UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS No. 3746

Real-Time Analysis of a Metagenomic Sample Obtained by Nanopore Based Sequencing Technology

Ivan Vujević

Zagreb, June 2016.

I would like to thank my professor Mile Šikić for his patience and constant motivation during the last three years.
Also, thank you to Krešimir Križanović for the help on this thesis.
I'd also like to thank my family and everybody who provided me support during this years.

CONTENTS

1. Introduction **1**

2. Overview **2**

2.1. Definitions 2

2.2. Metagenomics 3

2.3. Sequencing 4

2.3.1. Nanopore sequencing 5

2.4. SAM format 6

3. Material and methods **8**

3.1. Reducing database 8

3.2. Maximum likelihood estimation 9

3.2.1. Definition 9

3.2.2. Simple example 10

3.3. Mixture model 11

3.4. Expectation maximization algorithm 12

3.4.1. Definition 12

3.5. Method 14

3.5.1. Mapping against a reduced database 14

3.5.2. Determining which species are present in a sample 14

4. Implementation **18**

4.1. Core 18

4.2. Web application 21

5. Results **23**

6. Conclusion **34**

LIST OF FIGURES

2.1. Analytical strategies to determine which taxa are presented in a metagenome 4
2.2. Milestones in nanopore DNA sequencing 5
2.3. Nanopore sequencing 6

3.1. Likelihood function for 6 tails and 4 heads 11

4.1. Preparing reduced database 19
4.2. Workflow of the algorithm 20
4.3. Web page organisations 21
4.4. Home page of the web application 22
4.5. Graph on the web page for results 22

LIST OF TABLES

5.1. Pbsim parameters 23
5.2. Strains used for testing 24
5.3. Salmonella enterica coverrage x1, results obtained after the first substep 25
5.4. Salmonella enterica coverrage x1, results obtained after the second substep 25
5.5. Salmonella enterica coverrage x5, results obtained after the first substep 25
5.6. Salmonella enterica coverrage x5, results obtained after the second substep 26
5.7. Salmonella enterica coverrage x10, results obtained after the first substep 26
5.8. Salmonella enterica coverrage x10, results obtained after the second substep 26
5.9. Staphylococcus aureus coverage x1, results obtained after the first substep 27
5.10. Staphylococcus aureus coverage x1, results obtained after the second step 27
5.11. Staphylococcus aureus coverage x1, results obtained after the first substep 27
5.12. Staphylococcus aureus coverage x1, results obtained after the second step 28
5.13. Staphylococcus aureus coverage x10, results obtained after the first substep 28
5.14. Staphylococcus aureus coverage x10, results obtained after the second step 28
5.15. Klebsiella pneumoniae coverage x1, results obtained after the first substep 29
5.16. Klebsiella pneumoniae coverage x1, results obtained after the second step 29

5.17. Klebsiella pneumoniae coverage x5, results obtained after the first substep 29
5.18. Klebsiella pneumoniae coverage x5, results obtained after the second step 30
5.19. Klebsiella pneumoniae coverage x10, results obtained after the first substep 30
5.20. Klebsiella pneumoniae coverage x10, results obtained after the second step 30
5.21. Mixture of Salmonella enterica, Staphylococcus aureus, Klebsiella pneumoniae with the coverage x15, x10, x5 respectively; results obtained after the first step 31
5.22. Mixture of Salmonella enterica, Staphylococcus aureus, Klebsiella pneumoniae with the coverage x15, x10, x5 respectively; results obtained after the second step 31
5.23. Comparison of our method and Pathoscope, memory consumption and running time 32
5.24. Results obtained by Pathoscope for Salmonella enterica x1 set 32
5.25. Results obtained by our method for Salmonella enterica x1 set 33

1. Introduction

"We have discovered the secret of life" were the first words said by Watson and Crick on their entry at Eagle pub in Cambridge. Several hours before that, or rather, in the morning of February 28th, 1955, they discovered something that would completely change research in the field of human. They found structure, today known as deoxyribonucleic acid (DNA).

DNA is a structure that carries genetic informations, transfer characteristics from parents to their children. DNA is a part of all the living organisms, both the simplest and the most complex. When we talk about DNA, it is important to stress that in our body, besides our DNA, there are DNAs from all the other organisms that can be found in our body. Therefore, if our body is infected with some kind of disease, a DNA of the organism that causes that disease is present within the infected organism. Organisms that cause diseases in our body are called pathogenic organisms. Besides the pathogens there are also organisms that live in our body which do not cause any disease. In fact, they generally represent normal and ecologically important inhabitants of the human body.

The main goal of this thesis is to find an algorithm that could quickly and precisely detect strange (pathogenic) organisms from a sample.

The thesis is organized as follow: Chapter 2 gives overview of this problem which includes overview of the metagenomics and nanopore sequencing. Chapter 3 describes methods and algorithms important for understanding the other part of this thesis. Chapter 4 gives short overview of the implementation of this problem which includes implementation of a core, and the implementation of a web application which is used for the presentation of the results. Chapter 5 consists of the results of testing performed on a set of several genomes. In the end, the Chapter 6 gives a brief conclusion and insight into the future of this field of research.

2. Overview

2.1. Definitions

First, we will see some important definitions that are necessary for easier understanding of the main problem of this thesis.

A **gene** is a union of genomic sequences encoding a coherent set of potentially overlapping functional products [1].

A **genome** is an organism's complete set of DNA, a chemical compound that contains the genetic instructions needed to develop and direct the activities of every organism.

Sequencing means determining the exact order of the base pairs in a segment of DNA.

A **GI number** is series of digits that are assigned consecutively to each sequence record processed by NCBI.

A **TI number** is series of digits that are assigned consecutively to each clade of taxonomy tree.

Taxonomy is the science of naming, describing and classifying organisms and includes all plants, animals and microorganisms of the world. Taxonomy produces a hierarchy of groups of organisms; the organisms are assigned to groups based on similarities or dissimilarities of their characteristic. The classification system begins with three domains that encompass all living and extinct forms of life: archaea, bacteria and eukaryote. The main taxonomic ranks are: domain, kingdom, phylum, class, order, family, genus, and species.

RefSeq - The reference sequence database provides a comprehensive, integrated, non-redundant, well-annotated set of sequences, including genomic DNA, transcripts and proteins.

Assembly is a process through which short DNA sequence fragments are merged into a longer DNA sequence.

2.2. Metagenomics

Metagenomics is defined as the direct genetic analysis of genomes contained within an environmental sample. This field of study initially began with the cloning of environmental DNA, followed by functional expressions screening, and was than quickly complemented by direct random shotgun sequencing of environmental DNA.

Metagenomics proves access to the functional gene composition of microbial communities and thus gives a much broader description than phylogenetic surveys, which are often based only on the diversity of one gene. Metagenomics is also a powerful tool for generation novel hypotheses of microbial function.

The rapid and substantial cost reduction in next-generation sequencing has dramatically accelerated the development of sequence-based metagenomics. In fact, the number of metagenome shotgun sequence datasets has exploded in the past few years. [2]

A metagenome can be subject to three general analytical strategies that ultimately produce a profile of the taxa, phylogenetic lineages, or genomes present in the community as shown in figure 2.1.

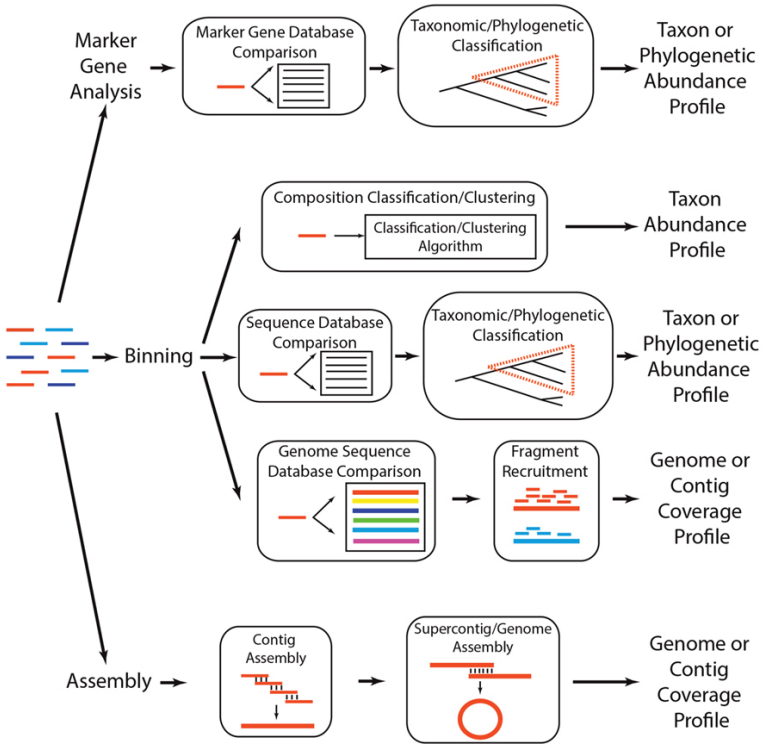

Figure 2.1: Analytical strategies to determine which taxa are presented in a metagenome [3]

Marker gene analysis is one of the most straightforward and computationally efficient ways of quantifying a metagenome's taxonomic diversity. This procedure involves comparing metagenomic reads to a database of taxonomically informative gene families (i.e., marker genes), identifying those reads that are marker gene homologs, and using sequence similarity to the marker gene database sequences to taxonomically annotate each metagenomic homolog. Since this approach involves comparing metagenomic reads to a relatively small database for the purpose of similarity search, marker gene analysis can be a relatively rapid way to estimate the diversity of metagenome [3]. This strategy is used in this work and will be discussed in more detail in the following chapters.

2.3. Sequencing

The beginning of the sequencing can be found in the research of Frederick Sanger who introduced fast sequencing methods [4]. The idea of sequencing the entire human genome was first proposed in discussions at scientific meetings organized by the US Department of Energy and others from 1984 to 1986 [5][6]. In 1990 the Department of Energy and National Institute of Health launched new project called Human genome

project with the goal of determining the human genome. The cost of the project was about 3 billion dollars and was expected to take 15 years. It is considered a key moment in the development of sequencing. The project was finished in April of 2013.

The results show that the human genome is constructed of 3 billion pairs of nucleotide bases and the average gene is consisted of 3000 bases. The total number of genes is about 20500; 99,99% bases are the same for all people.

Successful completion of the Human genome project created opportunity for ambitious project in the field of genetic engineering, including the searching for connections between the DNA sequence of some species and health, and an improved tracking of reactions on medical treatments. As a reaction on that there is a project called The Cancer Genome Atlas with a goal to determine the mutations of DNA causing the cancer in organs and tissues.

2.3.1. Nanopore sequencing

Nanopore sequencing has its origins in several laboratories during the 1980s [7]. In 1989, David Deamer jotted a seemingly implausible idea in his notebook, suggesting that it might be possible to sequence a single strand of DNA being drawn through a membrane's nanoscopic pore by electrophoresis. The milestones in nanopore DNA sequencing are shown on the figure 2.2.

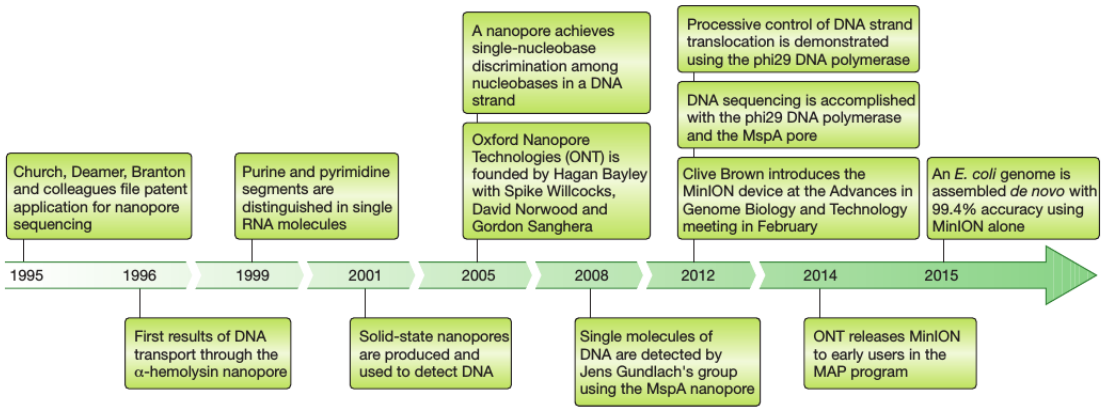

Figure 2.2: Milestones in nanopore DNA sequencing [7].

Certain porous transmembrane cellular proteins act as nanopores. Nanopores can also be made of silicon. Nanopore sequencing is based on theory when a nanopore is immersed in a conducting fluid and a potential is applied across it, an electric current is appeared. Nanopore sequencing technology identifies a nucleic acid sequence by threading a molecule through a pore with a diameter of a few nanometers [8]. That

pore might be a protein such as a-hemolysin that is embedded in a polymer membrane or a hole formed in a solid material such as silicon nitride. Voltage is applied across the membrane, creating an ionic current and an electrophoretic force that pulls the DNA through the opening. As the molecule zips through, it causes telltale fluctuations in the current that are specific to different DNA sequences. The technology can also be used to analyze RNA and proteins.

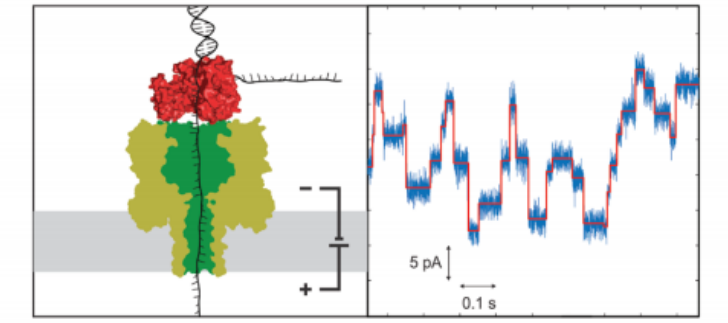

Figure 2.3: Nanopore sequencing [9]

2.4. SAM format

The Sequence Alignment/Map (SAM) format is a generic alignment format for storing reads alignments against reference sequences, supporting short and long reads produced by different sequencing platforms [10]. It is a TAB-delimited text format consisting of header section, which is optional, and an alignment section. Header lines start with '@' and if present, it must be prior to the alignments. Each alignment line has 11 mandatory fields for flexible or aligner specific information. These are:

1. QNAME: The query name.
2. FLAG: Combination of bitwise FLAGs.
3. RNAME: Reference sequence name of the alignment. If @SQ header is present, RNAME must be present in one of the header lines. An unmapped read has '*' at this field.
4. POS: 1-based leftmost mapping position of the first matching base.
5. MAPQ: Mapping quality. It equals $-10\,log_{10}\,Pr(\text{mapping position is wrong})$, rounded to the nearest integer. A value 255 indicates that the mapping quality is not available.

6. CIGAR: The cigar string of an alignment.
7. RNEXT: Reference name of the next read. This field is set to '*' when the information is unavailable, and set as '=' if reference name of next read is identical to the name of the current read.
8. PNEXT: Position of the next read.
9. TLEN: The observed length of the template.
10. SEQ: segment sequence.
11. QUAL: ASCII of base quality plus 33.

3. Material and methods

Overview of methods and data used in this work is presented in this chapter, and serves as a prerequisite for the understanding of the algorithms and problem-solving technique presented in Chapter 4.

3.1. Reducing database

Reduced database is used because it allows faster mapping and decrease the possibility of false positive hits. There are several ways how is possible to reduce database; one way can be to use specific k-mers as [11], and the other way can be to use specific markers.

At the core of [11] is a database that contains records consisting of a k-mer and the LCA of all organisms whose genomes contain that k-mer. Sequences are classified by querying the database for each k-mer in a sequence, and then using the resulting set of LCA taxa to determine an appropriate label for the sequence. Sequences that have no k-mers in the database are left unclassified.

There are two general methods by which marker genes are used to taxonomically annotate metagenomes. The first relies on sequence similarity between the read and the marker genes (e.g. Metaphlan [12]). The second approach uses phylogenetic information, which may take longer to calculate, but may also provide greater accuracy [3].

Metaphlan estimates the relative abundance of microbial cells by mapping reads against a reduced set of clade-specific marker sequences that are computationally preselected from coding sequences that unequivocally specific microbial clades at the species or higher taxonomic levels and cover all main functional categories.

In this work specific markers are used instead of specific k-mers. The reason why I used specific marker instead of specific k-mers is that the reads obtained by nanopore technologies have high percentage of errors, leading to a low number of true positive hits.

In order to reduce the database I used the same set of the markers as Metaphlan [12]. On the Metaphlan web page[1] a file named *markers_info.txt* can be found, which contains informations about clades specific markers. Each line in the file represents one marker. Lines start with GI annotation of a gene followed by its position in a genome. If a marker is on a reverse complement chain, its position starts with 'c'. It is possible that sometimes there are more positions which refer to the coding sequences without the introns. There are two fields required for reducing database in the second column of line; these are called 'ext' and 'clade'. A 'clade' field contains a name of the clade to which that marker belongs. Name starts with a letter indicates to which taxonomy group marker belong. This letter and the name of the clade are separated by "__" (e.g. s__Streptomyces_sp_KhCrAH_244). A letter can be one of the following:

a - all taxonomic levels;

k - kingdoms (Bacteria and Archea) only;

p - phyla only;

c - class only;

o - orders only;

f - family only;

g - genera only;

s - species only.

Field 'ext' can be empty, in case it is not, it contains written annotation of strains that also contain that marker but which are not under the given clade. Annotation of strains are given according to their assembly accession number (e.g GCF_000024865).

3.2. Maximum likelihood estimation

3.2.1. Definition

Let us assume that we have set of independent, identically distributed samples drawn from some known density function $p(\mathbf{x}|\boldsymbol{\theta})$. In this thesis vectors is written as a bold

[1] `https://bitbucket.org/biobakery/metaphlan2`

character. This density is a mixture of probability distributions, governed by a set of parameters $\boldsymbol{\theta}$:

$$\mathbf{x}^{(i)} \sim p(\mathbf{x}|\boldsymbol{\theta}).$$

Formally, there is a set $\mathcal{D} = \{x^{(1)}, x^{(2)}, ..., x^{(N)}\}$ of observed data by assumption generated from p. According to that, we define the probability of observing the data under the parameters $\boldsymbol{\theta}$ as:

$$p(\mathcal{D}|\boldsymbol{\theta}) = p(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) \equiv \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$$

With this, a probability density function of $\mathcal{D}$ governed by parameters $\boldsymbol{\theta}$ is defined. Likewise, this function can be observed as a function of $\boldsymbol{\theta}$ governed by fixed parameter $\mathcal{D}$. In that case, this function is called likelihood function and is written as $\mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$.

The goal is to estimate the parameter $\hat{\boldsymbol{\theta}}_{ML}$ which maximizes the likelihood function $\mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$. Formally, we need to estimate $\hat{\boldsymbol{\theta}}_{ML}$, such that:

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$$

Instead of maximizing the likelihood function, it is more practical to maximize the logarithm of the likelihood function. This function is called the log-likelihood function and can be written as:

$$\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = \ln \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) = \sum_{i=1}^{N} \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta})$$

3.2.2. Simple example

Let us suppose that we toss a coin 10 times and observe 6 tails (T) and 4 heads (H). Let us say that μ is a probability of the appearance of the head after one toss. Consequently, we can write $P(X = H|\mu) = \mu$ and $P(X = T|\mu) = 1 - \mu$. The likelihood function for this case is

$$\mathcal{L}(\mu|\mathcal{D}) = P(\mathcal{D}|\mu) = \mu^4(1 - \mu)^6$$

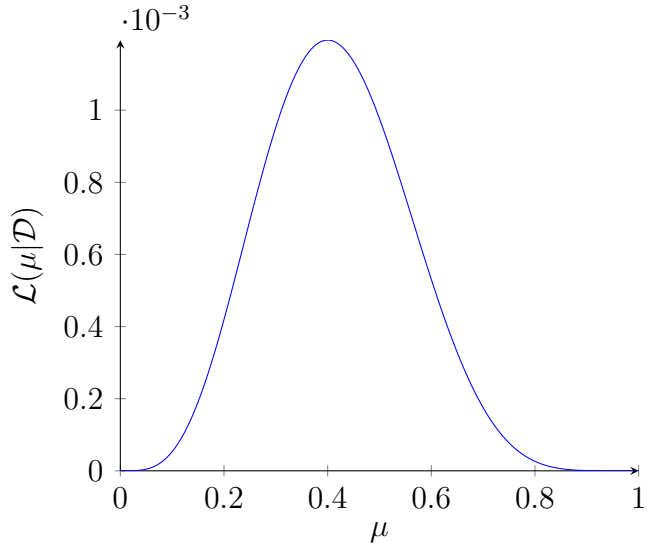

Figure 3.1: Likelihood function for 6 tails and 4 heads

From the figure 3.1 it can be seen that the likelihood function has maximum value for $\mu = 0.4$ and it is the estimated parameter by the maximum likelihood estimation method.

3.3. Mixture model

A mixture model is one in which a set of component models is combined to produce a richer model [13]:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|\boldsymbol{\theta}_k)$$

Let us assume that we have N reads and G genomes. Every read can origin from one or more genomes. Our task is to determine probability that i-th read origins from j-th genome. We define the mixture components as $p(\mathbf{x}|\boldsymbol{\theta}_k)$ with the parameters $\boldsymbol{\theta}_k$. Parameters π_k are mixture coefficients and for it worth that $\sum_{k=1}^{G} \pi_k = 1$. Now the mixture density can be expressed as:

$$p(\mathbf{x}) = \sum_{k=1}^{G} P(\mathcal{G}_k) p(\mathbf{x}|\mathcal{G}_k),$$

where $\pi_k = P(\mathcal{G}_k)$ is the prior distribution of chosen genome k and $p(\mathbf{x}|\boldsymbol{\theta}_k) = p(\mathbf{x}|\mathcal{G}_k)$ is a density of $\mathbf{x}$ with choosing genome k. A Bayesian rule is used in order to calculate the posterior distribution $P(\mathcal{G}_k|\mathbf{x})$:

$$P(\mathcal{G}_k|\mathbf{x}) = \frac{P(\mathcal{G}_k)p(\mathbf{x}|\mathcal{G}_k)}{\sum_{j=1}^{G} P(\mathcal{G}_j)p(\mathbf{x}|\mathcal{G}_j)} = \frac{\pi_k p(\mathbf{x}|\boldsymbol{\theta}_k)}{\sum_{j=1}^{G} \pi_j p(\mathbf{x}|\boldsymbol{\theta}_j)} \equiv h_k$$

and it is a possibility that **x** originates from the k-th genome.

Our task is to determine the parameters of this model:

$$\boldsymbol{\theta} = \{P(\mathcal{G}_k), \boldsymbol{\theta}_k\}_{k=1}^{G}.$$

We can estimate these parameters with the maximum likelihood estimation. The log-likelihood function for this case is:

$$\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = \ln \prod_{i=1}^{N} p(\mathbf{x}^{(i)}) = \ln \prod_{i=1}^{N} \sum_{k=1}^{G} \pi_k p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k) = \sum_{i=1}^{N} \ln \sum_{k=1}^{G} \pi_k p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k).$$

Obviously the maximization of this is not possible to solve in closed form. In order to solve this other methods are required. One of them, expectation maximization algorithm, is described in the next section.

3.4. Expectation maximization algorithm

Expectation maximization algorithm (EM algorithm) is an iterative method for solving the log-likelihood problem with the latent variables. A latent variable is variable whose realization could not be observed directly, as illustrated in the following example.

Let us assume that we have two coins, A and B. The possibility that we toss a head with the coin A is μ_A, and for B is μ_B. So, we have $P(A) = \mu_A$ and $P(B) = \mu_B$. These parameters are unknown so we need to estimate them.

If we know which coin is thrown we can simply estimate it using the maximum likelihood method described in 3.2. The problem of estimating μ_A and μ_B becomes more complex if we do not know which coin is thrown in which round. We only know the results of each toss. To solve this problem we do not use the maximum likelihood method. In order to solve this we need to use the EM algorithm.

3.4.1. Definition

The goal of this algorithm is to find parameters $\boldsymbol{\theta}$ which maximize the log-likelihood function $\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X})$. Model $p(\mathbf{X}|\boldsymbol{\theta})$ is expanded with a set of latent variables $\mathbf{Z}$ and

a joint density $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ is used. Marginal distribution $p(\mathbf{X}|\boldsymbol{\theta})$ can always be reconstructed as:

$$p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{Z} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

The set **{X,Z}** is called complete, whereas $\mathbf{X}$ is incomplete. According to that, $\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Z})$ is complete log-likelihood, and $\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X})$ is incomplete log-likelihood. Incomplete log-likelihood can be defined as:

$$\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X}) = \ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

whereas complete log-likelihood is:

$$\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Z}) = \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}).$$

The main difference between the complete and incomplete log-likelihood is that the incomplete could be solved in closed form, whereas the complete cannot. We do not know set $\mathbf{Z}$, it is the missing data. Therefore, it is not possible to work with the complete log-likelihood directly. Instead of directly observing the complete log-likelihood, the expectation of complete log-likelihood is tackled, $\mathbb{E}[\ln \mathcal{L}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Z})]$, what principally is the goal.

Maximization of expectation is achieved by alternation between two steps of EM algorithm: E-step and M-step. At an E-step (expectation step) we calculate the expectation of the complete log-likelihood with fixed parameters $\boldsymbol{\theta}^{(t)}$. It can be written as:

$$\begin{aligned} \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \mathbb{E}_{\mathcal{Z}|\mathcal{D},\boldsymbol{\theta}^{(t)}}[\ln\ \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}, \mathcal{Z})] \\ &= \mathbb{E}_{\mathcal{Z}|\mathcal{D},\boldsymbol{\theta}^{(t)}}[\ln\ \mathcal{L}(\mathcal{D}, \mathcal{Z}|\boldsymbol{\theta})] \\ &= \sum_{Z} P(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \end{aligned} \tag{3.1}$$

In a M-step (maximization step) new parameters $\boldsymbol{\theta}^{(t+1)}$ which maximize 3.1 need to be chosen:

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$$

This is make problem easier compared to the problem from the beginning of this section. In most cases it can be solved in the closed form. Below is a pseudocode of general EM algorithm.

Algorithm 1: General EM algorithm

Data: initialize parameters $\boldsymbol{\theta}^0$

$t \leftarrow 0$;

while *not convergency of ln* $\mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$ *or parameters* **do**

  E-step: calculate $P(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^t)$

  M-step: $\boldsymbol{\theta}^{(t+1)} \leftarrow \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$

    where $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \sum_Z P(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$

  $t \leftarrow t + 1$

3.5. Method

Previously we gave simple introduction to maximum likelihood estimation and EM algorithm what is necessary for better understanding this method, especially the second step where classifier is used to determine which species is present in a sample. There are two steps in our method. First we need to map metagenomic sample against reduced database and after that determine which species are present in a sample.

3.5.1. Mapping against a reduced database

In the first step we map metagenomic reads against a reduced database of clade-specific marker sequences. Clades are groups of genomes (organisms) that can be as specific as species or as broad as phyla. Clade-specific markers are coding sequences (CDS) that satisfy the conditions of being strongly conserved within the clade's genomes and not possessing substantial local similarity with any sequence outside the clade. This can be done very efficiently, as the reduced database contains only ~4% of sequenced microbial genes, and each read of interest has at most one match due to the markers' uniqueness [12]. Despite of that we can not uniquely map reads to the specific species because some marker could belong to the several species.

3.5.2. Determining which species are present in a sample

At this step we have output file with mapped reads. Each of R reads could be either mapped to one of the markers or be unmapped. The task is to determine which read originates from which species and to accordingly conclude which species are present in a sample. To solve this problem we use previously described mixture model which definition is given in the next paragraph.We define our model at a level of genomes

because in the second substep of this step we try to determine which genome is present in a sample, but there is not difference on which level of taxonomy tree we are.

We assume that reads are drawn from a small subset of unknown size from the pathogen genomes in the database. It assumes that each read is drawn from only one of the genomes in the subset. Parameters in the model represent the proportions of reads that originate from each genome as well as the proportion of the non-unique reads that are incorrectly assigned to each genome due to sequence similarity [14].

Let us say that we have vector $\mathbf{z} = (z_1, ..., z_G)$ where a $z_k = 1$ if a read originates from k-th genome, otherwise $z_k = 0$. Note that by assumption, one and only one element of the vector $\mathbf{z}$ can be equal to 1. We assume that that $\mathbf{z}$ follows a multinomial distribution, with probability of success:

$$P(z_k = 1) = \pi_k.$$

Also, we know that $\sum_k z_k = 1$ and according to that we can write:

$$P(\mathbf{z}) = \prod_{k=1}^{G} \pi_k^{z_k}.$$

For the unique reads, we know the template genome of interest or, in other words, we directly observe the genome indicator $\mathbf{z}$. In the case of the non-unique reads, the genome indicator $\mathbf{z}$ is the missing data. For the non-unique reads, the observations are partial mapping qualities for each of the genomes. These mapping probabilities are provided as posterior probabilities, which are scaled mapping qualities or relative likelihood alignment scores obtained from the algorithm. More specifically, for the i-th read we denote these mapping scores by $\mathbf{q}^{(i)} = (q_1^{(i)}, ...q_G^{(i)})$. For the non-unique reads, these represent the uncertainty in mapping and need to be rescaled, or equivalently these reads need to be reassigned to the correct template genome of origin. In order to do this, we define a second set of parameters, $\boldsymbol{\delta} = (\delta_1, ..., \delta_G)$ where δ_j is reassignment parameter that represents the proportions of the non-unique reads that need to be reassigned to the j-th genome. We can write our likelihood function $p(x|\mathbf{z}, \boldsymbol{\theta})$:

$$p(x^{(i)}|\mathbf{z}, \boldsymbol{\theta}) = \prod_{k=1}^{G} p(x^{(i)}|\boldsymbol{\theta}_k)^{z_k} = \prod_{k=1}^{G} (\delta_k^{1-y^{(i)}} q_k^{(i)})^{z_k}$$

where we defined our parameters $\boldsymbol{\theta}$ as $\delta_k^{1-y^{(i)}} q_k^{(i)}$ where $y^{(i)}$ is the indicator variable for unique reads. If $y^{(i)} = 1$ it means that the read i is unique and we do not need to use reassignment parameter δ for that read.

Joint distribution can be written using two previous formula:

$$p(x^{(i)}, \mathbf{z}|\boldsymbol{\theta}) = P(\mathbf{z})p(x^{(i)}|\mathbf{z}, \boldsymbol{\theta}) = \prod_{k=1}^{G} \pi_k^{z_k} \prod_{k=1}^{G} p(x^{(i)}|\boldsymbol{\theta}_k)^{z_k} = \prod_{k=1}^{G} \pi_k^{z_k} (\delta_k^{1-y^{(i)}} q_k^{(i)})^{z_k}$$

Now, knowing the joint distribution $p(x, \mathbf{z}|\boldsymbol{\theta})$ and $P(\mathbf{z})$ we can write complete log-likelihood function as:

$$\begin{aligned}
\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}, \mathcal{Z})) &= \ln \prod_{i=1}^{R} p(x^{(i)}, \mathbf{z}^{(i)}|\boldsymbol{\theta}_k) \\
&= \ln \prod_{i=1}^{R} \prod_{k=1}^{G} \pi_k^{z_k^{(i)}} p(x|\boldsymbol{\theta}_k)^{z_k^{(i)}} \\
&= \sum_{i=1}^{R} \sum_{k=1}^{G} z_k^{(i)} (\ln \pi_k + \ln p(x^{(i)}|\boldsymbol{\theta}_k))
\end{aligned}$$

We assume a priori that both $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ follow a Dirichlet distributions, the densities of which can been written as:

$$p(\boldsymbol{\pi}|\mathbf{a}) \sim \prod_{j=1}^{G} \pi_j^{a_j - 1}$$

$$p(\boldsymbol{\theta}|\mathbf{b}) \sim \prod_{j=1}^{G} \boldsymbol{\theta}_j^{b_j - 1}$$

If $a_j = 1$ for all genomes, this is equivalent to adding one unique read for each of the G genomes, and $a_j = n$ would be equivalent of adding n unique reads to the j-th genome. Similarly, $b_j = n$ is equivalent of adding n non-unique reads to the j-th genome.

In the previous section is given overview of an EM algorithm so now I will give only pseudocode of our method which includes closed form of each step.

Algorithm 2: EM algorithm for our method

Data: initialize parameters $\{\pi_j, \boldsymbol{q}_j, \boldsymbol{\delta}_j\}_{j=1}^{G}$

while *not convergency of ln* $\mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$ *or parameters* **do**

E-step:

Calculate $h_j^{(i)}$ using temporary value of parameters for each read $x^{(i)} \in D$ and each genome $j = 1, ..., G$:

$$h_j^{(i)} = \frac{\pi_j \delta_j^{1-y_i} q_j^{(i)}}{\sum_{k=1}^{G} \pi \delta_j^{1-y_i} q_j^{(i)}}$$

M-step:

Calculate new values for parameters using temporary values of h_j. For each genome $j = 1, ..., G$:

$$\pi_j = \frac{\sum_{i=1}^{R} h_j^{(i)} + a_j}{N + \sum_{j=1}^{G} a_k}$$

$$\delta_j = \frac{\sum_{i=1}^{R} (1-y^{(i)}) h_j^{(i)} + b_j}{\sum_{i=1}^{R} (1-y^{(i)}) + \sum_{j=1}^{G} b_j}$$

Calculate temporary value of log-likelihood function:

$$\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = \sum_{i=1}^{R} \ln \sum_{j=1}^{G} \pi_j p(x^{(i)}|\boldsymbol{\theta}_j)$$

Final score is π_j and in the first substep it represents the proportion of reads that are mapped to the species j. We know that each mapped read is not uniquely mapped to one species, it is uniquely mapped to one marker but that marker could belong to the more species and π_j will be distributed over these species so it could not achieve a high value. To solve this problem, in the second substep we map a metagenomic reads against a database that contains genomes of the first 5 species according to the value of π_j in the first substep. After the mapping, the classifier determines which genome is present in a sample and final π_j represents the proportion of reads that are mapped to the genome.

4. Implementation

This application can be divided into two parts. First part is the core of this application and it is written in Python. Second part is web application which starts real-time analysis and presents results during mapping. Web application is written in Java using Play Framework. In the next two sections are overview of each of this parts.

4.1. Core

Implementation of the application consists of four modules. All of these are implemented in python. First, reduced database need to be prepared. The task is, using the specific markers annotation, to construct the database that contains only marker genes in order to speed-up the mapping and reduce the possibility of false positive hits. As mentioned in the Chapter 3, the markers from Metaphlan were used in this work. This file contains only GI and position of specific gene used as marker. Therefore, the first task is to pair each GI from the marker file with a nucleotide sequence. Afterwards, the task is to find the taxonomy number of a specific clade. This algorithm works on the level of species so if some marker is determined for a clade above the species it is necessary to set that marker on the species below that clade first, and vice verse if that marker is unique for some strain. In order to get the taxonomy number and taxonomy subtree from the name of a clade I used *names.dmp* and *nodes.dmp* downloaded from the NCBI ftp server. The same clade could have many corresponding taxonomy numbers depending on the type of name (scientific name, equivalent name...). In this module, I used only the scientific name in order to get the taxonomy number. Names of the clades in the text file *markers_info.txt* are given replacing all the non alphanumeric characters with '_'. Similarly, we need to do the same thing with the names from *names.dmp*. Some markers can belong to other strains that are not under given the clade. Therefore, we also need to find the taxonomy number of the species above those strains.

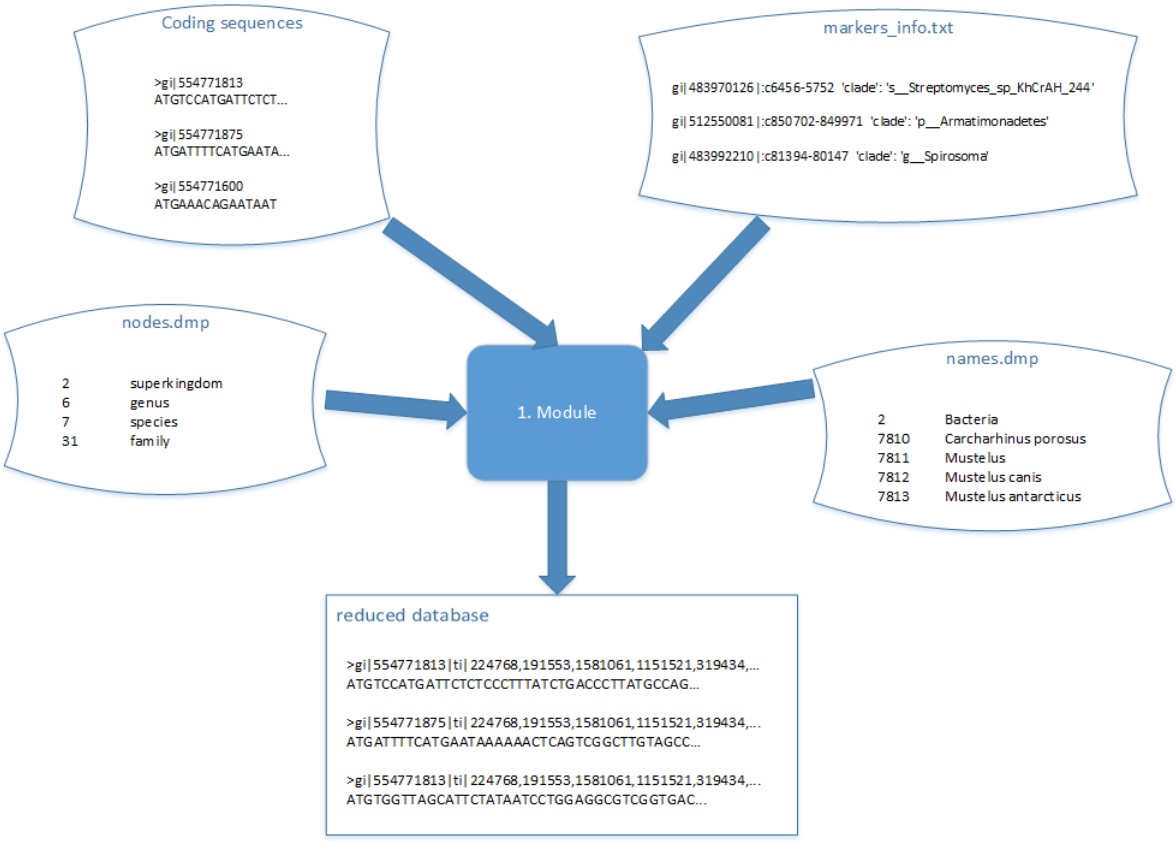

Figure 4.1: Preparing reduced database

Second module is used for mapping the sequenced reads against reduced database. This tool is adapted for real-time analysis of reads obtained by sequencer so this module reacts on every new set of reads and map it against the reduced database. Graphmap was used for mapping [15]. However, it is not mandatory; it is possible to use every tool that gives output in SAM format.

Next module is the core of the whole system. In this module I implemented an algorithm which reads the output from the previous module and detect species in the sample. Model used in this algorithm is described in Chapter 3, and the code is organised in a such a way that makes it easy to change the model of the algorithm.

Last module is used to produce the final output. This module gets results from previous module and gives them in appropriate format. These results are read by web application and presented to user.

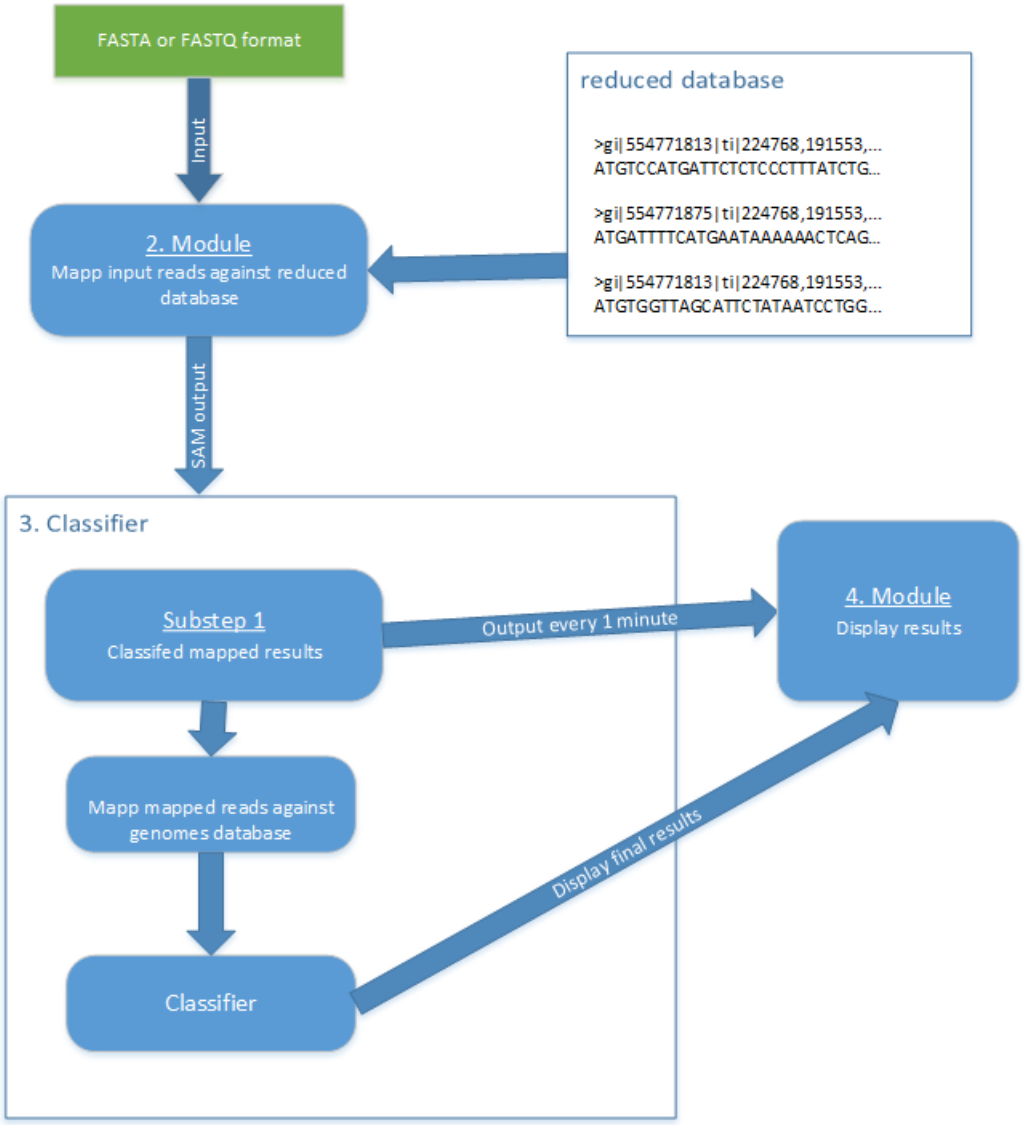

Figure 4.2: Workflow of the algorithm

4.2. Web application

Web application is written in Java using Play Framework and it follows the Model-view-controller (MVC) architectural pattern applied to the Web architecture.

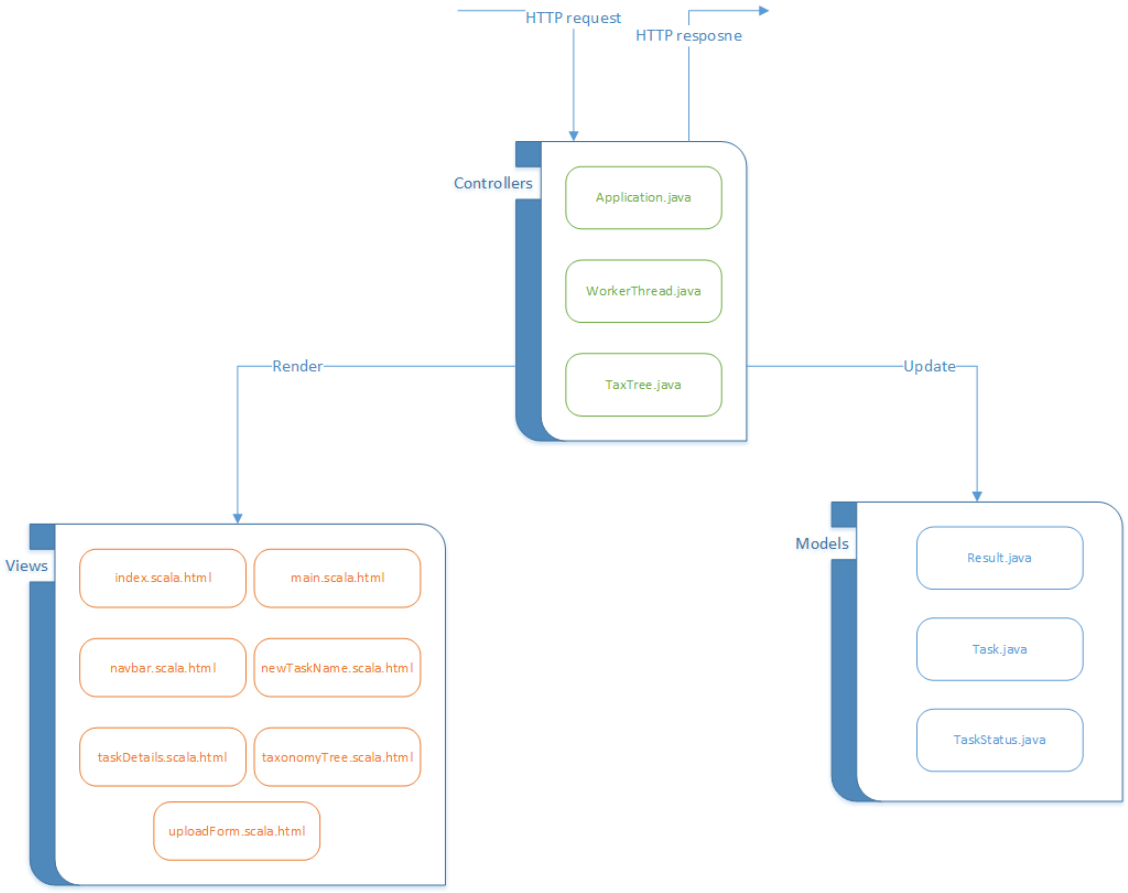

Figure 4.3: Web page organisations

The main function of the web is to upload the reads and to present the results. After the reads are uploaded the mapping process begin. As long as this process is active, the results are refreshed every minute. Home page of the web application on which are shown all started tasks is shown on the figure 4.4.

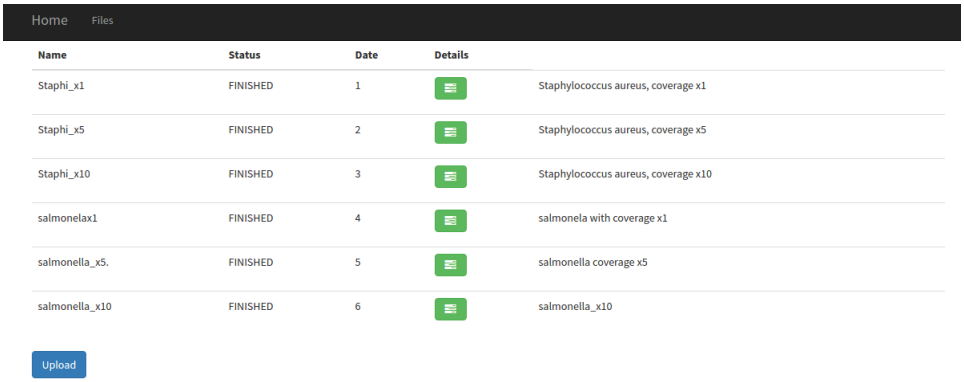

Figure 4.4: Home page of the web application

There is also a page which shows results from real-time analysis. It contains one graph on which are shown first 10 results from algorithm.

Figure 4.5: Graph on the web page for results

5. Results

Our method was tested on a sets of synthetic bacteria samples produced using PacBio reads simulator (PBSIM) [16]. PacBio sequencing is a method for real-time sequencing and does not require a pause between step. PacBio sequencing offers much longer read lengths and faster runs than SGS methods but is hindered by a lower throughput, higher error rate, and higher cost per base [17]. PBSIM produces a set of simulated reads in the FASTQ format and a list of alignments between a reference sequence and simulated reads in the MAF format. The parameters with which this tool was ran is given in the table 5.1.

option	**value**
data-type	CLR
depth	*one of* {1,5,10,15}
length-mean	9753
length-sd	4260
length-min	5
length-max	100000
accuracy-mean	0.9
accuracy-sd	0.05
accuracy-min	0.7
difference-ratio	50:30:20

Table 5.1: Pbsim parameters

The following table shows the strains that are used in testing.

GI	TI	species TI	Name
49240382	282458	1280	Staphylococcus aureus
374352002	1132507	28901	Salmonella enterica
1001954050	1263871	573	Klebsiella pneumoniae

Table 5.2: Strains used for testing

Each strain was tested with the different coverage against the marker database. We choose the coverage from the set of {x1, x5, x10} when we having the reads of one bacteria, and from the set of {x5, x10, x15} when having made the mixture of these bacteria. The following tables show the top 5 results with the different coverage for each of these bacteria. For each set we have two tables. In the first are the results after the first substep, and in the second table are the results after the mapping reads to the genomes. In the first column of a table is taxonomy number of the result, in the second is the name of the species or the strain, and in the third column is the proportion of reads that are mapped to that result. The goal of this testing is to find the strain from which the synthetic set was created with the high value of the final score.

Salmonella enterica

The test was performed on the synthetic sets created from the strain *Salmonella enterica subsp. enterica serovar Typhi str. P-stx-12* with the goal of getting this strain as the result after the second substep.

TI	Name	Final score
28901	Salmonella enterica	0.25
947561	Yersinia enterocolitica IP2222	0.002
914128	Serratia symbiotica str. Tucson	0.002
469595	Citrobacter sp. 30_2	0.002
1328380	Klebsiella pneumoniae MGH 48	0.002

Table 5.3: Salmonella enterica coverrage x1, results obtained after the first substep

TI	Name	Final score
1132507	Salmonella enterica subsp. enterica serovar Typhi str. P-stx-12	0.786
527001	Salmonella enterica subsp. enterica serovar Typhi str. Ty21a	0.012
209261	Salmonella enterica subsp. enterica serovar Typhi str. Ty2	0.012
220341	Salmonella enterica subsp. enterica serovar Typhi str. CT18	0.011
1320309	Salmonella enterica subsp. enterica serovar Bovismorbificans str. 3114	0.010

Table 5.4: Salmonella enterica coverrage x1, results obtained after the second substep

TI	Name	Final score
28901	Salmonella enterica	0.5
947561	Yersinia enterocolitica IP2222	0.002
914128	Serratia symbiotica str. Tucson	0.001
469595	Citrobacter sp. 30_2	0.001
1328380	Klebsiella pneumoniae MGH 48	0.001

Table 5.5: Salmonella enterica coverrage x5, results obtained after the first substep

TI	Name	Final score
1132507	Salmonella enterica subsp. enterica serovar Typhi str. P-stx-12	0.89
209261	Salmonella enterica subsp. enterica serovar Typhi str. Ty2	0.01
527001	Salmonella enterica subsp. enterica serovar Typhi str. Ty21a	0.008
220341	Salmonella enterica subsp. enterica serovar Typhi str. CT18	0.005
439843	Salmonella enterica subsp. enterica serovar Schwarzengrund str. CVM19633	0.002

Table 5.6: Salmonella enterica coverrage x5, results obtained after the second substep

TI	Name	Final score
28901	Salmonella enterica	0.17
947561	Yersinia enterocolitica IP2222	0.0002
914128	Serratia symbiotica str. Tucson	0.0001
469595	Citrobacter sp. 30_2	0.0001
1328380	Klebsiella pneumoniae MGH 48	0.0001

Table 5.7: Salmonella enterica coverrage x10, results obtained after the first substep

TI	Name	Final score
1132507	Salmonella enterica subsp. enterica serovar Typhi str. P-stx-12	0.89
220341	Salmonella enterica subsp. enterica serovar Typhi str. CT18	0.015
209261	Salmonella enterica subsp. enterica serovar Typhi str. Ty2	0.009
527001	Salmonella enterica subsp. enterica serovar Typhi str. Ty21a	0.009
295319	SSalmonella enterica subsp. enterica serovar Paratyphi A str. ATCC 9150	0.002

Table 5.8: Salmonella enterica coverrage x10, results obtained after the second substep

Staphylococcus aureus

The test was performed on the synthetic sets created from the strain *Staphylococcus aureus subsp. aureus MRSA252* with the goal of getting this strain as the result after the second substep.

TI	Name	Final score
1280	Staphylococcus aureus	0.16
2130	Ureaplasma urealyticum	0.03
5808	Cryptosporidium muris	0.005
29555	Mycoplasma canis	0.005
5833	Plasmodium falciparum	0.004

Table 5.9: Staphylococcus aureus coverage x1, results obtained after the first substep

TI	Name	Final score
282458	Staphylococcus aureus subsp. aureus MRSA252	0.74
46170	Staphylococcus aureus subsp. aureus	0.01
548473	Staphylococcus aureus subsp. aureus TCH60	0.01
585143	Staphylococcus aureus subsp. aureus 55/2053	0.01
703339	Staphylococcus aureus 04-02981	0.01

Table 5.10: Staphylococcus aureus coverage x1, results obtained after the second step

TI	Name	Final score
1280	Staphylococcus aureus	0.05
2130	Ureaplasma urealyticum	0.003
29555	Mycoplasma canis	0.0009
1345695	Clostridium saccharobutylicum DSM 13864	0.0009
748449	Halobacteroides halobius DSM 5150	0.0009

Table 5.11: Staphylococcus aureus coverage x1, results obtained after the first substep

TI	Name	Final score
282458	Staphylococcus aureus subsp. aureus MRSA252	0.89
46170	Staphylococcus aureus subsp. aureus	0.014
548473	Staphylococcus aureus subsp. aureus TCH60	0.004
585143	Staphylococcus aureus subsp. aureus 55/2053	0.004
703339	Staphylococcus aureus 04-02981	0.01

Table 5.12: Staphylococcus aureus coverage x1, results obtained after the second step

TI	Name	Final score
1280	Staphylococcus aureus	0.05
2130	Ureaplasma urealyticum	0.003
868864	Desulfurobacterium thermolithotrophum DSM 11699	0.001
1341181	Flavobacterium limnosediminis JC2902	0.001
5808	Cryptosporidium muris	0.001

Table 5.13: Staphylococcus aureus coverage x10, results obtained after the first substep

TI	Name	Final score
282458	Staphylococcus aureus subsp. aureus MRSA252	0.89
46170	Staphylococcus aureus subsp. aureus	0.014
585143	Staphylococcus aureus subsp. aureus 55/2053	0.004
548473	Staphylococcus aureus subsp. aureus TCH60	0.004
703339	Staphylococcus aureus 04-02981	0.01

Table 5.14: Staphylococcus aureus coverage x10, results obtained after the second step

Klebsiella pneumoniae

The test was performed on the synthetic sets created from the strain *Klebsiella pneumoniae ATCC BAA-2146* with the goal of getting this strain as the result after the second substep.

TI	Name	Final score
573	Klebsiella pneumoniae	0.0064
665944	Klebsiella sp. 4_1_44FAA	0.0057
749535	Klebsiella sp. MS 92-3	0.0057
1269006	Klebsiella pneumoniae 909957	0.0057
1182695	Klebsiella sp. KTE92	0.0054

Table 5.15: Klebsiella pneumoniae coverage x1, results obtained after the first substep

TI	Name	Final score
1263871	Klebsiella pneumoniae ATCC BAA-2146	0.75
1380908	Klebsiella pneumoniae JM45	0.02
72407	Klebsiella pneumoniae subsp. pneumoniae	0.016
484021	Klebsiella pneumoniae subsp. pneumoniae NTUH-K2044	0.016
1244085	Klebsiella pneumoniae CG43	0.016

Table 5.16: Klebsiella pneumoniae coverage x1, results obtained after the second step

TI	Name	Final score
573	Klebsiella pneumoniae	0.17
460086	Kribbella catacumbae	0.0006
1157680	Caulobacter sp. JGI 0001013-D04	0.00046
636	Edwardsiella tarda	0.0004
1182695	Klebsiella sp. KTE92	0.0004

Table 5.17: Klebsiella pneumoniae coverage x5, results obtained after the first substep

TI	Name	Final score
1263871	Klebsiella pneumoniae ATCC BAA-2146	0.92
484021	Klebsiella pneumoniae subsp. pneumoniae NTUH-K2044	0.008
72407	Klebsiella pneumoniae subsp. pneumoniae	0.007
1380908	Klebsiella pneumoniae JM45	0.007
1244085	Klebsiella pneumoniae CG43	0.005

Table 5.18: Klebsiella pneumoniae coverage x5, results obtained after the second step

TI	Name	Final score
573	Klebsiella pneumoniae	0.10
5808	Cryptosporidium muris	0.00019
749535	Klebsiella sp. MS 92-3	0.00018
936565	Klebsiella sp. OBRC7	0.00015
1206777	Pseudomonas sp. Lz4W	00013

Table 5.19: Klebsiella pneumoniae coverage x10, results obtained after the first substep

TI	Name	Final score
1263871	Klebsiella pneumoniae ATCC BAA-2146	0.94
1380908	Klebsiella pneumoniae JM45	0.0098
72407	Klebsiella pneumoniae subsp. pneumoniae	0.00567
1328324	Klebsiella pneumoniae subsp. pneumoniae KPNIH27	0.0039
1123862	Klebsiella pneumoniae subsp. pneumoniae Kp13	0.0031

Table 5.20: Klebsiella pneumoniae coverage x10, results obtained after the second step

Mixture of previos three datasets

The test was performed on the synthetic sets created of previously defined sets. Mixed set is combination of: *Salmonella enterica* with the coverage x15, *Staphylococcus aureus* with the coverage x10 and *Klebsiella pneumoniae* with the coverage x5. The goal is to find *Salmonella enterica* as the first result, *Staphylococcus aureus* as the second result and *Klebsiella pneumoniae* as the third result.

TI	Name	Final score
28901	Salmonella enterica	0.13
1280	Staphylococcus aureus	0.02
573	Klebsiella pneumoniae	0.009
749535	Klebsiella sp. MS 92-3	0.005
665944	Klebsiella sp. 4_1_44FAA	0.004

Table 5.21: Mixture of Salmonella enterica, Staphylococcus aureus, Klebsiella pneumoniae with the coverage x15, x10, x5 respectively; results obtained after the first step

TI	Name	Final score
1132507	Salmonella enterica subsp. enterica serovar Typhi str. P-stx-12	0.64
282458	Staphylococcus aureus subsp. aureus MRSA252	0.15
1263871	Klebsiella pneumoniae ATCC BAA-2146	0.064
220341	Salmonella enterica subsp. enterica serovar Typhi str. CT18	0.008
527001	Salmonella enterica subsp. enterica serovar Typhi str. Ty21a	0.005

Table 5.22: Mixture of Salmonella enterica, Staphylococcus aureus, Klebsiella pneumoniae with the coverage x15, x10, x5 respectively; results obtained after the second step

The previous tables show that our method is very precise and can precisely detect species presented in a metagenomic sample with the different coverage.

To compare our method with Pathoscope[18] we used Salmonella enterica set with the coverage x1 which require high precision because the coverage is small and the genome is not long. Table 5.23 shows memory consumption and running time of the our method and Pathoscope.

	Pathoscope	Our tool
RAM	400 GB	18 GB
Time	6324 s	485 s

Table 5.23: Comparison of our method and Pathoscope, memory consumption and running time

It is obvious that our method is roughly 13 times faster than Pathoscope and required 22 times less memory than Pathoscope.

The tables 5.24 and 5.25 show the final results obtained by Pathoscope and our method.

TI	Name	Final score
209261	Salmonella enterica subsp. enterica serovar Typhi str. Ty2	0.77
1132507	Salmonella enterica subsp. enterica serovar Typhi str. P-stx-12	0.05
1003191	Salmonella enterica subsp. enterica serovar Tennessee str. TXSC_TXSC08-19	0.005
1320309	Salmonella enterica subsp. enterica serovar Bovismorbificans str. 3114	0.005
984211	Salmonella enterica subsp. enterica serovar Anatum str. ATCC BAA-1592	0.005

Table 5.24: Results obtained by Pathoscope for Salmonella enterica x1 set

TI	Name	Final score
1132507	Salmonella enterica subsp. enterica serovar Typhi str. P-stx-12	0.786
527001	Salmonella enterica subsp. enterica serovar Typhi str. Ty21a	0.012
209261	Salmonella enterica subsp. enterica serovar Typhi str. Ty2	0.012
220341	Salmonella enterica subsp. enterica serovar Typhi str. CT18	0.011
1320309	Salmonella enterica subsp. enterica serovar Bovismorbificans str. 3114	0.010

Table 5.25: Results obtained by our method for Salmonella enterica x1 set

Comparison of the results from the tables 5.24 and 5.25 show that our method is more precise than Pathosope. The strain from which synthetic set was created is not given as the first result by Pathoscope, whereas our methoud gives it with the high value of the final score. The reason for that we can find in the fact that we avoid false positive hits using the reduced database in the first step, and mapping against genomes under given clades in the second substep.

6. Conclusion

Fast and cheap metagenomic sample analysis could be very useful for medical diagnosis. Traditional laboratory methods are either long-lasting or oriented towards a single pathogen species.

In this thesis we developed a tool for metagenomic sample analysis which use MinION sequencing devices by Oxford Nanopore Technologies (ONT). Since the databases of bacteria genomes could be very large so we decided to reduce database in order to make this method faster and resistant to false positive hits. Using markers from Metaphlan we reduced huge database to the more acceptable one with the size of 1 GB.

We showed that it is possible to build functional and precise tool for the metagenome analysis that could do real-time analysis. Presently,a mixture of bacteria with the human genome causes a lot of problem with the humans reads that are mapped to some bacteria markers. We need to find compromise between speed of real-time analysis and detection of human genome. In the future we can try with our own marker dataset instead of dataset from Metaphlan.

BIBLIOGRAPHY

[1] Mark B Gerstein, Can Bruce, Joel S Rozowsky, Deyou Zheng, Jiang Du, Jan O Korbel, Olof Emanuelsson, Zhengdong D Zhang, Sherman Weissman, and Michael Snyder. What is a gene, post-encode? history and updated definition. *Genome research*, 17(6):669–681, 2007.

[2] Torsten Thomas, Jack Gilbert, and Folker Meyer. Metagenomics-a guide from sampling to data analysis. *Microbial informatics and experimentation*, 2(1):1, 2012.

[3] Thomas J Sharpton. An introduction to the analysis of shotgun metagenomic data. 2014.

[4] Fred Sanger and Alan R Coulson. A rapid method for determining sequences in dna by primed synthesis with dna polymerase. *Journal of molecular biology*, 94(3):441–448, 1975.

[5] Joseph Palca. Human genome: Department of energy on the map. *Nature*, 321:371, 1986.

[6] Robert L Sinsheimer. The santa cruz workshop—may 1985. *Genomics*, 5(4):954–956, 1989.

[7] David Deamer, Mark Akeson, and Daniel Branton. Three decades of nanopore sequencing. *Nature Biotechnology*, 34(5):518–524, 2016.

[8] Vivien Marx. Nanopores: a sequencer in your backpack. *Nature methods*, 12(11):1015–1018, 2015.

[9] Tamas Szalay and Jene A Golovchenko. De novo sequencing and variant calling with nanopores using poreseq. *Nature biotechnology*, 33(10):1087–1091, 2015.

[10] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, et al. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[11] Derrick E Wood and Steven L Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol*, 15(3):R46, 2014.

[12] Nicola Segata, Levi Waldron, Annalisa Ballarini, Vagheesh Narasimhan, Olivier Jousson, and Curtis Huttenhower. Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature methods*, 9(8):811–814, 2012.

[13] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[14] Owen E Francis, Matthew Bendall, Solaiappan Manimaran, Changjin Hong, Nathan L Clement, Eduardo Castro-Nallar, Quinn Snell, G Bruce Schaalje, Mark J Clement, Keith A Crandall, et al. Pathoscope: species identification and strain attribution with unassembled sequencing data. *Genome research*, 23(10):1721–1729, 2013.

[15] Ivan Sovic, Mile Sikic, Andreas Wilm, Shannon Nicole Fenlon, Swaine Chen, and Niranjan Nagarajan. Fast and sensitive mapping of error-prone nanopore sequencing reads with graphmap. *bioRxiv*, page 020719, 2015.

[16] Yukiteru Ono, Kiyoshi Asai, and Michiaki Hamada. Pbsim: Pacbio reads simulator—toward accurate genome assembly. *Bioinformatics*, 29(1):119–121, 2013.

[17] Anthony Rhoads and Kin Fai Au. Pacbio sequencing and its applications. *Genomics, proteomics & bioinformatics*, 13(5):278–289, 2015.

[18] Changjin Hong, Solaiappan Manimaran, Ying Shen, Joseph F Perez-Rogers, Allyson L Byrd, Eduardo Castro-Nallar, Keith A Crandall, and William Evan Johnson. Pathoscope 2.0: a complete computational framework for strain identification in environmental or clinical sequencing samples. *Microbiome*, 2(1):1, 2014.

[19] J Handelsman, J Tiedje, L Alvarez-Cohen, M Ashburner, IKO Cann, EF Delong, WF Doolittle, CM Fraser-Liggett, A Godzik, JI Gordon, et al. The new science of metagenomics: revealing the secrets of our microbial planet. *Nat Res Council Report*, 13, 2007.

[20] Qichao Tu, Zhili He, and Jizhong Zhou. Strain/species identification in metagenomes using genome-specific markers. *Nucleic acids research*, 42(8):e67–e67, 2014.

[21] Andy Kilianski, Jamie L Haas, Elizabeth J Corriveau, Alvin T Liem, Kristen L Willis, Dana R Kadavy, C Nicole Rosenzweig, and Samuel S Minot. Bacterial and viral identification and differentiation by amplicon sequencing on the minion nanopore sequencer. *Gigascience*, 4(12):10–1186, 2015.

[22] Sissel Juul, Fernando Izquierdo, Adam Hurst, Xiaoguang Dai, Amber Wright, Eugene Kulesha, Roger Pettett, and Daniel J Turner. What's in my pot? real-time species identification on the minion. *bioRxiv*, page 030742, 2015.

[23] Benjamin Buchfink, Daniel H Huson, and Chao Xie. Metascope-fast and accurate identification of microbes in metagenomic sequencing data. *arXiv preprint arXiv:1511.08753*, 2015.

[24] Alexander L Greninger, Samia N Naccache, Scot Federman, Guixia Yu, Placide Mbala, Vanessa Bres, Doug Stryke, Jerome Bouquet, Sneha Somasekar, Jeffrey M Linnen, et al. Rapid metagenomic identification of viral pathogens in clinical samples by real-time nanopore sequencing analysis. *Genome medicine*, 7(1):1–13, 2015.

[25] Philip M Ashton, Satheesh Nair, Tim Dallman, Salvatore Rubino, Wolfgang Rabsch, Solomon Mwaigwisya, John Wain, and Justin O'Grady. Minion nanopore sequencing identifies the position and structure of a bacterial antibiotic resistance island. *Nature biotechnology*, 33(3):296–300, 2015.

[26] Robert C Edgar. Search and clustering orders of magnitude faster than blast. *Bioinformatics*, 26(19):2460–2461, 2010.

[27] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*, volume 2. Taylor & Francis, 2014.

[28] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[29] Jo Handelsman. Metagenomics: application of genomics to uncultured microorganisms. *Microbiology and molecular biology reviews*, 68(4):669–685, 2004.

[30] Ruth R Miller, Vincent Montoya, Jennifer L Gardy, David M Patrick, and Patrick Tang. Metagenomics for pathogen detection in public health. *Genome Med*, 5(9):81, 2013.

[31] Sofia Morfopoulou and Vincent Plagnol. Bayesian mixture analysis for metagenomic community profiling. *Bioinformatics*, 31(18):2930–2938, 2015.

[32] Eric S Lander, Lauren M Linton, Bruce Birren, Chad Nusbaum, Michael C Zody, Jennifer Baldwin, Keri Devon, Ken Dewar, Michael Doyle, William FitzHugh, et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.

[33] Timothy J Denison, Alexis Sauer-Budge, Jene A Golovchenko, Amit Meller, Eric Brandin, and Daniel Branton. Characterization of individual polymer molecules based on monomer-interface interactions, June 2 2015. US Patent 9,046,483.

[34] John J Kasianowicz, Eric Brandin, Daniel Branton, and David W Deamer. Characterization of individual polynucleotide molecules using a membrane channel. *Proceedings of the National Academy of Sciences*, 93(24):13770–13773, 1996.

[35] Andy Petrella. *Learning Play! Framework 2*. Packt Publishing, 2013.

[36] Jan Šnajder. *Machine learning*. 2014.

[37] Franck Picard. An introduction to mixture models. *Statistics for Systems Biology, Research Report*, (7), 2007.

Real-Time Analysis of a Metagenomic Sample Obtained by Nanopore Based Sequencing Technology

Sažetak

Brza i jeftina analiza metagenomoskog uzorka može biti korisna za dijagnostiku bolesti, kontrolu kvalitete hrane i utvrđivanje štetnih nametnika na biljkama. Tradicionalne laboratorijske metode su ili dugotrajne ili namijenjene za samo jednu vrstu. Uređaji za sekvenciranje MinION tvrtke Oxford Nanopore Technologies relativno su jeftini i pogodni za rad na terenu jer su lako prenosivi. Napravljen je alat koji u prvom koraku pronalazi sve organizme čije sekvence u svom dijelu imaju veliku sličnost s očitanim uzorcima, a nakon toga se utvrđuje koji organizmi su stvarno prisutni u tom uzorku.

Ključne riječi: metagenomika, dijagnostika, liječenje

Tool for fast searching of protein sequences in databases

Abstract

Fast and cheap metagenomic sample analysis could be very useful for medical diagnosis, food quality control and discovering harmful parasites on plants. Traditional laboratory methods are either long-lasting or oriented towards a single pathogen species. MinION sequencing devices by Oxford Nanopore Technologies are relatively cheap and suitable for application in the field. We developed a tool which in the first step discovers all organisms whose sequence have high similarity to the reads obtained from the samples and after that determines the organisms present in the sample.

Keywords: metagenomics, diagnostics, treatment