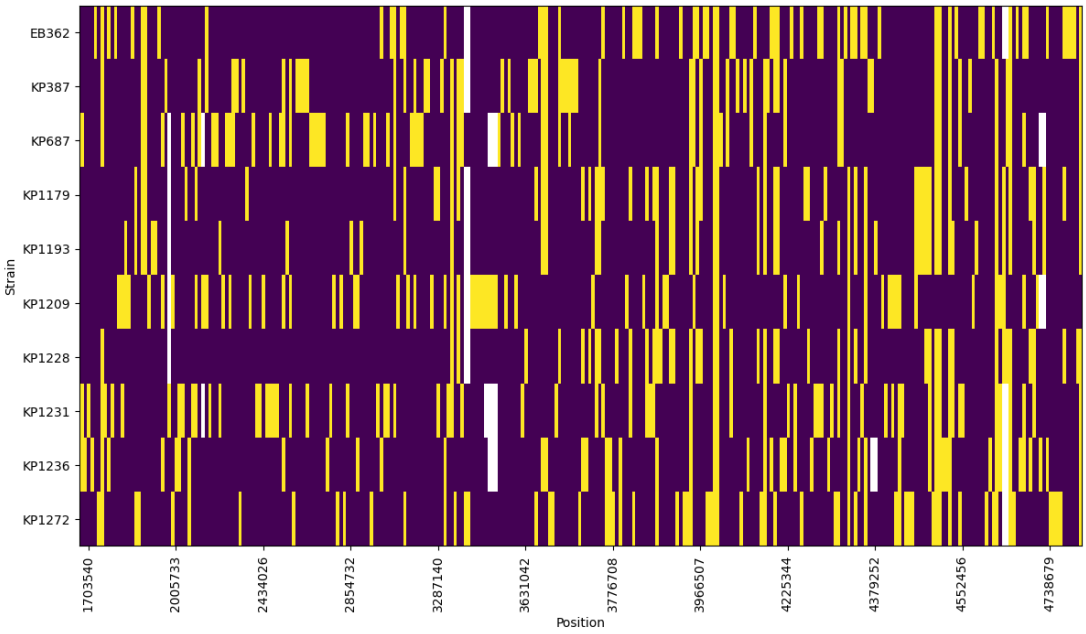

Fig. 5.2: Visualization of positions: White indicates deletions, yellow indicates methylations and purple indicates unmethylated positions.

methylations is given in Table 5.1. In addition, the table provides a comparison with the number of positions in the case that if a deletion were present in at least one strain, the position would be completely removed.

Tab. 5.1: Comparison of positions with and without deletions

	All positions	Unique	Common	Other
With deletions	531	328	2	201
Without deletions	510	311	2	197

To establish a meaningful distance matrix, a fitting metric was carefully selected. Considering the matrix's composition, which includes binary and sporadic "None" values, the Hamming distance appeared as the optimal choice. A `Hamming` function was created to calculate the distance between every pair of strains accurately. This function was used in populating a square matrix, initially filled with zeroes, with the computed distances based on the Hamming method. Figure 4.2 (B.3) shows an illustrative instance of this matrix, and the corresponding scripts are conveniently situated on the left (A.3).

The final analytical step involved plotting a dendrogram to visualize the relationships. The UPGMA (Unweighted Pair Group Method with Arithmetic Mean) technique was employed, leveraging the previously derived distance matrix as its

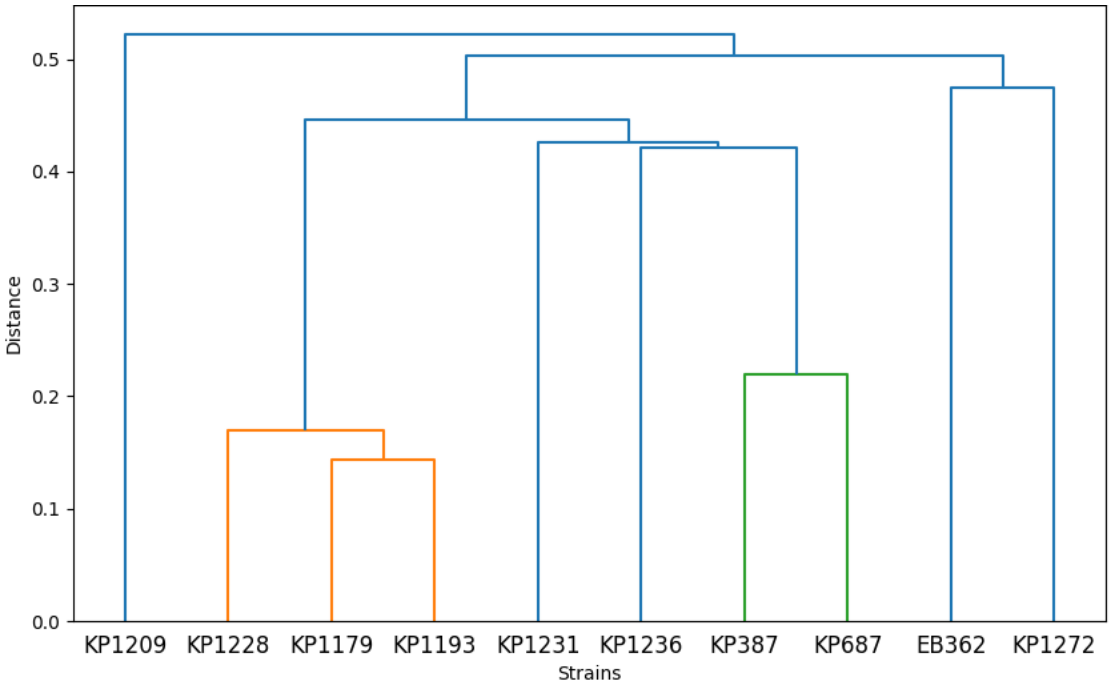

Fig. 5.3: Dendrogram of strains with deletions: Dendrogram constructed from detected methylations sites considering deletions.

foundational precursor. UPGMA, a clustering method, enabled the creation of a hierarchical representation that encapsulates the genetic associations among the strains.

To leverage the comprehensive functionality available within Python, the key libraries and functions were utilized, particularly `scipy.cluster.hierarchy` and `matplotlib`, to achieve the most robust results in analysis. Two dendrograms were generated for the purpose of comparison. Figure 5.3 showcases a dendrogram produced from a matrix that includes "None" values, a scenario that adds complexity to the analysis. In contrast, Figure 5.4 illustrates a dendrogram derived from the same data but without a focus on deletions. This approach enables us to assess the impact of "None" values on the resulting dendrogram, emphasizing the importance of data completeness in such analyses.

5.2 Discussion of results

Figure 5.1 suggests a connection between the number of methylations based on the established threshold of probability of occurrence. The trend of decline is similar across all strains. Moreover, a correlation between the sequencing method and the number of detected positions can be observed here. Strains that were sequenced

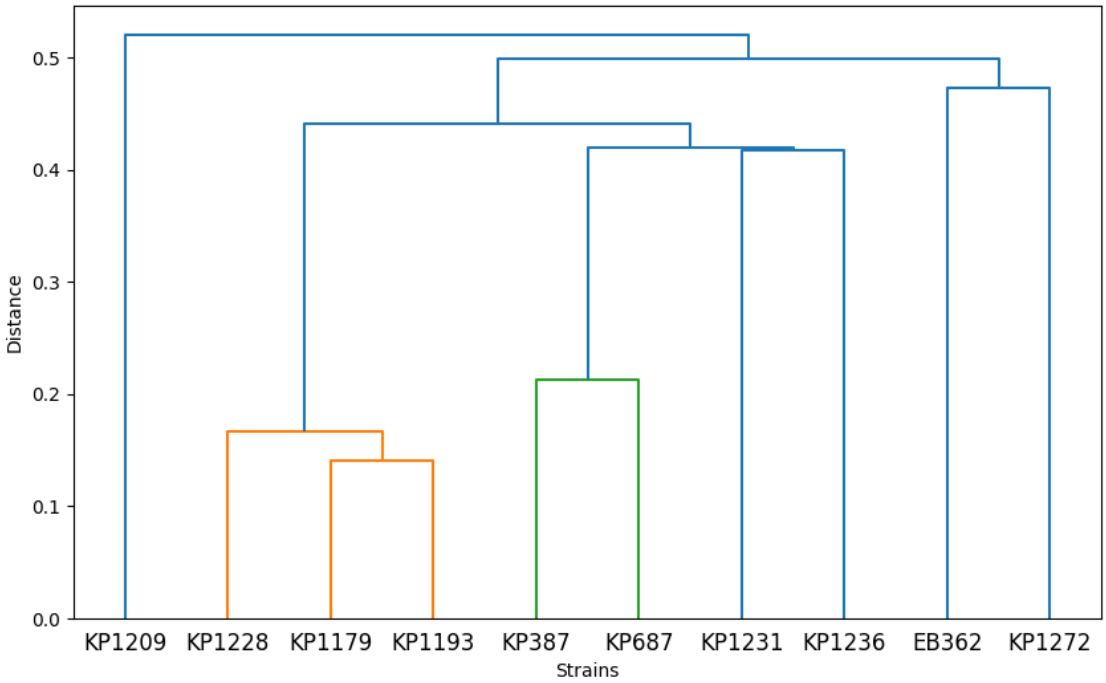

Fig. 5.4: Dendrogram of strains without deletions: Dendrogram constructed from detected methylations sites irrespective deletions.

alone have a higher number than strains that were multiplexed.

However, a deeper comparison is only provided by the dendrogram for the creation of which the data were filtered on the basis of various criteria. According to the dendrogram in Figure 5.3 and sequencing types, it is clear that KP1193, KP1179 and KP1228 are related. These three strains form a single cluster, confirming their belonging to the sequencing type ST551. This cluster stays the same even after considering the deletions (Fig. 5.4). In Figure 5.2 it can be seen that the deletions occur in these strains at mostly the same positions. The next cluster is formed by strains KP387 and KP687. Unlike the previous cluster, the strains in this one are not of the same sequencing type. While KP387 belongs to ST433, KP687 belongs to ST11. This would suggest that these species are functionally or otherwise similar, and thus could be compared. Just like with the previous cluster, strains within this cluster stay the same even without considering the deletions (Fig. 5.4). The difference is in the similarity of the clusters to each other. When considering the deletions, these clusters are more distant from each other; without considering them, they are, on the contrary, more similar. This shows that even a small change in the dataset can show differences that, in some cases, could be decisive.

Conclusion

It is evident that methylations have a crucial role in epigenetics, and further research in this area is expected to have significant benefits across multiple fields. Methylations provide additional regulatory control, and their in-depth investigation can shed light on their significance.

The theoretical background of the thesis was thoughtfully designed to ensure a comprehensive understanding. It successfully presented the necessary background information, making the practical part more accessible and understandable. Throughout this part, various tools were discussed, and their principles were explained, providing the readers with a clear understanding of their functionalities.

In the practical part of the thesis, the chosen tools, which were previously described, were applied to the data to detect methylations. By utilizing the knowledge gained from the theoretical section and the understanding of the tool's principles, the methodology for methylation detection was designed. As shown later, this methodology was able to implement the selected tools and successfully detect the methylations effectively. These data were then preprocessed for the most optimal results. The analytical part consisted of creating a distance matrix plotted into a dendrogram using UPGMA. This rendering made it relatively easy to evaluate the results based on the clusters created.

The initial finding of this thesis is a certain trend between the strains in relation to the number of methylations and the probability of their occurrence. Moreover, from this comparison, it can be observed that the number of detected positions is related to the sequencing method. For strains sequenced alone, a higher number of detected positions is seen than for multiplexed strains. As this is an experimental topic, the results were not known in advance. However, we hypothesized that strains with the same sequencing type should form a single cluster. This assumption was confirmed by assigning strains KP1179, KP1193 and KP1228 to a single cluster. This assumption was confirmed by assigning strains KP1179, KP1193 and KP1228 to a single cluster. On this basis, it can be assumed that methylation could be used for typing. The assignment of strains KP687 and KP387 to the same cluster despite different sequencing types suggests that typing based on methylation could yield even more accurate comparisons of similarities between strains.

I tried to process the essence of this topic so that it is understandable and provided a practical solution so that the reader can reproduce it himself and understand the meaning of the individual step. By presenting these findings, the thesis establishes a foundation for further research exploring the similarity of bacterial strains based on methylations.

Bibliography

[1] PASSARGE, Eberhard, [2018]. *Color atlas of genetics*. Fifth edition, revised and updated. Illustrated by Jürgen WIRTH. Stuttgart: Thieme. Flexibook. ISBN 978-3-13-241440-2.

[2] Bacterial DNA – the role of plasmids [online], 2014. New Zealand: Science Learning Hub [Accessed 2022-12-29]. Available at: https://www.sciencelearn.org.nz/resources/1900-bacterial-dna-the-role-of-plasmids

[3] BARON, Samuel, ed., 1991. Medical microbiology. 3rd ed. New York: Churchill Livingstone. ISBN 0-443-08671-0.

[4] O'DONNELL, M., L. LANGSTON a B. STILLMAN, 2013. Principles and Concepts of DNA Replication in Bacteria, Archaea, and Eukarya. Cold Spring Harbor Perspectives in Biology [online]. **5**(7), a010108-a010108 [Accessed 2022-12-07]. ISSN 1943-0264. Available at: doi:10.1101/cshperspect.a010108

[5] ABDELHALEEM, Mohamed, 2010. Helicases: An Overview. In: ABDELHALEEM, Mohamed M., ed. Helicases [online]. Totowa, NJ: Humana Press, 2009-9-15, s. 1-12 [Accessed 2022-12-07]. Methods in Molecular Biology. ISBN 978-1-60327-354-1. Available at: doi:10.1007/978-1-60327-355-8_1

[6] Introduction to Genomics: What's a Genome?, 2014. National Human Genome Research Institute [online]. Bethesda (Maryland): National Human Genome Research Institute, 11 October 2019 [Accessed 2022-12-29]. Available at: https://www.genome.gov/About-Genomics/Introduction-to-Genomics

[7] OCHMAN, H. a A. CARO-QUINTERO, 2016. Genome Size and Structure, Bacterial. In: Encyclopedia of Evolutionary Biology [online]. Elsevier, 2016, s. 179-185 [Accessed 2022-12-07]. ISBN 9780128004265. Available at: doi:10.1016/B978-0-12-800049-6.00235-3

[8] NI, Peng, Neng HUANG, Zhi ZHANG, et al., 2019. DeepSignal: detecting DNA methylation state from Nanopore sequencing reads using deep-learning. *Bioinformatics* [online]. **35**(22), 4586-4595 [Accessed 2022-11-27]. ISSN 1367-4803. Available at: doi:10.1093/bioinformatics/btz276

[9] SÁNCHEZ-ROMERO, María A, Ignacio COTA a Josep CASADESÚS, 2015. DNA methylation in bacteria: from the methyl group to the methylome. *Current Opinion in Microbiology* [online]. **25**, 9-16 [Accessed 2022-11-09]. ISSN 13695274. Available at: doi:10.1016/j.mib.2015.03.004

[10] SJAHPUTERA, Ozy, James M. KELLER, J. Wade DAVIS, et al., 2007. Relational Analysis of CpG Islands Methylation and Gene Expression in Human Lymphomas Using Possibilistic C-Means Clustering and Modified Cluster Fuzzy Density. IEEE/ACM Transactions on Computational Biology and Bioinformatics [online]. **4**(2), 176-189 [Accessed 2022-11-27]. ISSN 1545-5963. Available at: doi:10.1109/TCBB.2007.070205

[11] CASADESUS, Josep a David LOW, 2006. Epigenetic Gene Regulation in the Bacterial World. Microbiology and Molecular Biology Reviews [online]. **70**(3), 830-856 [Accessed 2023-01-02]. ISSN 1092-2172. Available at: doi:10.1128/MMBR.00016-06

[12] PORTNOY, D. A., David A. LOW, Nathan J. WEYAND a Michael J. MAHAN, 2001. Roles of DNA Adenine Methylation in Regulating Bacterial Gene Expression and Virulence. Infection and Immunity [online]. **69**(12), 7197-7204 [Accessed 2023-01-02]. ISSN 0019-9567. Available at: doi:10.1128/IAI.69.12.7197-7204.2001

[13] JANG, Hyun, Woo SHIN, Jeong LEE a Jeong DO, 2017. CpG and Non-CpG Methylation in Epigenetic Gene Regulation and Brain Function. Genes [online]. **8**(6) [Accessed 2022-11-27]. ISSN 2073-4425. Available at: doi:10.3390/genes8060148

[14] COLLIER, Justine, 2009. Epigenetic regulation of the bacterial cell cycle. Current Opinion in Microbiology [online]. **12**(6), 722-729 [Accessed 2023-01-02]. ISSN 13695274. Available at: doi:10.1016/j.mib.2009.08.005

[15] DNA Sequencing Fact Sheet, 2014. National Human Genome Research Institute [online]. Bethesda (Maryland): National Human Genome Research Institute [Accessed 2023-05-21]. Available at: https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Fact-Sheet

[16] Amplicons and Amplicon Sequencing, c2023. CD Genomics [online]. New York: CD Genomics, 3 December 2019 [Accessed 2023-05-20]. Available at: https://www.cd-genomics.com/blog/amplicons-and-amplicon-sequencing/

[17] SWAN, Kathryn A., Damian E. CURTIS, Kathleen B. MCKUSICK, Alexander V. VOINOV, Felipa A. MAPA a Michael R. CANCILLA, 2002. High-Throughput Gene Mapping in Caenorhabditis elegans. Genome Research [online]. **12**(7), 1100-1105 [Accessed 2023-04-28]. ISSN 1088-9051. Available at: doi:10.1101/gr.208902

[18] GREEN, Eric, 2014. Shotgun sequencing. National Human Genome Research Institute [online]. Bethesda (Maryland): National Human Genome Research Institute, 25 May 2023 [Accessed 2023-05-26]. Available at: https://www.genome.gov/genetics-glossary/Shotgun-Sequencing

[19] Amplicon Sequencing [online], c2023. Ebersberg: Eurofins Genomics [Accessed 2023-04-28]. Available at: https://eurofinsgenomics.eu/en/eurofins-genomics/material-and-methods/amplicon-sequencing/

[20] HEATHER, James M. a Benjamin CHAIN, 2016. The sequence of sequencers: The history of sequencing DNA. Genomics [online]. **107**(1), 1-8 [Accessed 2022-11-22]. ISSN 08887543. Available at: doi:10.1016/j.ygeno.2015.11.003

[21] HAMES, David a Nigel HOOPER, 2011. Biochemistry. 4th ed. New York: Garland Science. BIOS instant notes. ISBN 978-0-415-60845-9.

[22] DORADO, G., S. GÁLVEZ, H. BUDAK, T. UNVER a P. HERNÁNDEZ, 2019. Nucleic-Acid Sequencing. In: Encyclopedia of Biomedical Engineering [online]. Elsevier, 2019, s. 443-460 [Accessed 2022-12-28]. ISBN 9780128051443. Available at: doi:10.1016/B978-0-12-801238-3.08998-4

[23] PARK, Sason Y., c2023. Sanger Sequencing. Association for Diagnostics Laboratory Medicine/www.aacc.org/ [online]. Washington [cit. 2023-08-01]. Dostupné z: https://www.aacc.org/science-and-research/clinical-chemistry-trainee-council/trainee-council-in-english/pearls-of-laboratory-medicine/2014/sanger-sequencing

[24] Illumina Sequencing Technology: Highest data accuracy, simple workflow, and a broad range of applications. [online], 2010. San Diego: Illumina [Accessed 2022-12-29]. Available at: https://www.illumina.com/documents/products/techspotlights/techspotlight_sequencing.pdf

[25] BLEIDORN, Christoph, 2016. Third generation sequencing: technology and its potential impact on evolutionary biodiversity research. *Systematics and Biodiversity* [online]. **14**(1), 1-8 [Accessed 2022-11-22]. ISSN 1477-2000. Available at: doi:10.1080/14772000.2015.1099575

[26] WETTERSTRAND, Kris.A, 2014. Nanopore DNA Sequencing. National Human Genome Research Institute [online]. Bethesda (Maryland): National Human Genome Research Institute, 5 May 2022 [Accessed 2022-12-29]. Available at: https://www.genome.gov/genetics-glossary/Nanopore-DNA-Sequencing

[27] Types of nanopores [online], c2008-2022. Oxford Science Park: Oxford Nanopore Technologies [Accessed 2022-12-30]. Available at: https://nanoporetech.com/how-it-works/types-of-nanopores

[28] How nanopore sequencing works [online], c2008-2022. Oxford Science Park: Oxford Nanopore Technologies [Accessed 2022-12-30]. Available at: https://nanoporetech.com/how-it-works

[29] FOXMAN, Betsy, 2012. A Primer of Molecular Biology. In: Molecular Tools and Infectious Disease Epidemiology [online]. Elsevier, 2012, s. 53-78 [Accessed 2023-04-28]. ISBN 9780123741332. Available at: doi:10.1016/B978-0-12-374133-2.00005-8

[30] BAKER, Monya, 2012. De novo genome assembly: what every biologist should know. Nature Methods [online]. **9**(4), 333-337 [Accessed 2023-04-28]. ISSN 1548-7091. Available at: doi:10.1038/nmeth.1935

[31] An Overview of Genome Assembly, c2023. CD Genomics [online]. New York: CD Genomics [Accessed 2023-05-21]. Available at: https://www.cd-genomics.com/an-overview-of-genome-assembly.html

[32] What is de novo assembly?, c2016-2022. The Sequencing Center [online]. Fort Collins: The Sequencing Center [Accessed 2023-05-21]. Available at: https://thesequencingcenter.com/knowledge-base/de-novo-assembly/

[33] KYRIAKIDOU, Maria, Helen H. TAI, Noelle L. ANGLIN, David ELLIS a Martina V. STRÖMVIK, 2018. Current Strategies of Polyploid Plant Genome Sequence Assembly. Frontiers in Plant Science [online]. 9 [Accessed 2023-05-21]. ISSN 1664-462X. Available at: doi:10.3389/fpls.2018.01660

[34] KHAN, Abdul Rafay, Muhammad Tariq PERVEZ, Masroor Ellahi BABAR, Nasir NAVEED a Muhammad SHOAIB, 2018. A Comprehensive Study of De Novo Genome Assemblers: Current Challenges and Future Prospective. Evolutionary Bioinformatics [online]. **14** [Accessed 2023-05-21]. ISSN 1176-9343. Available at: doi:10.1177/1176934318758650

[35] KAHLKE, Tim, c2019. Genome Assembly with Minimap2 and Miniasm. Introduction to Long-Read Data Analysis [online]. GitHub [Accessed 2023-05-21]. Available at: https://timkahlke.github.io/LongRead_tutorials/ASS_M.html

[36] LI, Heng, 2016. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. Bioinformatics [online]. **32**(14), 2103-2110 [Accessed 2023-05-21]. ISSN 1367-4811. Available at: doi:10.1093/bioinformatics/btw152

[37] KAHLKE, Tim, 2019. Genome assembly using Flye. Introduction to Long-Read Data Analysis [online]. GitHub [Accessed 2023-05-21]. Available at: https://timkahlke.github.io/LongRead_tutorials/ASS_F.html

[38] KOLMOGOROV, Mikhail, Derek M. BICKHART, Bahar BEHSAZ, et al., 2020. MetaFlye: scalable long-read metagenome assembly using repeat graphs. Nature Methods [online]. **17**(11), 1103-1110 [Accessed 2023-05-27]. ISSN 1548-7091. Available at: doi:10.1038/s41592-020-00971-x

[39] TAMAZIAN, Gaik, Pavel DOBRYNIN, Ksenia KRASHENINNIKOVA, Aleksey KOMISSAROV, Klaus-Peter KOEPFLI a Stephen J. O'BRIEN, 2016. Chromosomer: a reference-based genome arrangement tool for producing draft chromosome sequences. GigaScience [online]. **5**(1) [Accessed 2023-05-21]. ISSN 2047-217X. Available at: doi:10.1186/s13742-016-0141-6

[40] KAHLKE, Tim, c2023. Basecalling using Guppy [online]. North America: GitHub [Accessed 2022-12-30]. Available at: https://timkahlke.github.io/LongRead_tutorials/BS_G.html

[41] LIU, Yang, Wojciech ROSIKIEWICZ, Ziwei PAN, et al., 2021. DNA methylation-calling tools for Oxford Nanopore sequencing: a survey and human epigenome-wide evaluation. Genome Biology [online]. **22**(1) [Accessed 2022-12-27]. ISSN 1474-760X. Available at: doi:10.1186/s13059-021-02510-z

[42] Processing more samples in less time, c2023. Illumina [online]. San Diego: Illumina [Accessed 2023-05-22]. Available at: https://www.illumina.com/techniques/sequencing/ngs-library-prep/multiplexing.html

[43] WICK, Ryan R., Louise M. JUDD, Kathryn E. HOLT a Mihaela PERTEA, 2018. Deepbinner: Demultiplexing barcoded Oxford Nanopore reads with deep convolutional neural networks. PLOS Computational Biology [online]. **14**(11) [Accessed 2023-05-22]. ISSN 1553-7358. Available at: doi:10.1371/journal.pcbi.1006583

[44] Demultiplexing, c2018-2019. Nanopype Documentation [online]. Nanopype Documentation [Accessed 2023-05-20]. Available at: https://nanopype.readthedocs.io/en/latest/rules/demux/

[45] Ont_fast5_api, c2023. GitHub [online]. Oxford Nanopore Technologies [Accessed 2023-05-26]. Available at: https://github.com/nanoporetech/ont_fast5_api

[46] WICK, Ryan R., Louise M. JUDD a Kathryn E. HOLT, 2019. Performance of neural network basecalling tools for Oxford Nanopore sequencing. Genome Biology [online]. **20**(1) [Accessed 2023-05-22]. ISSN 1474-760X. Available at: doi:10.1186/s13059-019-1727-y

[47] NAPIERALSKI, Adam a Robert NOWAK, 2022. Basecalling Using Joint Raw and Event Nanopore Data Sequence-to-Sequence Processing. *Sensors* [online]. **22**(6) [Accessed 2023-05-28]. ISSN 1424-8220. Available at: doi:10.3390/s22062275

[48] Re-squiggle Algorithm [online], c2017-2018. Oxford Nanopore Technologies [Accessed 2023-04-28]. Available at: https://nanoporetech.github.io/tombo/resquiggle.html

[49] FENG, Yilin, Gulsum GUDUKBAY AKBULUT, Xulong TANG, Jashwant Raj GUNASEKARAN, Amatur RAHMAN, Paul MEDVEDEV, Mahmut KANDEMIR a Thomas LENGAUER, 2022. GPU-accelerated and pipelined methylation calling. Bioinformatics Advances [online]. **2**(1) [Accessed 2023-01-02]. ISSN 2635-0041. Available at: doi:10.1093/bioadv/vbac088

[50] Megalodon Algorithm Details [online], c2019. Oxford Nanopore Technologies [Accessed 2023-05-26]. Available at: https://nanoporetech.github.io/megalodon/algorithm_details.html

[51] WU, Keh-Ming, Ling-Hui LI, Jing-Jou YAN, et al., 2009. Genome Sequencing and Comparative Analysis of Klebsiella pneumoniae NTUH-K2044, a Strain Causing Liver Abscess and Meningitis. Journal of Bacteriology [online]. 2009-07-15, 191(14), 4492-4501 [cit. 2023-08-07]. ISSN 0021-9193. Dostupné z: doi:10.1128/JB.00315-09

Symbols and abbreviations

4mC	N4-methylcytosine
5mC	5-methylcytosine
6mA	N6-methyladenine
CcrM	cell cycle-regulated methylase
CGI	CpG island
CSV	Comma-separated values
Dam	DNA adenine methylase
DNA	deoxyribonucleic acid
ddNTP	dideoxynucleotide triphosphate
GPU	Graphics Processing Unit
MTase	methyltransferase
ONT	Oxford Nanopore Technologies
PCR	polymerase chain reaction
ssDNA	single-stranded DNA
TSV	Tab-separated value

A Structure of the attached files

```
/
├── Python
│   ├── OutputFiles
│   │   ├── AllPositions_sorted.tsv
│   │   ├── Deletions.tsv
│   │   ├── Matrix.txt
│   │   └── Matrix_NoneValues.tsx
│   ├── call_modification_frequency.py
│   ├── Deletions.py
│   ├── DistanceMatrixUPGMA.py
│   ├── Matrix.py
│   ├── MedianCounter.py
│   ├── MergeSort.py
│   └── Preprocessing.py
└── ShellScript
    ├── GenomeAssembly.sh
    ├── Demultiplexing.sh
    └── MethylationDetection.sh
```