BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER SYSTEMS
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

TOOL FOR VISUALIZATION OF MICROBIOME DATA

NÁSTROJ PRO VIZUALIZACI MIKROBIOMOVÝCH DAT

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **SILVIA MIŠÁKOVÁ**
AUTOR PRÁCE

SUPERVISOR **STANISLAV SMATANA, Ing.**
VEDOUCÍ PRÁCE

BRNO 2021

Brno University of Technology
Faculty of Information Technology

Department of Computer Systems (DCSY) Academic year 2020/2021

Bachelor's Thesis Specification

Student: **Mišáková Silvia**
Programme: Information Technology
Title: **Tool for Visualization of Microbiome Data**
Category: Biocomputing
Assignment:

1. Analyse existing tools for visualization of microbial data (Emperor) and for general multidimensional data visualization (Tensorflow Embedding Projector). Asses their strengths and weaknesses with a focus on the needs of microbial research.

2. Based on your findings, propose new tool for microbial data visualization, which addresses some of the shortcomings of existing methods, especially relating to the ease of use and usefulness of their interface.

3. Implement the proposed tool.

4. Evaluate achieved results and discuss possibilities of future continuation.

Recommended literature:

- According to supervisor's instructions

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Smatana Stanislav, Ing.**
Head of Department: Sekanina Lukáš, prof. Ing., Ph.D.
Beginning of work: November 1, 2020
Submission deadline: May 12, 2021
Approval date: October 30, 2020

Abstract

This Bachelor's thesis focuses on the development of a new tool for the visualization of microbiome data. The developed tool uses Principal Component Analysis (PCA) and Principal Coordinates Analysis (PCoA) for dimension reduction. Bray-Curtis difference and UniFrac distance metric are used for distance matrix calculation. Then, the processed data is colored based on user-selected metadata.

The results are presented by two types of graphs. The first is a bar chart and shows the proportion of each principal component. The second, scatter chart, visualize the final result of the required analysis.

As part of the development of the tool, the possibility to download the calculated matrix and also a table of the first N main components calculated by the analysis were added.

Abstrakt

Táto práca sa zameriava na vytvorenie nového nástroja pre vizualizáciu mikrobiomových dát. Vytvorený nástroj používa pre redukciu dimenzií analýzu hlavných komponent (PCA) a analýzu hlavných súradníc (PCoA). V prípade výpočtu dištančnej matice sú použité metriky Bray-Curtis odlišnosť a UniFrac. Spracované dáta sú následne ofarbené na základe užívateľom zvolených metadát.

Výsledky sú prezentované pomocou dvoch typov grafov. Prvý z nich je stĺpcový a zobrazuje podiel každej hlavnej zložky. Druhý, bodový graf, vizualizuje konečný výsledok požadovanej analýzy.

V rámci práce bola pridaná možnosť stiahnuť si vypočítanú maticu a taktiež tabuľku prvých N hlavných zložiek vypočítaných danou analýzou.

Keywords

16S rRNA, Bioinformatics, Bray–Curtis dissimilarity, Distance matrix, DNA, Emperor, Meta-genomics, Microbiome, Microbiota, PCA, PCoA, Phylogenetic tree, RNA, Tensorflow Embedding Projector, UniFrac, visualization

Klíčová slova

16S rRNA, bioinformatika, Bray-Curtis odlišnosť, dištančná matica, DNA, Emperor, fylogenetický strom, metagenomika, mikrobióm, mikrobiota, PCA, PCoA, RNA, Tensorflow Embedding Projector, UniFrac, vizualizácia

Reference

MIŠÁKOVÁ, Silvia. *Tool for Visualization of Microbiome Data.* Brno, 2021. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Stanislav Smatana, Ing.

Rozšířený abstrakt

Mikrobióm je súbor všetkých mikróbov, respektíve mikroorganizmov (napr. baktérií, vírusov, prvkov, húb), žijúcich nie len vo vnútri, ale aj na povrchu tela. Pre ľudský život je priam nevyhnutný. Pomáha s trávením, produkciou rôznych vitamínov, reguláciou imunitného systému a chráni organizmus pred baktériami, ktoré spôsobujú rôzne ochorenia. Pokiaľ mikrobióm nefunguje správne, môže dochádzať k rôznym autonómnym chorobám (cukrovka, svalová dystrofia), tráviacim a kožným problémom, a taktiež môže ovplyvňovať aj našu náladu a podporiť vznik depresie. Pre udržanie mikrobiómu v čo najlepšom stave je veľmi dôležitá zdravá a vyvážená strava a pohyb.

Mikrobiómové dáta sa získavajú odobraním potrebnej vzorky, jej spracovaní a následnou sekvenáciou. Nad získanými sekvenciami sa vykonáva kvalitatívna alebo kvantitatívna analýza. Výsledky tejto analýzy sú n-rozmerné dáta, ktoré sú pre človeka len ťažko čitateľné. Preto sa ďalej spracovávajú metódami na redukciu dimenzií do 2D alebo 3D priestoru. Takto spracované dáta môžu byť následne vizualizované do grafov, prípadne ofarbované na základe príslušných metadát a ďalej analyzované. Cieľom tejto práce bolo vytvoriť nový nástroj pre vizualizáciu mikrobiómových dát.

Úvodná časť a druhá kapitola sa venuje podrobnejšiemu uvedeniu do problému najmä po biologickej stránke. Sú tu vysvetlené základné pojmy ako napríklad DNA, RNA, metagenomika, 16S rRNA či mikrobióm. Tieto pojmy sú potrebné pre komplexné pochopenie danej problematiky.

V tretej kapitole sú popísané metódy a metriky, ktoré navrhnutý nástroj používa. Pri redukcii dimenzií ide o analýzu hlavných komponentov (Principal Component Analysis) a analýzu hlavných súradníc (Principal coordinates analysis). V prípade výpočtu beta diverzity sa používajú metriky Bray-Curtis odlišnosť a UniFrac (vážený a nevážený). Dôležitou súčasťou metriky UniFrac je aj fylogenetický strom.

Nasledujúca kapitola je venovaná analýze a porovnaniu existujúcich nástrojov Emperor a Tensorflow Embedding Projector. Každý z týchto nástrojov sa zameriava na iný spôsob redukcie n-rozmerných dát do 2D alebo 3D priestoru. Cieľom nového nástroja je zachovanie a prepojenie niektorých silných stránok týchto nástrojov, ktoré by mohli byť pre cieľovú skupinu užívateľov potrebné a eliminácia ich nedostatkov.

Piata kapitola popisuje návrh samotného nástroja. Je tu popísaná základná funkcionalita a rozloženie kľúčových prvkov nástroja. Taktiež sa tu nachádza popis vstupných a výstupných dát. Vstupné dáta sa zadávajú do formulára vo webovom rozhraní. Výstupom nástroja sú dva typy grafov a súbory na stiahnutie. Jeden z grafov je stĺpcový a zobrazuje podiely jednotlivých hlavných zložiek v analýze. Druhý graf je bodový a zobrazuje samotný výsledok požadovanej analýzy, ktorý je ofarbovaný podľa príslušných metadát. Užívateľ si taktiež môže stiahnuť vypočítanú dištančnú maticu alebo tabuľku prvých N hlavných zložiek vypočítaných danou analýzou.

Ďalšia kapitola popisuje implementáciu. Zaoberá sa výberom programovacích jazykov, použitými nástrojmi, knižnicami, technológiami a komunikáciou medzi nimi.

Kapitola valuation sa zameriava na testovanie nástroja a dosiahnuté výsledky.

V závere sa nachádza zhrnutie dosiahnutých výsledkov a návrhy na prípadné rozšírenie a vylepšenie nástroja.

Tool for Visualization of Microbiome Data

Declaration

I hereby declare that this Bachelor's thesis was written as an original work by the author under the supervision of Ing. Stanislav Smatana. I have listed all the literary sources, publications, and other sources which were used during the preparation of this thesis.

.....
Silvia Mišáková
May 17, 2021

Acknowledgements

I would like to thank my supervisor Ing. Stanislav Smatana for supervising this thesis and for all the valuable ideas he gave me and also, to Janka Puterová for her advice on the text.

Contents

Chapter 1

Introduction

Nowadays, technology is everywhere we look and it is very significant for our lives. People are trying to use its potential in various industries. Biology and medicine are no exceptions.

Bioinformatics is one of the most popular interdisciplinary scientific fields in the world. It uses a combination of informatics, biology, biochemistry, statistics, and applied mathematics for data analysis and its interpretation. In this scientific discipline problems are usually solved at the molecular level.

The best-known areas of research in this field include genome annotation, sequence, gene expression level, or gene regulation analysis protein structure prediction, and comparative genomics. To be able to process a huge amount of data, experts in this discipline try to develop new software tools and methods.

This thesis focuses on the visualization of microbiome data. Microbiota is a term used to describe all microorganisms living inside and outside a human body and their interactions. It may contain both beneficial and non-beneficial bacteria. These microorganisms live in the whole body, but one of the most important is situated in the human intestine. Mainly the intestinal microbiota is connected to other systems. Brain, emotional and cardiovascular health are based on it too [37].

Nowadays, science allows the extraction of information about microbiota. This data represents a huge amount of information. In many cases, it is really difficult to analyze it.

In this thesis, an existing tool for visualization of microbial data (Emperor [47]) and for general multidimensional data visualization (TensorFlow Embedding Projector [43]) are compared. Based on this comparison, the new tool is designed. It should be more user-friendly and tries to keep strengths and eliminates weaknesses of existing tools.

This thesis is divided into eight chapters.

Chapter 2, is dedicated to the necessary biological terms such as DNA, RNA, metagenomics, 16S rRNA, microbiota. Theoretical knowledge from the field of biomedicine, necessary for understanding and creating this thesis, is described in Chapter 2.

In the third chapter, named "Methods used for microbiome analysis", are described calculation methods. Terms such as PCA, PCoA, metrics for beta diversity computation - Bray-Curtis and UniFrac are explained there. For UniFrac are also really important phylogenetic trees.

The fourth chapter, 4, describes existing tools Emperor and TensorFlow Embedding Projector. The weaknesses of these tools, which the new tool can solve, are mentioned here.

Chapter 5 summarizes the functional and design requirements from a user perspective. This chapter deals with the needs of potential users of this tool. There is also a description of input and output file types.

The sixth chapter „Implementation“ is dedicated to used methods, libraries and implementation details of this tool.

In the chapter, named ”Evaluation“, testing of this tool is described. There is also a comparison of average calculation times of analyzes.

Chapter 8 is a summary of the achieved results and suggestions for possible extension and improvement of the tool.

Chapter 2

Introduction into biological terms

The first part of this chapter will be describe the structure and function of biological molecules: deoxyribonucleic acid (DNA) and ribonucleic acid (RNA), the difference between them, and also the important concept of sequencing. The next part is more focused on metagenomics and the microbiome, which is important for the human body.

2.1 Deoxyribonucleic acid

Almost all organisms are made up of cells that contain DNA which carries genetic information. Deoxyribonucleic acid includes three basic parts: the deoxyribose molecule, nitrogenous bases, and the remainder of phosphoric acid.

DNA not only encodes the cell itself but also affects the overall evolution and properties of organisms. In prokaryotic organisms, it is located in the cytoplasm. In eukaryotic organisms, it occurs in the cell nucleus.

Deoxyribonucleic acid is a long sequence of nucleotides which are substances from nucleobases. They participate in the storage and transmission of energy. They are also a part of a biological synthesis, regulation of enzymes, protein production, and cellular communication [3].

For DNA are important four basic bases: adenine (A), guanine (G), cytosine (C) , and thymine (T). Adenine and guanine are among the purines - aromatic organic compounds of five carbon atoms and four nitrogen atoms. Cytosine and thymine are pyrimidines - simple aromatic compounds composed of four carbon atoms and two nitrogen atoms.

A certain region of DNA on a chromosome, called a gene, is the basic unit of genetic information that is encoded by the order of the bases. This information is stored, transmitted and by its analysis can be determined from which organism-specific cells come from.

By joining two complementary polynucleotide strands, a DNA molecule is formed. This connection is called a double helix (Figure 2.1). It was discovered in 1953 by James Watson and Francis Crick, who later received the Nobel Prize for it. This double helix is characterized by the formation of hydrogen bonds between the opposite bases. On the one side of this bond are hydrogen and other strongly electro-negative element (for example oxygen or nitrogen). On the other side is an atom with a free electron pair (e.g. nitrogen). Normally, these hydrogen bonds are formed between guanine and cytosine and between adenine and thymine, not other combinations. So the two complementary fibers, that could form a double helix, could look like this: ACCGT and TGGCA [4].

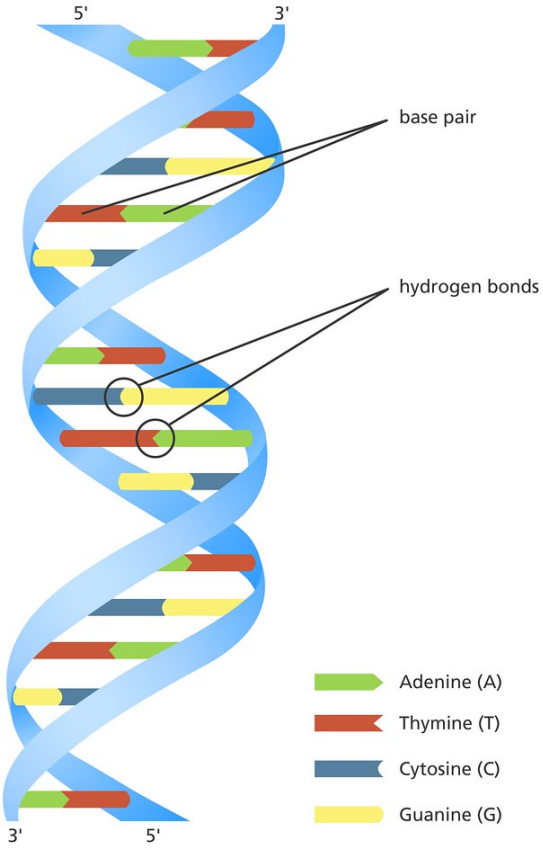

Figure 2.1: The double helix of DNA consists of hydrogen bonds that arise between bases of guanine and cytosine and between adenine and thymine. (Genome Research Limited 2016) [14]

Knowledge of genetic information is used not only in medicine to diagnose diseases. It is also used in agriculture in genetic engineering (genetic engineering[1]), the study of evolutionary development, or forensic science to identify the offender [32].

2.2 Ribonucleic acid

In some organisms, genetic information is carried by RNA. RNA differs from DNA in the way, that it contains ribose instead of the deoxyribose molecule, and thymine (T) is replaced by uracil (U). Thus, pairing occurs between bases guanine-cytosine and adenine - uracil. Instead of a double helix, RNA usually consists of only one strand of ribonucleotides which can be packaged into various three-dimensional shapes. Packing is important from a stability point of view [3].

RNA is responsible for translating the genetic code - passing nucleic acids information from DNA to protein. Proteins are an essential part of all living organisms because they participate in all processes at a cellular level. They form the cell wall, par-

[1] change of genetic information for instance by inserting or deleting a genome

ticipate in the transport of other molecules, accelerate reactions, ensure signal transmission into the cell and also have a defensive function. Proteosynthesis, the protein production, is mostly represented in the cytoplasm. It is a central dogma of molecular biology. (Figure 2.2) It takes place in three steps. The first is DNA replication - the creation of new complementary strands to the original. The second step is transcription - the mRNA molecule is created according to the DNA. This mRNA transport information from the nucleus to the cytoplasm. The last step provides translation of nucleic acid sequence to RNA.

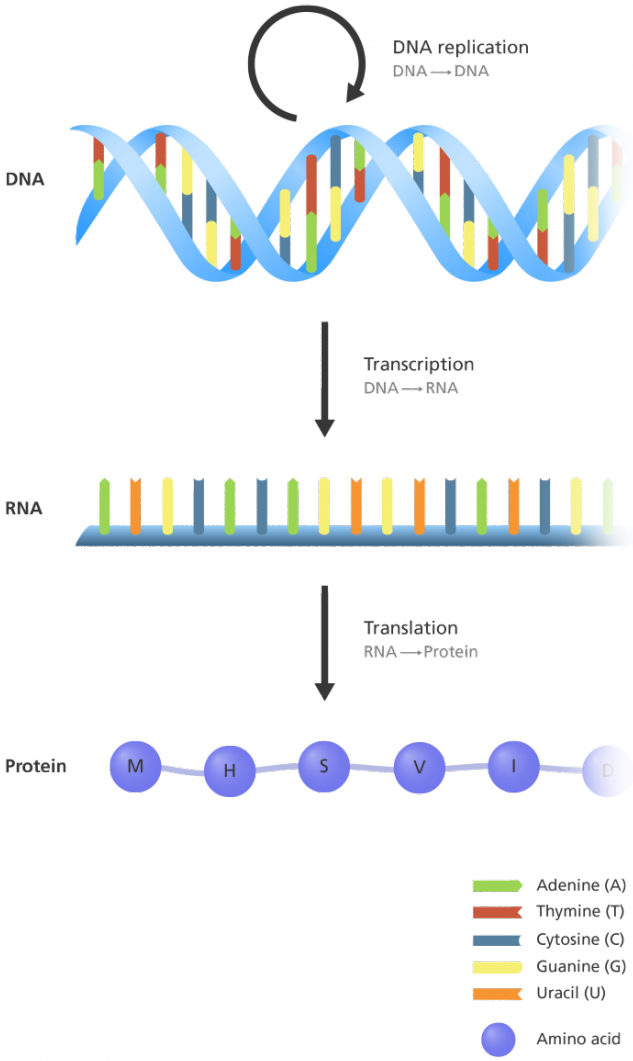

Figure 2.2: Process of proteosynthesis. Firstly, DNA is replicated - the creation of new complementary strands to the original. The second step is transcription - the mRNA molecule is created according to the DNA. This mRNA transport information from the nucleus to the cytoplasm. The last step provides translation of nucleic acid sequence to RNA. (Genome Research Limited 2021) [15]

The best-known types of RNA are mediator, ribosomal, and transfer RNA.

Mediator (messenger) mRNA is produced by transcription of DNA using the enzyme RNA polymerase in the nucleus. Transcription is the creation of a DNA copy - a DNA

double helix becomes an RNA single helix. So, mRNA is a transcript of a part of the genetic information transmitted to the ribosome.

The largest representation in the cell has ribosomal rRNA which is a catalytic component of the ribosome. The ribosome is a really important structure for a translation whose main function is the creation of proteins according to the mRNA template. The mRNA with a transcript of genetic information is attached to it and base on this information the specific amino acids are connected to tRNA.

Transfer tRNA is important for translation, which is part of proteosynthesis. (Figure 2.2) At least one tRNA for each amino acid exists. It selects and carries a specific amino acid and combining it with every three bases from genetic information. Then tRNA aligns it at the correct site in the polypeptide chain on the ribosome and finally, protein is formed [16].

2.3 Sequencing

Sequencing is the conversion of nucleotide sequences into a string that consists of characters from the alphabet ACGT for DNA and ACGU for RNA. The sequencing makes it easier for the scientist to detect mutations or study of evolutionary development of organisms.

One of the best-known sequencing methods is pyrosequencing [45]. For pyrosequencing, the synthesis of a sequence is characteristic and it is widely used in the search for new DNA sequences. It uses the synthesis of a complementary strand by polymerase and at the same time analyzes its activity [6].

However, in genome sequencing, it would be difficult and expensive to check nucleotide by nucleotide because the genome is very large. The human DNA sequence contains circa $3x10^9$ base pairs [44]. Therefore, two methods have been developed that can be used for genome sequencing.

One of them is the Whole-genome shotgun method. It is based on the genome division into random fragments of random size which are further sequenced. Finally, the sequences are most often sorted based on the search for overlap areas [9].

The second one is clone-by-clone sequencing which works on the opposite principle as the shotgun method. The genome is divided into parts and then is created a map of them. These parts are inserted into an artificial plasmid and make their copies. The DNA of these copies gradually breaks down into smaller fragments that are sequenced. After sequencing, these fragments are rejoined and returned to the genome according to the map. This process is complicated with a very large amount of data [30].

The result of sequencing is not only string consists of specific characters and also the quality or quantity of each sequence represented in a specific sample. In the case of targeted sequencing in diagnostics is the result often only information about the presence or absence of known sequences. This information is saved into text files, which are easily processed and can be further analyzed or visualized.

2.4 Metagenomics

In the last ten years, the scientific field of meta-genomics, which uses modern genomic technologies, has begun to develop. This field studies various genetic materials, microorganisms such as bacteria, archaea, fungi, viruses, and the structure of the genome in their natural environment. It helps us to understand how microorganism communities work and how stable they are. The difference between meta-genomics and genomics is that meta-genomics

usually does not use cultured (maintained or propagated microorganisms in the laboratory) samples for its research. It uses sequencing of total DNA optionally RNA from the whole environment. This makes it possible to better analyze a microbial community (the community of all microbes, microorganisms) that cannot be cultivated. DNA optionally RNA isolated from a certain environment at a certain time is called a metagenome. For this reason, meta-genomics has the opportunity to find new sequences, molecules, natural products, or antibiotics.

Metagenomics is classified into two groups: whole meta-genomics and targeted meta-genomics. The whole meta-genomics uses the Shotgun method for sequencing which is depicted in Figure 2.3. Targeted metagenomics is based on a sequenced-based approach and function-based analysis. This involves obtaining a specific genome by PCR (Polymerase Chain Reaction) [13]. The goal of this method is to identify microbes by using a specific gene. In the case of a microbiome, this specific gene is most often a 16s rRNA. Based on 16s rRNA, individual species of bacteria are determined [29].

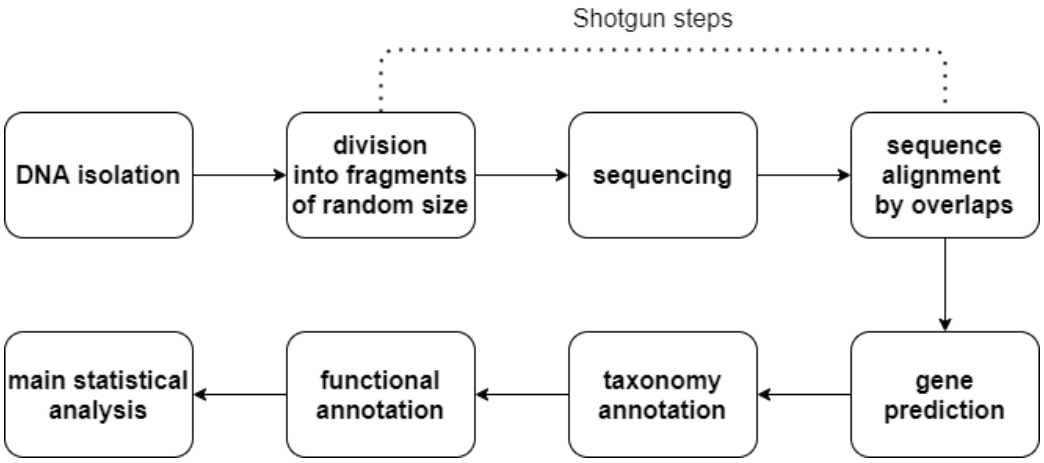

Figure 2.3: One of the methods, which use meta-genomics for sequencing is the shotgun method which works with random fragments from different parts of the genome.

2.5 Human Microbiota

The microbiota is very important for human life, but relatively unexplored too. It is very difficult to predict its evolution and behavior. Microbiota is found everywhere, affects the function of the whole body, and is influenced by everything - the environment in which it is, diet and exercise, and in small children even the way they are born [50].

The human microbiota is a set of all microbes, respectively microorganisms, living not only inside, but also on the surface of the body. Its amount increases significantly only until about the third year. On the other hand, its composition, the representation of groups, changes throughout the whole life. Because of the huge amount of it in the human body, it is often referred to as another organ. One of its main functions is to train human immunity and it is a really important part of human metabolism. It can also affect a person's mood or emotions. Its imbalance, dysbiosis (the predominance of hostile bacteria) often causes digestive problems, chronic inflammation, and even mental illness. Many analyses showed that the microbiota of a healthy person and a person with diseases is significantly different

in many cases. However, the question remains, whether this difference is the cause or even the consequence of these diseases [50].

Every human microbiota is specific and original. Even the gut microbiota of the same individual differs in different parts of the gut. For this reason, in the future, it could also be used as an identifying element in forensics [50].

Metagenomic methods are used for its initial extraction of genetic information. Only thanks to these methods we can analyze the microbiome because the majority of bacteria are not cultivated. The result can be further investigated in several ways. One of them is DNA analysis, where the sequences are compared with known ones that are already in the database and a match is wanted between them. Hybridization is used to determine if a specific template matches the sequence being searched.

The analysis can be qualitative or quantitative. In a qualitative analysis, the device returns information about how confident it is that the sequence is actually in the sample. In quantitative analysis, it returns the number of times a given sequence is in a sample. The resulting microbiome data can contain two types of data. The first type is the abundance of sequence variants in the samples and the second type is metadata, which are the properties of the analyzed samples (for example, the place or method of sampling). Subsequently, the data containing an abundance of sequence variants in the samples can be visualized, linked to metadata, and evaluated [37].

2.6 16S rRNA

Sequence analysis using 16S rRNA is relatively exact and widespread. The main reason is that the 16S rRNA gene is relatively large and is found in every cell in prokaryotes and archaea organisms. In eukaryotes, the equivalent is 18S rRNA. 16S rRNA has been used for phylogenetic studies mainly because its region is conserved and mutates relatively slowly. For analysis, the entire 16S rRNA gene is usually not used but only some specific region is selected. It is due to the limitations of the most commonly used sequencing technology - Illumina [46].

Variable regions are parts of a gene that are used primarily to distinguish related species. They provide specific information for the identification of bacteria at the gene level. We distinguish two types of regions. The first type contains regions that do not change or change only very slowly. The second group consists of regions that have undergone significant changes [3]. Between the individual regions, there are highly conserved areas that are used for phylogenetic analysis and assumption of the taxonomic level.

16S rRNA analysis is used to identify and compare individual sequences in a given sample. The first step is the sequencing of the fragment and its subsequent comparison with the specific region. Based on this step, the sequences are grouped into individual groups and form clusters of similar sequences. These groups are identified and the number of representatives of these groups is counted. A table that contains the 16S rRNA variants and their abundance in individual samples is created and saved to a text file which is ready for further analysis [48].

Advantages of using 16S rRNA include versatility, found in almost all bacteria, the function does not change over time, sufficient size, cost of analysis, no problem with horizontal gene transfer, conserved structure, and sequence [22].

The main disadvantage of using 16s rRNA is that when it is applied to slightly different species, they provide only limited information [22].

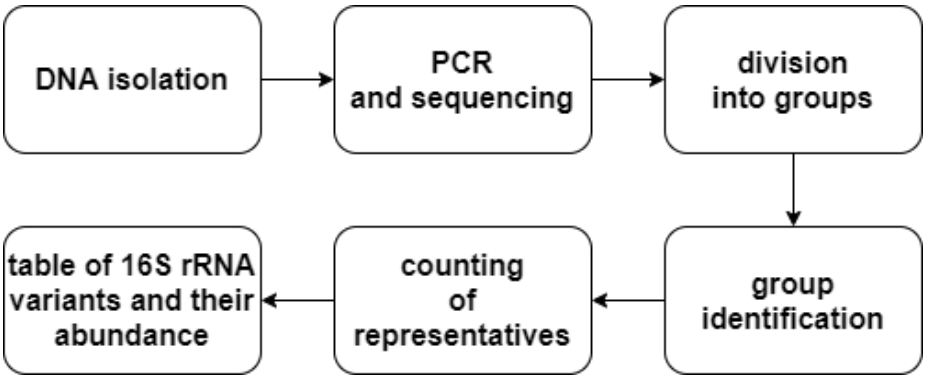

Figure 2.4: Identification and comparison of individual sequences in a given sample with 16S rRNA.

Chapter 3

Methods used for microbiome analysis

The first part of this chapter is dedicated to statistical methods that are used in the designed tool. There is a more detailed description of the PCA and PCoA methods and also the differences between them. In the second part, the concept of the phylogenetic tree and the methods which are used for its creation is explained. The last part is the explanation of what beta diversity is and the metrics for its calculation.

3.1 Principal Component Analysis

Principal Component Analysis (PCA) is a method used for reducing dimensions, so it also allows to visualize multidimensional data. Its goal is to reduce data size while trying to preserve as much information from the initial variables as possible. The price for this simplification is a partial loss of accuracy [42].

Before the application of the method, it is very important to standardize the data - to eliminate deviations of the initial variables. This step prevents the distortion of the result and performs the transformation of the variables to an equal scale. Standardization is described by a formula

$$SV = \frac{IV - AV}{SD} \tag{3.1}$$

where:

SV - standardized value,

IV - initial value,

AV - average value,

SD - standard deviation.

Subsequently, it is necessary to find out whether there are any relationships (correlations) between the individual values. A covariance matrix is used for this. It is a symmetric matrix in which elements are all possible pairs of variables. Positively calculated values indicate data correlation, negative indicates the opposite.

The next step is the identification of eigenvectors - principal components (PCs) and the calculation of eigenvalues. Principal components are linear combinations of the original variables. Each PC is characterized by a degree of variability (scattering) and is perpendicular to the previous one. Eigenvector is a nonzero vector which direction does not change when the linear transformation is applied to it. The eigenvalue is

a coefficient that indicates the number of variances carried in each principal component. Percentage of variance (Figure 3.1) can be calculated by a formula

$$PV = \frac{E}{SE} \tag{3.2}$$

where:

PV - percentage of variance,

E - eigenvalues,

SE - sum of all eigenvalues.

As a result, the eigenvectors are ranked from the largest eigenvalue to the lowest. This means that the first PC1 has the largest percentage of variance - it contains the most information. The second contains the largest scatter that was not contained in PC1 and is also perpendicular to PC1. The third has the largest percentage variance that was not contained in PC1 and PC2. PC3 is also perpendicular to PC1 and PC2. The rest of the PCs follow the same pattern. [42]

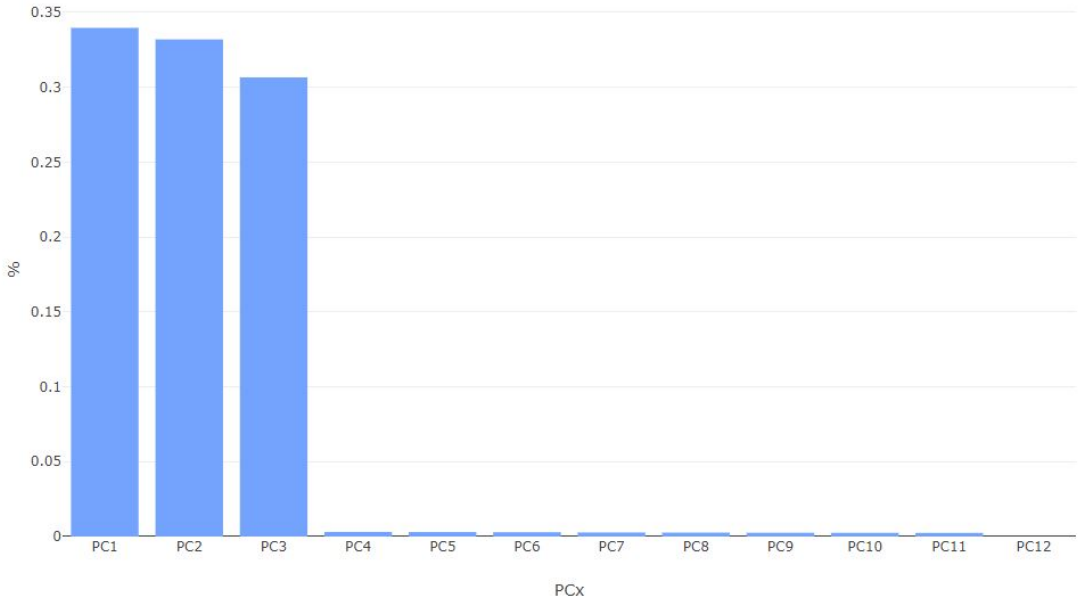

Figure 3.1: Example of a percentage of the variance of principal components. According to this plot, the most important principal components were PC1, PC2, and PC3.

In other words, principal components are just new axes that provide a better overview of original data. The main advantage of using principal components is the smaller data size compared to the initial ones because components that contain little information can be ignored. The resulting data are also no longer correlated and at the same time, they capture almost the entire variability of the original variables.

Finally, it is necessary to project the initial data on the new axis using feature vectors, which is an n-dimensional vector of features that represents an object. In this case, it is a matrix of vectors, which is made up of components that contain a sufficient amount of information [42].

3.2 Principal Coordinates Analysis

Multidimensional scaling (MDS) [5] is a method used for reducing dimensions and makes it easier to visualize the distance between objects. The input is a matrix of relationships between individual objects. It can be a similarity matrix - larger numbers mean greater similarity between individual objects. If it is a matrix of differences, larger numbers mean less similarity. The input matrix is then converted to a distance matrix - distances between individual objects are calculated. The method tries to decompose all objects into the resulting space in a way that distances from the distance matrix are preserved as much as possible. (Figure 3.2) Similar objects are placed next to each other, different apart from each other. One of the multidimensional scaling methods is classical MDS, otherwise known as Principal Coordinates Analysis (PCoA) or Torgeson Scaling.

DISTANCE MATRIX					
	A	B	C	D	E
A	0	210	430	1500	960
B	210	0	230	1310	800
C	430	230	0	1080	670
D	1500	1310	1080	0	1330
E	960	800	670	1330	0

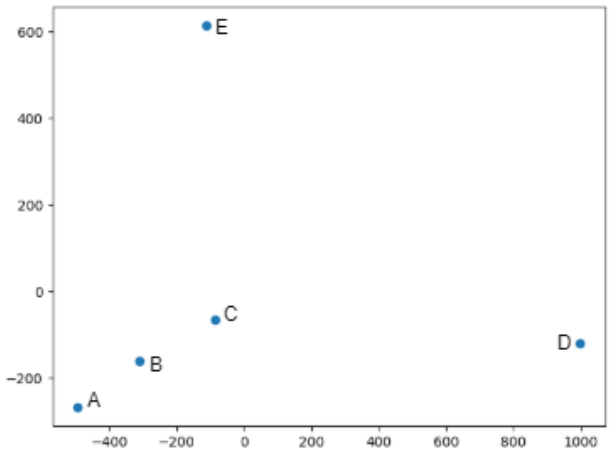

Figure 3.2: Example of the result of PCoA. Each sample from the distance matrix is recalculated by PCoA. This ensures the lower dimensionality of the data, which can be visualized for example into 2D or 3D graphs.

PCoA is based on a similar principle as PCA. The difference is that at the beginning the input matrix of relations is converted to a distance matrix, which is composed of distances that are calculated by any distance metric. PCoA compares this matrix with the initial data using the evaluation stress function. The stress function measures the degree of correspondence between the original data and the distance matrix. The best stress value (suitability indicator) is zero [5] and it can be calculated by a formula

$$stress = \sqrt{\frac{\sum\sum(f(x_{ij}) - d_{ij})^2}{scale}} \tag{3.3}$$

where:

d_{ij} - Euclidian distance,

$f(x_{ij})$ - function of original data,

scale - constant scaling factor.

Then, the coordinates of each object are adjusted to achieve minimized stress. These steps are repeated until the stress value begins to decrease. PCoA returns a matrix of coordinates.

Using this method may not always be appropriate. It may happen that even if the best configuration is found for a given number of dimensions, the data is relatively skewed. It

is due to the high-stress value [5]. In this case, it is recommended to replace the method with another or increase the number of required dimensions. However, increasing the number of dimensions no longer guarantees data simplification.

PCoA is slower than PCA and the axes formed by MDS cannot be interpreted in terms of variability. The advantage is that it allows the omission of information in the input data [5].

3.3 Phylogenetic tree

The phylogenetic tree captures relationships, the similarity between individual taxonomic units based on morphological or genetic similarity, provided that they have some common ancestor. 16S rRNA sequences are often used to create it.

This tree is based on the same tree structure as is know from computer science or mathematics. It is a non-oriented graph formed by root, nodes, branches, and leaves. Root represents the evolutionary oldest common ancestor. It is distinguished between rooted and non-rooted trees, depending on whether the root is in the tree or not. Nodes represent hypothetically the last common ancestors. Their DNA is normally not available. Each branch in the tree is one evolutionary line. Edges are the relationships between individual nodes. Their length can express the period of evolution from one organism to another or the number of mutations that would be required for a given sequence to move to another. Leaves are unique sequences, taxonomic units, existing organisms that are examined [32].

There are several methods for calculating the phylogenetic tree [25] [23] [8]. However, for all methods, the input sequences must first be aligned. This task solves Multiple sequence alignment [10] and is one of the NP-completely problems. (Figure 3.3)

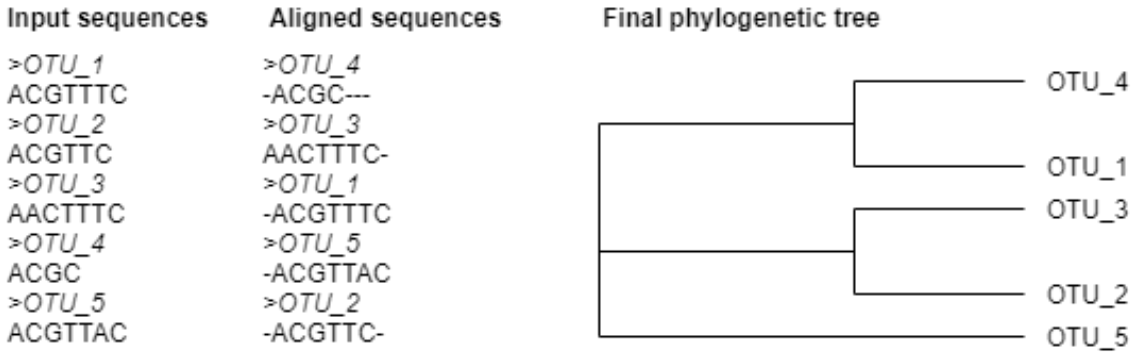

Figure 3.3: Example of the phylogenetic tree created from input sequences. At first, these sequences are aligned and then based on their similarity the tree is constructed.

Distance methods

In the beginning, the distances between the individual sequences are calculated and written in the distance matrix. The pair of sequences that are the closest to each other is then selected from the matrix and joined into a single cluster. This cluster is also marked in the emerging tree, where at the beginning are only individual sequences as leaves.

The distances are recalculated again, but with the difference that the sequences with the shortest distance is replaced by the resulting cluster. This process is repeated until all pairs are joined, only one cluster remains and a complete tree is created.

Parsimony methods

This method uses multiple sequence alignment as input and it aims to achieve the lowest possible parsimony score. The parsimony score expresses the sum of the ratings of all the branches in the tree. In the beginning, a scoring matrix is created, which expresses the ratings of the branches between the individual characters of the input alphabet. Based on this matrix, the score of individual nodes is calculated.

Finding a tree with this method can be used in two ways. The first is The Small Parsimony Problem that already has a tree and the goal is to find the best evaluation of internal tree nodes. The big parsimony problem tries to not only find the best parsimony score but it is also looking for a specific tree shape. For this step, it would be necessary to create all possible phylogenetic trees, which would be very demanding and therefore heuristic methods are used for it.

Heuristic methods

If the number of all trees is too large, heuristic methods are used. Initially, a random tree is selected and the number of evolutionary changes is counted in it. Subsequently, some two branches are swapped and the number of evolutionary changes is recalculated. From these two trees, the one with fewer changes is selected and some branches are swapped again. The random selection is repeated several times. The resulting tree is the one in which there have been the least changes.

Maximum likelihood

In this statistical method, the probability that the input data were generated by a given model (phylogenetic tree shape) is calculated. In other words, the algorithm tries to find a tree that is the closest to the input data. The algorithm returns the tree that has the highest probability. If the input data is noisy, the method can be relatively unreliable.

Bayesian methods

They are statistical methods, which calculate the probability that a given tree was generated from input data. The algorithm also returns the tree that has the highest probability. If the input data is noisy, these methods are more reliable than maximum likelihood methods, because they use prior information on how the phylogenetic tree should generally look like.

Bootstrapping

In this method are repeatedly creating partial phylogenetic trees on samples of the input data and the average of all partial trees is the final tree.

3.4 Beta diversity

Biodiversity expresses the variability of all living organisms. There are three biodiversity types. The first is alpha diversity, which works with the number of species in the sample. Another is beta diversity, which focuses on the difference in species composition between samples. The last is gamma diversity, which takes into account the number of species in the wider locality.

In this thesis, beta diversity is used to calculate the distance matrix for PCoA. Beta diversity answers the question of whether sample A is more different in composition than sample B or than sample C. Several methods are used to calculate diversity.

Bray–Curtis dissimilarity

It is a non-phylogenetic method - the phylogenetic tree is not required for its calculation. The result of the Bray-Curtis dissimilarity calculation (BC_{jk}) is a value between zero and one. If this value is equal to zero, it is the same composition, all species are shared. If the result is equal to one, the patterns have nothing in common - they do not share any species [38].

The calculation is performed using a formula [38]

$$BC_{jk} = \frac{\sum_i |X_{ij} - X_{ik}|}{\sum_i (X_{ij} + X_{ik})} \tag{3.4}$$

where:

X_{ij} - frequency of OTU i in sample j,

X_{ik} - frequency of OTU i in sample k.

INPUT DATA				
	OTU1	OTU2	OTU3	OTU4
A	2	1	0	1
B	0	3	0	4
C	0	2	6	2

Bray Curtis →

RESULT			
	A	B	C
A	0	0.63636	0.71429
B	0.63636	0	0.52941
C	0.71429	0.52941	0

Figure 3.4: Two samples are selected and according to the frequency of OTUs in it, Bray-Curtis dissimilarity is calculated. There is an example of a result.

The example of the result of using this method is shown in Figure 3.4.

In the case, that not all OTUs (Operational taxonomic units) are the same size in all samples, the input data must be adjusted. [38].

UniFrac

UniFrac is one of the most used phylogeny based distance metrics - a phylogenetic tree is needed for its calculation. It is another method that is used for the expression of the difference.

Unweighted UniFrac considers only whether the OTU is in the sample or not. It takes only the proportion of branch lengths that are presented in only one sample and branch lengths that are presented in at least one sample. In Figure 3.6 is an example of using unweighted UniFrac. The formula for computation is

$$unweightedUniFrac = \frac{U}{S} \tag{3.5}$$

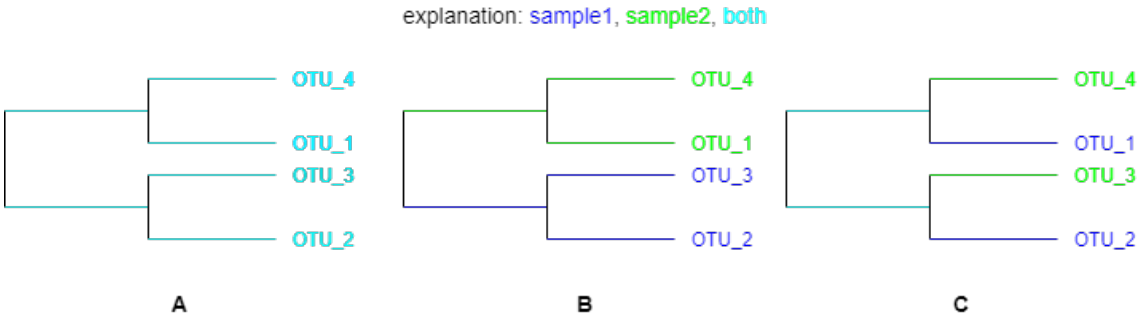

Figure 3.5: A used phylogenetic tree may contain branches that belong to only one sample, both samples, or none of them. If all branches belong to both samples, it is spoken about a UniFrac value of null (Figure A). On the other hand, if one whole branch consists of only one sample and the other consists of the only second sample, the UniFrac value is one (Figure B). If half of the branch belongs to one sample and the other half belongs to the other, the UniFrac value fluctuates around 0.5 (Figure C) [26].

where:

U - branch lengths which are presented in only one sample,

S - branch lengths which are presented in at least one sample.

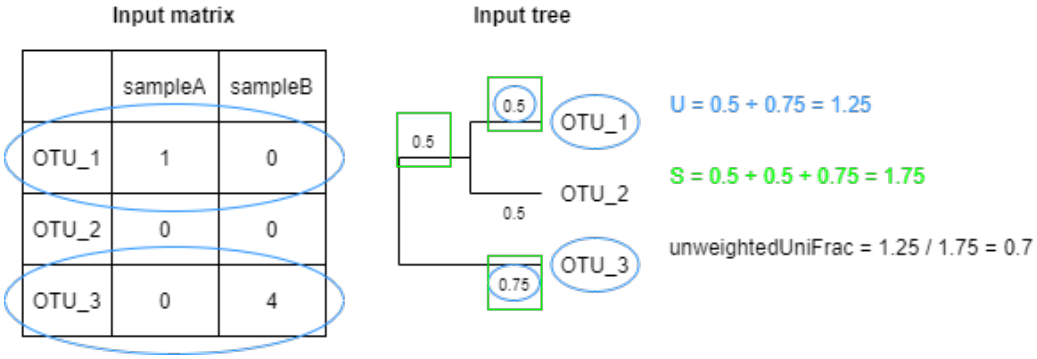

Figure 3.6: Computation of unweighted UniFrac with input matrix and phylogenetic tree. Branches, which are presented only in one sample are OTU1 (only in sampleA) and OTU3 (only in sampleB). The green color represents all branches presented in at least one sample.

Weighted UniFrac considers not only whether a given OTU is present in the sample, but also its frequency. The formula [27] for computation is

$$weightedUniFrac = \sum b_i \left| \frac{A_i}{A_T} - \frac{B_i}{B_T} \right| \tag{3.6}$$

where:

b_i - length of branch i,

A_i, B_i - numbers of sequences that descend form branch i in A and B,

A_T, B_T - total numbers of sequences in communities A and B.

Chapter 4

Existing tools for visualization of microbiome data

In this chapter, the most common existing tools that allow looking at microbiome data are described. The first part is dedicated to Emperor, which plots results from PCoA. The second part described TensorFlow Embedding Projector which works for example withk PCA, T-SNE, or UMAP.

4.1 Emperor

Emperor [47] is a tool designed especially for the visualization of microbial data. This tool primary uses principal coordinates analysis (PCoA), although it does not calculate it. The PCoA result is expected just as input for visualization and it can be calculated for example by Qiime [7] (Figure 4.1). One of the disadvantages of this tool is that it can not work also with PCA.

```
pc vector number    1       2       3       4       5
PC.636      0.333514620475  0.081687        0.25081 0.103746        2.50106743521e-09
PC.635      0.258145479886  -0.22437        -0.25446        -0.0290044      2.50106743521e-09
PC.356      -0.270663791669 -0.23983        0.18965 -0.118931       2.50106743521e-09
PC.481      -0.0437436047686        0.30751 -0.077078       -0.20627        2.50106743521e-09
PC.354      -0.277252703922 0.074945        -0.10898        0.245681        2.50106743521e-09


eigvals     0.329912543767  0.2147172       0.1815366       0.1261003       -3.127915774e-17
% variation explained       38.68853        25.18276        21.2717 14.85772        3.667478e-15
```

Figure 4.1: Example of input dataset for visualization by Emperor [47]. Input dataset must already be a PCoA result, this tool just visualizes it, it can not compute it. The input file consists of two parts. The first part is the matrix of projected coordinates of original input data, the second is a list of eigenvalues and a line with the percentages of variation explained at each dimension.

Even though this tool generates HTML documents with visualization, it is only a software tool. This means that if the user wants to use it, he has to install it and all its dependencies, especially QIIME, locally.

Emperor is used for visualization of any QIIME [7] or scikit-bio [40] compliant datasets. This means that its input can be file in the format fasta, blast, newick, and similar biological format, but the processing of common excel tables is not allowed (file should always be saved as tab-delimited text) and the JSON file is recommended only for smaller datasets.

For visualization, Emperor can use two metadata categories. The gradient category determines the order in which samples are connected together. The trajectory category determines how samples are grouped. If plots are generated with this tool, only columns where all values are numeric will be accessible as a trajectory category,

The result of visualization is plotted to the canvas, where is also displayed the number of observed samples and the possibility to change the type of visualization (PCoA or Parallel). On the right side of this canvas are controllers for changing for instance opacity, sphere scaling, metadata for coloring, or dimensions. The tool can show sample identifiers and a list of categories from the mapping file for coloring. Each component in the graph can be hidden or visible. This tool offers visualization only in 3D space and visualization in 2D space is missing.

Emperor also allows metadata-based coloring, which is important for clear visualization. In this way, for example, vegetarians can be color-coded from carnivores or adults from children. However, this coloring is based only on the selection of one metadata category, different categories for coloring cannot be combined. It means that data can be colored by type or method of isolation, but not by both of these criteria at the same time.

4.2 TensorFlow Embedding Projector

Unlike Emperor, TensorFlow Embedding Projector [43] is a web-based tool used for general multidimensional data visualization. For preparation data for visualization, this tool can use one of three techniques - UMAP, T-SNE, and PCA. For this thesis, the most important is PCA. This tool does not deal with PCoA, so it does not allow the calculation of beta diversity.

Both input vectors and metadata for PCA must be in tsv format and tab-separated. (Figure 4.2) Additionally, single-column metadata or input vectors can not have a header row. If the input format is not in a proper format, the tool is relatively unstable and not very user-friendly - it usually crashes.

The result of PCA analysis can be plotted in 2D or 3D space with this tool. The canvas header contains options for working with a visualized graph, as well as the number of plotted samples and the number of available dimensions. The result of the PCA analysis and other details can be saved by the user in txt format. TensorFlow Embedding Projector allows the publishing of embedding visualization and also input data.

Like Emperor, this tool allows metadata-based coloring based only on the selection of one metadata category, different categories for coloring cannot be combined. It again means that data can be colored by one metadata category, but not by more categories of these metadata at the same time.

The percentage of variance is expressed only in numbers in the left-down corner, where is a selection of the principal components, which should be used to plot the graph. This tool does not offer any graphical representation of this information.

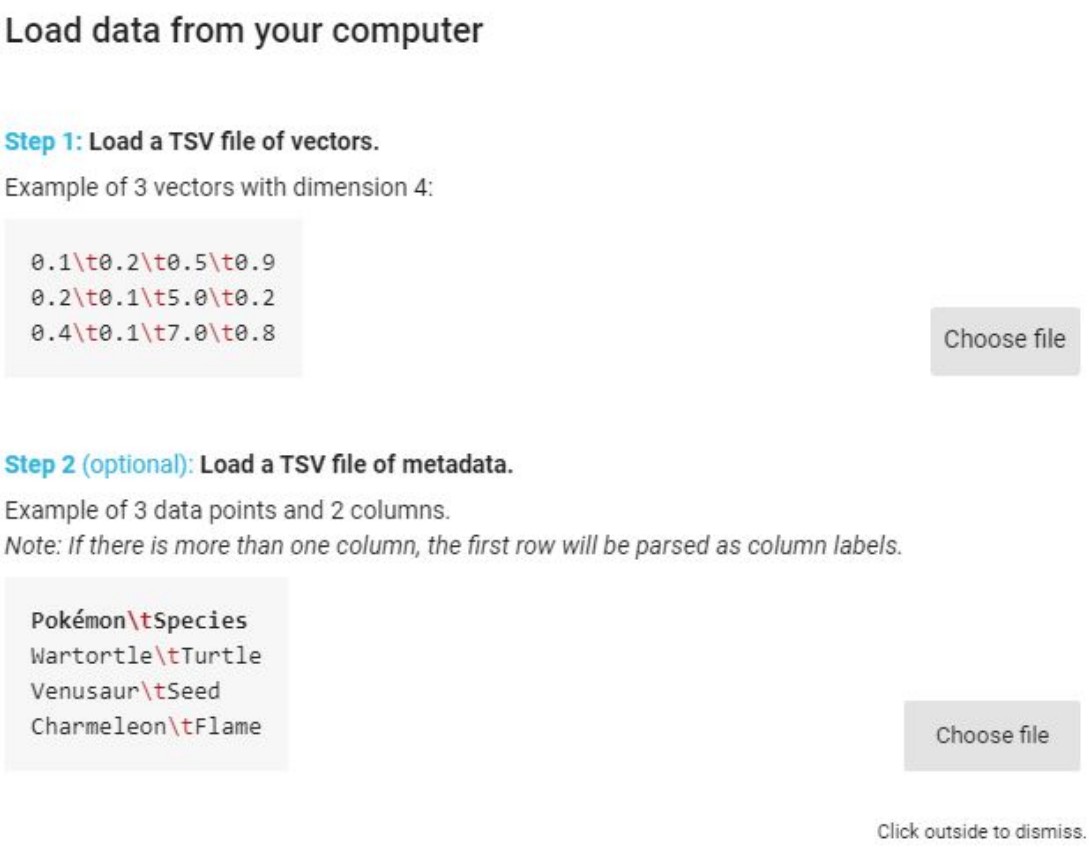

Figure 4.2: Example of input data for visualization by TensorFlow Embedding Projector. Input dataset must be in a proper tsv format separated by a tab. (TensorFlow Embedding Projector 2021) [43]

Chapter 5

Analysis of requirements and tool design

The aim of this work is to design and create a tool for the visualization of microbiome data, which would minimize the weaknesses of existing tools and meet the main requirements of the target group of microbial researchers, who should use them.

In this chapter, the main aim of this work is described more detail. The next part of the chapter is dedicated to tool design and its functions. In the last part, a description of input and output files is mentioned.

5.1 Functionality requirements

The main goal of the proposed tool is to link the methods that use the already existing tools, which are mentioned in the Chapter 4. Although the emperor allows the visualization of the PCoA result, it cannot calculate the analysis itself. Contrariwise, the possibility of PCA calculation and visualization is completely missing. On the other hand, the TensorFlow Embedding Projector can not only visualize the PCA, but it can also calculate it. The disadvantage of this tool is that the possibility of calculation and visualization of the PCoA is missing. TensorFlow Embedding Projector has the advantage over Emperor in that it can render graphs in both 2D and 3D space (Emperor can only render them in 3D) and is fully web-based - Emperor is a software tool that just generates HTML documents with the final visualization. On the other hand, Emperor allows a larger number of input file types, TensorFlow only accepts the very limited input format described in the subchapter 4.2. Both of these tools allow to color data by only one metadata category; different categories cannot be combined in non of them. There is also an absence of a graphical interpretation of the percentage of variance in both tools. A comparison of these tools can also be found in the Figure 5.1 The new proposed tool should eliminate the weaknesses of these tools and get as close as possible to the needs of microbial research.

In the new tool, the user will be able to choose whether he wants to use principal component analysis or principal coordinates analysis to reduce the dimensionality of the data. These methods were explained in more detail in the Chapter 3. Because PCoA also uses a distance matrix for its calculation, the user is given a choice of several metrics to calculate it. In the case of calculating the distance matrix, there is also the option of downloading it in several formats. The same download option applies to a file that contains any number of first principal components that have been calculated using dimen-

	Emperor	TensorFlow Embedding Projector	proposed tool
PCA calculation	✗	✓	✓
PCA visualization	✗	✓	✓
PCoA calculation	✗	✗	✓
PCoA visualization	✓	✗	✓
dimensionality	3D	2D, 3D	2D, 3D
web / SW tool	SW, result in HTML document	web	web
format of input files	QIIME or scikit-bio compliant datasets	tsv and tab-separated format	Biom-format, Excel, JSON, text files
data coloring according to one category	✓	✓	✓
data coloring according to a combination of categories	✗	✗	✓
graphical interpretation of the percentage of variance	✗	✗	✓

Figure 5.1: Comparison of the tools. The proposed tool is designed base on advantages and disadvantages of existing tools and try to get as close as possible to the needs of microbial research.

sionality reduction methods. The representation of the individual principal components is also reflected in the bar graph. The result of reducing the dimensionality of the data should be projected into both 2D and 3D space. For better clarity of this result, the user can choose any combination of metadata according to which the result should be analyzed. The graphs that will be the output of the analysis should allow basic work with them, such as zoom in, zoom out, rotating, scrolling, or downloading them.

5.2 Basic tool design

The tool was originally designed as a simple desktop application in Python without any database that the user will be able to install and use locally. The user interface, which is shown in Figure 5.2, consisted of a canvas with a graph that represented the result of the analysis and a form in which the user entered requests for analysis. Although the tool designed in this way met the basic requirements for functionality, it was not so easily accessible to the user. The solution was to modify the desktop application and create a web tool.

The new design of the web tool consists of three parts. The first part is the user interface, where the user enters the required parameters for the calculation into the form. The result of the analysis is also plotted to the user interface. The second is the calculation part, where the analysis and work with requests take place. The last part is the database in which the calculated data are stored for later visualization or for downloading files with transformed data or distance matrix.

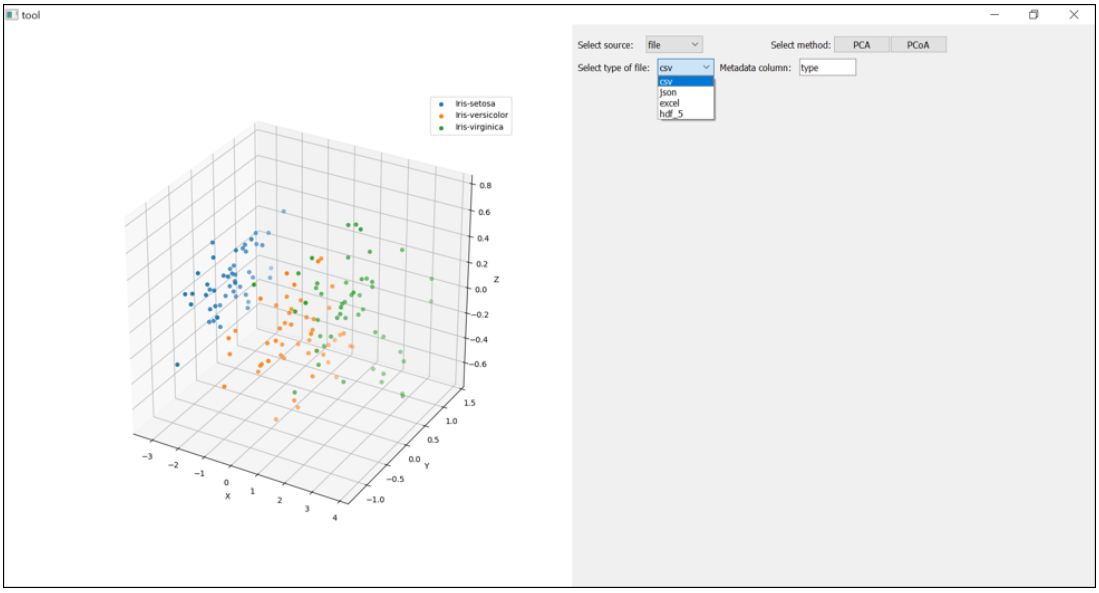

Figure 5.2: The tool was originally designed as a desktop application in Python, with which the user can work locally on their computer.

5.3 Individual parts and their communication

The tool is based on sending various requests from the User Interface (UI) to the Application Programming Interface (API). At the API level, based on the type of request, it is decided which event will be executed. If it is necessary, the data is stored or retrieved from the database using the API. After serving the API request, the response is sent back to the user interface. Here the answer is processed and the result is interpreted to the user. This communication is shown in Figure 5.3.

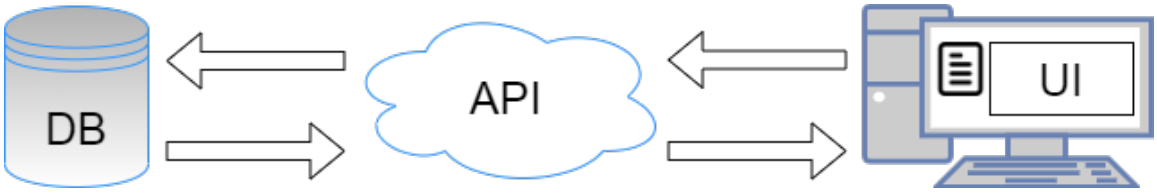

Figure 5.3: The tool consists of three parts that communicate with each other by sending requests.

5.3.1 User interface

The user interface includes forms and a canvas into which the resulting graph is drawn.

In the forms, the design of which can be found in Figure 5.4, the user enters the parameters that are needed for the calculation. In the case of a PCA form, it includes information about the dimensionality of the space into which the result should be presented - the choice is 2D or 3D. Input data in the form of files are also necessary for the calculation. The first part of input data is for the type of file and the file, which contains the data from which the analysis will be calculated. The second part contains the metadata file and also its

type. In the case of working with a biom-format file, the user enters only one file, which contains all data for analysis and the necessary metadata too. The format of these files is described in more detail in section 5.4. After loading the metadata file, the user will see a list of categories from which he can choose the ones according to which he wants to color the output. At the bottom is a button for confirming and sending data to the API. In the case of the PCoA form, only the option of selecting the method of calculating the distance matrix is added.

After the graph is drawn, new sections are added to the forms. In the case of PCA, there is the option to enter the number of first X principal components and the type of file in which these transformed data should be saved. In the case of PCoA, the option of selecting the type of file in which the calculated distance matrix should be stored is also added.

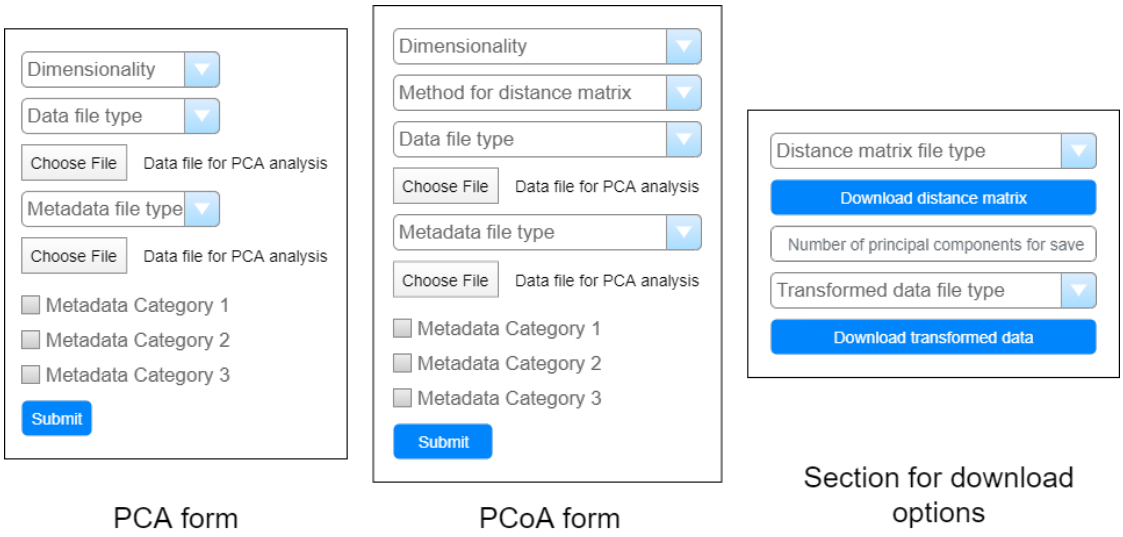

Figure 5.4: Design of forms for the user in the user interface. Data from these forms is sent to the API, where it is further processed. The content of the section for download options depends on the type of analysis. In the case of PCA, there is only the section for download transformed data. In the case of PCoA, there is also the section for download distance matrix.

The result of the analysis in the form of graphs is visualized on the canvas. The result of the Percentage of Variance of maximal first fifty principal components is visualized in a 2D bar graph. A maximum of fifty was chosen to keep the graph concise.

The result of the dimension reduction itself is plotted in a scatter 2D or 3D graph according to the user's choice. In this graph, for each plotted point, there is the name of the sample and the name of the group to which it belongs according to the selected metadata. The chart also includes a legend that can be hidden or shown at any time.

5.3.2 Application Programming Interface and database

At the API level, what should be done is selected based on the data received from the user interface. It can be a request that will be served immediately, but it can also be a more complex type of task that takes a long time to calculate. More complex tasks, in this case, include those in which the calculation of a phylogenetic tree is necessary

(for example, the calculation of the distance matrix using UniFrac). These jobs are queued and waiting to be serviced.

The database stores information about individual requests that may still be needed for further requests. One record in the database can contain a unique ID, a computed matrix, transformed data, or an overall analysis result that is needed for visualization on the user interface side. The proposal of the database is shown in the Figure 5.5.

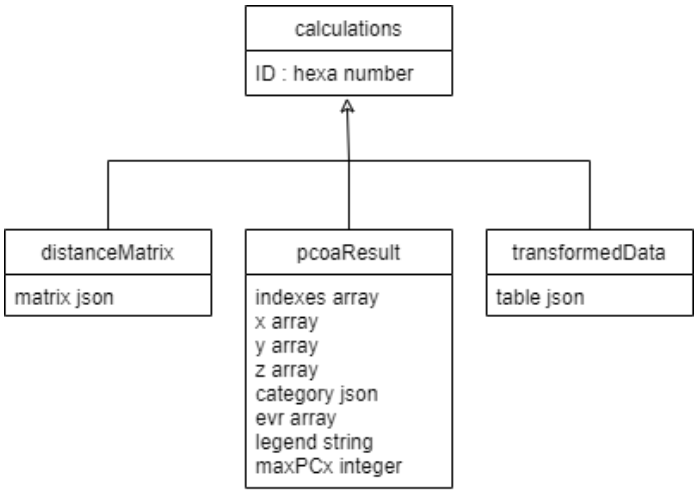

Figure 5.5: To the database by ID, results of PCoA, distance matrices, or table of transformed data can be saved.

5.3.3 User interface - Application Programming Interface communication

The UI sends an API request to the user based on filling the form. Each request has its specific format and expects a specific response. A table of requests and responses to them can be found in Figure 5.6.

When a PCA calculation or a PCoA calculation using the Bray-Curtis for distance matrix calculation is requested, UI sends the parameters needed for these calculations. The user interface expects a response with the data needed for plotting the graph and a unique ID with which can be sent requests related o this session. In the case of PCoA with matrix calculation using one of the UniFrac methods, the UI sends the user's e-mail together with the necessary parameters for analysis. Instead of the result, only the ID under which the calculation takes place is sent in response. It is for the reason that the calculation can take a long time and wait for the result can be uncomfortable for the user. In this case, after the calculation is completed, the user is notified by e-mail, in which he finds a link to the output in the form of a graph.

The file save request contains only the file that the user wants to upload. The answer is the path where the file is saved at the level of API.

In the case of a PCoA UniFrac request, the user interface receives in response the session ID under which the calculation is registered. If the user chooses to wait despite a longer calculation and does not close the web page, the UI uses this ID to create a new request every 5000 ms, asking if the calculation is already over. If the calculation has already finished, the API returns a response with the data necessary for a plot of the graph and the UI

TYPE OF REQUEST	PARAMETERS OF REQUEST	RESPONSE
save file	file	path to file saved on API side
PCA calculation	data needed for analysis	data needed for plotting the graph and a unique ID
PCoA calculation using the Bray-Curtis	data needed for analysis	data needed for plotting the graph and a unique ID
PCoA calculation using the UniFrac	data needed for analysis and user's e-mail	a unique ID
is PCoA calculation using the UniFrac completed?	ID of calculation	null in case of uncompleted calculation or data needed for plotting
get result of analysis by ID from e-mail	ID of calculation	data needed for plotting the graph
save transformed data	number of principal components, type of file for saving	file with transformed data in the proper format
save distance matrix	type of file for saving	file with distance matrix in the proper format

Figure 5.6: Each request send from UI to API contains specific information. According to the type of request, the response is created on API side and send back to the user interface.

stops querying for the result. If the calculation has not yet been completed, the user interface continues to query.

If the user does not want to wait for the PCoA result using UniFrac and closes the page, after the calculation he will receive a notification e-mail with a link where he will find the graphical output of his analysis. After opening the link, the UI sends a request for data results with the given session ID to get important data for visualization. The required ID is a part of the line found in the e-mail.

Another type of request is the storage of transformed data. In this case, the UI sends a request with the ID that belongs to the given session, the number of the first X principal components which transformed data should be downloaded, and the type of file in which this data will be stored. The response to this request is a file of the requested type.

When requesting a file with a distance matrix, the request again contains the ID of the current session and the type of file in which the matrix is to be stored.

Examples of individual communications can be found in Figure 5.7.

5.3.4 Application Programming Interface - Database communication

In the case of PCA, only the transformed data and the unique ID of the specific request are stored in the database at the API level. In the case of PCoA, with the calculation of the distance matrix using the Bray-Curtis method, the calculated distance matrix is stored in the database together with the request ID and transformation data. In the case of PCoA analysis, with the calculation of the distance matrix using one of the UniFrac methods, the request ID, transformation data, the distance matrix, and also all the information that will be necessary for the visual representation of the result are stored in the database.

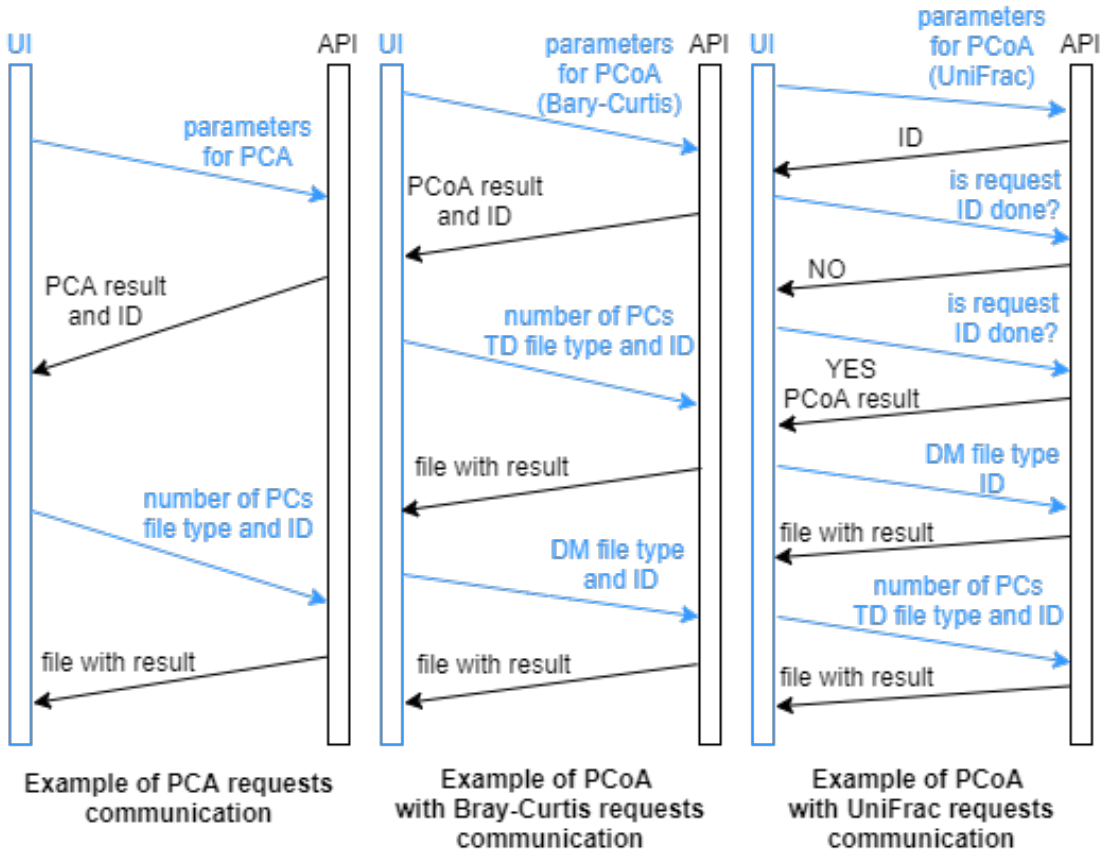

Figure 5.7: Examples of communication between the user interface and API for different types of requests. The types of requests and the responses to them depend on the type of used analysis.

When an API request is delivered with a specific ID to store a file with transformed data, matrix, or return a PCoA result using UniFrac, a query is sent to the database with that ID. The database returns all the information which are stored for this ID and this information can be further processed.

Examples of API and database communication are shown in Figure 5.10.

5.4 File formats

In Subsection 5.3.1 was already mentioned that for the calculation of PCA or PCoA with subsequent visualization of the result, two types of data are needed - data for the calculation of principal components and metadata, according to which the analysis result will be colored. This data must be provided to the tool in one of the supported file formats described in this chapter, which are text files, JSON, Excel, or biom-format files. A small description of the file format will be also displayed in tooltips in input forms.

In the case of a metadata file, it is also necessary for the first column to contain the names of each sample and for the header to include the names of the individual metadata categories. In the case of a file for the calculation of the principal components, the first column must contain either the names of the samples in the same order as they were given in the metadata file or the names of the individual sequences. If the first column contains

sequence names, the header must contain the names of the samples. If the first column contains the names of samples, the header must contain the names of the sequences.

In case the user requires PCA or PCoA calculation with distance matrix calculation using Bray-Curtis, the individual sequences can be named arbitrarily. However, if he wants to use a matrix calculation method that also requires the construction of a phylogenetic tree, whole sequences must be used as sequence names. This means that, for example, for PCA calculation, the sequence ACGTCTT can be stored in the table as a whole sequence, but also under some abbreviation (e.g. OTU1). For PCoA calculation using the UniFrac metric to calculate the distance matrix, the sequence can only be stored as an ACGTCTT string so that the sequence can be used to create a phylogenetic tree.

When the user wants to download transformed data and distance matrix, there is also a choice of three formats in which he can save the data. These formats are CSV, XLS, JSON.

Text files

For this tool, files with the extension csv, data, tsv, txt fall into this category. When file types are used, each column must be separated by typical delimiters such as a comma, semicolon, or tab. If the user does not use a space in his data in another way as a column separator (for example, in the name of individual samples or metadata categories), space is also allowed as a separator.

JSON format

There are several ways to write JSON files. In the case of this tool, for clarity of the entry was chosen the type, where the value that belongs to the given index is written as an object. An example of this format is in Figure 5.8.

```
{
  "sample1": {"OTU_1": "5", "OTU_2": "3", "OTU_3": "1", "OTU_4": "2", "OTU_5": "0"},
  "sample2": {"OTU_1": "4", "OTU_2": "3", "OTU_3": "4", "OTU_4": "0", "OTU_5": "2"}
}
```

Figure 5.8: Example of JSON format. The key is the name of the sample and the value are parameters. Values can be metadata attributes or n-dimensional coordinates.

Excel format

Files with extensions xlsx, xlsm, xltx, and xltm are supported. In this case, the general rules concerning the first column and heading mentioned above apply.

Biom-format

If the user decides to use this type of file, a small change occurs. The data needed for the calculation of the principal components and metadata are in one file. An example of this is shown in Figure 5.9. After loading the file in biom format, the possibility to load the file with metadata is hidden and immediately a list of categories according to which the data can be colored displays. The above-mentioned requirements concerning the formation of a phylogenetic tree apply in this case too.

```
{
    "id":null, "type": "OTU table", "generated_by": "QIIME revision 1.4.0-dev",
    "rows":[
            {"id":"AACCTCAAAA", "metadata":null},
            {"id":"AACTGG", "metadata":null},
            {"id":"AATCGTTATAAAA", "metadata":null},
            {"id":"AAGAAGCAATGACGGG", "metadata":null},
            {"id":"AATAAAGT", "metadata":null}
        ],
    "columns": [
            {"id":"Sample1", "metadata":{"meta1":"CGCT", "meta2":"CATG", "meta3":"0", "meta4":"false"}},
            {"id":"Sample2", "metadata":{"meta1":"CGCA", "meta2":"CATG", "meta3":"0", "meta4":"true"}},
            {"id":"Sample3", "metadata":{"meta1":"CGCT", "meta2":"CAAG", "meta3":"1", "meta4":"false"}},
            {"id":"Sample4", "metadata":{"meta1":"CACT", "meta2":"CATG", "meta3":"0", "meta4":"true"}}
            ],
    "matrix_type": "dense", "matrix_element_type": "int",
    "shape": [5,4],
    "data":  [[0,0,1,0],
              [5,1,0,2],
              [0,0,1,4],
              [2,1,1,0],
              [0,1,1,0]]
}
```


Figure 5.9: Example of biom-format file. This type of file contains both data for the calculation of principal components and also metadata. Another file is not needed.

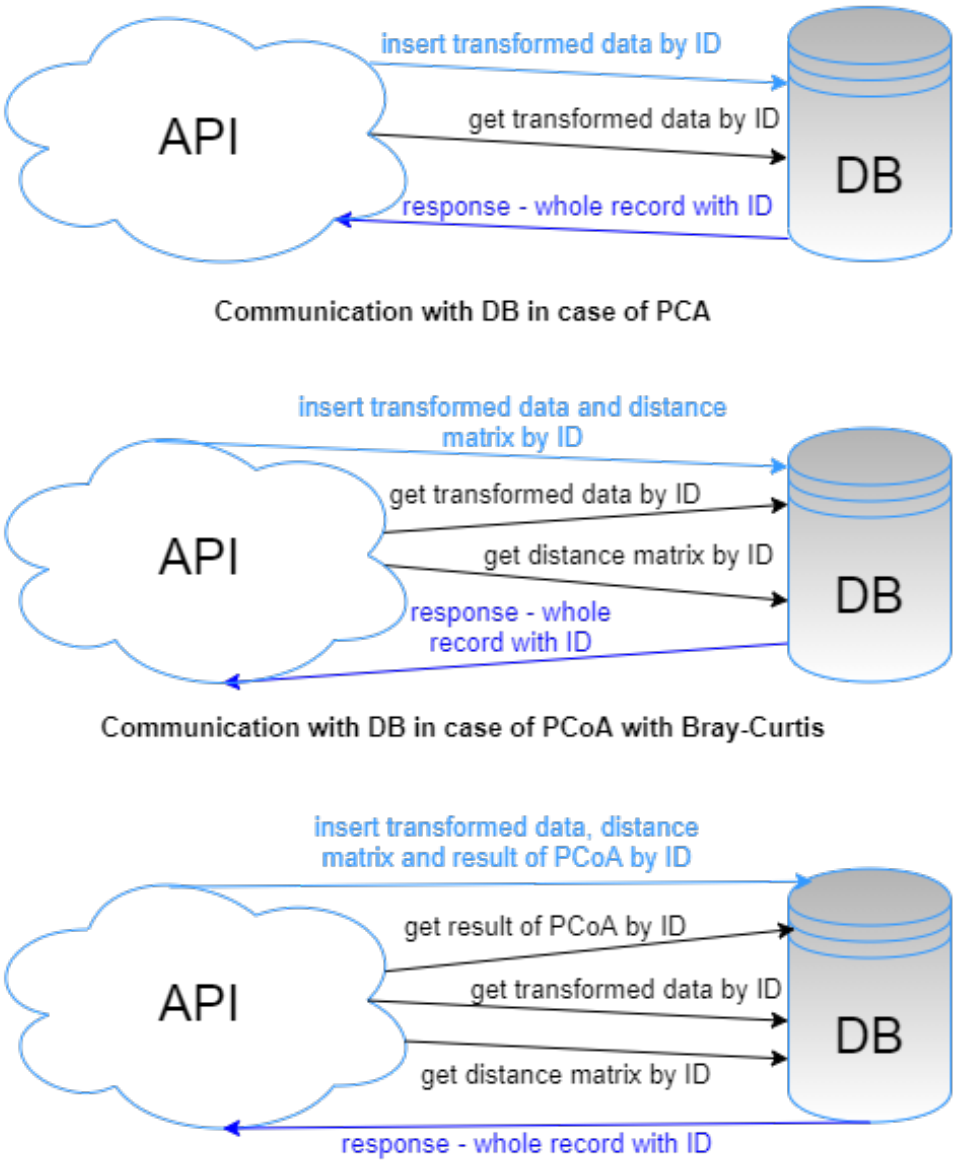

Figure 5.10: Examples of communication between the database and API for different types of requests. To the database can be saved or from it can be got transformed data, distance matrix or result of dimensionality reduction.

Chapter 6

Implementation

This chapter will describe the implementation of the proposed tool according to the tool design and functions, which were described in the previous chapters. The chapter will be divided into four parts - Implementation of user interface and its functions, API implementation, Implementation of the database and Testing. In each of implementation sections, the libraries and functions used by this tool will be explained in more detail. The source code can also be found in Appendix A.

6.1 Implementation of user interface and its functions

The UI of the tool is implemented in JavaScript using the library React [49], which is used to create graphical interfaces. Communication between the UI and the API is possible by sending a request in JSON format. HTML, CSS, and set of react component libraries React Suite [39] and package react-tooltip [51] are used to create individual components of the graphical interface.

Redux [2], React Redux [1], and simple selector library for redux - reselect [36] are used to manage store, which is a container status of variables. In the container, the status of all global variables that are used across the entire application is stored. In the case of this tool, the states of variables, which are used for subsequent plotting of graphs or further communication with the API, are stored in this container. The initialization of these variables can be found in Figure 6.1.

A small Formik library [12] is used to facilitate the creation of forms. The functions `pcaForm()` and `pcoaForm()` are used to display forms, and the next custom function `resetAll()` is used to set their default values.

The forms, among other components, also include input components of the type „file“, which are used to load the file from the user. The FileReader object [28] and its properties onloadend, onload and readAsText are used to open these files and load their content. The SheetJS js-xlsx package [41] is important for working with xlsx files. Custom functions `onChangeDataFile` and `onChangeMetadataFile` are also used during reading files from the user. Arguments of these functions are the current download event and the JSON, in which the data is stored for subsequent sending to the API. The file is sent to the API by the POST method, where it is processed and the path, where the file was stored on the server, is returned in the response. In the case of function `onChangeDataFile`, also the other form components are shown or hidden according to the selected file type. For this step, the function `onSelectFileType` is used. The tool also uses several other array

```
const baseInitialState = {
  x: [],          // arrays of x coordinates for each data category
  y: [],          // arrays of y coordinates for each data category
  z: [],          // arrays of z coordinates for each data category
  category: [],   // specific categories according to which data are colored
  evr: [],        // explained variability ratio for axes
  colGraph: [],   // data for the percentage of variance graph
  legend: [],     // order of name of metadata in categories
  id: [],         // ID of session
  maxPCx: [],     // the number of columns in the percentage of variance graph
  indexes: [],    // arrays of names or indexes of samples contained in each category
  showLegend: [], // is legend visible?
};
```

Figure 6.1: Initial states of variables in container status of variables. Their state is changed after the response from API is received and these variables are used for plotting the graph or download matrix or transformed data.

functions, which are used to correctly process the file according to its type and to obtain the header from the loaded metadata files. This tool uses FileSaver [17] to save the file on the client-side.

After filling these forms, the data from them in JSON format are sent by the array function using the POST method to the API. The content of sending JSON is set based on which confirmation button is pressed. In the case that the „Submit“ button is pressed, the entire JSON, the initialization of which is shown in Figure 6.2, is sent to the API. As a response, the data needed to plot graphs is returned. In the case of PCoA using the UniFrac, the ID for further requesting is returned. In this case, also functions `getMail()` and `waitingForData(id)` is used. The function `getMail()` is used to obtain and verify the email address from the user to send a link of the result of PCoA analysis using the UniFrac method. The function `waitingForData(id)` provides regular requests by ID of calculation using the POST method (in the interval of 5000 ms) to determine whether the result is ready for plotting or the calculation is still in progress. If the result is ready, the sending of requests is terminated and the result is visualized. If the download button for transformed data is pressed, the JSON content is the ID of the calculation to which the request relates, the type of file in which the data will be stored and the number of principal components which will be included in the resulting file. If the button for downloading the distance matrix is pressed, the contents of the JSON are the ID of the calculation to which the request relates and the type of file in which the matrix will be stored. When the button to download the matrix or download the transformed data is pressed, the response is the file, which is then saved to the user's computer.

After analyzing the javascript libraries that are used to work with graphs, the React Plotly.js library [21] was chosen to represent the result of the analyzes. This library allows the visualization of 2D and 3D scatter or bar graphs and using config provides all the functions that are needed for the desired work with graphs such as zoom in, zoom out, rotating, scrolling, or downloading them.

Some custom functions have been added to the functions of this library. The functions `hideLegendButton()` and `showLegendButton()` provide hiding and showing the graph legend. For preparing the result from analysis for plotting the graph is used function `prepareData()`. The function `pcxGraph()` is used for the input data processing to display

```
initialValues={{
    coloring: '',           // metadata for coloring
    dimension: '2D',        // type of dimensionality
    downloadType: 'csv',    // type of file for downloading transformed data
    file: '',               // path on the server to file for principal components calculation
    fileType: 'csv',        // type of file for principal components calculation
    mail: '',               // email address for sending the response
    matrix: '',             // method for matrix calculation
    matrixType: 'csv',      // type of file for downloading matrix
    metaFile: '',           // path on the server to metadata file
    metaFileType: 'csv',    // type of metadata file
    numberOfPC: '',         // number of principal components for the file with transformed data
    type: 'PCA',            // type of analysis
}}
```

Figure 6.2: Structure of JSON format, which is sent to API in case of a request for analysis calculation. Default values are set according to the required analysis.

the percentage of variance bar graph and the function `resultGraph()` is used for the preparing input dataset with results of the analysis for plotting the scatter graph.

If the tool is opened via a link, which is in the e-mail, a request is sent to the API using the POST method. The content of the request is the ID by which the calculation of the given analysis is registered. The expected answer is in JSON format. The content of this JSON is the data needed for subsequent visualization. This data is stored in the variable state container and has the same structure as the data in Figure 6.1.

6.2 API implementation

The API is programmed in Python version 3.8 using external programs MAFFT [24], FastTree [33] and several libraries such as numpy [19], pandas [35] or scikit-bio [40].

The API is called based on requests in JSON format that come from the user interface. The Flask library [18] is used to receive these requests and send responses to them (again in JSON format). Flask was chosen because of simple connection with JavaScript and the possibility of working on the principle multiple clients - one server. Depending on the type of request, it is either resolve immediately or queued for the worker and waits until it can be served. Each request is specific and different commands are executed to process it. This subchapter describes the procedures for processing these individual requests.

File save request

The file which should be saved and then used for the selected analysis is part of this type of request. A special folder is created to store all these files. The file is stored by using os.path [34] and the path to it is sent in a response to the request. This response is recorded on the UI side to preserve information about where the file was stored.

Analysis calculation request

Part of the request is JSON, which content is all the data needed for the analysis. The structure of this JSON is in the Figure 6.2. Initially, it is found whether the purpose is PCA or PCoA and, in the case of PCoA, also by which method the distance matrix

will be calculated. In the case of PCA or PCoA using Bray-Curtis, the request can be processed directly and a JSON with the data important for plotting the graph is sent in response. The example of JSON, which is contained in the response, is shown in Figure 6.3.


```
category: "{\"0\": \"0\", \"1\": \"1\"}"
colGraph: "[0.587859367574921, 0.3926786051659457, 0.01946202725913328, 1.1649549666563485e-35]"
evr: "[0.587859367574921, 0.3926786051659457, 0.01946202725913328]"
id: "3614a764b57a11eba30c5c80b6615755"
indexes: "[[\"Sample1\", \"Sample2\", \"Sample4\"], [\"Sample3\"]]"
legend: "\"meta3\""
maxPCx: 4
x: "[[3.4322176593802776, -0.32649274693096986, -1.403676691453398], [-1.7020482209959091]]"
y: "[[0.26920757533842304, -1.5744157135186239, 2.6259639637430734], [-1.3207558255628726]]"
z: "[[-0.14969709422223731, 0.5381613579839342, 0.09489471530706595], [-0.4833589790687627]]"
```


Figure 6.3: The example of JSON from the sending response. This JSON consists of another JSON that expresses the names of final metadata categories (key category), an array with the data for the percentage of variance plot (key colGraph), an array with values that express the explained variability ratio for graph axes (key evr), ID of calculation, array of arrays with names of samples for each final metadata category (key indexes), name of metadata categories for a legend, number of principal components in the result (key maxPCx) and arrays with resulting coordinates.

However, handling a request to calculate PCoA using the UniFrac method can take several minutes because it requires calculating the phylogenetic tree and aligning the sequences, which is one of the NP-complete problems. Therefore, this type of request is moved to the queue for the worker which uses threading [34], and only the ID under which the calculation is registered is sent to the UI in response. If the UI is active, it regularly uses this ID to send a request to know if the calculation is already completed. When the worker completes the calculation of the request, the necessary data is saved in the database, the request is marked as processed, and an e-mail (Figure 6.4) about the completion is sent to the user. For sending e-mail libraries smtplib [34] and ssl [34] are used.

The first step of both PCA and PCoA calculation is to process the files that have been uploaded using the form in the user interface. The content of these files is loaded into the dataFrame using pandas [35]. The selected method for loading these files depends on the file type. Subsequently, if it is necessary (for example, if the first column of metadata does not match the first column of data for the calculation of principal components), the created DataFrame is transformed.

In the case of PCoA, it is also necessary to calculate the distance matrix from the data for principal components calculation. For this calculation, the tool uses the skbio.diversity library [40]. The phylogenetic tree is also part of the calculation of the distance matrix by UniFrac methods. The calculation of the phylogenetic tree is provided by the external program FastTree [33]. The input for it is aligned sequences stored in the Newick file format. Sequence alignment is provided by an external MAFFT [24] program. The input of this program is sequences read from a file, which is uploaded by the user to a form, saved in fasta format. The output of the tool is a fasta file with aligned sequences. After creating the phylogenetic tree, the distance matrix is calculated. In the case of PCA, the calculation of the distance matrix is omitted.

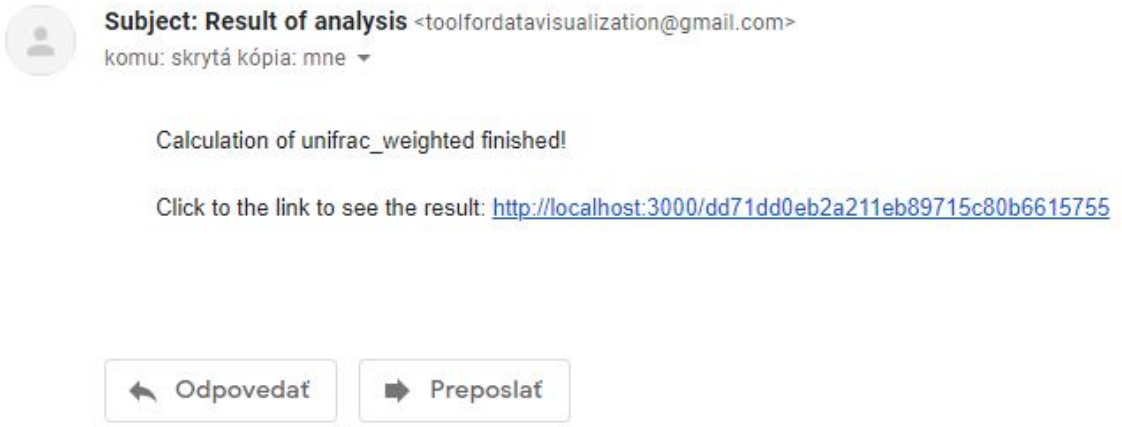

Figure 6.4: After completing the PCoA with the calculation of the distance matrix using UniFrac methods, an e-mail is sent to the user. In the mail is a link under which the user can open his visualization.

Then follows the calculation of PCA or PCoA and the division of data into groups according to selected metadata categories. In the case of PCA, sklearn.decom- position [31] is used, in the case of PCoA, skbio.stats.ordination [40] is used.

After the analysis calculation, the data is prepared for sending or storage in the database. Based on the filtering, the calculated values are divided into groups according to metadata and all data necessary for subsequent rendering on the user interface side are sent in JSON format. This JSON is almost identical to the data structure in the store (Figure 6.2), only the showLegend item is missing.

Result requirement for PCoA using Unifrac

If the user decides to wait for the PCoA result using UniFrac and does not close the tool webpage, after receiving the ID under which the calculation is registered, the UI starts sending requests whether the calculation is already completed or not at regular intervals (5000 ms). The API then sends a query for the record with the given ID to the database. If the calculation is still processed and the result is not yet in the database, the API sends a response in JSON format with empty content. When the calculation is completed, the important data for visualization is sent to the UI in JSON format.

Request to retrieve data based on email

If the user opens a link from the e-mail that was delivered to him, the opened UI sends a request to obtain the result, which is registered under the ID in the database. This ID is part of the sent link. The API receives the request and sends a query for a record with this ID to the database. After receiving the response API sends it in the appropriate format to the UI.

Request to store matrix or transformed data

Part of this type of request is the ID under which the required data should be saved in the database, the type of file in which they will be stored, and, in the case of transformed data, the number of principal components that will be the content of the file. After retrieving the data from the database, they are processed using pandas dataFrame and stored in a file of the appropriate type. After saving and processing using codecs [34], this file is in the form of a response to a JSON request sent to the UI. Files, created during the process, that are no longer needed are deleted.

6.3 Implementation of the database

This tool uses library sqlite3 [20] for data storage. In the beginning, a connection to the database then four tables are created. These tables contain the identifiers of the individual calculations, the results of the PCoA analysis using UniFrac, the resulting distance matrices, and the transformed data. Based on requests from the API, data is inserted into or obtained from these tables. Several functions are created for these operations.

- `insertDBResult(item)` - this function is used for storing the result from the variable item in a table with the results of PCoA analysis using the UniFrac method. The function is called after the end of the PCoA calculation using the UniFrac method.
- `insertDBDistanceMatrix(item)` - the function allows storing the data about the distance matrix from the variable item in a table with the resulting distance matrices. The function is called after the PCoA calculation is completed.
- `insertDBTransformedData(item)` - the function allows storing the result of the computation of the principal components from the variable item into a table with the resulting transformed data. The function is called after completing any analysis.
- `insertDBIdentificator(item)` - this function is used for storing the calculation identifier from the variable item in a table with calculation identifiers. The function is called when a new request is accepted.
- `getItemDBResult(idValue)` - the function allows obtaining a record of the result of PCoA analysis using UniFrac according to the corresponding value idValue. The function is called upon a request from the UI to determine whether the PCoA calculation using UniFrac is completed or not. If it is not completed, the database returns a response in the form of an empty field. If the calculation has been completed, the database returns the appropriate record. The function is also called after opening the line from the received email.
- `getItemDBDistanceMatrix(idValue)` - this function allows obtaining distance matrix data using the ID from variable idValue. It is called when a request to store a distance matrix arrives at the API.
- `getItemDBTransformedData(idValue)` - the function is used to retrieve transformed data from a transformed data table using the idValue variable. It is called when a request to store a distance matrix arrives at the API.

- `getItemDBIdentificator(idValue)` - this function allows to find out whether the generated ID from idValue is already in the database and a new one needs to be generated or this ID can be used as an identifier of the new calculation. If the ID is already in the table of calculation identifiers, it returns True, otherwise False.

Chapter 7

Evaluation

The tool has been tested on several data sets of different sizes and types. These datasets have been specifically generated to resemble as closely as possible the data used in the microbiome analysis. The goal of this testing was to observe whether the visualizations from this tool meet input data expectations. An example of specifically generated input data is given in Appendix A and the example of the result is shown in Figure 7.3. The tool was also tested on the existing Iris Data Set [11], which was subsequently modified in several ways to verify the functionality of the tool.

The average calculations of the individual analyzes can be found in the graphs in the Figure 7.1 and in Figure 7.2.

The dependence of time on the number of items in the dataset is linear. In the case of the PCA or PCoA calculation using the Bray-Curtis method with the number of items in the dataset up to 18000, the calculation takes only milliseconds. On the other hand, the PCoA calculations using any UniFrac method can take a few seconds or minutes. The PCoA calculation method using Bray-Curtis was evaluated as the fastest method for reducing dimensionality based on testing. The PCoA method with the calculation of the distance matrix using weighted UniFrac was evaluated as the most time-consuming.

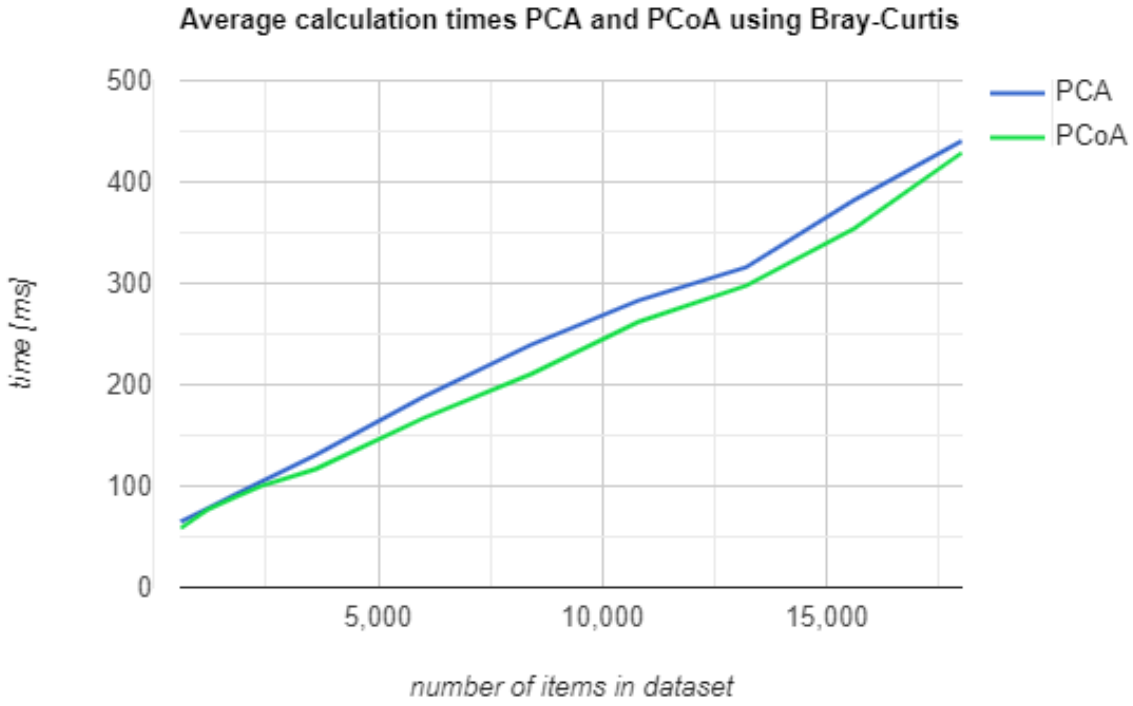

Figure 7.1: The graph shows the average times in the calculation of PCA and PCoA using Bray-Curtis in ms depending on the number of elements in the dataset with an abundance of sequence variants.

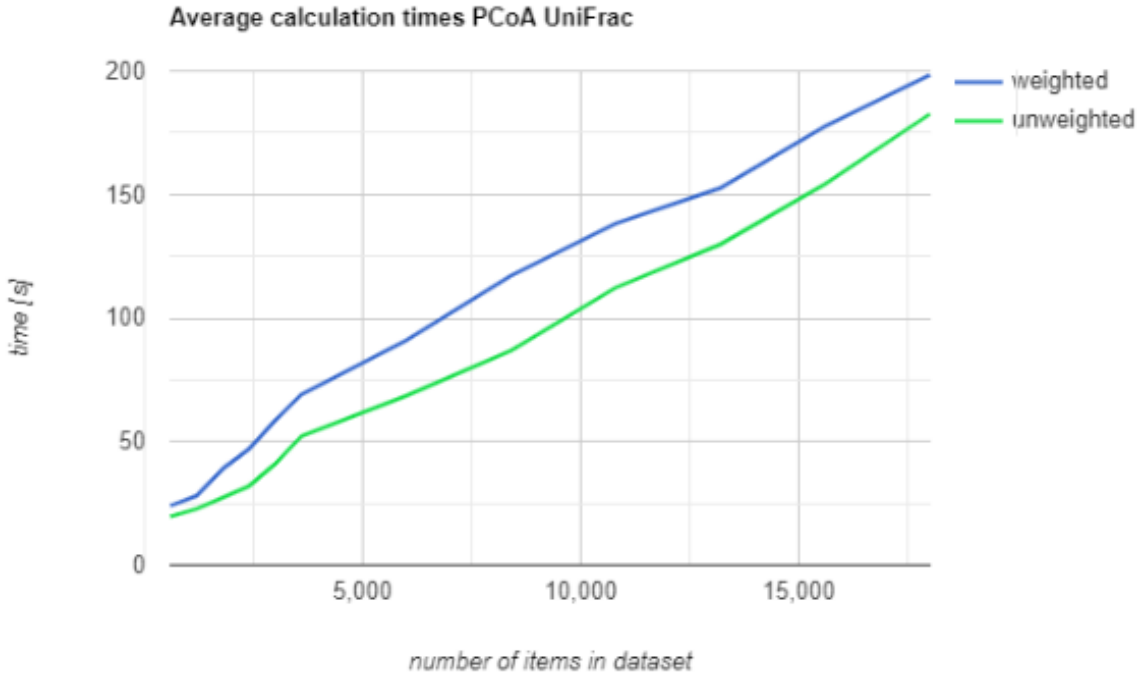

Figure 7.2: The graph shows the average times in the calculation of PCoA using UniFrac weighted and UniFrac unweighted in seconds depending on the number of elements in the dataset with an abundance of sequence variants.

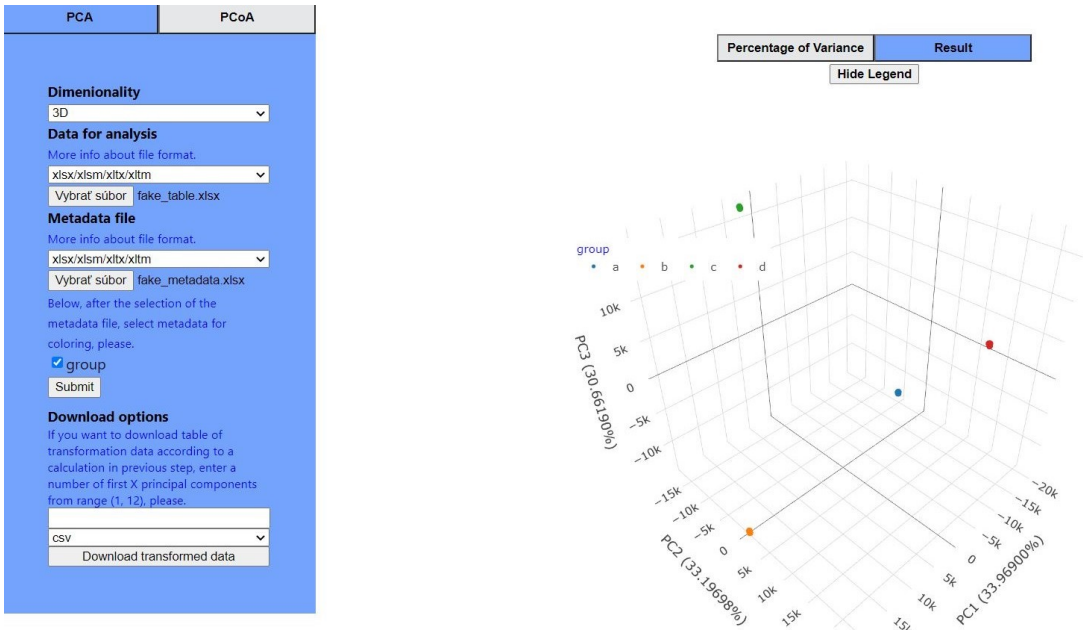

Figure 7.3: The visualization of data, which are in Appendix A. The dataset was designed that way, that the data should form four clusters. Each cluster should consist of three samples.

Chapter 8

Conclusion

The aim of this work was to analyze existing tools for visualization of microbial data (Emperor) and for general multidimensional data visualization (TensorFlow Embedding Projector). Based on the identified strengths and weaknesses, a new tool for visualization of microbial data was designed. The proposed tool addresses some of the identified shortcomings of existing tools.

The proposed tool meets the basic requirements that are necessary for microbial research and tries to make the user's work as easy as possible. It can be used to calculate and visualize both PCA and PCoA. In the case of PCoA calculation, there are three methods to calculate the distance matrix - Bray-Curtis, UniFrac weighted and UniFrac unweighted. The result of the calculated matrix can be saved in CSV, JSON, or Excel format. The same option to save the file also applies to the calculated values of principal components. The first values of the percentage of variance are also plotted. The results of the analyzes can be visualized in 2D or 3D space and colored based on any combination of metadata categories. Although the tool is designed as a web tool, deployment on a server has not yet taken place.

The currently proposed tool as x, y, or z axes takes the first two or three principal components, which have the highest value of explained variability ratio. In the future, the user could be able to choose his own choice of principal components that would represent the x, y, and z axes. Other methods such as UMAP and t-SNE, which are part of the Tensorflow Embedding Projector, could also be added to methods for dimension reduction. The proposed tool currently expects two types of files - a file with metadata and a file with an abundance of sequence variants. Another improvement of the tool could be that it will also calculate the abundance of sequence variants itself. With the addition of new functionalities, the user interface of the tool would also be modified.

Bibliography

[1] Abramov, D. *React Redux*. 2015 [cit. 12. 05. 2021]. Available at: https://github.com/reduxjs/react-redux.

[2] Abramov, D. and Clark, A. *Redux*. 2015 [cit. 12. 05. 2021]. Available at: https://github.com/wwayne/react-tooltip.

[3] Alberts, B. *Molecular Biology of the Cell*. 9th ed. New York: Taylor and Francis Group, 2014. ISBN 978-1-31573-536-8. Available at: https://www.proquest.com/docview/2148817083/4ABD4B3E14F8407EPQ/1?accountid=17115.

[4] Bates, A. D. *DNA topology*. 2nd ed. Oxford University Press, 2005. ISBN 978-0-19-850655-3.

[5] Borg, I. and Groenen, P. *Modern Multidimensional Scaling: Theory and Applications*. 2nd ed. Springer New York, 2005. Springer Series in Statistics. ISBN 978-0387-25150-9. Available at: https://books.google.cz/books?id=duTODldZzRcC.

[6] Brown, S. M. *Next-generation DNA sequencing informatics*. 2nd ed. Cold Spring Harbor Laboratory Press, 2015. ISBN 978-1-62182-123-6.

[7] Caporaso, J., Kuczynski, J. and Stombaugh, J. e. a. QIIME allows analysis of high-throughput community sequencing data. *Nature Methods*. 7th ed. 2010, no. 5, p. 335–336. ISSN 1548-7105. Available at: https://doi.org/10.1038/nmeth.f.303.

[8] Cavalli-Sforza, L. L. and Edwards, A. W. F. Phylogenetic analysis: models and estimation procedures. *Evolution*. 21st ed. 1967, no. 3, p. 550–570. DOI: https://doi.org/10.1111/j.1558-5646.1967.tb03411.x.

[9] Chen, K. and Pachter, L. Bioinformatics for Whole-Genome Shotgun Sequencing of Microbial Communities: e24. *PLoS Computational Biology*. 1st ed. july 2005, no. 2. DOI: http://dx.doi.org/10.1371/journal.pcbi.0010024.

[10] Edgar, R. C. and Batzoglou, S. Multiple sequence alignment. *Current Opinion in Structural Biology*. 16th ed. 2006, no. 3, p. 368–373. DOI: https://doi.org/10.1016/j.sbi.2006.04.004. ISSN 0959-440X. Available at: https://www.sciencedirect.com/science/article/pii/S0959440X06000704.

[11] Fisher, R. A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*. 7th ed. 1936, no. 2, p. 179–188. DOI: https://doi.org/10.1111/j.1469-1809.1936.tb02137.x. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x.

[12] Formium, I. *Formik*. 2020 [cit. 12. 05. 2021]. Available at: https://github.com/formium/formik.

[13] Garibyan, L. and Avashia, N. Polymerase Chain Reaction. *Journal of Investigative Dermatology*. 133th ed. 2013, no. 3, p. 1–4. Available at: https://doi.org/10.1038/jid.2013.1.

[14] Genome Research Limited. *What is DNA?* January 2016 [cit. 26. 04. 2021]. Available at: https://www.yourgenome.org/facts/what-is-dna.

[15] Genome Research Limited. *What is the 'Central Dogma'?* January 2016 [cit. 15. 04. 2021]. Available at: https://www.yourgenome.org/facts/what-is-the-central-dogma.

[16] Gorodkin, J. and Ruzzo, W. L. *RNA sequence, structure, and function : computational and bioinformatic methods*. 1st ed. Humana Press, 2014. ISBN 978-1-62703-708-2.

[17] Greys, E. *FileSaver.js*. 2011 [cit. 11. 05. 2021]. Available at: https://github.com/eligrey/FileSaver.js.

[18] Grinberg, M. *Flask Web Development: Developing Web Applications with Python*. 1stth ed. O'Reilly Media, Inc., 2014. ISBN 1449372627.

[19] Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P. et al. Array programming with NumPy. *Nature*. 585th ed. Springer Science and Business Media LLC. september 2020, no. 7825, p. 357–362. DOI: 10.1038/s41586-020-2649-2. Available at: https://doi.org/10.1038/s41586-020-2649-2.

[20] Häring, G. *Sqlite3*. 2006 [cit. 15. 04. 2021]. Available at: https://github.com/python/cpython/tree/3.9/Lib/sqlite3/.

[21] Johnson, A., Parmer, J., Parmer, C. and Sundquist, M. *Plotly*. 2012 [cit. 11. 05. 2021]. Available at: https://plotly.com/.

[22] Johnson, J. S., Spakowicz, D. J., Bo Young, H., Petersen, L. M., Demkowicz, P. et al. Evaluation of 16S rRNA gene sequencing for species and strain-level microbiome analysis. *Nature Communications*. 10th ed. 2019, no. 1, p. 1–11, [cit. 20. 04. 2021]. DOI: http://dx.doi.org/10.1038/s41467-019-13036-1.

[23] Jones, N. C. *Introduction to bioinformatics algorithms*. 1st ed. MIT Press, 2004. ISBN 0-262-10106-8. Available at: https://books.google.cz/books?id=p_qzpkNVcUwC&printsec=frontcover&hl=sk&source=gbs_ViewAPI&redir_esc=y#v=onepage&q&f=false.

[24] Katoh, K. *MAFFT*. 2007 [cit. 09. 05. 2021]. Available at: https://mafft.cbrc.jp/alignment/software/.

[25] Krane, D. E. *Fundamental concepts of bioinformatics*. 1st ed. Benjamin Cummings, 2002. ISBN 0-8053-4633-3.

[26] Lozupone, C. and Knight, R. UniFrac: a New Phylogenetic Method for Comparing Microbial Communities. *Applied and Environmental Microbiology*. 71th ed. American Society for Microbiology Journals. 2005, no. 12, p. 8228–8235, [cit. 12. 01. 2021]. DOI: 10.1128/AEM.71.12.8228-8235.2005. ISSN 0099-2240. Available at: https://aem.asm.org/content/71/12/8228.

[27] Lozupone, C. A., Hamady, M., Kelley, S. T. and Knight, R. Quantitative and Qualitative beta Diversity Measures Lead to Different Insights into Factors That Structure Microbial Communities. *Applied and Environmental Microbiology*. 73th ed. American Society for Microbiology Journals. 2007, no. 5, p. 1576–1585, [cit. 18. 01. 2021]. DOI: 10.1128/AEM.01996-06. ISSN 0099-2240. Available at: https://aem.asm.org/content/73/5/1576.

[28] Mozilla and contributors individual. *FileReader*. 2005 [cit. 12. 05. 2021]. Available at: https://github.com/mdn/content/blob/main/files/en-us/web/api/filereader/index.html.

[29] Nelson, K. and White, B. *Metagenomics: Theory, Methods and Applications*. 1st ed. Norfolk: Caister Academic Press, 2010. 171-182 p. ISBN 978-1-904455-54-7.

[30] Olson, M. V. Clone by clone by clone. *Nature*. 409th ed. Feb 01 2001, no. 6822, p. 816–818, [cit. 10. 03. 2021]. DOI: http://dx.doi.org/10.1038/35057271. ISSN 1476-4687.

[31] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 12th ed. 2011, no. 85, p. 2825–2830.

[32] Pevsner, J. *Bioinformatics and functional genomics*. 3rd ed. Wiley Blackwell, 2015. ISBN 978-1-118-58178-0. Available at: https://books.google.cz/books?id=dgmeCAAAQBAJ&printsec=frontcover&hl=sk&source=gbs_ViewAPI&redir_esc=y#v=onepage&q&f=false.

[33] Price, M. *FastTree*. 2009 [cit. 09. 05. 2021]. Available at: http://www.microbesonline.org/fasttree/.

[34] Python Software Foundation. *The Python Standard Library — Python 3.9.5 documentation*. 2001 [cit. 04.02.2021].

[35] Reback, J., McKinney, W., Bossche, J., Augspurger, T., Cloud, P. et al. *Pandas-dev/pandas: Pandas*. 2020 [cit. 08. 04. 2021]. Available at: https://doi.org/10.5281/zenodo.3509134.

[36] Redux.js. *Reselect*. 2018 [cit. 09. 05. 2021]. Available at: https://github.com/reduxjs/reselect.

[37] Reid, A. and Greene, S. *FAQ: human microbiome*. General Information. American Academy of Microbiology, 2013. I,II,1-16 p. Available at: https://www.proquest.com/reports/faq-human-microbiome/docview/1655684906/se-2?accountid=17115.

[38] Ricotta, C. and Podani, J. On some properties of the Bray-Curtis dissimilarity and their ecological meaning. *Ecological Complexity*. 31st ed. 2017, p. 201–205. DOI: https://doi.org/10.1016/j.ecocom.2017.07.003. ISSN 1476-945X.

[39] RSUITE TEAM. *React Suite*. 2016 [cit. 12. 05. 2021]. Available at: https://github.com/rsuite/rsuite.

[40] SCIKIT BIO DEVELOPMENT TEAM. *Scikit-bio: A Bioinformatics Library for Data Scientists, Students, and Developers*. 2020. Available at: http://scikit-bio.org.

[41] SHEETJS TEAM. *Sheetjs*. 2013 [cit. 12. 05. 2021]. Available at: https://github.com/SheetJS/sheetjs.

[42] SHLENS, J. *A Tutorial on Principal Component Analysis*. 2014. Available at: https://arxiv.org/pdf/1404.1100.pdf.

[43] SMILKOV, D. and BIG PICTURE. *Embedding Projector - Visualization Of High-Dimensional Data*. 2015 [cit. 06.04.2021]. Available at: https://www.tensorflow.org/.

[44] SUNDARARAJAN, V. S., MALIK, G., IJAQ, J., KUMAR, A., DAS, P. S. et al. HYPO: A database of hypothetical human proteins. *BioRxiv*. 25th ed. Cold Spring Harbor Laboratory. 2017, no. 8. DOI: 10.1101/202887. Available at: https://www.biorxiv.org/content/early/2017/10/13/202887.

[45] SZEBERÉNYI, J. Problem-solving test: Pyrosequencing. *Biochemistry and Molecular Biology Education*. 41th ed. 2013, no. 2, p. 112–115. DOI: https://doi.org/10.1002/bmb.20688. Available at: https://iubmb.onlinelibrary.wiley.com/doi/abs/10.1002/bmb.20688.

[46] VAN DIJK, E. L., AUGER, H., JASZCZYSZYN, Y. and THERMES, C. Ten years of next-generation sequencing technology. *Trends in Genetics*. 30th ed. 2014, no. 9, p. 418–426. DOI: https://doi.org/10.1016/j.tig.2014.07.001. ISSN 0168-9525. Available at: https://www.sciencedirect.com/science/article/pii/S0168952514001127.

[47] VAZQUEZ BAEZA, Y., PIRRUNG, M., GONZALEZ, A. and KNIGHT, R. *EMPeror: a tool for visualizing high-throughput microbial community data* [online]. November 2013 [cit. 20. 09. 2020]. Available at: https://biocore.github.io/emperor/.

[48] VETROVSKY, T. and BALDRIAN, P. The Variability of the 16S rRNA Gene in Bacterial Genomes and Its Consequences for Bacterial Community Analyses. *PLoS ONE*. 8th ed. Public Library of Science. 2013, no. 2. ISSN 1932-6203. Available at: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0057923.

[49] WALKE, J. *React*. 2013 [cit. 20. 04. 2021]. Available at: https://github.com/facebook/react.

[50] YONG, E. *I contain multitudes*. 1st ed. Vintage, 2017. ISBN 978-17-8470-017-1.

[51] ZIXIAO, W. *React Tooltip*. 2020 [cit. 10. 05. 2021]. Available at: https://github.com/wwayne/react-tooltip.

Appendix A

Contents of the Attached DVD

- doc/ - source files of this text
- Dockerfile - file for preparing Docker
- inputFiles/ - example of input files
- project.tar - source files of the implemented tool
- README.md
- start.sh - script to run a tool for Docker
- xmisak03.pdf - electronic version of this text