**BRNO UNIVERSITY OF TECHNOLOGY**
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

**FACULTY OF INFORMATION TECHNOLOGY**
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

**DEPARTMENT OF COMPUTER SYSTEMS**
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

# BIOINFORMATIC TOOL FOR ESTIMATION OF ABUNDANCES OF BACTERIAL FUNCTIONAL MOLECULES IN BIOLOGICAL SAMPLES BASED ON 16S RRNA METAGENOMIC DATA

BIOINFORMATICKÝ NÁSTROJ PRO ODHAD ABUNDANCE BAKTERIÁLNÍCH FUNKČNÍCH MOLEKUL

V BIOLOGICKÝCH VZORCÍCH NA ZÁKLADĚ METAGENOMICKÝCH DAT 16S RRNA

**MASTER'S THESIS**
DIPLOMOVÁ PRÁCE

**AUTHOR** **Bc. MICHAELA BIELIKOVÁ**
AUTOR PRÁCE

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

BRNO 2019

**Brno University of Technology**
Faculty of Information Technology

Department of Computer Systems (DCSY) Academic year 2018/2019

# Master's Thesis Specification

21657

| | |
|---|---|
| Student: | **Bieliková Michaela, Bc.** |
| Programme: | Information Technology Field of study: Bioinformatics and biocomputing |
| Title: | **Bioinformatic Tool for Estimation of Abundances of Bacterial Functional Molecules in Biological Samples Based on 16S rRNA Metagenomic Data** |
| Category: | Biocomputing |

Assignment:

1. Study basic principles of metagenomics and its applications to the analysis of microbiome using the 16s rRNA amplicon sequencing. Research existing methods for estimation of bacterial functional molecule abundance based on 16s rRNA metagenomic data. Focus primarily on the tools PICRUSt, Tax4Fun and Paprica.
2. Evaluate advantages and disadvantages of existing methods. Based on your evaluation, propose a new tool for estimation of bacterial functional molecule abundance. The tool can implement completely new estimation method, or can serve as a consensual tool, which appropriately combines outputs of other existing methods in order to achieve better estimation accuracy.
3. Implement the proposed tool and evaluate its performance on an appropriate testing dataset.
4. Evaluate achieved results and discuss possibilities of future continuation of the project.

Recommended literature:

- Based on instructions from the supervisor.

Requirements for the semestral defence:

- Completion of tasks 1 and 2 from the specification.

Detailed formal requirements can be found at http://www.fit.vutbr.cz/info/szz/

| | |
|---|---|
| Supervisor: | **Smatana Stanislav, Ing.** |
| Head of Department: | Sekanina Lukáš, prof. Ing., Ph.D. |
| Beginning of work: | November 1, 2018 |
| Submission deadline: | May 22, 2019 |
| Approval date: | October 26, 2018 |

# Abstract

Humans are host to an enormous variety of microbes, bacterial, archaeal, fungal, and viral. Some of these can cause serious diseases, but others, particularly gut microbiome, are essential to human life. Unfortunately, the gut microbiome is not well documented, since it contains thousands of different kinds of bacteria most of which cannot be cultivated in laboratories, and we do not know all of its functions. The recent solution to this problem seems to be high-throughput sequencing in combination with bioinformatics tools for functional profile prediction. In this thesis, bioinformatics tools for functional profile prediction will be introduced, along with their advantages and disadvantages. The goal of this thesis is to create a new tool for functional profile prediction, which can either employ a consensus of the existing tools or can be a brand new tool inspired by these.

# Abstrakt

Ľudské telo je prostredím pre život neuveriteľného množstva mikróbov. Niektoré z nich môžu spôsobovať rôzne choroby, ale ďalšie, napríklad črevný mikrobióm, sú pre život a zdravie človeka nepostrádateľné. Nanešťastie, črevný mikrobióm nie je detailne preštudovaný, pretože obsahuje tisíce rôznych druhov baktérií, z ktorých väčšina sa nedá kultivovať v laboratórnych podmienkach. Riešením tohto problému sú nové rýchle metódy sekvenovania v kombináciou s bioinformatickými nástrojmi na výpočet funkčného profilu baktérií vo vzorke. V tejto práci si predstavíme existujúce nástroje predpovedajúce funkčný profil, a následne navrhneme nový nástroj, ktorý môže implementovať konsenzus nad výsledkami existujúcich nástrojov, alebo sa môže jednať o úplne nový nástroj.

# Keywords

bioinformatics, metagenomics, bacterial functional profile, KO profile, 16S rRNA, PiCRUST

# Kľúčové slová

bioinformatika, metagenomika, bakteriální funkční profil, KO profil, 16S rRNA, PiCRUST

# Reference

BIELIKOVÁ, Michaela. *Bioinformatic Tool for Estimation of Abundances of Bacterial Functional Molecules in Biological Samples Based on 16S rRNA Metagenomic Data*. Brno, 2019. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Stanislav Smatana

# Rozšírený abstrakt

Ľudské telo je domovom veľkého množstva rôznych mikróbov, baktérií a vírusov. Niektoré z nich sú škodlivé a môžu spôsobiť rôzne ochorenia, ale veľa ďalších je pre život človeka nevyhnutných. Konkrétne črevný mikrobióm je veľmi dôležitý pre správne fungovanie tráviaceho systému. V posledných rokoch bolo dokázané, že zmeny stavu črevného mikrobiómu spôsobujú rôzne zdravotné komplikácie od porúch trávenia až po psychické problémy, ako napríklad depresia. Napriek tomu zostáva črevný mikrobióm človeka stále neprebádaný, čo je spôsobené hlavne veľkým množstvom prítomných druhov baktérií a nemožnosťou niektoré z nich pestovať v laboratóriách. Moderné metódy mikrobiológie tieto prekážky odstraňujú prostredníctvom efektívnych metód sekvenovania a metagenomiky. [21, 28]

Keďže väčšina baktérií prítomná v črevnom mikrobióme ešte nebola preštudovaná a zdokumentovaná, hlavná otázka pri ich analýze nie je aké druhy baktérií sa vo vzorke nachádzajú, ale aké majú funkcie (napr. trávenie cukrov, tukov, rezistencia k antibiotikám). Hľadanie odpovede na túto otázku sa nazýva predikcia funkčného profilu a je hlavným zameraním tejto práce. Predikcia funkčného profilu je založená na zistení, že druhy baktérií s podobnou RNA sekvenciu majú podobné funkcie, zatiaľ čo druhy baktérií, ktorých sekvencie sa veľmi odlišujú, majú odlišné funkčné profily. [28]

V tejto práci popíšem existujúce nástroje pre zisťovanie funkčného profilu podľa sekvencií, konkrétne PiCRUST, Tax4Fun a Paprica. Budem sa zaoberať metódami, ktoré využívajú pre predikciu, ktoré RNA databázy používajú, a ich výhody a nevýhody. Cieľom práce je vytvoriť nový nástroj na predikciu funkčného profilu, ktorý môže implementovať konsenzus nad spomínanými nástrojmi, alebo sa môže jednať o úplne nový nástroj, ktorý bude existujúcimi prístupmi iba inšpirovaný. Mal by implementovať viacero metód pre predikciu funkčného profilu, vrátane nových metód, ktoré nie sú použité v žiadnom existujúcom nástroji. Tieto metódy budú porovnané medzi sebou a potom s najpoužívanejším existujúcim nástrojom — s PiCRUSTom.

V texte práce najprv definujem základné pojmy z bioinformatiky v rozsahu potrebnom pre pochopenie ďalších častí práce. Popisujem dva typy analýzy vzorky — analýzu bakteriálnej kompozície, ktorá odpovedá na otázku, ktoré druhy baktérií sú prítomné vo vzorke, a predikciu funkčného profilu, ktorá sa zaoberá tým, aké majú tieto baktérie funkcie a v akom množstve sú dané funkcie zastúpené. Dôraz je kladený na predikciu funkčného profilu, ktorú popisujem podrobnejšie a vysvetľujem aj metódy, ktoré sa pre ňu dajú použiť. Tieto metódy sa dajú podľa princípu na ktorom sú založené rozdeliť do troch skupín — metódy založené na vzdialenosti, na fylogenetickom strome a na lineárnej regresii. Popísané existujúce nástroje používajú metódy založené na fylogenetických stromoch, zvyšné dve skupiny sú mojou autorskou prácou.

V samostatnej kapitole sa venujem existujúcim nástrojom, popisujem ich a porovnávam ich výhody a nevýhody. Vysvetľujem, prečo som sa rozhodla namiesto konsenzusu implementovať úplne nový nástroj a v ďalšej kapitole popisujem jeho návrh a implementáciu.

Posledná kapitola je venovaná experimentom s implementovanými metódami. Najprv popisujem testovací framework a spôsob vyhodnotenia. Navrhujem novú metódu na vyhodnocovanie presnosti predikcie funkčného profilu, ktorá sa namiesto všetkých bakteriálnych funkcií sústreďuje len na tie najviac špecifické. Potom robím experimenty pre každú spomínanú skupinu metód. Skúšam rôzne úpravy prístupov a parametrov s cieľom zlepšenia celkovej korelácie očakávaného a odhadovaného funkčného profilu. Následne porovnávam najlepšie výsledky pre každú skupinu metód navzájom. Z metód založených na vzdialenosti dávala najlepšie výsledky predikcia založená na prahu relatívnej podobnosti, z metód založených na fylogenetických stromoch bol najlepší strom vytvorený cez neighbour-

joining, a lineárna regresia najlepšie funguje cez predikciu podľa nezarovnanej RNA sekvencie. Spomedzi skupín metód vykazujú najlepšie výsledky fylogenetické stromy. Na záver porovnávam svoj výsledok s výsledkom získaným z vlastnej implementácie Picrustu. V tomto porovnaní je môj nástroj mierne lepší ako pre všetky bakteriálne funkcie, tak aj pre špecifické.

V závere zhrniem dosiahnuté výsledky a diskutujem možnosti ďalšieho rozšírenia práce. Jedným z možných smerov je predstavenie upravenej metódy vyhodnocovania nástrojov bioinformatickej komunite. Takisto vytvorený nástroj, vzhľadom na to, že vykazuje lepšie výsledky ako Picrust, má potenciál na publikovanie, čo bude možné po dôkladnejšom otestovaní.

# Bioinformatic Tool for Estimation of Abundances of Bacterial Functional Molecules in Biological Samples Based on 16S rRNA Metagenomic Data

## Declaration

Hereby I declare that this term project was prepared as an original author's work under the supervision of Ing. Stanislav Smatana.

........................
Michaela Bieliková
May 20, 2019

## Acknowledgements

I would like to thank my supervisor Ing. Stanislav Smatana for his guidance, advice, and patience. Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum provided under the program "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated.

# Contents

# Chapter 1

# Introduction

Humans are hosts to an enormous variety of microbes, bacterial, archaeal, fungal, and viral. Some of these are invaders that can cause serious diseases, but there is a lot of microbes that are essential to human life. Particularly gut microbiome is crucial for the regular function of the digestion tract. In the last years, it was proven that irregularities in the gut microbiome are linked to many conditions ranging from digestion tract diseases like inflammatory bowel disease to psychic conditions like depression. Unfortunately, because of a big variety of present bacterial species and the impossibility to cultivate most of them in laboratories, science knows only little about the gut microbiome. Modern approaches in microbiology, specifically high-throughput sequencing and metagenomics, seem to be able to solve these problems and allow us to study microbiome thoroughly and understand how it is connected to human health. [21, 28]

Since most of the bacteria present in the gut microbiome has not been studied yet, the main question is not which species of bacteria a specific sample contains, but instead what can the bacteria in this sample do (i.e. lipid digestion or resistance to antibiotics). This task is called functional profile prediction and it will be the main focus of this thesis. Functional profile prediction is based on the observation, that bacteria species with similar RNA sequence tend to have similar functions, whereas between species with small RNA similarity the functional profile differs. [28]

In this thesis, existing bioinformatics tools for functional profile prediction will be described, namely PiCRUST, Paprica, and Tax4Fun. We will discuss methods they use for prediction, which RNA database they use, and their advantages and disadvantages. The goal of this thesis is to create a new tool for functional profile prediction, which can either be a consensus tool built on PiCRUST, Paprica, and Tax4Fun, or a brand new tool inspired by these. It should implement various methods for functional profile prediction, including new algorithms that are not used in any of the mentioned tools, and compare the accuracy of the new methods with existing tools.

In Chapter 2 I explain fundamental bioinformatics concepts to an extent to make the thesis understandable for informaticians with no knowledge of biology. Next, I describe the process of bacterial composition analysis. In Chapter 3, I discuss functional profile prediction. In Chapter 4 I describe existing bioinformatic tools for this purpose. In Chapter 5, I will introduce the new tool. In Chapter 6, I discuss the results of the experimental evaluation of the created tool. Summarization and possible future improvements will be given in Chapter 7.

# Chapter 2

# Bacterial composition analysis

This chapter contains a theoretical background for this thesis. First, we will define the fundamental concepts of bioinformatics. Then, we will discuss 16S rRNA, KEGG Orthologs and the details of bacterial composition analysis of a given sample.

## 2.1 Genetics

Genetics is a part of science that studies the way characteristics are transformed from one generation to the next. The main focus of genetics are molecules called DNA, where all the genetic information of an organism is stored. [2]

*DNA* stands for deoxyribonucleic acid. It has a form of a strand consisting of nucleotides, which are built from a ribose sugar, a phosphate group, and a nitrogen base. There are four nitrogen bases that can be part of a DNA strand - adenine, thymine, cytosine, and guanine. The three-dimensional structure of DNA, which can be seen in Figure 2.1, is a spiral called double helix. It consists of two strands which are bound by hydrogen bonds between the nitrogen bases. The rules of base pairing say that adenine always pairs with thymine and cytosine with guanine. [20]

The genetic information is coded by the order of nitrogen bases in the strands. A specific segment of the DNA strand that holds the information needed for a certain function is called a *gene*. The function may be coded directly, or the coded information may be a template for a protein performing the function. Intuitively, we can imagine DNA sequence as a prescription, which holds the information to creating molecules that body cells need to survive and function properly. [20]

Another way of coding information in living organisms is *RNA*, which stands for ribonucleic acid. RNA has two main differences from DNA - it contains ribose sugar instead of deoxyribose sugar and uracil instead of thymine. The three-dimensional structure of RNA is not as conserved as in DNA. RNA does not form a double helix, instead, there are multiple local patterns that can be formed, such as bulges or hairpins. [25]

There are many different RNA types (mitochondrial RNA, ribosomal RNA, transfer RNA, ...) that have different roles in the body cells. For instance, some of them play an important role in protein synthesis, others can inhibit gene expression or expression of transposons. [25]

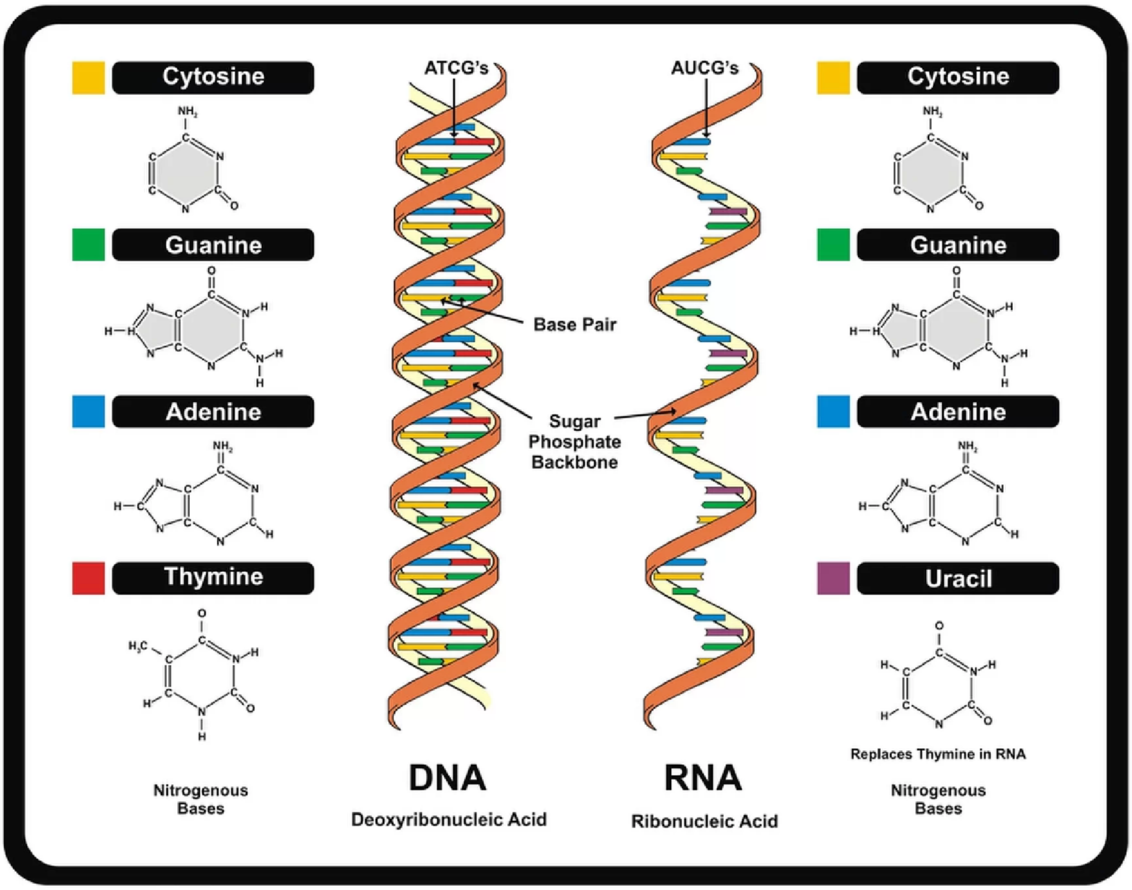

Figure 2.1: The structure of DNA and RNA. This image was taken from the article *DNA: Definition, Structure & Discovery* by Rachael Rettner [1]

## 2.2 Metagenomics

Metagenomics is a study of genetic material recovered directly from samples. It does not require isolating the DNA of individual species, neither cultivating in laboratories. [26, 32]

There are two main types of analysis. The first one is taxonomic, where the main question is: Which bacteria are present in the given sample? The second one, the main focus of this thesis, is functional: What can the bacteria in this sample do? [28]

The basic workflow of the taxonomic analysis based on 16S rRNA can be found in Figure 2.2. As input, we have all sequences from a given sample. We perform OTU picking, which clusters them in groups called OTUs. OTUs are then assigned taxonomy according to a reference database and a corresponding estimated abundance. The individual steps will be described in more detail further in this chapter. [28]

Functional analysis will be discussed in Chapter 3.

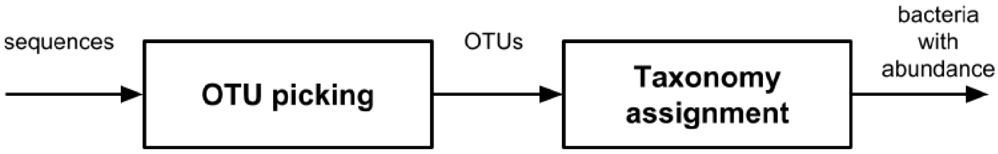

Figure 2.2: Diagram showing steps of bacterial composition analysis process

To minimize the length of the DNA sequence that must be processed in taxonomic or functional analysis, only a part of genetic information, called marker gene, is used. Marker gene needs to have the following attributes:

- It is present in every organism we want to study
- It is unique for every species
- It is similar for closely related species and different for non-related species

For bacteria, a commonly used marker gene is *16S rRNA*. It contains conserved regions, that are consistent among all species and can help identify the position of 16S rRNA in the data, and variable regions, that are different and allow us to compare the variable regions to identify species or predict functions. [21, 28]

### 2.2.1 OTU picking

The result of sequencing the sample is a lot of sequences of the marker gene, which need to be clustered into distinct groups, since some of which can belong to the same species. The sequencing methods are not perfect so the data can contain some sequencing errors, meaning the sequences of the same species are not exactly identical. In reality, sequences with 95%-99% similarity are assigned in one cluster. [28]

These distinct groups are called *OTU - operational taxonomic units*. When we lack named species corresponding to a particular variant of the marker gene, OTUs are often used instead of species, even though they are not species per se. Commonly used algorithm for OTU picking is uclust, which can be found in a tool for microbe analysis called Qiime [12, 17]. Qiime offers three different strategies:

- **de-novo**: sequences are clustered against each other, without any external reference
- **closed-reference**: sequences are clustered against a reference database, the ones without a corresponding reference record are discarded
- **open-reference**: first sequences are clustered against a reference database, then the ones without a corresponding reference are clustered de-novo

After OTU picking, Qiime is also able to assign taxonomy — names of known species — to the obtained OTUs. This is achieved by comparing OTU sequences to a reference database of 16s rRNA. Quiime is currently using Greengenes database and offers multiple methods for taxonomy assignment other than uclust, including BLAST, RDP classifier or mothur, that differ in the approach to determining the most probable taxonomy to the OTUs. [13]

## 2.3 Databases

There are multiple databases of 16S rRNA data. In this thesis, we will refer to Greengenes [4, 18] and Silva [15].

Greengenes is a database containing solely 16S rRNA sequences. The most recent version is from May 2013, and it is no longer updated. It contains experimental datasets created with the PhyloChip 16S rRNA microarray. [4, 18]

Silva databases have a wider data range, it contains also 20/23S RNA. The most recent version is from April 2018, and the next release is planned at June/July 2019. It is developed and maintained by the Microbial Genomics and Bioinformatics Research Group in Bremen, Germany. The data is available in the form of raw sequences, alignments across the whole data, precomputed phylogenetic trees and other formats. Silva databases are a part of the ARB project, which provides a graphically oriented software for sequence database handling and data analysis. [15]

# Chapter 3

# Functional profile prediction

Functional analysis is a process of finding the answer to the question „What can the bacteria in this sample do?“. We want to find the different metabolic functions of organisms in the sample, as well as to estimate their abundance - what portion of organisms have this function. [22, 28].

The basic principle is comparing OTUs to a reference database that contains the functional profile of previously studied organisms and finding the best match (usually sequencing errors are taken into account, so we are looking for 95%-99% match). For the OTUs that are not paired with a known organism, we can search for the most similar organisms and deduce the functional profile from them. [28]

Functional profiles have the form of KO identifiers with abundance in the corresponding sequences. KO identifiers refer to molecular functions and can be found in the Kegg Orthology database [6].

## 3.1 Functional prediction workflow

The workflow of functional profile prediction can be seen in Figure 3.1. The input is a table, where every row contains an OTU identifier and abundances of the given OTU in the sample, and DNA sequences representing the OTUS. The usual workflow consists of looking up the representative sequences of the OTU clusters in the reference functional profile database, combining the found results and then dealing with the OTUs for which no functional profile was found. The strategy of dealing with OTUs without known functional profile differs from tool to tool and will be further discussed in section 4 and Chapter 5.

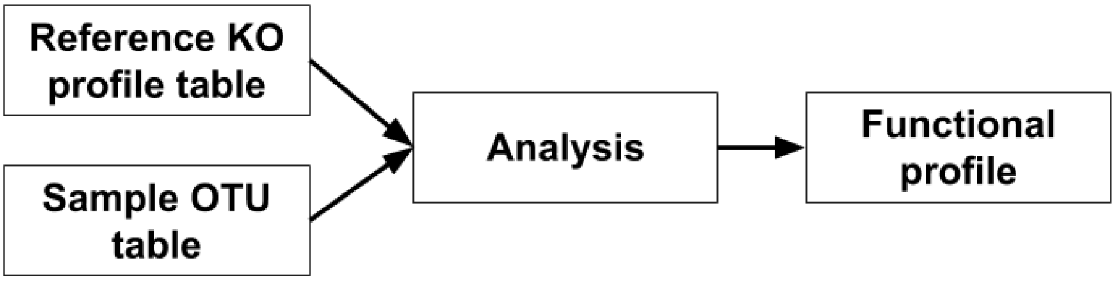

Figure 3.1: Diagram showing steps of functional profile prediction

## 3.2 Functional analysis methods

There are many different algorithms for functional analysis. In this thesis, we will focus on two distinctive groups — phylogenetic tree based methods, which are used in existing bioinformatic tools, and distance based methods, which are my original work. We will describe each group to an extent needed to understand the rest of this thesis.

### 3.2.1 Distance based algorithms

This group of algorithms is based on analyzing the representative sequences of given OTUs and comparing them to reference sequences with known functional profiles. The resulting functional profile is then inferred by the most similar reference OTUs.

To speed up the search, the similarity is usually precomputed and stored in a distance matrix. The rows and columns of the distance matrix represent the OTUs and the numbers in the matrix represent the distance of OTUs in the corresponding row and column. The smaller the distance, the most similar the OTUs are. To find the most similar OTUs to a given one, it is only needed to find the smallest values in the corresponding row or column.

To compute the similarity between OTUs, different methods can be used. Some of them simply count the number of equal characters in their alignment. Others punish the differences according to their evolutionary probability. Because of the different chemical nature of the nucleotides in RNA, certain changes in the sequences are more probable than the others. There are various matrices that express the probability of interchange between the nucleotides.

### 3.2.2 Phylogenetic tree based algorithms

This group is based on constructing a phylogenetic tree which is a graph that represents evolutionary relations between organisms. Each node of such a tree represents a species. Some of them, specifically the leaves, are living species, while the others are only estimated. The common parent of two nodes is their most probable evolutionary ancestor. [31]

An example of a phylogenetic tree can be seen in Figure 3.2. This is a tree where the lengths of individual lines between nodes represent the estimated time of evolution. If the line is short, the nodes it connects are very similar, since the time for evolution is short which implies fewer changes in the genome compared to the long lines. [31]

From the phylogenetic tree, we can estimate the evolutionary distance between different species. Using this distance, it is possible to infer a correct combination of known functional profiles for all species for which the functional profile was not found in the reference database.

The inference of unknown functional profiles can be done by finding the nearest nodes with known profiles. We can search for a certain number of known profiles, or limit the search by sequence similarity to the investigated. After we have a set of nodes with known profiles, we compute a consensus profile based on the distance to the investigated node - closer nodes have a bigger weight than the more distant ones.

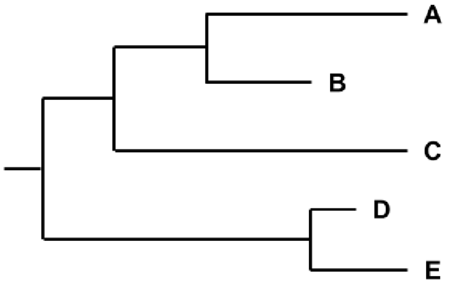

Figure 3.2: Phylogenetic tree example

There are two groups of methods for phylogenetic tree construction. Distance-matrix methods first precompute the distance matrix between all sequences and then cluster them to compute the tree so that the distance between clustered nodes is the smallest possible. Examples are Neighbor-joining and UPGMA. The second group of methods is character based. It looks directly at the sequences and tries to explain the changes between sequences representing different species. This group includes Parsimony methods and Maximal likelihood method. [29]

A simple way to construct phylogenetic trees is to use MEGA [23] — Molecular Evolutionary Genetics Analysis — which is a software that offers various sequence analysis. The trees were computed from multiple sequence alignment of sequences from Greengenes that have a known functional profile. To compare their accuracy of different types of phylogenetic trees, I have constructed all types available in this tool:

- **Neighbour joining** — this method uses bottom-up clustering based on distance matrix between all sequences. In each step, the nodes that are closest to each other and at the same time farthest from the rest of the sequences are clustered into a higher-level cluster. [24]
- **UPGMA (unweighted pair group method with arithmetic mean)** — this method also uses the distance matrix. In each step, it clusters the nodes that are closest to each other. [24]
- **Maximal likelihood** — this method uses a mathematical model based to build the tree. From all the possibilities, it chooses the tree that most probably explains the observed changes between sequences. [24]
- **Minimal evolution** — in this method, the tree is built with the fewest possible changes required to explain the differences in the observed data. The evaluation of the changes is based on giving a score to each branch in the tree. [24]
- **Maximal parsimony** — this method considers the shortest possible tree that explains the given data the best one. [24]

## 3.3 Linear Regression

One of the new approaches I implemented in this thesis uses linear regression to estimate the unknown profiles. The basic idea is to use the known profiles to create a model, and then use the model to predict unknown profiles. Therefore, I include some basic theory of linear regression.

The aim of linear regression is to build a statistical model which will reflect the relationship that may exist between explanatory variables and a scalar response (also called the dependent variable). For multiple explanatory variables, the resulting model is called a multiple linear regression model. The relationships are modeled using linear predictor functions whose unknown parameters are estimated from the data. [27]

The statistical model gained by linear regression can be described by an equation which can be seen below. The variable $y$ is the dependent variable, $x_i$ are the explanatory variables and $\beta_i$ are the coefficient we want to compute by fitting the model to the data. [27]

$$y = \beta_1 x_1 + \beta_1 x_2 + ... + \beta_n x_n + \epsilon \tag{3.1}$$

To find the $\beta$ coefficients, the least square method is often used, which means that the overall solution minimizes the sum of the squares of the errors made in the results of every single equation. [27]

# Chapter 4

# Existing tools

In this section, we will discuss existing tools for predicting the abundances of bacterial functional molecules. From the many tools available I have chosen PiCRUST, Tax4Fun, and Paprica, as they are well-known and widely used. Other tools are also available, but they do not have such a reputation and acceptance in the scientific community.

We will discuss how each tool computes KO profile, what method is used for dealing with OTUs with unknown KO profiles, used reference database, and advantages and disadvantages. PiCRUST is described in most detail since the tool created in this thesis is inspired by it.

In the last part of this Chapter, we will summarize and compare the existing tools and advocate the need for a new tool.

## 4.1 PiCRUST

Picrust is short for phylogenetic investigation of communities by reconstruction of unobserved states. It is a bioinformatics software package implemented in Python and R, freely available under the GPL [19, 11].

The work-flow of Picrust can be seen in Figure 4.1. It can be divided into two parts, Gene content inference, and Metagenome inference. Picrust uses the Greengenes database, specifically versions 13.5 and 18may2012. Using Greengenes is the biggest disadvantage of this tool - Greengenes is an old database, that is no longer updated, which makes results acquired using Picrust also out-of-date. Despite this fact, Picrust is still widely used in many bioinformatics projects.

### 4.1.1 Gene content inference

In this step, Picrust takes the whole reference tree from Greengenes and precomputes the KO profiles. The result is a KO profile for every bacteria in Greengenes. This step is independent of the sample, so it only needs to be calculated once. The creators of Picrust precalculated the data for Greengenes versions 13.5 and 18may2012 and then published the resulting data, which can be downloaded on Picrust website.

To predict the unknown functional profiles, gene content table from IMG is used, which is a table containing functional profiles for known genomes. The reference OTU tree is compared with the gene content table and sequences with the unknown functional profile are identified. Then an ancestral state reconstruction algorithm is used to create a phylogenetic tree featuring all OTUs from the reference tree. For OTUs with an unknown

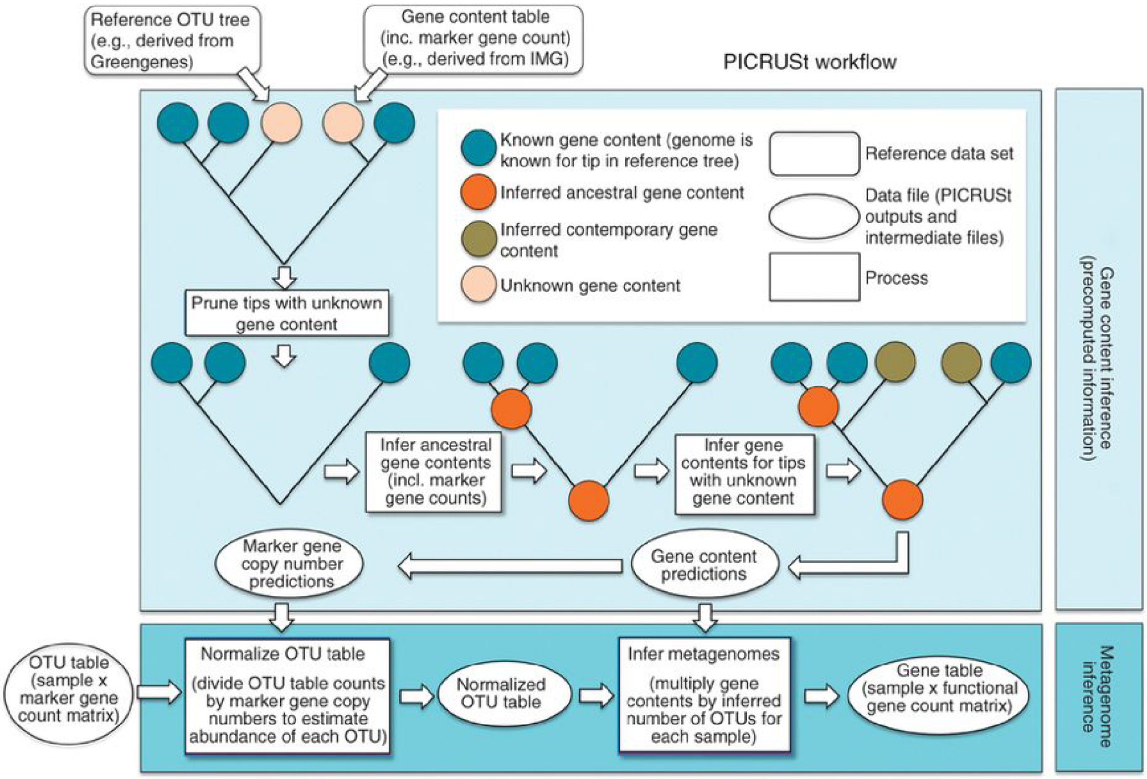
Reference OTU tree (e.g., derived from Greengenes)
Gene content table (inc. marker gene count) (e.g., derived from IMG)
PICRUSt workflow
Known gene content (genome is known for tip in reference tree)
Inferred ancestral gene content
Inferred contemporary gene content
Unknown gene content
Reference data set
Data file (PICRUSt outputs and intermediate files)
Process
Prune tips with unknown gene content
Infer ancestral gene contents (incl. marker gene counts)
Infer gene contents for tips with unknown gene content
Marker gene copy number predictions
Gene content predictions
Gene content inference (precomputed information)
OTU table (sample x marker gene count matrix)
Normalize OTU table (divide OTU table counts by marker gene copy numbers to estimate abundance of each OTU)
Normalized OTU table
Infer metagenomes (multiply gene contents by inferred number of OTUs for each sample)
Gene table (sample x functional gene count matrix)
Metagenome inference

Figure 4.1: Diagram showing the workflow of two main use-cases in a tool for functional profile prediction, Picrust. This picture is originally from the paper *Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences* by Morgan G. I. Langille [19]

functional profile, an estimated profile is computed, using the position of the given OTU in the phylogenetic tree and the closest OTUs with a known functional profile.

Although Picrust website features instructions for running Gene content inference with data from any user-desired database, in reality, executing all the steps of the instructions is difficult and time-consuming. The creators of Picrust indirectly acknowledged it by releasing Picrust 2 [10], that is different from Picrust mainly in allowing different reference databases.

### 4.1.2 Metagenome inference

This steps takes an user-provided table of OTUs, and using the gene content table from the previous step, predicts metagenomic content of the given sample. The prediction is done by summing up the functional profiles (obtained in the previous step) corresponding to OTUs in the input table while taking into account their abundance.

The provided OTUs must be closed-reference picked against the desired version of Greengenes since on this level Picrust cannot deal with OTUs with unknown functional profiles. In a case the input table was not close-reference picked, Picrust offers a script that will fix the input table by removing all OTUs that are not featured in the precomputed table.

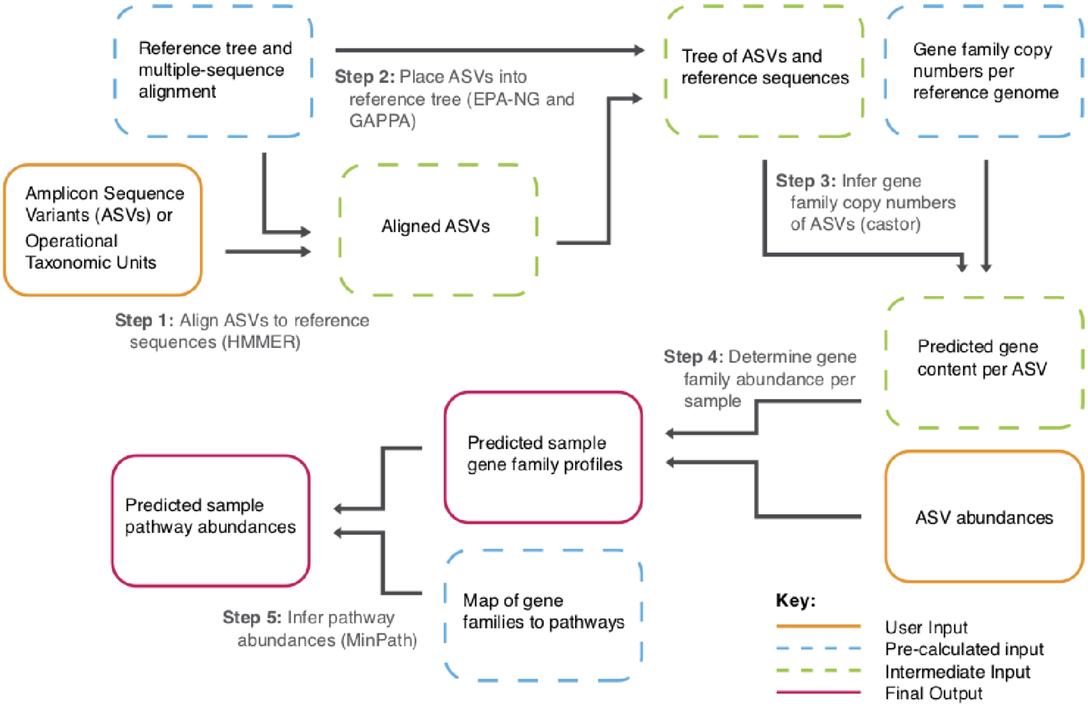
Reference tree and multiple-sequence alignment
Step 2: Place ASVs into reference tree (EPA-NG and GAPPA)
Tree of ASVs and reference sequences
Gene family copy numbers per reference genome
Amplicon Sequence Variants (ASVs) or Operational Taxonomic Units
Aligned ASVs
Step 3: Infer gene family copy numbers of ASVs (castor)
Step 1: Align ASVs to reference sequences (HMMER)
Predicted gene content per ASV
Step 4: Determine gene family abundance per sample
Predicted sample gene family profiles
Predicted sample pathway abundances
ASV abundances
Map of gene families to pathways
Step 5: Infer pathway abundances (MinPath)
Key:
User Input
Pre-calculated input
Intermediate Input
Final Output

Figure 4.2: Diagram showing the workflow of Picrust2. This picture is originally from the wiki of Picrust2 [10]

### 4.1.3 Summary

Advantages of Picrust are transparency and good documentation. Disadvantages are the usage of the Greengenes database, the difficulties of using a different reference database and the requirement that input data must be closed-reference picked against Greengenes.

## 4.2 Picrust2

Picrust2 is a newer version of Picrust. As of now, it is still in Beta version. It offers the same basic functions as Picrust, but it is much easier to use with user-provided reference data. [10]

The documentation of Picrust2 is much more detailed and thorough as in Picrust. The data flow can be seen in Figure 4.2. Steps 1 and 2 correspond with Picrust gene content inference and Steps 3 to 5 with metagenome inference. The steps can be run individually or as a whole pipeline. When possible, Picrust2 allows users to provide own reference data or choose the computation method and set its parameters simply by command line arguments.

On the other hand, Picrust2 is more resource-consuming than Picrust. To run the first step of Picrust2 pipeline, alignment and tree creation, at least 16GB of RAM is needed, and even that may not be enough, dependent on the input data.

## 4.3 Tax4Fun

Tax4Fun is an open-source package for R using Silva database. It can predict functional capabilities of a metagenome, as well as a metabolic profile [30, 16].

Tax4Fun uses a different strategy for OTUs with unknown profiles than Picrust. Where Picrust builds the ancestral tree based on the nearest neighbor method, Tax4Fun adds sequence similarity check. Since nearest neighbor in a tree always exists, Picrust links all OTUs, even if their distances in the tree are large. Tax4Fun links the nearest neighbors, only if the sequences have certain minimum sequence similarity, and then applies a linear transformation. Because of this, Tax4Fun should be more efficient for metagenomes with a large proportion of not well-characterized bacteria.

The results of the comparison between Picrust and Tax4Fun, published in the paper *Tax4Fun: Predicting functional profiles from metagenomic 16S rRNA data* [30], indicate that Tax4Fun is more accurate. Unfortunately, since the two tools use different reference databases, this could be caused by the better quality of Silva database data compared to Greengenes data. To truly prove that method used within Tax4Fun is more efficient, comparison on the same database would be needed. As reasoned in 4.1.1, this is currently not possible.

An advantage of Tax4Fun over Picrust is the implementation in R. While Picrust needs to be installed and run on Linux-based system, Tax4Fun is an R package so it can be used on any operating system with R installed. R also has a simple and easy-to-use user interface, RStudio, and is more popular among non-informatics than Python. The reference data of Tax4Fun are from Silva database, which is more up-to-date than Greengenes in PiCRUST.

## 4.4 Paprica

Paprica is another Python library. To create a functional profile, Paprica uses phylogenetic placement instead of an OTU-based approach [7, 8].

In the OTU-based approach, the reads from a sample with a certain similarity are clustered into one OTU. The gained OTUs are then compared to OTUs of known metagenomes and the functional profile is inferred. The disadvantage of this method is the clustering into OTUs since similar 16S rRNA sequences do not always imply the same functional profile. For example, many different genomes with different functional profiles have been sequenced for Escherichia coli. Although we have knowledge about different E-coli genomes and the corresponding functional profiles, the 16S rRNA gene of all variants is very similar so the OTU-based approach will treat different E-coli variants as one.

In Paprica, there is a precomputed phylogenetic reference tree of 16S rRNA genes from each completed genome. Internal nodes contain a consensus genome which takes into account all the child nodes.

The input of Paprica is not a set of OTUs as in the previous tools, but raw non-clustered sequences. To find a corresponding functional profile, Paprica tries to find the most similar sequences in the reference tree.

The advantage of this approach is that the resolution changes based on the sequence coverage of the reference tree. For the well-studied organisms, we are able to distinguish variants that are very similar, while also having good results for unknown organisms. The disadvantage is the input format incompatible with other popular prediction tools.

## 4.5 Summary

The goal of this thesis was to either create a consensus tool from existing bioinformatic tools or to create a new one that will be inspired by them. Based on the research of PiCRUST, Tax4Fun, and Paprica, I came to the conclusion that creating a consensus tool that will use them would be more difficult than creating a new tool. PiCRUST is incompatible with the other two because of a different reference database and the inability to deal with OTUs that are not closed-reference picked against Greengenes. Paprica is incompatible because of a different required input format. The usual input for functional profile prediction is a list of OTUs with their abundances, but Paprica requires the reads before OTU clustering. If we wanted to use Paprica in the consensus tools, we would either have to find a way to automatically download sequences corresponding to the OTUs in the input file, or the input format would have to be the reads and we would have to implement OTU picking on them to be able to use PiCRUST and Tax4Fun.

Since two of the three studied tools have a feature that makes them incompatible with the others, I concluded that it will be easier and more efficient to create a new tool and focus on implementing and comparing different prediction methods than to try to make PiCRUST, Tax4Fun, and Paprica compatible.

To not disregard the research of these tools, I will use the acquired information in the new tool created in this thesis. The PiCRUST dataflow is easy to understand and makes sense so the dataflow of the new tool will be inspired by it. Different ancestral state reconstruction algorithms implemented in PiCRUST will also serve as an inspiration in our own ancestral state reconstruction. Tax4Fun is easy to use and has the most transparent output format, which is a table of KO with the estimated abundance in the sample. The same output format will be used in the new tool. Paprica introduces a wholly different point of view to functional profile prediction. It would be interesting to implement some method inspired by Paprica and then compare the results with other, OTU based methods. Even a comparison with other phylogenetic tree based methods might be valuable.

# Chapter 5

# Design of the created tool

The goal of this thesis is to create a tool for functional profile prediction. It could have implemented a consensus of existing tools described in 4, but after a detailed study of these tools, we determined that creating a consensus tool is not possible. The detailed reasoning for this can be found in 4.5.

Therefore, we decided to build a new tool. It should be usable with any user-provided reference database and implement different methods for dealing with OTUs with unknown functional profiles. The goal is to design and implement various of these methods, based either on sequence similarity or on a phylogenetic tree, evaluate them on the various quantity of OTUs with unknown profiles in the sample, and determine which method is the best, alternatively, which method is the best for certain species of bacteria. The source code of the tool is available in GitHub repository [3] and on the storage medium attached to this thesis.

## 5.1 Implementation

The tool consists of multiple modules, which can be seen in Figure 5.1. Arrows indicate the data flow in the tool.

The green modules are data sources:

- **KO profiles**: a table of known species with corresponding KO profiles. The data used in the implementation and evaluation is from Greengenes, but a switch with a different data table is possible.

- **OTU similarity**: this is a source of similarity between OTUs with known and with unknown functional profiles. It can either be a similarity matrix or a phylogenetic tree, or anything else, that somehow represents the similarity between OTUs. We aim to experiment with a similarity matrix based on different scoring matrices and different methods for phylogenetic tree construction.

The yellow modules will be the same for every method for dealing with unknown OTUs:

- **Input parser**: this module extracts OTU identifiers and abundances in the given sample from the input file.

- **Known profile resolver**: this module takes the OTU identifiers obtained from input parser and tries to find them in the KO profiles table. For the found OTUs it creates a KO profile respecting the given abundances and sends it to Output generator. The

OTUs which are not found in the KO profiles table is forwarded to Unknown profile resolver.

- **Output generator**: this module takes the two KO profiles, combines them into one, and generates the output file.

The pink module will deal with OTUs with unknown KO profiles:

- **Unknown profile resolver**: this module gets the OTU identifiers, that were not found in the KO profiles table. By using one of the methods described in 6 it finds the most similar OTUs with known functional profiles. Finally, it creates a KO profile respecting the similarity and the abundance in the given sample and sends it to the Output generator.

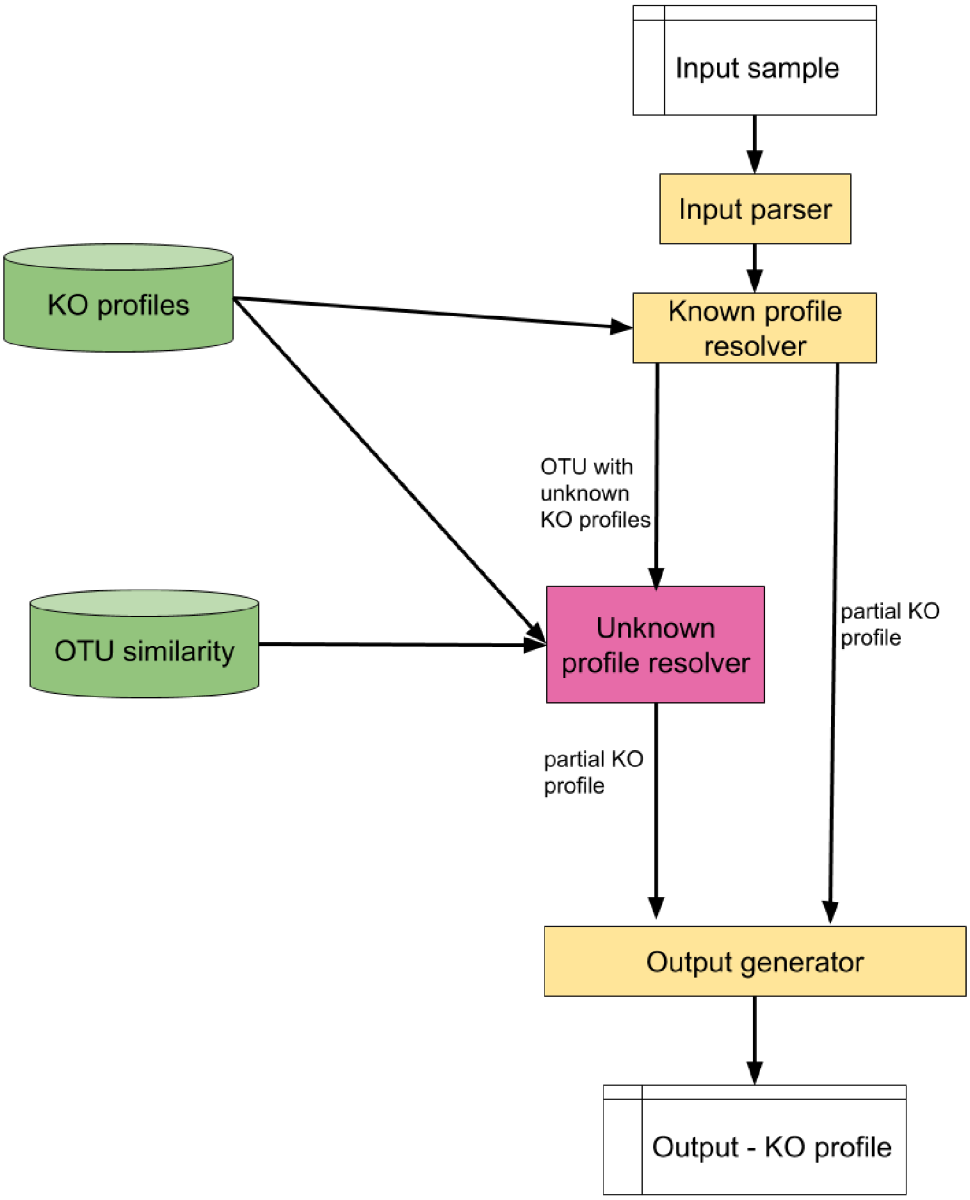

Figure 5.1: New tool design. White rectangles show the input and output of the tool, green color denotes the data sources, yellow rectangles are the processes which are the same in every unknown functional profile resolving algorithm and the pink rectangle is the unknown profile resolver, which implements various methods for functional profile prediction. Where needed, input and output data of a process are described next to the corresponding arrows.

## Chapter 6

# Evaluation of unknown reference profiles prediction methods

In this section, we will describe the method for profile prediction that is implemented in the created tool. We will discuss what settings have been tried and what were the reasons behind them and the expected results.

Testing data — known functional profiles — have been retrieved from the IMG database [5]. IMG stands for Integrated Microbial Genomes and it contains already sequenced and annotated microbial data.

## 6.1 Evaluation framework

For the purpose of testing the accuracy of the implemented methods, I have also implemented an evaluation framework. We can see the logic of evaluation in Figure 6.1. I use reference KO profiles table as an input, from which I generate the samples which will be tested. Then I compute the expected results using the reference profiles table. This result is saved. After that, the tool is run on the same samples. The results from the tool are correlated with the expected results.

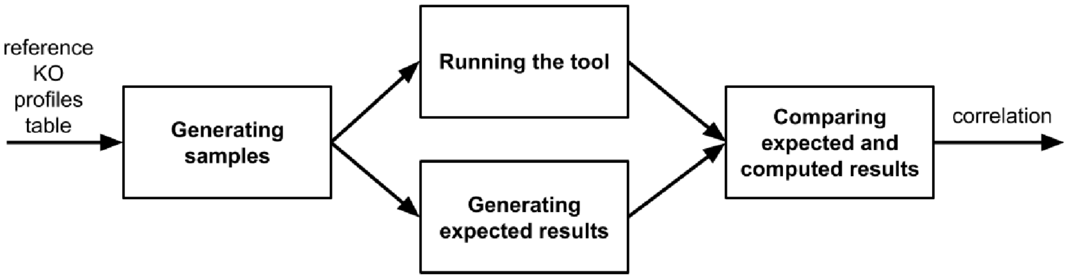
reference KO profiles table
Generating samples
Running the tool
Generating expected results
Comparing expected and computed results
correlation

Figure 6.1: Workflow of the framework used to test the created tool

## 6.2 Evaluation strategy

For each method, the accuracy was tested on 10 artificial samples. To simulate missing functional profiles, a part of reference KO profile table was randomly deleted. The ratio of the deleted table was incrementally increased, from 0% to 90%, to see how much the accuracy drops with more profiles missing. Since the deletion from the reference table was randomized, this step was performed 10 times.

To summarize, for each ratio of missing functional profiles, we performed 100 tests.

The correlation between the expected and the computed result is computed as the Pearson Product-Moment Correlation, which shows the linear association between two vectors. The formula for estimation of the Pearson coefficient can be seen below, where $r$ is the coefficient, $x$, and $y$ are the vectors and $n$ is the size of the vector. The values of the coefficients can range from -1 to 1, where 0 means no association between the vectors, values bigger than 0 show positive association and values smaller than 0 show negative association. The proximity to -1 and 1 show the strength of the association. Generally, values bigger than 0.5 are considered a strong association. [9]

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}}$$

## 6.3 Distance based methods

This group of methods was created for this thesis and as of now none of them is used in bioinformatic tools described in 4. The methods are based on finding OTUs with the most similar sequences that have a known functional profile. Then we take the corresponding profiles and compute the unknown profile as their average.

The simplest way to get the similarity between sequences is to count the number of the same characters in their aligned sequences. However, this method is not biologically accurate, since the probability of exchange of different nucleotide pairs is not the same. Therefore, I have experimented with using scoring matrices, that penalize the differences between sequences more accurately.

A comparison between arithmetic and average weighted by similarity score has also been part of the experiments. The weighted average should be more precise since the more similar OTUs will have a bigger influence on the resulting KO profile. On the other hand, the difference between the similarity score of the most similar OTUs is not big — sequences usually differ only in a couple of bases — so the difference in the weight might be minimal and the results could strongly copy the non-weighted average.

Another thing to experiment with is the number of required similar sequences. Fewer sequences ensure, that the reference OTUs are more related to the original one, but more sequences can bring variety and include some functional traits that are not present in the smaller selection.

### Evaluation

The initial evaluation was performed on $N = 10$ and $N = 4$. As we can see in Figure 6.2, if more than 50% of the OTUs have known functional profiles, the correlation of correct and predicted profile is more than 0.9. After the ratio of the known profile drops under 40%, the accuracy falls and the variance of the correlations rises. The rise in the variance

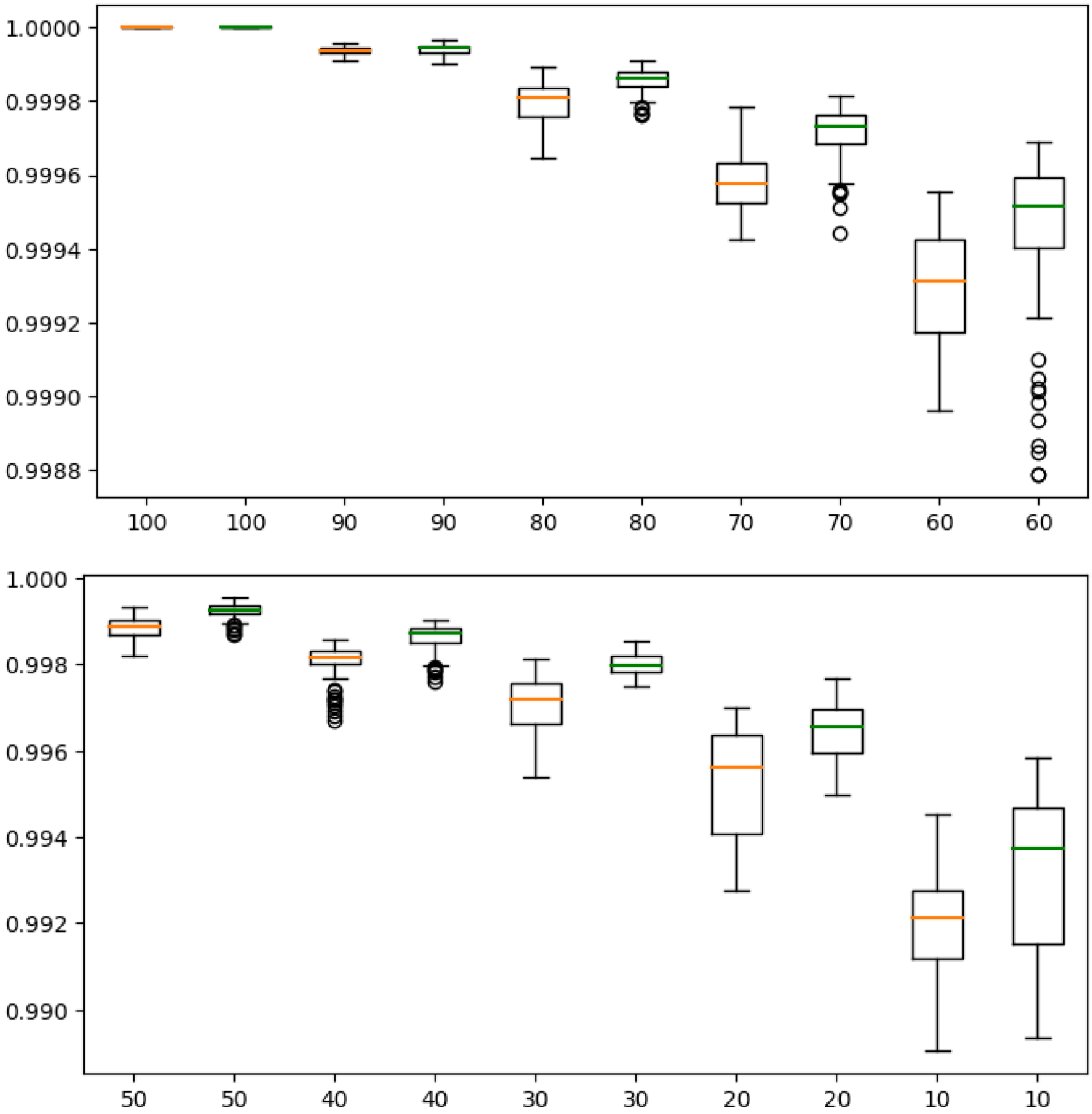

Figure 6.2: Results of the experiments with unknown functional profile estimation. The orange data show correlations of an average of 10 most similar OTU profiles, while the green data shows the same result for 4 similar OTUs. The y-axis shows the correlation between expected and computed results. The x-axis shows the ratio of sequences that had a known functional profile.

suggests that with such a small ratio of known profiles, the result is highly dependent on which profiles stay in the reference table.

As for the comparison between an average of 4 and 10 similar OTUs, 4 similar OTUs prove to be more accurate. The difference is not big with a higher ratio of known OTUs, but under 40%, 4 similar OTUs have a 0.002 better correlation. This indicates that it might be to search for fewer closely related OTUs when estimating an unknown functional profile.

To confirm this hypothesis, I have performed more tests with different numbers of similar OTUs from which the missing profiles are inferred. Results of these experiments can be seen in Figure 6.3. The chart shows only experiments where 30 to 10 percent of the OTUs had known profiles because when over 30% of the profiles are known, all the variants have high accuracy and the differences between them are not so significant.

As we can see, the hypothesis that fewer similar sequences mean more accurate results were confirmed. The best option seems to be taking only one closest profile (for 10% of the reference table having known profiles) or counting the average of two most similar sequences (for more than 10% of profiles known). The biological reasoning behind this is that the more sequences we take, the more different they are from the original one. However, this reasoning is not very accurate, as the similarity of sequences is dependent on the reference data. For some species of bacteria, we have more studied variants with known profiles, while for others the number is smaller. The solution might be limiting not the exact number of similar sequences, but limiting the similarity by a certain threshold, or computing weighted average of the profiles. These variants I also experimented with and they will be described later.

## 6.4 Random method

Overall, the acquired result from the first implemented the distance-based method described in 6.3 was better than was expected. When only 10% of OTUs in a sample have known functional profiles, the correlations are still higher than 0.99. This may be caused by **common metabolic functions**: each bacteria must have basic functions for translation, transcription, and processing of common metabolites. This is the part of the functional profile that is the same for every species of bacteria, independent on the sample. A number of KO specific for a certain species of bacteria is much smaller so it might not have that much of an effect on the correlation.

To test whether the average method is really that successful or if the accurate results are caused by the wrong evaluation method, I have implemented a random method. For every unknown functional profile, it takes 4 random OTUs with known functional profile and computes their average. The number 4 was chosen based on experiments in 6.3. More OTUs were proven to cause a drop in accuracy, while less could also lead to worse accuracy, as averaging more profiles can help smooth specific features of the randomly chosen sequences. The results of this experiment can be seen in Figure 6.4. As we can see, the median of the random method for only 10% of known functional profiles has a correlation of more than 0.94 with the expected results.

For more accountable results, I have tried to eliminate the effect of common metabolic functions. In this experiment, while counting the correlation of the KO profiles, KOs that are present in a certain number of OTUs, for instance in more than 95%, are ignored. This way the correlation will be counted only on the most specific KOs for all species, which could provide more accurate results than comparing all bacterial functions, including the ones common for every species.

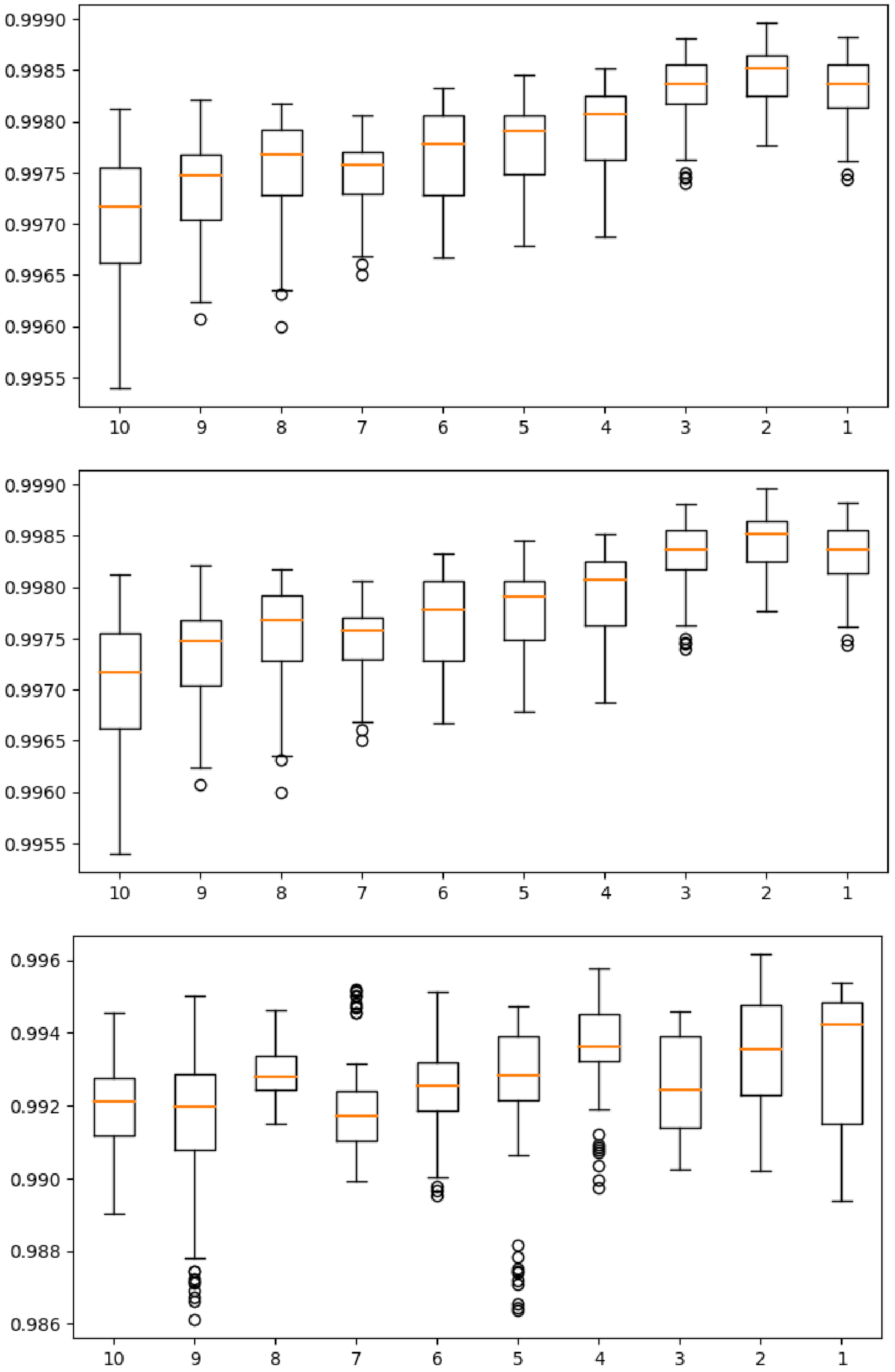

Figure 6.3: Results of the experiments with number of similar sequences that are taken into account. The y-axis shows correlation between expected and computed results. The x-axis shows the number of sequences that were used to compute the functional profile. The figures represent experiments when the ratio of 30%, 20% and 10% of the KO profiles was known, respectively.

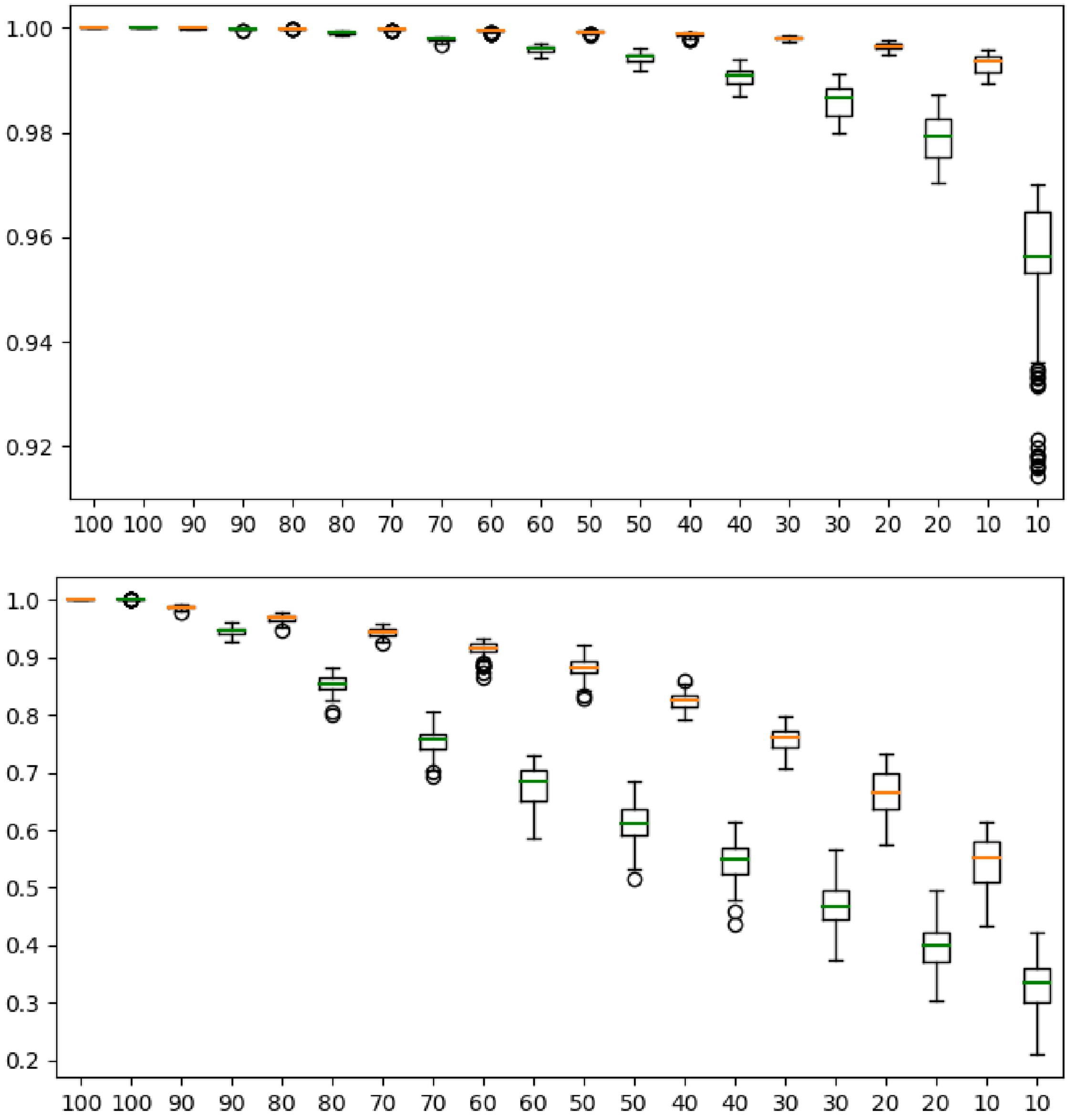

Figure 6.4: Comparison of average and random based methods for functional prediction. The upper chart shows evaluation on all bacterial functions, the graph below on the rarest 1% of bacterial functions. Orange data represent the average method and green data the random method. The x-axis represents the ratio of known profiles in the sample, the y-axis represents the correlation between expected and computed result.

We can see the results in Figure 6.4. Both charts show results from the same experiment, but in the second chart, only KOs which are present in less than 1% of the items in the reference table was included in the evaluation. We can see that the correlation between the expected and computed results significantly dropped in both average and random method. This confirms the hypothesis that the classic evaluation with all the functions is strongly influenced by common metabolic functions and does not reflect the accuracy of more specific bacterial functions.

The advantage of evaluation on more specific KOs is that the difference between random and average method is much more significant — with only 10% of the functional profiles known, the difference is more than 0.2. This is important because evaluation in which even the random method achieves the correlation values of 0.99 can be misleading and result in unfair comparisons between different methods.

To eliminate the effect of common metabolic functions on the results, all the following methods for functional profile prediction will feature both the results for all KOs and for the most specific KOs. My experiments with different thresholds of specificity indicated that the difference between methods is most visible when only KOs present in less than 1% of the reference data (which is still more than 1 100 KOs) are correlated. Therefore, I will further use 1% as the default KO threshold.

The effect on common metabolic functions on the evaluations is not always a problem. In various research areas, for instance, gut bacteria, we usually want to know if it all the basic functions are present in high enough abundance, so the focus of our interest is actually the common metabolic functions. As the gut microbiome contains thousands of species, the more specific functions might not be well documented so this information would not be that significant.

I will be using the thresholds because the focus of this thesis is to compare different methods and the results acquired with thresholds make comparing easier and more intuitive by making the differences in correlations bigger.

## 6.5 Variants of distance-based methods

This section contains experiments with different variants of distance-based methods.

### 6.5.1 Distance matrix

The first modification I have tried was to use a transition/transversion scoring matrix for nucleotides for computing the distance between OTU pairs. I expected the results to be better than normal average described before. Surprisingly, the results were worse. This was not expected, as counting the distance between OTUs based on matrix should be more biologically accurate than just counting the number of corresponding symbols in sequences. The result can have multiple reasons:

- **wrong scoring matrix** — there are various types of scoring matrices. I have used the transition/transversion scoring matrix, which does not yield good results, but other matrices might be more successful.

- **wrong score for gaps** — there are multiple ways to deal with gaps in the sequence alignment. They can have no effect on the score, they can be punished and it is possible to have a different score for smaller and larger gaps. I used the uniform punishing of each gap, which might have been the wrong approach.

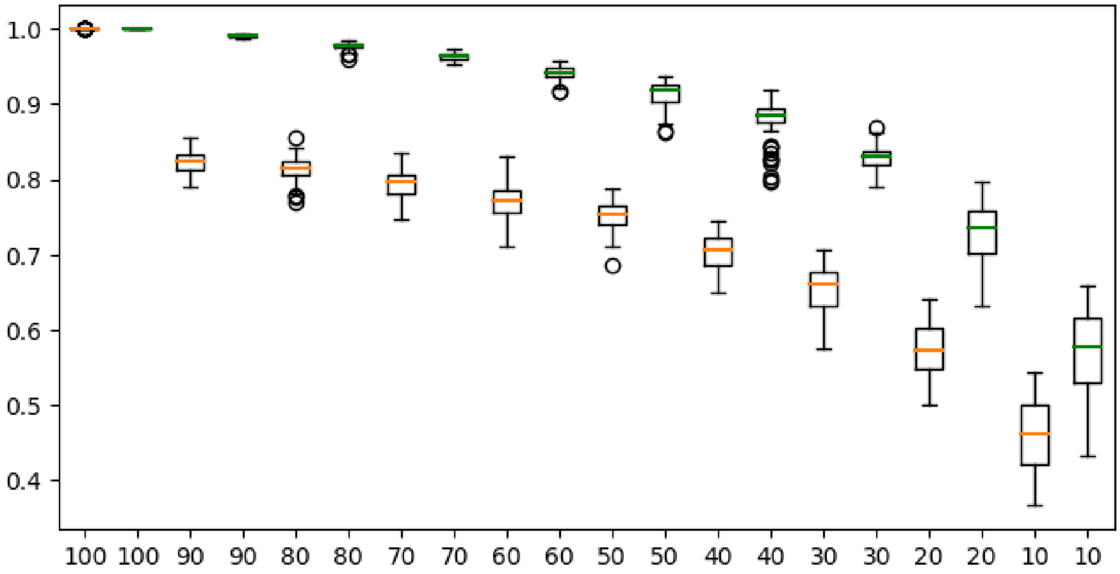

Figure 6.5: Results of two methods using scoring matrices for functional profile prediction, evaluated for the rarest functions. The orange data represent simple transition/transversion matrix, the green data matrix using the Tamura-Nei model. The x-axis represents the ratio of known profiles in the sample, the y-axis represents the correlation between expected and computed result.

- **global alignment** — the distance is computed from the multiple sequence alignment. It could be better to get raw sequences for each pair of OTUs, align them and count the score from there, but this would be more time-consuming.

After the first, non-successful experiment, I have decided to try one more scoring matrix. I have computed it using the MEGA software [23] and used transition/transversion matrix in combination with the Tamura-Nei model. This time, the results were significantly better, as can be seen in Figure 6.5. The accuracy is also higher in comparison with the method using only the number of corresponding nucleotides as the distance metric. Therefore, we can conclude that the first evaluated scoring matrix was simply not suitable for this prediction, but the idea to respect the biologic basis of the nucleotides is correct.

### 6.5.2 Weighted average

This modification is based on not computing a simple average of similar sequences, but a weighted one, based on how similar the sequences are. Although the best number of similar OUTs to estimate from proved to be 2, I have included 4 most similar OTUs in the weighted average, since less would be too little to see the effect of the weighted average, since the distance of two most similar sequences is usually much alike, as they differ only in a couple of bases. According to Figure 6.3, using more than 4 sequences leads to worse accuracy. As we can see in Figure 6.6, the accuracy of the weighted average method is slightly better than with the normal average method, but overall the curve defined by the medians is very similar.

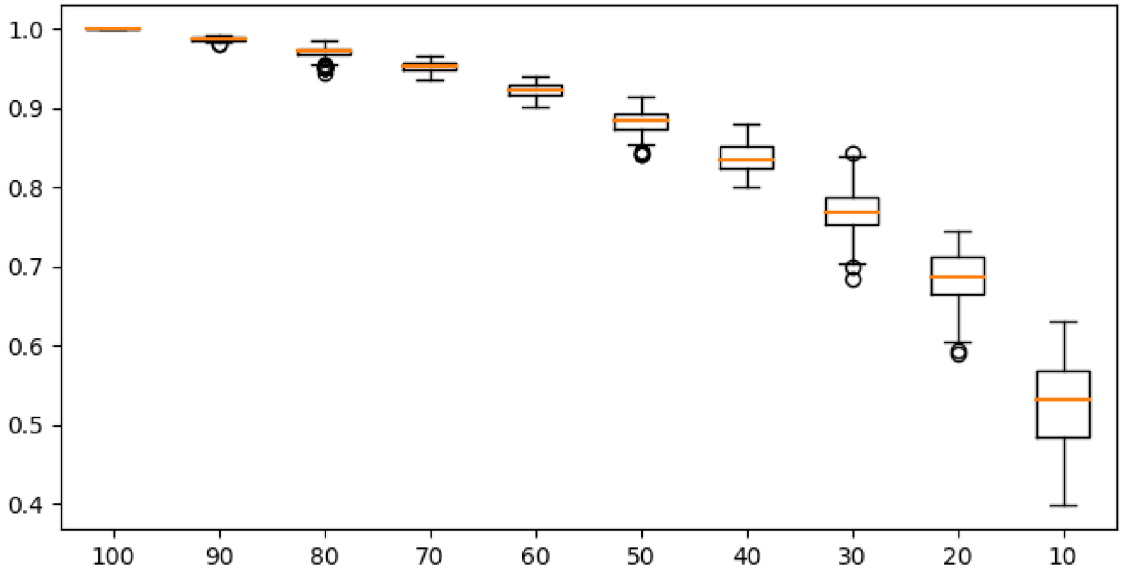

Figure 6.6: Results of weighted average method for functional profile prediction, evaluated for the rarest functions. The x-axis represents the ratio of known profiles in the sample, the y-axis represents the correlation between expected and computed result.

### 6.5.3 Limit by ratio

In this method, the goal is not to find $N$ sequences similar to he one with unknown profile, but to find all sequences with similarity bigger than a given threshold. The idea behind this is that if we have OTU whose closest 4 OTUs bear the similarity of 90%, 89%, 89%, and 88%, and for another OTU the closest with the similarity of 88%, 40%, 35%, and 30%, in the first case it makes sense to include all 4 sequences in the estimation. However, in the second case, the first sequence is very similar, but the others are quite different so it can be expected that including all of them will decrease the accuracy. How often the second situation arises is dependent on the quality of the reference data. If for every sequence found in the sample we have similar enough counterparts in the reference database, the results will be accurate using the normal average method. But if we do not have ideal reference data, which is the case in real life as a lot of the bacterial species are not documented, the limit by threshold makes more sense than limiting the number of similar items.

In this method, the relative similarity of two OTUs is computed as the number of identical symbols in their representative aligned sequences divided by the length of these sequences. As the source of the alignment, I used the global alignment of all Greengenes OTUs, which means that all relative similarity scores have the same divisor — the length of global alignment, which is 7682 symbols. If we require 99% identity, then 77 symbols in the global alignment can be different. To test what identity yields the best results, I have experimented with the similarity threshold of 99% and 97%.

We can see the results in Figure 6.7. The threshold of 99% seems to produce more accurate results, which confirms the hypothesis that it is better to look for less more similar OTUs than for more OTUs that are not that similar.

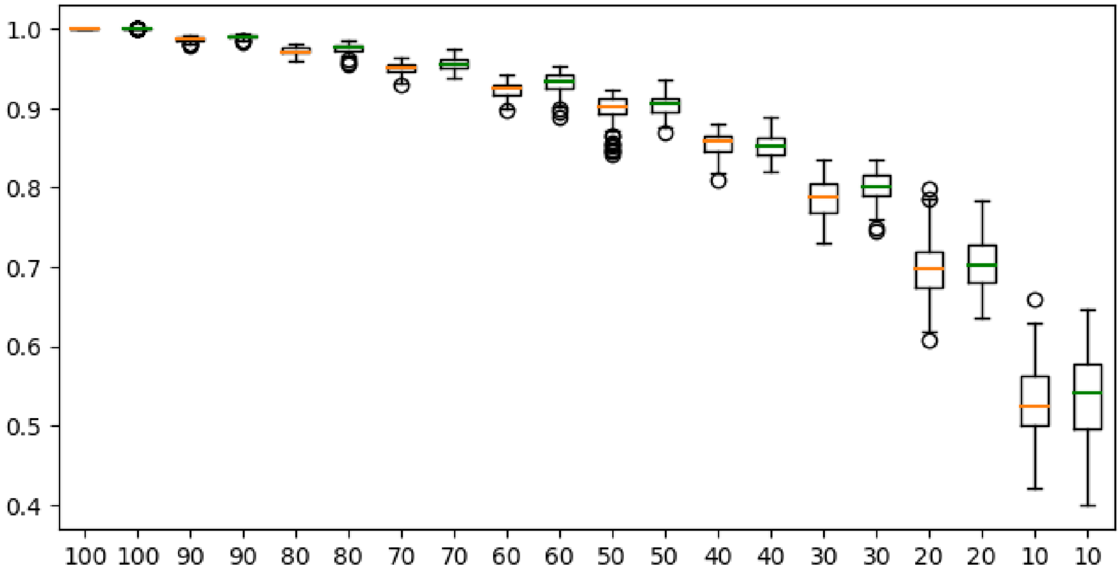

Figure 6.7: Comparison of two methods for functional profile prediction based on the similarity threshold. The green box-plot shows the threshold of 99% and the orange box-plot of 97%. The evaluation was performed on the rarest bacterial functions. The x-axis represents the ratio of known profiles in the sample, the y-axis represents the correlation between expected and computed result.

## 6.6 Aggregated results of distance based methods

We can see the aggregated results of all distance-based methods in Figure 6.8. Table 6.1 shows us the AUC metric — Area Under The Curve. Bigger AUC means a more accurate prediction method. As we can see, the random method is significantly worse than all the others, which was expected.

Next methods have minimal differences in accuracy. Normal average methods are slightly worse than threshold methods, which can be expected as they are more naive. I chose to include stats for 2 and 4 sequences in the aggregated results, as 4 is a number of sequences that are also used in the weighted average and random method, and 2 sequences produced the best results according to Figure 6.3, where I experimented with the number of sequences from 1 to 10.

| Method | AUC |
|---|---|
| random | 59.37 |
| average - 4 | 77.26 |
| average - 2 | 77.33 |
| weighted average | 77.75 |
| threshold - 97 | 78.44 |
| threshold - 99 | 78.91 |
| matrix | 80.33 |

Table 6.1: AUC — Area Under Curve metric, computed from curves in Figure 6.8

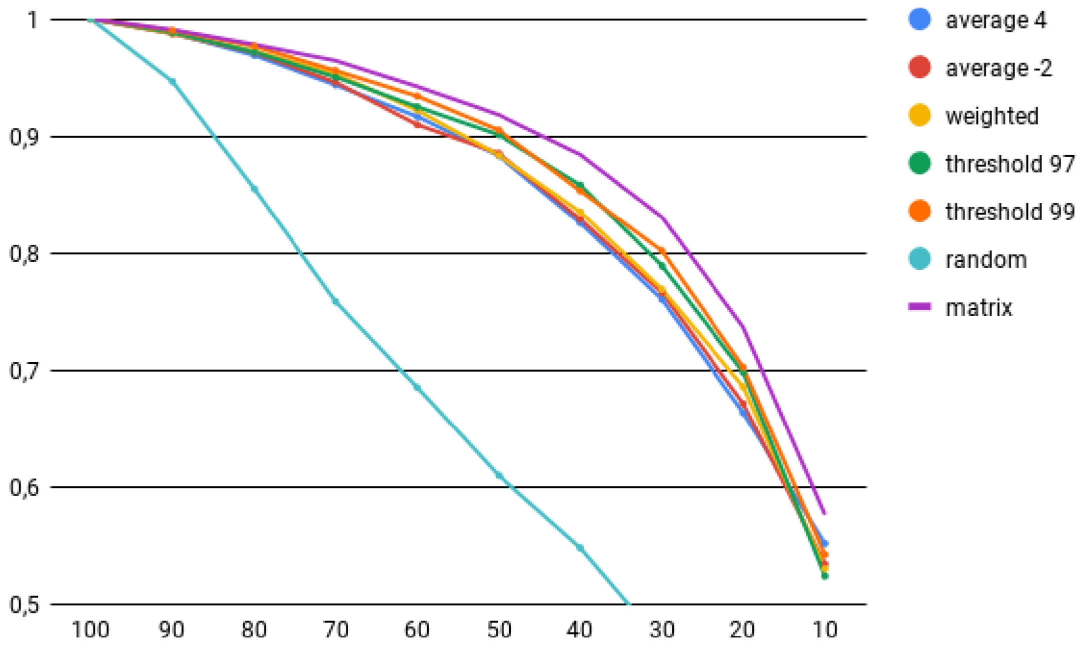

Figure 6.8: A chart comparing the accuracy of distance-based methods for functional profile prediction. The x-axis represents the ratio of known profiles in the sample, the y-axis represents the correlation between expected and computed result. Values rendered in the graph represent the median of all correlations earned in the evaluation.

The weighted average is slightly better than the normal average, as the sequences which are more different from the target OTU have less influence on the result as the more similar ones.

The threshold method proves to be quite accurate. The method respects the number of similar OTUs with the functional profile in the reference data. If there is a lot of similar OTUs, we include all of them in the result estimation. If there is less, we often end with only one or two profiles to use in prediction. Other OTUs which would influence the result estimation in normal or weighted average are not included and do not skew the estimated profile. As for ideal parameters, the best setting seems to be the threshold 99%, although the threshold 97% is almost as successful. The final setting of the threshold should be determined with the focus on the nature of the particular experiment.

The most accurate method is the one using the scoring matrix as a distance metric between sequences. This is not a surprise, as it is the only one that respects the biologic and chemical nature of the nucleotides and the probability of their exchange.

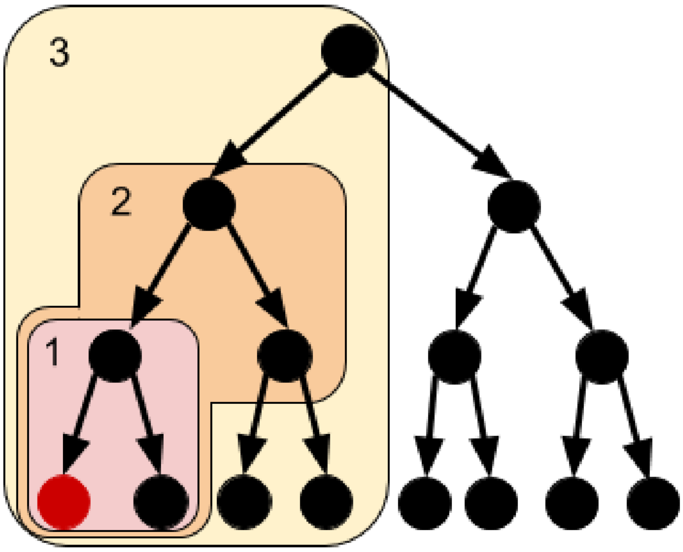

Figure 6.9: A visualization of search of the closest node with the known functional profile in a phylogenetic tree. The red circle represents the node with the unknown profile. The colored regions represent the area of the tree that is searched in each iteration. The red region is searched first, the orange region second, and the yellow region third. After that the searched area increases following the same pattern.

## 6.7 Phylogenetic tree based methods

A different approach to OTU similarity computation is a phylogenetic tree. The main idea is to compare the differences between OTUs, determine which OTUs might have a common evolutionary ancestor, and then create a phylogenetic tree using this information. The KO profile is based on the closest ancestors with known KO profiles, with respect to the distance between the tree nodes.

All the well-known existing bioinformatic tools — Picrust, Tax4Fun, and Paprica — estimate the functional profile using phylogenetic tree based methods, therefore this approach is not new and it has already been evaluated and documented. To avoid reinventing the wheel, the extent of my experiments with phylogenetic tree-based methods is not as large as with distance-based methods. I am mainly focusing on comparing different types of phylogenetic trees and the inference of functional profile from given trees is simple and straight-forward.

To create different types of trees, I have used already mentioned tool called MEGA [23]. The inference of functional profile from the phylogenetic tree I implemented is incremental. Since I compute the trees from all Greengenes sequences, OTUs with unknown functional profiles are a part of the tree as well as those with known profiles. The idea of searching in the tree can be seen in Figure 6.9. For each node with unknown profile, I first search the area in the closest proximity — the siblings. If enough similar sequences are found, the search ends. If not, the children of siblings and also siblings of the parent of the target node are searched. The area which is searched is increased in each iteration until the sufficient amount of nodes with known profiles is found.

The results of the comparison of trees can be found in Figure 6.10 and Table 6.2. We can see that the results form two distinctive groups — trees constructed using Minimal Evolution and Maximal Parsimony method are significantly worse than the other trees according to the Area Under Curve metric. The chart in Figure 6.10 also confirms this statement.

To better compare the other three types of trees (UPGMA, Neighbor-joining, and Maximal likelihood), we have to look at Figure 6.11, because while the previous two results use only the median of all correlations, box-plot shows us all computed results. The biggest differences can be seen in the variance of correlations. UPGMA and Maximal likelihood tend to have a bigger variance than Neighbor-joining and overall tend to produce more outliers. With regard to these facts, Neighbor-joining seems to be the tree constructing method that produces the most accurate results, even though the Maximal likelihood has slightly better AUC value.

Table 6.2: AUC — Area Under Curve metric, computed from curves in Figure 6.10

| **Tree type** | **AUC** |
|---|---|
| Maximal parsimony | 76.82 |
| Minimal evolution | 77.25 |
| UPGMA | 80.39 |
| Neighbour joining | 80.54 |
| Maximal likelihood | 80.72 |

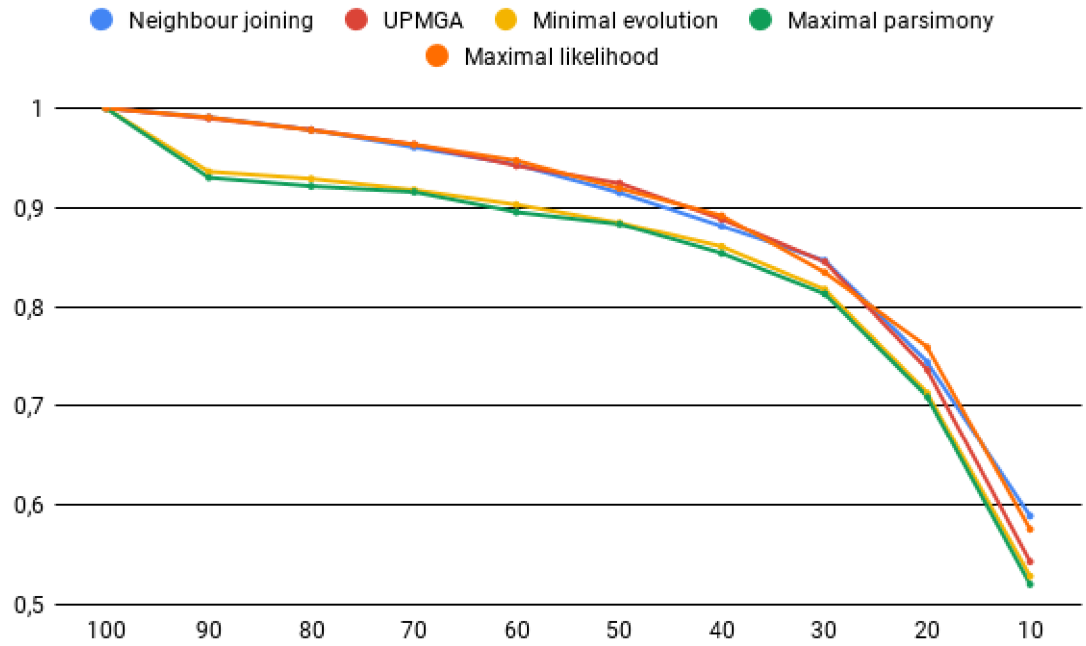

Figure 6.10: A chart comparing the accuracy of various types of phylogenetic trees used in functional profile prediction. The x-axis represents the ratio of known profiles in the sample, the y-axis represents the correlation between expected and computed result. Values rendered in the graph represent the median of all correlations earned in the evaluation.

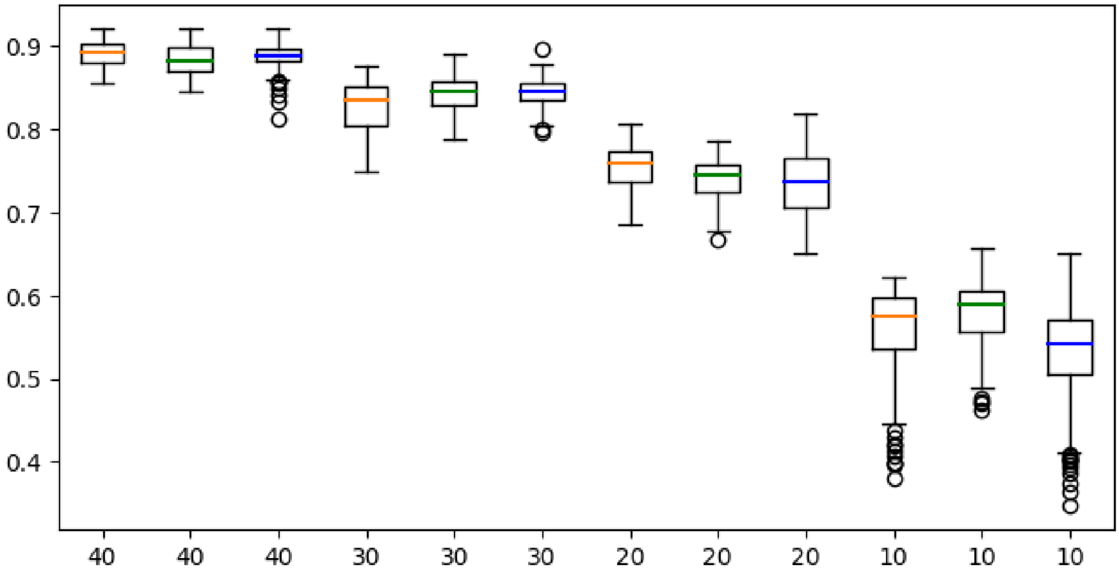

Figure 6.11: Comparison of three most successful types of trees used in my experiments with phylogenetic tree based method for functional profile prediction. The orange data represent the maximum likelihood tree, the green data the neighbor-joining tree, and the blue data the UPGMA tree. The x-axis represents the ratio of known profiles in the sample, the y-axis represents the correlation between expected and computed result.

## 6.8 Linear regression based method

This is a new method for functional profile prediction. Distance-based methods and phylogenetic tree-based methods are both based on biologic principles, but by implementing linear regression I try to bring in a more computer-science based approach.

To create linear regression models, I used Python library scikit-learn [14]. It is fast, easy to use and offers methods for model creation as well as for saving the trained model into a file and then loading and using it in prediction.

To predict functional profiles by linear regression, we have to decide what will be considered a model, and what the corresponding explanatory and dependent variables will be. To predict the whole functional profile with one model seems too ambitious. We have over 6800 KOs and predicting all of them using only one model would require a big training set and even then it might not work very well. The Greengenes database contains roughly 2500 species that have known KO profiles, which might not be enough for such a complex model. Therefore, I have decided to create one regression model for each KO. This approach can also be convenient if we needed to predict only certain KOs.

Next, we have to determine what explanatory and dependent variables will be. I have decided to infer the profile directly from the sequence of nucleotides from the global and local alignment.

### Prediction from DNA sequence

The second hypothesis is that the information about the functional profile can be found in the 16S rRNA sequence. To test this hypothesis, I have experimented with a model that uses RNA sequence as the explanatory variable and KO value as a dependent variable.

Unfortunately, scikit only works with float values when training the model. Therefore, I had to transform the sequence into an array of float values by doing the frequency analysis of triplets — for each of 64 possible combinations of three nucleotides in a row, I counted how many times they are in the corresponding DNA sequence. The 64 values were then passed as to scikit as explanatory variables.

The results can be seen in Figure 6.12. The most interesting thing to notice is that there is no exponential decrease in accuracy. Through 90% to 20% of the reference table being used for training the model, the results are consistent, and the drop comes only in the switch from 20% to 10%. This means that the number of OTUs needed to fully train the model is roughly 20% of the reference table, which is about 500 sequences. Then the accuracy of the model does not rise even when we increase the volume of the training data.

**Global alignment vs. not aligned sequences**

As this approach is directly connected to the OTU sequences, not just a distance matrix, it is important to note that the previous results were obtained using raw sequence data. When using the global alignment, the models' accuracy was much worse, as we can see in Figure 6.13. The average correlation was 0.2, which is considered a weak correlation. The results also had really high variance — for some testing data the correlation was as high as 0.9, but for other data the correlation was negative. Therefore we can conclude, that the regression from raw not-aligned sequences is better than the one using globally aligned data.

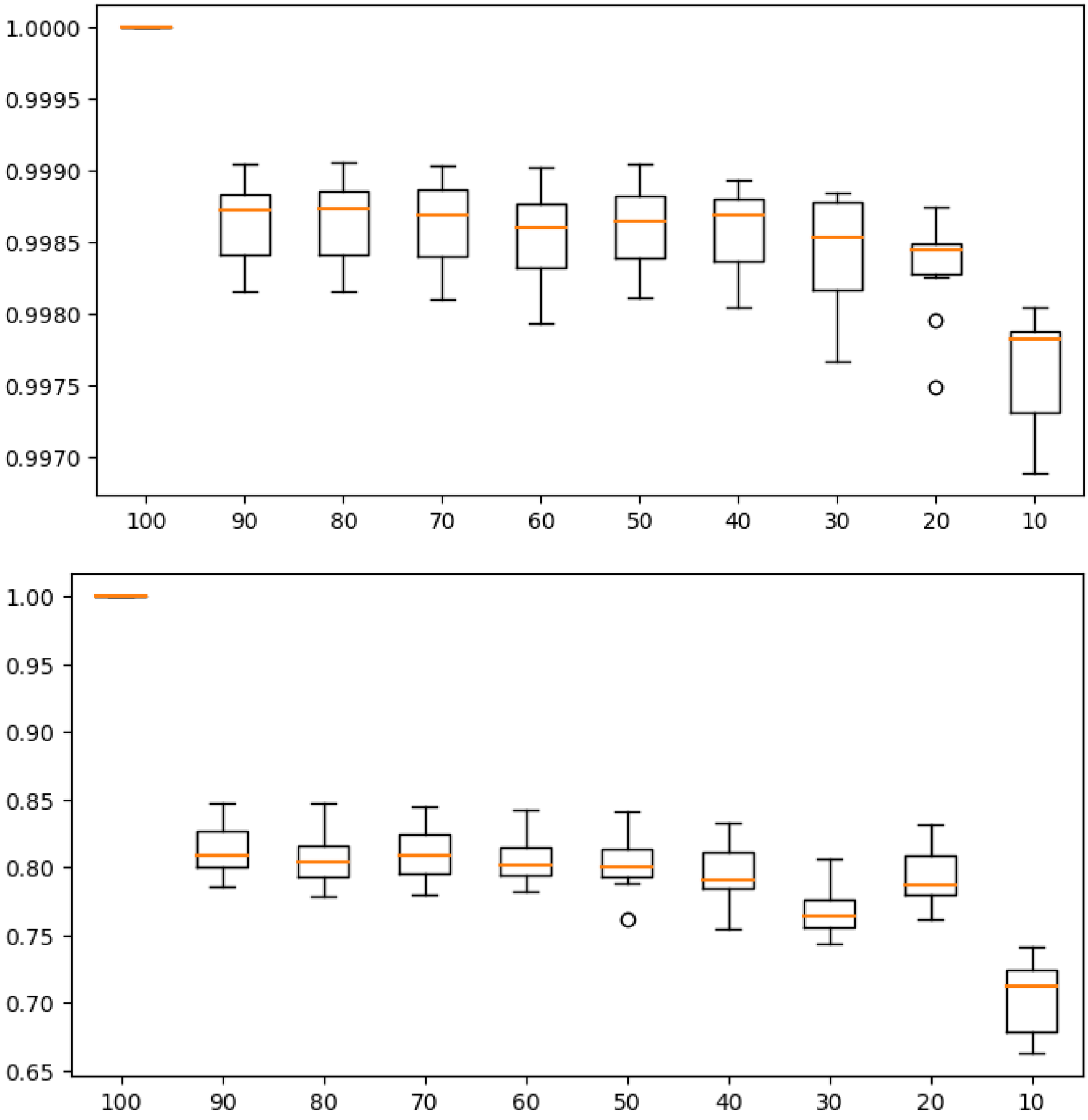

Figure 6.12: Linear regression based method. Regression from DNA sequences. The upper chart shows evaluation on all bacterial functions, the graph below on the rarest 1% of bacterial functions. The x-axis represents the ratio of known profiles in the sample, the y-axis represents the correlation between expected and computed result.

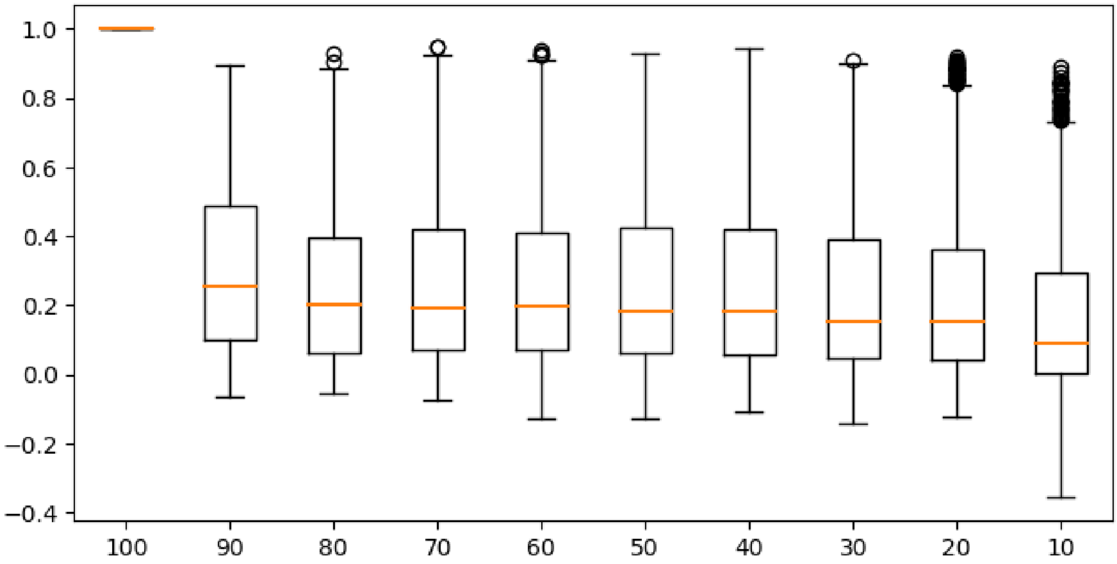

Figure 6.13: Linear regression-based method. Regression from DNA sequences experiments with global alignment. The upper chart shows evaluation on all bacterial functions, the graph below on the rarest 1% of bacterial functions. The x-axis represents the ratio of known profiles in the sample, the y-axis represents the correlation between expected and computed result.

## 6.9 Aggregated results

In this section, I will compare the distance based, phylogenetic tree based and linear regression based methods by analyzing and contrasting the best results from each group. From the distance based group, the best results were achieved using the method based on the scoring matrix. From phylogenetic trees, the most accurate was the inference from the tree acquired by Neighbor-joining algorithm. In linear regression, the best-trained model was the one that predicted the data based on the non-aligned nucleotide sequence.

The aggregated results can be seen in Table 6.3 and Figure 6.14. We can see that the phylogenetic tree method is strictly better than the distance based one, both in the AUC metric and visually in the chart. However, the comparison between the neighbor-joining method and the linear regression is more tricky. By the AUC metric, the neighbor-joining method is better, but if we have only 20% or 10% of the reference table known, the regression method shows much stronger correlation.

As the neighbor-joining method is better by the AUC metric and also in most of the ratio of known reference data, it could be concluded that it is the best from my implemented methods. On the other hand, the experiments with linear regression also offer valuable knowledge. It seems that the accuracy is quite stable, as in the results do not fluctuate between different experiments. The only significant drop can be observed when we move from 20% to 10% data having a known functional profile.

| **Method** | **AUC** |
|---|---|
| Matrix | 80.33 |
| Neighbour joining | 80.54 |
| Regression | 72.20 |

Table 6.3: AUC — Area Under Curve metric, computed from curves in Figure 6.14

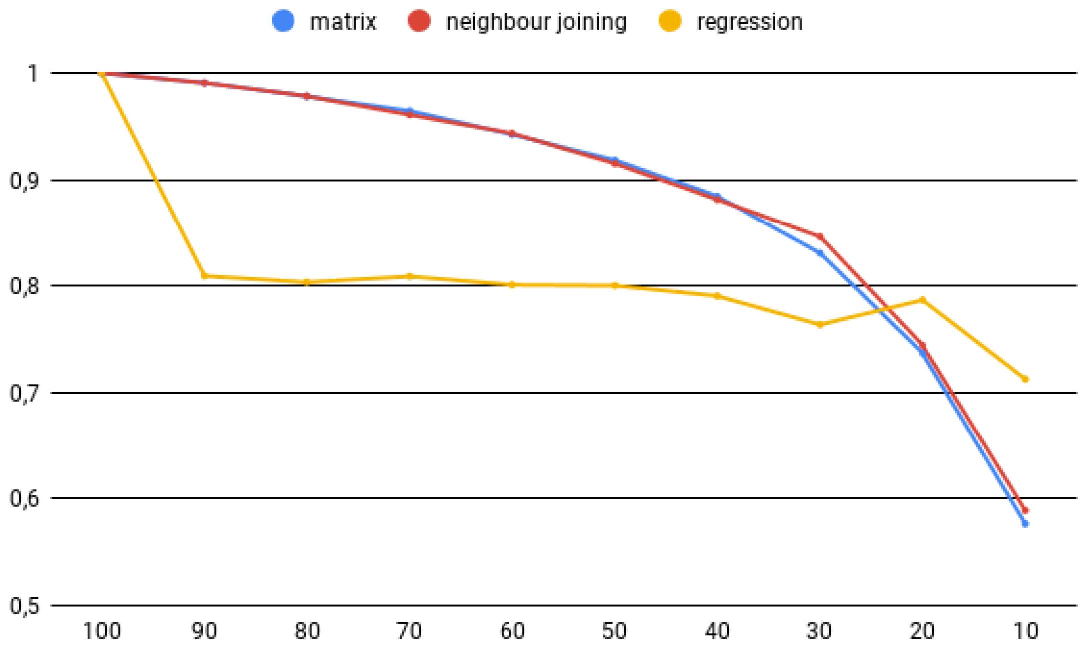

Figure 6.14: A chart comparing the accuracy of the best variant from all three groups of methods for functional profile prediction. The x-axis represents the ratio of known profiles in the sample, the y-axis represents the correlation between expected and computed result. Values rendered in the graph represent the median of all correlations earned in the evaluation.

## 6.10 Comparison with existing tools

Since I have tested and evaluated my tool on Greengenes data, I have decided to compare the accuracy of my tool with Picrust. Unfortunately, I was not able to run the original Picrust genome prediction, which would allow me to evaluate the accuracy for different ratios of reference table missing, due to various complications with execution.

Therefore, I have done my own implementation of most of the Picrust pipeline. I have created a tree from all my testing sequences using a maximum likelihood method in MEGA, which is the first step of Picrust pipeline. To perform hidden state prediction in the tree, I have used the original Picrust 2 script, that allows user-supplied trait tables. The prediction of bacterial function abundance was also my own implementation, which followed the same procedure as Picrust.

As we can see in Figure 6.15, results from Picrust have the same characteristics as my own methods, besides linear regression. The drop in the accuracy is more significant when less than 50% of the reference table is known. When we evaluate Picrust on all bacterial functions, the accuracy of computed and expected results is really high, but when we focus on the most specific bacterial functions, the accuracy drops.

The comparison between the best results from my tool and Picrust can be found in Figure 6.16 and Tables 6.4 and 6.5. As we can see, for all bacterial functions the results are very similar. Although the neighbor-joining method is better by the AUC metric, the results are minimal. The evaluation of the most specific bacterial functions is more conclusive. By the chart and also by the AUC metric, the neighbor-joining method is more accurate.

| **Method** | **AUC** |
|---|---|
| Neighbour joining | 89.93 |
| Picrust | 89.86 |

Table 6.4: AUC — Area Under Curve metric, computed from curves in Figure 6.16 for all bacterial functions

| **Method** | **AUC** |
|---|---|
| Neighbour joining | 80.54 |
| Picrust | 77.74 |

Table 6.5: AUC — Area Under Curve metric, computed from curves in Figure 6.16 for the rarest bacterial functions

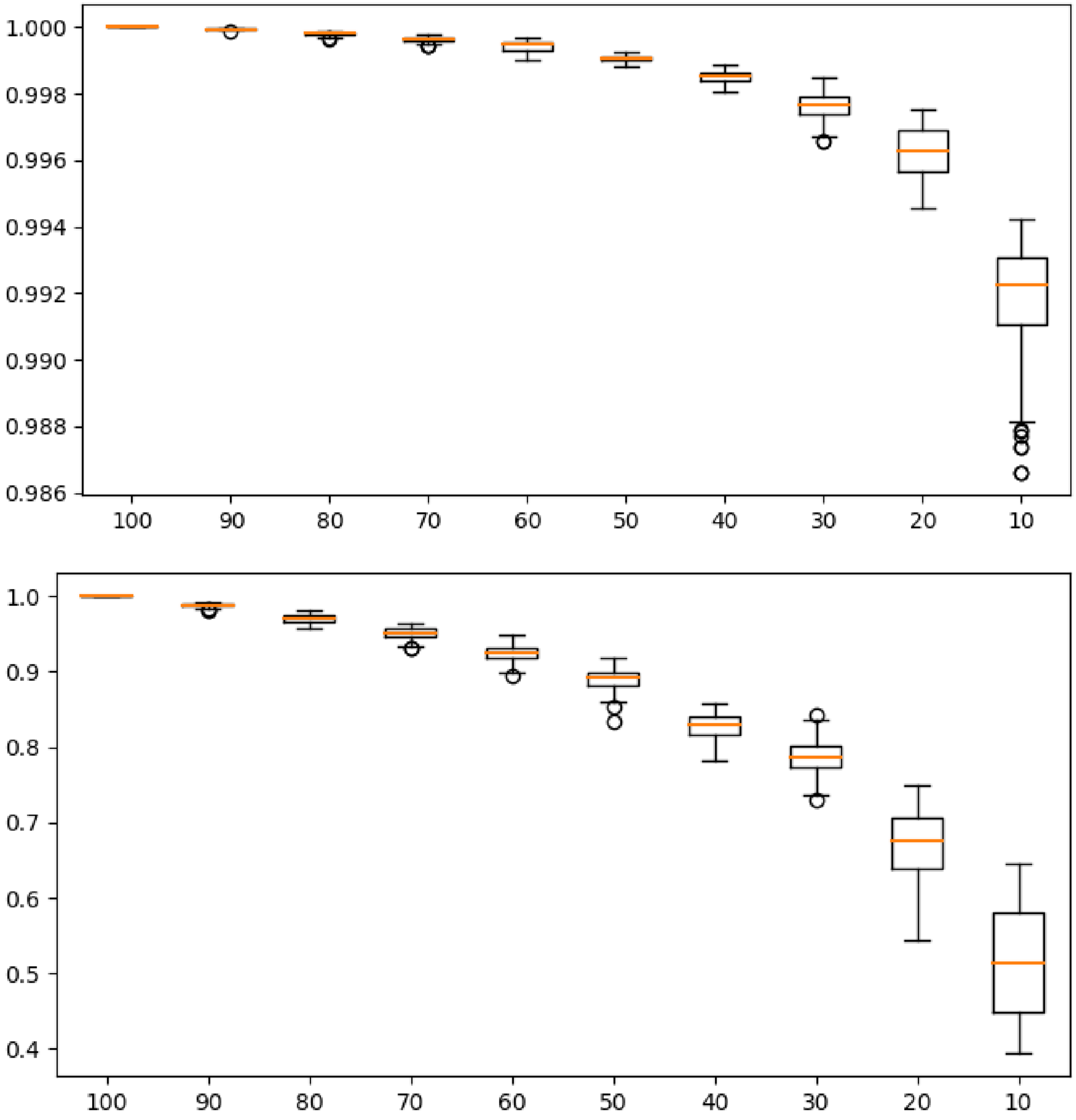

Figure 6.15: Evaluation of my implementation of Picrust. The upper chart shows evaluation on all bacterial functions, the graph below on the rarest 1% of bacterial functions. The x-axis represents the ratio of known profiles in the sample, the y-axis represents the correlation between expected and computed result.

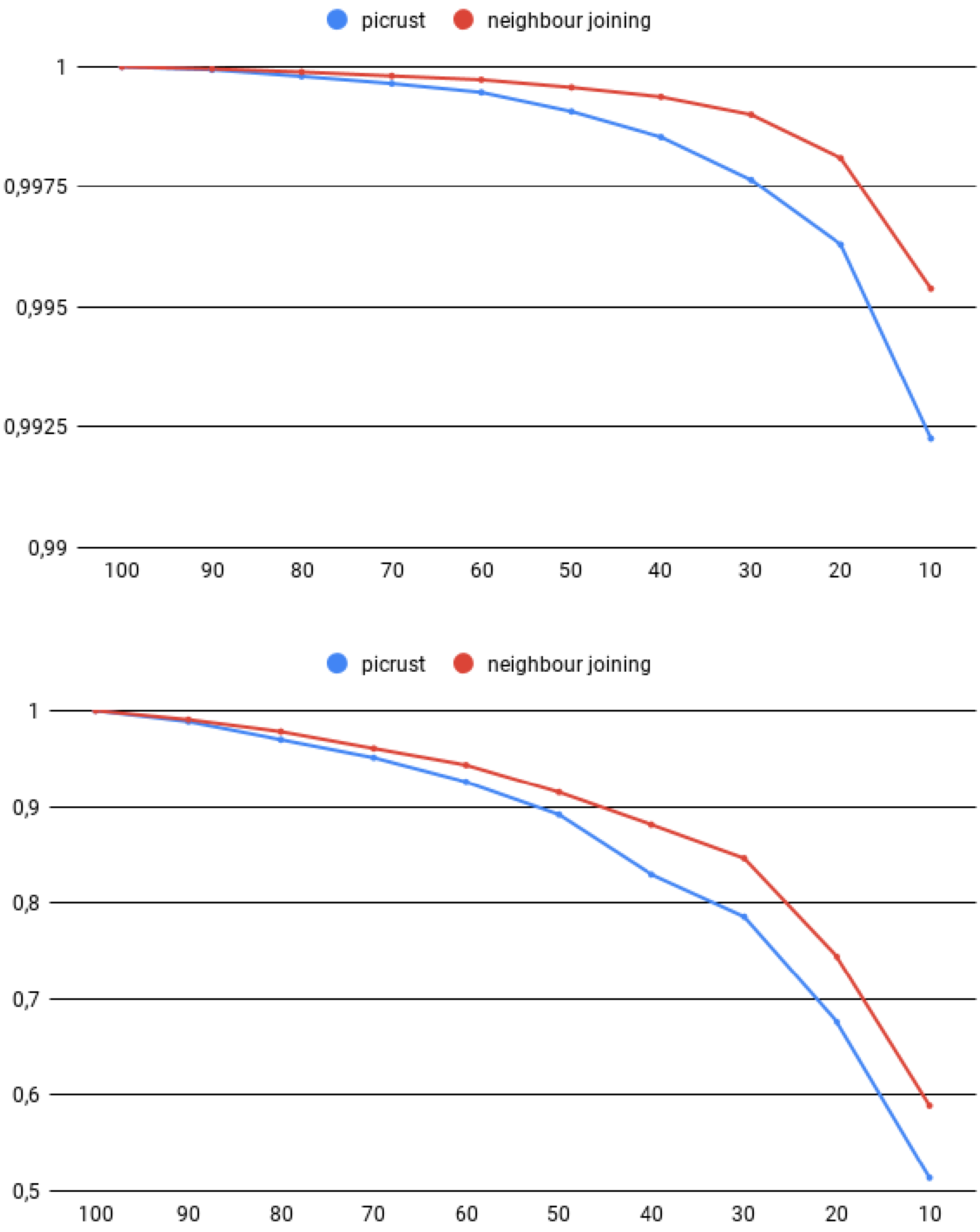

Figure 6.16: Comparison of my implementation of Picrust with my the results from my tool. The upper chart shows evaluation on all bacterial functions, the graph below on the rarest 1% of bacterial functions. The x-axis represents the ratio of known profiles in the sample, the y-axis represents the correlation between expected and computed result.

# Chapter 7

# Conclusion

The goal of this term project was to design a tool for predicting functional profiles from a given sample.

First, the necessary theoretical background was given. After that, I studied the most used existing bioinformatics tools for functional profile prediction - Picrust, Tax4Fun, and Paprica — to determine if a consensus tool can be built over them, but I decided to rather create a new tool that is inspired by them. I described the design and implementation of the tool.

The rest of the paper was focused on evaluating and comparing multiple methods for functional profile prediction. I have implemented and experimented with three types of methods — distance based, a phylogenetic tree based and linear regression. After the initial experiments, I have proposed a new method for evaluating functional precision tools' accuracy, that is focused on only the most specific bacterial function in place of all functions. I have experimented with variants to each type of method and came to the conclusion that the phylogenetic tree based algorithm combined with the neighbor-joining method for tree construction yields the best results.

The last part of the thesis is a comparison of my tool with Picrust. First I have evaluated Picrust both on all and the specific bacterial functions. Then I compared it to the best results earned from my tool. I came to the conclusion that the mentioned phylogenetic tree based method combined with the neighbor-joining tree construction algorithm is more accurate than Picrust. Unfortunately, the results are not completely convincing, as I did not use the original implementation of Picrust, but used my own implementation. Once the tool is more thoroughly tested, it could be published and presented to the bioinformatics community.

The future work based on this thesis may also include evaluating commonly used tools for functional profile prediction with my newly proposed evaluation method. The comparison on the results would be a valuable material eligible for publication.

# Bibliography

[1] DNA: Definition, Structure & Discovery. [Online]. [Visited 10.03.2019]. Retrieved from: https://scikit-learn.org/stable/documentation.html

[2] Genes, DNA, RNA. [Online]. [Visited 14.01.2019]. Retrieved from: https://biologyguide.app/notes/?f=genes

[3] Github respository of the created tool. [Online]. Retrieved from: https://github.com/Miskaaa/thesis

[4] Greengenes database webpage. [Online]. [Visited 28.12.2018]. Retrieved from: http://greengenes.secondgenome.com/

[5] IMG database webpage. [Online]. [Visited 28.12.2018]. Retrieved from: https://img.jgi.doe.gov/

[6] KEGG Orthology database. [Online]. [Visited 4.12.2018]. Retrieved from: https://www.genome.jp/kegg/ko.html

[7] Paprica github repository. [Online]. [Visited 4.12.2018]. Retrieved from: https://github.com/bowmanjeffs/paprica

[8] Paprica introduction tutorial. [Online]. [Visited 4.12.2018]. Retrieved from: https://www.polarmicrobes.org/introducing-paprica/

[9] Pearson Product-Moment Correlation. [Online: https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php]. [Visited 1.4.2019]. Retrieved from: https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php

[10] Picrust 2. [Online]. [Visited 26.12.2018]. Retrieved from: https://github.com/picrust/picrust2/wiki

[11] Picrust webpage. [Online]. [Visited 4.12.2018]. Retrieved from: http://picrust.github.io/picrust/

[12] QIIME - otu picking. [Online]. [Visited 28.12.2018]. Retrieved from: http://qiime.org/tutorials/otu_picking.html

[13] QIIME - taxonomy assignment. [Online]. [Visited 13.01.2019]. Retrieved from: http://qiime.org/scripts/assign_taxonomy.html

[14] scikit-learn: Machine Learning in Python. [Online]. [Visited 20.04.2019].
Retrieved from: https://www.livescience.com/37247-dna.html

[15] Silva database webpage. [Online]. [Visited 28.12.2018].
Retrieved from: https://www.arb-silva.de/

[16] Tax4Fun webpage. [Online]. [Visited 4.12.2018].
Retrieved from: http://tax4fun.gobics.de/

[17] Caporaso, J. G.; Kuczynski, J.; Stombaugh, J.; et al.: QIIME allows analysis of high-throughput community sequencing data. *Nature Methods*. vol. 7, no. 5. April 2010: pp. 335–336. doi:10.1038/nmeth.f.303.
Retrieved from: https://doi.org/10.1038/nmeth.f.303

[18] DeSantis, T. Z.; Hugenholtz, P.; Larsen, N.; et al.: Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB. *Applied and Environmental Microbiology*. vol. 72, no. 7. 2006: pp. 5069–5072. ISSN 0099-2240.

[19] G I Langille, M.; Zaneveld, J.; Gregory Caporaso, J.; et al.: Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences. *Nature biotechnology*. vol. 31. 08 2013.

[20] Genetics Home Reference: What is DNA? – Genetics Home Reference – NIH. January 2019. [Online; visited 20.1.2019].
Retrieved from: https://ghr.nlm.nih.gov/primer/basics/dna

[21] Hiergeist, A.; Reischl, U.; Gessner, A.: Multicenter quality assessment of 16S ribosomal DNA-sequencing for microbiome analyses reveals high inter-center variability. *International journal of medical microbiology : IJMM*. vol. 306 5. 2016: pp. 334–342.

[22] Jo, J.-H.; A. Kennedy, E.; Kong, H.: Research Techniques Made Simple: Bacterial 16S Ribosomal RNA Gene Sequencing in Cutaneous Research. *Journal of Investigative Dermatology*. vol. 136. 03 2016: pp. e23–e27.

[23] Kumar, S.; Stecher, G.; Tamura, K.: MEGA7: Molecular Evolutionary Genetics Analysis version 7.0 for bigger datasets. *Molecular Biology and Evolution*. vol. 33(7). 2016.

[24] Li, Y.: How to Build a Phylogenetic Tree. [Online]. [Visited 2.4.2019.2018].
Retrieved from: http://guava.physics.uiuc.edu/~nigel/courses/598BIO/498BIOonline-essays/hw2/files/hw2_li.pdf

[25] Mandal, A.: What is RNA? August 2018. [Online; visited 20.1.2019].
Retrieved from:
https://www.news-medical.net/life-sciences/What-is-RNA.aspx

[26] Marco, D.: *Metagenomics: Theory, Methods and Applications*. Caister Academic Press. 2010. ISBN 978-1-904455-54-7.

[27] Montgomery, D.; Peck, E.; Vining, G.: *Introduction to Linear Regression Analysis*. Wiley Series in Probability and Statistics. Wiley. 2012. ISBN 9780470542811.
Retrieved from: https://books.google.sk/books?id=0yR4KUL4VDkC

[28] Morgan, X. C.; Huttenhower, C.: Chapter 12: Human Microbiome Analysis. In *PLoS Computational Biology*. 2012.

[29] Orr, I.: Introduction to Phylogenetic Analysis. [Online]. [Visited 2.4.2019.2018]. Retrieved from: https://bip.weizmann.ac.il/education/course/introbioinfo/03/lect12/phylogenetics.pdf

[30] P Aßhauer, K.; Wemheuer, B.; Daniel, R.; et al.: Tax4Fun: Predicting functional profiles from metagenomic 16S rRNA data. *Bioinformatics (Oxford, England)*. vol. 31. 05 2015.

[31] Rintoul, D.; Bear, R.: Taxonomy and phylogeny. [Online]. 2015. [Visited 30.3.2019.2018].
Retrieved from:
http://cnx.org/contents/12696f5e-80cb-4399-9449-b753db45280b@7

[32] Smith, Y.: What is Metagenomics? August 2018. [Online; visited 1.5.2019].
Retrieved from:
https://www.news-medical.net/life-sciences/What-is-Metagenomics.aspx

# Appendix A

# Storage Medium

The storage medium contains an electronic version of this text report and the data of the created bioinformatic tool:

- source codes of the tool
- bash script for easy testing
- user guide
- GPL v.3 — gnu general public license
- example data:
  - KO profile tables with various ratio of known profiles
  - precomputed phylogenetic trees
  - trained models for prediction using linear regression
  - testing samples and corresponding expected results