Bioinformatic methods for the analysis and comparison of metagenomes and metatranscriptomes

Ph.D. Thesis

submitted to the
Faculty of Technology,
Bielefeld University, Germany
for the degree of Dr. rer. nat.

by

Christina Ander

April, 2014

Referees:

Prof. Dr. Jens Stoye
apl. Prof. Dr. Andreas Tauch

Gedruckt auf alterungsbeständigem Papier nach DIN-ISO 9706.
Printed on non-aging paper according to DIN-ISO 9706.

Summary

Microbial communities play an important role in the whole life of planet earth. The fields metagenomics and metatranscriptomics have developed to reveal the taxonomic composition and functional diversity of heterogeneous microbial communities. With the development of new sequencing methods studies in those fields are accelerated. At the same time the new sequencing methods provide a challenge for bioinformatics to process and store a high amount of data.
In the scope of this thesis, methods for the analysis of metagenome and metatranscriptome data were developed. At first, the taxonomic classifier `metaBEETL` was developed and implemented. `metaBEETL` is based on the Burrows-Wheeler transformation and analyses metagenome sequences to gain a taxonomic profile of microbial communities. Using several bias controls, it provides accurate taxonomic profiles while being memory efficient. In this thesis the accuracy of the classifier is shown by the analysis of an artificial metagenome dataset.
Secondly, the rich client software platform `Metrans` was developed for the analysis and comparison of metatranscriptome datasets. `Metrans` consists of a pipeline designed for the analysis of metatranscriptomes. It also includes storage and visualization of the analysis results. The software is currently used in a number of projects. The analysis of a metatranscriptome gained from the infected ear of a man and a time series from tidal flat are presented as analysis examples in this thesis.

Contents

Chapter 1

Introduction

A vast majority of the biosphere of our planet is populated by microbial organisms. The total number of *prokaryotes* on earth has been estimated to be $4 - 6 \times 10^{30}$ cells [100]. *Bacteria*, *microeukaryotes* and *archea* can exist in nearly any environment on earth. In oceanic dead zones, which contain only a minimal amount of oxygen, microbial life still strives, even though no other life forms can be found [14]. Living microbial organisms can also be found in hot springs with a chemical mixture which would kill any other life form [41, 42], or in nearly freezing water [94]. It is unknown for most of those extreme habitats, how living cells can survive yet microbial life is still striving there. Microbes are essential for higher life forms on earth, as a source for nutrients and the primary recyclers of dead matter. In soil they contribute to plant health and nutrition, soil structure and fertility [48]. In the open ocean, microbes are the foremost source of nitrogen fixation [15] and are suspected to hold a central position in the conversion of organic matter into higher trophic levels [19]. Most of these organisms live in complex microbial communities that are adapted to a certain habitat.

Microbial communities in soil and ocean are influenced by human life, but only have a small immediate effect on it. Whereas communities living in direct contact to the human body have an impact on human health and life conditions, they are also directly influenced by human behavior. It has been estimated that the human body carries more microbial cells (10^{14}) than human cells (10^{13}) [6]. In theory human bodies are free of microbes before birth and the microbial community is assembled shortly after birth depending on environmental exposure [7]. Even though this assembly process is similar, each individual has unique microbial communities [53]. It has been suggested that over the million years of coevolution, the interaction between a healthy host and its microbial organisms has reached a Nash equilibrium [8]. The Nash equilibrium is a concept of game theory, where all players choose a certain strategy, based on all possible strategies of the other players. Even though

microbial communities and the human host can live without each other, the relationship between those two has mutual benefits. For humans those benefits include the extraction of extra energy resources from food, stimulation of innate and adaptive immune system, colonization resistance against pathogens and provision of accessory growth factors [17, 85]. Microbial communities gain a livable environment with direct nutrients. The Nash equilibrium only holds for healthy humans as hosts. Not only pathogens infecting the host can become a health problem for humans. Shifts in the microbiota in humans are suspected to play a role in human diseases like diabetes, asthma or cancer [36, 1]. The major areas of human microbiota currently investigated include: gut, skin, lungs and mucous membranes [90]. Over the last years research found positive as well as negative effects of microbial communities inside and on humans. Even though a high amount of research has been done in this field, it is still unclear how the influence of humans is carried out on a molecular level.

Humans do not only share a host relationship with microbial communities. Since the beginning of civilization, microbes were used in fermentation processes for beer, wine or bread [51]. Fermentation is not the only process where microorganisms can be used by the industry. Today many product components that are used daily in our diet or in health products could not be produced without the industrial use of microorganisms [13]. The discovery of microbial enzymes over the last years led to a wide range of new application fields of microbial organisms. For many years, new enzymes or enzymatic functions from microbes have been discovered and then used in biotechnological processes [89]. Those enzymes can be used for the industrial production of chemicals or the reduction of environmental pollution [58]. Another aspect of biotechnological use of microbes is the renewable energy sector. Biogas as well as bioethanol are produced by microbial communities that degenerate plant products [32].

Because of the wide range of living environments, the interaction with humans and the biotechnological application of microbial organisms, the research of those organisms has a long history and is still ongoing. Microbial research includes different aspects, such as:

- Species abundance in the microbial community.
- Potential metabolic functions.
- Active/non-active metabolic functions.
- Interaction between members of the microbial community.
- Interaction between microbes and their host.

For many years research on microbes has been exploring all those aspects. Here we will introduce the basic background of the biology of microbes and present potential research methods for investigating microbial communities.

1.1. Research on microbial communities

The first recorded observation of microbial life was as early as 1665-1683 by Robert Hooke and Antoni van Leeuwenhoek using microscopes [26]. Till the discovery of the structure of the deoxyribonucleic acid (DNA) double helix by Watson and Crick in 1953 [97], most microbial research relied on microscopes. Since the development of sequencing techniques in 1975, all life forms can be studied at a more detailed level using genetic information.

1.1.1. Central biological dogma

Since the discovery of DNA, the typing of organisms can be done either at a phenotype or at a genotype level. Phenotypes describe all characteristics of an organism, like physical properties, behavior and development. Most of these characteristics are encoded in the DNA, making up the genotype of an organism. DNA is made up of four nucleotides *adenine* (A), *guanine* (G), *cytosine* (C) and *thymine* (T), also called *bases*. The nucleotides are attached to a phosphate-deoxyribose backbone. Nucleotides form chemical bonds with each other, stabilizing the DNA in a double helix. A schematic example of this is shown in Figure 1.1. The automatic formation of the bonds between two compatible nucleotides (A with T and C with G) is called *annealing*. This either stabilizes the double helix or is used by the enzyme DNA polymerase to produce a complementary strand of the DNA. In *eukaryotes* (animals, plants, fungi, and protists) the DNA is mostly stored in the nucleus organized in linear chromosomal structures. In *prokaryotes* (*bacteria* and *archea*) DNA is mostly stored in circular chromosomes in the cytoplasm. DNA sequence stretches are composed of genes, which are transcribed into ribonucleic acid (RNA) encoding for one or more proteins, thereby influencing the phenotype of the organism. For the transcription step the RNA polymerase binds to the DNA, seperates the double helix and produces a complementary antiparallel RNA strand, keeping the bonding pair cytosine and guanine but replacing thymine with *uracil* (U). If a gene coding region was transcribed, the produced RNA is called *messenger RNA* (*mRNA*). It remains single stranded and therefore degenerates faster than DNA.

Since similar characteristics can be encoded by different genetic makeup, similar phenotypes give only a small implication on the shared genotype. The genetic sequence of an organism is inherited by the parents of that organism, either through

[1]Source:http://en.wikipedia.org/wiki/DNA, Figure slightly adapted.

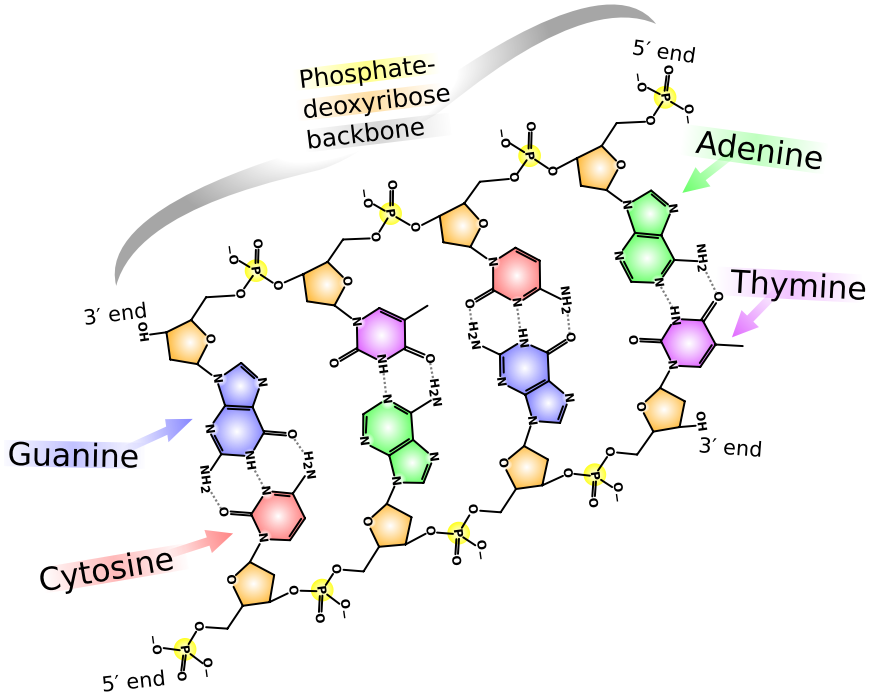

Figure 1.1.: Chemical structure of DNA. Hydrogen bonds between guanine (blue) and cytosine (red) and adenine (green) and thymine (purple) are shown as dotted lines[1].

sexual reproduction, involving two parents, or through cell division with only one parent. The genotype of a species changes gradually through mutations of single nucleotides or rearrangement, separating the species over time genetically from other species while sharing the same ancestors. Over time this builds a tree like structure of the degree of kinship between species, which is also called the *tree of life*. Finding the correct ancestors of two species and classifying them to the proper position in the tree of life is called *taxonomic classification*. This is often done on basis of the genotype rather than the phenotype, because phenotypes can look similar even though the species are not closely related.

1.1.2. DNA Sequencing

DNA sequencing is a technique to determine the sequential arrangement of the nucleotides in a given strand of DNA. The development of new and better sequencing techniques since the discovery of the DNA double helix till today is still ongoing. In 1975 the first sequencing technique was published by Sanger and Coulson [81]. The determination of the DNA code was based on incorporation of chain-terminating nucleotides and the enzymatic function of the DNA polymerase. For this technique

a DNA fragment had to be cloned in a plasmid vector. Two years later Maxam and Gilbert developed a sequencing technique based on chemical modification of specific bases and the cleavage of the DNA at those nucleotides [59]. In both methods the length of the sequenced DNA is restricted to a maximum of 1000 nucleotides, as a result of insufficient separation of larger DNA molecules that differ in length by only one nucleotide. A continuous string of genetic sequence gained by sequencing is called a *read*. Since the Maxam-Gilbert method is technical more complex, Sanger sequencing was the most used sequencing method for many years.

In recent years so called "next-generation" sequencing (NGS) methods have been developed. Using those techniques a cloning step is no longer necessary and through massive parallelization millions of reads can be produced in a short amount of time [93]. The amount of reads gained by those sequencing techniques often masks sequencing errors in single reads. The accuracy of single nucleotides is dependent on their placement in the read. Accuracy generally drops towards the 3' end of the read, but single oligonucleotides can also cause specific sequencing errors.

The first technique was developed by *454 Life Science*, which generates 400-600 megabases in a ten hour run. This technique generates one million reads in a 24 hour run. The produced reads have an accuracy of 99.9%. At this time the most cost efficient sequencing methods are *sequencing by synthesis* (*Illumina*) and *sequencing by ligation* (*SOLiD*) [57]. Sequencing by synthesis can produce up to 3 billion reads with a read length up to 250 bp in a ten day run. The accuracy of those reads is at 98%. Sequencing by ligation produces 1.2 to 1.4 billion reads with a read length up to 80 bp in one to two weeks, the reads have an accuracy of 99.9%. To gain longer reads Pacific Bioscience uses *single-molecule real-time sequencing*, gaining reads with an average length of 5,500 bp to 8,500 bp. The single reads of this method have an accuracy of 87%, through alignment methods they can achieve 99.99%. Removing the cloning step is the biggest advantage of the NGS sequencing methods, since this step was time and cost expensive. In addition direct sequencing of DNA makes it possible to obtain genetic code from non culturable microbes.

1.1.3. History of metagenomics

Metagenomics is the research on genetic material isolated from a mixture of microorganisms instead of a single organism. The term metagenomics was first defined in 1998 by Handelsmann et al. by proposing to clone environmental DNA fragments into BAC vectors [31]. Transformation of those BAC vectors into *Escherichia coli* host cells gave the opportunity to screen the cultures for interesting metabolic functions. This technique made it possible to access certain genes and gives a rough overview of the species present in the microbial community. It was first used in 2000 by Rondon et al. who cloned DNA fragments of a soil community [76]. A phylogenetic

as well as a functional overview over the microbial community was gained. Rondon discovered that the cloned DNA fragments coded for antibacterial activity and found phylogenetic marker genes similar to genes from culture dependent methods. Such antibacterial genes are a potential thread to host microbes, causing BAC vector transformation to be highly selective. Regardless of this disadvantage a number of significant projects relied on this cloning method. One of the biggest was the sampling of the Sargasso Sea [92]. A total of 1.045 billion base pairs were sequenced of seawater samples from the Sargasso Sea near Bermuda, showing that the amount of microorganisms with an unknown genome is enormous. The sequences gained in those big projects led to a number of new discoveries even years later [70, 69, 37] when they were screened once again for new enzymatic functions.

The development in the metagenomic field was relatively slow till next generation sequencing techniques were developed. In 2006 the 454 sequencing technique was used to directly sequence a metagenome, comparing physically close sites of a mine [21]. Difference in the potential metabolic functions as well as the microbial composition was found. Since then, metagenomic research using NGS has accelerated.

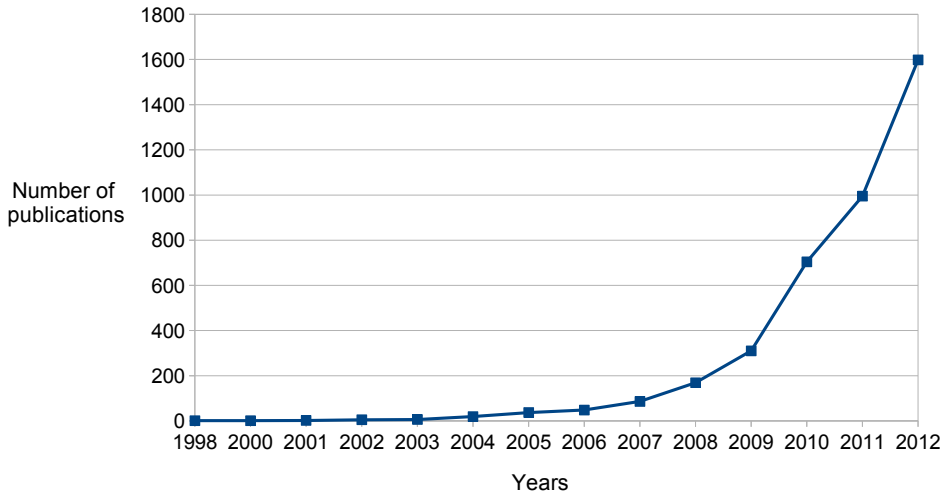

Figure 1.2.: Number of papers published over the years which either contain the term 'metagenomics' or 'metagenome' [2].

The number of papers published with the term metagenome has risen exponentially from 2006 to 2012 (Figure 1.2). Shortly after the invention of the new sequencing

[2] http://www.ncbi.nlm.nih.gov/pubmed

techniques, various big metagenome projects started. For example the Human Microbiome Project tries to strategically catalog all possible microbial communities of the human body [91]. The Earth Microbiome Project analyzes microbial communities across the whole globe, researching different environmental factors and their effect on soil microbiome [27]. These projects are mostly well organized, acting globally and connecting research all over the world. All metagenome projects share research techniques. There are different techniques to investigate the composition and metabolic functions of microbial communities. In the next section we will present a number of those research techniques.

1.1.4. Research techniques for non culturable microorganisms

In this section we cover the currently used research techniques for microbes that can not be cultured singularly. We give a short overview of the most used techniques, as shown in Figure 1.3. A more substantial background will be given chapter 2 and 3 of the dissertation, where bioinformatic analysis approaches for certain research methods are presented. Each technique can be used to achieve a certain set of research objectives as presented before.

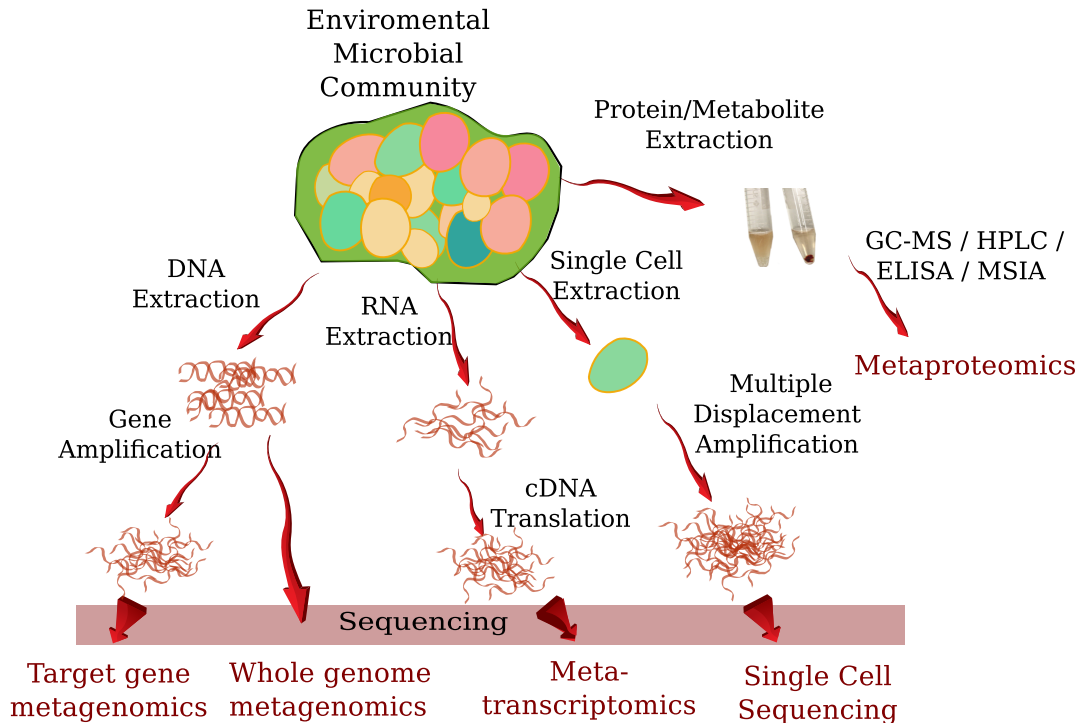

Figure 1.3.: Research techniques for microbes, that are not culturable as a mono culture.

Targeted gene metagenomics is used to gain the information about species abundance in a microbial community. Specific marker genes are sequenced instead of whole DNA. Most of the published human microbiome studies focus on the sequencing of the 16S ribosomal ribonucleic acid (16S rRNA). The 16S rRNA is the smaller part of the ribosome in *bacteria* and *archea* [68]. The ribosome serves as primary site for protein synthesis in biological cells and is therefore crucial for survival. The whole DNA of the community is extracted and the targeted gene is amplified by *polymerase chain reaction* before sequencing. The resulting sequences are used to create a taxonomic profile of the microbiome. A broader background as well as advantages and disadvantages of this technique will be given in Section 2.1.1.

Whole genome metagenomics detects genetically encoded metabolic functions and taxonomic composition. The whole DNA of the environment is extracted, fragmented and sequenced using random primers without the amplification of certain sequences. The result is a set of sequences originating from coding as well as non coding regions of all microbial genomes in the community. The sequences are used to obtain a taxonomic profile as well as to access the repertoire of genes. A deeper introduction to whole genome metagenomics will be given in Section 2.1.2.

Metatranscriptomics is a relatively new research field, where expressed genes of microbial communities are studied, by means of studying their RNA. Since the transcription of DNA to RNA is the first step in gene expression, it can be used to find the expression level of certain metabolic functions in the community. The total RNA of the community is isolated, the mRNA possibly enriched, then translated into cDNA and sequenced. The resulting sequence data is a mixture of sequences from expressed genes and ribosomal RNA of different organisms. The extraction of RNA instead of DNA has the advantage that only living organisms are studied since RNA degrades faster than DNA [20]. A deeper introduction of metatranscriptomics and the needed analysis is given in chapter 3. Like any study of transcripts, metatranscriptomics has the disadvantage that an expressed RNA does not automatically lead to a metabolic product. Another disadvantage is that non-active organisms are not accessed. The amount of RNA in a cell is constantly changing, therefore a high amount of samples at different time points is more important in this field than in metagenomics.

Metaproteomics has emerged as a research field to study the structure and function of the proteins in an environmental community. The proteins of the whole environment are extracted and their mass-to-charge ratios are studied using mass spectrometry methods [95]. The proportions of mass to charge is individual for each protein and can be used to identify and analyze metabolic compounds. The amount

of analyzed probes should be similar to that in metatranscriptomics, since protein content of cells is also in flux.

Single cell sequencing is used to obtain the whole genome of one microbial organism. A minimum of DNA material is needed for sequencing. For unculturable microbes it was not possible to obtain that material until the development of *Multiple Displacement Amplification* (MDA) [87]. MDA makes it possible to obtain a high amount of large size DNA molecules of high quality for sequencing. The process starts with the single stranded DNA of a single cell, random hexamer primers are annealed to the DNA. A highly error resistant DNA polymerase is used to amplify the DNA sequences at constant temperature. After sequencing, the reads can be assembled to the original genome of the microbe, facing problems like nonuniform coverage of the genome and chimeric sequences produced by the MDA method [67].

1.1.5. Bioinformatic challenges in modern microbial research

There are several challenges for bioinformatic researchers in microbial research based on reads gained from NGS. Since the reads are gained from a mixture of microbes it is not possible to map them to a single organism. Therefore, the reads must be classified functionally and taxonomically to gain functional and taxonomic profiles of the microbial community. There are several methods to classify the reads, a number of taxonomic classification methods will be presented in chapter 2. Many of the functional classification methods rely on a comparison of the reads with databases containing already classified reads. Those databases often contain sequences from microbes that can be grown in a mono-culture since those are easier to sequence and research. Therefore, they are not very similar to sequences gained from a mixed microbial community, making it hard to produce reliable functional profiles. Analysis results in research of microbial communities are highly complex, since they are from different organisms, adding the taxonomic source as a new level of complexity. New visualizations are needed for the combined taxonomic and functional profiles of microbial communities. The high amount of fast produced sequencing data also poses a challenge to bioinformatics. The sequences must be analyzed and stored in a cost efficient way. At best methods should work on compressed data or should at least compress the data when it is not analyzed. If sequence analysis is done by non informatics, the software used should be user friendly and contain easy to start tools for the analysis.

1.2. Overview of this thesis

In this chapter we have given an overview of research methods for microbial organisms and communities. We presented shortly the central biological dogma and

gave an introduction to sequencing. The most commonly used research methods for microbial communities were also presented. The data obtained by those research methods needs to be analyzed, normalized and compared to obtain more information about microbial communities. In this thesis we present new bioinformatic methods for data analysis of sequence data of metagenomes and metatranscriptomes. In chapter 2 we will present the taxonomic classifier `metaBEETL`. It was developed for fast taxonomic classification of whole genome metagenomic datasets and its accuracy was tested on a simulated profile. In chapter 3 we will present the new software platform `Metrans` for the analysis and comparison of metatranscriptome datasets. In each of those chapters an introduction in the background of the new methods will be given, the methods and results will be presented and discussed and a summary and outlook for that particular method will be given. A conclusion and outlook over the whole thesis will be presented in chapter 4.

Chapter 2

metaBEETL - A taxonomic classifier for whole genome metagenome reads

The abundance of different species in an environmental microbial community is called a *taxonomic profile*. Taxonomic profiles are an important part in fundamental microbial research and are also used in medical research. Changes in the taxonomic profile of a microbial community can indicate the health status of its host. Therefore, the taxonomic profile of a community could help possible diagnoses, if it is reliable and obtained fast. Here we present `metaBEETL`, a fast and accurate taxonomic classifier for whole genome metagenomics sequences. The classifier specializes in the fast creation of accurate profiles of well researched environments. `metaBEETL` uses new normalization steps to avoid biases like genome length and copy number variations of genes, creating accurate profiles of microbial communities.

First we introduce taxonomic classification of microbial communities in Section 2.1. This includes the research methods *targeted gene metagenomics* (Section 2.1.1) and *whole genome metagenomics* (Section 2.1.2). Available methods for taxonomic classification are presented in Section 2.1.3. As a sequence compression and comparison method we present the *Burrows-Wheeler transformation*, on which the classifier `metaBEETL` is based in Section 2.2. The classifier and its classification methods are shown in Section 2.3. The results of accuracy testing of `metaBEETL` are given in Section 2.4. The accuracy test will be discussed in Section 2.5 and an outlook will be given.

2.1. Introduction to taxonomic classification of microbial communities

Taxonomic classification is the scientific field of grouping organisms together on the basis of their characteristics. Those groups, called *taxa* (singular *taxon*), are assigned a name and a rank. Taxa, sharing a set of features, can be aggregated to form a super group of higher rank. These features can either be taken from their phenotypes or from their genotypes. Since those types differ, it can happen that one organism has different taxonomies, depending on the classification features. Currently the most common used taxonomy is created by the National Center for Biotechnology Information (NCBI) [82]. This taxonomy is entirely based on genetic sequence similarity. It arranges the taxa in a tree like structure containing six major levels, which are from top to bottom: *superkingdom, phylum, class, order, family, genus* and *species*. For each level exist a different number of sub-levels, for example *subclass* or *subphylum*. Those sub-levels do not occur in all classifications. Table 2.1 shows examples for the taxonomic classification of the gray wolf and of *Staphylococcus agnetis* based on the NCBI taxonomy. The deeper a taxon is placed into the taxonomic tree, the more individual characteristics do the organisms in it share. To classify a genetic sequence to a low rank in the tree, it has to be either long enough to find a sufficient amount of characteristics or contain a number of traits which are unique to a taxonomic group at that rank.

Rank	**Taxon** (gray wolf)	**Taxon** (*Staphylococcus agnetis*)
Superkingdom	*Eukaryota*	*Bacteria*
Phylum	*Chordata*	*Firmicutes*
Class	*Mammalia*	*Bacilli*
Order	*Carnivora*	*Bacillales*
Family	*Canidae*	*Staphylococcaceae*
Genus	*Canis*	*Staphylococcus*
Species	*Canis lupus*	*Staphylococcus agnetis*

Table 2.1.: Taxonomic classification of the major ranks of the gray wolf (*Canis lupus*) and *Staphylococcus agnetis* according to the NCBI taxonomy .

A taxonomic profile of a microbial community is the overall abundance of all taxa in that community. To obtain it, the reads in a metagenomic dataset have to be taxonomical classified. In metagenomics most sequenced reads are quite short. Therefore, it is often possible to classify the reads only to a higher taxonomic level. Nevertheless for gut microbial communities it has been shown that even changes in the taxonomical profile at a rank as high as phylum can be related to diet and obesity

of the host [65]. Therefore, a major research focus in metagenomics is finding the abundance of all taxa at all ranks in a microbial community. The genetic sequences needed for taxonomic classification can either be obtained by *targeted gene metagenomics* or by *whole genome metagenomics*. Both methods are based on the assumption that species occurring in a higher abundance in the community will result in more sequenced reads than less abundant species. Therefore, the taxonomic profile can be obtained by classifying the reads and aggregating the counts of the classification.

2.1.1. Targeted gene metagenomics

All living organism have *essential genes*, which are needed for survival. Those genes are under evolutionary pressure to remain stable over time and are often widely spread in the taxa of one taxonomic group. For example, the ribosomal 16S rRNA gene can be found in all microbial genomes and is mostly stable, containing nine hypervariable regions. Because of their stability, essential genes are often used for taxonomic classification of organisms. Over the years many single cultures were taxonomically classified using only the sequence information of their essential genes without obtaining the whole genome. Therefore, there exist many more reference sequences of essential genes than of whole genomes.

The taxonomic composition of a microbial community can also be obtained by using the sequence of essential genetic regions [99]. For this the whole DNA is extracted, the targeted genetic region is amplified by *polymerase chain reaction* (PCR) and afterwards sequenced [78]. During PCR the targeted single stranded DNA is used as a *template* to amplify it to thousands of copies of itself. Small stretches of DNA (*primers*) anneal to the template starting the elongation of the DNA, creating a double helix. This helix is broken apart by temperature changes and the elongation restarts on both single strands. The repetition of this cycle creates a high number of copies of the template. Part of the targeted genetic region has to be known, to create specific primers for the PCR. Those primers can be designed for either single species or whole taxonomical groups, depending on how preserved the region is.

For taxonomical profiling of a microbial community, targeted gene metagenomics has two advantages compared to whole genome metagenomics. Since the essential part of the genetic material is amplified before sequencing, the amount of reads required for a sound taxonomic profile is less than in whole genome metagenomics. Additionally, the taxonomic profile includes taxonomic groups that contain organisms whose essential genes, but not whole genomes, have been sequenced.

Targeting single genetic regions also has potential biases. First of all, the primers used during PCR do not cover all parts of the taxonomic tree equally. Primer design is not possible if the targeted genetic region of a certain species was not sequenced

yet. Therefore, an inconsistent mixture of primers with different annealing affinity for different branches of the taxonomic tree is used. Secondly, the elongation step during the PCR can also lead to so called *chimeric sequences*. In a chimeric sequence one part of the sequence was derived from a different organism than the other part, making it impossible to classify the sequence correctly. The amount as well as the composition of chimeric reads differ from sample to sample. Thirdly, essential genes are often present at multiple places in the genome. For example one microbial genome can contain up to 15 copies of the 16S rRNA gene [49]. This creates a huge bias if targeted to gain the taxonomic profile. This bias can be considered if the copy number of the targeted region is known. Unfortunately, those copy numbers vary even between closely related species. Some of those biases can be avoided using whole genome metagenomics.

2.1.2. Whole genome metagenomics

To counteract the biases of targeted gene metagenomics and to gain information of potential expressed functions, the complete DNA of the microbial community is analysed. After DNA extraction, it is sheared to an expected read size and sequenced. Those reads represent randomly drawn parts of all genomes of the community, excluding possible biases from the extraction, shearing and sequencing step. A functional as well as a taxonomical profile can be computed from those reads. The functional profile offers information about the existence and amount of potential expressed genes in the microbial community. This sheds light on the question how microbes can survive in extreme environments and can be used to identify possible pathogens. Classification of the reads can be improved by increasing their length. It is possible to gain a longer sequence by assembling the short NGS reads. Unfortunately the assembly process is a possible source for biases. If reads from different species are assembled into one sequence, it is not possible to obtain a correct taxonomic classification to species level.

Even though the biases from the PCR are no longer an issue, there exist other sources of bias that should be kept in mind. First of all, the length of the different genomes is probably the bias which has the most impact on the profile. Microbial genomes can be between 0.2 Mbp [60] and 10 Mbp [16] base pairs long. The taxonomic profile could be skewed towards microbes with larger genomes since those produce more sequenced reads. Secondly, as in targeted gene metagenomics, copy number variations of genes can result in a bias of the reads leaning towards a species with a high number of certain genes. At last, another possible bias in the taxonomic profiles are *plasmids*. Plasmids are small circular stretches of DNA that are often transferred between bacteria, sometimes even crossing the species border. Therefore, reads originating from plasmids could be classified to species different from those they originated from. Even though these biases exist, whole genome metage-

nomics has the advantage of having less bias than targeted gene metagenomics and it also can show the metabolic potential of a community as well as mutations to already sequenced genomes.

2.1.3. Taxonomic classification methods

Given a set of sequence reads of a microbial community the first step to obtain its taxonomic profile is taxonomically classifying all reads. Existing taxonomic classifiers use either *composition based* or *comparison based* methods. Composition based methods analyze reference sequences to gain a number of distinct characteristics. Those characteristics make up a model of the reference. The classification of a read starts with the computation of the characteristics of that read. Afterwards only the characteristics of the model and the read are compared, not the sequence in itself. On the other hand homology based methods depend on the direct sequence comparison of the read to a reference database. In this Section we will give a short overview of used methods and present a number of taxonomic classifiers.

Composition based methods extract sequence features (e. g. GC-content or oligonucleotide frequency) from a set of reference sequences. Those features are used to build a classification model. Model building often makes use of *interpolated Markov models* (IMMs) [79], *naive Bayesian classifiers* [103] or *k-means/k-nearest-neighbor* [46] algorithms. Since short sequences contain only limited amount of features, the accuracy of classification highly depends on the length of the sequences to be classified. Comparison methods do not rely on single features and can therefore accurately classify short sequences. Therefore, some classifiers like `PhymmBL` include an additional similarity search [10]. The classification methods can be divided by chosen characteristics as well as the learning model of the classifiers. Most classifiers use oligonucleotide frequency as characteristic for different sequences, for example the classifiers `NBC` [77] and the `RDP Classifier` [96]. In both methods a *naive Bayesian* classifier is trained and used for classification. `NBC` classifies reads from whole genome metagenomics, while the `RDPclassifier` specializes in 16S rRNA sequences. *Support Vector Machines* are another possible model for oligonucleotide frequencies, they are used by `PhyloPythia` [61] . Some methods try to upgrade the oligonucleotide frequency count with other characteristics of the genome sequence. `RAIphy` [66] scores the oligonucleotide counts relative to their abundance, so that oligonucleotides occurring in many genomes gain more weight in the classification process. Another possibility to use the composition of the sequences is the creation of *Markov Models*. Those are used in `PhymmBL` and `SCIM` [47] as Interpolated Markov Models.

The main advantage of composition based classifiers is the speed of the classification. The main effort of those methods lies in building the model for classification,

fortunately this is only needed if the reference changes. In contrast the classification step is much faster and more frequently used. Unfortunately the relatively short NGS reads have only a limited amount of characteristics that can be used for classification. Therefore, most composition based classifiers have minimal read length of at least 250 bases in order to produce reliable taxonomic profiles.

Comparison based methods use direct comparison of the sequences to a reference instead of precomputed models. With this type of methods shorter sequences can be classified at the cost of compute time and power. Taxonomic classifiers can be categorized by *type of comparison*, *classification method* and *reference*.

Most comparison based methods use a BLAST [2] version for the comparison step and afterward classify the reads with a certain similarity to the reference database. BLAST (Basic Local Alignment Search Tool) is a heuristic algorithm developed by Altschul et al. in 1990 for the comparison of sequences with large databases. For fast comparison, BLAST uses a seeding method to find parts of the query sequences with high similarity to the database sequences. If a seed is found, the sequence parts of query and database sequences surrounding the seed are aligned. If this alignment meets certain criteria, e. g. overall similarity or alignment length, the corresponding sequence in the database is called a *hit* for the query sequence. MEGAN [39], jMOTU/Taxonerator [43], MetaPhyler [56], MG-RAST [62], MTR [28] and MARTA [35] are using a simple BLAST or BLASTX search. Other methods like CARMA [25] and SOrt-ITEMS [64] use a reciprocal BLASTX search to improve taxonomic classification. Only the classifiers FACS [88] and Genometa [18] do not use BLAST. FACS [88] uses Bloom filters to index their reference database and Genometa relies on the alignment programs BOWTIE [50] or BWA [54]. This speeds up the comparison step, but restricts the size of the indexed database.

Since the comparison step of the taxonomic classification is rather time consuming, there are classifiers that restrict the database size. A restricted database can be used as reference for classification if biologically meaningful sequences are selected for this database. jMOTU uses reference sequences from the *Bar Code of Life* project. In this project, sequences unique to a taxonomic group are studied. MG-RAST restricts the references to protein coding sequences. Based on the idea of targeting gene metagenomics, MetaPhyler uses marker genes as a reference. Genometa relies on a database consistent of one genome for each genus. The advantages of database restriction are reduced misclassifications and a speed up of the comparison step. The disadvantage is the small number of sequences classified. MEGAN as well as CARMA have shown that the best classification can be achieved using all available sequences from the NCBI *Non Redundant Database* (NCBI-NR) as reference.

Most comparison based taxonomic classifiers use a *Lowest Common Ancestor* (LCA) approach for taxonomic classification of the sequences. In this approach sequence with more than one hit in the reference database are placed in the taxonomic tree at the lowest common ancestor of all those hits. Of the presented classifiers only MTR and Genometa do not use a variation of the LCA classification. MTR clusters the hits of different sequences and classifies the clustered sequences depending on the overlap in their hit. Genometa uses no further taxonomic classification but maps the sequences to the corresponding genomes.

In 2012, Bazinet et al. compared the different classification methods [5]. They used four different datasets, taken from literature, and compared the results of true positively (TP) and false positively (FP) classified reads. Of the comparison based classifiers CARMA, MEGAN, MetaPhyler and MG-RAST were compared, stating problems running other possible classification tools. In this study the classifiers MEGAN and CARMA presented the best possible classification for the given datasets. Therefore, we will present those two classifiers in more depth. Only after this study, the classifier Genometa was published. Like metaBEETL, Genometa is based on the *Burrows Wheeler Transformation* (BWT) [12] and will therefore be presented in more detail as well. An overview of the three classifiers will be given in order of their publishing date.

MEGAN was one of the first developed comparison based classifiers. It has an easy to use graphical user interface, for starting the classification and displaying trees of the resulting taxonomic profile. MEGANs taxonomic classification depends on a full BLAST output of the metagenome reads compared to any database compatible with the NCBI taxonomy. BLASTN, BLASTX or BLASTZ can be used for the comparison step. The BLAST hits in a certain *bit score* range are mapped to the NCBI taxonomy . The bit score in a BLAST comparison is a score to determine the significance of the hit according to database size. MEGAN assigns the LCA to each read and stores the results internally as read-taxon match. The aggregation of the read classifications form a taxonomic profile, represented as a tree in the user interface. The read-taxon matches can be exported from MEGAN in different formates including cvs. Additional features have been added to MEGAN since the first release. In 2009 for example statistical methods for comparison of metagenome datasets were added [63]. Relying on a BLAST comparison is the drawback of MEGAN. First of all, the comparison step can not be started from the graphical user interface of MEGAN. Therefore, a minimum amount of computer skills to obtain the comparison results are needed to use MEGAN. Secondary, the BLAST comparison is computationally expensive.

CARMA is a taxonomic classification tool, also relying on BLAST for the comparison step. CARMA3 also contains additionally a classification option for 16S rRNA sequences. The BLASTX comparison to the *NCBI non redundant protein sequence database* proofs to result in the most true positive classified reads [24]. Instead of using all hits in a certain range to generate the LCA of one read, CARMA refines the results further using a reciprocal BLAST search. For each read with a hit, a new database is constructed, consisting of all sequences of the hits and the query sequence. Then BLASTp is used to compare the sequence of the first hit against the newly constructed database. All hits are ordered by decreasing homology, hits between the first hit and the original query sequence are used for the taxonomic classification of the query sequence. This refines the taxonomic classification and prevents false positive assignments. CARMA uses the bit scores of the reciprocal search to further refine the taxonomic classification to a lower level in the taxonomic tree. This reciprocal search results in a detailed classification but is unfortunately computationally quite expensive.

Genometa is a tool for taxonomic classification of metagenome reads from well studied environments. It uses either BWA or BOWTIE to compare the reads to an integrated database. Those mapping algorithms use the BWT to index a reference sequence. The index is used to map query reads to the reference in a time efficient manner. Genometa uses those mappings for metagenomics analysis by showing differences (mismatches, deletions, insertions) between the mapped reads compared to a number of reference genomes, therefore gaining a taxonomic classification of the reads on the level of strain and species. With this, Genometa filled a gap since no other classifier showed single nucleotide polymorphism between metagenome reads and reference genomes. Since the whole reference has to be contained in memory for mapping, both BWA and BOWTIE restrict the size of the reference database. For this reason, Genometa uses for each genus exactly one reference sequence. Unfortunately this can result in a low amount of classified reads if the originating genomes do not exist in the reference.

In summary the existing taxonomic classifiers based on sequence comparison are either slow through the usage of BLAST, or their reference database is restricted in size. Classifiers based on BLAST also have the disadvantage that single mutations can not be shown. Using the BWT index for comparison of the sequences proved to be much faster. Currently BWT based mappers restrict the database size, because a random access to the reference index is needed for the comparison.

2.2. Algorithmic background

In the previous section we presented several approaches for taxonomic classification of NGS metagenomic sequences. In most methods the comparison to a biological database presents the computational bottleneck. To speed up the comparison the reference can be indexed using the Burrows-Wheeler transformation (BWT). The BWT transforms a text to a more compressable state, additionally creating an index of the text. This index can be used for a pattern search that is faster than on the original text. BWT-based aligners such as `Bowtie` [50] and `BWA` [54] facilitate rapid matching of a set of sequences to a reference genome by converting the genome to a compressed index and using error-tolerant modifications of the basic 'backward search' strategy to check for matches to individual query sequences. Here we will present the Burrows-Wheeler transformation as well as the search method published by Ferragina and Manzini in 2000 (*FM-Backward Search*) [22].

2.2.1. Burrows-Wheeler transformation

In 1994, Burrows and Wheeler proposed an algorithm for lossless compression of texts, today known as the Burrows-Wheeler transformation [12]. The BWT of a string $t = t_1 \dots t_n$, of length n containing characters of alphabet Σ, is defined by computing all n rotations of the string and ordering those alphabetically. For example the string $t = AGTAGTCA$ can be rotated to the strings $R(t) = \{AGTAGTCA, GTAGTCAA, TAGTCAAG, AGTCAAGT, GTCAAGTA, TCAAGTAG, CAAGTAGT, AAGTAGTC\}$. If those rotations are ordered alphabetically it will result in the matrix M shown in Figure 2.1. The last column of the matrix M is the BWT of the text,

M:

AAGTAGTC
AGTAGTCA
AGTCAAGT
CAAGTAGT
GTAGTCAA
GTCAAGTA
TCAAGTAG
TAGTCAAG

Figure 2.1.: Example of the matrix M with ordered permutations of the text $t = AGTAGTCA$. The first column (in blue) are the lexicographically ordered characters of t, the last column is the $BWT(t)$.

here $BWT(t) = CATTAAGG$. The first column contains the ordered characters of t,

in this example: $f = AAACGGTT$.

The BWT in itself is not a compression algorithm but the rotation results in a text which is more compressable than the original one. The BWT of a non random text contains longer stretches of the same characters than the original text. On genetic level a high compression is possible since base triplets encode for amino acids, for example the triplet *TAG* encodes for the amino acid *isoleucin*. A gene encoding for a protein containing a high amount of *isoleucin* will contain a high number of the triplet *TAG*. When the rotations of the gene sequence is sorted, *AG* will sort together, resulting in a long stretch of *T*s in the BWT of the gene. A text with longer stretches of one character can be compressed more efficiently using Huffman [38] or Run Length encoding.

It is possible to use suffix arrays instead of all rotations of a given string to compute the BWT. For this, a special character \$, which is lexicographical smaller than any $c \in \Sigma$, is appended at the end to the given text. This ensures that the alphabetically order of the suffixes of the text is the same as the ordered rotations of the text. While the order of the suffixes remains the same, the $BWT(t)$ changes through appending \$. BWT creation of a string t can be described in four simple steps.

1. The character \$ is appended at the end of the string t.
2. All $n + 1$ suffixes of t\$ are build.
3. The suffixes are ordered lexicographically.
4. The character occurring in t directly before the start of each suffix is taken to form the BWT transformed string $BWT(t)$.

Figure 2.2 shows an example of BWT creation of t\$ based on suffixes. Using this way to create the BWT it is easer to see that it shares useful characteristics with *suffix trees*.

The original text t can be gained from the $BWT(t)$ by exploiting certain characteristics of the BWT. The ordered characters of t give the string f, for $t = AGTAGTCA\$$ this would be $f = \$AAACGGTT$. An array $C[.]$, of length $|\Sigma| + 1$ can be obtained from the BWT, such that $C[c]$ contains the total number of characters in t which are lexicographically smaller than c. Therefore, $C[c]$ is also the index of the first occurrence of c in f. The matrix $Occ(c, q)$ denotes the number of occurrences of character c in the q long prefix $BWT(t)[1, q]$ of the $BWT(t)$ for any $c \in \Sigma$ and $q \leq n$. An example for $C[.]$ and $Occ(c, q)$ for the $BWT(t)$ is given in Tables 2.2 and 2.3 respectively. Using $Occ(c, q)$ and $C[.]$ it is possible to map the $BWT(t)$ to the ordered characters of t. The mapping can be used to obtain the text t from the $BWT(t)$ using array $C[.]$, matrix $Occ(c, q)$ and the ordered characters of t f, as can be seen in Algorithm 1.

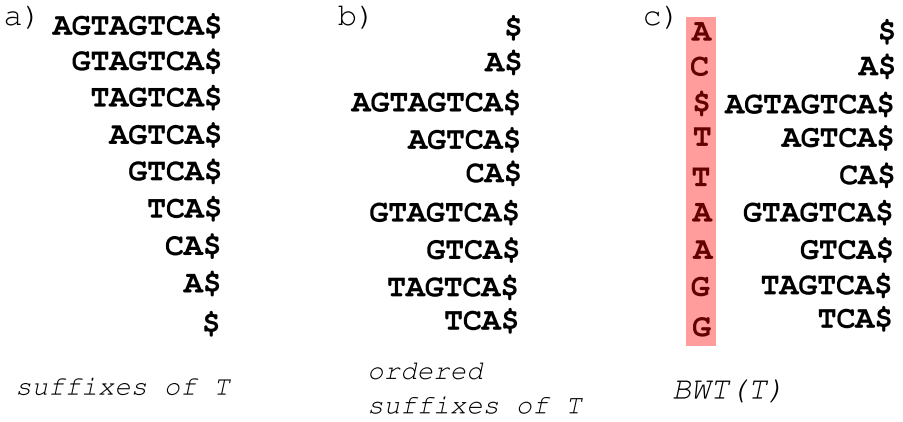

Figure 2.2.: Example how the $BWT(t)$ of the text $t = AGTAGTCA\$$ is created, after appending \$. a) shows all suffixes of t, b) shows the lexicographically ordered suffixes of t, part c) shows the $BWT(t)$ of t.

\$	A	C	G	T
0	1	4	5	7

Table 2.2.: Array $C[.]$ for the text $BWT(t) = AC\$TTAAGG$, where $C[c]$ is the number of characters in the whole string which are lexicographically smaller than c.

Occ(c,q)	1	2	3	4	5	6	7	8	9
\$	0	0	1	1	1	1	1	1	1
A	1	1	1	1	1	1	3	3	3
C	0	1	1	1	1	1	1	1	1
G	0	0	0	0	0	0	0	1	2
T	0	0	0	1	2	2	2	2	2

Table 2.3.: Matrix $Occ(c,q)$ for the text $BWT(t) = AC\$TTAAGG$, where $Occ(c,q)$ counts all occurrences of character c in the $BWT(t)$ before position q.

Figure 2.3 shows the single steps of gaining $t = AGTAGTCA\$$ from $BWT(t)$ for $BWT(t) = AC\$TTAAGG$ using $C[.]$ and $Occ(c,q)$.

Algorithm 1: Pseudo code to obtain t from the $BWT(t)$ using the array $C[.]$ and the matrix $Occ(c,q)$.

```
1 i ← 1 , t ← f[1], m = 0 ;
2 while (i ≤ n) do
3     m ← C[BWT(t)[i]] + Occ[BWT(t)[i], i];
4     c ← f[m];
5     t ← ct;
6     i ← i + 1;
7 end
```

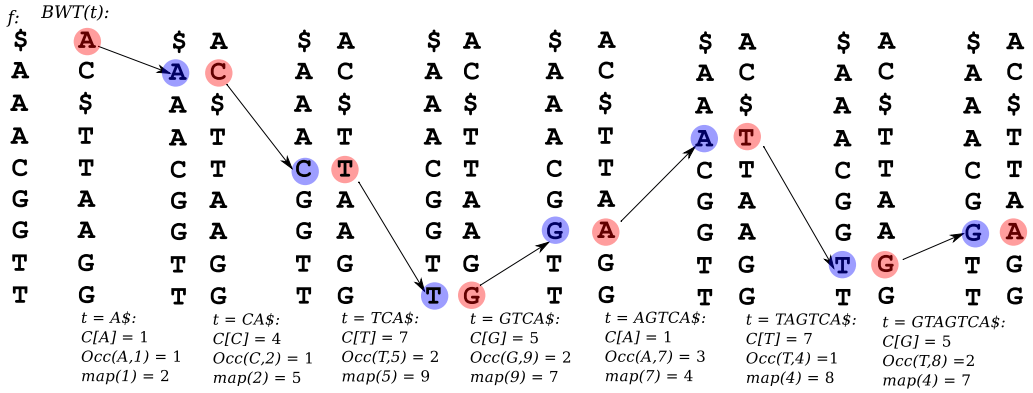

Figure 2.3.: Gaining text t through the $BWT(t)$ using the array $C[.]$ and the matrix $Occ(c,q)$.

2.2.2. Ferragina-Manzini backward search

The matrix $Occ(c,q)$ and the array $C[.]$ can also be used as index to search in the compressed text $t = t_1 \dots t_n$ for all occurrences of the pattern $p = p_1 \dots p_k$ of length k. The search algorithm starts at the end of the pattern and goes forward to the first character. It was first described by Ferragina and Manzini [22] giving the index the name *FM-index*. The algorithm exploits the following two properties of the matrix M:

i) since the matrix M of the text t is sorted and contains all possible suffixes of t, all possible occurrences of p occur in a continuous set of rows, also called a *Q-interval*.

ii) this set of rows has a starting position f_Q and an ending position l_Q, where f_Q is the *lexicographically smallest position* of the pattern p in the ordered column f.

The interval between f_Q and l_Q defines a contiguous stretch in the $BWT(t)$. Searching for pattern p in text t is done by finding the positions f_Q and l_Q of the Q-interval in matrix M, where $Q = p$. The value $l_Q - f_Q + 1$ gives the total number of occurrences of p in t.

To find l_Q and f_Q in k steps using only $Occ(c, q)$ and $C[.]$, the Algorithm 2 is used. The algorithm initiates with the last character of the pattern ($c = p_k$) and starts at the end of the pattern going to the beginning. At the first step of the FM backward search f_Q and l_Q are initiated with $f_Q = C[c] + 1$ and $l_Q = C[c + 1]$. $c + 1$ donates the character lexicographically following c, if c is the lexicographically biggest character $c + 1$ then $C[c + 1] = n$. For example searching for the pattern $p = AGT$ in

Algorithm 2: Pseudo code to find the number of occurrences of pattern p in text t using the indexes of t.

```
1  i ← k , c ← p[k], f_Q ← C[c] + 1 , l_Q ← C[c + 1];
2  while ((f_Q ≤ l_Q) and (i ≥ 1)) do
3      c ← p[i − 1];
4      f_Q ← C[c] + Occ(c, f_Q − 1) + 1;
5      l_Q ← C[c] + Occ(c, l_Q);
6      i ← i − 1 ;
7  end
8  if l_Q ≤ f_Q then
9      no rows prefixed by p[1, k]
10 else
11     return ⟨f_Q, l_Q⟩
12 end
```

the string $t = AGTAGTCA\$$ with the help of the $BWT(t)$ and the index $Occ(c, q)$ and $C[.]$ would include the following steps:

1. $i = 1$, $f_Q = C[T] + 1 = 7 + 1 = 8$, $l_Q = C[c + 1] = 9$ gives the interval $Q = [8, 9]$ of the matrix of suffixes starting with T.

2. Find those suffixes starting with G, where the following character is T: $i = 2$, $f_Q = C[G] + Occ(G, 7) + 1 = 5 + 0 + 1 = 6$, $l_Q = C[G] + Occ(G, 9) = 5 + 2 = 7$, interval $Q = [6, 7]$ of suffixes starting with GT.

3. $i = 1$, $f_Q = C[A] + Occ(A, 5) + 1 = 1 + 1 + 1 = 3$ and $l_Q = C[A] + Occ(A, 7) = 1 + 3 = 4$, interval: $Q = [3, 4]$ of suffixes starting with AGT.

Figure 2.4 shows the example of finding all occurrences of the pattern $p = AGT$ in the BWT of $t = AGTAGTCA\$$. This small example finds pattern $p = AGT$ in t two times ($4 - 3 + 1 = 2$). At each step of the search it is known at which point in

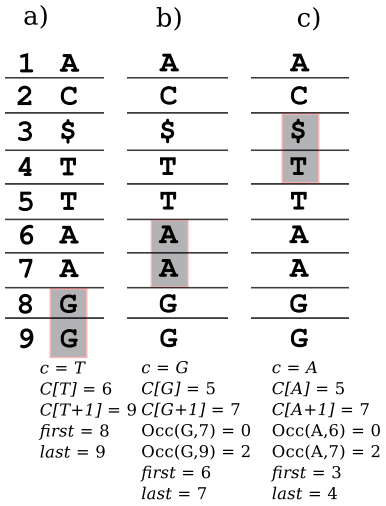

Figure 2.4.: Example of FM-Backward search for the pattern $p = AGT$ in the indexed text $t = AGTAGTCA\$$. **a)** shows the initial step of the algorithm, **b)** the first iteration of the loop and **c)** the last one. Grey boxes indicate the current Q-interval in the $BWT(t)$.

the $BWT(t)$ the suffixes can be found. Once the array $C[.]$ and the matrix $Occ(c, q)$ are computed, the time it takes to search for a pattern p depends on the length of p and not on the length of the searched text or the number of occurrences in of p in t. Furthermore, this pattern search does not need the original text anymore. Therefore, it can be done on compressed texts. During each step of the search the BWT has to be accessed depending on the results of the preceding step, leading to a random access of the BWT.

The FM backward search is used by mapping programs like `BWA` or `Bowtie`. They create the BWT of a reference sequence and use the index for fast comparison of sequence reads to the reference. Since the search needs random access to the BWT, either the BWT or $Occ(c, q)$ and $C[.]$ have to be kept in memory during the comparison. Therefore, the size of the reference sequence is limited. In the next section we will present an extension of the FM-Backward search, enabling the BWT to be remain on disk without missing any patterns.

2.3. metaBEETL - Taxonomic classification of whole genome metagenomic sequences

The BWT is often used to index reference sequences to achieve fast sequence comparison. Unfortunately, the size of the reference sequence is restricted, since it has to be

held in memory. Here we present the novel taxonomic classifier `metaBEETL` which was published 2013 at RECOMB-Seq [3]. `metaBEETL` is based on the sequence comparison program `BEETL` [4] and was created in collaboration with Illumina. It uses the all-against-all backward search of `BEETL`, which will be presented in Section 2.3.1 This method was adjusted for taxonomic classification of metagenome shotgun reads as presented in 2.3.2.

2.3.1. Simultaneous all-against-all backward search

Since the amount of sequenced reads is rising faster than the development of computational memory space it has become important to use new methods, which do not hold every needed information in memory, but effectively access the information on disk [44]. Sequential access to this information is essential for the containment on disk. Instead of the FM-Backward search, where certain patterns are searched in a text, we can find all occurring patterns P of length k in text t by going through the $BWT(t)$ k times. If done so to two or more text simultaneously, this provides the possibility to test of distinct or coexistent occurring patterns in both texts. This method was first used in the sequence comparison program `BEETL` and was adapted for the taxonomic classification of the sequences for `metaBEETL`.

Let t be the text to be searched, containing characters of the alphabet Σ, excluding the special character \$ that is included in Σ. $BWT(t)$ is the Burrows-Wheeler transformation of t\$. If all possible patterns $P = \{p_1 \dots p_{|\Sigma|^k}\}$ of length k are tested, it is possible to go through the index of t k times sequentially to find all occurring patterns of length k. The index can therefore be held on disk and does not have to be kept in memory. In difference to the original `BEETL`, in `metaBEETL` not all occurring patterns are tested but only those occurring in the sequence dataset and the comparison database. $O(\Sigma)$ defines the lexicographically ordered characters of Σ. The BWT can be divided into $|\Sigma|$ buckets $B = B_1 \dots, B_{|\Sigma|}$ because it corresponds to the lexicographically ordered suffixes of t. The characters in bucket B_i are in t directly followed by the character $O(\Sigma)_i$. Figure 2.5 shows an example for the division of the BWT of the string $t = AGTAGTCA\$$ into five buckets.

To describe the *all-against-all backward search* for all P we use the definition of Q-intervals, as defined in 2.2.2. To find all occurring patterns of length k, we need k iterations to go through the $BWT(t)$ k times. At iteration j let $Q_j = [f_{Q_j}, l_{Q_j})$ be the interval of an occurring jth suffix of at least one searched pattern p. Let cQ be the extension of the suffix Q with the preceding character c. To find all possible extensions of Q with any character $c_i \in \Sigma$ we can go sequentially through $BWT(t)$ with the help of the following method. f_{c_iQ} and l_{c_iQ} of c_iQ in bucket B_i can be obtained by the amount of c_i in range Q and the number of c_i in the $BWT(t)$ before the start of Q. Those numbers are acquired for all $c \in \Sigma$ while going once through

$B_\$$	$	A
B_A	A$	C
	AGTAGTCA$	$
	AGTCA$	T
B_C	CA$	T
B_G	GTAGTCA$	A
	GTCA$	A
B_T	TAGTCA$	G
	TCA$	G

Figure 2.5.: Division of the $BWT(t) = AC\$TTAAGG$ into the five buckets $B_\$, B_A, B_C, B_G$ and B_T, depending on the first letter of the ordered suffixes.

the $BWT(t)$. In the array r of length $|\Sigma|$ let $r[c_i]$ count all occurrences of c_i in the $BWT(t)$ before f_Q. To obtain the length of the interval c_iQ, the array $o[c]$ keeps the number of occurrences of all c in the interval Q of the $BWT(t)$. At iteration j those arrays are updated with the new counts for each c_i which occurs in Q. At the end of stage j the arrays contain all Q intervals for all $(j+1)$-suffixes, with $f_{c_iQ} = r[c_i]$ and $l_{c_iQ} = f_{c_iQ} + o[c_i]$. The algorithm starts with the search for the extension of suffixes of length one, which correspond respectively to one of the files. At each iteration the arrays $r[c]$ and $o[c]$ are updated and the extensions of the patterns are stored on disk in F files, where F_i contains Q-intervals of all $(j+1)$-suffixes starting with the character c_i in lexicographic order. In the next iteration the files $F_1, \ldots, F_2, \ldots, F_{|\Sigma|}$ are read sequentially to obtain the lexicographic order of the suffixes for the next iteration. The $BWT(t)$ is traversed sequentially and for each suffix the arrays are updated again and stored in F files. An example for this for the text $t = AGTAGTCA\$$ for all shared patterns with $k = 1$ can be found in Figure 2.6. Further iterations can be found in the appendix in Figures A.1 and A.2.

A Q-interval is not extended to cQ if no extension is possible or the only possible extension is \$. Using this, one can find all possible patterns of a certain length k in a Burrows-Wheeler transformed text by making k passes through the $BWT(t)$. To find all shared patterns of two texts, this can be done by making sequential passes through both BWTs simultaneously. Accessing the indexes in a sequential way is cache-efficient if one or both of the indexes do fit in RAM. More importantly it also makes it feasible to compare them while they are both held on disk, thus preventing available RAM from constraining the sizes of the indexes that can be compared. Moreover, indexing the query sequences exploits redundancy within them since each distinct pattern is compared with the reference index exactly once, even if it has multiple occurrences among the queries.

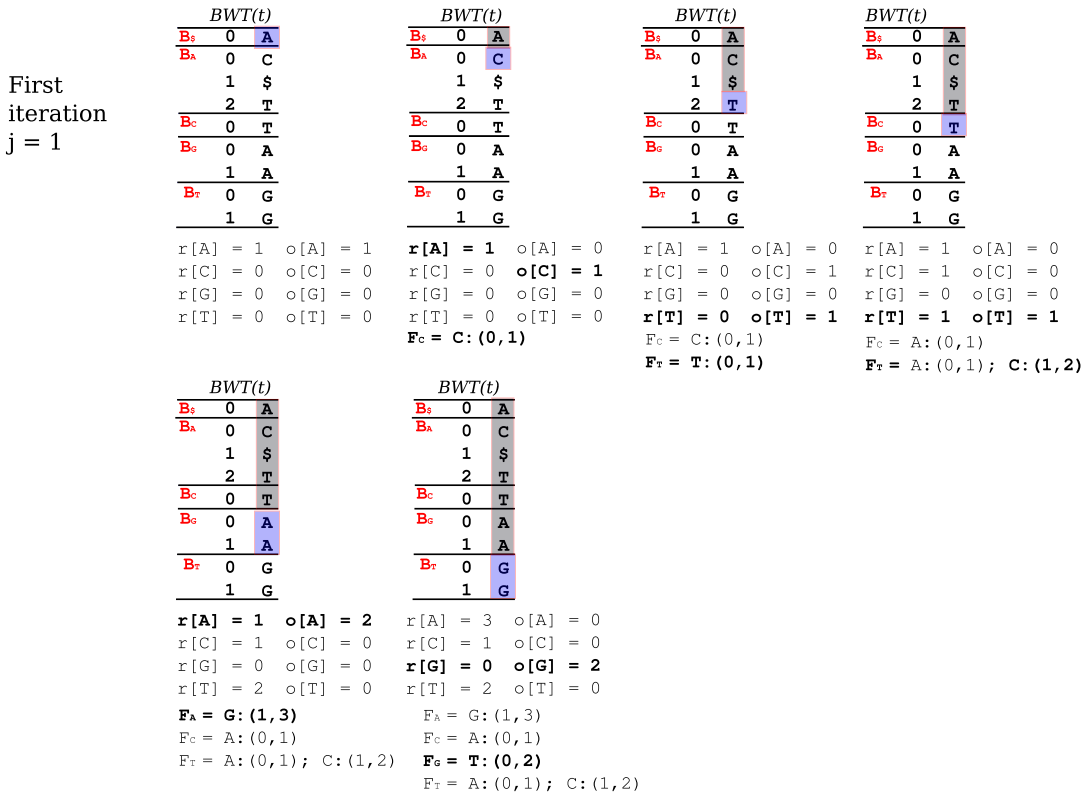

Figure 2.6.: First iteration of the all-against-all backward search. Grey boxes indicate the already gone through part of the $BWT(t)$, blue boxes are the part currently analysed. Array $r[c]$ contains the number of characters read in the BWT before the current section. Array $o[c]$ contains the numbers of character in one Q-interval. The newly found cQ intervals are stored in F.

2.3.2. Taxonomic classification using the all-against-all backward search

The all-against-all backward search was developed to compare two read sets, while the reads could be held in compressed state on disk. Here we present a method to use this search as a taxonomic classifier for compressed metagenome shotgun reads. For this a collection of reference sequences has first to be indexed as a reference database.

Database creation

To index a collection of sequences, first the concept of the BWT must be generalized from a single string to a collection of n texts. A straightforward way to do this is to imagine each member of the collection is terminated by a distinct member of a set of

special characters $\$_1, \ldots, \$_n$ that satisfy $\$_1 < \cdots < \$_n$ and are lexicographically less than any characters of Σ that the rest of the text is drawn from. A generalized BWT is built for the sequence collection $G = \{g_1, \cdots, g_n\}$. This is a 'one-time' procedure that only needs to be repeated when sequences are added to (or removed from) the collection, therefore simplicity is at this prioritized over efficiency: first suffix arrays for all members of G are built (which can be done in parallel), then they are merged, by reading the suffix arrays element-by-element from disk into a Fibonacci heap. A Fibonacci heap (a priority queue) is a collection of trees. In those trees the key of a child is always greater than or equal to the key of the parent [23]. Here the suffixes pose as keys, so that the lexicographic order of all suffixes can be found fast. Using copies of the sequences held in RAM, the relative ordering between suffixes from different members of the collection is determined. This enables us to build not only the generalized BWT but also to obtain the arrays A and C. Array A holds the original position of the suffix in the sequence, C holds the information of which sequence the suffixes originated from, such that the suffix at position $A[i]$ of member $C[i]$ of G is the i-th smallest suffix in the collection. Together, A and C form a *generalized suffix array* of G.

In `metaBEETL` G contains all genome sequences as well as their reverse complements of *bacteria*, *archeae* and *viruses* from the NCBI genome sequence database. The elements of C are used as keys for an array T of 8-vectors such that $T[i] =$ *{superkingdom, phylum, class, order, family, genus, species, strain}* describes the classification of the i-th member of G. Each member of the 8-vector is a taxonomic id according to the NCBI-taxonomy.

Classification

The search for shared patterns P of length k between the set of reads R and the genomes G is done as described in section 2.3.1. At iteration k, the Q-intervals of all k-mers that are present in either or both of $BWT(R)$ and $BWT(G)$ are considered in lexicographic order. Intervals found only in $BWT(R)$ or only in $BWT(G)$ are not extended. For each k-mer Q of a minimal length that is present in both R and G, we extract from C the subarray $C[f_Q], C[f_Q + 1], \ldots, C[l_Q]$ whose elements encode the origin of the symbols in the Q-interval $[f_Q, l_Q)$ of $BWT(G)$. For each level, starting with *strain* moving to *superkingdom* the corresponding taxonomic indexes $T[C[f_Q]][l] = T[C[f_Q + 1]][l] = \cdots = T[C[l_Q]][l]$ of the genomes are compared with each other. At each level only those taxonomic indexes are considered, where more than a 80% of the genomes $C[f_Q], C[f_Q + 1], \ldots, C[l_Q]$, share that classification, this excludes outliers in the classification. Therefore, the k-mer is classified to the deepest taxon in the tree of life that most of the originating genomes share as classification. Turning to $BWT(R)$, the size $f'_Q - l'_Q + 1$ of the Q-interval $[f'_Q, l'_Q]$ gives the number

of occurrences of Q in the reads. This results in a number of k-mers in the read set R with a known taxonomic classification.

Bias control

In `metaBEETL` two possible sources for bias in the data are considered. First the bias arising from the copy number variations is considered by removing k-mers that occur more than once in one genome from the taxonomic classification. For this the array $C[f_Q], \ldots, C[l_Q]$ is checked for entries occurring more than once before classification. Second, per-read statistics such as these must be normalized by genome size to obtain a statistic that reflects the relative abundance of microbial cells [75]. To achieve this, the occurrences of all k-mers specific to a given taxon are aggregated and then divided by the mean lengths of the genomes within that taxon. Further biases, like different sequencing depth or amount of sequences which can be classified should are also considered, either by division by the amount of reads sequenced or the amount of classified k-mers. The optimal k for a given experiment is determined empirically and depends on the accuracy and length of its reads: the greater specificity of longer k-mers is weighed against the fact that sequencing errors and genomic variations cause fewer reads to be classified as k becomes close to the read length. K-mers as short as 10 bases are more likely to appear in so many genomes that the only possible classification is at superkingdom level.

2.4. Accuracy tests of metaBEETL

Here we will present the results of the accuracy test of `metaBEETL`. An artificial whole genome metagenome sequence dataset was generated to test accuracy. The classification of this dataset from `metaBEETL` was compared to the classification from `CARMA`, `MEGAN` and `Genometa`.

2.4.1. Reference database

We downloaded the set of all NCBI RefSeq microbial sequences[1] and the associated NCBI taxonomy[2] on October 2nd 2012. This comprised 2097 genomes from *bacteria*, viruses and *archaea*, from which plasmid sequences were excluded to reduce the possibility of wrong taxonomic profiles through bacterial conjugation and copy number variation of plasmids in different microbes. The BWT and generalized suffix array of the remaining 2020 sequences and their reverse complements were generated as described in Section 2.3.2.

[1] ftp://ftp.ncbi.nlm.nih.gov/genomes/Bacteria/
[2] ftp://ftp.ncbi.nih.gov/pub/taxonomy/

2.4.2. Accuracy test on a simulated metagenome

To test the accuracy of a taxonomic classifier, a dataset is needed consisting of reads with known taxonomic classification. Therefore, we tested `metaBEETL` with a simulated metagenome dataset, comparing the classification results with three other classifiers. We simulated a metagenome containing equal proportions of microbes from fifteen organisms whose genomes are present in the NCBI Nucleotide database, having genome sizes ranging from 0.2 Mbp to 11 Mbp with an average of 3.3 Mbp. An overview of those genomes and the simulated reads can be found in Table 2.4. `MetaSim` [75] was used to simulate 100,000 Illumina read pairs of length 80 bp.

Name	Taxonomic id	Size	Fraction in simulation	Read count
Blattabacterium sp. str. BPLAN	600809	0.64 Mb	6.67 %	2372
Borrelia hermsii DAH chromosome	314723	0.92 Mb	6.67 %	3616
Candidatus Blochmannia pen. str. BPEN	291272	0.79 Mb	6.67 %	3122
Candidatus Sulcia muelleri DMIN	641892	0.24 Mb	6.67 %	3122
Candidatus Zinderia insecticola CARI	871271	0.21 Mb	6.67 %	816
Catenulispora acidiphila DSM 44928	479433	10.47 Mb	6.67 %	41950
Chloroflexus aggregans DSM 9485	326427	4.68 Mb	6.67 %	18684
Clostridium sp. BNL1100	755731	4.61 Mb	6.67 %	18248
Deinococcus radiodurans R1	243230	3.06 Mb	6.67 %	12066
Escherichia coli DH1	536056	4.63 Mb	6.67 %	18400
Fluviicola taffensis DSM 16823	755732	4.63 Mb	6.67 %	18258
Frankia sp. CcI3	106370	5.43 Mb	6.67 %	21282
Geobacter bemidjiensis Bem	404380	4.61 Mb	6.67 %	18344
Mycoplasma pneumoniae M129	272634	0.82 Mb	6.67 %	3286
Yersinia enterocolitica subsp. e. 8081	150052	4.62 Mb	6.67 %	18548

Table 2.4.: Composition of simulated metagenomic dataset: An even distribution of microbes was simulated.

	metaBEETL	CARMA3	MEGAN	Genometa
Memory	1 GB	13 GB	13 GB	3 GB
Time	46 min	18 h 35 m	14 h 58 m	2 min

Table 2.5.: Running time and memory requirements of the tested classifiers on the simulated data set. CARMA3 and MEGAN were run on a compute cluster, using 100 nodes. Genometa was run on a laptop with four CPUs available. metaBEETL was run on an SSD drive. Memory consumption was taken at peak memory usage for one thread. All times are taken as wall clock times. For CARMA3 and MEGAN the time for the longest running time of the 100 threads was taken, the average time for MEGAN was 12h 15m and for CARMA 12h 30m.

While this number is only a fraction of an actual sequencing run, its size was chosen to allow the BLASTX alignments needed by MEGAN and CARMA3 to finish in reasonable time on the hardware available to us.

Comparison of computational costs

We used CARMA3 [25] and MEGAN 4.0 [40] as the most recent versions of the programs. Aligning the reads to a set of reference sequences dominates the computational cost of MEGAN and CARMA3. Of the configurations tested in [25], aligning the reads to the NCBI NR database with BLASTX maximized the number of reads correctly classified by both programs, so we did the same with our data. These alignments were calculated on a cluster of 100 nodes, each node having at least 124GB memory available. The number of cores per node varied between 2 to 48, each having a clock speed of 2.0GHz. Genometa and metaBEETL both ran on a single CPU Intel Xeon machine having eight 3.0GHz cores and 64Gb of shared RAM, to which we had sole access for our tests. metaBEETL needed only 200Mb of RAM but its index of reference genomes and its temporary files were stored on an attached solid-state hard drive to facililate the large amount of disk I/O that metaBEETL needs to do.

Timings for the four methods are given in Table 2.5: The very different computational requirements of the BLAST-based and BWT-based tools make a like-for-like comparison difficult, but the advantage of the BWT-based methods is clear: metaBEETL finishes an order of magnitude more quickly on a single CPU than the BLAST-based methods do on a 100 node cluster.

Genometa, whose compute time is predominantly taken up by BWA alignments, is in turn an order of magnitude faster than metaBEETL, but our prototype implementation has considerable scope for optimization. At the moment, the reference BWT string is stored as ASCII, whereas a compressed format would greatly reduce

Taxonomic Level	metaBEETL		CARMA3		MEGAN		Genometa	
	TP	*FP*	*TP*	*FP*	*TP*	*FP*	*TP*	*FP*
Super-kingdom	64.64	0.00	80.58	0.08	89.35	0.00		
Phylum	64.64	0.00	79.45	0.19	88.30	0.02		
Class	64.64	0.00	78.77	0.19	87.86	0.02		
Order	64.64	0.00	77.81	0.11	87.37	0.02		
Family	64.63	0.00	75.84	0.18	85.61	0.05	56.56	2.60
Genus	64.63	0.01	66.13	0.26	75.65	0.32	54.94	4.22
Species	64.62	0.02	25.96	0.12	55.36	0.6	54.72	4.45

Table 2.6.: Comparison of the percentage of correctly classified (true positive - TP) and incorrectly classified (false positive - FP) reads of the simulated metagenome between the classifiers, metaBEETL, CARMA3, MEGAN and Genometa.

the I/O that dominates metaBEETL's runtime. Moreover, it is likely that any given sample will contain only a small proportion of the 2020 genomes that are present in the database. Therefore, indexing the BWT string of the reference database should reduce I/O still further by allowing metaBEETL to jump directly to the relevant areas of the BWT instead of reading the entire string on every pass.

Comparison of accuracy of the taxonomic profiles

CARMA3, MEGAN and Genometa were run with default parameters and metaBEETL was run with a *k*-mer length of 50. Table 2.6 shows the percentage of reads correctly and incorrectly classified by the four tools at all taxonomic levels. Overall metaBEETL classified 129,290 reads, CARMA 161,315 reads, MEGAN 178,717 reads and Genometa 118,340 reads. The smaller number of reads classified by metaBEETL compared with CARMA and MEGAN is likely explained by metaBEETL's discarding of *k*-mers occurring multiple times in a reference genome. Genometa requires a curated database (only one reference per genus, for instance) and we thus used the database provided by Genometa[3]. We manually checked that all the genomes used in the simulated sample were contained in this database. Importantly, metaBEETL is the best of the four tools in correctly classifying reads at the species level and misclassifies the fewest reads at all taxonomic levels.

An obvious way to assess the performance of a metagenomic classifier is simply to count the number of correctly classified reads, but we have already observed that copy number changes and different genome sizes can prevent the relative read

[3]http://genomics1.mh-hannover.de/genometa/index.php?Site=Download

Taxonomic Level	metaBEETL	CARMA3	MEGAN	Genometa
Super-kingdom	1.0	1.0	1.0	–
Phylum	7.47	22.44	22.89	–
Class	7.48	25.70	23.84	–
Order	9.45	24.26	24.23	–
Family	9.39	22.15	19.60	–
Genus	10.85	26.22	21.56	38.82
Species	10.59	19.02	22.44	38.16

Table 2.7.: Comparison of the simulated taxonomic profile of an artificial metagenome and the predicted profiles from metaBEETL, CARMA3, MEGAN and Genometa. We compared profiles using the Euclidean distance to the simulated profile. Results from Genometa were only available at levels genus and species.

counts from correctly reflecting the relative abundances of the microbes they are sequenced from. For this reason we decided not to perform comparisons solely based on the number of classified reads but also based on the expected taxonomic profile. We used the Euclidean distance $\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$ to compute the distance between computed and simulated taxonomic profile. Here q_i is the percentage of simulated reads in taxon i and q_i the percentage of reads classified to taxon i. The distance of the computed profiles to the simulated profile using the Euclidean distance can be found in Table 2.7.

We can see that metaBEETL produces a taxonomic profile closer to the simulated ground truth than the other classifiers. For Genometa, we could only generate the taxonomic profiles at the genus and species levels, because Genometa does not produce higher level taxonomic classifications. The taxonomic profiles for the levels phylum to species can be found in the appendix Figures A.3 to A.8. The difference in the classification originates from under- as well as overestimation of different taxa from CARMA3, MEGAN and Genometa. The more accurate classification from metaBEETL derived from the new bias control it is using. For CARMA3 and MEGAN the differences in classification can not be explained by the difference in the comparison databases, because the NCBI NR database includes the reference sequences used by metaBEETL. On the other hand the difference in database may be the cause for the higher distance from the true profile to the one produced by Genometa. By using a curated database with a limited size, Genometa loses the advantage of having a broader spectrum of references. Therefore, even though Genometa was much faster in the analysis of the simulated metagenome, the resulting taxonomic profile shows the greater differences to the simulated profile. metaBEETL performed much faster

Taxonomic Level	$k-mer$ 25	$k-mer$ 40
Superkingdom	1.0	1.0
Phylum	45.73	50.60
Class	46.56	51.56
Order	45.05	50.51
Family	11.38	29.70

Table 2.8.: Comparison of the simulated taxonomic profile of an artificial metagenome and the predicted profiles from `metaBEETL` against a modified database. Results of two k-mer sizes are shown.

than `CARMA` and `MEGAN`, while using less memory, which makes it possible to analyze large whole genome metagenome data sets. That `metaBEETL` is nearer to the true taxonomic profile shows that the bias reduction through removing sequences occurring more than once in the genome and the normalization for genome size gives `metaBEETL` an advantage over other classifiers.

2.4.3. metaBEETL accuracy test on a modified database

A key challenge of metagenomic studies is the fact that the majority of microbes cannot be grown as a single culture. This leads to the problem that reference data used for taxonomic classification is probably lacking the genomes of the microbes naturally occurring in the environment. To test the accuracy of the taxonomical classification of `metaBEETL` in the absence of the direct reference sequences, we masked our reference database at a certain level in the tree of life. To test how accurate the classification would be if a small branch of the taxonomic tree is missing, we masked all microbial reference sequences in the database that share the same genus as the microbes in the simulated metagenome. In this test no classifier can give the correct taxonomic profile on the levels of strain, species or genus, but the taxonomic profile on higher levels still need to be as close to the correct one as possible. For comparison with the correct profile we also used the Euclidean distance. `metaBEETL` produced taxonomic profiles with a distance ranging from 11.38 to 50 depending on taxonomic level and chosen k-mer size, see Table 2.8. The test on the modified database showed that the accuracy of `metaBEETL` got higher the more reads were simulated.

2.5. Discussion of metaBEETL

In this chapter we presented `metaBEETL`, an algorithm for the taxonomic classification of sequencing reads from whole genome metagenomic experiments. `metaBEETL` relies on indexed representations of both the input reads and the reference

genomes for fast comparison. We demonstrated on real and simulated data that its performance is competitive to BLAST-based metagenomic classifiers such as `CARMA3` and `MEGAN`, while scaling better to the large data sets generated by next-generation sequencing technologies. `metaBEETL` was presented at RECOMB-Seq 2013.

Like Genometa, `metaBEETL` relies on BWT-based text indexing, but there are fundamental differences in the two approaches. Genometa uses standard read mapping tools to perform its alignments, meaning its overall runtime is faster. However, the BWA and Bowtie aligners both have upper limits of around 3 Gb on the total volume of reference sequence that they can index, which will become an issue as the number of available bacterial genome sequences increases. Moreover, this reliance also means its ability to handle ambiguous matches is limited: a strain from each species must be hand-chosen to be added to the index as an exemplar of that species. In contrast, the bespoke nature of our BWT index allows us to distinguish between different strains and to assign reads to a higher phylogenetic order when a strain-specific match is not possible. Since the k-mer analysis of the sequence data shows similarity to the composition based methods, a comparison with those classifiers could also be interesting.

In many ways, our current implementation does not fully exploit the information present in the indexes. Instead of relying on an empirically chosen k-mer size, a future version could aggregate information from multiple values of k to continue to extend only those sequences that are not yet long enough to be specific at the strain level. Moreover, k-mers that are specific to a given strain can be used to identify novel variants within that strain.

Chapter 3

Metrans - a software platform for the analysis of metatranscriptomes

Metagenomic research gives insight into the taxonomic composition as well as the functional potential of microbial communities. However, it does not distinguish between expressed and non expressed genes. In *metatranscriptomics* the activity levels of members of a microbial community are researched by analyzing transcribed DNA.

Here we present `Metrans`, a software platform for the analysis and comparison of metatranscriptomes. The biological functions of ribonucleic acid (RNA) in microbial cells and possible analysis of RNA will be presented in Section 3.1, including an introduction to metatranscriptomics. The analysis pipeline and the software specifications of `Metrans` will be presented in Section 3.2. Finally we present an analysis example in Section 3.3 and discuss the results in Section 3.4.

3.1. Transcription analysis

Transcription of DNA into RNA is an essential step to produce proteins in a living cell, as introduced in Section 1.1.1. The stretch of DNA transcribed into RNA is called a *transcription unit* and encodes for at least one gene. If the transcribed section contains at least one gene coding for a protein, the RNA is called *messenger RNA* (mRNA). Other RNAs are called *non-coding* RNAs. To this group belong micro RNAs, lincRNAs, ribosomal RNAs (rRNA), transfer RNAs (tRNA) and RNAs coding for ribozymes. Most of these non-coding RNAs are folded in bigger RNA molecules. Folded RNAs are more stable than mRNAs, since those have no base pair bonds for stabilization [20]. Of the total RNA of an active cell, rRNA occurs in highest amount [29].

The isolated RNA of a cell is used to study the activity levels and the transcriptional responses to changes in the environment. These studies are called *RNA expression profiling* or *Transcriptomics*. The term *RNA-Seq* is used if NGS techniques are exploited to study the RNA expression of a single organism. In RNA-Seq the whole RNA of a mono-culture is isolated and sequenced. In difference to RNA-seq, in *metatranscriptomics* the RNA of a whole microbial community is studied. The main goal of analyzing the RNA of environmental organisms is to detect active organisms and to study the activity levels of specific metabolic functions in the community.

The main goal of metatranscriptomics is to research the activity levels of microorganisms and track changes in gene expression compared to shifts of environmental variables. For this the amount of specific mRNA in the microbial cells is determined. Since the transcription of a gene is the first step in gene expression, the amount of mRNA in the microbial cells can be used as an indicator for metabolic activities.

Commonly *DNA microarrays* are used for RNA expression profiling. A DNA microarray, also known as *DNA chip* or *biochip*, is a collection of DNA spots attached to a solid surface. Each spot contains 10^{-12} moles of specific DNA sequence as a *probe*. The nucleiacid sequences in the analysis sample are labeled with a molecular marker. Sequences hybridizing to the probe are called *targets*. Their relative abundance in the sample is determined by their hybridization to the probe. Typically microarrays are used to study the transcriptional reaction of single organisms [83]. However, there are some microarrays available to study the transcription in whole microbial communities [98, 102, 11]. Each of those arrays is designed for a specific environment. Unfortunately microarrays can only be designed if the reference sequence for an organisms is known. This is the main disadvantage of microarrays due to the fact that a large number of microorganisms are not yet sequenced. Since the development of NGS methods, RNA expression profiles can be studied by sequencing the isolated RNA after transcription into complementary DNA (cDNA).

The first NGS based metatranscriptome was obtained from a soil sample and revealed that *archea* are much more active in *amino oxidation* than the more studied *bacteria* [52]. In the last years a growing number of metatranscriptome studies have been published [34, 71, 55, 80]. The isolation of environmental RNA is quite challenging therefore the amount of gained RNA is often not enough for sequencing. A solution for this is *Multiple Displacement Amplification* (MDA). This technique is used to multiply small amount of genetic material in a sample the same way it is used for single cell sequencing, as described in Section 1.1.4.

To detect expression levels of genes, a high amount of mRNA in the sample is needed. There are protocols available to enrich the amount of mRNA in the sam-

ple, this is called *mRNA enrichment* or *rRNA depletion* [33]. Metatranscriptomes sequenced without prior mRNA enrichment can contain up to than 95% rRNA [86]. The rRNA sequences can not be used to study activity levels of certain metabolic functions, however they can be used to identify active organisms. Since an active cell has more ribosomes than an inactive one, more rRNA from active cells will be sequenced than from inactive ones. Using this to find out active taxa is only possible if no mRNA enrichment was done since those methods favor certain taxa.

Since metatranscriptome sequences originate from a mixture of different and often unknown organisms, it is not possible to analyze the sequences by mapping them to certain genomes, like it is done with RNA-Seq data. Therefore, sequence reads must be classified functionally and taxonomically by comparison to sequences with known classification. For this the quality of the sequences must be very high, since sequencing errors are not as easy to detect as in the mapping. To compare different datasets the sequencing depth also has to be considered. Since longer genes produce longer transcripts, the results of both classifications need to be normalized by gene length. For further normalization, sequencing depth as well as the amount of rRNA in the dataset have to be considered.

So far analysis of sequenced metatranscriptomes was either done by hand or relying on the web server *Metagenomic Rapid Annotations using Subsystems Technology* (`MG-RAST`). `MG-RAST` is an open source web service for functional and taxonomic analysis of metagenomes [62]. `MG-RAST` is based on the *SEED framework*[1]. SEED is a collection of genome sequences that are annotated with the RAST technology. In MG-RAST metagenomic sequences can be uploaded in fasta or fastq format and are automatically analyzed. Analysis tools are based on a BLAST comparison against the NCBI-NR database. The result counts are normalized by the total number of results. `MG-RAST` offers a number of different visualizations and comparison with openly available metagenome dataset. Using `MG-RAST` for metatranscriptomes has the disadvantage that result counts are not normalized by gene length, making it harder to compare the expression of long and short genes. An additional problem is, that `MG-RAST` will not analyze duplicate sequences. Therefore, only a small fraction of the sequences will be analyzed if the RNA had to be multiplied by MDA. `MG-RAST` also does not offer to filter non-coding RNAs.

The analysis of a metatranscriptome dataset includes quality control, test of sequencing depth, removal of non-coding RNA sequences, as well as the construction of taxonomic and functional profiles. Comparing the taxonomic results of a metatranscriptome and a metagenome can differentiate between the overall activity levels and the number of certain taxa in one community. Correlation of the results

[1] http://www.theseed.org/wiki/Main_Page

of several metatranscriptomes from the same environment with changed conditions shows the reaction of the microbial community to these changes. Comparability of the taxonomic and functional profiles requires specific normalization steps.

3.2. Metrans - analysis pipeline and software structure

`Metrans` is an open source software platform for the easy analysis of metatranscriptomes. The software has an user interface for the start of the analysis and the visualization of the analysis results. The analysis is done by an automated pipeline, presented in this section. Further, an overview of the data storage, software architecture, user interface and available visualizations will also be presented here.

3.2.1. Metrans analysis pipeline

Figure 3.1 shows an overview of the single steps as well as the corresponding visualizations of intermediate results in `Metrans`. Each of the following paragraphs

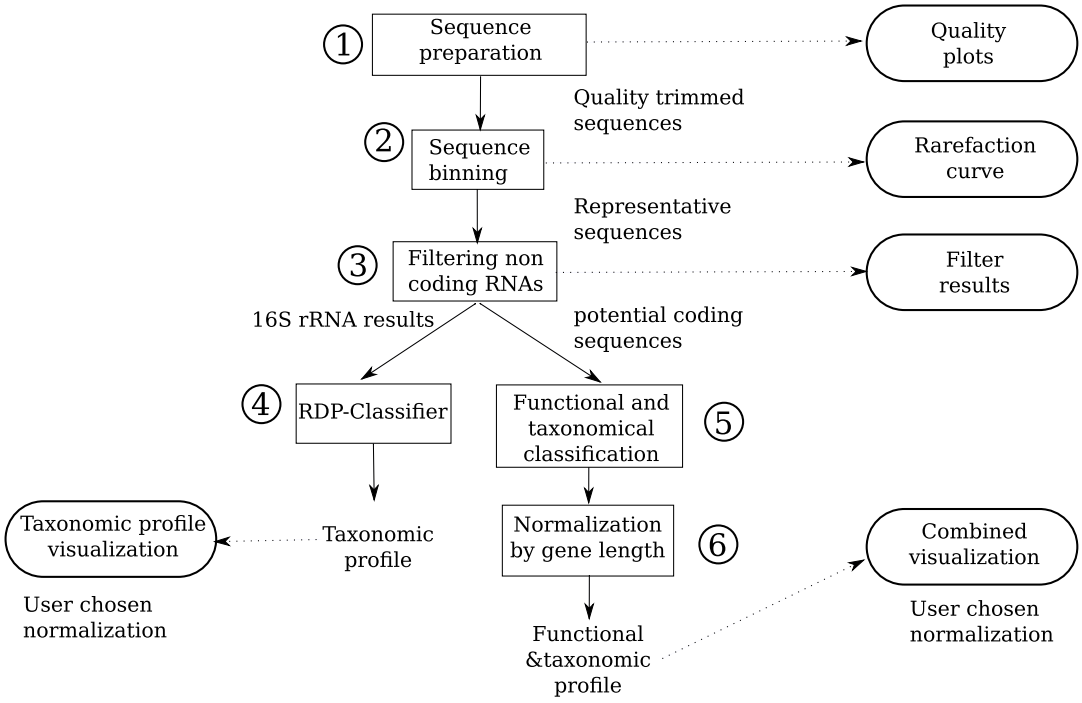

Figure 3.1.: Overview of the pipeline steps (rectangles) and the corresponding visualizations (ovals) in `Metrans`.

will first concentrate on the conceptual description of the corresponding analysis step and then on its implementation in `Metrans`. The parameters of all of the steps

can be configured by the user before executing the program. Default parameters for all tools have either been tested with good results or were taken from the literature. Single steps of the pipeline can be easily adjusted for an individual analysis.

Sequence preparation

The first step in any kind of DNA sequence analysis is to ensure the quality of the sequenced reads (Figure 3.1, step **1**). Since sequence quality changes greatly depending on the position within the read, a sliding window approach is often used for quality trimming. The window slides over the read, and if the average quality within the window drops below a certain threshold, the read is cut at the beginning of the sliding window.

In `Metrans` this quality trimming is performed immediately while loading a metatranscriptome dataset in fastq format. The size of the sliding window, as well as the quality threshold, based on the standard Phred-33 format, are set by the user. If the read length is shorter than the sliding window, the average quality of the read is used. Reads shorter than 35 bp are excluded from further analysis. The reads are stored in binary format in order to reduce disk space by 50 percent.

Sequence binning

The next analysis step is testing whether an adequate sequencing depth was reached (Figure 3.1, step **2**). If the sequencing depth is not deep enough, low expressed genes will not be found [27]. Binning the sequences according to sequence similarity will result in a number of bins for each expressed gene. Whether sufficient sequencing depth is reached, can be seen by comparing the amount of singletons (bins with only one sequence) to the amount of all sequenced reads. Due to the fact that sequencing errors occur only in a small amount of reads in Illumina or 454 sequencing datasets, the binning can also be used to find those sequencing errors [101]. Using the representative sequence of each bin in further processing, errors will have less impact on the further analysis.

`Metrans` employs the binning program *dnaclust*, developed by Ghodsi *et al.* (2011). It uses a greedy algorithm, employing a k-mer filter to avoid a high amount of computationally intensive alignments. To create a multiple alignment of the sequences in one bin, *dnaclust* uses the center star method with the longest sequence chosen as the center sequence. *Dnaclust* has been shown to be fast and reliable. To ensure better binning, the user can divide the sequences to be binned according to their length into several subsets. This ensures that many short sequences binned with a longer one will not reduce the length of the representative sequence. The bins are stored by saving the representative sequence of each bin. Individual reads are saved

then by storing their variations (mismatches, deletions and insertions) from the representative sequence, see Figure 3.2 as example. Representative sequences and the variations are stored in a binary format to reduce space further. This reduces disk

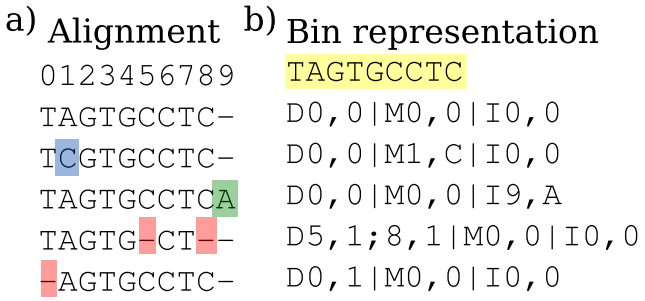

Figure 3.2.: Bin representation in `Metrans`. The alignment (left) of the sequences in the bin contains three deletions (red), one mismatch (blue) and one insertion compared to the representative sequence of that alignment. On the right the representation of the Bin in `Metrans` is shown. Representative sequence as well as mismatches, deletions and insertions according to the representative sequence are stored.

storage notably, depending on the success of the binning step.

Filtering non-coding RNAs

One of the most important steps in metatranscriptome analysis is the removal of sequences with no functional coding (Figure 3.1, step **3**). Doing this early in the analysis pipeline has several advantages. First of all, the comparison of the dataset with ribosomal RNA gene databases is faster than the comparison to a database containing protein coding genes. Firstly, protein coding databases are bigger in size than databases containing non-coding RNAs. Secondly, they contain amino acid sequences; therefore the reads have to be translated for the comparison. The second advantage for an early filter step is that further analysis will be accelerated by removing ribosomal RNA sequences since they are highly abundant in metatranscriptome datasets without prior depletion of rRNA. The third reason is the occurrence of annotation errors in sequence databases containing protein sequences. It could be shown that even well curated databases contain wrongly annotated sequences [84]. A wrongly annotated ribosomal RNA can have the effect that a metabolic function is falsely predicted as highly expressed in the microbial community. On the other hand, false positive classifications as non-coding RNAs during the filtering could lead to missing functional classifications in the subsequent analysis. Therefore, relatively relaxed parameters in the filtering step should be used, since sequences that are not identified as rRNAs in the dataset can be detected later and removed accord-

ingly.

The filtering of non-coding RNAs in Metrans is accomplished by the comparison of the representative sequences of all bins to databases of known non-coding RNAs using BLAST [2]. These databases like LSU, SSU [68] and RFAM [30] can be either loaded as a fasta file or as a blastable database. The user also has the option to translate the databases and use BLASTX for the comparison. This results in a higher amount of sequences being filtered out. The user can choose BLAST thresholds like E-value and identity as well as the minimal alignment overlap with the representative sequence. Sequences identified as non-coding RNAs will be tagged and will not be considered in the subsequent analysis.

RDP Classifier If the mRNA of the metatranscriptome is not enriched the 16S rRNA content of the sequence reads can be used to gain a taxonomic profile of the active organisms in the microbial community. In Metrans this is done using the RDP Classifier [96] (Figure 3.1, step **4**). The results of the comparison against the small subunit database are used for this purpose. Sequences similar to the 16S rRNA sequences are taxonomically classified using a naïve Bayesian classification. This taxonomic profile can be compared to the taxonomic profile from the functional analysis for sanity checking.

Combined functional and taxonomical analysis

After the filtering of potential non-coding RNAs, the remaining sequences are analyzed further to create taxonomic and functional profiles of the metatranscriptome (Figure 3.1, step **5**). Both types of profiles are connected with each other for data representation and visualization. It is important to start this step of the analysis with a database that is highly manually curated, because of possible misannotations. Further in the analysis, other less curated databases can be used.

For the *functional classification* of the representative sequences, Metrans uses BLAST-X [2] as comparison tool with functional databases. The user can choose parameters like E-value, minimal sequence overlap and identity as cutoff criteria of hits considered in the classification. For each sequence all hits above those cutoffs will be considered. The descriptions of the corresponding database sequences are obtained using the program *fastacmd* and are searched for certain keywords like EC-numbers, COG-categories or SwissProt-identifications. If a description does not contain any keywords, the whole description will be taken into account. The read is classified with the most frequently occurring functional assignment in the result list. The result count for a function is the aggregated number of sequences in the bins where the representative sequence was classified to this function.

For simultaneous *taxonomic classification*, the BLAST results are mapped back to the taxonomic information from the NCBI Taxonomy [82]. The description of the sequence in the databases contains individual annotation for the organisms it originated from. Therefore, the back mapping from the encoded name of the organisms to the NCBI Taxonomy is integrated individually for each database. In `Metrans` the taxonomical classification for the following reference databases are integrated: SwissProt [9], KEGG [45], PFAM [74] and EggNog [72]. Of all organisms found in the taxonomic assignment of a specific function, the *lowest common ancestor* will be assigned as taxonomic classification to the representative sequence. The taxonomical classifications of those reads are combined to a taxonomic profile.

Functional as well as taxonomical results are normalized by the length of the gene from which the classification originated (Figure 3.1 step **6**). This normalization is based on the RPKM-value (reads per kilobase per million), which is used in RNA-Seq experiments. Further normalization, by the read count of the metatranscriptome or by the total number of results can be chosen during visualization. The classified reads are tagged with an individual id, indicating the classifier and the classification. For custom analysis the user can load own databases or sequence files and use the graphical user interface to create a functional profile.

3.2.2. Data storage and operating system

Data representation in `Metrans` is project based. A project contains a number of metatranscriptome datasets. Each project requires a physical folder on disk, where metatranscriptome reads and analysis results can be stored. The representative reads of the metatranscriptome datasets are stored in binary format in order to reduce disk space. `Metrans` combines file- and database-based storage of analysis results and read information. This enables fast access to the analysis results, while offering the possibility to obtain reads responsible for the results. The results of the analysis are stored either in a H2[2] or in a MySQL[3] database. The database schema for the storage of the results can be seen in Figure 3.3. Storing the results in a H2 database has the advantage that the whole project can easily be exported to other machines. Although most tools used in `Metrans` depend on a Unix- or Linux-based platform, `Metrans` runs platform independently and analysis results can be viewed on a machine with other operating systems, once the analysis is finished. Since the reads are tagged with the identifier of the profiles, single reads can be obtained according to the tags of the results.

[2]http://www.h2database.com/html/main.html
[3]http://www.mysql.de

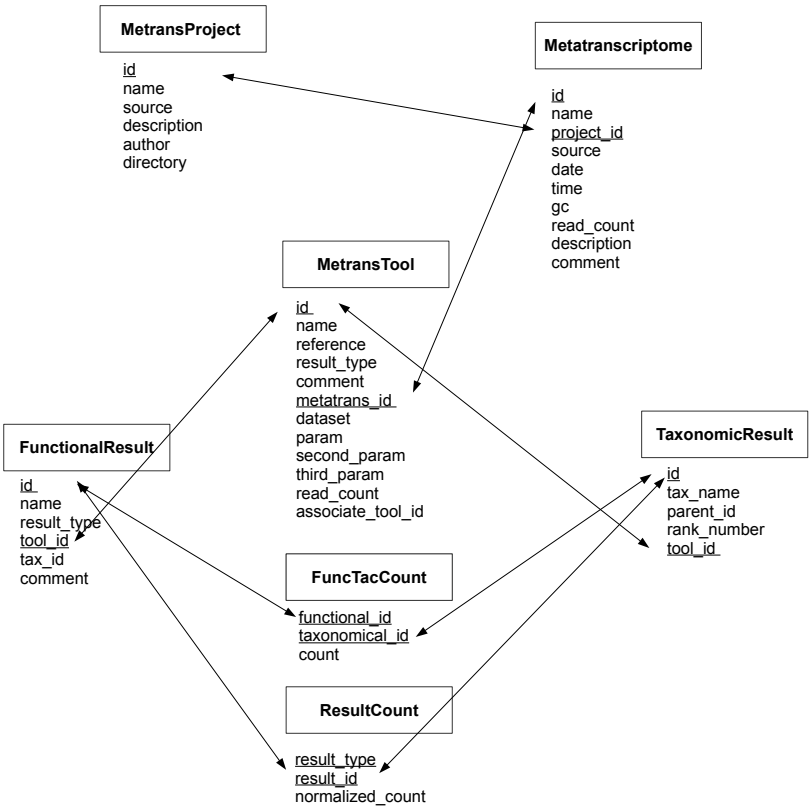

Figure 3.3.: Database schema of Metrans. Arrows indicate the association of the different tables by the key of the tables. The underlined words are the keys.

3.2.3. Software architecture

`Metrans` is fully implemented in Java, which has the advantage that it is portable to various operating systems. The visualization is based on *JFreeChart*[4] and the *prefuse* library[5]. `Metrans` is based on the *NetBeans Platform*[6], which allows a modular software design. New modules, like different classification tools or new visualizations, can easily be integrated in `Metrans` by implementing the corresponding interfaces. The extension to existing interfaces is loaded automatically without updating other modules. For parallelization, tools started from the `Metrans` platform can either be distributed to a *Sun Grid Engine* or run on a user chosen number of CPUs on a multi core machine.

[4]http://www.jfree.org/jfreechart
[5]http://prefuse.org/
[6]https://netbeans.org/features/platform/

3.2.4. Graphical user interface

Metrans was built for biologist users with a minimal background in computer science. Thus the primary goal of the development was to provide good usability. Figures 3.7 to 3.6 give an overview of single aspects of the user interface.

Projects, metatranscriptome datasets and available analysis results are presented in a tree like structure in a clipboard on the left side of the window (Figure 3.4, a and Figure 3.7 a). A double click on a node of the tree opens a new tab in the main window. This tab will allow to select either possible analysis steps of the pipeline (Figure 3.4 b) or a visualizations of the results (Figure 3.7 b, c and d). Tools of the pipeline can be started either with the context menu of the metatranscriptome node (Figure 3.4 a) or through the pipeline overview (Figure 3.4 b). For all tools, wizards

Figure 3.4.: Example how to start the pipeline through the user interface. The user can decide either to start the single steps through the pipeline overview in the main window (right) or using the context menu of the metatranscriptome dataset node (left).

(Figure 3.5) guide the user through the needed options to start the program. The wizards ensure that all required variables are set. When starting the same tool for a different metatranscriptome, all feasible variables for program calls are stored and presented to the user as an option. All options for references and tools in Metrans

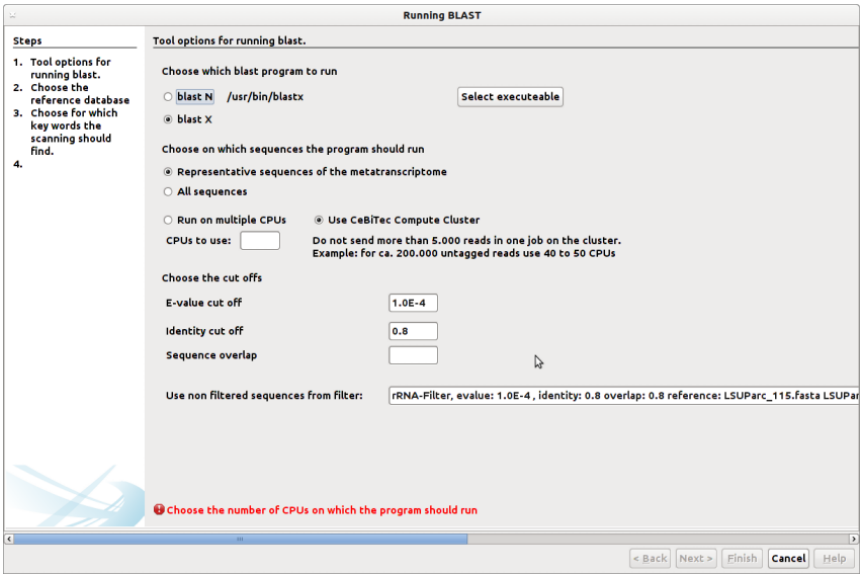

Figure 3.5.: Wizard for the start of the classification of sequences. The wizard guides the user through the single steps of starting the tool.

can be adjusted permanently, using the options menu (see Figure 3.6). Selection of results includes: dataset, classification reference and visualization. The results will be visualized as seen in the right part of Figure 3.7. Figure 3.7 **b** shows the visualization of a taxonomic tree of one dataset with heat maps for each node. The saturation of the color in the heat map indicates the amount of normalized results for one taxon. The visualization window of taxonomic or functional profiles is divided in two parts. Adjustments like normalization or tax level can be made for visualized data (Figure 3.7 **c**). Possible parameters of the visualization can be adjusted at the bottom of the visualization (Figure 3.7 **d**).

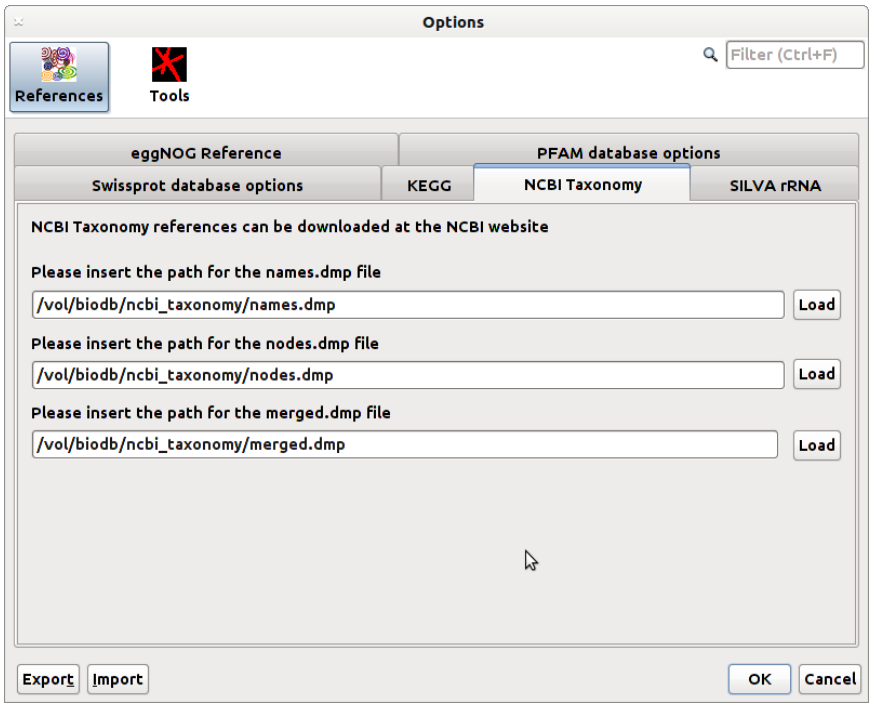

Figure 3.6.: Options window in `Metrans`. Options for all tools and references are adjustable here.

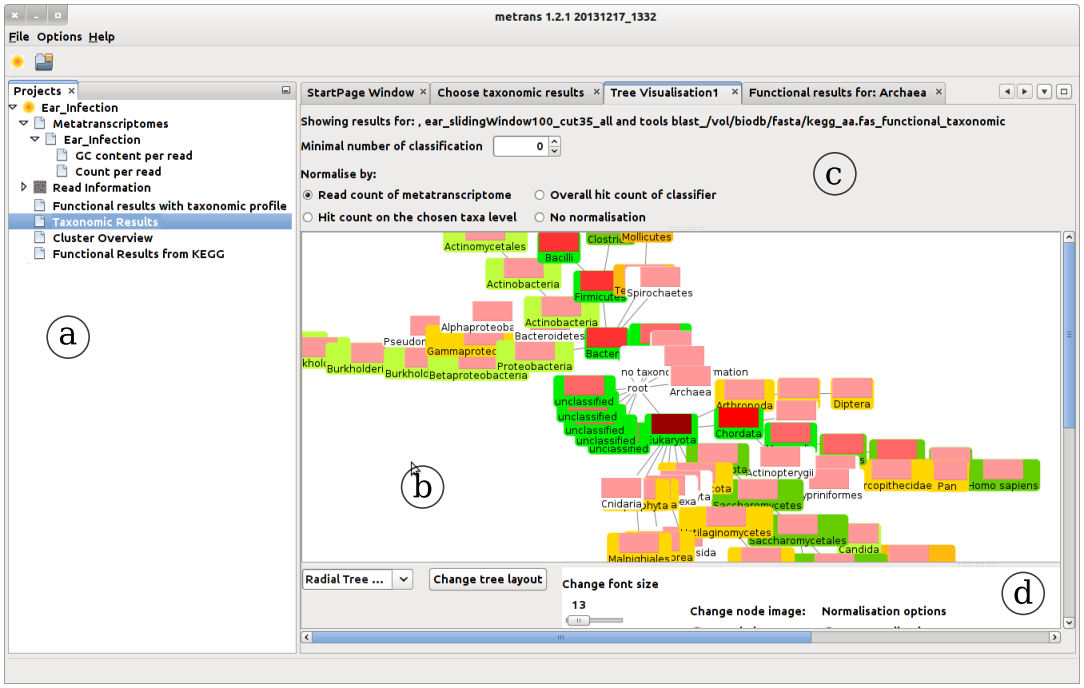

Figure 3.7.: Result visualization in `Metrans`. Part **a** shows the project tab, where all possible results can be seen. In the main part on the right side part **b** show the overall options for a visualization, part **c** shows the visualization of a taxonomic tree and part **d** are special options for the visualization of trees.

Result visualizations Metrans has several visualization options for taxonomic and functional profiles. Taxonomic results can be visualized as trees or as bar charts. Tree visualizations are available as *Linked Tree*, *Radial Tree* and *Balloon Tree*, see Figures 3.8 to 3.10. The Linked Tree is the best visualization for the exploration of the different tax levels through the taxonomic profile. Both the Balloon Tree as well as the Radial Tree are ideal to compare different data sets or classifiers. Single results

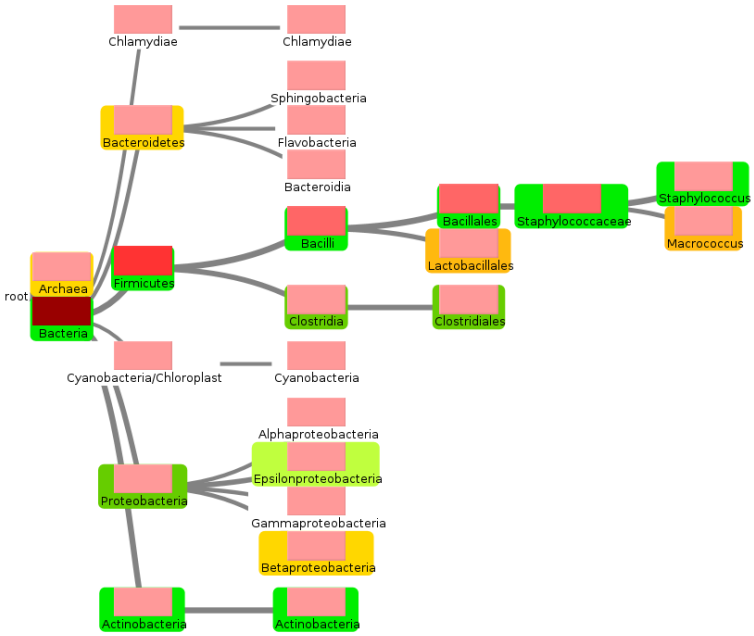

Figure 3.8.: Linked Tree visualization of taxonomic profile. Each node shows a heat map visualization of the amount of results. Width of edge depends on the amount of results.

can be shown as node images either as *heat maps* (Figure 3.9) or as *bar charts* (Figure 3.10). The color schema of the heat maps are depending on the dataset. Depending on the number of results the color for one dataset varies from strong to light, see Figure 3.9. Strong color indicates more results than light one. A click on one of the nodes will open the functional profile corresponding to this node, if there is one. Functional results can be visualized as *Stacked Bar Charts*. The height of the bar indicates the amount of classified reads, normalized at the user's choice, see Figure 3.11. The parts of the bar are the taxonomic classifications at a user chosen taxonomic level. Heights of the parts indicate the amount of taxonomic classifications in the functional result. All visualizations can be exported directly from Metrans in png or jpeg/jpg format for further usage.

Figure 3.9.: Radial Tree visualization of taxonomic profile. The tree shows results of two different classifiers as heat map representation. Red as well as blue color varies from strong to light depending on the number of results.

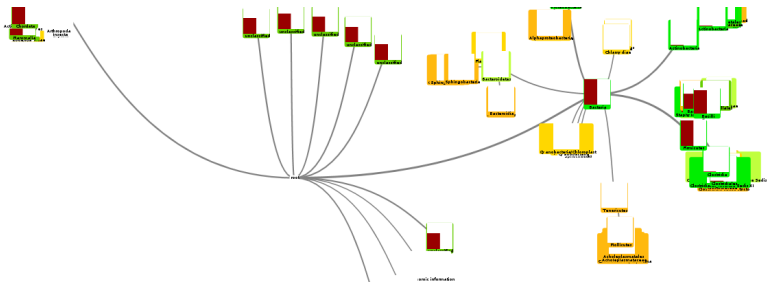

Figure 3.10.: Balloon Tree visualization of taxonomic profile. Amount of classification is shown as bar charts on each taxonomic node.

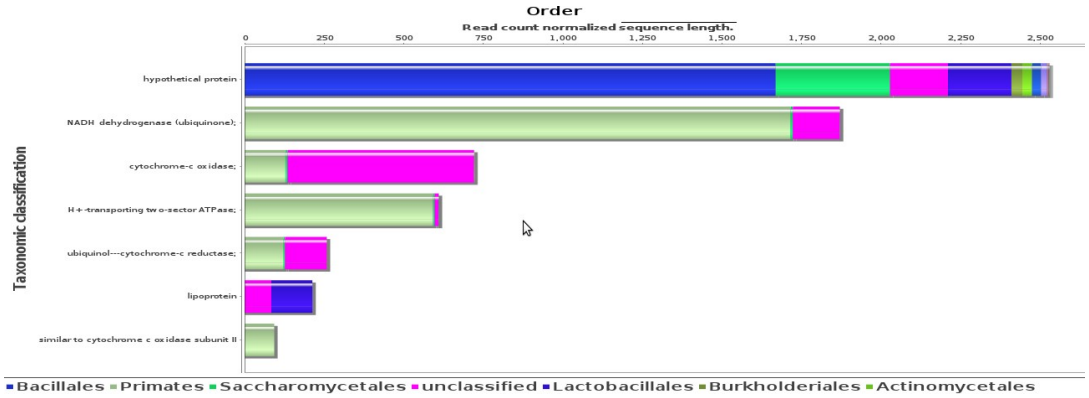

Figure 3.11.: Stacked bar chart of functional results at taxonomic level Genus. Each bar presents one functional prediction. The parts of the bar are the taxonomic predictions in this function.

3.3. Application examples

The Metrans platform was used to analyze a number of projects. Here we present the features of it on two of those projects. The analysis results of a metatranscriptome gained from the ear of a 77-year-old male will be shown in depth. Furthermore, we will show the analysis results of a metatranscriptome time series from tidal flat.

Metatranscriptome gained from a middle ear infection

The sample for the metatranscriptome was gained from a man suffering from middle ear infection (*otitis media*). Total RNA was extracted from the sample, cDNA was synthesized and amplified isothermally. A cDNA library was constructed for sequencing on an Illumina HiSeq 2000. Sequencing yielded 143,923,092 reads of 100 bases in length. For quality control, the window size was set to 100 bp and the average Q-value cut-off to 35. The size of the window was set equal to the read length, since there was a clear drop in quality at base 39 to Q-value 22. After quality control, 64,804,207 reads remained in the analysis pipeline. Reads were binned at a 98% identity threshold, resulting in 5,000,434 bins, reducing the sequences to be analyzed to 7.71% of the reads. A total number of 2.995.125 bins contained only one sequence, indicating either a high amount of sequencing errors or an insufficient sequencing depth. Filtering removed 55.89% of the reads, corresponding to 1,203,818 bins, using the databases RFAM (version 10.1), LSU and SSU (version 111) based on BLASTN with a minimal overlap of 80% and a minimal identity threshold of 80%.

During the functional classification using SwissProt, Metrans indicated a false positive assignment. The functional entry *Uncharacterized protein ORF91* in the uniprot database is likely a wrongly annotated 16S rRNA gene, not included in the current version of the SSU and too dissimilar from the included genes to be filtered out. To remove these sequences, the filter step was repeated using not only BLASTN against the DNA sequences, but also BLASTX against the translated amino acid sequences of the filter databases. This step highly reduced the amount of sequences classified as false positives in the functional analysis. Using the updated filter, 84.29% of all reads were removed. All newly removed sequences were checked by comparing them by BLAST against the NCBI-NT and the NCBI-NR databases [73]. The comparison against the amino acid sequences in the NCBI-NR database resulted mostly in hypothetical or uncharacterized proteins, as well as in one enzyme of the primary metabolism. Comparison of the newly filtered sequences against the NCBI-NT database showed only 16S rRNA as well as mitochondrial sequences. Since the amount of incorrectly classified sequences was highly reduced, it was decided to keep the updated filter step with BLASTX for this dataset. For the taxonomic classification of the sequences the RDP Classifier, as well as the LCA approach with the databases SwissProt, KEGG and COG/EGGNOG were used.

The comparison of the taxonomic profiles from these three databases shows great diversity in the assigned taxa. All possible taxonomic profiles show *Staphylococcus* as one of the most active genera of the microbial community (Figure 3.12). The

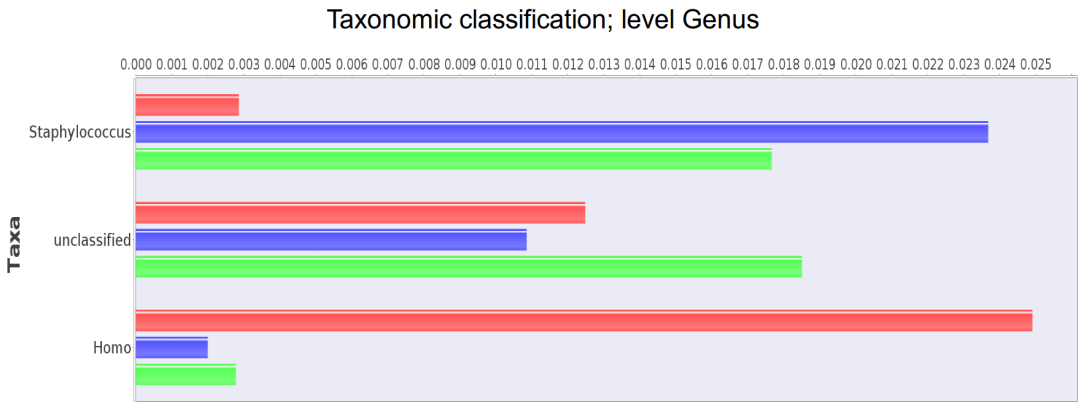

Figure 3.12.: The three most abundant taxa on tax level Genus, classification based on the databases SwissProt (red bar), COG/EGGNOG (blue bar) and KEGG (green bar). The amount of classified reads is normalized by reference length and the total amount of reads classified by the corresponding classifier.

difference in the taxonomic profiles can be attributed to the different amounts of reference sequences from certain organisms in the databases. Different strains of

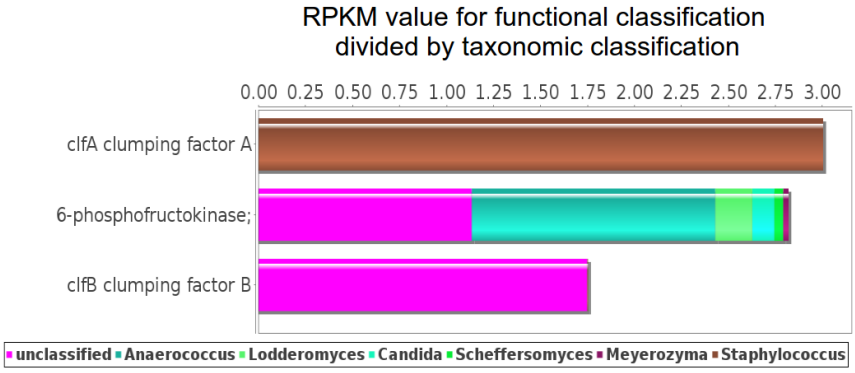

Figure 3.13.: RPKM-value of the virulence factors *clumping factor A*, *clumping factor B* and the gene of the primary metabolism *6-phosphofructokinase*. The different colored parts of the bars indicate the amount of sequences classified to a certain taxon. The classification is based on the SwissProt database.

Staphylococcus are a known source of infections in the human body. A number of

virulence factors like *clumping factors A* and *B* were found at an expression level comparable to that of genes of the primary metabolism like *6-phosphofructokinase* (Figure 3.13). The classification of the virulence factors to the taxon *Staphylococcus* indicates a possible infection by a bacterial strain of this genus. Unfortunately no classification to a specific strain was possible.

Metatranscriptomes time series from tidal flat

In another project `Metrans` was used for the analysis of six metatranscriptomes taken from a tidal flat surface. Samples were taken at two-hour intervals from 06:50 in the morning to 16:50 in the afternoon. Total RNA was isolated, mRNA enriched and translated into cDNA. The cDNA was sequenced on an Illumina MiSeq sequencer as *paired end sequences*. To gain paired end sequences, the DNA is sequenced from both sides. If the DNA strand is smaller than the sequencing length, the sequences overlap and can be used to gain longer reads. Table 3.1 shows an overview of the sequences gained from the tidal flat. In this example the removal of rRNA was only done with `BLASTN`. An overview of all binning and rRNA removal results can be seen in Table 3.2. The analysis with `Metrans` showed a shift in the amount of sequences classified to enzymes of the photosynthesis cycle during the

Sample time	Tide level	Number of reads
06:50	low tide	404,183
08:50	late low tide	573,364
10:50	rising tide	648,488
12:50	high tide	562,189
14:50	falling tide	529,437
16:50	early low tide	534,292

Table 3.1.: Sample data of tidal flat metatranscriptome probes

Sample	Number of bins	Single read bins	non coding RNA
06:50 low tide	318,485	73.92%	25.05%
08:50 late low tide	167,898	22.85%	82.26%
10:50 rising tide	164,353	19.30%	84.03%
12:50 high tide	123,425	16.50%	88.14%
14:50 falling tide	272,793	45.60%	55.79%
16:50 early low tide	338,498	57.42%	40.47%

Table 3.2.: Results of binning at 99% sequence similarity and filtering of non coding RNA

day. Only a few of those sequences could be classified at least at phylum level. There were mostly *Cyanobacteria* present (see Figure 3.14).

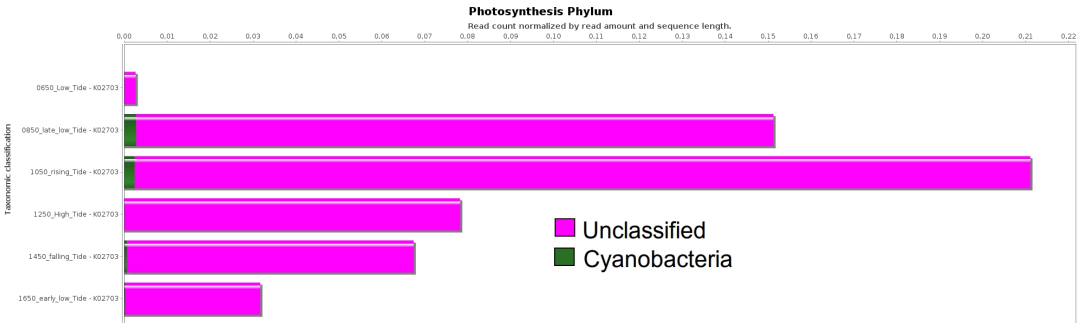

Figure 3.14.: Taxonomic classification at level phylum of the most abundant Ko-Number (K02703) that mapped to the Photosynthesis cycle based on the KEGG database. This Ko-Number has the definition *photosystem II P680 reaction center D1 protein*.

3.4. Conclusion

`Metrans` is a freely available software platform for the analysis of metatranscriptome datasets. It is easy to use and wizards guide the user through all steps of the pipeline. `Metrans` can analyze large datasets to generate both taxonomic as well as functional profiles. Especially the binning step makes it possible to analyze big data sets, while at the same time masks sequencing errors. The combined visualization of the taxonomical and functional profiles provide the opportunity to find out if certain metabolic functions are shared between taxa. It can also show the functional metabolism of certain taxa. Those profiles can be used to gain an overview of the active organisms, as well as of the expressed metabolic functions in the microbial environment. To perform further analysis, all sequences corresponding to any functional or taxonomic result can be downloaded from `Metrans`. All results are normalized according to gene length as well as with a user chosen normalization feature, such as the overall number of results or the overall number of reads in the metatranscriptome. Since the amount of non-coding RNA in the sample influences the amount of reads that are classified in a functional category, the normalization by the number of reads is not adequate. Therefore, the normalization by number of results is included.

Since there are some steps in the pipeline that are problematic for certain kind of data sets, it is possible to skip certain steps of the pipeline. For example, binning data sets with shallow sequencing depth may shorten the representative reads

unduly and does not remove sequencing errors, since the size of the bins is then quite small. Furthermore, certain mRNA enrichment methods deplete the rRNA unevenly, making a comparison of the taxonomic profile from the `RDP Classifier` and the functional classification nonsensical.

The modular structure of `Metrans` makes the skipping of pipeline steps and the integration of new ones easier. For the integration of new modules existing interfaces can be implemented. Some of those modules could be statistical modules to make comparison of different datasets easier. Furthermore a module for direct comparison of the metatranscriptome and metagenome sequences would provide the user with the possibility not only to compare analysis results but also to compare directly the sequenced reads.

Chapter 4

Conclusion and outlook

Microbial communities play an important role in the life cycle of planet earth. Research on microbial communities concentrates on understanding the species abundance in those communities, the potential and expressed metabolic functions as well as the interaction between the microbes within the community or their host. Since the development of next generation sequencing methods research on those communities has accelerated. The NGS methods simultaneously provide the opportunity to find out more about microbial communities and pose a challenge for analysis and storage. The work of this thesis contributes to the field metagenomics and metatranscriptomics with the development of new bioinformatic methods.

For fast taxonomic analysis of whole shotgun metagenome sequence data the software `metaBEETL` was developed. It relies on Burrows-Wheeler transformed sequences, so that compressed sequence data can be classified without decompression. `metaBEETL` is based on the software `BEETL` that uses the all-against-all backward search to analyze sequence data while the compressed data is held on disk. With new bias control methods `metaBEETL` generates reliable taxonomic profiles for whole shotgun metagenome reads. Since `metaBEETL` is based on exact k-mer counts it only provides reliable taxonomic profiles for already well researched communities. Even though with each year more microbial communities are researched, further development of the software should concentrate on this challenge. One possibility would be the translation of the sequences in amino acids before classification since amino acid sequences are often more preserved than nucleotide acids.

For the analysis and comparison of metatranscriptome sequence data the Rich Client software platform `Metrans` was developed. `Metrans`combines different analysis tools to a pipeline to gain combined taxonomic and functional profiles for metatranscriptome sequence data. The pipeline includes binning of the reads, filtering of non-coding RNA, taxonomic classification of the 16S rRNA sequences and tax-

onomic and functional classification through the comparison of the sequences to databases containing already annotated sequences. Including wizards for easy employment of the pipeline tools and the different visualizations of the analysis results, `Metrans` offers an user friendly way to analyze metatranscriptome data. To projects analyzed with `Metrans` were shown. For the metatranscriptome from the infected ear of a 77-year-old male confirmed the diagnosis of a *Staphiilococcus* infection. Even thoug the metatranscriptome time series from tidal flat surface was done with a low amunt of sequences, the analysis showed increasing expression in the photosyntehsis pathway during the day. For further development more analysis tools and visualizations could be included More importantly, would be statistical analysis tools to compare different metatranscriptome datasets.

Further development of bioinformatic tools for metatranscriptome and metagenome datasets would be the comparison of those datasets. To compare metagenome and metatranscriptome datasets two methods are currently used. The first method is the assembly of the metagenome reads according to sequence similarity, gaining longer sequences (*contigs*). Afterwards, genes are predicted on the contigs. The metatranscriptome reads are mapped on the contigs and the further analysis is similar to established RNA-Seq analysis methods. While this method is fast, it has several bias sources. First of all, if the microbial community is highly heterogeneous, reads from different species are assembled in one heterogeneous contig. In those contigs sequence variations that are only present in a small amount of the species are hidden. If sequence variations are high enough, transcripts of species occurring in a small amount will not be mapped. Secondly, it can happen that reads from low occurring species are not assembled in a contig, therefore the transcripts can not be mapped, losing transcript and genome information. An other method is to compare the results of a functional and taxonomical analysis of both datasets. This takes a long time and does not give information for sequences that were not observed before and therefore have no classification. However, a direct comparison of metagenome and metatranscriptome reads would give an insight about difference in the amount of expression and the existence of genes without the biases of the currently used methods.

To directly compare metatranscriptome and metagenome reads, the method used in `metaBEETL` can be modified as follows: The list of k-mers shared between the metagenome and the metatranscriptome reads can be utilized to find levels of occurrences of sequences in the datasets without classification. Normalized with the read counts of the respective datasets, those levels can be used to analyze a number of metatranscriptomes from microbial communities in different environmental conditions compared to their respective metagenomes. Therefore, it would be possible to find differences in expression levels without the biases introduced by classifica-

tion of the reads or the assembly. Integrating this method in the `Metrans` pipeline offers the opportunity for further analysis steps and the comparison of metatranscriptomes with their respective metagenomes.

Acknowledgements

First, I would like to thank Prof. Dr. Jens Stoye and Prof. Dr. Andreas Tauch for their support and advice during my PhD project. I am also very grateful for the advice and good collaboration of Eugenie Fredrich, in whom I found not only a good colleague but also a friend during my PhD project.
I would like also to thank my colleagues and friends for the good advice and important discussions, especially Pina Krell and Dr. Annelyse Thévenin for proof reading my thesis. I want to acknowledge the "CLIB Graduate Custer" for funding during my PhD project. I also wish to thank Dr. Ole Schulz-Trieglaff and Dr. Anthony Cox for giving me the wonderful opportunity to work at Illumina in Little Chesterford and for making my stay in England a really pleasant one.
Words can not express enough how grateful I am for my partner Dr. Carsten Gnörlich, who helped, encouraged and supported me during my whole PhD project. I also want to thank my dear friend Imani for standing by me in the hard and the good times. I am also grateful for my parents, who always had an open ear or advice for me. At last I want to thank all my friends who supported me during this period that are not mentioned here.

List of Figures

List of Tables

Bibliography

[1] M. Z. Alam, A. Haque, Q. Alam, M. A. Kamal, and A. M. Abuzenadah. A Possible Link of Gut Microbiota Alteration in Type 2 Diabetes and Alzheimer's Disease Pathogenicity: An Update. *CNS Neurol Disord Drug Targets*, 2013.

[2] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403–410, 1990.

[3] C. Ander, O. B. Schulz-Trieglaff, J. Stoye, and A. J. Cox. metaBEETL: high-throughput analysis of heterogeneous microbial populations from shotgun DNA sequences. *BMC Bioinformatics*, 14 Suppl 5:S2, 2013.

[4] M. J. Bauer, A. J. Cox, and G. Rosone. Lightweight BWT construction for very large string collections. In *CPM 2011*, vol. 6661 of *LNCS*, 219–231. Springer, 2011.

[5] A. L. Bazinet and M. P. Cummings. A comparative evaluation of sequence classification programs. *BMC Bioinformatics*, 13:92, 2012.

[6] R. D. Berg. The indigenous gastrointestinal microflora. *Trends Microbiol.*, 4(11):430–435, 1996.

[7] M. J. Blaser and S. Falkow. What are the consequences of the disappearing human microbiota? *Nat. Rev. Microbiol.*, 7(12):887–894, 2009.

[8] M. J. Blaser and D. Kirschner. The equilibria that allow bacterial persistence in human hosts. *Nature*, 449(7164):843–849, 2007.

[9] B. Boeckmann, M. C. Blatter, L. Famiglietti, U. Hinz, L. Lane, B. Roechert, and A. Bairoch. Protein variety and functional diversity: Swiss-Prot annotation in its biological context. *C. R. Biol.*, 328(10-11):882–899, 2005.

[10] A. Brady and S. Salzberg. PhymmBL expanded: confidence scores, custom databases, parallelization and more. *Nat. Methods*, 8(5):367, 2011.

[11] S. E. Bulow, C. A. Francis, G. A. Jackson, and B. B. Ward. Sediment denitrifier community composition and nirS gene expression investigated with functional gene microarrays. *Environ. Microbiol.*, 10(11):3057–3069, 2008.

[12] M. Burrows and D. J. Wheeler. A block sorting data compression algorithm. Tech. rep., DIGITAL System Research Center, 1994.

[13] G. Campbell-Platt. Fermented foods â a world perspective. *Food Research International*, 27(3):253 – 257, 1994.

[14] D. E. Canfield, F. J. Stewart, B. Thamdrup, L. De Brabandere, T. Dalsgaard, E. F. Delong, N. P. Revsbech, and O. Ulloa. A cryptic sulfur cycle in oxygen-minimum-zone waters off the Chilean coast. *Science*, 330(6009):1375–1378, 2010.

[15] D. G. Capone. Marine nitrogen fixation: what's the fuss? *Curr. Opin. Microbiol.*, 4(3):341–348, 2001.

[16] A. Copeland, A. Lapidus, T. Glavina Del Rio, M. Nolan, S. Lucas, *et al.* Complete genome sequence of Catenulispora acidiphila type strain (ID 139908). *Stand Genomic Sci*, 1(2):119–125, 2009.

[17] C. C. Crowe, W. E. Sanders, and S. Longley. Bacterial interference. II. Role of the normal throat flora in prevention of colonization by group A Streptococcus. *J. Infect. Dis.*, 128(4):527–532, 1973.

[18] C. F. Davenport, J. Neugebauer, N. Beckmann, B. Friedrich, B. Kameri, *et al.* Genometa–a fast and accurate classifier for short metagenomic shotgun reads. *PLoS ONE*, 7(8):e41 224, 2012.

[19] E. F. DeLong and D. M. Karl. Genomic perspectives in microbial oceanography. *Nature*, 437(7057):336–342, 2005.

[20] M. P. Deutscher. Degradation of RNA in bacteria: comparison of mRNA and stable RNA. *Nucleic Acids Res.*, 34(2):659–666, 2006.

[21] R. A. Edwards, B. Rodriguez-Brito, L. Wegley, M. Haynes, M. Breitbart, D. M. Peterson, M. O. Saar, S. Alexander, E. C. Alexander, and F. Rohwer. Using pyrosequencing to shed light on deep mine microbial ecology. *BMC Genomics*, 7:57, 2006.

[22] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, FOCS '00, 390–. IEEE Computer Society, Washington, DC, USA, 2000.

[23] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, 1987.

[24] W. Gerlach. *Taxonomic classification of metagenomic sequences*. Ph.D. thesis, Bielefeld University, 2012.

[25] W. Gerlach and J. Stoye. Taxonomic classification of metagenomic shotgun sequences with CARMA3. *Nucleic Acids Res.*, 39(14):e91, 2011.

[26] H. Gest, R. Hooke, and A. V. Leeuwenhoek. The discovery of microorganisms by Robert Hooke and Antoni Van Leeuwenhoek, fellows of the Royal Society. *Notes Rec R Soc Lond*, 58(2):187–201, 2004.

[27] J. A. Gilbert, D. Field, Y. Huang, R. Edwards, W. Li, P. Gilna, and I. Joint. Detection of large numbers of novel sequences in the metatranscriptomes of complex marine microbial communities. *PloS one*, 3(8):e3042, 2008.

[28] F. Gori, G. Folino, M. S. Jetten, and E. Marchiori. MTR: taxonomic annotation of short metagenomic reads using clustering at multiple taxonomic ranks. *Bioinformatics*, 27(2):196–203, 2011.

[29] R. I. Griffiths, A. S. Whiteley, A. G. O'Donnell, and M. J. Bailey. Rapid method for coextraction of DNA and RNA from natural environments for analysis of ribosomal DNA- and rRNA-based microbial community composition. *Appl. Environ. Microbiol.*, 66(12):5488–5491, 2000.

[30] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S. R. Eddy. Rfam: an RNA family database. *Nucleic Acids Res.*, 31(1):439–441, 2003.

[31] J. Handelsman, M. R. Rondon, S. F. Brady, J. Clardy, and R. M. Goodman. Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products. *Chem. Biol.*, 5(10):R245–249, 1998.

[32] T. Hasunuma, F. Okazaki, N. Okai, K. Y. Hara, J. Ishii, and A. Kondo. A review of enzymes and microbes for lignocellulosic biorefinery and the possibility of their application to consolidated bioprocessing technology. *Bioresour. Technol.*, 135:513–522, 2013.

[33] S. He, O. Wurtzel, K. Singh, J. L. Froula, S. Yilmaz, *et al.* Validation of two ribosomal RNA removal methods for microbial metatranscriptomics. *Nat. Methods*, 7(10):807–812, 2010.

[34] I. Hewson, R. S. Poretsky, S. T. Dyhrman, B. Zielinski, A. E. White, H. J. Tripp, J. P. Montoya, and J. P. Zehr. Microbial community gene expression within colonies of the diazotroph, Trichodesmium, from the Southwest Pacific Ocean. *ISME J*, 3(11):1286–1300, 2009.

[35] M. Horton, N. Bodenhausen, and J. Bergelson. MARTA: a suite of Java-based tools for assigning taxonomic status to DNA sequences. *Bioinformatics*, 26(4):568–569, 2010.

[36] K. Hosoki, A. Nakamura, K. Kainuma, M. Sugimoto, M. Nagao, *et al.* Differential activation of eosinophils by bacteria associated with asthma. *Int. Arch. Allergy Immunol.*, 161 Suppl 2:16–22, 2013.

[37] J. Hu and J. L. Blanchard. Environmental sequence data from the Sargasso Sea reveal that the characteristics of genome reduction in Prochlorococcus are not a harbinger for an escalation in genetic drift. *Mol. Biol. Evol.*, 26(1):5–13, 2009.

[38] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the Institute of Radio Engineers*, 40(9):1098–1101, 1952.

[39] D. H. Huson, A. F. Auch, J. Qi, and S. C. Schuster. MEGAN analysis of metagenomic data. *Genome Res.*, 17(3):377–386, 2007.

[40] D. H. Huson, S. Mitra, H. J. Ruscheweyh, N. Weber, and S. C. Schuster. Integrative analysis of environmental sequences using MEGAN4. *Genome Res.*, 21(9):1552–1560, 2011.

[41] W. P. Inskeep, D. B. Rusch, Z. J. Jay, M. J. Herrgard, M. A. Kozubal, *et al.* Metagenomes from high-temperature chemotrophic systems reveal geochemical controls on microbial community structure and function. *PLoS ONE*, 5(3):e9773, 2010.

[42] D. S. Jones, H. L. Albrecht, K. S. Dawson, I. Schaperdoth, K. H. Freeman, Y. Pi, A. Pearson, and J. L. Macalady. Community genomic analysis of an extremely acidophilic sulfur-oxidizing biofilm. *ISME J*, 6(1):158–170, 2012.

[43] M. Jones, A. Ghoorah, and M. Blaxter. jMOTU and Taxonerator: turning DNA Barcode sequences into annotated operational taxonomic units. *PLoS ONE*, 6(4):e19 259, 2011.

[44] S. D. Kahn. On the future of genomic data. *Science*, 331(6018):728–729, 2011.

[45] M. Kanehisa. A database for post-genome analysis. *Trends Genet.*, 13(9):375–376, 1997.

[46] J. M. Keller, M. R. Gray, and Jr. A fuzzy k-nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, 15:580–585, 1985.

[47] D. R. Kelley and S. L. Salzberg. Clustering metagenomic sequences with interpolated Markov models. *BMC Bioinformatics*, 11:544, 2010.

[48] J. L. Kirk, L. A. Beaudette, M. Hart, P. Moutoglis, J. N. Klironomos, H. Lee, and J. T. Trevors. Methods of studying soil microbial diversity. *J. Microbiol. Methods*, 58(2):169–188, 2004.

[49] J. A. Klappenbach, P. R. Saxman, J. R. Cole, and T. M. Schmidt. rrndb: the Ribosomal RNA Operon Copy Number Database. *Nucleic Acids Res.*, 29(1):181–184, 2001.

[50] B. Langmead, C. Trapnell, M. Pop, and S. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25+, 2009.

[51] J. L. Legras, D. Merdinoglu, J. M. Cornuet, and F. Karst. Bread, beer and wine: Saccharomyces cerevisiae diversity reflects human history. *Mol. Ecol.*, 16(10):2091–2102, 2007.

[52] S. Leininger, T. Urich, M. Schloter, L. Schwark, J. Qi, G. W. Nicol, J. I. Prosser, S. C. Schuster, and C. Schleper. Archaea predominate among ammonia-oxidizing prokaryotes in soils. *Nature*, 442(7104):806–809, 2006.

[53] R. E. Ley, C. A. Lozupone, M. Hamady, R. Knight, and J. I. Gordon. Worlds within worlds: evolution of the vertebrate gut microbiota. *Nat. Rev. Microbiol.*, 6(10):776–788, 2008.

[54] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[55] Y. W. Lim, R. Schmieder, M. Haynes, D. Willner, M. Furlan, *et al.* Metagenomics and metatranscriptomics: Windows on CF-associated viral and microbial communities. *J. Cyst. Fibros.*, 2012.

[56] B. Liu, T. Gibbons, M. Ghodsi, T. Treangen, and M. Pop. Accurate and fast estimation of taxonomic profiles from metagenomic shotgun sequences. *BMC Genomics*, 12 Suppl 2:S4, 2011.

[57] L. Liu, Y. Li, S. Li, N. Hu, Y. He, R. Pong, D. Lin, L. Lu, and M. Law. Comparison of next-generation sequencing systems. *J. Biomed. Biotechnol.*, 2012:251 364, 2012.

[58] D. R. Lovley. Cleaning up with genomics: applying molecular biology to bioremediation. *Nat. Rev. Microbiol.*, 1(1):35–44, 2003.

[59] A. M. Maxam and W. Gilbert. A new method for sequencing DNA. *Proc. Natl. Acad. Sci. U.S.A.*, 74(2):560–564, 1977.

[60] J. P. McCutcheon and N. A. Moran. Functional convergence in reduced genomes of bacterial symbionts spanning 200 My of evolution. *Genome Biol Evol*, 2:708–718, 2010.

[61] A. C. McHardy, H. G. Martin, A. Tsirigos, P. Hugenholtz, and I. Rigoutsos. Accurate phylogenetic classification of variable-length DNA fragments. *Nat. Methods*, 4(1):63–72, 2007.

[62] F. Meyer, D. Paarmann, M. D'Souza, R. Olson, E. M. Glass, *et al.* The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics*, 9:386, 2008.

[63] S. Mitra, B. Klar, and D. H. Huson. Visual and statistical comparison of metagenomes. *Bioinformatics*, 25(15):1849–1855, 2009.

[64] M. Monzoorul Haque, T. S. Ghosh, D. Komanduri, and S. S. Mande. SOrt-ITEMS: Sequence orthology based approach for improved taxonomic estimation of metagenomic sequences. *Bioinformatics*, 25(14):1722–1730, 2009.

[65] E. F. Murphy, P. D. Cotter, S. Healy, T. M. Marques, O. O'Sullivan, *et al.* Composition and energy harvesting capacity of the gut microbiota: relationship to diet, obesity and time in mouse models. *Gut*, 59(12):1635–1642, 2010.

[66] O. U. Nalbantoglu, S. F. Way, S. H. Hinrichs, and K. Sayood. RAIphy: phylogenetic classification of metagenomics samples using iterative refinement of relative abundance index profiles. *BMC Bioinformatics*, 12:41, 2011.

[67] S. Nurk, A. Bankevich, D. Antipov, A. A. Gurevich, A. Korobeynikov, *et al.* Assembling single-cell genomes and mini-metagenomes from chimeric MDA products. *J. Comput. Biol.*, 20(10):714–737, 2013.

[68] G. J. Olsen, N. Larsen, and C. R. Woese. The ribosomal RNA database project. *Nucleic Acids Res.*, 19 Suppl:2017–2021, 1991.

[69] G. Piganeau, Y. Desdevises, E. Derelle, and H. Moreau. Picoeukaryotic sequences in the Sargasso sea metagenome. *Genome Biol.*, 9(1):R5, 2008.

[70] G. Piganeau and H. Moreau. Screening the Sargasso Sea metagenome for data to investigate genome evolution in Ostreococcus (Prasinophyceae, Chlorophyta). *Gene*, 406(1-2):184–190, 2007.

[71] V. Poroyko, J. R. White, M. Wang, S. Donovan, J. Alverdy, D. C. Liu, and M. J. Morowitz. Gut microbial gene expression in mother-fed and formula-fed piglets. *PLoS ONE*, 5(8):e12 459, 2010.

[72] S. Powell, D. Szklarczyk, K. Trachana, A. Roth, M. Kuhn, *et al.* eggNOG v3.0: orthologous groups covering 1133 organisms at 41 different taxonomic ranges. *Nucleic Acids Res.*, 40(Database issue):D284–289, 2012.

[73] K. D. Pruitt, T. Tatusova, and D. R. Maglott. NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res.*, 33(Database issue):D501–504, 2005.

[74] M. Punta, P. C. Coggill, R. Y. Eberhardt, J. Mistry, J. Tate, *et al.* The Pfam protein families database. *Nucleic Acids Res.*, 40(Database issue):290–301, 2012.

[75] D. C. Richter, F. Ott, A. F. Auch, R. Schmid, and D. H. Huson. MetaSim: a sequencing simulator for genomics and metagenomics. *PLoS ONE*, 3(10):e3373, 2008.

[76] M. R. Rondon, P. R. August, A. D. Bettermann, S. F. Brady, T. H. Grossman, *et al.* Cloning the soil metagenome: a strategy for accessing the genetic and functional diversity of uncultured microorganisms. *Appl. Environ. Microbiol.*, 66(6):2541–2547, 2000.

[77] G. L. Rosen, E. R. Reichenberger, and A. M. Rosenfeld. NBC: the Naive Bayes Classification tool webserver for taxonomic classification of metagenomic reads. *Bioinformatics*, 27(1):127–129, 2011.

[78] R. K. Saiki, S. Scharf, F. Faloona, K. B. Mullis, G. T. Horn, H. A. Erlich, and N. Arnheim. Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia. *Science*, 230(4732):1350–1354, 1985.

[79] S. L. Salzberg, A. L. Delcher, S. Kasif, and O. White. Microbial gene identification using interpolated Markov models. *Nucleic Acids Res.*, 26(2):544–548, 1998.

[80] J. G. Sanders, R. A. Beinart, F. J. Stewart, E. F. Delong, and P. R. Girguis. Metatranscriptomics reveal differences in in situ energy and nitrogen metabolism among hydrothermal vent snail symbionts. *ISME J*, 7(8):1556–1567, 2013.

[81] F. Sanger and A. R. Coulson. A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *J. Mol. Biol.*, 94(3):441–448, 1975.

[82] E. W. Sayers, T. Barrett, D. A. Benson, S. H. Bryant, K. Canese, *et al.* Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, 37(Database issue):5–15, 2009.

[83] M. Schena, R. A. Heller, T. P. Theriault, K. Konrad, E. Lachenmeier, and R. W. Davis. Microarrays: biotechnology's discovery platform for functional genomics. *Trends Biotechnol.*, 16(7):301–306, 1998.

[84] A. M. Schnoes, S. D. Brown, I. Dodevski, and P. C. Babbitt. Annotation error in public databases: Misannotation of molecular function in enzyme superfamilies. *PLoS Comput Biol*, 5(12):e1000 605, 2009.

[85] K. Smith, K. D. McCoy, and A. J. Macpherson. Use of axenic animals in studying the adaptation of mammals to their commensal intestinal microbiota. *Semin. Immunol.*, 19(2):59–69, 2007.

[86] R. Sorek and P. Cossart. Prokaryotic transcriptomics: a new view on regulation, physiology and pathogenicity. *Nat. Rev. Genet.*, 11(1):9–16, 2010.

[87] C. Spits, C. Le Caignec, M. De Rycke, L. Van Haute, A. Van Steirteghem, I. Liebaers, and K. Sermon. Whole-genome multiple displacement amplification from single cells. *Nat Protoc*, 1(4):1965–1970, 2006.

[88] H. Stranneheim, M. Kaller, T. Allander, B. Andersson, L. Arvestad, and J. Lundeberg. Classification of DNA sequences using Bloom filters. *Bioinformatics*, 26(13):1595–1600, 2010.

[89] J. Trevors. Bacterial biodiversity in soil with an emphasis on chemically-contaminated soils. *Water, Air, and Soil Pollution*, 101(1-4):45–67, 1998.

[90] P. J. Turnbaugh, R. E. Ley, M. Hamady, C. M. Fraser-Liggett, R. Knight, and J. I. Gordon. The human microbiome project. *Nature*, 449(7164):804–810, 2007.

[91] P. J. Turnbaugh, R. E. Ley, M. Hamady, C. M. Fraser-Liggett, R. Knight, and J. I. Gordon. The human microbiome project. *Nature*, 449(7164):804–810, 2007.

[92] J. C. Venter, K. Remington, J. F. Heidelberg, A. L. Halpern, D. Rusch, *et al.* Environmental genome shotgun sequencing of the Sargasso Sea. *Science*, 304(5667):66–74, 2004.

[93] K. V. Voelkerding, S. A. Dames, and J. D. Durtschi. Next-generation sequencing: from basic research to diagnostics. *Clin. Chem.*, 55(4):641–658, 2009.

[94] D. A. Walsh, E. Zaikova, C. G. Howes, Y. C. Song, J. J. Wright, S. G. Tringe, P. D. Tortell, and S. J. Hallam. Metagenome of a versatile chemolithoautotroph from expanding oceanic dead zones. *Science*, 326(5952):578–582, 2009.

[95] D. Z. Wang, Z. X. Xie, and S. F. Zhang. Marine metaproteomics: Current status and future directions. *J Proteomics*, 2013.

[96] Q. Wang, G. M. Garrity, J. M. Tiedje, and J. R. Cole. Naïve Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Appl. Environ. Microbiol.*, 73(16):5261–5267, 2007.

[97] J. D. Watson and F. H. Crick. Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, 1953.

[98] S. Weckx, J. Allemeersch, R. Van der Meulen, G. Vrancken, G. Huys, P. Vandamme, P. Van Hummelen, and L. De Vuyst. Development and validation of a species-independent functional gene microarray that targets lactic acid bacteria. *Appl. Environ. Microbiol.*, 75(20):6488–6495, 2009.

[99] W. G. Weisburg, S. M. Barns, D. A. Pelletier, and D. J. Lane. 16S ribosomal DNA amplification for phylogenetic study. *J. Bacteriol.*, 173(2):697–703, 1991.

[100] W. B. Whitman, D. C. Coleman, and W. J. Wiebe. Prokaryotes: the unseen majority. *Proc. Natl. Acad. Sci. U.S.A.*, 95(12):6578–6583, 1998.

[101] X. Yang. Error correction and clustering algorithms for next generation sequencing. In *Proceedings of the 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum*, IPDPSW '11, 2101–2104. IEEE Computer Society, Washington, DC, USA, 2011.

[102] Y. You, C. Fu, X. Zeng, D. Fang, X. Yan, B. Sun, D. Xiao, and J. Zhang. A novel DNA microarray for rapid diagnosis of enteropathogenic bacteria in stool specimens of patients with diarrhea. *J. Microbiol. Methods*, 75(3):566–571, 2008.

[103] H. Zhang. The optimality of naive bayes. In V. Barr and Z. Markov, eds., *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*. AAAI Press, 2004.

Appendix A

metaBEETL

A.1. All-against-all backward search

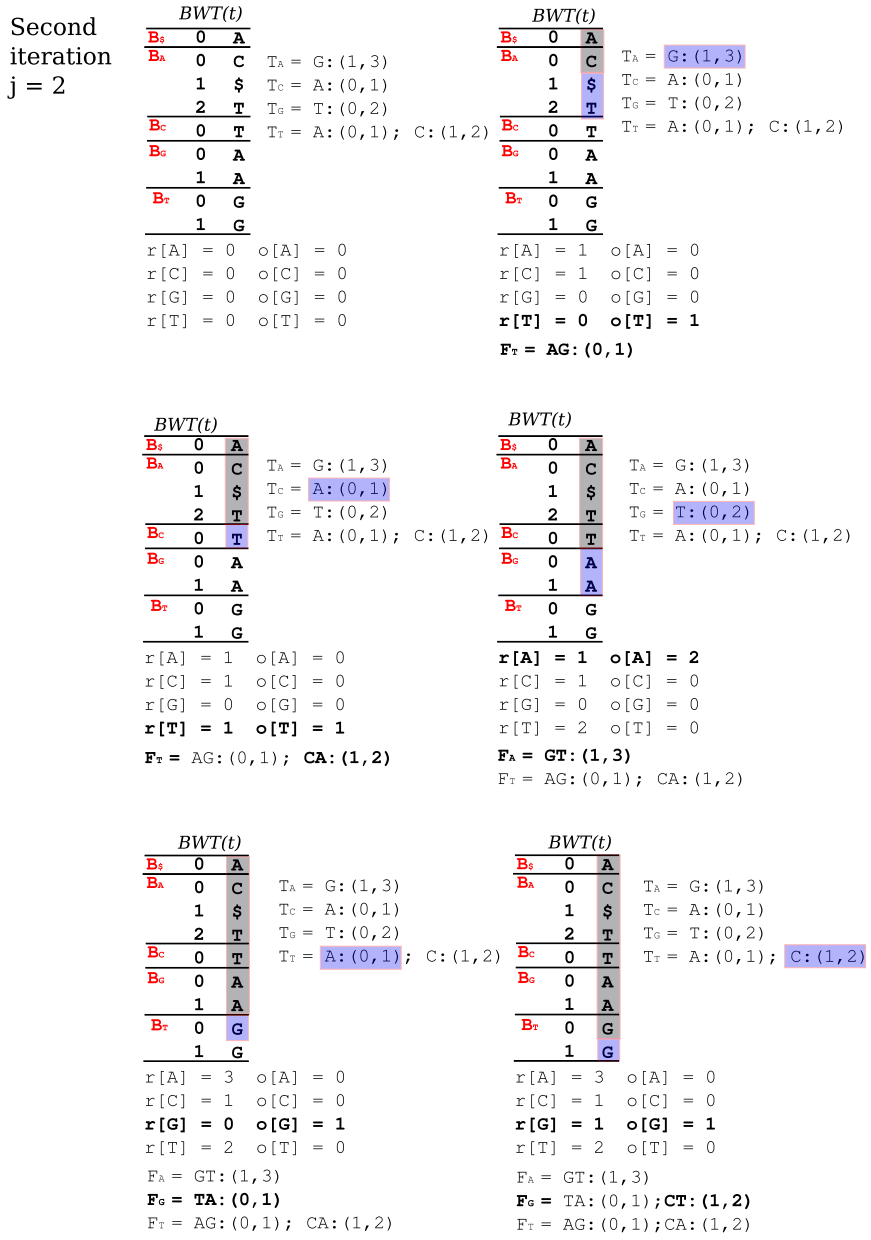

Figure A.1.: Second iteration of the all against all backward search. Grey boxes indicate the read part of the $BWT(t)$. Array $r[c]$ contains the number of characters read in the BWT before the current section. Array $o[c]$ contains the number of character in one Q-interval.

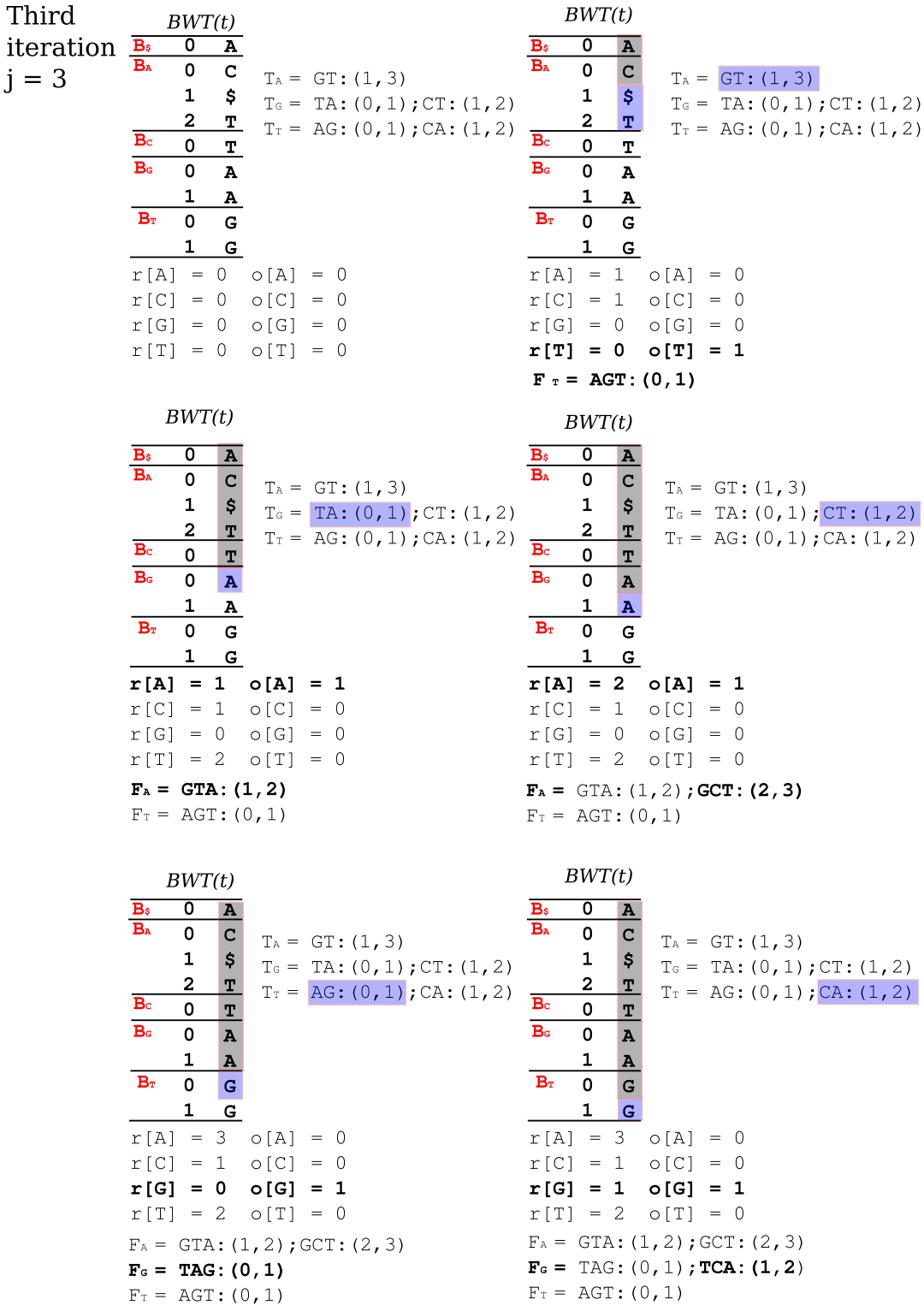

Figure A.2.: Third iteration of the all against all backward search. Grey boxes indicate the read part of the $BWT(t)$. Array $r[c]$ contains the number of characters read in the BWT before the current section. Array $o[c]$ contains the number of character in one Q-interval.

A.2. Taxonomic profiles of simulated data with classifiers metaBEETL, CARMA3, MEGAN and Genometa

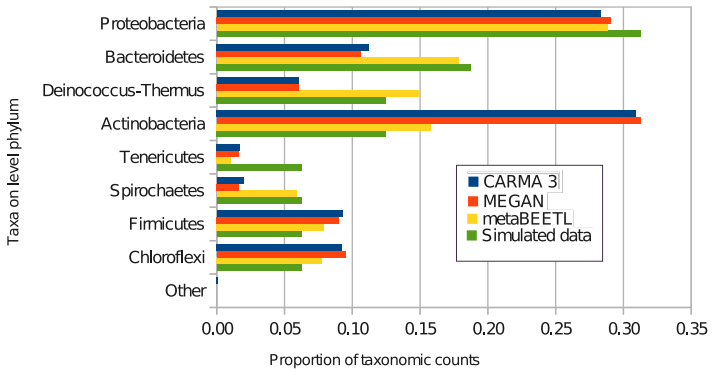

Figure A.3.: Phylum-level composition of simulated data compared with classifications produced by metaBEETL, CARMA3 and MEGAN.

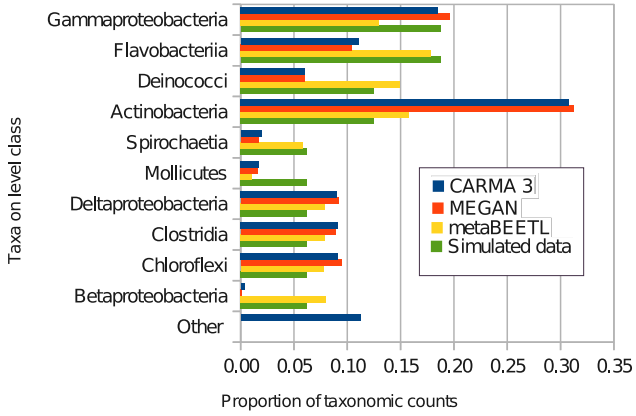

Figure A.4.: Class-level composition of simulated data compared with classifications produced by metaBEETL, CARMA3 and MEGAN.

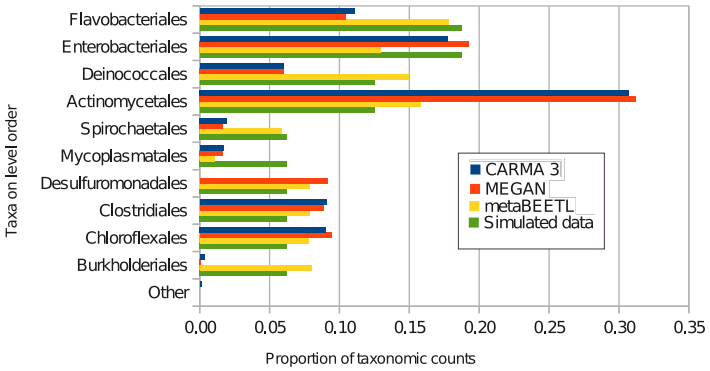

Figure A.5.: Order-level composition of simulated data compared with classifications produced by metaBEETL, CARMA3 and MEGAN.

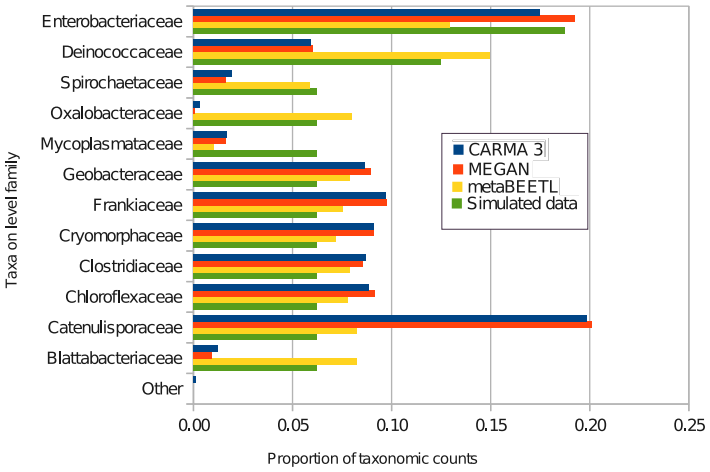

Figure A.6.: Family-level composition of simulated data compared with classifications produced by metaBEETL, CARMA3 and MEGAN.

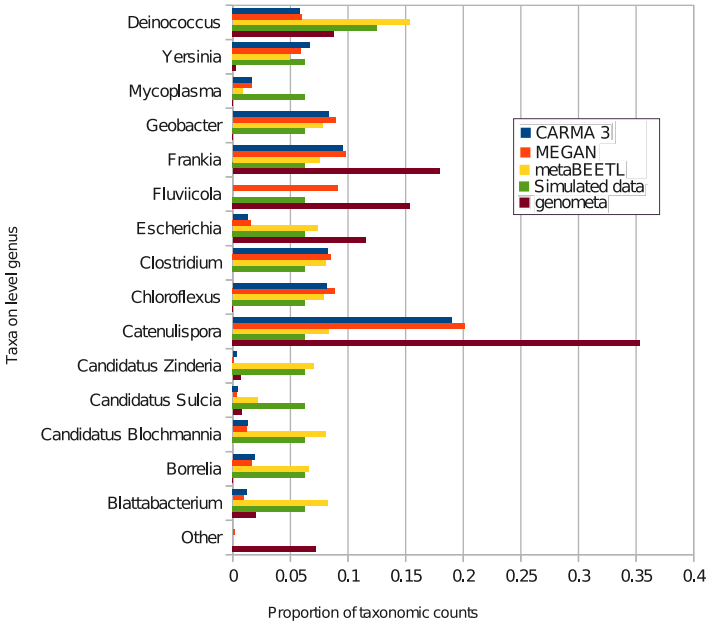

Figure A.7.: Genus-level composition of simulated data compared with classifications produced by metaBEETL, CARMA3 and MEGAN.

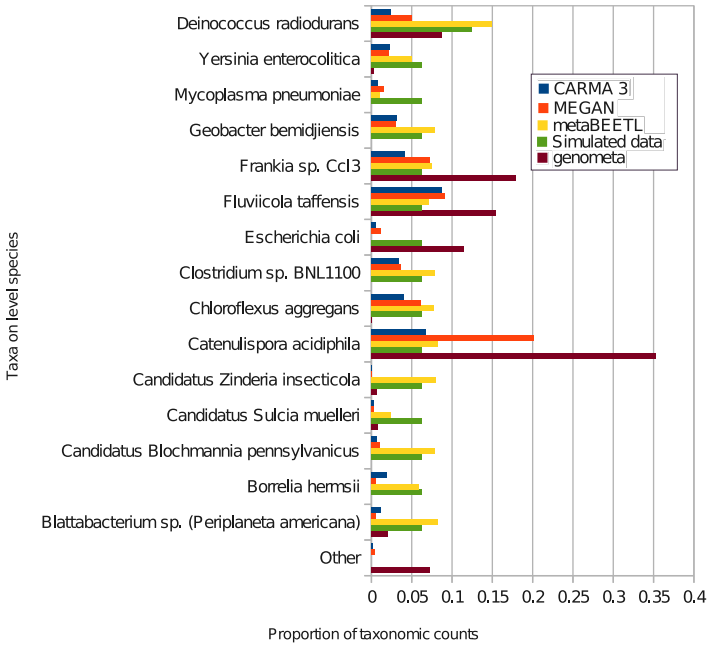

Figure A.8.: Species-level composition of simulated data compared with classifications produced by metaBEETL, CARMA3, MEGAN and Genometa.