Methods for large-scale Microbiome Analysis using MEGAN

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät der
Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Dipl.-Inform (Bioinformatik)
HANS-JOACHIM RUSCHEWEYH
aus Dettelbach

Tübingen
2014

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	22.10.2014
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter	Prof. Dr. Daniel Huson
2. Berichterstatter	Prof. Dr. Julia-Stefanie Frick

Abstract

The capability of next generation sequencers of emitting enormous volumes of data at a moderate cost has changed the field of sequence based research areas, such as metagenomics or studies estimating microbial diversity by using the 16S rRNA gene. While early studies investigated relatively small samples in isolation, current studies effectively target questions that require deeper sequencing of a larger number of samples. As a consequence of this development it becomes increasingly difficult to perform the computational component of the analysis on a desktop computer. As a matter of fact, even if the computationally intensive parts are outsourced to a more powerful environment, users still face datasets outgrowing the size of their home computers.

This development disagrees with the policy of MEGAN - a widely accepted, powerful and user-friendly tool for metagenomics - to perform qualitative analysis on local data files. To overcome this limitation, we developed MEGANServer. MEGANServer allows bioinformaticians to retain data files on a server with sufficient resources. Furthermore, we extended MEGAN to communicate with MEGANServer and by that enable researchers to perform their analysis on a home computer regardless the actual data size. Moreover, to overcome the complexity introduced by the growing number of samples, selection of datasets of interest is automated by metadata-based grouping. In addition, following the analysis strategy of the 16S rRNA studies, datasets can be opened applying different strategies, for instance as merged data, in order to provide a deeper insight on taxonomic and/or functional distribution.

Furthermore, and as a consequence of a development in which metagenomics and 16S rRNA studies are converging, we extended MEGAN to also deal with sequences that stem from a targeted approach. More precisely, we have developed a pipeline that covers the entire workflow, starting from pre-processing and, in a final step, allowing qualitative analysis using MEGAN. For that, we took advantage of a novel aligner, namely MALT, that in combination with a placement algorithm, namely the Majority Vote LCA, introduced recently in MEGAN, is not only capable of assigning more than 99% of reads to the correct genus, but lowers the rate of false positives to a value close to 0%.

We believe that, by the additional utilization of the different data access strategies implemented in MEGANServer, MEGAN in combination with MALT and the Majority Vote algorithm is now fully capable of serving as a powerful, yet user-friendly analysis tool for 16S rRNA sequencing data.

Acknowledgements

My first and foremost gratitude goes to my supervisor Prof. Dr. Daniel Huson, for accepting me as a member of his group, for the constant support and for giving me the freedom to explore. Using this freedom with all its facets was, at first, challenging but in retrospective I can say that this was probably the most valuable lesson you taught me.

I also extend my thanks to my collaborators Prof. Dr. Julia-Stefanie Frick, Prof. Dr. Barbara Stecher and Isabell Flade for the chance to take part in your exciting projects. Thank you for helping me with all the biology-related questions and for the discussions we had.

I also acknowledge the Bildungsministerium für Bildung und Forschung for the funding.

The time at the office is most valuable if shared with great people. I was really lucky to be part of a fun and stimulating environment here at the Algorithms in Bioinformatics group. Of my colleagues one person most certainly stands out. Cuong, thank you for the past two years and all the discussions we had. I wish you all the best for your future.

I'm also grateful to all my friends from Tübingen, Volkach and all around the world. You are probably not aware of this but each of you has contributed with a small fraction to my work.

But most of all I have to thank my family. Thank you for believing in me and all the encouragement and support throughout the years.

In accordance with the standard scientific protocol, I will use the personal pronoun *we* to indicate the reader and the writer, or my scientific collaborators and myself.

Contents

Chapter 1

Introduction

MEGAN is a widely accepted, powerful and user-friendly tool which allows users to perform metagenomic analysis even on a home computer. First released in 2007 [Huson et al., 2007] in order to facilitate taxonomic analysis of ancient mammoth bones [Poinar et al., 2006], it is now available in its fifth version [Huson, 2014b]. In the past years, the function pool was continuously expanded, so much so that the latest version not only supports taxonomical but also functional analysis using the SEED [Overbeek et al., 2005], KEGG [Kanehisa and Goto, 2000] and COG [Muller et al., 2010] classifications [Mitra et al., 2011a]. Furthermore, and as a result of the rapid development of sequencing technologies, reflected in plummeting prices and growing number of emitted sequences, one can determine a change in the study layout. While early metagenomic studies investigated single samples in isolation, in recent studies the focus is on collecting greater number of samples in order to identify differences or similarities among their taxonomic of functional distribution. To cope with the need of researchers, MEGAN introduced functions that support several comparison strategies [Mitra et al., 2010; Huson et al., 2009; Mitra et al., 2009].

The possibility to generate more and larger samples allows researchers to investigate metagenomic datasets at a previously inaccessible depth, but also lead to data sizes which outgrow the capacity of desktop computers of the researchers. For example, an average sized study of 12 permafrost soil samples (see [Mackelprang et al., 2011]) includes 250 million reads and requires, after alignment, 165GB of disk space. As a consequence of the growing sizes sharing of datasets with colleagues also becomes increasingly cumbersome. In order to allow researchers to perform their analysis, regardless the data sizes, on a desktop computer using MEGAN, in this thesis we present MEGANServer. With

MEGANServer one outsources the storage of metagenomic datasets to a different computer and accesses their content via MEGAN. Furthermore, driven by the accumulated fashion in which datasets are stored on MEGANServer, we will introduce new functions to MEGAN. This includes the extensive usage of metadata to identify datasets of interest, and using boolean expressions and different strategies to open datasets such as merging, splitting or extracting. The development of MEGANServer is covered in Part II.

Besides the study of metagenomics, MEGAN can also be used for studies that assess microbial diversity by analyzing sequences originating from the bacterial 16S rRNA gene [Mitra et al., 2011b, 2013]. The importance of supporting both study types is underlined by the fact that ever since the *Human Microbiome Project* both fields are converging. As a result, not only MEGAN but other tools that were initially developed for one field extended their workflows to bundle both analysis types (see MG-RAST and QIIME [Meyer et al., 2008; QIIME, 2014]) in one analysis framework.

In this thesis we will adopt the idea of Mitra et al. [2011b] of using MEGAN for visual analysis of 16S rRNA data. However, we will develop a new pipeline that covers the entire analysis process. For that undertaking, we first describe a routine for preprocessing of sequencing data in Chapter 6. Then we introduce a novel approach of accurate taxonomic placement using alignment in Chapter 7.

Additionally, and due to the nature of studies performed on sequences originating from the 16S rRNA gene which often lead to large numbers of samples, we will show in Part IV how analysis of 16S rRNA samples can profit from the accumulated fashion in which datasets are stored on MEGANServer.

Before we discuss the development of MEGANServer, we will present some background information on next generation sequencing as well as introduce the field of metagenomics in Part I.

Part I

Sequencing and Sequence Analysis

Sequencing and sequence analysis are the backbone for any metagenomic or 16S rRNA study. In this chapter we will briefly discuss sequencing technologies as well as introduce the goals behind metagenomics and 16S rRNA studies.

Chapter 2

Next Generation Sequencing

The advent of next generation sequencing (NGS) led to progress in metagenomics and other previously sequencing independent fields. Contrary to automated Sanger sequencers, the dropping prices and the enormous volumes of data generated by NGS sequencers allow researchers to address questions that quantitatively assess the impact of bacterial communities in terms of diversity and functional content.

The term next generation sequencing is intrinsically tied to pyrosequencing, a technique using bioluminescence to detect nucleotide incorporation during DNA synthesis. This methodology dates back to 1986 [Walker et al., 2007; Nyrén, 2001; Ronaghi et al., 1998] and was taken up in 2005 [Margulies et al., 2005] marking a milestone for massive parallel sequencing techniques. This approach was then made commercially available by 454 Life Sciences. A concept that also takes advantage of the emission of light as a signal for nucleotide incorporation is based on reversible dye-terminators and was made commercially available in 2004 by Solexa [Bennett, 2004] (today Illumina).

Even though Illumina and 454 are both based on a similar concept, their techniques and the results of sequencing vastly differ. As a consequence, Illumina is capable of emitting large volumes of short sequences[1] and is therefore the first choice for metagenomic studies. A 454 run, on the other hand, produces only a fraction of that data but outperforms Illumina in terms of quality and length. As a consequence, 454 sequencers are the first choice for amplicon sequencing.

Because later chapters will present in detail the pre-processing and an analysis pipeline exclusively designed for data emitted by 454 sequencers, in the following we

[1] Illumina announced the 1 Terabase run for this year. For further information, see `http://www.illumina.com/products/hiseq-sbs-kit-v4.ilmn`

would like to briefly summarize the methodology and work-flow preceding the actual sequencing.

Emulsion PCR As a sequencing-by-synthesis platform, 454 depends on a measurable light signal as a result of a nucleotide being incorporated. Since single fluorescence events remain undetected by the imaging system used, amplification of templates is required. To do so, 454 employs the emPCR [Mardis, 2008; Dressman et al., 2003] technique. First, a single template sequence is ligated to a bead. The bead is subsequently enclosed in an emulsion, and the amplification of the template sequence can be conducted, resulting in a loaded bead with approximately ten million identical sequences [Sciences, 2014]. This process is performed for the entire library in parallel, resulting in ~1-1.6 million loaded beads[2].

Loading the PicoTiterPlate$^{\text{TM}}$ The loaded beads are placed on a PicoTiterPlate. The plate contains 1.6 million wells, which are designed in such a way that exactly one bead fits in a single well.

Sequencing & Imaging Besides the beads, each well is equipped with the necessary reagents for synthesis, except for the nucleotides. They are not added because they would interfere with the synchronous fashion of incorporating one specific nucleotide and measuring the signal sequencing process. The method of sequencing is separated in 800-1,000 flows, where at each flow a new nucleotide is added to the PicoTiterPlate. Some of these nucleotides are incorporated; the unbound ones are washed away. Incorporated nucleotides trigger a biochemical process resulting in emission of light. The intensity of the light signal is stronger if, in a homopolymeric region, several nucleotides are incorporated during a single flow.

Errors It is known that homopolymeric regions increase the strength of the emitted signal in a linear fashion only when 6 or less nucleotides are incorporated [Margulies et al., 2005; Balzer et al., 2010]. Therefore, longer homopolymeric stretches would introduce an accuracy bias. Nevertheless, the main source of error is introduced as a side effect of the emPCR. The synthesis among identical sequences attached to a single bead loses its synchrony with increasing read length, influencing light signals and results in uncertain base-calls. This leads to a drop of accuracy at ~400bp [Margulies et al., 2005]. To

[2] For further information, see `http://454.com/products/technology.asp`

overcome this bias, by the end of 2013, Roche introduced an improved flow cycle, with the goal to further extend sequence length.

Chapter 3

Metagenomics & 16S rRNA Analysis

3.1 Introduction

Microbes are essential to all life. Microbes on and inside our body outnumber the number of cells we have [Berg, 1996]. Besides covering all that is living, microbes also cover the entire surface of the earth as well [Whitman et al., 1998]. Microbes, even though found everywhere, are highly specialized to the environmental factors of their ecological niche [Xie et al., 2011]. While microbes inhabiting the surface of deep-sea vents are fueled by sulfur oxidation [Sievert et al., 2008], microbes living in the human gut take a different role, express different genes and react differently to changes in the environment [Arumugam et al., 2011; Ley et al., 2006; Qin et al., 2010]. What all microbes have in common is a relatively tight relationship with their environment, other microbes and, for example, in the human gut, with the human host [Peterson et al., 2009].

The field of traditional genomics is dependent on the ability to isolate a single organism, and to culture such an organism in the absence of other microbes. This approach may conflicts and eventually presents a problem when considering the tight environmental relationship these microbes need to survive. Most microbes will not grow in isolation [Rappé and Giovannoni, 2003] and, therefore, their genomic content is not accessible. Quantitatively, this means, that more than 99% of microbes cannot be cultured with current methods and stay beyond the reach of traditional genomic research. Nowadays, the endeavors in research address not only *true* biodiversity, but also interaction patterns between microbes - issues which cannot be fully answered by genomics.

Metagenomics, on the other hand, explores the genomic content of an entire microbial community. Being culture independent, metagenomics enables one to grasp some of the 99% of the microbes not being covered by genomics, resulting in findings of new genes and species. For instance, one of the first large-scale metagenomics studies, conducted by the *Global Ocean Survey*, led to finding six million genes. New protein families were detected at a linear rate [Yooseph et al., 2007], implying that a deeper sequencing would have lead to discovery of many more new families.

Another aspect of metagenomics is to unravel community dynamics at various levels. For example, during the *Human Microbiome Project*, metagenomic samples have been taken from 242 individuals from 15 different body sites to find a link between our health and the microbiome. One of the findings of the *Human Microbiome Project* indicates that there is no single human microbiome at the taxonomic level. However, at functional level similarities across individuals were found [The Human Microbiome Project Consortium, 2012b].

Therefore, the aim of metagenomics is to study uncultured organisms in order to draw conclusions about the true diversity of microbial communities, and to explore their functional potential, their inter-community cooperation and their reaction to induced environmental change.

The study of microbial diversity using the 16S rRNA gene follows, for the most part, the same principles as metagenomics and attempts to answer similar questions. In contrary to metagenomics, for which sequences originate from the entire genome, for this approach, sequences only from the 16S rRNA gene are sequenced and used for downstream analysis. Consequently, 16S rRNA analysis is generally considered to be the first choice for taxonomic analysis, and, due to the lower costs, also used in studies where the functional content is not of interest.

In this chapter we will describe the analysis of both study types from three viewpoints. First, we will discuss typical sources of samples. Secondly, we will describe analysis goals of both fields. Finally, we will elaborate on computational aspects of analysis.

3.2 Sampling

Considering the spread of microbial life, an extensive number of different sources of samples are investigated in current studies. In order to find answers to a diverse set of questions such as the role of microbiota in carbon fixation in permafrost [Mackelprang et al., 2011], discovery of efficient biofuels [Hess et al., 2011], unraveling the evolutional development of pathogens [Schuenemann et al., 2013], or the role of the human microbiome in disease development [The Human Microbiome Project Consortium, 2012a], a great diversity of samples - soil, water, ancient bones, extreme environments - in combination with medical models, are studied.

However, the first step in all these studies is the retrieval of a single or a number of samples from a particular environment. This step is followed by the extraction of DNA. Subsequently, one of the sequencing methods, predominantly Illumina for metagenomics and 454 for 16S rRNA, is applied. The resulting sequences serve as input for downstream analysis.

Metadata Collection While early studies performed metagenomics or 16S rRNA studies on single or small number of samples, there is now an increasing number of projects that involve multiple samples collected systematically [Turnbaugh et al., 2007]. Moreover, greater attention is being paid to the general problem of recording relevant environmental parameters (so-called metadata). The importance of metadata is underlined by the fact that the *Genomics Standards Consortium* (GSC) was established in 2005 and published a minimal set of metadata (MiMS, MiGS, MIMARKS) to be collected for every experiment [Yilmaz et al., 2011]. Today, all major resources for metagenomic studies and data have implementations to store metadata compliant with the conditions of the GSC[1].

However, from a researcher's point of view, the main objective of metadata collection is the enhanced analysis potential, especially in studies that aim at comparing a larger number of samples. As a result one can correlate taxonomic and functional properties such as abundance shifts between datasets with environmental factors.

3.3 Analysis Goals

Despite the variety of problems addressed by metagenomics and 16S rRNA studies, and the differences between samples, one can condense their analysis goals to three questions.

[1] A positive side effect of metadata collection is to store datasets of former studies in a reusable fashion [Vines et al., 2014].

Who is out there? For the most part and with regard to the DNA extraction from a whole community, the microbial diversity in a sample is unknown. Thus, the first task is to quantitatively identify all species present in a sample.

What are they doing? The *Human Microbiome Project* concluded that, even though individuals' microbiota significantly differs at the taxonomic level, the functional content seems to be stable. Hence, the second question addresses the functional content of a sample. Due to the nature of targeted sequencing in 16S rRNA studies, this question can only be answered by metagenomics.

How do they compare? Studies, in particular those with a medical background, are not designed to observe samples in isolation. Instead, the focus is to identify differences on the taxonomic and functional level in samples retrieved from, for example, both experimental and control groups.

3.4 Computational Aspects

The advances in sequencing and thus the growing read counts in datasets lead to a variety of strategies and computational approaches to tackle the three analysis goals. Despite algorithmic differences, the intention is to find the correct taxonomic and functional identity for every read. To do so, most methods use a database guided approach, taking advantage of publicly available databases such as NCBI-NR or NCBI-NT [Benson et al., 2006] or REFSEQ [Pruitt et al., 2007] for metagenomics and Silva [Yilmaz et al., 2013] or Greengenes [McDonald et al., 2012] for 16S rRNA studies.

The main idea on which most methods are based is defined as follows: two sequences that share a common ancestor will be more similar when compared to an alignment of two sequences that do not share common ancestry. Therefore, tools usually compare reads against a reference database to find the *best possible* match. To keep in pace with the continuously growing datasets as well as reference databases it is computationally not feasible to search for the *optimal* match. For this reason, new algorithms apply different steps to lower the computation time. BLAST [Altschul et al., 1990], the gold standard for pairwise alignment for more than a decade, for example, applies a pre-filtering step to consecutively perform an alignment on the pre-filtered reference sequences only. Modern techniques such as Bowtie2 [Langmead and Salzberg, 2012] take advantage of the Burrows Wheeler Transformation. The transformation supports fast identification of high identity matches. Hence, these methods improve runtime significantly but lack sensitivity[2]. An alternative approach takes advantage of machine learning techniques to detect species specific patterns such as GC-content or k-mer frequencies. Two implementations that fall in the machine learning category are the Naive Bayesian Classifier [Rosen et al., 2011] for metagenomics data and the RDP classifier [Wang et al., 2007] for 16S rRNA data. Both methods perform reasonably fast but lack specificity. For example, the RDP classifier fails to assign reads to the species level. To overcome the setbacks of BLAST and NBC, namely runtime and accuracy, Nico Weber proposed a hybrid approach [Weber, 2013]. Since the performance of BLAST partly depends on the size of the reference database, it is possible to use NBC to assign reads to phylum level and consecutively run BLAST on the smaller reference databases in parallel.

At the time BLAST was developed, the limiting factor was, even though reference databases were considerably small, the main memory. Therefore, one of the main goals was to develop software that would minimize the memory footprint. Surprisingly, even though in the following years the memory prices dropped, alignment software developers still paid too much attention to minimizing the memory footprint and thereby, artificially lowered the performance of their tools in terms of speed and/or accuracy[3]. In combination with spaced-seeds techniques and using a reduced alphabet, modern mapping algorithms such as PAUDA [Huson and Xie, 2014], DIAMOND [Buchfink et al., 2014], MALT [Huson, 2014a] and to some extent Rapsearch2 [Zhao et al., 2012] perform at a reasonable speed combined with high accuracy.

In combination with tools, such as MEGAN [Huson et al., 2011], which are capable of extracting significant information from the results of the methods introduced above, one can identify the taxonomical and functional content and, thereby, find answers for the first two analysis goals.

The third question, on the other hand, is concerned with the problem of detecting differences between samples. Depending on the task, possible approaches include comparing the abundances of taxonomic and functional content (MEGAN) or applying phylogenetic methods to create principal component analysis (MEGAN, R packages such as ade4 [Dray and Dufour, 2007] or vegan [Dixon, 2003]). A relatively new method, LEfSE [Segata et al., 2011], applies statistical tests in combination with metadata on the taxonomic or functional content of a number of datasets determining the taxa or

[2] Bowtie2 can be tweaked to be more sensitive. The runtime will increase accordingly.

[3] The alignment problem can be described an tradeoff between accuracy, runtime and memory footprint.

genes which have significantly different abundances among samples.

Part II

Software for Metagenomic Analysis

This part focusses on the design ideas and implementation of MEGANServer. MEGANServer is a web application created with the purpose to store, analyze, filter and manage datasets used by the metagenomic analysis tool MEGAN. To do so, we first introduce MEGAN and the features important to MEGANServer and run through use cases. The use cases impact and drive the design process for the MEGANServer software. Finally, we discuss the implementation and how to use MEGANServer for metagenomic data.

Chapter 4

MEGAN

In this chapter we will briefly introduce MEGAN. We will walk through use cases in order to derive data access patterns. These help to design the MEGANServer software.

4.1 Introduction

MEGAN, short for MEtaGenome ANalyzer, is software developed to provide answers to the three metagenomic questions: *Who is out there? What are they doing?* and *How do they compare?* The first version of MEGAN, initially developed to analyze DNA from an ancient mammoth bone [Poinar et al., 2006], was released in 2007 [Huson et al., 2007] and is now available in its fifth version [Huson, 2014b]. MEGAN is developed with the intention of providing an user-friendly stand-alone tool for metagenomics, metaproteomics, metatransscriptomics and up to some degree, also targeted sequencing analysis.

To do so MEGAN supports four classifications, namely NCBI Taxonomy [Federhen, 2012], SEED [Overbeek et al., 2005], KEGG [Kanehisa and Goto, 2000] and COG [Muller et al., 2010]. MEGAN uses precomputed alignments from e.g. a SAM file, BLAST output or a CSV file to map reads to nodes of a classification tree. This information is used to grasp the taxonomical and functional potential of a metagenomic dataset. Furthermore and concerning the third metagenomic question, additional features of MEGAN support comparison between datasets either with regard to abundances within classifications or by applying β-Diversity measures. To provide these features MEGAN currently stores dataset specific information in a binary file, namely read-match-archive (RMA).

In this chapter we introduce MEGAN from the data access perspective with the goal to identify access patterns which help to design the MEGANServer software. The RMA file, as the current data backend, is reviewed in Section 4.2. The focus is to derive important information such as to distinguish static from dynamic data or to discover interactions between internal data structures. We then discuss common use cases in order to infer data access patterns.

4.2 Data Structures

The quality of software design depends significantly on the knowledge about data objects and the connections among them. In this section we introduce the main data structures of MEGAN.

The backbone of a metagenomic dataset is built up from a set of sequences and an associated set of alignments. In a step performed prior to analysis, MEGAN scans the content of every alignment with the intention of identifying correlated taxa and genes. All data extracted from a single alignment forms a match. A read incorporates a number of associated matches and taxonomical as well as functional identifiers. Besides reads and matches the third data structure is the header. It contains a variety of additional information such as metadata and global information about the dataset. Figure 4.1 depicts a sketch of the main data structures and their relations with each other.

4.2.1 Header

The header section embodies all information which is not directly associated to a match or a read. Its main task is to store global information such as the name of the dataset, the number of reads, the number of matches and store the metadata. Additionally, data concerning visualization, such as which nodes are visible and which are collapsed is included. The content of a header is highly dynamic[1], so that subsets change continuously. There is exactly one header per dataset and the size is limited to a few kilobytes. Therefore, the impact of any modification to this data object on the performance, is negligible.

[1] The content of dynamic data objects can be changed while using MEGAN. On the other hand, static data objects will be read-only.

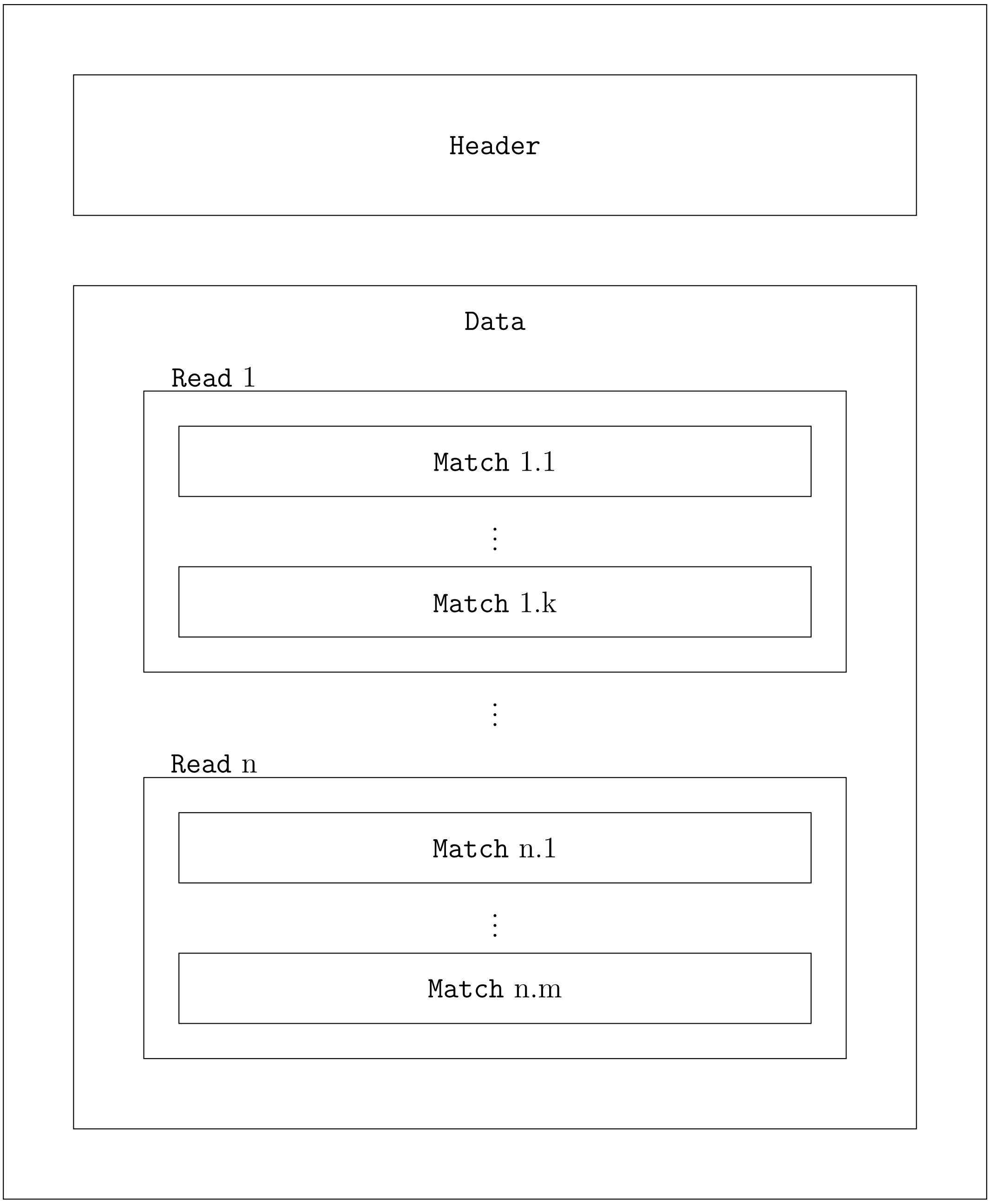


Figure 4.1: ***Overview of data structures****: A set of matches is assigned to one read. All reads form the data section. General information on the dataset is provided by the header section.*

4.2.2 Read

A read incorporates all information gathered to represent a single sequence. This includes the sequence itself, its classification identifiers, a reference to the paired sequence (if present) and a set of associated matches as shown in Listing 4.1 and Figure 4.1. Most fields are static and do not undergo any change after the enclosing read has been created. The classification identifiers may be changed in the process of classification recalculation (see Section 4.3.3).

Listing 4.1: Conceptual structure of a read

```
long uid; //Unique id of the read
String readHeader; //Sequence name
String readSequence; //Sequence

int taxonId; //NCBI - taxonomical id
int seedId; //Subsystems - functional id
int cogId; //Cog/Eggnog - functional id
int keggId; //Kegg - functional id

int readWeight; //Readweight
long mateReadUId; //Paired end information
byte mateType; //Paired end information
float complexity; //Sequence complexity

Match[] matches; //Matches
```

4.2.3 Match

A match contains all information extracted from a single alignment and is associated with exactly one read. The fields mainly span classification identifiers and scores that estimate the quality and significance of a match. All variables, as shown in Listing 4.2, are static. Hence, they will never undergo any change after being initially created.

4.3 Use Cases

Understanding how MEGAN is being used and which data is being accessed at which time is critical information when designing a data backend. This applies especially for software dealing with large volumes of data, e.g. MEGAN datasets can easily exceed 50GB.

Listing 4.2: Conceptual structure of a match

```
long uid; //Unique id of the match
String alignment; //Alignment text

int taxonId; //NCBI - taxonomical id
int seedId; //Subsystems - functional id
int cogId; //Cog/Eggnog - functional id
int keggId; //Kegg - functional id

float bitScore; //Quality measure
float expected;; //Quality measure
String refSeqId; //Id of the refSeq database
float percentIdentity; //Alignment coverage
```

To do so the first step in software design is to analyze typical use cases for data access patterns, for example. Use cases help to untangle what seems to be uncoordinated access to the file system and reveal patterns. Patterns are very important in terms of prediction of later data access. For example, with the knowledge that a data object will be requested shortly after its first usage, it is beneficial to the overall performance to apply caching in order to avoid a second, potentially slow, file access.

We will go through a number of typical use cases with the goal of identifying hidden data access patterns. Additionally, for every use case in which the result size is not predetermined, we will discuss the performance impact if the result sizes are arbitrary large.

4.3.1 Taxonomic Overview

As discussed in the introduction of this chapter, the main visualization of MEGAN is built of a tree derived from one of the four classifications (COG, NCBI, SEED and KEGG). For example, Figure 4.2 depicts the distribution of reads among taxa for a permafrost dataset [Mackelprang et al., 2011] collapsed at phylum level[2]. The size of the node correlates with the number of reads assigned. For this view, MEGAN supports typical tree operations such as collapsing and expanding of nodes to explore abundances at different levels.

The view is composed of two data objects. First, there is the raw tree structure, not

[2] There are no differences among classifications at the data access level. Therefore, and for brevity, when we discuss examples from the taxonomic point of view, the same statements apply also to the three other classifications.

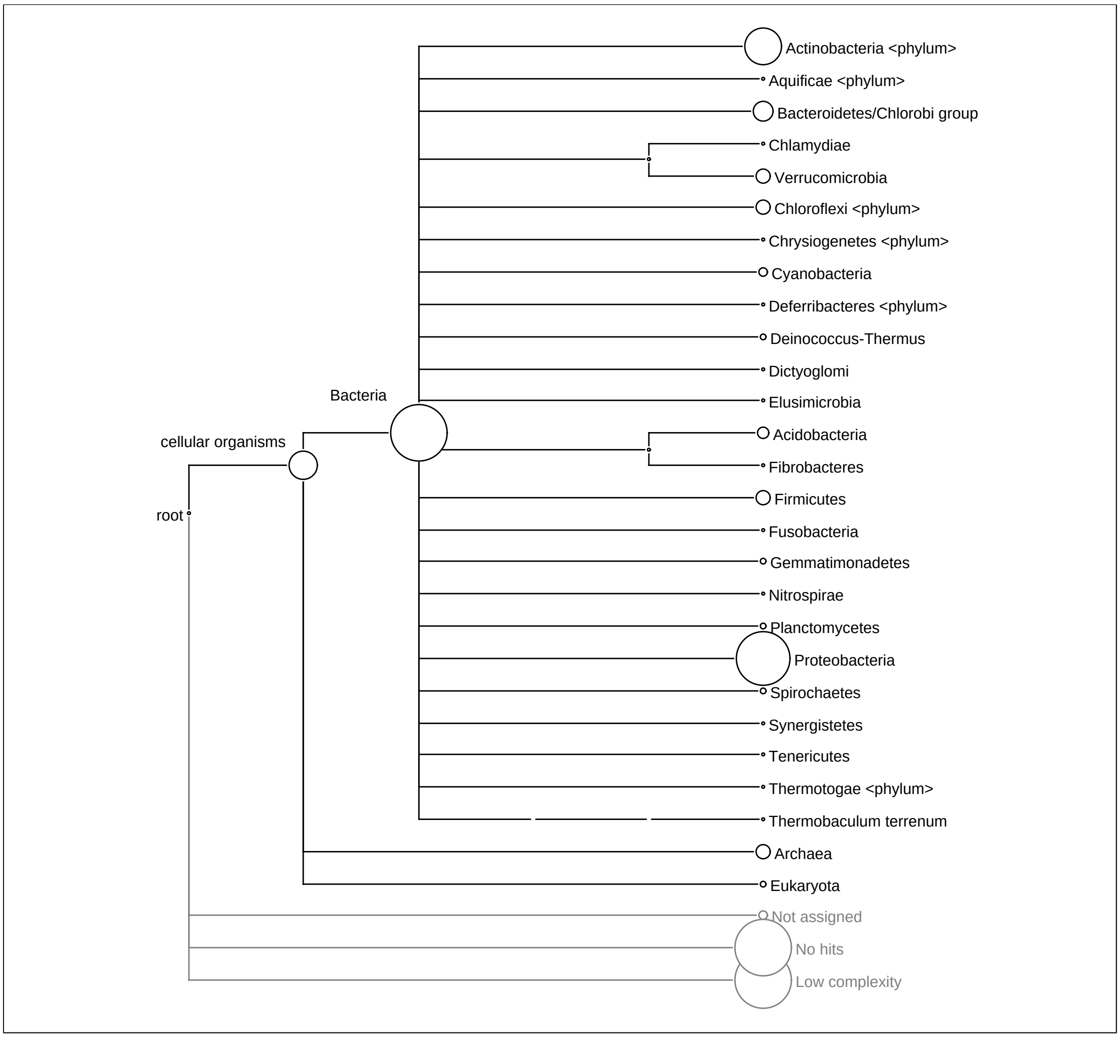


Figure 4.2: Tree representation of reads on the NCBI taxonomy on the phylum level. The size of each node indicates the number of reads assigned. Nodes may be collapsed or expanded to different taxonomic levels.

specific to the dataset but specific to a classification. Therefore, the tree is stored in a file other than the actual dataset and is not of further interest.

The second object reflects the taxonomic composition of a single sample. It is composed of the taxonomical information of all reads and represents the number of reads assigned to a node (taxon). The generic structure is shown in Equation 4.1. Equation 4.2 and 4.3 depict the mapping for *Actinobacteria* and *Chlamydiae* from the permafrost dataset depicted in Figure 4.2.

$$\textit{Node} \rightarrow \textit{Number of reads assigned} \tag{4.1}$$

$$\textit{Actinobacteria} \rightarrow \textit{190,092} \tag{4.2}$$

$$\textit{Chlamydiae} \rightarrow \textit{1,735} \tag{4.3}$$

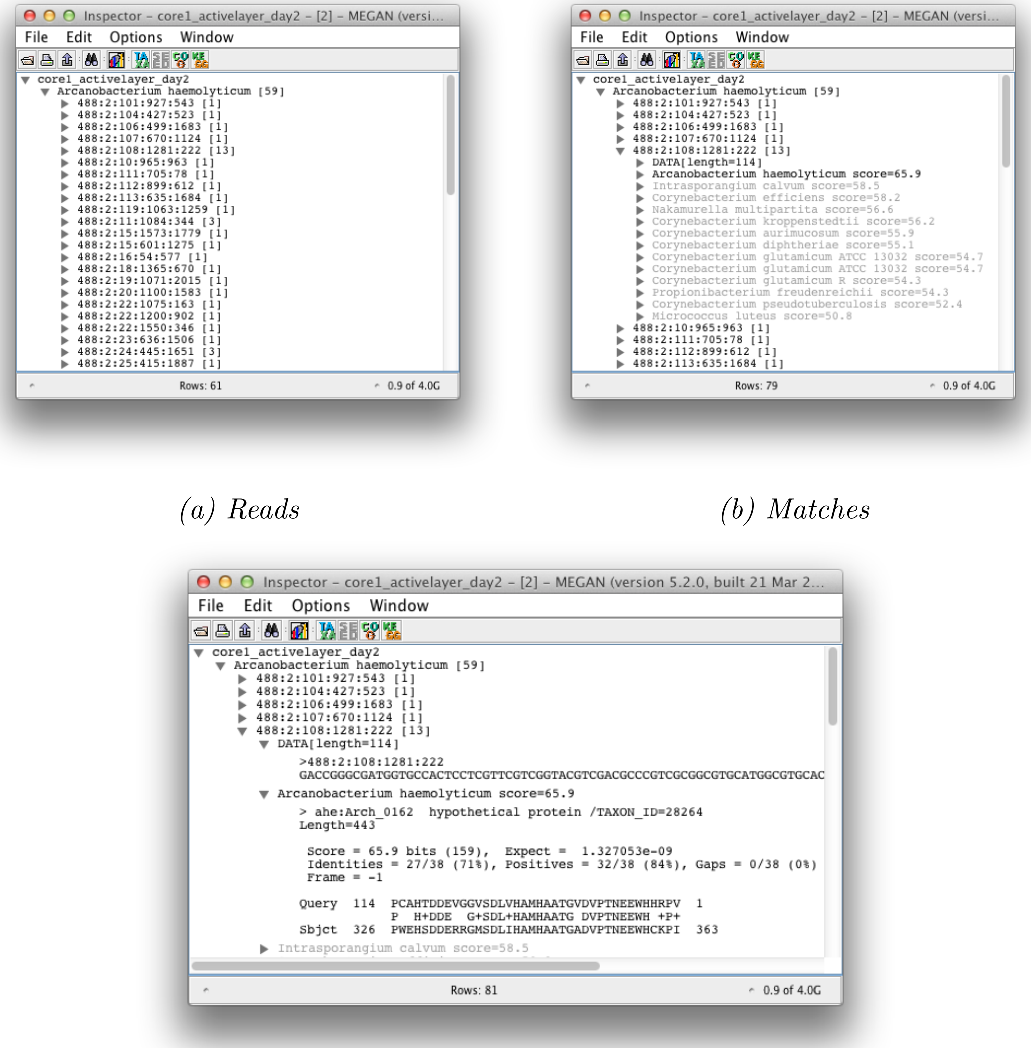
MEGAN requests the mapping when opening a data file. Considering the relatively small number of taxa present in a sample (rarely more than 1,000), the size of the mapping rarely exceeds a few kilobytes. On the other hand considering the large number of reads incorporated in metagenomic datasets, it is a computationally expensive step to access the taxonomical information of all reads in order to create this mapping.

4.3.2 Inspector

The representation as a tree allows one to explore the taxonomic composition of a sample in an aggregated fashion, omitting all read and match specific information. To explore the content of reads and matches in a more detailed way, MEGAN provides an additional visualization, the Inspector. After selecting a taxon, one can explore the content of assigned reads, as seen in Figure 4.3. The structure of the view divides data access in a sequence of three steps:

- As shown in Figure 4.3a, the initial view incorporates the read names in combination with the number of associated matches, assigned to a specific taxon, here *Arcanobacterium haemolyticum*.
- On read selection all associated matches are uncollapsed and presented by the taxon name (see Figure 4.3b).
- Finally, Figure 4.3c shows the lowest level of data access. After selecting a single match, its alignment text is displayed.

Even though MEGAN handles these three steps as independent data requests, one can immediately recognize a pattern. Reads are accessed and matches being counted in a first step. A subset of these matches is accessed a second time when exploring their taxonomic identity and individual matches are requested a third time in order to inspect their alignment. It is beneficial to the performance to store matches, which have been accessed in the second step, in a cache to avoid one expensive I/O request.



```
▼ core1_activelayer_day2
  ▼ Arcanobacterium haemolyticum [59]
    ▶ 488:2:101:927:543 [1]
    ▶ 488:2:104:427:523 [1]
    ▶ 488:2:106:499:1683 [1]
    ▶ 488:2:107:670:1124 [1]
    ▶ 488:2:108:1281:222 [13]
    ▶ 488:2:10:965:963 [1]
    ▶ 488:2:111:705:78 [1]
    ▶ 488:2:112:899:612 [1]
    ▶ 488:2:113:635:1684 [1]
    ▶ 488:2:119:1063:1259 [1]
    ▶ 488:2:11:1084:344 [3]
    ▶ 488:2:15:1573:1779 [1]
    ▶ 488:2:15:601:1275 [1]
    ▶ 488:2:16:54:577 [1]
    ▶ 488:2:18:1365:670 [1]
    ▶ 488:2:19:1071:2015 [1]
    ▶ 488:2:20:1100:1583 [1]
    ▶ 488:2:22:1075:163 [1]
    ▶ 488:2:22:1200:902 [1]
    ▶ 488:2:22:1550:346 [1]
    ▶ 488:2:23:636:1506 [1]
    ▶ 488:2:24:445:1651 [3]
    ▶ 488:2:25:415:1887 [1]
```

(a) Reads

```
▼ core1_activelayer_day2
  ▼ Arcanobacterium haemolyticum [59]
    ▶ 488:2:101:927:543 [1]
    ▶ 488:2:104:427:523 [1]
    ▶ 488:2:106:499:1683 [1]
    ▶ 488:2:107:670:1124 [1]
    ▼ 488:2:108:1281:222 [13]
      ▶ DATA[length=114]
      ▶ Arcanobacterium haemolyticum score=65.9
      ▶ Intrasporangium calvum score=58.5
      ▶ Corynebacterium efficiens score=58.2
      ▶ Nakamurella multipartita score=56.6
      ▶ Corynebacterium kroppenstedtii score=56.2
      ▶ Corynebacterium aurimucosum score=55.9
      ▶ Corynebacterium diphtheriae score=55.1
      ▶ Corynebacterium glutamicum ATCC 13032 score=54.7
      ▶ Corynebacterium glutamicum ATCC 13032 score=54.7
      ▶ Corynebacterium glutamicum R score=54.3
      ▶ Propionibacterium freudenreichii score=54.3
      ▶ Corynebacterium pseudotuberculosis score=52.4
      ▶ Micrococcus luteus score=50.8
    ▶ 488:2:10:965:963 [1]
    ▶ 488:2:111:705:78 [1]
    ▶ 488:2:112:899:612 [1]
    ▶ 488:2:113:635:1684 [1]
```

(b) Matches

Inspector – core1_activelayer_day2 – [2] – MEGAN (version 5.2.0, built 21 Mar 2…
File Edit Options Window
Rows: 81
0.9 of 4.0G

```
▼ core1_activelayer_day2
  ▼ Arcanobacterium haemolyticum [59]
    ▶ 488:2:101:927:543 [1]
    ▶ 488:2:104:427:523 [1]
    ▶ 488:2:106:499:1683 [1]
    ▶ 488:2:107:670:1124 [1]
    ▼ 488:2:108:1281:222 [13]
      ▼ DATA[length=114]

          >488:2:108:1281:222
          GACCGGGCGATGGTGCCACTCCTCGTTCGTCGGTACGTCGACGCCCGTCGCGGCGTGCATGGCGTGCAC
      ▼ Arcanobacterium haemolyticum score=65.9

          > ahe:Arch_0162  hypothetical protein /TAXON_ID=28264
          Length=443

           Score = 65.9 bits (159),  Expect =  1.327053e-09
           Identities = 27/38 (71%), Positives = 32/38 (84%), Gaps = 0/38 (0%)
           Frame = -1

          Query  114  PCAHTDDEVGGVSDLVHAMHAATGVDVPTNEEWHHRPV  1
                      P  H+DDE  G+SDL+HAMHAATG DVPTNEEWH +P+
          Sbjct  326  PWEHSDDERRGMSDLIHAMHAATGADVPTNEEWHCKPI  363
      ▶ Intrasporangium calvum score=58.5
```

(c) Alignment

Figure 4.3: ***Inspection of Arcanobacterium haemolyticum:*** *59 reads have been assigned to the Arcanobacterium taxon. (a) The names of the 59 reads are listed below the taxon name. Read '488:2:108:1281:222' has 13 matches. The taxon names of the matches are listed in (b). The alignment for one match is shown together with the read sequence in (c).*

4.3.3 (Re)calculation of Classifications

The ultimate goal of MEGAN is to find the correct taxonomic and functional identity for every read as the quality of downstream analysis greatly profits from an accurate assignment. To do so, MEGAN applies a two step algorithm. First, the initial taxonomic and functional identity is based on the taxa and genes found in its matches. The matches are pre-filtered by quality, so that only significant alignments will be taken to consideration. In the second step, reads assigned to rare taxa are considered to be noise, and are reassigned to more frequent but related taxa. Finally, the resulting taxonomic and functional assignments for each read are written back to the data file.

Within the scope of MEGAN, this is the most resource intensive workflow, not only in terms of main memory consumption but also in terms of runtime.

First, MEGAN retrieves all reads one by one and fetches the classification information of their associated matches. Even though matches are pre-filtered beforehand and therefore reduce the quantity of data to be transferred, in software design one has to consider the worst case in which all matches have to be retrieved. After finishing the calculation one has to update the classification identifiers in all reads.

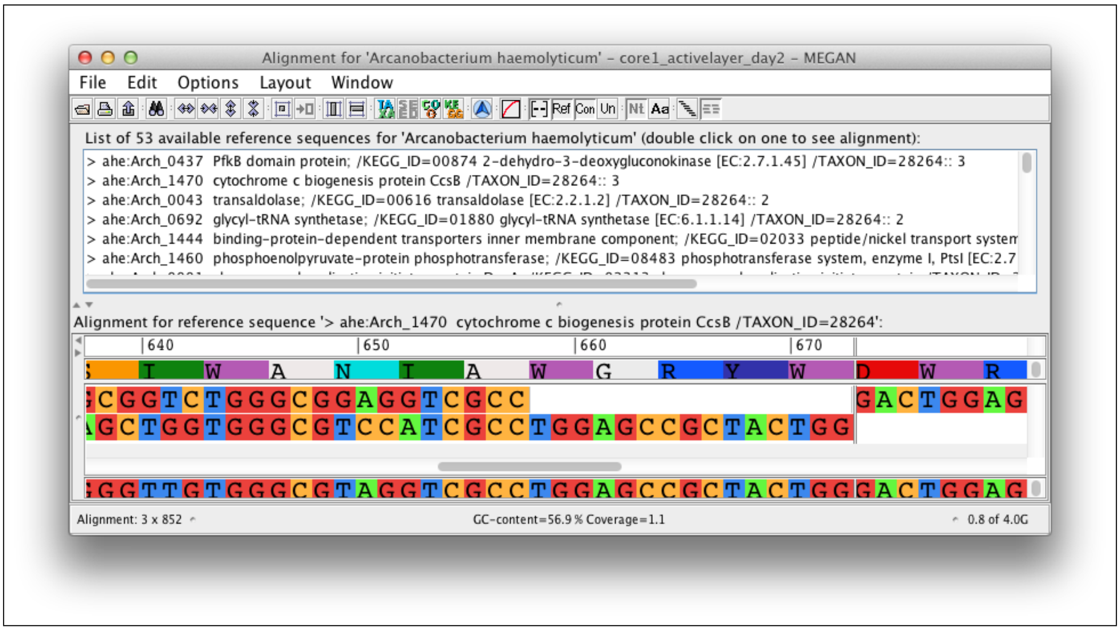
The runtime of this task can be greatly reduced by accessing data inside reads and matches in a selective way and only transmitting data that is needed for that specific task. For the recalculation of classifications, only the taxonomical and functional identifiers of each match are required. Therefore, transmitting the alignment text does not give any additional information, but only inflates the data to be transferred.

4.3.4 Alignment Viewer

With the release version of 5, a new feature introduced to MEGAN allows one to explore alignments in a more elaborate manner, when compared to the traditional Inspector window (see Figure 4.3c). The Alignment Viewer is capable of visually aligning multiple sequences to a number of references. References and sequences are extracted from reads and their associated matches.

Similar to the routine which uses the Inspector window, reads from a taxon are requested, together with a set of quality-filtered matches. Alignment text is extracted from matches used as a reference and the read sequences are aligned against this reference.

In contrast to the previous use cases where alignment text was accessed in a selective fashion (Inspector, see Section 4.3.2) or completely omitted (Recalculation of classifications: see Section 4.3.3), the Alignment Viewer requests and accesses alignment text for all matches.



Alignment for 'Arcanobacterium haemolyticum' – core1_activelayer_day2 – MEGAN
File
Edit
Options
Layout
Window
Ref
Con
Un
Nt
Aa

```
List of 53 available reference sequences for 'Arcanobacterium haemolyticum' (double click on one to see alignment):
> ahe:Arch_0437  PfkB domain protein; /KEGG_ID=00874 2-dehydro-3-deoxygluconokinase [EC:2.7.1.45] /TAXON_ID=28264:: 3
> ahe:Arch_1470  cytochrome c biogenesis protein CcsB /TAXON_ID=28264:: 3
> ahe:Arch_0043  transaldolase; /KEGG_ID=00616 transaldolase [EC:2.2.1.2] /TAXON_ID=28264:: 2
> ahe:Arch_0692  glycyl-tRNA synthetase; /KEGG_ID=01880 glycyl-tRNA synthetase [EC:6.1.1.14] /TAXON_ID=28264:: 2
> ahe:Arch_1444  binding-protein-dependent transporters inner membrane component; /KEGG_ID=02033 peptide/nickel transport system
> ahe:Arch_1460  phosphoenolpyruvate-protein phosphotransferase; /KEGG_ID=08483 phosphotransferase system, enzyme I, PtsI [EC:2.7

Alignment for reference sequence '> ahe:Arch_1470  cytochrome c biogenesis protein CcsB /TAXON_ID=28264':
     |640      |650      |660      |670
 S I W A N I A W G R Y W D W R
 GCGGTCTGGGCGGAGGTCGCC                GACTGGAG
 AGCTGGTGGGCGTCCATCGCCTGGAGCCGCTACTGG

 GGGTTGTGGGCGTAGGTCGCCTGGAGCCGCTACTGGGACTGGAG

Alignment: 3 x 852          GC-content=56.9 % Coverage=1.1          0.8 of 4.0G
```

Figure 4.4: Alignment Viewer for Arcanobacterium haemolyticum. Sequences are extracted from reads and aligned against references.

4.4 Conclusion

In this chapter we discussed a subset of MEGAN's functionality with the focus on the I/O. While programs that handle a small amount of data do not need to focus on data access, in the case of metagenomics, considering rapidly growing read counts, efficient I/O handling is crucial to the performance of software. With regard to the ultimate goal of designing software capable of storing and accessing MEGAN's datasets in an efficient manner, we summarize lessons learned in this chapter:

Ordered vs. unordered Reads are not expected to be sorted. The order in which they are sent to MEGAN can be arbitrary. The matches, on the other hand, must be ordered by decreasing quality.

Aggregate The Main Viewer requires an aggregated mapping of the taxonomic identity of all reads. We expect that the performance boost of a pre-calculated mapping outweighs the rise in complexity to maintain the mapping at an up-to-date status.

Selectivity One paradigm when handling large quantities of data is to be as selective as possible in order to avoid overhead. When MEGAN requests a match only to extract the taxonomical identifier, it is not beneficial to the performance to transmit also the alignment text. Only access and transmit data which is requested.

Caching I/O costs, even to fast database instances are expensive. The access pattern of the Inspector shows that MEGANServer can profit from caching of previously loaded reads and matches.

Chapter 5

MEGANServer

In this chapter we will introduce MEGANServer, an add-on to MEGAN that is capable of outsourcing the often very large metagenomic datasets on to a web server environment.

5.1 Introduction

The main focus, when implementing the metagenomic analysis tool MEGAN, is on the usability. Every user, tech savvy or not, should be capable of downloading, installing and running the software within a couple of minutes. The same principle applies to the implementation of the graphical user interface. Additionally and in contrary to other tools that promise great functionality but lack usability, MEGAN is able to close the gap between functionality and being user-friendly.

Driven by the huge success of next generation sequencing technologies leading to large volumes of data at moderate costs, it has become a trend to apply metagenomic sequencing to a large range of research areas (see Review *Metagenomics Research Review*, [Illumina, 2012]). While an analysis pipeline could be carried to execute on a normal home computer before, nowadays these pipelines need to be executed on hardware with larger computing power, such as servers or clusters. As a result, plenty of metagenomic pipelines such as MG-RAST [Meyer et al., 2008] or Camera [Seshadri et al., 2007], to name the most popular ones, have emerged in the past few years that offer researchers the possibility to upload their data files and perform one click begin-to-end analysis.

MEGAN, on the other hand, follows a different strategy. Analysis pipelines should be flexible and data should remain the researcher's property. Hence, the traditional approach of how MEGAN handles metagenomic datasets is to perform analysis on a home computer and store data files locally. Local access is easy to implement, relatively fault safe and enables fast access. But, there are obvious disadvantages. One has to store and organize large datasets, often spanning several gigabytes, locally, and is forced to duplicate data in order to share the analysis with colleagues. It is only a matter of time, till metagenomic datasets will outgrow the researcher's desktop computer.

To overcome this setback we introduce an add-on to MEGAN, namely MEGAN-Server. The goal of MEGANServer is to allow users to move their datasets from their hard drive to a web server environment without losing the comfort of accessing their datasets in the same way as when they would be stored locally. This approach works well, considering that, even though the average size of metagenomic data easily exceeds several gigabytes, the actual data requests of MEGAN span between a few kilobytes to a few megabytes.

This chapter is divided in four sections. First, we explain the ideas and the goals behind the MEGANServer project. Secondly, we define the specifications and requirements. Next, we describe the design of the MEGANServer software. Finally, we show how MEGANServer can be used as a MEGAN data backend and also, how it extends the current functionality.

5.2 Ideas & Goals

Handling metagenomic datasets in times in which not only the number of sequences per sample but also the number of samples are growing is challenging. To cope with the raw quantity of data, MEGAN introduced already three data formats, each of them replacing the biases of the former one but facing new challenges. The first version, using cleartext as storage solution, reached its limits very early. The second version, and the first variant of the so called read-match-archive (RMA), used compression techniques and changed the format from cleartext to binary. The files shrunk to smaller sizes but the file structure made updates computationally costly. For example, the recalculation of the classifications required major reorganization of the file structure and therefore, performed poorly in terms of runtime. The third version, the RMA2 format, removed this bias by differentiating between static and dynamic content. Yet the performance, in terms of creation, cannot keep up with the speed in which new aligners e.g. DIAMOND [Buchfink et al., 2014] emit results. RMA3 is currently under development.

As a result of this development, in 2010 we started working on different strategies to handle metagenomic datasets. In the diploma thesis (see [Ruscheweyh, 2010]) we presented an alternative approach using relational databases with the goal of replacing the conventional file structure. As shown in the thesis, relational databases outperform conventional file structures, but only under the condition that the data is relational and dynamic. This applies only to a subset of MEGAN's data. While the classification identifiers, for example, profit from their relational nature, accelerating access and introducing new features to MEGAN, the static data such as parts of the read object and all data of a match object inflate the database and therefore introduce a lot of negative side effects.

Even though the software worked on a local computer, we faced numerous problems when setting up the program on a computer cluster. For example, the lack of multi-user functionality and the fact that the implementation lacked performance, due to complexity, using ssh for remote access were major setbacks. With that in mind, we launched the MEGANServer project. The goal was to develop software capable of storing and organizing metagenomic datasets remotely and which are exclusively accessible via MEGAN. The performance should be comparable to that of local access. Access should be based on modern web service technology to overcome remote access problems. MEGANServer must be capable of handling multiple users and support some basic level of security.

Additionally, a previously unseen aspect became increasingly relevant. Metagenomic sequencing and analysis underwent a shift from exploring single datasets in isolation to analyzing communities over a period of time or before and after a medical treatment. Researchers weren't looking at individual samples anymore but rather were trying to identify the key taxa or genes in certain scenarios. "*How do the samples from the sick individuals compare against the healthy ones?*", "*Can we identify the taxa which drive disease development?*" Rare taxa or genes considered to be noise in individual samples would be lifted above the detection limit if similar samples could be pooled.

Considering the accumulated fashion in which datasets are stored on MEGANServer using relational database technology, merging, pooling, splitting or simply searching would be computationally feasible. As a consequence, we decided to update the project goal. Information collected along with the actual sequencing sample, so called metadata, would be incorporated in MEGAN datasets. This data could consequently be used to search for datasets of interest or applied to differentiate between different groups, subsequently merged and then compared in MEGAN.

5.3 Planning Phase

It is known that the quality of software depends on the time invested in the planning phase. Even though there are approaches such as extreme programming which follow different paradigms, the traditional strategy in software development reserves a maximum of one third of project time for coding [Brown, 2013; Anderson et al., 2010]. For the two thirds dedicated to planning, the field of software engineering has developed a wide range of strategies such as the prototype, the incremental model and scrum [Schwaber, 1997]. For the development of MEGANServer we decided to follow the 'Big Design Up Front' (BDUF) strategy [Brown, 2013]. In contrast to other software development models, BDUF follows the principle of creating complete and well designed software all at one go rather than starting off with a small prototype and then extending the functionality step by step. The obvious disadvantage of the model is the static character of the software, and the fact that the development process causes belated deployment. On the other hand, if the BDUF strategy is based on a working prototype, like in our case, these biases are mostly removed.

The BDUF principle splits software development in six blocks, namely requirements, specification, design, implementation, verification and maintenance. The two last blocks are omitted for brevity. Since a discussion on the implementation would go beyond the scope of this thesis, we will cover only the essentials in the design block.

5.3.1 Requirements

The purpose of the MEGANServer project is to liberate MEGAN users from the need to store large datasets on their local computer. Data should be stored at a web server and accessed through MEGAN. The development process starts off by defining hard and soft criteria the software has to meet.

Accessibility & Performance MEGANServer has to grant easy and fast access to incorporated datasets. It should not matter to the user whether the current sample is stored on the local filesystem or it is accessed through a web service. The performance of standard commands executed against a remote dataset must be comparable to commands executed locally. In order to avoid firewall or proxy problems, traffic has to follow the demands of a standard protocol and should be routed through a standard port.

Functionality All functions implemented in MEGAN and supported by local storage must be supported by MEGANServer as well. Additionally, due to the accumulated fashion in which datasets are stored, new functions can be introduced. An interface to store and utilize metadata has to be included. Merging of datasets should be supported.

Security Datasets managed by a MEGANServer instance must be secured to prohibit unwanted access. Only dataset owners can access and/or grant access to their data.

Extendibility The program MEGAN is under ongoing development. New visualizations and analysis tools are continuously being added. MEGANServer must be extendable to support continuing development without large changes in design.

Fault Proof If MEGAN loses the connection to MEGANServer in the process of transmitting data, MEGANServer should recognize the disconnection to free memory and close connections to databases.

ACID MeganServer must be capable of handling access from multiple users. Updates of one user should not be allowed to affect reads of other users. In computer science the different flavors of data security are described as the ACID (Atomicity, Consistency, Isolation and Durability) criteria. The first three criteria must be supported by MEGANServer, or in other words, MEGANServer must be transactional. The fourth criteria is the task of a database management system.

Uploading - Bypass Web services are not designed to support fast upload of large quantities of data. To handle the sheer size of metagenomic datasets, we require an alternative approach for the upload of datasets to a MEGANServer instance.

5.3.2 Specification

‘Don’t repeat yourself’ is one of the fundamental rules in software development. If the result of another software project publishes a solution to a problem you are facing, why not take advantage of that piece of software? The rule is so fundamental that entire frameworks are wrapped around that idea, such as Ruby on Rails[1]. Java, undoubtably one of the most used programming languages [TIOBE, 2014], comes with a whole variety

[1] http://rubyonrails.org/

of public software packages which are free for use. For example, in the central Maven Repository, a database that curates software packages build with Apache's Maven[2], one can find source code from over 70,000 unique software projects [Sonatype, 2014].

In this section we will introduce a number of different libraries and frameworks which will be used throughout the entire implementation process.

Spring Framework

One idea behind the Java language development was to take the pain out of certain fields of software development and furthermore to create programs that would run, even if precompiled, everywhere. Nowadays, Java runs on over 8 billion devices [Oracle, 2014] e.g. the wrapper of the Android operating system[3] is entirely written in Java. Unfortunately, in the context of enterprise technology, Java offers with JavaEE[4] a solution that is considered to be rather poor:

> *"The projects using JavaEE technology have to place more emphasis on satisfying specific API's rather than developing actual business logic."*
> [Wolff, 2010] (translation mine)

The Spring Framework[5], first introduced in 2002 [Johnson, 2004], is one of the approaches which combine enterprise computing and Java. The primary concern is to be both comprehensive and modular. Spring handles source code that is not part of the logic as well as it offers standardized interfaces for all important business purposes e.g. security, data access, cloud computing, integration.

We will take advantage of the Spring Framework throughout the entire implementation. However, the nature of Spring is that it serves its purpose best when it is invisible to the actual logic, not implementing its own features but bringing many technologies together. Even though all components we will introduce will be managed by Spring, we will discuss its impact only where necessary.

[2] `http://maven.apache.org/`

[3] `http://www.android.com/`

[4] `http://www.oracle.com/technetwork/java/javaee/overview/index.html`

[5] `http://spring.io/projects`

Servlet Container

The nature of Java web applications imposes the necessity to launch these in a so-called servlet container. For this purpose we chose Tomcat[6]. There are Tomcat installers for all operating systems and the configuration process is relatively simple. We want that MEGANServer can be installed by most users. Tomcat seems to offer suitable features to support this goal.

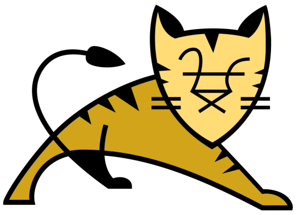


Figure 5.1: Tomcat

Storage

Since early versions relying on relational databases only proved to violate our performance requirement, we decided to divide objects in a dataset in two groups. The first group, mainly classification data, would remain in the relational database. The static content of reads (see Section 4.2.2) and matches (see Section 4.2.3), on the other hand, should be stored in a more appropriate data format.

PostgreSQL As a relational database management system we chose PostgreSQL[7]. PostgreSQL is an open source project with contributions from a huge and helpful community. It has also proven to be a suitable solution to store and manage large data by supporting a wide range of tweaking parameters. For ease of use we also tried Java database management systems, such as H2[8], which incorporate all logic in a Java archive thereby omitting the entire installation process. Unfortunately, even though the developers claim differently, H2 does not seem to be designed for large data as there is a significant drop in performance when facing the size of metagenomic data.

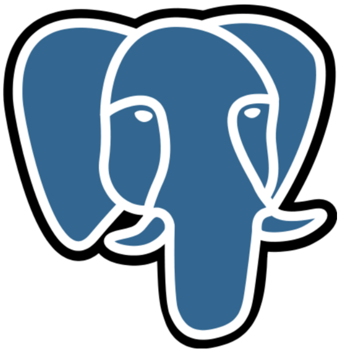


Figure 5.2: PostgreSQL

[6] http://tomcat.apache.org/

[7] http://www.postgresql.org/

[8] http://www.h2database.com/html/main.html

CouchDB We decided to use CouchDB[9] as a storage solution for the static content of reads and matches. Additionally, we will use lightcouch[10] as the Java interface. CouchDB is a simple document orientated NoSQL database management system. Each document, for example, a match or a read, is transformed in JSON and subsequently stored in a tag value manner. The tag, in our case either the read or the match identifier, serve as input for a B+ tree index which is used to provide ultra fast access to single documents.

Figure 5.3: CouchDB

Queuing

One of the main goals of MEGANServer is to provide fast access to reads and matches. Normally, for example, following the Alignment Viewer use case (see Section 4.3.4), one does not know the number of reads a request will return. If only a small number of reads is requested, one can load all reads from the databases, package them in a single object and transfer this package to MEGAN. This strategy, applied to a request that delivers a large number of reads, has the potential to stall MEGAN and introduce memory leaks in MEGANServer. A naive solution would be to apply streaming techniques on both sides, transferring read by read. This conflicts with the non-negligible overhead of serializing Java objects in order to transfer them via a web connection. The optimal solution incorporates both approaches, supports streaming on chunks of reads.

We solved this issue by implementing an additional software, namely the JobQueue. The input of the JobQueue is a request of MEGAN, asking for an indefinite number of reads. The JobQueue hands out tickets under which MEGAN iteratively requests small chunks of the reads, therefore solving both, the stalling and the memory problem. Unused tickets, for example, when a MEGAN client crashes during access, will be automatically invalidated and the memory freed.

Metadata

MEGANServer allows to gather a larger number of datasets in one repository. Search methods need to be installed to provide users functionality to identify datasets of interest. Besides basic text search on names or metadata of datasets, we decided that our software

[9] `http://couchdb.apache.org/`

[10] `http://www.lightcouch.org/`

would profit from a feature that allows the selection of all datasets that fall in a certain scenario. For example, if one is interested in including all datasets that originate from patients that are both, female and sick, one has to apply a boolean expression such as:

$$\text{`}\textit{Gender}\text{'} = \text{`}\textit{Female}\text{'}\ \textbf{AND}\ \text{`}\textit{Health Status}\text{'} = \text{`}\textit{Sick}\text{'} \tag{5.1}$$

For the purpose of parsing and evaluating a boolean expression we chose the Mozilla Rhino[11] library which is a Javascript engine entirely written in Java.

Web Service

Since MEGANServer and MEGAN run on different virtual machines we need to establish inter-process communication in order to transfer data between both JVMs. Java is shipped with an inbuilt solution for that task, namely remote method invocation (RMI)[12]. Using the Java serializer and its own protocol, RMI is capable of fast data transfer at a relatively low overhead. However, RMI is not an option because it uses its own protocol and not a standard protocol such as http. Therefore RMI is most certainly blocked by any sensitive firewall or proxy. We believe that the traditional web services based on the SOAP [Curbera et al., 2002] or REST [Fielding and Taylor, 2002] specifications such as Axis2[13] or Jersey[14] are no options either. Cleartext serializers would create large overhead and thereby throttle the speed. Since the communication will be between two Java programs, the optimal solution is to use a web service that relies on a Java serializer (size and speed) and uses the http protocol (accessibility). Potential candidates include Hessian2[15], Burlap[16] and the HttpInvoker[17]. As seen in tests (see [Miquel, 2014], [wuqingren2316, 2014]), the HttpInvoker achieves a performance comparable to RMI, even for larger objects. The main reason seems to be that HttpInvoker relies on the Spring Java serializer in comparison to the other two implementations that take advantage of the standard Java serializer. Hessian2 and Burlap perform well transmitting small objects but fail to efficiently serialize large objects. Therefore, we chose

[11] `https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Rhino`

[12] `http://www.oracle.com/technetwork/java/javase/tech/index-jsp-138781.html`

[13] `http://axis.apache.org/axis2/java/core/`

[14] `http://jersey.java.net/`

[15] `https://github.com/takafan/hessian2`

[16] `http://www.caucho.com/resin-3.0/protocols/burlap.xtp`

[17] `http://docs.spring.io/spring/docs/4.0.5.RELEASE/spring-framework-reference/html/remoting.html#remoting-httpinvoker`

to use the HttpInvoker for our web service.

Security

The consequence of offering a service located on a web server is the exposure of data and functionality to the internet. Access control such as authentication is essential. Additionally, in order to support different types of permission, such as read/write or read only access, an authorization process has to be employed. Spring Security[18] offers a solution to both authentication and authorization, and supports simple configuration via annotations.

We will use Spring Security for authentication and authorization in combination with digest authentication so that passwords are not sent in cleartext over the network.

Caching

One can significantly improve performance by caching reads and matches (see use case: Inspector 4.3.2). There are highly advanced implementations such as Redis[19], Java Caching System[20], EhCache[21] or MemCached[22]. However, even though all of these implementations offer striking performance, the increase of complexity these packages would cause in our software is undesirable. We chose a simple cache implementation distributed along with the Google Guava package[23].

Transactions

The essence of server software is rooted in the idea of offering services and data to a larger number of users. In comparison to single user software, one has to employ additional data safety strategies. Data can be invalid and inconsistent at certain points of its lifespan. For example, a dataset which is currently under update has an invalid state. Parts of the dataset still carry old values whereas other parts might represent the updated values. And what happens if the update fails? For example, blocking access to inconsistent data or automated rollback are two aspects of transactional behavior[24].

[18] `http://projects.spring.io/spring-security/`

[19] `http://redis.io/`

[20] `http://commons.apache.org/proper/commons-jcs/`

[21] `http://ehcache.org/`

[22] `http://memcached.org/`

[23] `https://code.google.com/p/guava-libraries/`

[24] `synchronized` is not transactional.

For that matter we will use Spring's own implementation, the TransactionManager[25].

Additional Packages

In contrast to client software, which in case it crashes can be restarted by a user, a server software is required to be self-governing. MEGANServer should be ideally fault tolerant, self updating and memory consistent. Both Java and Spring ship with libraries which support the criteria stated before. We will take advantage of aspect oriented programming (AOP) [Kiczales et al., 1997] to apply timer functions, exception handling and logging support without adding any line of source code.

The timer manages the caches and keeps the JobQueue free from abandoned requests and triggers a function to frequently scan the database for recently uploaded datasets. For logging we use the generic interface, Simple Logging Facade for Java (SLF4J)[26] primarily because incorporated packages do not use a standardized logging library. Every exception will be caught, logged and processed using the interception framework in Spring's AOP implementation.

5.4 Design

The process in which software packages are systematically plugged in together in such a way that they meet the requirements, in which interfaces are defined and, scopes and the modular structure of the software are discussed, is the design stage.

Figure 5.4 depicts the conceptual design of the MEGANServer environment. A number of MEGAN clients access a web server hosting a MEGANServer instance. Tomcat is used as the runtime environment for MEGANServer. Communication with CouchDB and PostgreSQL is established using additional protocols.

On a level closer to the actual implementation (see Figure 5.5) one can see the modular composition in which the software is designed. There are five programmatically independent areas. `Authentication` is located at the outermost position. The credentials of an incoming request are verified using information stored in the user database. A successful request is routed to the `Dispatcher`. The `Dispatcher` evaluates the request type and determines a suitable handler. Requests, resulting in data of undetermined size, are forwarded to the `Large Data Handler`. Data requests that result in data

[25] http://docs.spring.io/spring/docs/4.0.5.RELEASE/spring-framework-reference/html/transaction.html

[26] http://www.slf4j.org/

sizes transmittable in a single chunk are transferred to the `Small Data Handler`. Requests that affect user entries, such as changing credentials, adding or removing users are transmitted to the `User Data Handler`. Programmatically speaking, each handler is the implementation of a simple interface, using the Spring Framework, translated in an HttpInvoker servlet. All three handlers implement methods to either retrieve or write data. To do so, all methods access the `Data Access Object`. Large data requests are detoured through the `JobQueue` which is capable of breaking large data requests to smaller chunks to speed up transfer without losing streaming behavior. The `Data Access Object` determines the type of data and requests information either from the PostgreSQL or from the CouchDB instance. Similar behavior is implemented in the `UserService` using the UserDB as a data backend. Caches and transaction support are omitted for brevity but belong in the `Data Access Object`

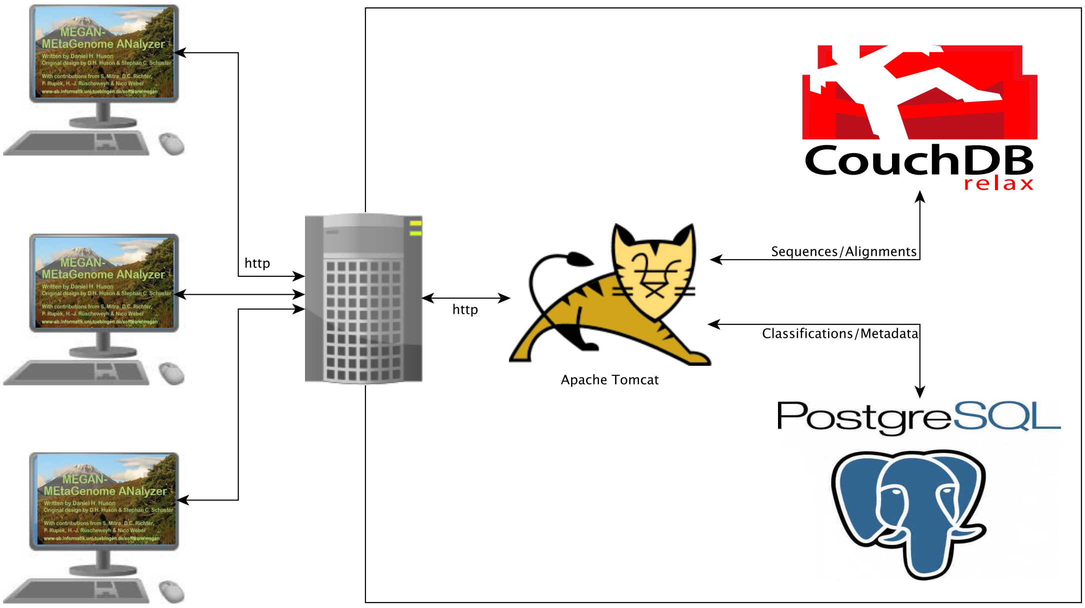


*Figure 5.4: **The MEGANServer Environment**: The MEGANServer software is embedded in the Tomcat servlet engine. Data is stored in CouchDB or PostgreSQL. A number of MEGAN clients communicate with the web server of MEGANServer via http.*

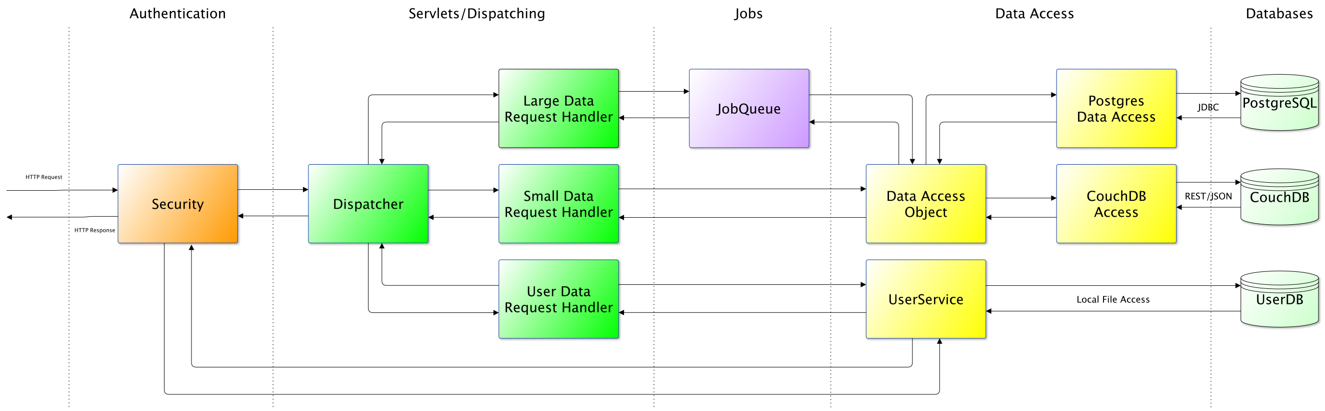


Figure 5.5: ***Design of MEGANServer****:* ***Authentication****: Credentials for incoming connections being validated.* ***Servlets/Dispatching****: The dispatcher parses the request and routes it to the correct servlet. The handlers take advantage of the HttpInvoker interface to unwrap http requests to Java classes and to communicate with the data access object.* ***Jobs****: The JobQueue translates large data requests to smaller chunks and hands out tickets under which data can be accessed.* ***Data Access****: Transactional data access to databases. Caching is implemented for reads matches and header data.* ***Databases****: Raw data access either via JDBC, REST or direct file access.* ***Not Shown****: Authorization, Caching, Exception Handling.*

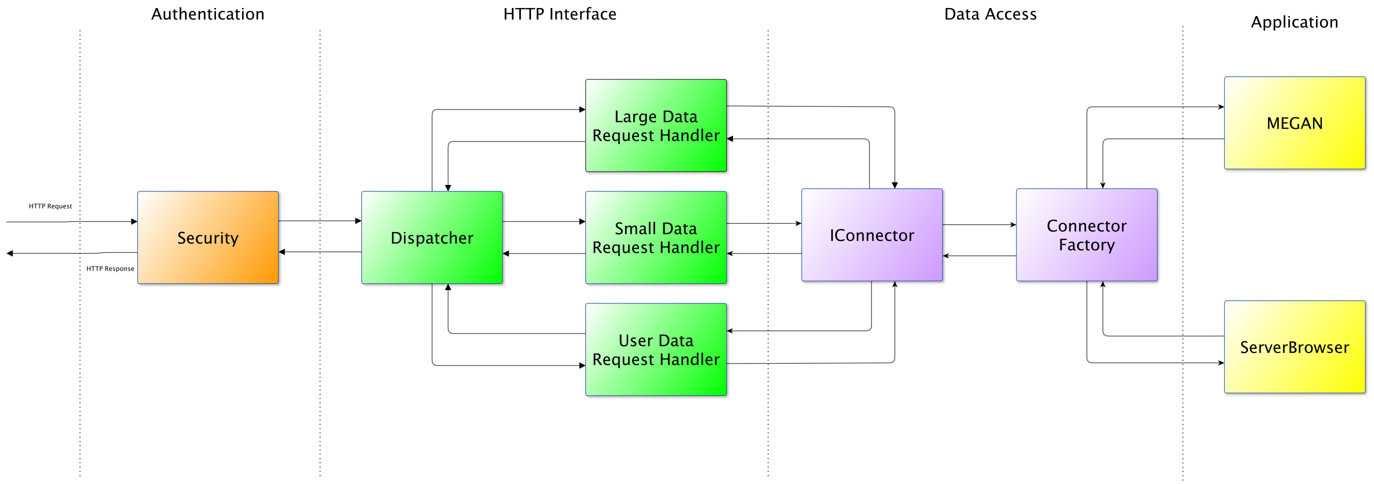


Figure 5.6: ***Design of the MEGAN Add-on***: ***Application***: *Either MEGAN or the ServerBrowser of MEGAN request data from the Connector.* ***Data Access***: *The ConnectorFactory maintains available IConnectors. The IConnector is the universal data interface MEGAN.* ***HTTPInterface***: *The IConnector requests data through the handlers. They are the interfaces of the associated implementations on the server side. The Dispatcher either separates or forks requests.* ***Authentication***: *The http requests are post-processed and enriched with credentials.*

For the communication with MEGANServer we implemented an add-on to MEGAN (see Figure 5.6). In order to access data, `MEGAN` or the `ServerBrowser` use the universal data interface in MEGAN, the `IConnector`. The `ConnectorFactory` manages `IConnector`s for all available MEGANServer instances. The `IConnector` delegates method invocations to one of the three handler interfaces. The `Dispatcher` bundles and translates requests to agree with the http protocol. During `Authentication` the http headers of each outgoing request are enriched with credentials.

5.5 Using MEGANServer

The previous sections aimed at introducing MEGANServer from a technical point of view. Here, we will discuss how the end-user benefits from MEGANServer. First, we will describe how to upload datasets to a MEGANServer instance and evaluate the performance in terms of runtime and storage usage. Secondly, we will introduce the ServerBrowser which is the graphical front-end to the end-user of MEGAN. Through the ServerBrowser one can easily search and access datasets present on a MEGANServer instance. Finally, we will discuss how MEGANServer extends the repertoire of functions of MEGAN, such as pooling of datasets or automated dataset selection using metadata.

To demonstrate features of MEGANServer we decided to use data from a typical metagenomic study. In the study (see [Mackelprang et al., 2011]), three drill core samples from Hess Creek, Alaska were extracted from either a permafrost or the overlaying active soil layer. Samples from two cores were extracted, resulting in four samples, two of each, permafrost and active layer. All samples were incubated at 5°C for seven days. Material was extracted at day zero, two and seven, resulting in twelve samples ready for metagenomic sequencing. Illumina sequencing lead to 250 million reads. 420 million matches were found using the PAUDA aligner. Alignments and reads were imported to MEGAN and written to twelve RMA files.

5.5.1 Uploading

During early stages of MEGANServer we implemented upload functionality within MEGAN. Datasets would be uploaded to MEGANServer, read by read, in the process of creation, and routed through the HttpInvoker web service. However, the overhead in terms of size and time were tremendous. Not surprisingly, creating a dataset locally, copying it to the server on which MEGANServer is hosted and subsequently uploading this dataset to the databases directly, resulted in a significant speedup (100-1,000 fold). Furthermore, this approach is driven by the fact that new aligners, such as MALT, are capable of emitting MEGAN files directly.

The `MSUploader` tool is shipped along with MEGANServer. `MSUploader` takes any of the three MEGAN filetypes as input and moves their data directly to both databases. In Table 5.1 one can see the performance of `MSUploader` for the twelve datasets. The 250 million reads, together with their associated 420 million matches, being copied in roughly 40 hours resulting in a performance of 17 million entries[27] per hour. If reads and matches are not of interest, one can translate a MEGAN file to a summary file, leading to a constant upload time of two seconds.

Sample	Reads	Matches	Size(GB)			Time (hh:mm)	
			RMA	Summary	DB	Summary	DB
1	12,116,336	16,566,839	6.8	0.0001	13.5	00:02	01:31
2	14,733,774	34,957,415	13.0	0.0001	25.9	00:02	03:12
3	35,550,968	71,340,056	27.3	0.0001	54.3	00:02	06:45
4	11,466,987	16,604,831	6.0	0.0001	11.9	00:02	01:30
5	33,687,302	67,614,854	25.9	0.0001	51.5	00:02	06:10
6	15,725,557	14,132,054	6.5	0.0001	12.9	00:02	01:50
7	33,376,178	59,334,732	23.0	0.0001	45.8	00:02	05:43
8	16,419,468	18,705,234	8.0	0.0001	15.9	00:02	02:00
9	15,564,330	21,713,772	8.9	0.0001	17.7	00:02	01:55
10	11,697,059	16,955,173	6.9	0.0001	13.7	00:02	01:36
11	14,026,860	29,380,225	11.1	0.0001	22.1	00:02	02:43
12	32,117,490	52,984,661	21.1	0.0001	41.9	00:02	05:08
all	246,482,309	420,289,846	165.5	0.001	330.1	00:20	39:03

Table 5.1: ***Upload to MEGANServer****: Twelve samples incorporate ∼250 million reads and ∼420 million matches. The size of rma type doubles the size compared with the file structure and takes roughly one hour for 17 million entries (reads+matches). Upload time and space consumption of samples with the summary option is constant and negligible.*

As shown in Table 5.1, storage inside a MEGANServer instance doubles the space requirements compared to an RMA file. The alignment text, as a part of each match, constitutes ∼80-90% of the size. In order to lower disk space required we apply compression. To do so we merge alignment text from all matches of a read and compress

[27] An entry is either a read or a match.

the result. Since, the performance of compression algorithms depend on the input text size[28], larger texts are compressed more efficiently, therefore leading to a smaller data object. For metagenomic datasets, where reads often have no or few matches, our approach to merge alignment texts for each read and subsequently apply compression does not show its true potential. However, if applied to datasets where reads have tens to the hundreds of matches, e.g. in amplicon sequencing datasets, we can show that the size of MEGANServer datasets drops under the size of the initial MEGAN file (see Chapter 8).

5.5.2 ServerBrowser

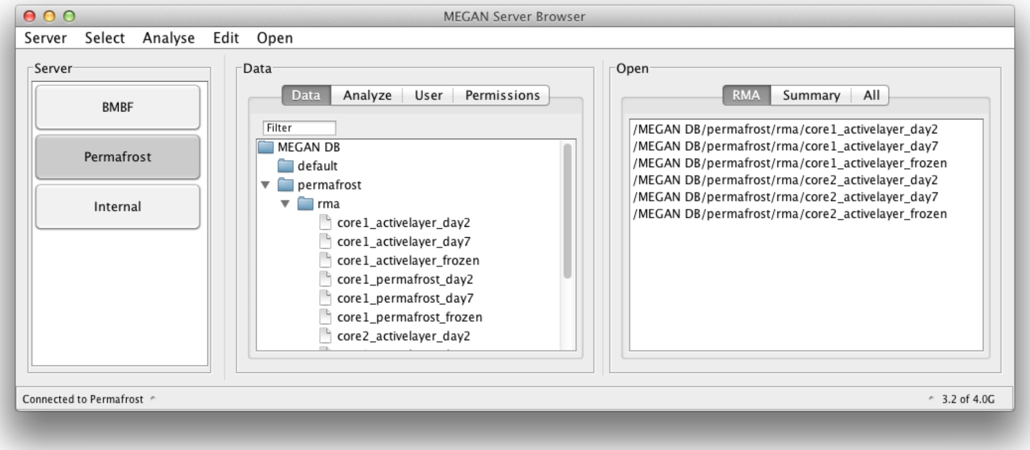
Not only in bioinformatics and many other research fields which depend on software, but also in our daily life, we use programs in order to facilitate workflows that we would not be able to perform without the help of a computer. Among others, two factors seem to be critical when deciding if software is useful to us or not. First, functionality defines which set of functions software can offer and with what performance. Secondly, the usability defines how easy it is to handle and access the functionality of a program. However, these factors often contradict each other. Hence, if a program has a high functionality letting the user decide about a large set of parameters, it is likely that most users will be overstrained by the complexity. As a result researchers will take advantage of simpler tools which are easier to use but are more likely to give results of a lower quality.

MEGAN stands out as a metagenomic software filling the gap between these conflicting factors by offering an easy-to-use and understand graphical interface and also a wide variety of functions. With this in mind, we added a window in MEGAN, facilitating the access to datasets that are hosted on a MEGANServer instance.

The ServerBrowser as depicted in Figure 5.7 is partitioned in three sub-windows. On the left, one can select, edit, delete and add existing MEGANServer instances. The center view shows datasets as found on the MEGANServer instance. Furthermore one can search for datasets with two different filtering functions, either by name or by metadata. Datasets selected in the center window appear on the rightmost view. MEGANServer supports merging, transforming, comparing, grouping or solely opening datasets. Datasets affected by these operations are shown in this part of the window.

A typical MEGAN analysis begins with selecting one or a number of datasets of interest by hand and individually inspecting their content. In a next step, these datasets

[28] There performance also depends on the text structure. But we cannot influence the text structure.



MEGAN DB
- default
- permafrost
  - rma
    - core1_activelayer_day2
    - core1_activelayer_day7
    - core1_activelayer_frozen
    - core1_permafrost_day2
    - core1_permafrost_day7
    - core1_permafrost_frozen
    - core2_activelayer_day2

```
/MEGAN DB/permafrost/rma/core1_activelayer_day2
/MEGAN DB/permafrost/rma/core1_activelayer_day7
/MEGAN DB/permafrost/rma/core1_activelayer_frozen
/MEGAN DB/permafrost/rma/core2_activelayer_day2
/MEGAN DB/permafrost/rma/core2_activelayer_day7
/MEGAN DB/permafrost/rma/core2_activelayer_frozen
```

*Figure 5.7: **The Server Browser**: On the left one chooses, adds or edits MEGAN-Server instances. The view in the center shows the main data representation of MEGAN-Server, a file system like structure. On the right one can see selected datasets which can subsequently be opened in MEGAN.*

are compared to, for example, estimate β-Diversity in between samples. Selection by hand is cumbersome, especially if the number of samples is large or the selection criteria are complex. Also, MEGAN treats each dataset individually. The merging of datasets is not supported, or at least not achievable without reimporting entire alignment files.

With MEGANServer we aim at automating the selection of datasets. Furthermore we extend the functionality by enabling merging of datasets in such a way that MEGAN recognizes the result as an individual dataset. By that we support the idea that analysis of scenarios has a higher chance to reveal information rather than analysis of single datasets. For both undertakings - first, automation of selection and second, merging of datasets - metadata plays a significant role. In order to demonstrate the functionality we will take advantage of the permafrost data that comes along with a small set of metadata (see Table 5.2).

Automated Selection In order to automate selection we take advantage of boolean expressions such as those shown in the Equations 5.2, 5.3 or 5.4. The ServerBrowser parses one expression and results to a list of datasets which meet this criteria (see Figure 5.8).

Sample	Core	Soil	Day
1	1	activelayer	0
2	2	permafrost	2
3	2	permafrost	0
4	1	activelayer	7
5	2	activelayer	2
6	1	permafrost	0
7	2	activelayer	7
8	1	permafrost	7
9	1	permafrost	7
10	1	activelayer	2
11	2	permafrost	7
12	2	activelayer	0

Table 5.2: ***Metadata****: Three types of metadata collected along with the metagenomic samples. 'Core' distinguishes between the different sampling sites whereas the 'Soil' describes at which depth the sample has been collected. 'Day' shows the time elapsed since incubation start of the samples*

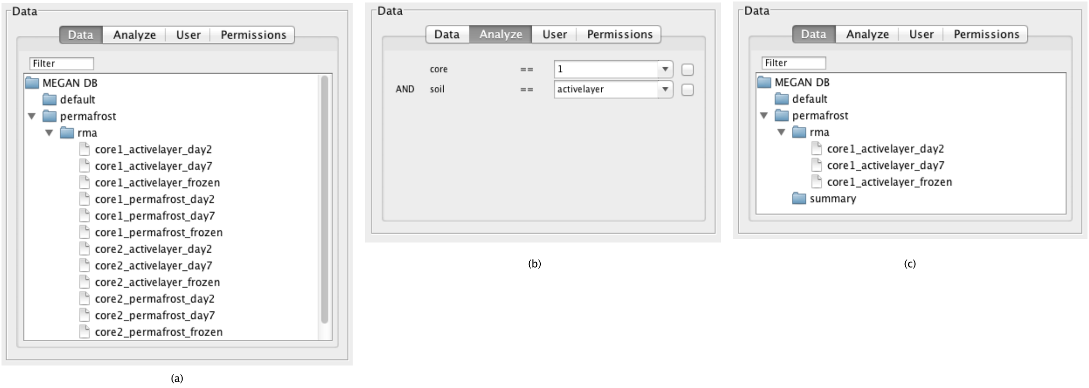
$$\text{`}\mathit{Core}\text{'} = \text{`}\mathit{1}\text{'} \tag{5.2}$$

$$\text{`}\mathit{Core}\text{'} = \text{`}\mathit{1}\text{'}\ \textbf{AND}\ \text{`}\mathit{Soil}\text{'} = \text{`}\mathit{activelayer}\text{'} \tag{5.3}$$

$$\text{`}\mathit{Core}\text{'} = \text{`}\mathit{1}\text{'}\ \textbf{AND}\ \text{`}\mathit{Soil}\text{'} = \text{`}\mathit{permafrost}\text{'} \tag{5.4}$$

Automated Projection After selection, the user can decide how to open datasets of interest. Currently there are five options:

- **Individual:** Open selected datasets individually with one MEGAN window per dataset. When browsing through the data, for example using the Inspector, missing data will be loaded from the MEGANServer instance.
- **Summary:** Extract a summary from each dataset and store individually to the local hard drive. Then open each dataset in MEGAN. These datasets contain no reference to MEGANServer and can later be used in offline mode.
- **Compare:** Perform the **Summary** command and subsequently compare the datasets using MEGAN's inbuilt algorithm which supports three modes, namely


Data
Data | Analyze | User | Permissions
Filter

MEGAN DB
default
permafrost
rma
core1_activelayer_day2
core1_activelayer_day7
core1_activelayer_frozen
core1_permafrost_day2
core1_permafrost_day7
core1_permafrost_frozen
core2_activelayer_day2
core2_activelayer_day7
core2_activelayer_frozen
core2_permafrost_day2
core2_permafrost_day7
core2_permafrost_frozen

(a)

Data
Data | Analyze | User | Permissions

core == 1 □
AND soil == activelayer □

(b)

Data
Data | Analyze | User | Permissions
Filter

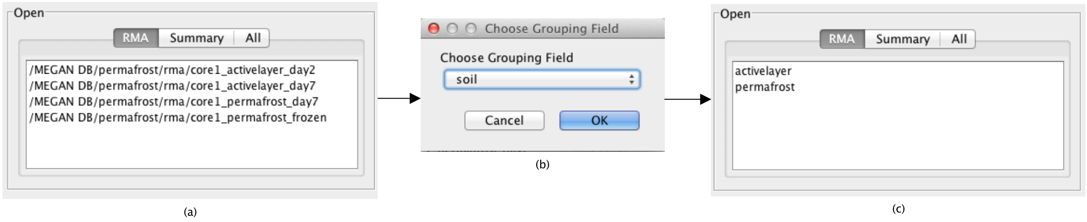
MEGAN DB
default
permafrost
rma
core1_activelayer_day2
core1_activelayer_day7
core1_activelayer_frozen
summary

(c)

*Figure 5.8: **Metadata usage**: (a) shows all twelve datasets present in the MEGANServer instance. In (b) one applies a boolean expression on the metadata, selecting only datasets of interest. As a result only datasets that satisfy the criteria stated in the boolean expression are visible (c).*

absolute, normalized and sub-sampled.

- **Merge:** Selected datasets are merged in such a way that MEGAN recognizes them as a single dataset. To do so MEGANServer creates an artificial dataset on the fly at no additional memory or cpu costs. To MEGAN it is a normal dataset of the RMA type, supporting inspection of reads and matches or visualizing alignments.
- **Group:** Selected datasets are grouped based on metadata (see Figure 5.9). The resulting groups can then be opened with either of the previous commands.



/MEGAN DB/permafrost/rma/core1_activelayer_day2
/MEGAN DB/permafrost/rma/core1_activelayer_day7
/MEGAN DB/permafrost/rma/core1_permafrost_day7
/MEGAN DB/permafrost/rma/core1_permafrost_frozen

(a)

Choose Grouping Field: soil

(b)

activelayer
permafrost

(c)

*Figure 5.9: **Grouping**: (a) Previously selected datasets. (b) Group datasets based on metadata. (c) The view containing initial datasets has been replaced with a view containing the merged datasets.*

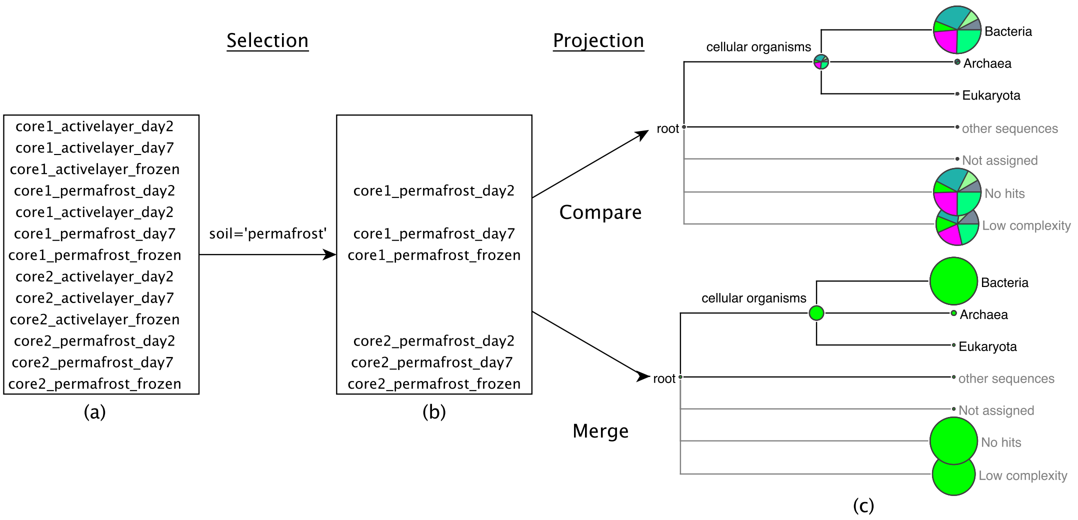


*Figure 5.10: **Selection - Projection:** (a) All datasets present on a MEGANServer instance. (b) Use a boolean expression to choose datasets of interest. (c) Open the selection in different ways.*

5.6 Big Data - Challenging Software Design

The goal was and is to implement a data repository that not only offers access as fast as the file system and implements multi-user capabilities but also is space efficient. These are ambitious goals, yet, when implementing software handling small sized data, the complexity overhead is not visible to the end-user. Whether an operation takes 0.5 seconds to perform due to introduction of routines that guarantee transactional behavior or it takes 0.1 seconds otherwise, will remain unnoticed by the user. On the other hand, if one operation usually takes 5 minutes and with a server version requires 25 minutes, the user might stick to the version using local access. However, the implementation of routines that guarantee data validity at all times is essential for a multi-user environment. Unfortunately, this, in combination with the fact that MEGAN deals with datasets of arbitrary sizes, adds up to a significant performance drain.

For the developer this means that MEGANServer must be designed in such way that the implementation compensates for the loss of performance. To do so we experimented with a number of implementations and database backends. The final solution in which we introduced a blocked streaming behavior in combination with extensive caching seems to offer the best performance in terms of memory use and cpu cost. Tests prove that the retrieval of data via MEGANServer can compete and even outperform the traditional approach when neglecting biases introduced by, for example, a slow internet connection.

5.7 Conclusion

In this chapter we introduced MEGANServer, a software with which one can store MEGAN files on a web server without losing the comfort of using MEGAN. Datasets are stored using modern database technologies offering striking performance and being fault proof. Access to MEGANServer is granted via MEGAN only and secured using basic authentication and authorization schemes, thereby, preventing illegal access. Besides providing a powerful storage for datasets, we were also able to extend the function pool. To do so, we picked up ideas that originate from 16S rRNA analysis. There, large numbers of datasets are not stored individually but in one file and metadata is used to divide or merge datasets of interest. Therefore, we implemented methods that allow metadata guided merging of datasets to overcome the limitation of comparing single datasets to support the goal of comparing scenarios such as *healthy* vs. *sick*.

Part III

MEGAN for Targeted Sequencing

This part focusses on the development of a 16S rRNA sequence analysis pipeline which, in contrast to the typical pipeline, employs alignment in combination with the metagenomics analysis tool MEGAN and a new taxonomic placement algorithm, namely Majority Vote LCA. We will show that this approach assigns more than 99% of sequences to the correct genus and compare β-Diversity plots generated in MEGAN with plots created by typical 16S rRNA analysis pipelines, such as mothur or QIIME.

Chapter 6

16S rRNA Pre-processing

The rapid development of sequencing technologies, reflected in plummeting prices and growing number of emitted sequences has lead to a large number of projects that use DNA or RNA sequences to answer a diverse range of questions. Despite the variety of technologies, what all next (and third) generation sequencers have in common is the production of erroneous and short sequences. Therefore, when it comes to sequence length and quality, Sanger is still considered to be the gold standard [Technologies, 2014]. With Sanger sequencing one can emit high quality sequences with a length up to 1,000 base pairs. The number of emitted sequences is, however, small compared to the numbers produced by next generation sequencing technologies such as 454 or Illumina. In general, one can say that sequencing is a tradeoff between price, length, accuracy and emitted volume.

For the projects in which we have taken part, the main focus is on the analysis of the prokaryotic small subunit of ribosomal RNA (short 16S rRNA). We decided to use Roche's 454 sequencer. This technology offers a higher quality and length compared to the typical sequencing technology for metagenomic samples, namely Illumina, but emits a smaller number of sequences for a higher price. However, the main goal when applying targeted sequencing is to unravel the "*true*" taxonomical diversity [Hughes et al., 2001] within a sample and for that task, a smaller number of sequences is generally considered to be sufficient:

> *The advantages of having large numbers of samples at shallow coverage (∼1,000 sequence per sample) clearly outweigh having a small number of samples at greater coverage for many datasets, suggesting that the focus for*

> *future studies should be on broader sampling that can reveal association with key biological parameters rather than on deeper sequencing.*
> [Kuczynski et al., 2010]

Furthermore, greater sequence length is favorable in order to distinguish different species based on sequence similarity. In fact, over the past years and even though the technology was initially developed to facilitate assembly [Rothberg and Leamon, 2008], 454 was the quasi standard for sequencing of the 16S rRNA gene[1]. This is underlined by the fact that major analysis pipelines, such as QIIME or mothur, were exclusively developed for 454 data.

Currently, 454 Life Sciences offers two sequencers, the GS Junior System and the GS FLX+ System; of these two the latter is advertised to be capable of emitting "Sanger-like read length" [Roche Diagnostics GmbH, 2014] if the newest Titanium reagents are used. However, even though the new flow cycle produces longer reads, it does not change the fact that the data is still erroneous. Assembly can, as the research field for which 454 was initially developed, due to increased coverage, deal with sequencing errors very well. However, for 16S rRNA analysis, sequence errors introduce a large variety of problems. For example, clustering, a step in which the diversity of a sample is estimated by bundling similar sequences in so-called OTUs or Phylotypes, e.g at 97% to achieve species level resolution, is largely affected by sequence errors, resulting in an estimation of a wrong and a much higher number of species, which will ultimately influence the outcome of the downstream analysis. As a consequence, the impact of sequencing errors led to a debate about if or how errors are responsible for species assigned to the so-called "rare biosphere" [Kunin et al., 2010; Sogin et al., 2006; Huse et al., 2007, 2010; Lynch et al., 2012].

In general, the Phred Score [Ewing and Green, 1998; Ewing et al., 1998] is used to describe sequence quality, which estimates the correctness of a base-call by a probability. As shown in Table 6.1, a score of 10 estimates that the assigned base has a 90% chance to be correct, or that the chances of the assigned base to be incorrectly introduced are 1 in 10. In this manner, a stretch of 100 bases incorporates 10 false base-calls.

For the 454 technology it is known that sequence quality is negatively correlated to proceeding sequence length, introduced by a lack of synchrony among incorporation of nucleotides to an amplified template sequence as described in Chapter 2. Having this evidence as a rationale, numerous quality control tools [Schmieder and Edwards, 2011;

[1] 454 is currently losing its leading role to Illumina and PacBio.

Phred Score	Base-call accuracy
10	90%
20	99%
30	99.9%
40	99.99%

Table 6.1: Phred Score as the probability if a base is correct.

Patel and Jain, 2012; Gordon and Hannon, 2010; Andrews, 2010] were developed, all promising to clear the input sequencing data of erroneous sequences or/and to cut off the low quality tail stretch of the input sequencing data. Naive approaches consist of arbitrarily picking one quality score, parsing every sequence from beginning to end and, once one base under the threshold is found, it is assumed that the rest of the sequence is of low quality and is disposed off.

This action results in short sequences, but more importantly, it ignores the fact that the low quality base could have been an outlier followed by a number of higher quality bases. Because of the shortcomings of this approach a new one was developed. The principle behind the sliding windows concept is: if the average score inside a sequence window drops under a threshold, the sequence is cut at the beginning of the sliding window, resulting in longer sequences and solving the outlier problematic. An alternative to the sliding window approach was introduced by Robert Edgar who argues that average quality scores are not a good indicator to distinguish correct base-calls from incorrect ones [Edgar, 2014].

However, the approaches mentioned above are based on the assumption that quality scores reasonably reflect the correctness of a base-call. A more advanced error-correction approach circumvents Phred Scores and evaluates the underlying flowgrams directly. During sequencing at each flow, one of the four nucleotides is added to the sequencing plate. If the nucleotide can be incorporated, light is emitted and optics measure the intensity. If more than one nucleotide is incorporated light of a higher intensity is emitted. Ideally, as seen in Table 6.2a the values are well distinguishable (resulting sequence: `>Seq1:CTTG`). However, the real output data contains floats rather than integers leading to a more complex base-calling process. The table with the idealized values translates Sequence 2 to `ACCTTT` whereas a naive translation of the realistic values leads to a sequence ACCTT[2]. The standard base-call protocol performs base-calls naively and quality values represent the deviation from the closest integer[3]. In order to optimize the base-calling process by applying advanced statical methods, Christopher Quince developed tools such as PyroNoise and AmpliconNoise [Quince et al., 2011] that use the expectation maximization algorithm leading to statistically more probable base-calls.

	A	C	T	G
>Seq1	0	1	2	1
>Seq2	1	2	3	0

(a) Idealized flow

	A	C	T	G
>Seq1	0.2	1.3	1.7	0.6
>Seq2	0.65	1.6	2.4	0.3

(b) Realistic flow

Table 6.2: Idealized and realistic flows resulting from 454 sequencing

Application of either of these tools drastically improves sequence quality by removing three of the four potential error sources listed below:

1. PCR applied before sequencing introduces substitution errors [Cline et al., 1996].
2. Platform specific errors. 454 sequencers struggles with longer stretches of homopolymers [Margulies et al., 2005].
3. Asynchrony of polymerase positions among an amplified template sequence.
4. 16S rRNA sample preparation is prone to chimeras formation [Haas et al., 2011].

Since chimeric sequences are not a product of poor sequencing performance or low quality but are introduced at a step prior to sequencing, they remain undetectable by either of the methods previously introduced. Chimeras are sequences that originate from two or more different sequences and are usually introduced during PCR as a result of an incomplete extension [Smyth et al., 2010]. The removal of chimeric sequences is challenging and no method has been developed yet that successfully removes all chimeric sequences from a sample. However, there are a number of tools that focus on the removal of chimeric sequences such as ChimeraSlayer [Haas et al., 2011], Bellerophon [Huber et al., 2004] or Decipher [Wright et al., 2012]. The UCHIME [Edgar et al., 2011; Edgar, 2013] algorithm embedded the major 16S rRNA analysis pipelines QIIME [Caporaso

[2] Or, if it translates to the correct sequence, than low Phred Scores are assigned to the critical areas.

[3] The explanation is a vast oversimplification.

et al., 2010b] and mothur [Schloss et al., 2009] argues that it outperforms all other tools in terms of sensitivity and speed. In order to detect chimeric sequences, UCHIME cuts reads in shorter sub-sequences and aligns these against a reference database. If the resulting alignments are assigned to different taxa, the read is considered to be an offspring of a chimeric formation.

In order to prepare sequences for downstream analysis and to remove as many errors as possible without losing too much of sequence length, we decided to follow the guideline of the mothur 454 pre-processing SOP and apply adaptions where required. The pipeline is based on the evidence found in a study [Schloss et al., 2011] that performed an in-detail error analysis:

- **Input Data:** Two files serve as input for the mothur pipeline. First, the standard flowgram format (SFF) file and a file with barcodes and names of samples incorporated in a sequencing run.
- **SFF Extraction:** The binary SFF file is translated to a human readable flow, fasta and quality file. Reads are demultiplexed and assigned to their associated sample.
- **Trimming:** Schloss et al. [2011] showed that sequences with more than two errors in the primer sequence and/or more than one error in the barcode have an increased potential to be erroneous. These sequences are removed together with sequences that are suspiciously long or short.
- **Denoising:** Application of the PyroNoise algorithm.
- **Dereplicating:** Identical sequences are merged. The result is equal to a clustering at 100%. This step is performed to lower computation time of forthcoming steps.
- **Align:** Sequences are aligned against the Silva reference database[4]. The result is used to remove sequences that match regions other than the sequenced 16S rRNA region.
- **Pre-cluster:** The dereplication step combines sequences that are identical and count their occurrences. Normally there are some sequences with higher abundance and a lot of singletons. The sequences with a higher abundance are considered to have a higher probability to be error free. Pre-clustering takes the

[4] A small subset of the actual Silva database formatted as a multiple sequence alignment.

singletons and compares these against the more abundant sequences at a user defined identity, for example at 99% identity, and, if successful, add the singletons to the bigger cluster.

- **Chimeras:** Chimeric sequences are removed using the UCHIME algorithm.
- **Contaminants:** Sequences are taxonomically assigned using the RDP classifier. Sequences that fail assignment to the kingdom level are considered to be contaminants and are removed.

For further analysis we do not use the mothur package. A self written script reformats the mothur output and emits one fasta file per sample.

Chapter 7

16S rRNA Analysis using MEGAN

7.1 Introduction

The ultimate goal when performing analysis on sequences that originate from the 16S rRNA gene is to identify the "*true*" taxonomic identity for each input sequence and, subsequently use this information to apply additional methods that, for example, calculate distributions among taxa, and α- or β-Diversity. Taxonomy-dependent approaches, for example, alignment, are generally considered to be the inferior choice for this task, since their performance and robustness is tightly correlated to the associated reference database and taxonomy. Therefore a taxonomy-dependent approach will struggle with assigning taxa to sequences which are not well represented or have no close relatives in the reference database [Armougom and Raoult, 2009; Schloss and Westcott, 2011; Huse et al., 2008]. Consequently, the typical 16S rRNA sequence analysis toolkits such as mothur or QIIME employ a different reference database independent approach (taxonomy-independent), namely clustering. In clustering the similarity among all input sequences is calculated and used to assign sequences to so-called operational taxonomic units (OTU). Thus, sequences from yet uncultured or unannotated microbes are captured by this approach as well. The optimal outcome of clustering is to create one OTU per species present in a sample with all sequences assigned that are likely to belong to this species. That is why clustering is normally performed at 97% similarity[1].

[1] 97% identity is the general rule of thumb to distinguish between species when comparing two sequences. Even though this rule is widely used also for shorter sequences, it was originally

In order to assign a taxonomic identity, one representative sequence of each OTU is selected[2] and taxonomy-dependent approaches such as alignment are applied. If the number of OTUs is reasonably small, this approach implies a major performance boost. However, the speed-up is impacted by the loss in resolution since the taxonomic identity of a representative sequence is carried out to all sequences assigned to the specific OTU.

A third aspect, why clustering is considered to be a superior approach when compared to taxonomy-dependent methods is based on the fact that distance based algorithms such as β-Diversity, will lead to better results when applied on a tree derived from a multiple sequence alignment of all representative sequences, hence, displaying the exact distances among the sample, rather than on a generic tree such as the NCBI or Silva taxonomy where sequences are assigned in the process of taxonomic binning. However, not all β-Diversity metrics take tree distances as the base for their calculations, and taxonomies such as Silva are known to be of high quality [Quast et al., 2013; Yilmaz et al., 2013; Schloss, 2009].

Independent of the approach, the ultimate goal of 16S rRNA analysis is to determine the correct taxon for each input sequence. In this chapter we will disprove the general concerns regarding taxonomic-dependent methods and introduce an approach which is capable of first, assigning more than 99% of the reads to the correct genus; second, lowering the false positive rate to a value close to 0%; third, showing that alignment is not a performance bottleneck; and finally, showing that as a consequence of our taxonomic placement we can create β-Diversity plots comparable to those created by taxonomy-independent approaches. To do so we apply MALT as aligner in combination with a new taxonomic placement method, namely Majority Vote which has been recently introduced to MEGAN.

To test the robustness of our approach we analyzed the 16s rRNA gene sequences of mouse gut flora. The experimental setting consisted of gnotobiotic mice fed with a defined set of bacteria. The 454 sequences from the V3-V6 region were retrieved from 28 samples consisting of feces, cecum or small intestine tissue biopsies.

defined to be true only for full length 16S rRNA sequences.

[2] There are more than a dozen techniques to determine which sequence is representative. For further information see: `http://qiime.org/scripts/pick_rep_set.html`

7.2 Material & Methods

7.2.1 Input Data & Databases

454 Sequencing Data

Twenty-eight samples were obtained from gnotobiotic mice which were initially colonized with four bacterial strains and after the first sampling were fed with 10 additional bacterial types. Sampling was performed at three time points, day 0, 10 and 20 and three sites, feces, cecum and small intestine, as depicted in Table 7.1.

Day	0	10	20
Feces	4	4	4
Cecum	0	3	4
Small Intestine	0	5	4

Table 7.1: Sampling Dates/Body Sites

Sequencing of the V3-V6 region of the 16S rRNA gene was performed on a 454 GS FLX sequencer at Eurofins. Following the pre-processing pipeline, described in Chapter 6, 28 fasta files, containing high quality reads (low quality reads filtered and chimeric sequences removed) were retrieved. A total number of 349,639 reads with sequence length ranging between 250 and 290 bp were further analyzed.

Reference Database

The reference database contains full-length 16S rRNA sequences from the twelve[3] bacterial strains present in the mice as described above. Their identifiers are: ASF361, ASF457, ASF519, Isol46, Isol48, KB1, YL2, YL31, YL32, YL44, YL45, YL58. The sequences were obtained by Sanger sequencing.

Silva Database

Version 115 of the 16S rRNA Silva database was used. The database contains 479,726 quality filtered, full-length 16S rRNA sequences clustered at 99% identity and can

[3] Two bacterial strains did not colonize and were not found in any of the 28 samples. For brevity we removed these references from downstream analysis.

be downloaded, along with the taxonomy files, from: `http://www.arb-silva.de/no_cache/download/archive/release_115/Exports/`.

For further information on the Material & Methods Section we provide a flowchart depicting every step performed in this section in Figure C.1.

7.2.2 Taxonomical Classification of the Full-Length Reference Sequences

The sequences in the reference database were aligned against the Silva database, using usearch [Edgar, 2010] and MALT [Huson, 2014a] at different percent identities. The usearch run was performed in the *search_global* mode resulting in a full alignment[4] with a query coverage of 100% and percent identities ranging from 100-96%. MALT alignment was performed in *semi-global* mode, using the *full-seed* approach for percent identities ranging between 100-96%. For every percent identity we counted the number of matches per genus for each input sequence, once for MALT and once for usearch. The resulting table, showing a summary of MALT alignments at genus level, is shown in Table 7.2. Since MALT and usearch mostly agreed on which matches have been found, we skip the usearch results for brevity (They can be found in the Appendix). In order to eliminate the possibility of the Silva database to be biased we additionally assigned the reference sequences using the RDP classifier, as shown in Table 7.3.

For further information one can find the exact number of matches assigned by MALT and usearch in the Tables C.1, C.2, C.3, C.4, C.5, C.6 and C.7 which can be found in the appendix.

7.2.3 Inferring the Correct Taxonomic Distribution

Sequences of the 28 pre-processed samples were merged to a single fasta file and compared against the reference database. As aligners MALT and usearch were used. The MALT run was performed in *semi-global* mode with identities ranging between 100% and 90%. Usearch was used in the *search_global* mode with a query coverage of 98% with identities ranging between 100% and 90%. The reason why we used usearch with a query coverage

[4] Each query sequence is aligned against each reference sequence. Results are sorted by percent identity.

Name	Taxon	100%	99%	98%	97%	96%
ASF361	Lactobacillus	100.0	100.0	100.0	100.0	100.0
ASF457	Mucispirillum	100.0	100.0	100.0	100.0	100.0
ASF519	Parabacteroides	0	100.0	100.0	100.0	100.0
ISOL46	Erysipelotrichaceae;IncertaeSedis	0	100.0	100.0	100.0	100.0
ISOL48	Bacteroides	0	100.0	100.0	100.0	100.0
KB1	Enterococcus	0	100.0	100.0	100.0	98.8
	Staphylococcus	0	0	0	0	0.3
	Melissococcus	0	0	0	0	0.3
	Bacillus	0	0	0	0	0.3
	Carnobacterium	0	0	0	0	0.3
YL2	Bifidobacterium	0	100.0	100.0	100	100
YL31	Flavonifractor	0	100.0	100.0	100.0	89.2
	Pseudoflavonifractor	0	0	0	0	10.8
YL32	Lachnospiraceae;IncertaeSedis	0	100.0	100.0	100.0	99.4
	Lachnospiraceae;uncultured	0	0	0	0	0.6
YL44	Akkermansia	0	100.0	100.0	100.0	100.0
YL45	Parasutterella	0	100.0	100.0	100.0	100.0
YL58	Blautia	0	0	0	0	0

Table 7.2: Percentage of matches assigned to each identified genus after alignment of full-length reference sequences against Silva using MALT at different percent identities (100-96). The color indicates the relationship of the genus to the most abundant genus of each reference sequence. Yellow indicates that the genus does not agree on genus level but on family level. Orange indicates that their taxonomic paths agree either on class or on order level but disagree on lower levels.

lower than 100%, is due to the fact that usearch calculates a global alignment, and subsequently cuts off terminal gaps, thus some alignments fail the 100% query coverage condition.

If sequences aligned against more than one reference, we considered the match with a higher percent identity to be the correct match. Table 7.4 shows the distribution among reference sequences for both tools. Sequences which could not be aligned were assigned to the *No_Hit* column. Figure 7.1 depicts the distribution resulting from a MALT alignment at 97% identity as a graph.

For further information, see the tables in the appendix. Table C.13 shows the number

		Confidence	
Name	Taxon	Full-Length	Trimmed
ASF361	Lactobacillus	1.00	0.98
ASF457	Mucispirillum	1.00	1.00
ASF519	Parabacteroides	1.00	1.00
Isol46	Erysipelotrichaceae;IncertaeSedis	1.00	0.99
Isol48	Bacteroides	1.00	1.00
KB1	Enterococcus	1.00	1.00
YL2	Bifidobacterium	1.00	1.00
YL31	Flavonifractor	1.00	1.00
YL32	Clostridium XlVa	1.00	1.00
YL44	Akkermansia	1.00	1.00
YL45	Parasutterella	1.00	0.91
YL58	Blautia	1.00	1.00

Table 7.3: Full-length and trimmed reference sequences taxonomically assigned by the RDP classifier.

%ID	Tool	ASF361	ASF457	ASF519	Isol46	Isol48	KB1	YL2	YL31	YL32	YL44	YL45	YL58	No_Hit
100	MALT	16	0	0	104	28	85	0	86	271	0	58	0	348,991
	USEARCH	16	0	0	104	28	85	0	86	271	0	58	0	348,991
99	MALT	8,447	30	119,671	417	43,126	104	21	238	35,389	120,520	8,324	643	12,709
	USEARCH	8,444	30	119,650	417	43,104	104	21	237	35,336	120,520	8,319	643	12,814
98	MALT	8,630	37	122,255	424	44,189	110	21	239	36,135	124,333	8,852	680	3,734
	USEARCH	8,626	37	122,165	424	44,126	110	21	239	36,069	124,240	8,840	680	4,062
97	MALT	8,634	37	122,495	424	44,252	110	21	239	36,186	124,655	8,893	682	3,011
	USEARCH	8,633	37	122,512	424	44,264	110	21	239	36,168	124,631	8,887	682	3,031
96	MALT	8,637	37	122,523	424	44,263	110	21	239	36,216	124,693	8,895	682	2,899
	USEARCH	8,636	37	122,624	424	44,291	110	21	239	36,214	124,697	8,896	684	2,766
95	MALT	8,637	37	122,555	424	44,293	110	21	239	36,232	124,693	8,895	682	2,821
	USEARCH	8,637	37	122,697	424	44,326	110	21	239	36,238	124,704	8,897	684	2,625
94	MALT	8,637	37	122,571	424	44,337	110	21	439	36,251	124,695	8,897	682	2,538
	USEARCH	8,637	37	122,728	424	44,375	110	21	439	36,259	124,708	8,899	684	2,318
93	MALT	8,639	37	122,581	425	44,343	110	21	444	36,253	124,696	8,899	682	2,509
	USEARCH	8,639	37	122,745	425	44,385	110	21	444	36,263	124,708	8,900	684	2,278
92	MALT	8,639	37	122,584	425	44,345	110	21	444	36,259	124,696	8,902	682	2,495
	USEARCH	8,639	37	122,746	425	44,385	110	21	444	36,269	124,709	8,902	684	2,268
91	MALT	8,641	38	122,592	425	44,348	110	21	444	36,261	124,697	8,904	683	2,475
	USEARCH	8,641	38	122,753	425	44,388	110	21	444	36,271	124,709	8,905	684	2,250
90	MALT	8,641	60	122,596	425	44,351	110	21	444	36,262	124,697	8,904	688	2,440
	USEARCH	8,641	60	122,755	425	44,388	110	21	444	36,273	124,709	8,905	684	2,224

Table 7.4: Distribution among reference sequences using MALT and usearch on the input sequences at different percent identities (100-90).

of reads that aligned or failed to align at different percent identities using the MALT software.

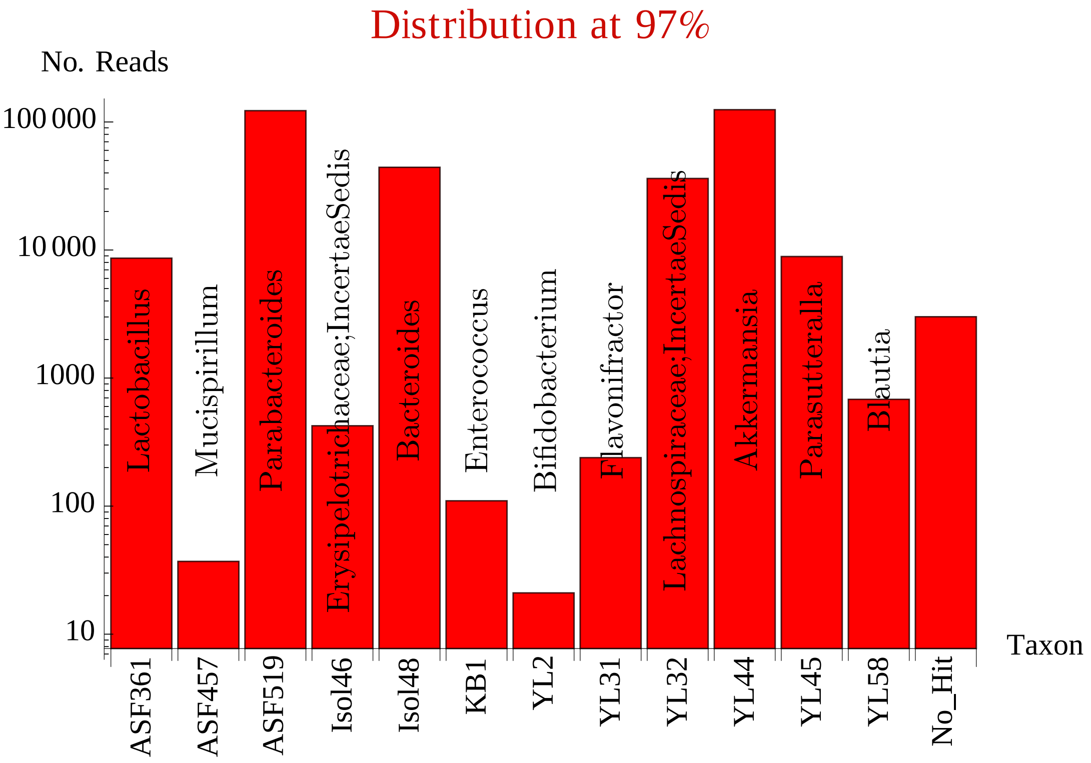


Figure 7.1: Distribution of reads among reference sequences using MALT at 97% identity on a logarithmic scale in combination with their proposed taxonomic identity.

7.2.4 Taxonomical Classification of the Trimmed Reference Database

We determined the average alignment start position and alignment length for each of the reference sequences by using results generated with MALT at 97% identity in the previous section (see Figure 7.1). Then, these stretches were extracted from the reference sequences in order to mimic typical but error-free reads.

These sequences were aligned against Silva and analyzed using the same protocol as described in Section 7.2.2. The summarized results are shown in Table 7.5. In order to eliminate the possibility of the Silva database to be biased we additionally assigned the reference sequences using the RDP classifier, as shown in Table 7.3.

For further information and the exact number of matches found by MALT and usearch see Tables C.8, C.9, C.10, C.11 and C.12.

Name	Taxon	100%	99%	98%
ASF361	Lactobacillus	100	97.9	98.1
	🟧 Streptococcus	0	1.3	0.6
	🟧 Allobaculum	0	0.8	0.3
ASF457	Mucispirillum	100	100	100
ASF519	Parabacteroides	100	96.2	96.5
	🟥 Lachnospiraceae;IncertaeSedis	0	3.8	3.5
ISOL46	Erysipelotrichaceae;IncertaeSedis	100	100	100
ISOL48	Bacteroides	100	100	100
KB1	Enterococcus	95.6	94.2	94.3
	🟧 Planomicrobium	2.2	2.3	2.3
	🟧 Staphylococcus	0.7	1.7	1.4
	🟧 Clostridium	0	0.5	0.5
	🟧 Epulopiscium	0	0	0.5
	🟧 Bacillus	1.4	1	1
YL2	Bifidobacterium	100	100	100
YL31	Flavonifractor	96.7	92.9	87
	🟨 Ruminococcaceae;IncertaeSedis	3.3	3.6	1.7
	🟨 Ruminococcaceae;uncultured	0	3.6	11.3
YL32	Lachnospiraceae;IncertaeSedis	97.2	96	81.5
	🟨 Lachnospiraceae;uncultured	0.9	1.5	17.2
	🟥 S24-7	0.9	1	0.6
	🟥 Anaerolineaceae;uncultured	0.4	0.3	0.2
	🟨 Roseburia	0	1	0.2
	🟧 Pseudobutyrivibrio	0	0	0.2
	🟧 Ruminococcus	0	0	0.2
	🟧 Anaerosporobacter	0	0	0.2
	🟨 Blautia	0.4	0.3	0.2
YL44	Akkermansia	100	100	100
YL45	Parasutterella	100	100	100
YL58	Blautia	0	93.1	88.2
	🟧 Christensenellaceae;uncultured	0	0.3	0.3
	🟨 Roseburia	0	0.3	0.3
	🟨 Dorea	0	0.3	0.3
	🟨 Lachnospiraceae;IncertaeSedis	0	1.9	2.8
	🟨 Pseudobutyrivibrio	0	2.4	2.3
	🟧 Ruminococcaceae;IncertaeSedis	0	0	0.3
	🟧 Peptostreptococcaceae;IncertaeSedis	0	0.3	0.3
	🟨 Lachnospiraceae;uncultured	0	0	5
	🟥 S24-7	0	0.3	0.3
	🟧 Ruminococcus	0	0.3	0.3

Table 7.5: Percentage of matches assigned to each identified genus after alignment of trimmed reference sequences against Silva using MALT at different percent identities (100-98). The color indicates the relationship of the genus to the most abundant genus of each reference sequence. Yellow indicates that genus does not agree on genus level but on family level. Orange indicates that they agree either on class or on order level but disagree on lower levels. Red indicates that this genus belongs to a different phylum.

7.2.5 Taxonomic Identity of Reference Sequences

The results retrieved in the Sections 7.2.2 and 7.2.4 were used to derive the taxonomic identity (at genus level) for each of the 12 bacterial strains present in the reference database. To do so, we checked results of full-length and trimmed reference alignments against Silva and picked for each of the 12 bacterial strains the genus which lead us to the highest number of matches (see Tables 7.5 and 7.2). Usearch and MALT agreed at all percent identities tested, so that each reference sequence could be assigned to a genus. We cross-validated the selected taxonomic identities by comparing these against the RDP generated taxonomic identities for trimmed and full-length reference sequences. The resulting taxonomic identity for each reference sequence is shown in Figure 7.1.

7.2.6 Alignment of Sequencing Data against Silva

We extracted the sequences that successfully aligned against the reference database at percent identities ranging between 100 and 96% (see Table 7.4), for example, 648 sequences resulted by aligning at 100% identity against the reference, and aligned these against the Silva database using MALT in *semi-global* mode, and percent identities ranging between 100 and 96%. The resulting 25 alignment files were imported to MEGAN using the SILVA taxonomy file and a mapping file generated by a self-written script. For taxonomic classification we applied MEGAN's Lowest Common Ancestor algorithm (LCA) with a minimal bitscore of 300 and set the toppercent parameter to 2%. We imported the alignment files a second time and this time applied the Majority Vote LCA algorithm at a 90% confidence for taxonomic placement.

We tried to perform the same task with usearch in *usearch_global* mode as proposed by the usearch developers, which uses unique k-mers as runtime reducing heuristic. The alignment failed due to an *out-of-memory* exception, since only the 32-bit version is free for academic use. As an alternative, we tried to perform the analysis using usearch in *search_global* mode, executing a full alignment, leading to unacceptable runtimes. We therefore removed usearch from our analysis pipeline.

7.2.7 Alignment of Dereplicated Sequences

In order to lower the computation time we performed the MALT alignment a second time, but this time we dereplicated the input sequences beforehand, thereby reducing the sample size and alignment time ∼40-fold. Replication of resulting alignments after completion lead to the exact same result as discussed above.

7.2.8 Comparing LCA against Majority Vote LCA

For each read we extracted the taxonomic path, from the MEGAN files, generated in Section 7.2.6, once for the normal LCA and once for the Majority Vote LCA algorithm at different percent identities and compared the paths against the *true* taxonomic distribution results (see Table 7.4).

Based on the outcome of the test, taxonomic placement of a read is categorized as follows: “Same” is designated to those reads which were successfully placed in the correct genus; “Higher” is assigned to the reads that were mapped to a higher taxonomic level, nevertheless they share a common path with the correct genus; “No Hit” describes all reads that, even though they could be aligned against the reference database, could not be aligned against the Silva database or led to low quality matches; Reads which did not fall in any of the previous categories are false positives and are labeled as “Different”. Examples for each category are listed in Table 7.6.

Summarized results of the genus level comparison are shown in Table 7.7.

Class	Taxonomic Path
Reference	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Blautia
Same	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Blautia
Higher	Firmicutes;Clostridia
Different	Bacteroidetes;Bacteroidia;Bacteroidales;S24-7
No Hit	-

Table 7.6: Example for correct or incorrect taxonomic assignment.

7.2.9 β-Diversity using QIIME and MEGAN

From each of the 28 samples, we removed sequences that did not align at 97% identity against the reference database, and analyzed the remaining data using QIIME in de-novo clustering mode and MALT, using 97% identity in combination with the Silva database, MEGAN and the Majority Vote algorithm for taxonomic placement.

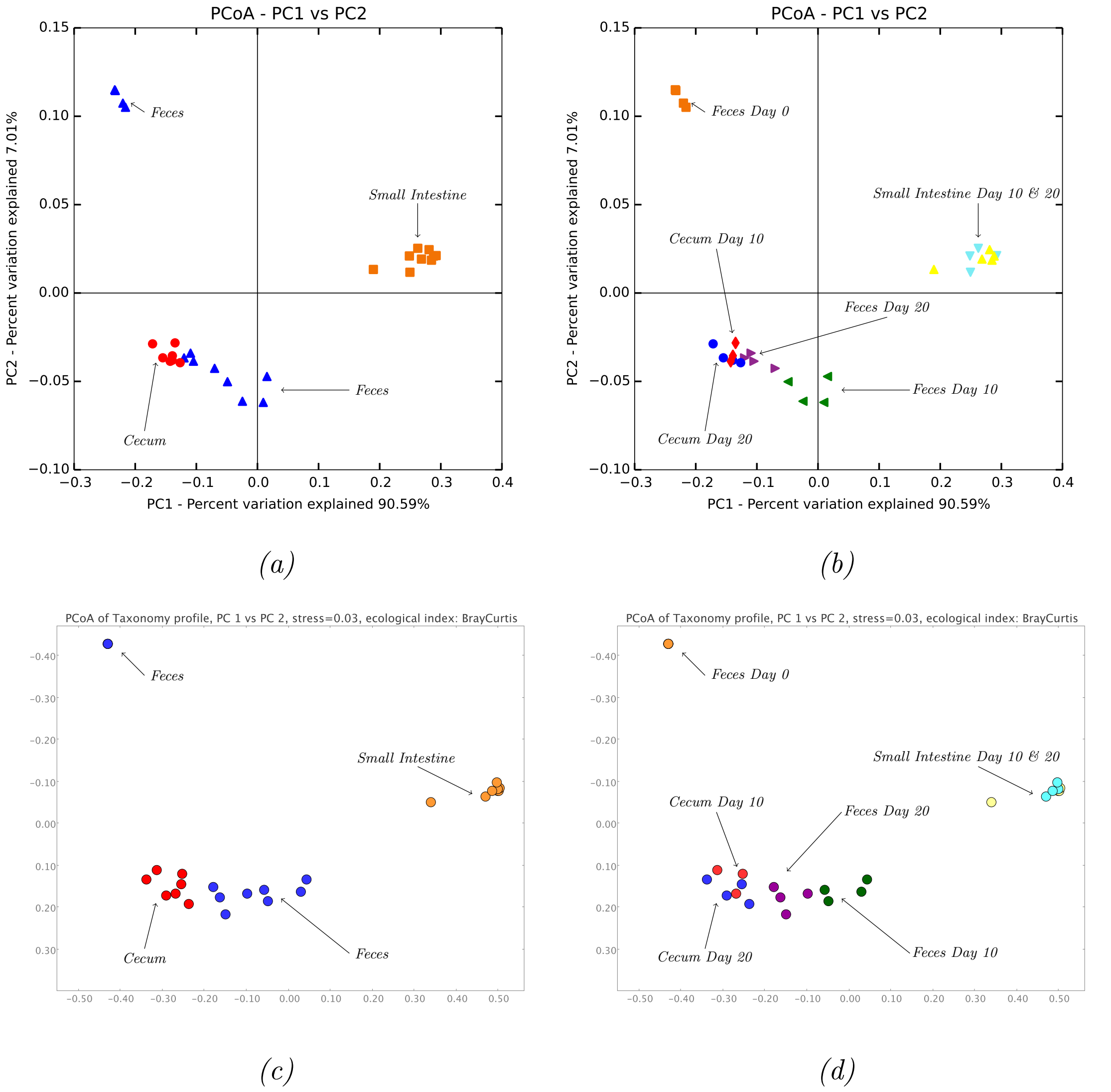
In QIIME, for each OTU one representative sequence was extracted, using the *cluster_seed* picking strategy. Multiple sequence alignment was performed on the set of

					Percent Identity at Alignment against Silva									
					100		99		98		97		96	
	%Id	Count	%Reads		LCA	MV	LCA	MV	LCA	MV	LCA	MV	LCA	MV
Percent Identity at Alignment against Reference	100	648	0.19	Same	206	648	206	648	206	648	206	648	206	648
				Higher	442	0	442	0	442	0	442	0	442	0
				Different	0	0	0	0	0	0	0	0	0	0
				No Hit	0	0	0	0	0	0	0	0	0	0
	99	336,930	96.37	Same	120,790	121,404	300,554	336,930	180,914	336,929	180,914	336,929	180,914	336,929
				Higher	615	1	36,376	0	156,016	1	156,016	1	156,016	1
				Different	0	0	0	0	0	0	0	0	0	0
				No Hit	215,525	215,525	0	0	0	0	0	0	0	0
	98	345,905	98.93	Same	120,790	121,404	300,956	337,346	188,299	345,895	186,553	345,895	186,553	345,895
				Higher	615	1	36,396	6	157,606	10	159,352	10	159,352	10
				Different	0	0	1	1	0	0	0	0	0	0
				No Hit	224,500	224,500	8,552	8,552	0	0	0	0	0	0
	97	346,628	99.14	Same	120,790	121,404	300,956	337,346	188,575	346,172	187,037	346,606	186,989	346,607
				Higher	615	1	36,397	6	157,618	18	159,588	15	159,637	15
				Different	0	0	3	4	2	5	2	6	2	6
				No Hit	225,223	225,223	9,272	9,272	433	433	1	1	0	0
	96	346,740	99.17	Same	120,790	121,404	300,957	337,347	188,579	346,176	187,079	346,648	187,060	346,696
				Higher	616	1	36,399	6	157,630	20	159,610	23	159,676	22
				Different	0	1	6	9	5	18	5	23	4	22
				No Hit	225,334	225,334	9,378	9,378	526	526	46	46	0	0

Table 7.7: ***Accuracy of Taxonomic Placement at Genus Level:*** *Sequences that align against the reference database at different percent identities - Rows - were extracted and aligned using MALT at different percent identities against the Silva NR99 115 database - Columns - and subsequently imported to MEGAN. The table shows that both algorithms, Lowest Common Ancestor (LCA) and Majority Vote LCA (MV), assign reads free of false positives. However, the LCA tends to place reads on the correct taxonomic path, but on higher levels than the genus level. The MV algorithm assigns more than 99% of reads to the correct genus.*

representative sequences using the PyNast [Caporaso et al., 2010a] algorithm. The resulting data served as input for the tree building tool fasttree [Price et al., 2009, 2010]. The tree served as input for the weighted Unifrac [Lozupone et al., 2006] distance metric. The transformation of the resulting distance matrix to a β-Diversity plot was performed using an internal QIIME routine. The resulting β-Diversity is depicted in Figures 7.2a and 7.2b.

After import to MEGAN, we compared all samples using the sub-sampling method. β-Diversity plots are generated using the Bray-Curtis dissimilarity metric [Bray and Curtis, 1957]. Results are depicted in Figures 7.2c and 7.2d.



(a)

(b)

(c)

(d)

Figure 7.2: β-diversity plots created with QIIME ((a) and (b)) and MEGAN ((c) and (d)): Colors and shapes differentiate between sample types. Figures (a) and (c) are colored by sample source. Figures (b) and (d) differ, additionally, between collection dates as explained in Table 7.1.

7.3 Results & Discussion

7.3.1 Not Under-representation but False Positives influence Taxonomic Analysis using Alignments

When using taxonomy-dependent methods for 16S rRNA sequence analysis, the assumption that the taxonomic information of those sequences which have no match in the reference database will be lost is a constant concern. As shown in Table 7.2 only two bacterial strains out of 12 have an exact 16S rRNA sequence match in the Silva database. However, if we perform alignment with relaxed percent identity criteria ranging from 99 to 96%, we see, as shown in Table 7.2, that although the exact sequence is not present in the Silva database the correct taxonomy identity can be assigned. These results can be verified by using the RDP classifier to taxonomically bin sequences, as shown in Tables 7.3.

However, due to the sequencers length limitations, the majority of studies performs taxonomic identification not on full-length 16S rRNA sequences but on shorter stretches. This is why we repeated the analysis, with the difference that this time we used reference sequences trimmed to a position and length which is typical for our "real" sequencing data. As we can see in Table 7.5, the alignment of each trimmed reference sequence finds the same most abundant genus as the alignment of the full-length reference. For this reason, we can say that during the downstream analysis, our taxonomic assignment is not biased by reduced sequence length of ∼250-300bp.

Contrary to the general concern, we identified a potential source of error not in under-representation but in false positives. We found that already at alignment with 99% identity (see Table 7.5) some sequences matched the Silva database at a phylum different to the correct taxonomic identity (see Figure 7.1). Further relaxation of the percent identity criteria to 98% added more false positives, thereby contradicting the rule according to which a 97% identity distinguishes sequences at species level. However, some reference sequences aligned to the correct genus only, and the number of cases where sequences were correctly assigned to the matching genus were generally by one order of magnitude higher when compared to the incorrect.

The results of the taxonomic analysis presented in this section are summarized in Figure 7.1 where the correct identity of each input sequence is found up to genus level. Assignment to the species level was not possible due to two reasons. First, at species level the same input sequence matched multiple references with identical identity, score and alignment. Secondly, most species that were matched did not have a name but were referenced as "*uncultured species*", adding no further information.

7.3.2 Deriving "true" Taxonomic Distribution: Alignment of 454-Sequences against the Reference Database

Unlike in mock communities which are generated in-vitro by mixing bacteria in known proportions and are sequenced, our approach contains sequences from an in-vivo experiment. As a consequence and in order to verify results presented in downstream analysis, first, we need to derive the correct taxonomic distribution by applying alignment methods using the reference database and, secondly, remove sequences that have a high chance of being contaminants, such as bacterial sequences that do not stem from any of the sequences in the reference database.

To do so, we aligned 349,639 sequences against the reference database, using MALT and usearch for percent identities ranging from 100-90%. Even though we would expect that following pre-processing, sequences would be free of the majority of errors, in our setting we saw only 648 sequences that could be aligned at 100% identity (see Table 7.4). However, the majority of sequences could be aligned at 99% identity. At 94% identity no further significant changes in the ratio between assigned and not assigned could be detected, implying that roughly 2,500 sequences are contaminants, originating from different bacterial species or representing an artifact of pre-processing.

The number of reads that are assigned to each of the reference sequences in the sample at different percent identities is depicted in Table 7.4 and Figure 7.1. Leaving results at 100% identity aside, when using the distribution at 99% identity as a reference, one can see in Figure 7.3, that through the course of 97 - 93% there are only minor changes in the distribution of reads among reference sequences. Changes in the distribution between 99% and 98% are limited to ± 3% in relation to the distribution one would expect by adding previously unassigned reads (Example: If 10% of reads are assigned to one reference and the number of all reads is 100, then if we add another 50 reads we would expect, considering the initial 10% to be representative, that 5 of the 50 additional reads are assigned to this reference sequence. On the other hand, if 10 more reads are assigned, the ratio changes from 10% (10 of 100) to 13% (20 of 150) which is a rise of 30%.). Despite the fact that we detected only a minor change of ± 3%, this poses the question if sequences that stem from certain bacteria are more prone to sequencing errors and/or if the accuracy suffers for intragenomic variation of the 16S rRNA gene

[Coenye and Vandamme, 2003; Clayton et al., 1995]. YL31 and ASF457 have significant changes of abundance, ranging between +20% for ASF457 at 98% identity to +80% at 94% identity for YL31. One explanation for these drastic changes might be the fact that both taxa are relatively rare, making up for only 0.13% of all reads. In other words, an additional 7 reads for ASF457 would result in a 20% rise in abundance. Another explanation for the increase of 80% for YL31 at 94% identity could be the introduction of contaminants which belong to the same genus or family. Based on the evidence summarized in Table 7.4 and shown in Figure 7.3, we concluded that the distributions retrieved from the alignment performed at 99% and above identity do not correctly represent the sample distribution and that a percent identity lower than 95%, most likely, introduces contaminants.

Furthermore, alignment performed with MALT leads to overall same results as usearch in full alignment mode as shown in Table 7.8. In addition, the analysis of reads that both tools assigned to different taxa, prompts us to conclude that the deviation stems from the methods applied to calculate the alignment using either a global or semi-global approach (See discussion at Section 7.3.4). In the context of this study, one can say that MALT finds, due to the nature of semi-global alignment, better and longer alignments.

%Id	Same	Different	% Different
100	349,639	0	0
99	349,534	105	0.03
98	349,311	328	0.09
97	349,457	182	0.05
96	349,470	169	0.04
95	349,427	212	0.06
94	349,406	233	0.06
93	349,401	238	0.06
92	349,405	234	0.06
91	349,408	231	0.06
90	349,412	227	0.06

Table 7.8: Number of Reads that align to the same or different taxon using MALT and usearch for varying percent identities.

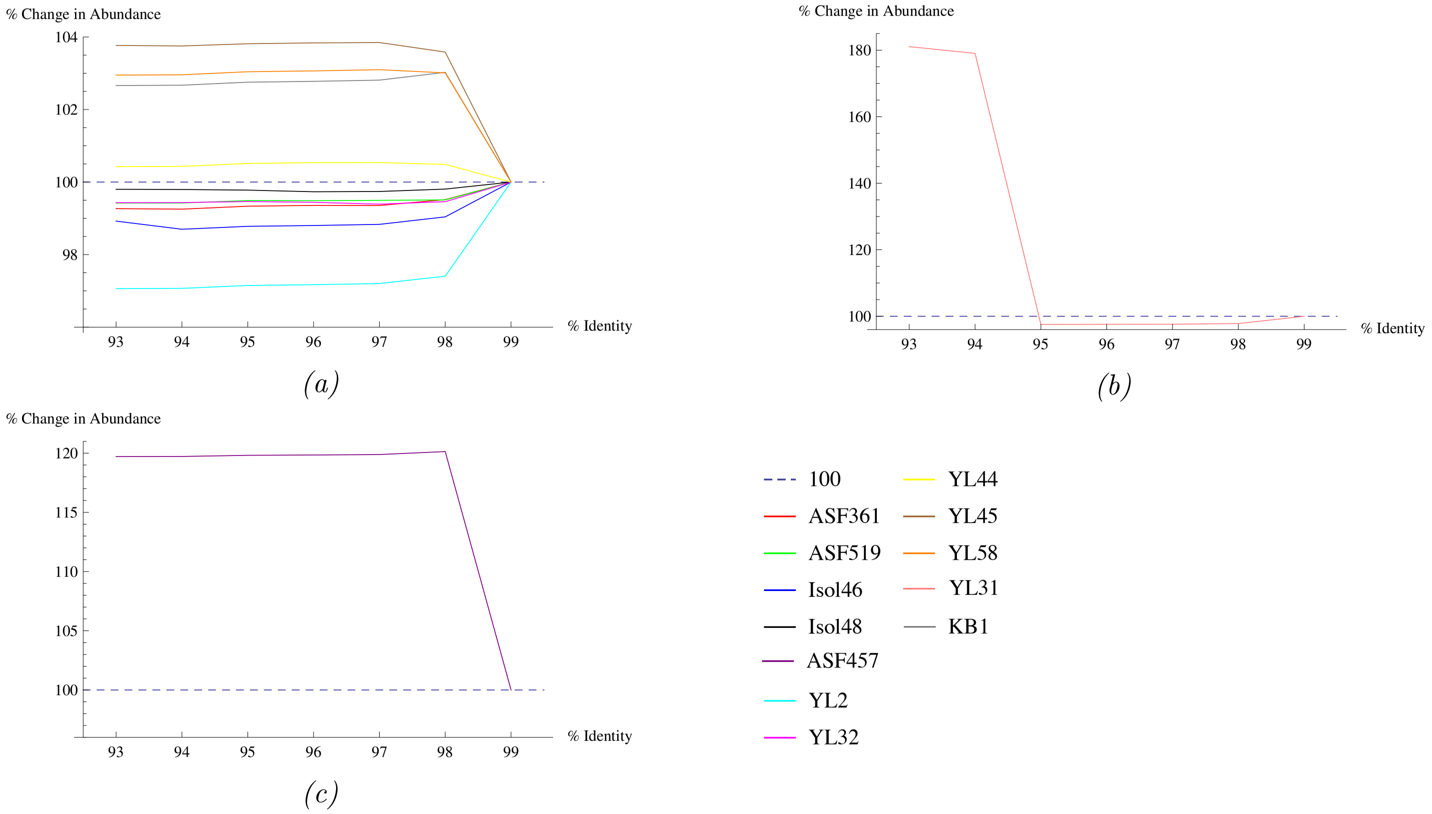


Figure 7.3: ***Distribution of reads among taxa using MALT****: The graphs show how the distribution among taxa changes from 99-93% identity, when using the distribution at 99% as a reference and expecting proportional growth with the growing number of reads, failed to align at higher percent identities. (a) The abundances among 10 of 12 references did not undergo significant changes (±3%). (b) A significant raise of abundance for YL31 at 94% identity. (c) A significant change of abundance for ASF457 at 98% identity.*

7.3.3 Taxonomic Assignment using MALT and MEGAN - Accurate and Free of False Positives

Traditionally, MEGAN assigns reads to nodes on the taxonomy using the lowest common ancestor algorithm (LCA). Before placement, the matches of each read are filtered to remove those of low quality. The remaining matches are considered significant, and are subsequently used as input for the lowest common ancestor algorithm. The algorithm is known to be relatively robust against false positives [Huson et al., 2007], since reads are assigned to the ancestor of all matches, as seen in Figure 7.5d. In the field of metagenomics, where one often finds very few matches per read that can pass the quality filter, and alignments are created using databases not specialized for taxonomic placement but rather to unravel the genetic content, this approach seems to be well accepted. However, for our analysis, we believe that the standard algorithm leads to a too conservative and thereby, to a taxonomic placement too close to the root. This assumption is fueled by the taxonomic assignments listed in Tables 7.2 and 7.5 where already at 99% identity reference sequences match different phyla. We assume that for real sequencing data, that may contain errors, the rate of false matches would be even higher. For that case, the lowest common ancestor will fail to accurately map the sequence to the correct genus. On the other hand, the Majority Vote LCA as an alternative approach to the LCA takes advantage of the fact that the number of matches for the correct genus is in general much higher than the count of false positives.

Majority Vote Lowest Common Ancestor As described above, prior to taxonomic placement, matches are quality filtered. This pre-processing step is performed in order to eliminate matches considered not to be significant, with the goal of increasing accuracy of taxonomic placement. The Majority Vote algorithm (MV) extends the functionality of the traditional lowest common ancestor approach by applying an additional filter, that eliminates matches not before but in the course of finding the LCA.

Figure 7.5 depicts the workflow of the MV. Initially, pre-filtered matches are placed on nodes of the taxonomy tree. For example, in Figure 7.5a, 90 out of 100 matches are placed on the *Clostridia* node. Parent nodes inherit the number of matches from their subtree, resulting in assignment of 95 matches to *Firmicutes*. The LCA algorithm traverses the tree leaf to root and places the read on the lowest node in which every match is captured. As a result, applying the LCA on the example tree places the read on the *Bacteria* node as seen in Figure 7.5d. The MV applies the same algorithm as the LCA but relaxes the condition of how many matches have to be embodied in the subtree from all matches (in LCA) to a user defined fraction. Therefore, a fraction of 90% assigns the read to *Clostridia* (see Figure 7.5b), a fraction of 95% places the read on *Firmicutes* (see Figure 7.5c) and a fraction of 100% leads to the same taxonomic placement as the LCA.

The LCA is free of False Positives, the Majority Vote Algorithm is also accurate Knowing the taxonomic identity for each reference sequence as derived from the Silva database (see Figure 7.1) and the distribution among reads (see Table 7.4) enables us to prove that MALT is capable of aligning most reads. Furthermore, MEGAN is capable of assigning reads avoiding false positives (less than 0.1%) and, finally, that the Majority Vote algorithm using a threshold of 90% of matches, assigns more than 99% of reads to the correct genus.

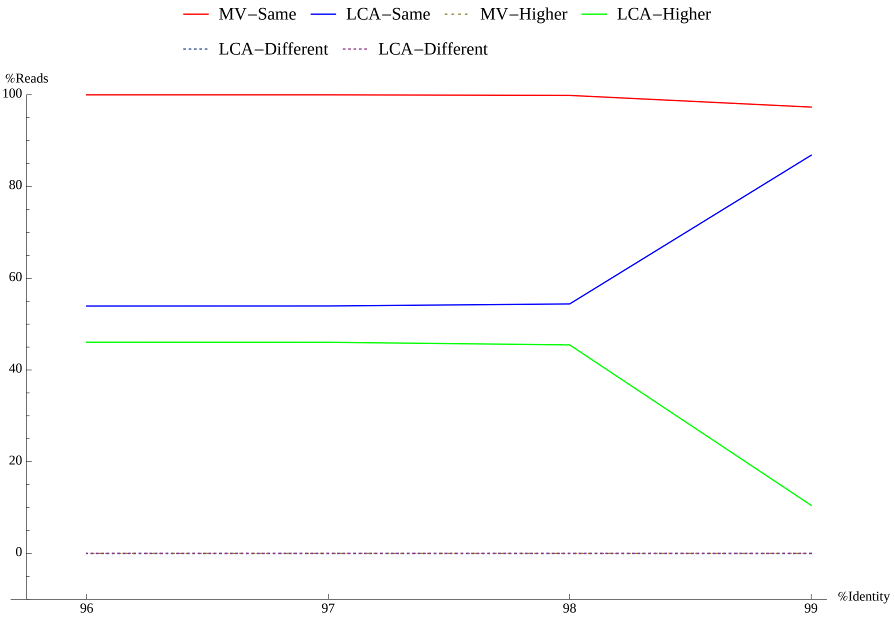
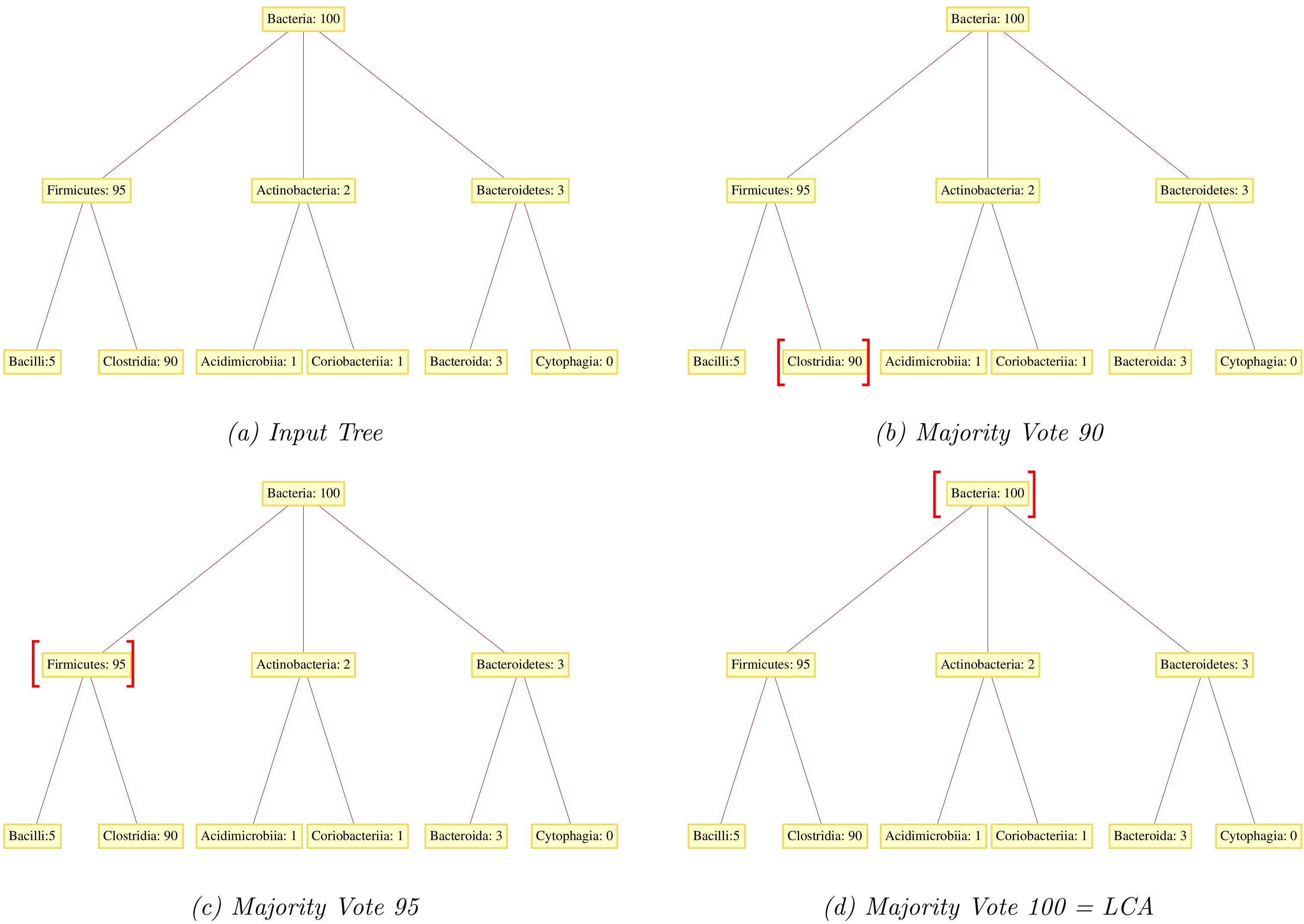


Figure 7.4: Accuracy of taxonomic placement using different percent identities in combination with the LCA and MV algorithms.



(a) Input Tree

(b) Majority Vote 90

(c) Majority Vote 95

(d) Majority Vote 100 = LCA

*Figure 7.5: **The Majority Vote LCA**: (a) The input tree: A number of matches is assigned to each node (e.g. 3 to Bacteroida). Central nodes, inherit matches from their child nodes. (b) Result of the Majority Vote algorithm at 90% is highlighted in red. (c) Result of the Majority Vote algorithm at 95% is highlighted in red. (d) Result of the LCA algorithm, equal to a Majority Vote at 100% is highlighted in red.*

As shown in Table 7.7, using the LCA and the MV algorithm on reads that were previously aligned using MALT against the Silva database, we can see that:

- MALT is capable of finding matches for most reads. The accuracy and performance of the alignment algorithm is underlined by the fact that (see Table 7.7) the number of reads which could not be aligned, drops to zero, if the percent identities of both alignment steps (rows and columns) are equal. Implying that MALT is capable of dealing with sequence errors and can still find the correct alignment.

- the rate of false positives ranges between 0 and 0.1% regardless the percent identities applied during alignment. However, as shown in Table 7.7, we can identify a rise in false positives when allowing sequences to align against the reference database only at 96% identity and when the criteria for alignment against Silva are also relaxed. The analysis of reads that lead to false positives shows that false positives are found on family level only and, that the source of errors lies in reads that are supposed to be placed to the *Lachnospiraceae;IncertaeSedis* genus but are falsely assigned to the *Lachnospiraceae;uncultured* genus. Another possible explanation implies that through the relaxed percent identity criteria the reads that we classify as false positives actually stem from different bacterial strains and therefore represent contamination.

- the MV assigns more than 99% of reads to the correct genus, regardless of the alignment parameters. Figure 7.4 shows the performance of the MV compared against the LCA. The LCA assigns roughly 80% of reads to the correct genus when using conservative alignment parameters at 99% identity but loses accuracy when allowing matches that align at only 98%. A comparison between LCA and MV at 97% identity after alignment against Silva is depicted in Figure 7.6 and shows that the majority of reads which the LCA maps to a higher taxonomic level are assigned to the *Bacteria* node and stem mostly from *Parabacteroides* and *Lachnospiraceae;IncertaeSedis*, therefore distorting the entire taxonomic distribution.

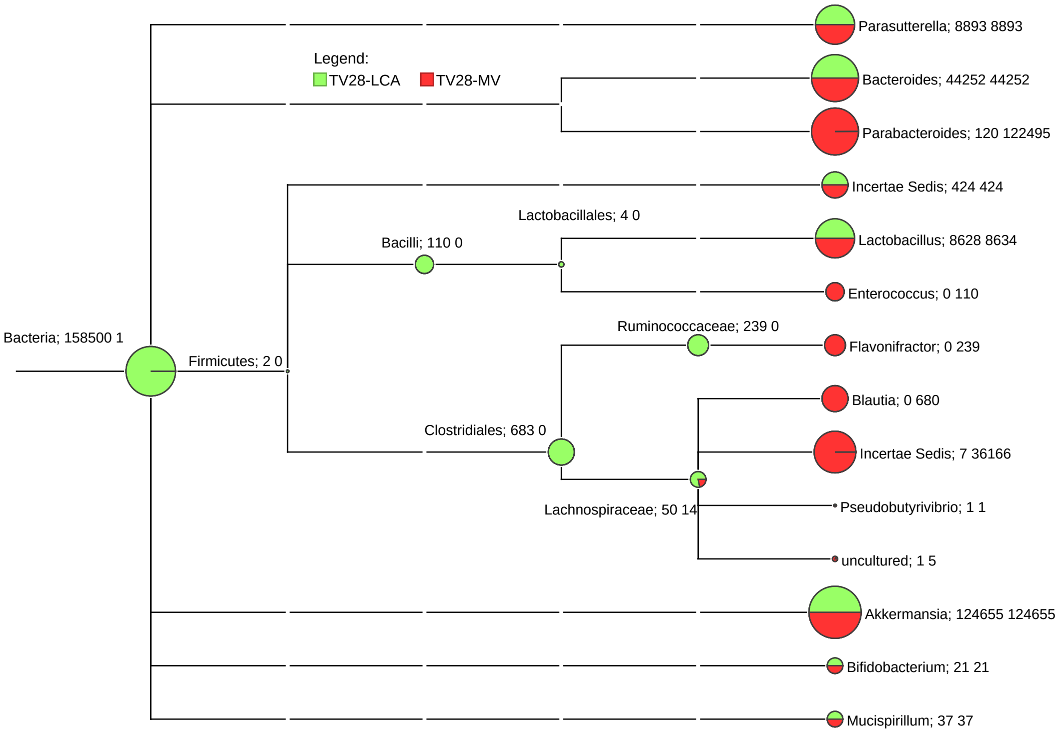


Figure 7.6: Comparison of taxonomic placement methods: Lowest Common Ancestor against Majority Vote.

7.3.4 Alignment of 16S rRNA Sequences - Semi-Global Outperforms Local and Global

In the course of this study we proved that MALT is capable of aligning 16S rRNA sequences in a way that MEGAN can translate the output to nearly optimal taxonomic assignment. We believe that the rationale behind the very good performance of MALT is in fact the semi-global alignment setting, which seems to be the superior choice as method for alignment of 16S rRNA sequences. In this section, we will discuss why we believe that the two other approaches often used for 16S rRNA analysis, namely local and global alignment, may lead to less significant results.

As input data for pairwise alignment we have, as depicted in Figure 7.7a, a reference sequence and a query sequence. Unlike in whole genome sequencing where random pieces of DNA are sequenced, possibly stemming from two different genes, in targeted sequencing, query sequences always stem from only one gene, or in other words, the query is a sub-sequence of the reference. Therefore, in this case the goal of pairwise alignment is to create an alignment which covers the entire query sequence, thus allowing us to retrieve only the most significant alignments.

Here, we describe three alignment methods used for 16S rRNA analysis:

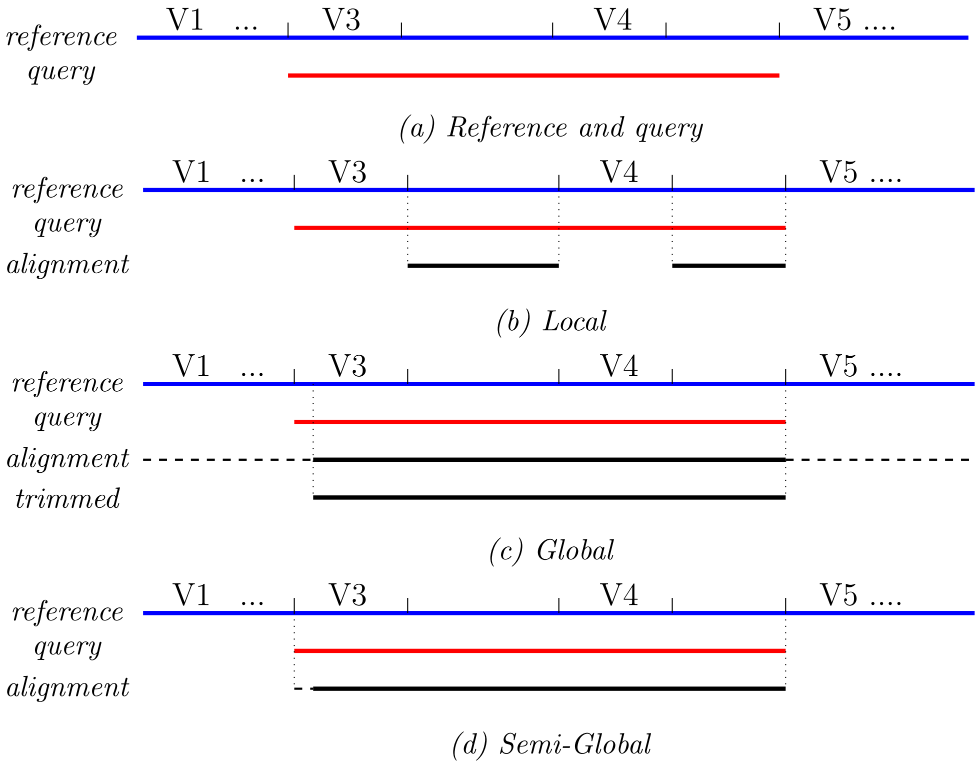
- **Local:** Local alignment finds high-similarity regions and implicitly ignores low-similarity ones. Therefore, local alignment on 16S rRNA sequences can result in several alignments for one query sequence, potentially omitting subsets of the hypervariable regions (see Figure 7.7b).
- **Global:** During global alignment, both sequences - reference and query - are aligned end-to-end. This approach solves the gap problems arising in local alignment. Nonetheless, the alignment retrieved has terminal gaps (see Figure 7.7c) which result in a biased percent identity and bitscore. Cutting terminal gaps away (as done by default by certain programs) solves the problem but it might lead to an alignment shorter than the query sequence.
- **Semi-Global:** Methods, such as MALT, supporting semi-global alignment, align the entire query sequence against the reference as depicted in Figure 7.7d. Terminal gaps are not cut off and may influence bitscore and percent identity negatively.

To summarize, we can say that local alignment can lead to multiple alignments per query which have a high percent identity but omit areas used to differentiate between species, such as variable regions. By cutting off terminal gaps located in the query, global alignments result in a high score and a high percent identity. Since, alignment for different query sequences may vary in the number of terminal gaps, this approach negatively influences the credibility of the percent identity parameter. Finally, semi-global alignment produces the most credible results even though scores and percent identities may be lower compared to global alignment.

However, in the process of evaluating and comparing alignments produced by either MALT with semi-global alignment and usearch with global alignment, we observed an additional bias introduced by global alignment. Possibly due to the algorithm usearch uses to calculate global alignments, gaps at the end of query sequences are introduced where MALT aligned these areas avoiding gaps at all.

7.3.5 Accurately estimating β-Diversity using MEGAN

β-Diversity is an important tool to detect compositional differences between samples and is essential for 16S rRNA analysis especially when considering the complexity introduced



(a) Reference and query

(b) Local

(c) Global

(d) Semi-Global

Figure 7.7: Methodologies for 16S rRNA pairwise sequence alignment.

by the large number of samples. To create plots first, distances or dissimilarities between samples are calculated using e.g. a tree or the taxonomic distribution in combination with a metric such as Unifrac. The resulting distance matrix accurately describes how samples differ but due to the high dimensionality this is inaccessible to the human eye. In the following step, the matrix undergoes a transformation in which distances are projected to two- or three dimensional coordinates with the aim of losing as little variance as possible. Finally, this data is plotted, displaying distances between samples.

Major 16S rRNA platforms such as QIIME or mothur but also the metagenomic software MEGAN support β-Diversity plots. Following the plan to establish MEGAN as a tool for 16S rRNA analysis, we will show that MEGAN is capable of accurately describing distances between samples for 16S rRNA data.

Contrary to the taxonomic placement analysis we presented in the previous sections, for this section we have no *correct* β-Diversity to which we could compare our results. Therefore, we compare results created in MEGAN with those created in QIIME and correlate the findings with the metadata as described in Table 7.1.

Figure 7.2 depicts the β-Diversity plots generated with QIIME using the weighted Unifrac distance metric and β-Diversity plots generated with MEGAN using the Bray-Curtis dissimilarity. Both approaches lead to highly similar clustering, with sampling time and site proving to be the discriminating metadata. First, feces samples taken at day 0 harboring only three bacterial strains significantly differ from all other samples. Secondly, feces samples collected at day 10 and 20 underwent a taxonomical shift, induced by feeding of 10 additional bacterial strains on day 0, and consequently a notable difference between them and day 0 samples is visible in form of an additional cluster. Additionally, one can detect compositional differences between day 10 and 20. Furthermore, cecum samples from both, day 10 and 20 form a third cluster located in the direct neighborhood of the feces samples. Finally, samples taken from the small intestine create a fourth and distant cluster.

Taking the plots generated by QIIME and additional information about the samples in account, we can most certainly say that MEGAN in combination with our taxonomic placement approach is capable of correctly measuring distances between samples and, therefore, of successfully discriminating sample types based on metadata.

7.3.6 Computational and Algorithmic Challenges of 16S rRNA Analysis

The main objective of clustering, in terms of 16S rRNA analysis is to identify exactly one OTU per species present in a sequencing sample. Knowing that the number of species is by far smaller than the number of sequences, this approach significantly reduces the runtime for taxonomic assignment as well as improves the accuracy of downstream analysis which depends, in the case of β-Diversity, on a tree built from representative sequences. However, clustering can also lead to poor results, introducing new species originating from e.g. sequencing errors [Huse et al., 2010; Marco, 2010; Quince et al., 2009] or falsely add rare species to clusters formed by a highly abundant species thereby losing resolution.

Schloss analyzed factors that influence the outcome and the quality of clustering (see [Schloss, 2010]) and, consequently, a new generation of clustering tools emerged (or was updated), such as ESPRIT [Cai and Sun, 2011; Sun et al., 2009], uclust [Edgar, 2010], uparse [Edgar, 2013], CROP [Hao et al., 2011], Muscle [Edgar, 2004], SLP [Huse et al., 2010] or CD-Hit [Huang et al., 2010]. However, comparison of these tools shows that the species richness is still overestimated 2 to 10-fold [Chen et al., 2013; Bonder et al., 2012]. For example, clustering, using QIIME on our data, as described in the previous section, lead to 58 OTUs at 97% similarity, and by that, overestimating the number of species by a factor of 5.

Nonetheless and even though alignment based methods do not suffer from similar setbacks, clustering is still the first choice for the analysis of 16S rRNA sequences, and the low computational cost is one of the reasons for this. Clustering and taxonomic assignment as performed on 349,639 sequences in the previous section using QIIME, took ∼10 minutes using a single core and 4GB of memory. Alignment and import to MEGAN, on the other hand, took ∼2:30 hours, required 32 cores and ∼60GB of memory. However, our approach in which we dereplicate sequences before alignment and rereplicate alignments after computation, lowered the runtime to ∼6 minutes at 32 cores and reduced the main memory requirements to ∼40GB. We believe that both, runtime and memory footprint can further be decreased by trimming the full-length 16S rRNA database sequences to the area in which the query sequences are located using V-Xtractor [Hartmann et al., 2010].

Nonetheless, the performance of clustering is greatly influenced by the complexity of the input samples. If sequences have varying length or are biased by undetected sequencing errors, the runtime of clustering is significantly increased whereas the runtime of the alignment approach is relatively linear.

In other words, typical 16S rRNA analysis pipelines, such as mothur or QIIME, are faster and less expensive approaches, however they only offer overall abundance information among the taxa, not more, not less. On the other hand MEGAN offers a large variety of tools, for instance, users can inspect alignments with the aim of verifying taxonomic assignments, and with these “extra” tip the balance in favor of more accurate results. For this reason, and considering that the computation has to be performed once per study, computational overhead becomes somewhat less important.

7.4 Conclusion

While studying biodiversity, the key to a successful analysis and the starting point for downstream analysis lies in the correct taxonomic assignment of input sequences. In this chapter we introduced a new analysis approach for 16S rRNA sequences which proved to be capable of assigning more than 99% of all input sequences to the correct genus. To do so, we established a pipeline using MALT as aligner and Silva as reference database. For taxonomic placement we implemented a new algorithm, namely Majority Vote, and added its functionality to MEGAN. We also showed that MEGAN is capable of creating β-Diversity plots similar to those generated by QIIME, when applying our approach.

Furthermore, we believe that by enabling MEGAN to deal with 16S rRNA data, less tech-savvy users will profit from the user-friendly graphical interface which MEGAN offers, when compared to typical command-line based 16S rRNA pipelines.

Part IV

Building Blocks, Falling into Place

In this part we will show how MEGANServer, as introduced in Chapter 5, in combination with MEGAN and the 16S rRNA analysis pipeline, introduced in the previous chapter, can be combined in such way that it can serve as an accurate and user-friendly 16S rRNA analysis platform.

Chapter 8

Accurate Analysis of a Large-Scale 16S rRNA Project Using MEGAN and MEGANServer

8.1 Introduction

In the previous chapter, we introduced a novel taxonomy-dependent approach for 16S rRNA analysis for which we showed that alignment in combination with analysis in MEGAN can lead to an accurate and performant taxonomic placement, provided it is applied on 454-sequencing data. With MEGAN, we can open and extract relevant information on the datasets either in isolation to derive a taxonomic profile, for example, or compare a set of samples by measuring their β-Diversity. Contrary to the typical 16S rRNA analysis pipeline, MEGAN offers a user-friendly graphical interface which enables also less tech-savvy users to browse and analyze data on their own.

A feature MEGAN cannot offer but is essential to all 16S rRNA analysis tools is the capability to utilize metadata to split, merge and/or search for datasets. This is due to the fact that MEGAN treats and maintains datasets individually - datasets are aligned and imported to MEGAN separately. That is why, comparative analysis of a larger number of samples in MEGAN require additional efforts to guarantee that, for example, all datasets underwent the exact same treatment in terms of alignment and import to MEGAN. The standardized storage, maintenance and access to datasets is, on the other hand, one of the benefits when using MEGANServer. Taking into account that MEGANServer is also capable of using metadata, in a way essential to 16S rRNA analysis, we can claim that, with MEGANServer as a data backend, MEGAN's function pool is also suited to perform large-scale 16S rRNA analysis.

In this chapter, we present our analysis pipeline, which combines findings from previous chapters. First, sequencing data is pre-processed as discussed in Chapter 6. Secondly, we derive the taxonomic content for each sample with the method developed in Chapter 7. Finally, we upload the resulting files to MEGANServer (see Chapter 5) and provide access through MEGAN, in order to review their content. For demonstration, we apply the pipeline on 16S rRNA sequencing data which stems from a study (see [Gronbach et al., 2014]) that investigated endotoxicity of certain mouse gut flora[1].

8.2 Study Background

This study investigates how the gut flora, specific bacteria and/or biological molecules influence the development of colitis in mice. For this purpose, germ-free C57BL/6J-Rag^{1tm1Mom} (Rag1$^{-/-}$) mice were colonized with two types of complex intestinal microbiota. Mice with the Endo^{hi} (high endotoxicity) microbiota developed colitis shortly after transfer of CD4^{+}CD62L^{+} T cells, whereas mice colonized with the Endo^{lo} (low endotoxicity) microbiota maintained homeostasis. The fundamental difference between both microbiota could be identified as a low proportion of *Bacteroidetes* in combination with a high proportion of *Enterobacteriaceae* in Endo^{hi} and the exact opposite proportions for the Endo^{lo} microbiota.

Assuming that the increased endotoxicity was caused by the lipopolysaccharide (LPS) of *Enterobacteriaceae*, mice of both microbiota were administered *Escherichia coli* JM83 (high endotoxic LPS as in *Enterobacteriaceae*, $E.coli_{WT}$) and *Escherichia coli* JM83 + htrB$_{PG}$ (mutated, low endotoxic LPS, similar to that of *Bacteroidetes*, $E.coli_{MUT}$). Regardless of the initial gut flora, treatment with $E.coli_{WT}$ caused colitis, whereas mice receiving $E.coli_{MUT}$ preserved homeostasis. This experiment was repeated, omitting the *E.coli* and directly administering either highly endotoxic LPS (LPS^{hi}) or low endotoxic LPS (LPS^{lo}), leading to the same results as the previous one. In a fourth

[1] We will not provide an in-depth analysis of the data. This is the task of another PhD thesis. We will give a brief introduction on the data, how we analyzed the data and how to use MEGANServer in combination with MEGAN to retrieve results.

experiment, *Bacteroides vulgatus*, a bacterium that is known to protect against *E.coli*-induced colitis [Waidmann et al., 2003], was administered at different time points, once before colitis development and once during early stages of disease, to test whether the progression of the disease is reversible.

8.3 Experimental Setup & Sequencing Data

Mice of both microbiota underwent treatment as described in the previous section and as depicted in Figure 8.1. Treatment expected to lead to homeostasis is colored in green. The color red indicates that these mice are expected to develop colitis. For each of these treatments, fecal samples of mice with either Endo^{lo} or Endo^{hi} microbiota were collected at four time points, week -1, 0, 3 and 6.

Altogether, 237 samples from 75 mice were collected and sequenced with a 454 GS-FLX+ sequencer at Eurofins.

8.4 16S rRNA Analysis Pipeline

Six sequencing runs resulted in 2 million sequences with an average sequencing length of 511bp. This includes 28bp for primer and barcode at the front, and the low-quality tail at the end of the sequence.

Pre-Processing Pre-processing, as described in Chapter 6, emitted 1.3 million high quality sequences with an average length of 281bp. Sequences that have been removed were for the most part either too short (shorter than 200bp after quality control) or were identified as chimeric sequences (∼20%).

The smallest sample contains 869, the largest sample 18,490 sequences, at an average of 5,394 sequences per sample. The standard deviation is σ=3,362. The shortest sequence is 253bp, the longest sequence is 293bp with a standard deviation of σ=9.

Taxonomic Assignment For taxonomic assignment we applied the pipeline that we introduced in Chapter 7. Before alignment we merged all input files and de-replicated their sequences. With that step, we reduced the number of sequences to be aligned from 1.3 million to 0.2 million. Sequences were aligned using MALT in *semi-global* mode at a percent identity of 95. The number of matches was restricted to 100. The alignment required 115 minutes using 32 cores with 47GB of main memory. Subsequently, we

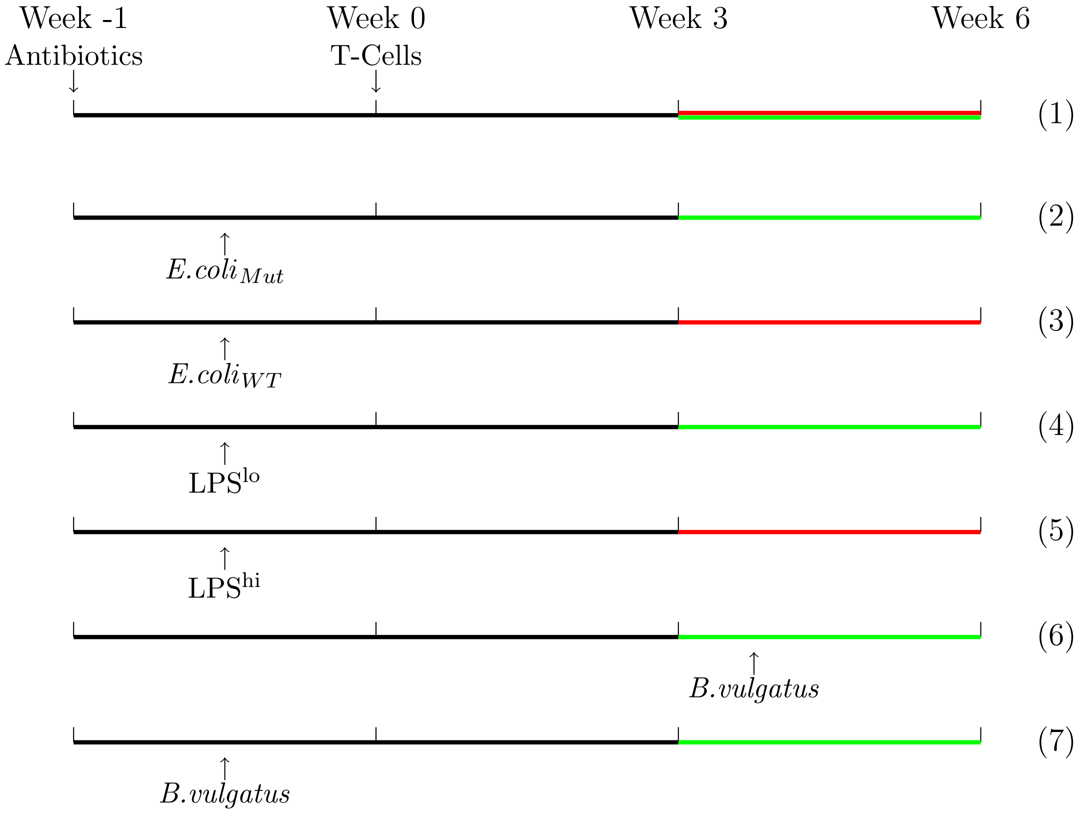


*Figure 8.1: **Experimental Setup**: Mice of both microbiota are treated with antibiotics at week -1. At week 0, T-Cells are administered. Treatment with bacteria or LPS is performed at either day -3 or week 3. The color indicates the expected health state at week 6. The two colored bar at (1) indicates that the health status depends on the microbiota.*

re-replicated and demultiplexed the resulting alignment file. Import to MEGAN of 1.3 million sequences and 85.5 million matches required 86 minutes. The Majority Vote algorithm applied at 90% confidence, mapped 95% and 98.5% of reads to genus and to family level, respectively. Of the 1.3 million reads, 16,500 (1.26%) could not be aligned or lead to only low quality matches. The resulting 237 MEGAN files require 39GB of disk space.

Upload to MEGANServer The 237 MEGAN files were uploaded to a MEGAN-Server instance. Upload of all files required 7 hours and consumed 32.2GB disk space. If the files are uploaded in summary format, the runtime is reduced to 30 seconds and only 3MB of disk space is required.

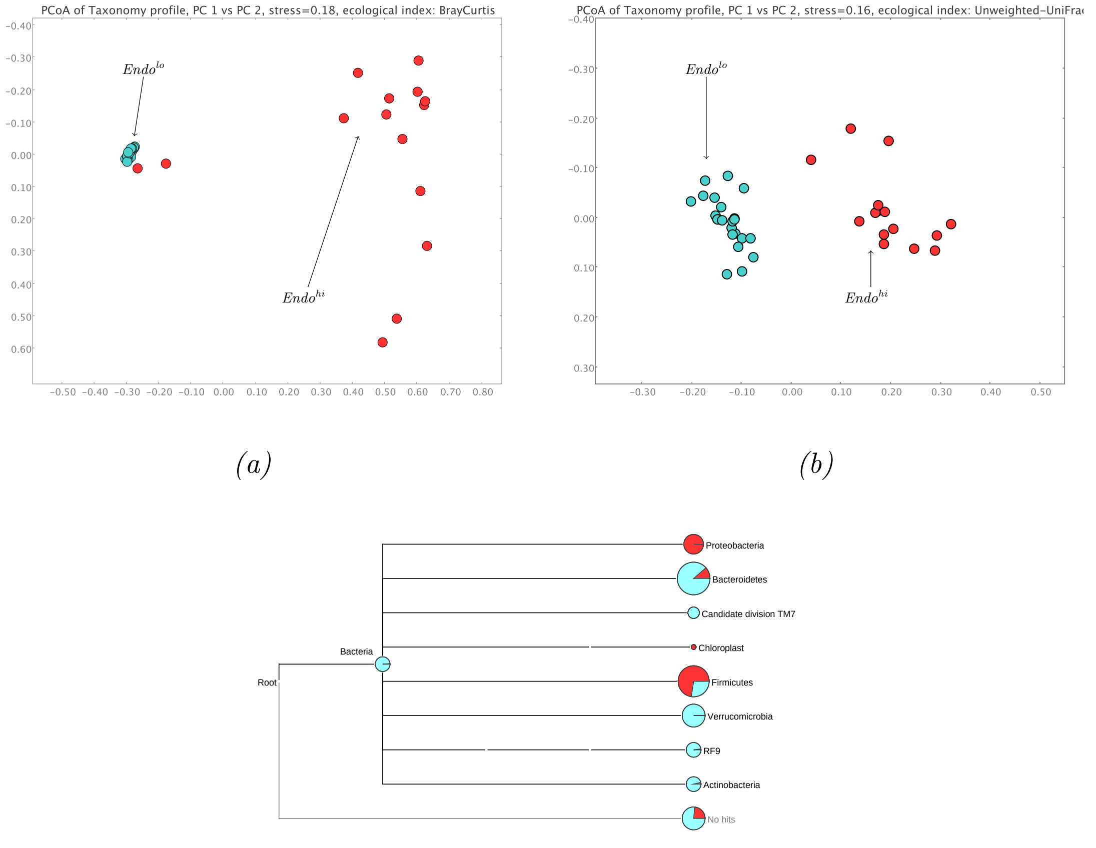
Each dataset was enriched with 32 types of metadata following the recommendations of the *Genomic Standards Consortium.* The fields are: *SampleID, BarcodeSequence, LinkerPrimerSequence, MouseNumber, MouseName, SamplingTime, EndOfExperiment, PoolPrepProtokol, PoolPrep, SequencingDate, ExperimentNumber, AnimalFacility, Supplier, Gender, Spleen, MLN, cLP, Feces, Comment, HistoScore, Healthstate-Organ, HealthStateHisto, HealthStateEnd, ExpectedHealthEnd, Treatment, Antibiotics, Birthday, DateTcellTransfer, Parents, CellCountMLK, CellCountcLP, cLPCD3CD4.*

8.5 MEGANServer for 16S rRNA analysis

Once the upload is completed, the taxonomical content and the differences among samples can be accessed. We will explain how the functionality of MEGANServer, in combination with the analysis capabilities of MEGAN, can help to assess underlying patterns.

Compositional Differences between Endo$^{\mathrm{hi}}$ and Endo$^{\mathrm{lo}}$ Assuming that the taxonomic composition in the intestinal microbiota plays a major role in disease development, a comparison of samples collected at week -1 and originating from Endo$^{\mathrm{hi}}$ microbiota with those which stem from Endo$^{\mathrm{lo}}$ microbiota should reveal differences. Since sampling at week -1 was not performed thoroughly, we compare samples from week 0 by estimating their β-Diversity. To select datasets, we used the metadata analyzer with the boolean expression *SamplingTime = 'week0'* and opened resulting datasets in a comparison file. The β-Diversity, applying the Bray Curtis dissimilarity (see Figure 8.2a), detects compositional differences between Endo$^{\mathrm{hi}}$ and Endo$^{\mathrm{lo}}$. Whereas the samples that stem from mice with an Endo$^{\mathrm{lo}}$ microbiota seem to cluster very well, the samples from mice with an Endo$^{\mathrm{hi}}$ microbiota show a scattered pattern. That could be due to a faster response to the different treatments at day -3.

The clear separation among microbiota using Unifrac as a β-Diversity metric (see Figure 8.2b) suggests that there are indeed taxonomical differences between microbiota. Since a naive comparison of all 35 samples (14 from Endo$^{\mathrm{hi}}$, 24 from Endo$^{\mathrm{lo}}$) would lead to an imprecise result due to a bias introduced by sub-sampling among all 35 samples, for example, we need to merge samples before comparison. Thereby, we extend the functionality from the comparison of samples to the comparison of scenarios. To do so, we merge all samples that stem from Endo$^{\mathrm{lo}}$ mice at week 0 in one dataset. The second dataset incorporates all samples that stem from Endo$^{\mathrm{hi}}$ mice at week 0. Taxonomic distribution of these two datasets at phylum level are shown in Figure 8.2c.



(a) *(b)*

(c)

Figure 8.2: (a) and (b): β-Diversity Plots of Endo^{hi} *and Endo*^{lo} *(regardless the treatment) microbiota at week 0. In (c) the same data is grouped by microbiota and depicted as taxonomic distribution at phylum level (log-based scale).*

Development of Taxonomic Distribution after $\boldsymbol{E.coli_{MUT}}$ **Treatment** Regardless of the initial microbiota, the treatment with $E.coli_{MUT}$ resulted in mucosal homeostasis, (see Figure 8.1 (2)). That leads to the question of if and how the treatment altered the microbiota during the course of the 6 week experiment period. To identity changes over a period of time we need to compare samples from both microbiota which underwent treatment with $E.coli_{MUT}$ at 3 time points, namely week 0, 3 and 6. The boolean expressions that need to be evaluated are shown in Figure 8.3. For each of the expressions, the resulting samples are merged and 6 new datasets are created. The most abundant phyla are shown in Figure 8.4 and lead to the conclusion that, compared to the Endo^{lo} microbiota, the Endo^{hi} underwent a larger taxonomical shift.

'AnimalFacility' = *'Endo*^{hi}*'* **AND** *'SamplingTime'* = *'week 0'* **AND** *'Treatment'* = *'E.coli*$_{MUT}$*'*
'AnimalFacility' = *'Endo*^{hi}*'* **AND** *'SamplingTime'* = *'week 3'* **AND** *'Treatment'* = *'E.coli*$_{MUT}$*'*
'AnimalFacility' = *'Endo*^{hi}*'* **AND** *'SamplingTime'* = *'week 6'* **AND** *'Treatment'* = *'E.coli*$_{MUT}$*'*
'AnimalFacility' = *'Endo*^{lo}*'* **AND** *'SamplingTime'* = *'week 0'* **AND** *'Treatment'* = *'E.coli*$_{MUT}$*'*
'AnimalFacility' = *'Endo*^{lo}*'* **AND** *'SamplingTime'* = *'week 3'* **AND** *'Treatment'* = *'E.coli*$_{MUT}$*'*
'AnimalFacility' = *'Endo*^{lo}*'* **AND** *'SamplingTime'* = *'week 6'* **AND** *'Treatment'* = *'E.coli*$_{MUT}$*'*

Figure 8.3: Boolean expressions to extract datasets after E.coli$_{MUT}$ *treatment at three sampling times for both microbiota.*

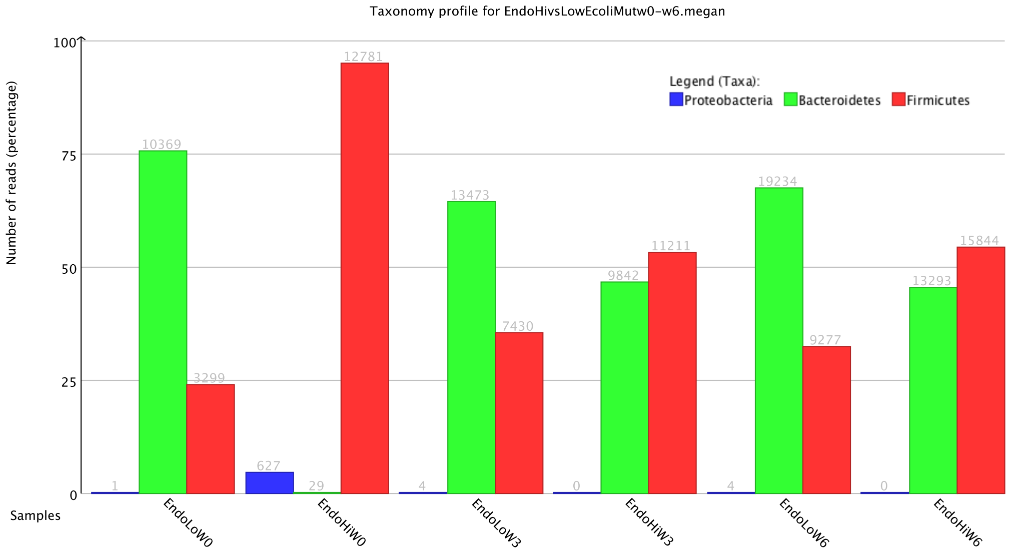


Figure 8.4: Development among most abundant phyla in Endo^{hi} *and Endo*^{lo} *microbiota after treatment with E.coli*$_{MUT}$*.*

8.6 Conclusion

In this chapter we introduced a novel analysis pipeline for 16S rRNA sequencing data. The pipeline covers the entire analysis process, which begins with raw input sequences and ends with providing an accurate taxonomic description, visually accessible using MEGAN.

The pipeline begins with pre-processing raw sequencing data, as described in Chapter 6. In this step erroneous sequences were discarded and/or low quality tails of sequences were removed. The remaining high quality sequences were used as input for the alignment and taxonomic placement using MALT and the Majority Vote algorithm. The combination of both tools led to a fast and accurate taxonomic placement as described in Chapter 7. In order to provide enhanced capabilities in terms of comparing and to effectively use metadata, resulting datasets were uploaded to MEGANServer. Finally, visual inspection of data was provided using the MEGAN software.

Part V

Conclusion

The capability of next generation sequencers of emitting enormous volumes of data at a moderate cost has changed the field of metagenomics. While early studies investigated relatively small samples in isolation, current studies effectively target questions that require deeper sequencing of a larger number of samples. As a consequence of this development it becomes increasingly difficult to perform the computational component of the analysis on a desktop computer. In fact, for that reason, we can observe a change in how studies are conducted. Bioinformaticians develop analysis tools for large-scale sequencing data and perform the calculation of alignments, for example, on a specialized computing environment. Furthermore, they provide resulting data files to medical staff who then qualitatively analyses the data using MEGAN, to correlate environmental parameters to changes in taxonomical distributions, for instance. Consequently, due to the increasing sequencing volumes growing file sizes, qualitative analysis on desktop computers becomes increasingly difficult. Files simply outgrow hard disks of normal home computers. Thus a different approach is needed to organize data files. For that reason, we developed MEGANServer. MEGANServer allows bioinformaticians to retain data files on a server with sufficient resources. Furthermore, we extended MEGAN to communicate with MEGANServer and by that enable researchers to perform their analysis on a home computer regardless the actual data size. Moreover, to overcome the complexity introduced by the growing number of samples, selection of datasets of interest is automated by metadata-based grouping. In addition, following the analysis strategy of the 16S rRNA studies, datasets can be opened applying different strategies, for instance as merged data, in order to provide a deeper insight on taxonomic and/or functional distribution.

In fact, the fields of metagenomics and microbiome studies are converging, with respect to the 16S rRNA based analysis. They ask similar questions, rely on similar analysis methods and base their findings on the same visualizations. Therefore, we extended MEGAN in such a way that it can now also deal with sequences that stem from a targeted sequencing approach. More precisely, we have developed a pipeline that covers the entire workflow, starting at pre-processing and, in a final step, allowing qualitative analysis using MEGAN. For that, we took advantage of a novel aligner, namely MALT, that in combination with a placement algorithm, namely the Majority Vote LCA, introduced recently in MEGAN, is capable of assigning more than 99% of reads to the correct genus and lowers the rate of false positives to a value close to 0%. We believe that, by the additional utilization of the different data access strategies implemented in MEGANServer, MEGAN is now fully capable of serving as a powerful, yet user-friendly analysis tool for 16S rRNA sequencing data.

Part VI

Appendix

Appendix A

Contributions

MEGANServer

Hans-Joachim Ruscheweyh (HJR) and Daniel Huson (DH) contributed to this project. HJR designed and implemented MEGANServer, the ServerBrowser and the MSUploader. DH defined methods for the global data access interface (IConnector) and updated MEGAN.

MEGAN for Targeted Sequencing

Hans-Joachim Ruscheweyh (HJR), Daniel Huson (DH) and Barbara Stecher (BS) contributed to this project. Sequencing data and the reference database were generated by BS. DH implemented the aligner MALT. MEGAN was implemented by DH with additions from HJR. HJR, DH and BS conceived the study. HJR conducted the analysis, implemented the Majority Vote algorithm and wrote scripts for analysis and data transformation.

MEGANServer for Accumulated Targeted Sequencing

Hans-Joachim Ruscheweyh (HJR), Daniel Huson (DH), Julia-Stefanie Frick (JSF) and Isabell Flade (IF) contributed to this project. JSF and IF generated sequencing data and conceived the study. HJR and DH conducted the analysis.

Appendix B

Publications

B.1 Publications

Gronbach, K., Flade, I., Holst, O., Lindner, B., Ruscheweyh, HJ., Wittmann, A., Menz, Sarah., et al. **“Endotoxicity of Lipopolysaccharide as a Determinant of T-Cell Mediated Colitis Induction in Mice”** Gastroenterology (2013).

Huson, D., Mitra, S., Ruscheweyh, HJ., Weber, N. and Schuster, SC., **“Integrative analysis of environmental sequences using MEGAN4”** Genome research 21, no. 9 (2011): 1552-1560.

B.2 Publications in Preparation

Ruscheweyh, HJ., Huson DH., **“Webserver-supported storage of metagenomic datasets using MEGANv5”**

Ruscheweyh, HJ., Stecher, B., Huson DH., **“Taxonomy-dependent microbiome analysis using MALT and MEGAN”**

Brugiroux S., Beutler M., Ruscheweyh HJ., Diehl M., Berry D., Loy A., Huson D., Heesemann J. and Stecher B., **“The Oligo-MM: a novel gnotobiotic model to study the mechanism of colonization resistance in mice”**

Beutler M., Brugiroux S., Ruscheweyh HJ., Ring D., Berry D., Loy A., Huson D., Heesemann J. and Stecher B, **“The impact of Salmonella-infection on the composition of the Oligo-MM”**

Appendix C

Supplements

C.1 16S rRNA Analysis using MEGAN

C.1.1 Material & Methods Flowchart

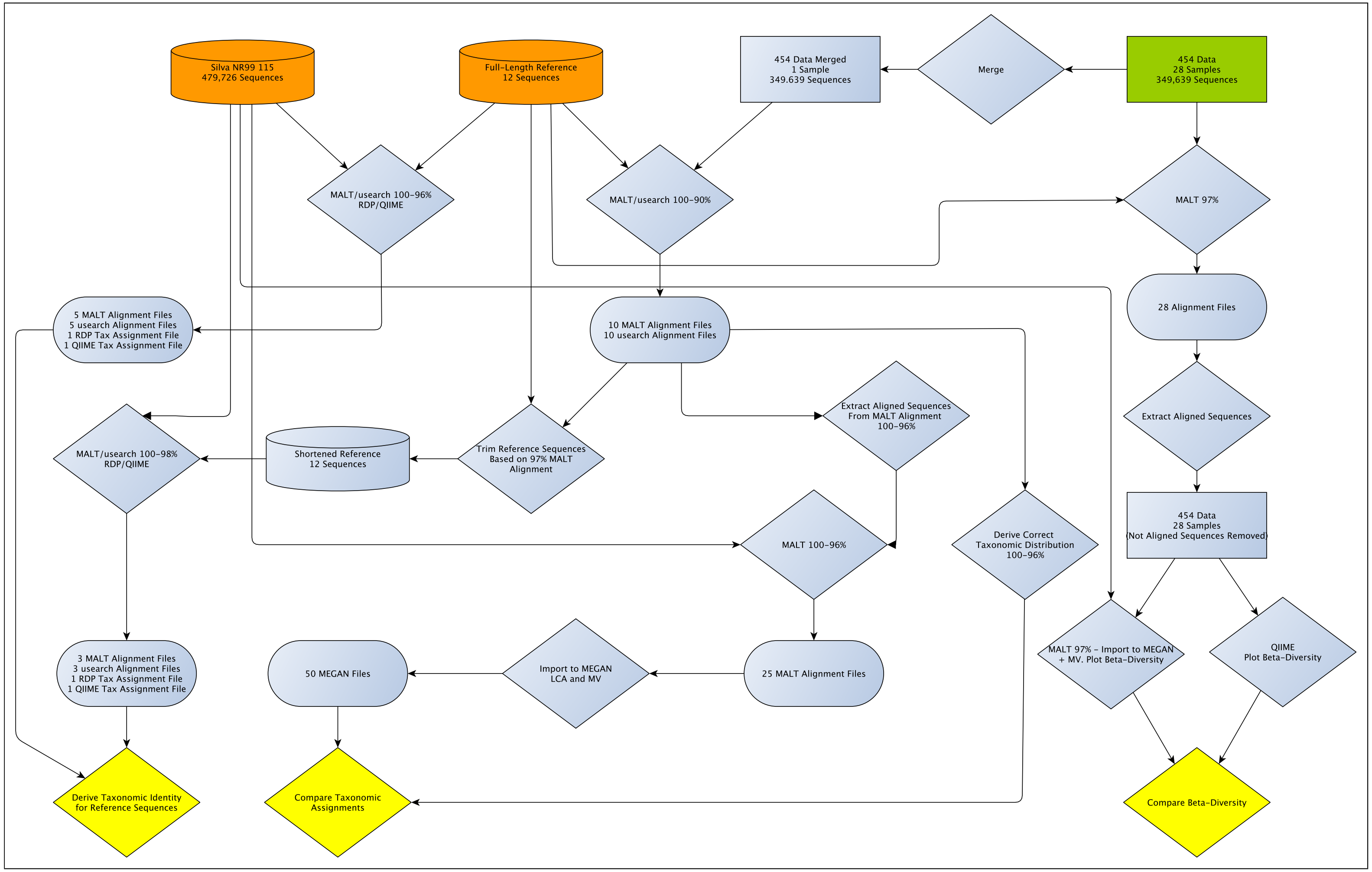


Figure C.1: Analysis workflow for Chapter 7. Input data is colored green. Databases are colored orange. Results are colored yellow.

C.1.2 Taxonomic Assignment of Full-Length Reference Sequences

	Taxon	Confidence
ASF361	Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus	1.0
ASF457	Deferribacteres;Deferribacteres;Deferribacterales;Deferribacteraceae;Mucispirillum	1.0
ASF519	Bacteroidetes;Bacteroidia;Bacteroidales;Porphyromonadaceae;Parabacteroides	1.0
Isol46	Firmicutes;Erysipelotrichia;Erysipelotrichales;Erysipelotrichaceae;Erysipelotrichaceae;IncertaeSedis	1.0
Isol48	Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides	1.0
KB1	Firmicutes;Bacilli;Lactobacillales;Enterococcaceae;Enterococcus	1.0
YL2	Actinobacteria;Actinobacteria;Actinobacteridae;Bifidobacteriales;Bifidobacteriaceae;Bifidobacterium	1.0
YL31	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Flavonifractor	1.0
YL32	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Clostridium XlVa	1.0
YL44	Verrucomicrobia;Verrucomicrobiae;Verrucomicrobiales;Verrucomicrobiaceae;Akkermansia	1.0
YL45	Proteobacteria;Betaproteobacteria;Burkholderiales;Sutterellaceae;Parasutterella	1.0
YL58	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Blautia	1.0

Table C.1: Taxonomic assignment of full length reference sequences using the rdp classifier.

	Taxon	MALT	USEARCH
ASF361	Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus	1	1
ASF457	Deferribacteres;Deferribacteres;Deferribacterales;Deferribacteraceae;Mucispirillum	2	2
ASF519	NO HIT	0	0
ISOL46	NO HIT	0	0
ISOL48	NO HIT	0	0
KB1	NO HIT	0	0
YL2	NO HIT	0	0
YL31	NO HIT	0	0
YL32	NO HIT	0	0
YL44	NO HIT	0	0
YL45	NO HIT	0	0
YL58	NO HIT	0	0

Table C.2: Number of database matches at genus level for full-length reference sequences against the Silva NR99 115 database using 100% identity.

	Taxon	MALT	USEARCH
ASF361	Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus	44	42
ASF457	Deferribacteres;Deferribacteres;Deferribacterales;Deferribacteraceae;Mucispirillum	5	5
ASF519	Bacteroidetes;Bacteroidia;Bacteroidales;Porphyromonadaceae;Parabacteroides	11	9
ISOL46	Firmicutes;Erysipelotrichia;Erysipelotrichales;Erysipelotrichaceae;IncertaeSedis	6	6
ISOL48	Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides	8	8
KB1	Firmicutes;Bacilli;Lactobacillales;Enterococcaceae;Enterococcus	107	108
YL2	Actinobacteria;Actinobacteria;Bifidobacteriales;Bifidobacteriaceae;Bifidobacterium	1	1
YL31	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Flavonifractor	9	9
YL32	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;IncertaeSedis 0	50	50
YL44	Verrucomicrobia;Verrucomicrobiae;Verrucomicrobiales;Verrucomicrobiaceae;Akkermansia	43	42
YL45	Proteobacteria;Betaproteobacteria;Burkholderiales;Alcaligenaceae;Parasutterella	3	4
YL58	NO HIT	0	0

Table C.3: Number of database matches at genus level for full-length reference sequences against the Silva NR99 115 database using 99% identity.

	Taxon	MALT	USEARCH
ASF361	Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus	73	80
ASF457	Deferribacteres;Deferribacteres;Deferribacterales;Deferribacteraceae;Mucispirillum	6	6
ASF519	Bacteroidetes;Bacteroidia;Bacteroidales;Porphyromonadaceae;Parabacteroides	21	21
ISOL46	Firmicutes;Erysipelotrichia;Erysipelotrichales;Erysipelotrichaceae;IncertaeSedis	41	37
ISOL48	Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides	11	12
KB1	Firmicutes;Bacilli;Lactobacillales;Enterococcaceae;Enterococcus	160	161
YL2	Actinobacteria;Actinobacteria;Bifidobacteriales;Bifidobacteriaceae;Bifidobacterium	7	7
YL31	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Flavonifractor	39	36
YL32	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;IncertaeSedis	89	90
YL44	Verrucomicrobia;Verrucomicrobiae;Verrucomicrobiales;Verrucomicrobiaceae;Akkermansia	167	165
YL45	Proteobacteria;Betaproteobacteria;Burkholderiales;Alcaligenaceae;Parasutterella	4	5
YL58	NO HIT	0	0

Table C.4: Number of database matches at genus level for full-length reference sequences against the Silva NR99 115 database using 98% identity.

	Taxon	MALT	USEARCH
ASF361	Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus	95	117
ASF457	Deferribacteres;Deferribacteres;Deferribacterales;Deferribacteraceae;Mucispirillum	6	7
ASF519	Bacteroidetes;Bacteroidia;Bacteroidales;Porphyromonadaceae;Parabacteroides	31	31
ISOL46	Firmicutes;Erysipelotrichia;Erysipelotrichales;Erysipelotrichaceae;IncertaeSedis	68	80
ISOL48	Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides	25	26
KB1	Firmicutes;Bacilli;Lactobacillales;Enterococcaceae;Enterococcus	195	204
YL2	Actinobacteria;Actinobacteria;Bifidobacteriales;Bifidobacteriaceae;Bifidobacterium	49	33
YL31	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Flavonifractor	49	51
YL32	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;IncertaeSedis	263	264
YL44	Verrucomicrobia;Verrucomicrobiae;Verrucomicrobiales;Verrucomicrobiaceae;Akkermansia	245	266
YL45	Proteobacteria;Betaproteobacteria;Burkholderiales;Alcaligenaceae;Parasutterella	6	6
YL58	NO HIT	0	0

Table C.5: Number of database matches at genus level for full-length reference sequences against the Silva NR99 115 database using 97% identity.

	Taxon	MALT	USEARCH
ASF361	Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus	110	138
ASF457	Deferribacteres;Deferribacteres;Deferribacterales;Deferribacteraceae;Mucispirillum	10	8
ASF519	Bacteroidetes;Bacteroidia;Bacteroidales;Porphyromonadaceae;Parabacteroides	36	36
ISOL46	Firmicutes;Erysipelotrichia;Erysipelotrichales;Erysipelotrichaceae;IncertaeSedis	75	98
ISOL48	Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides	187	183
KB1	Firmicutes;Bacilli;Lactobacillales;Enterococcaceae;Enterococcus	347	313
	Firmicutes;Bacilli;Bacillales;Staphylococcaceae;Staphylococcus	1	1
	Firmicutes;Bacilli;Lactobacillales;Enterococcaceae;Melissococcus	1	0
	Firmicutes;Bacilli;Bacillales;Bacillaceae;Bacillus	1	1
	Firmicutes;Bacilli;Lactobacillales;Carnobacteriaceae;Carnobacterium	1	2
YL2	Actinobacteria;Actinobacteria;Bifidobacteriales;Bifidobacteriaceae;Bifidobacterium	53	54
YL31	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Flavonifractor	58	61
	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Pseudoflavonifractor	7	5
YL32	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;IncertaeSedis	352	345
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;uncultured	2	1
YL44	Verrucomicrobia;Verrucomicrobiae;Verrucomicrobiales;Verrucomicrobiaceae;Akkermansia	264	196
YL45	Proteobacteria;Betaproteobacteria;Burkholderiales;Alcaligenaceae;Parasutterella	6	6
YL58	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Blautia	0	1

Table C.6: Number of database matches at genus level for full-length reference sequences against the Silva NR99 115 database using 96% identity.

	Taxon
ASF361	Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus
ASF457	Deferribacteres;Deferribacteres;Deferribacterales;Deferribacteraceae;Mucispirillum
ASF519	Bacteroidetes;Bacteroidia;Bacteroidales;Porphyromonadaceae;Parabacteroides
Isol46	Firmicutes;Erysipelotrichi;Erysipelotrichales;Erysipelotrichaceae;Eubacterium
Isol48	Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides
KB1	Firmicutes;Bacilli;Lactobacillales;Enterococcaceae;Enterococcus
YL2	Actinobacteria;Actinobacteria;Bifidobacteriales;Bifidobacteriaceae;Bifidobacterium
YL31	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Oscillospira
YL32	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae
YL44	Verrucomicrobia;Verrucomicrobiae;Verrucomicrobiales;Verrucomicrobiaceae;Akkermansia
YL45	Proteobacteria;Betaproteobacteria;Burkholderiales;Alcaligenaceae;Sutterella
YL58	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Blautia

Table C.7: Taxonomic assignment of full length reference sequences using QIIME.

C.1.3 Taxonomic Assignment of Trimmed Reference Sequences

	Taxon	Confidence
ASF361	Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus	0.98
ASF457	Deferribacteres;Deferribacteres;Deferribacterales;Deferribacteraceae;Mucispirillum	1.00
ASF519	Bacteroidetes;Bacteroidia;Bacteroidales;Porphyromonadaceae;Parabacteroides	1.00
Isol46	Firmicutes;Erysipelotrichia;Erysipelotrichales;Erysipelotrichaceae;Erysipelotrichaceae;IncertaeSedis	1.00
Isol48	Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides	1.00
KB1	Firmicutes;Bacilli;Lactobacillales;Enterococcaceae;Enterococcus	1.00
YL2	Actinobacteria;Actinobacteria;Actinobacteridae;Bifidobacteriales;Bifidobacteriaceae;Bifidobacterium	1.00
YL31	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Flavonifractor	1.00
YL32	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Clostridium XlVa	1.00
YL44	Verrucomicrobia;Verrucomicrobiae;Verrucomicrobiales;Verrucomicrobiaceae;Akkermansia	1.00
YL45	Proteobacteria;Betaproteobacteria;Burkholderiales;Sutterellaceae;Parasutterella	0.91
YL58	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Blautia	1.00

Table C.8: Taxonomic assignment of shortened reference sequences using the rdp classifier.

	Taxon	MALT	USEARCH
ASF361	Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus	78	78
ASF457	Deferribacteres;Deferribacteres;Deferribacterales;Deferribacteraceae;Mucispirillum	6	6
ASF519	Bacteroidetes;Bacteroidia;Bacteroidales;Porphyromonadaceae;Parabacteroides	15	15
ISOL46	Firmicutes;Erysipelotrichia;Erysipelotrichales;Erysipelotrichaceae;IncertaeSedis	40	40
ISOL48	Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides	11	11
KB1	Firmicutes;Bacilli;Lactobacillales;Enterococcaceae;Enterococcus	131	131
	Firmicutes;Bacilli;Bacillales;Planococcaceae;Planomicrobium	3	3
	Firmicutes;Bacilli;Bacillales;Staphylococcaceae;Staphylococcus	1	1
	Firmicutes;Bacilli;Bacillales;Bacillaceae;Bacillus	2	2
YL2	Actinobacteria;Actinobacteria;Bifidobacteriales;Bifidobacteriaceae;Bifidobacterium	5	5
YL31	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Flavonifractor	30	30
	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;IncertaeSedis	1	1
YL31	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;IncertaeSedis	211	212
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;uncultured	2	4
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Blautia	1	1
	Bacteroidetes;Bacteroidia;Bacteroidales;S24-7	2	1
	Chloroflexi;Anaerolineae;Anaerolineales;Anaerolineaceae;uncultured	1	1
YL44	Verrucomicrobia;Verrucomicrobiae;Verrucomicrobiales;Verrucomicrobiaceae;Akkermansia	113	113
YL45	Proteobacteria;Betaproteobacteria;Burkholderiales;Alcaligenaceae;Parasutterella	1	2
YL58	NO HIT	0	0

Table C.9: Number of database matches at genus level for shortened reference sequences against the Silva NR99 115 database using 100% identity.

	Taxon	MALT	USEARCH
ASF361	Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus	144	143
	Firmicutes;Bacilli;Lactobacillales;Streptococcaceae;Streptococcus	2	2
	Firmicutes;Erysipelotrichia;Erysipelotrichales;Erysipelotrichaceae;Allobaculum	1	1
ASF457	Deferribacteres;Deferribacteres;Deferribacterales;Deferribacteraceae;Mucispirillum	6	6
ASF519	Bacteroidetes;Bacteroidia;Bacteroidales;Porphyromonadaceae;Parabacteroides	26	25
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;IncertaeSedis	1	1
ISOL46	Firmicutes;Erysipelotrichia;Erysipelotrichales;Erysipelotrichaceae;IncertaeSedis	74	70
ISOL48	Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides	13	13
KB1	Firmicutes;Bacilli;Lactobacillales;Enterococcaceae;Enterococcus	165	162
	Firmicutes;Bacilli;Bacillales;Planococcaceae;Planomicrobium	4	4
	Firmicutes;Bacilli;Bacillales;Staphylococcaceae;Staphylococcus	3	2
	Firmicutes;Clostridia;Clostridiales;Clostridiaceae;Clostridium	1	1
	Firmicutes;Bacilli;Bacillales;Bacillaceae;Bacillus	2	3
YL2	Actinobacteria;Actinobacteria;Bifidobacteriales;Bifidobacteriaceae;Bifidobacterium	49	8
YL31	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Flavonifractor	52	58
	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;uncultured	2	1
	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;IncertaeSedis	2	1
YL32	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;IncertaeSedis	312	316
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;uncultured	5	3
	Bacteroidetes;Bacteroidia;Bacteroidales;S24-7	3	3
	Chloroflexi;Anaerolineae;Anaerolineales;Anaerolineaceae;uncultured	1	1
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Blautia	1	1
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Roseburia	3	2
YL44	Verrucomicrobia;Verrucomicrobiae;Verrucomicrobiales;Verrucomicrobiaceae;Akkermansia	279	277
YL45	Proteobacteria;Betaproteobacteria;Burkholderiales;Alcaligenaceae;Parasutterella	6	6
YL58	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Blautia	287	287
	Firmicutes;Clostridia;Clostridiales;Christensenellaceae;uncultured	1	1
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Roseburia	1	1
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Dorea	1	1
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;IncertaeSedis	6	6
	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;IncertaeSedis	1	1
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Pseudobutyrivibrio	8	8
	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Ruminococcus	1	1
	Firmicutes;Clostridia;Clostridiales;Peptostreptococcaceae;IncertaeSedis	1	1
	Bacteroidetes;Bacteroidia;Bacteroidales;S24-7	1	1

Table C.10: Number of database matches at genus level for shortened reference sequences against the Silva NR99 115 database using 99% identity.

	Taxon	MALT	USEARCH
ASF361	Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus	160	158
	Firmicutes;Bacilli;Lactobacillales;Streptococcaceae;Streptococcus	2	2
	Firmicutes;Erysipelotrichia;Erysipelotrichales;Erysipelotrichaceae;Allobaculum	1	1
ASF457	Deferribacteres;Deferribacteres;Deferribacterales;Deferribacteraceae;Mucispirillum	7	7
ASF519	Bacteroidetes;Bacteroidia;Bacteroidales;Porphyromonadaceae;Parabacteroides	28	28
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;IncertaeSedis	1	1
ISOL46	Firmicutes;Erysipelotrichia;Erysipelotrichales;Erysipelotrichaceae;IncertaeSedis	86	81
ISOL48	Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides	13	13
KB1	Firmicutes;Bacilli;Lactobacillales;Enterococcaceae;Enterococcus	199	170
	Firmicutes;Bacilli;Bacillales;Planococcaceae;Planomicrobium	5	5
	Firmicutes;Bacilli;Bacillales;Staphylococcaceae;Staphylococcus	3	3
	Firmicutes;Clostridia;Clostridiales;Clostridiaceae;Clostridium	1	2
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Epulopiscium	1	1
	Firmicutes;Bacilli;Bacillales;Bacillaceae;Bacillus	2	2
YL2	Actinobacteria;Actinobacteria;Bifidobacteriales;Bifidobacteriaceae;Bifidobacterium	64	10
YL31	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Flavonifractor	54	54
	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;IncertaeSedis	1	1
	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;uncultured	7	4
YL32	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;IncertaeSedis	374	478
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;uncultured	79	108
	Bacteroidetes;Bacteroidia;Bacteroidales;S24-7	3	3
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Blautia	1	3
	Chloroflexi;Anaerolineae;Anaerolineales;Anaerolineaceae;uncultured	1	1
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Anaerosporobacter	1	1
	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Ruminococcus	1	1
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Pseudobutyrivibrio	1	2
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Roseburia	2	2
YL44	Verrucomicrobia;Verrucomicrobiae;Verrucomicrobiales;Verrucomicrobiaceae;Akkermansia	354	345
YL45	Proteobacteria;Betaproteobacteria;Burkholderiales;Alcaligenaceae;Parasutterella	6	6
YL58	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Blautia	352	1,536
	Firmicutes;Clostridia;Clostridiales;Christensenellaceae;uncultured	1	1
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Roseburia	1	10
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Dorea	1	1
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;IncertaeSedis	11	52
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Pseudobutyrivibrio	9	27
	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;IncertaeSedis	1	1
	Firmicutes;Clostridia;Clostridiales;Peptostreptococcaceae;IncertaeSedis	1	1
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;uncultured	20	47
	Bacteroidetes;Bacteroidia;Bacteroidales;S24-7	1	1
	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Coprococcus;	0	1
	Actinobacteria;Coriobacteriia;Coriobacteriales;Coriobacteriaceae;Collinsella	0	2
	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Ruminococcus	1	5

Table C.11: Number of database matches at genus level for shortened reference sequences against the Silva NR99 115 database using 98% identity.

	Taxon
ASF361	Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus
ASF457	Deferribacteres;Deferribacteres;Deferribacterales;Deferribacteraceae;Mucispirillum
ASF519	Bacteroidetes;Bacteroidia;Bacteroidales;Porphyromonadaceae;Parabacteroides
Isol46	Firmicutes;Erysipelotrichi;Erysipelotrichales;Erysipelotrichaceae;Eubacterium
Isol48	Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides
KB1	Firmicutes;Bacilli;Lactobacillales;Enterococcaceae;Other
YL2	Actinobacteria;Actinobacteria;Bifidobacteriales;Bifidobacteriaceae;Bifidobacterium
YL31	Firmicutes;Clostridia;Clostridiales;Ruminococcaceae;Oscillospira
Yl32	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae
YL44	Verrucomicrobia;Verrucomicrobiae;Verrucomicrobiales;Verrucomicrobiaceae;Akkermansia
YL45	Proteobacteria;Betaproteobacteria;Burkholderiales;Alcaligenaceae;Sutterella
YL58	Firmicutes;Clostridia;Clostridiales;Lachnospiraceae;Blautia

Table C.12: Taxonomic assignment of shortened reference sequences using QIIME.

C.1.4 Alignment of 454 Reads against Reference Database

% Identity	Assigned	% Assigned	Not Assigned	% Not Assigned
100	648	0.19	348,991	99.81
99	336,930	96.37	12,709	3.63
98	345,905	98.93	3,734	1.07
97	346,628	99.14	3,011	0.86
96	346,740	99.17	2,899	0.83
95	346,818	99.19	2,821	0.81
94	347,101	99.27	2,538	0.73
93	347,130	99.28	2,509	0.72
92	347,144	99.29	2,495	0.71
91	347,164	99.29	2,475	0.71
90	347,199	99.30	2,440	0.70

Table C.13: Percentage of reads that successfully align at certain percent identity using MALT in semiglobal mode with a database created from the twelve input sequences.

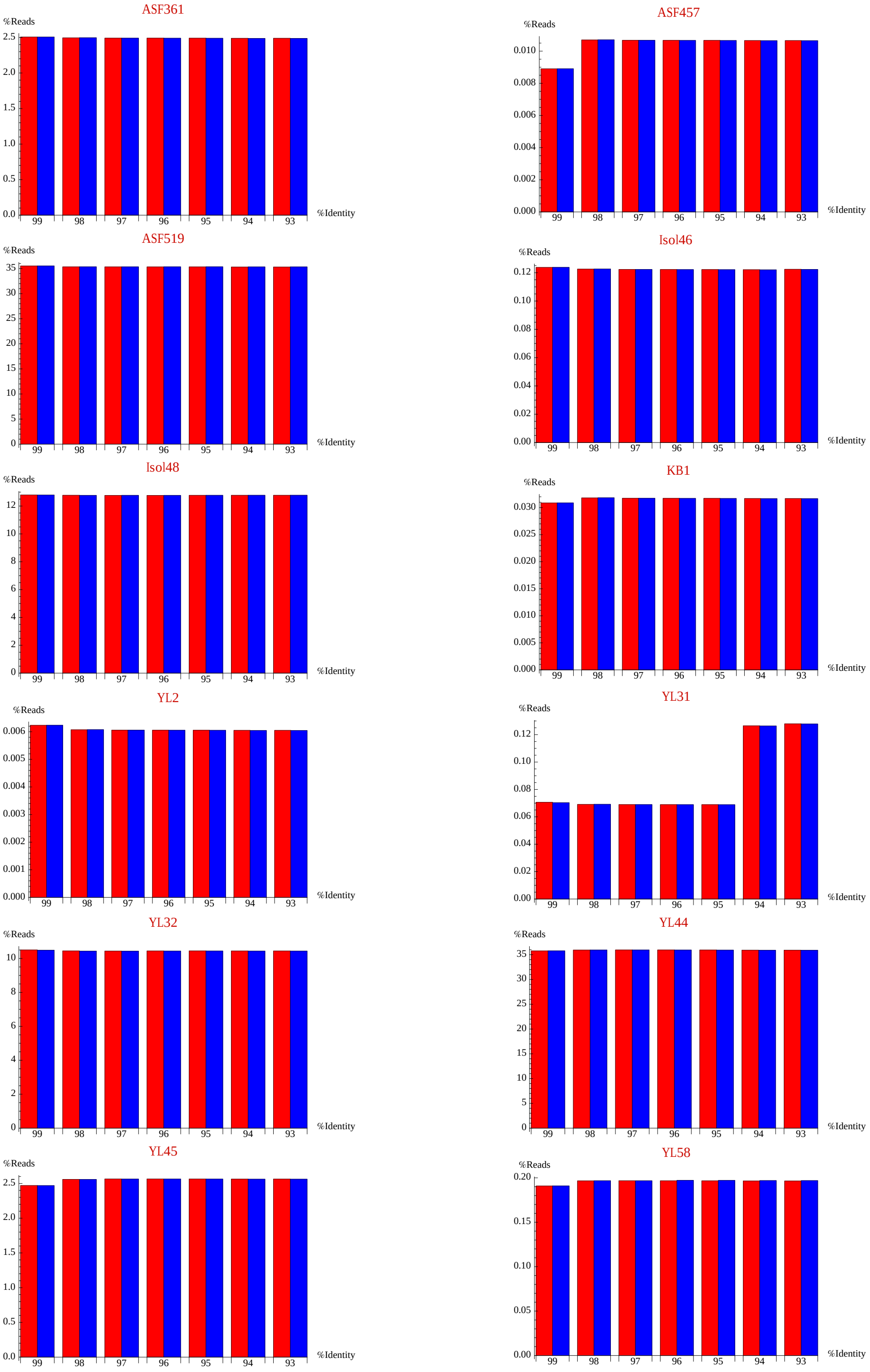


Figure C.2: Percent of reads assigned to reference sequences at different percent identities using ***MALT*** *and* ***usearch***

Bibliography

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403 – 410.

Anderson, J., McRee, J., and Wilson, R. (2010). *Effective UI: The art of building great user experience in software*. O'Reilly Media, Inc.

Andrews, S. (2010). FastQC: A quality control tool for high throughput sequence data. `http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/`. [Online; accessed 31-May-2014].

Armougom, F. and Raoult, D. (2009). Exploring microbial diversity using 16S rRNA high-throughput methods. *Journal of Computer Science & Systems Biology*, 2(1).

Arumugam, M., Raes, J., Pelletier, E., Le Paslier, D., Yamada, T., Mende, D. R., Fernandes, G. R., Tap, J., Bruls, T., Batto, J.-M., Bertalan, M., Borruel, N., Casellas, F., Fernandez, L., Gautier, L., Hansen, T., Hattori, M., Hayashi, T., Kleerebezem, M., Kurokawa, K., Leclerc, M., Levenez, F., Manichanh, C., Nielsen, H. B., Nielsen, T., Pons, N., Poulain, J., Qin, J., Sicheritz-Ponten, T., Tims, S., Torrents, D., Ugarte, E., Zoetendal, E. G., Wang, J., Guarner, F., Pedersen, O., de Vos, W. M., Brunak, S., Dore, J., Weissenbach, J., Ehrlich, S. D., and Bork, P. (2011). Enterotypes of the human gut microbiome. *Nature*, 473(7346):174–180.

Balzer, S., Malde, K., Lanzn, A., Sharma, A., and Jonassen, I. (2010). Characteristics of 454 pyrosequencing data enabling realistic simulation with flowsim. *Bioinformatics*, 26(18):i420–i425.

Bennett, S. (2004). Solexa ltd. *Pharmacogenomics*, 5(4):433–438.

Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., and Wheeler, D. L. (2006). GenBank. *Nucleic Acids Research*, 34(Database issue):D16–D20.

Berg, R. D. (1996). The indigenous gastrointestinal microflora. *Trends in Microbiology*, 4(11):430–435.

Bonder, M. J., Abeln, S., Zaura, E., and Brandt, B. W. (2012). Comparing clustering and pre-processing in taxonomy analysis. *Bioinformatics*, 28(22):2891–2897.

Bray, J. R. and Curtis, J. T. (1957). An ordination of the upland forest communities of southern Wisconsin. *Ecological Monographs*, 27(4):325–349.

Brown, S. (2013). Software architecture for developers. *Coding the Architecture.*

Buchfink, B., Xie, C., and Huson, D. H. (2014). Fast and sensitive protein alignment using DIAMOND. `http://ab.inf.uni-tuebingen.de/software/diamond/`. [Online; accessed 05-May-2014].

Cai, Y. and Sun, Y. (2011). ESPRIT-Tree: hierarchical clustering analysis of millions of 16S rRNA pyrosequences in quasilinear computational time. *Nucleic Acids Research.*

Caporaso, J. G., Bittinger, K., Bushman, F. D., DeSantis, T. Z., Andersen, G. L., and Knight, R. (2010a). PyNAST: a flexible tool for aligning sequences to a template alignment. *Bioinformatics*, 26(2):266–267.

Caporaso, J. G., Kuczynski, J., Stombaugh, J., Bittinger, K., Bushman, F. D., Costello, E. K., Fierer, N., Pena, A. G., Goodrich, J. K., Gordon, J. I., Huttley, G. A., Kelley, S. T., Knights, D., Koenig, J. E., Ley, R. E., Lozupone, C. A., McDonald, D., Muegge, B. D., Pirrung, M., Reeder, J., Sevinsky, J. R., Turnbaugh, P. J., Walters, W. A., Widmann, J., Yatsunenko, T., Zaneveld, J., and Knight, R. (2010b). QIIME allows analysis of high-throughput community sequencing data. *Nature Methods*, 7(5):335–336.

Chen, W., Zhang, C. K., Cheng, Y., Zhang, S., and Zhao, H. (2013). A comparison of methods for clustering 16S rRNA sequences into OTUs. *PLoS ONE*, 8(8):e70837.

Clayton, R. A., Sutton, G., Hinkle, P. S., Bult, C., and Fields, C. (1995). Intraspecific variation in small-subunit rRNA sequences in GenBank: Why single sequences may not adequately represent prokaryotic taxa. *International Journal of Systematic Bacteriology*, 45(3):595–599.

Cline, J., Braman, J. C., and Hogrefe, H. H. (1996). PCR fidelity of pfu DNA polymerase and other thermostable DNA polymerases. *Nucleic Acids Research*, 24(18):3546–3551.

Coenye, T. and Vandamme, P. (2003). Intragenomic heterogeneity between multiple 16S ribosomal RNA operons in sequenced bacterial genomes. *FEMS Microbiology Letters*, 228(1):45–49.

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., and Weerawarana, S. (2002). Unraveling the web services web: an introduction to SOAP, WSDL, and UDDI. *Internet Computing, IEEE*, 6(2):86–93.

Dixon, P. (2003). VEGAN, a package of R functions for community ecology. *Journal of Vegetation Science*, 14(6):927–930.

Dray, S. and Dufour, A.-B. (2007). The ade4 package: implementing the duality diagram for ecologists. *Journal of Statistical Software*, 22(4):1–20.

Dressman, D., Yan, H., Traverso, G., Kinzler, K. W., and Vogelstein, B. (2003). Transforming single DNA molecules into fluorescent magnetic particles for detection and enumeration of genetic variations. *Proceedings of the National Academy of Sciences*, 100(15):8817–8822.

Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797.

Edgar, R. C. (2010). Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19):2460–2461.

Edgar, R. C. (2013). UPARSE: highly accurate OTU sequences from microbial amplicon reads. *Nature Methods*, 10(10):996–998.

Edgar, R. C. (2014). Average Q score. `http://www.drive5.com/usearch/manual/avgq.html`. [Online; accessed 31-May-2014].

Edgar, R. C., Haas, B. J., Clemente, J. C., Quince, C., and Knight, R. (2011). UCHIME improves sensitivity and speed of chimera detection. *Bioinformatics*, 27(16):2194–2200.

Ewing, B. and Green, P. (1998). Base-calling of automated sequencer traces using phred. ii. error probabilities. *Genome Research*, 8(3):186–194.

Ewing, B., Hillier, L., Wendl, M. C., and Green, P. (1998). Base-calling of automated sequencer traces using phred. i. accuracy assessment. *Genome Research*, 8(3):175–185.

Federhen, S. (2012). The NCBI taxonomy database. *Nucleic Acids Research*, 40(Database issue):D136–D143.

Fielding, R. T. and Taylor, R. N. (2002). Principled design of the modern web architecture. *ACM Transactions on Internet Technology (TOIT)*, 2(2):115–150.

Gordon, A. and Hannon, G. (2010). Fastx-toolkit. *FASTQ/A short-reads pre-processing tools (unpublished) http://hannonlab.cshl.edu/fastx_toolkit*.

Gronbach, K., Flade, I., Holst, O., Lindner, B., Ruscheweyh, H.-J., Wittmann, A., Menz, S., Schwiertz, A., Adam, P., Stecher, B., Josenhans, C., Suerbaum, S., Gruber, A. D., Kulik, A., Huson, D., Autenrieth, I. B., and Frick, J.-S. (2014). Endotoxicity of lipopolysaccharide as a determinant of T-Cell mediated colitis induction in mice. *Gastroenterology*, 146(3):765 – 775.

Haas, B. J., Gevers, D., Earl, A. M., Feldgarden, M., Ward, D. V., Giannoukos, G., Ciulla, D., Tabbaa, D., Highlander, S. K., Sodergren, E., Methé, B., DeSantis, T. Z., The Human Microbiome Consortium, Petrosino, J. F., Knight, R., and Birren, B. W. (2011). Chimeric 16S rRNA sequence formation and detection in sanger and 454-pyrosequenced PCR amplicons. *Genome Research*, 21(3):494–504.

Hao, X., Jiang, R., and Chen, T. (2011). Clustering 16S rRNA for OTU prediction: a method of unsupervised bayesian clustering. *Bioinformatics*.

Hartmann, M., Howes, C. G., Abarenkov, K., Mohn, W. W., and Nilsson, R. H. (2010). V-Xtractor: An open-source, high-throughput software tool to identify and extract hypervariable regions of small subunit (16S/18S) ribosomal RNA gene sequences. *Journal of Microbiological Methods*, 83(2):250 – 253.

Hess, M., Sczyrba, A., Egan, R., Kim, T.-W., Chokhawala, H., Schroth, G., Luo, S., Clark, D. S., Chen, F., Zhang, T., Mackie, R. I., Pennacchio, L. A., Tringe, S. G., Visel, A., Woyke, T., Wang, Z., and Rubin, E. M. (2011). Metagenomic discovery of biomass-degrading genes and genomes from cow rumen. *Science*, 331(6016):463–467.

Huang, Y., Niu, B., Gao, Y., Fu, L., and Li, W. (2010). CD-HIT suite: a web server for clustering and comparing biological sequences. *Bioinformatics*, 26(5):680–682.

Huber, T., Faulkner, G., and Hugenholtz, P. (2004). Bellerophon: a program to detect chimeric sequences in multiple sequence alignments. *Bioinformatics*, 20(14):2317–2319.

Hughes, J. B., Hellmann, J. J., Ricketts, T. H., and Bohannan, B. J. M. (2001). Counting the uncountable: Statistical approaches to estimating microbial diversity. *Applied and Environmental Microbiology*, 67(10):4399–4406.

Huse, S. M., Dethlefsen, L., Huber, J. A., Welch, D. M., Relman, D. A., and Sogin, M. L. (2008). Exploring microbial diversity and taxonomy using SSU rRNA hypervariable tag sequencing. *PLoS Genetics*, 4(11):e1000255.

Huse, S. M., Huber, J. A., Morrison, H. G., Sogin, M. L., and Welch, D. M. (2007). Accuracy and quality of massively parallel DNA pyrosequencing. *Genome Biology*, 8(7):R143–R143.

Huse, S. M., Welch, D. M., Morrison, H. G., and Sogin, M. L. (2010). Ironing out the wrinkles in the rare biosphere through improved OTU clustering. *Environmental Microbiology*, 12(7):1889–1898.

Huson, D., Richter, D., Mitra, S., Auch, A., and Schuster, S. (2009). Methods for comparative metagenomics. *BMC Bioinformatics*, 10(Suppl 1):S12.

Huson, D. H. (2014a). MALT. `http://ab.inf.uni-tuebingen.de/software/malt/`. [Online; accessed 15-June-2014].

Huson, D. H. (2014b). MEGAN5. `http://ab.inf.uni-tuebingen.de/software/megan5/`. [Online; accessed 05-May-2014].

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

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

Huson, D. H. and Xie, C. (2014). A poor man's BLASTX - high-throughput metagenomic protein database search using PAUDA. *Bioinformatics*, 30(1):38–39.

Illumina (2012). Metagenomics research review. `https://www.science.smith.edu/cmbs/documents/metagenomicsresearchreview.pdf`. [Online; accessed 16-June-2014].

Johnson, R. (2004). *Expert one-on-one J2EE Design and Development*. John Wiley & Sons.

Kanehisa, M. and Goto, S. (2000). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30.

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J.-M., and Irwin, J. (1997). *Aspect-oriented programming*. Springer.

Kuczynski, J., Liu, Z., Lozupone, C., McDonald, D., Fierer, N., and Knight, R. (2010). Microbial community resemblance methods differ in their ability to detect biologically relevant patterns. *Nature Methods*, 7(10):813–819.

Kunin, V., Engelbrektson, A., Ochman, H., and Hugenholtz, P. (2010). Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates. *Environmental Microbiology*, 12(1):118–123.

Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357–359.

Ley, R. E., Turnbaugh, P. J., Klein, S., and Gordon, J. I. (2006). Microbial ecology: Human gut microbes associated with obesity. *Nature*, 444(7122):1022–1023.

Lozupone, C., Hamady, M., and Knight, R. (2006). UniFrac - an online tool for comparing microbial community diversity in a phylogenetic context. *BMC Bioinformatics*, 7:371–371.

Lynch, M. D. J., Bartram, A. K., and Neufeld, J. D. (2012). Targeted recovery of novel phylogenetic diversity from next-generation sequence data. *ISME J*, 6(11):2067–2077.

Mackelprang, R., Waldrop, M. P., DeAngelis, K. M., David, M. M., Chavarria, K. L., Blazewicz, S. J., Rubin, E. M., and Jansson, J. K. (2011). Metagenomic analysis of a permafrost microbial community reveals a rapid response to thaw. *Nature*, 480(7377):368–371.

Marco, D. (2010). *Metagenomics: Theory, methods and applications*. Horizon Scientific Press.

Mardis, E. R. (2008). Next-generation DNA sequencing methods. *Annual Review of Genomics and Human Genetics*, 9(1):387–402. PMID: 18576944.

Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y.-J., Chen, Z., Dewell, S. B., Du, L., Fierro, J. M., Gomes, X. V., Godwin, B. C., He, W., Helgesen, S., Ho, C. H., Irzyk, G. P., Jando, S. C., Alenquer, M. L. I., Jarvie, T. P., Jirage, K. B., Kim, J.-B., Knight, J. R., Lanza, J. R., Leamon, J. H., Lefkowitz, S. M., Lei, M., Li, J., Lohman, K. L., Lu, H., Makhijani, V. B., McDade, K. E., McKenna, M. P., Myers, E. W., Nickerson, E., Nobile, J. R., Plant, R., Puc, B. P., Ronan, M. T., Roth, G. T., Sarkis, G. J., Simons, J. F., Simpson, J. W., Srinivasan, M., Tartaro, K. R., Tomasz, A., Vogt, K. A., Volkmer, G. A., Wang, S. H., Wang, Y., Weiner, M. P., Yu, P., Begley, R. F., and Rothberg, J. M. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380.

McDonald, D., Price, M. N., Goodrich, J., Nawrocki, E. P., DeSantis, T. Z., Probst, A., Andersen, G. L., Knight, R., and Hugenholtz, P. (2012). An improved greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *ISME J*, 6(3):610–618.

Meyer, F., Paarmann, D., D'Souza, M., Olson, R., Glass, E., Kubal, M., Paczian, T., Rodriguez, A., Stevens, R., Wilke, A., Wilkening, J., and Edwards, R. (2008). The metagenomics RAST server – a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics*, 9:386–386.

Miquel (2014). Middleware remoting protocol migration. `http://blog.nominet.org.uk/tech/2007/03/13/middleware-remoting-protocol-migration/`. [Online; accessed 18-May-2014].

Mitra, S., Forster-Fromme, K., Damms-Machado, A., Scheurenbrand, T., Biskup, S., Huson, D., and Bischoff, S. (2013). Analysis of the intestinal microbiota using SOLiD 16S rRNA gene sequencing and SOLiD shotgun sequencing. *BMC Genomics*, 14(Suppl 5):S16.

Mitra, S., Gilbert, J. A., Field, D., and Huson, D. H. (2010). Comparison of multiple metagenomes using phylogenetic networks based on ecological indices. *ISME J*, 4(10):1236–1242.

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

Mitra, S., Rupek, P., Richter, D., Urich, T., Gilbert, J., Meyer, F., Wilke, A., and Huson, D. (2011a). Functional analysis of metagenomes and metatranscriptomes using SEED and KEGG. *BMC Bioinformatics*, 12(Suppl 1):S21.

Mitra, S., Stark, M., and Huson, D. (2011b). Analysis of 16S rRNA environmental sequences using MEGAN. *BMC Genomics*, 12(Suppl 3):S17.

Muller, J., Szklarczyk, D., Julien, P., Letunic, I., Roth, A., Kuhn, M., Powell, S., von Mering, C., Doerks, T., Jensen, L. J., and Bork, P. (2010). eggNOG v2.0: extending the evolutionary genealogy of genes with enhanced non-supervised orthologous groups, species and functional annotations. *Nucleic Acids Research*, 38(Database issue):D190–D195.

Nyrén, P. (2001). Method of sequencing DNA based on the detection of the release of pyrophosphate and enzymatic nucleotide degradation.

Oracle (2014). Learn about Java technology. `http://www.java.com/en/about/`. [Online; accessed 15-May-2014].

Overbeek, R., Begley, T., Butler, R. M., Choudhuri, J. V., Chuang, H.-Y., Cohoon, M., de Crcy-Lagard, V., Diaz, N., Disz, T., Edwards, R., Fonstein, M., Frank, E. D., Gerdes, S., Glass, E. M., Goesmann, A., Hanson, A., Iwata-Reuyl, D., Jensen, R., Jamshidi, N., Krause, L., Kubal, M., Larsen, N., Linke, B., McHardy, A. C., Meyer, F., Neuweger, H., Olsen, G., Olson, R., Osterman, A., Portnoy, V., Pusch, G. D., Rodionov, D. A., Rckert, C., Steiner, J., Stevens, R., Thiele, I., Vassieva, O., Ye, Y., Zagnitko, O., and Vonstein, V. (2005). The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. *Nucleic Acids Research*, 33(17):5691–5702.

Patel, R. K. and Jain, M. (2012). NGS QC Toolkit: A toolkit for quality control of next generation sequencing data. *PLoS ONE*, 7(2):e30619.

Peterson, J., Garges, S., Giovanni, M., McInnes, P., Wang, L., Schloss, J. A., Bonazzi, V., McEwen, J. E., Wetterstrand, K. A., Deal, C., Baker, C. C., Di Francesco, V., Howcroft, T. K., Karp, R. W., Lunsford, R. D., Wellington, C. R., Belachew, T., Wright, M., Giblin, C., David, H., Mills, M., Salomon, R., Mullins, C., Akolkar, B., Begg, L., Davis, C., Grandison, L., Humble, M., Khalsa, J., Little, A. R., Peavy, H., Pontzer, C., Portnoy, M., Sayre, M. H., Starke-Reed, P., Zakhari, S., Read, J., Watson, B., and Guyer, M. (2009). The NIH human microbiome project. *Genome Research*, 19(12):2317–2323.

Poinar, H. N., Schwarz, C., Qi, J., Shapiro, B., MacPhee, R. D. E., Buigues, B., Tikhonov, A., Huson, D. H., Tomsho, L. P., Auch, A., Rampp, M., Miller, W., and Schuster, S. C. (2006). Metagenomics to Paleogenomics: Large-scale sequencing of mammoth DNA. *Science*, 311(5759):392–394.

Price, M. N., Dehal, P. S., and Arkin, A. P. (2009). FastTree: Computing large minimum evolution trees with profiles instead of a distance matrix. *Molecular Biology and Evolution*, 26(7):1641–1650.

Price, M. N., Dehal, P. S., and Arkin, A. P. (2010). FastTree 2 –approximately maximum-likelihood trees for large alignments. *PLoS ONE*, 5(3):e9490.

Pruitt, K. D., Tatusova, T., and Maglott, D. R. (2007). NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research*, 35(suppl 1):D61–D65.

QIIME (2014). Analysis of shotgun sequencing data. `http://qiime.org/tutorials/shotgun_analysis.html`. [Online; accessed 28-June-2014].

Qin, J., Li, R., Raes, J., Arumugam, M., Burgdorf, K. S., Manichanh, C., Nielsen, T., Pons, N., Levenez, F., Yamada, T., Mende, D. R., Li, J., Xu, J., Li, S., Li, D., Cao, J., Wang, B., Liang, H., Zheng, H., Xie, Y., Tap, J., Lepage, P., Bertalan, M., Batto, J.-M., Hansen, T., Le Paslier, D., Linneberg, A., Nielsen, H. B., Pelletier, E., Renault, P., Sicheritz-Ponten, T., Turner, K., Zhu, H., Yu, C., Li, S., Jian, M., Zhou, Y., Li, Y., Zhang, X., Li, S., Qin, N., Yang, H., Wang, J., Brunak, S., Dore, J., Guarner, F., Kristiansen, K., Pedersen, O., Parkhill, J., Weissenbach, J., Bork, P., Ehrlich, S. D., and Wang, J. (2010). A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, 464(7285):59–65.

Quast, C., Pruesse, E., Yilmaz, P., Gerken, J., Schweer, T., Yarza, P., Peplies, J., and Glöckner, F. O. (2013). The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Research*, 41(Database issue):D590–D596.

Quince, C., Lanzen, A., Curtis, T. P., Davenport, R. J., Hall, N., Head, I. M., Read, L. F., and Sloan, W. T. (2009). Accurate determination of microbial diversity from 454 pyrosequencing data. *Nature Methods*, 6(9):639–641.

Quince, C., Lanzen, A., Davenport, R. J., and Turnbaugh, P. J. (2011). Removing noise from pyrosequenced amplicons. *BMC Bioinformatics*, 12:38–38.

Rappé, M. S. and Giovannoni, S. J. (2003). The uncultured microbial majority. *Annual Reviews in Microbiology*, 57(1):369–394.

Roche Diagnostics GmbH (2014). GS FLX+ System: Sanger-like read lengths - the power of next-gen throughput. `http://454.com/downloads/GSFLXApplicationFlyer_FINALv2.pdf`. "[Online; accessed 30-May-2014]".

Ronaghi, M., Uhlén, M., and Nyrén, P. (1998). A sequencing method based on real-time pyrophosphate. *Science*, 281(5375):363–365.

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

Rothberg, J. M. and Leamon, J. H. (2008). The development and impact of 454 sequencing. *Nature Biotechnology*, 26(10):1117–1124.

Ruscheweyh, H.-J. (2010). Datenbankgestützte Analyse von Metagenomikdaten. Master's thesis, Universität Tübingen.

Schloss, P. D. (2009). A high-throughput DNA sequence aligner for microbial ecology studies. *PLoS ONE*, 4(12):e8230.

Schloss, P. D. (2010). The effects of alignment quality, distance calculation method, sequence filtering, and region on the analysis of 16S rRNA gene-based studies. *PLoS Computational Biology*, 6(7):e1000844.

Schloss, P. D., Gevers, D., and Westcott, S. L. (2011). Reducing the effects of PCR amplification and sequencing artifacts on 16S rRNA-based studies. *PLoS ONE*, 6(12):e27310.

Schloss, P. D. and Westcott, S. L. (2011). Assessing and improving methods used in operational taxonomic unit-based approaches for 16S rRNA gene sequence analysis. *Applied and Environmental Microbiology*, 77(10):3219–3226.

Schloss, P. D., Westcott, S. L., Ryabin, T., Hall, J. R., Hartmann, M., Hollister, E. B., Lesniewski, R. A., Oakley, B. B., Parks, D. H., Robinson, C. J., Sahl, J. W., Stres, B., Thallinger, G. G., Van Horn, D. J., and Weber, C. F. (2009). Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology*, 75(23):7537–7541.

Schmieder, R. and Edwards, R. (2011). Quality control and preprocessing of metagenomic datasets. *Bioinformatics*.

Schuenemann, V. J., Singh, P., Mendum, T. A., Krause-Kyora, B., Jäger, G., Bos, K. I., Herbig, A., Economou, C., Benjak, A., Busso, P., Nebel, A., Boldson, J. L., Kjellström, A., Wu, H., Stewart, G. R., Taylor, G. M., Bauer, P., Lee, O. Y.-C., Wu, H. H., Minnikin, D. E., Besra, G. S., Tucker, K., Roffey, S., Sow, S. O., Cole, S. T., Nieselt, K., and Krause, J. (2013). Genome-wide comparison of medieval and modern Mycobacterium leprae. *Science*, 341(6142):179–183.

Schwaber, K. (1997). Scrum development process. In *Business Object Design and Implementation*, pages 117–134. Springer.

Sciences, . L. (2014). How is genome sequencing done? `http://www.454.com/downloads/news-events/how-genome-sequencing-is-done_FINAL.pdf`. [Online; accessed 1-July-2014].

Segata, N., Izard, J., Waldron, L., Gevers, D., Miropolsky, L., Garrett, W. S., and Huttenhower, C. (2011). Metagenomic biomarker discovery and explanation. *Genome Biology*, 12(6):R60–R60.

Seshadri, R., Kravitz, S. A., Smarr, L., Gilna, P., and Frazier, M. (2007). CAMERA: A community resource for metagenomics. *PLoS Biology*, 5(3):e75.

Sievert, S. M., Hügler, M., Taylor, C. D., and Wirsen, C. O. (2008). Sulfur oxidation at deep-sea hydrothermal vents. In *Microbial Sulfur Metabolism*, pages 238–258. Springer.

Smyth, R., Schlub, T., Grimm, A., Venturi, V., Chopra, A., Mallal, S., Davenport, M., and Mak, J. (2010). Reducing chimera formation during PCR amplification to ensure accurate genotyping. *Gene*, 469(1–2):45 – 51.

Sogin, M. L., Morrison, H. G., Huber, J. A., Welch, D. M., Huse, S. M., Neal, P. R., Arrieta, J. M., and Herndl, G. J. (2006). Microbial diversity in the deep sea and the underexplored “rare biosphere”. *Proceedings of the National Academy of Sciences of the United States of America*, 103(32):12115–12120.

Sonatype (2014). Maven repository statistics. `http://search.maven.org/#stats`. [Online; accessed 15-May-2014].

Sun, Y., Cai, Y., Liu, L., Yu, F., Farrell, M. L., McKendree, W., and Farmerie, W. (2009). ESPRIT: estimating species richness using large collections of 16S rRNA pyrosequences. *Nucleic Acids Research*, 37(10):e76–e76.

Technologies, L. (2014). Life technologies receives FDA 510(k) clearance for diagnostic use of sanger sequencing platform and HLA typing kits. `http://www.reuters.com/article/2013/02/11/ca-life-fda-idUSnPnLA57279+160+PRN20130211`. [Online; accessed 31-May-2014].

The Human Microbiome Project Consortium (2012a). A framework for human microbiome research. *Nature*, 486(7402):215–221.

The Human Microbiome Project Consortium (2012b). Structure, function and diversity of the healthy human microbiome. *Nature*, 486(7402):207–214.

TIOBE (2014). TIOBE index for june 2014. `http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html`. [Online; accessed 1-July-2014].

Turnbaugh, P. J., Ley, R. E., Hamady, M., Fraser-Liggett, C., Knight, R., and Gordon, J. I. (2007). The human microbiome project: exploring the microbial part of ourselves in a changing world. *Nature*, 449(7164):804–810.

Vines, T. H., Albert, A. Y., Andrew, R. L., Débarre, F., Bock, D. G., Franklin, M. T., Gilbert, K. J., Moore, J.-S., Renaut, S., and Rennison, D. J. (2014). The availability of research data declines rapidly with article age. *Current Biology*, 24(1):94 – 97.

Waidmann, M., Bechtold, O., Frick, J.-S., Lehr, H.-S., Schubert, S., Dobrindt, U., Loeffler, J., Bohn, E., and Autenrieth, I. B. (2003). Bacteroides vulgatus protects against Escherichia coli-induced colitis in gnotobiotic interleukin-2-deficient mice. *Gastroenterology*, 125(1):162 – 177.

Walker, J., Marsh, S., and Nyrén, P. (2007). Methods in molecular biology. In *Pyrosequencing Protocols*, volume 373, pages 1–13. Humana Press.

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

Weber, N. (2013). *Computational approaches for analyzing ancient genomes and modern metagenomes*. PhD thesis, Universität Tübingen.

Whitman, W. B., Coleman, D. C., and Wiebe, W. J. (1998). Prokaryotes: the unseen majority. *Proceedings of the National Academy of Sciences*, 95(12):6578–6583.

Wolff, E. (2010). *Spring 3: Framework für die Java-Entwicklung*. DPunkt, Heidelberg, 3. edition.

Wright, E. S., Yilmaz, L. S., and Noguera, D. R. (2012). DECIPHER, a search-based approach to chimera identification for 16S rRNA sequences. *Applied and Environmental Microbiology*, 78(3):717–725.

wuqingren2316 (2014). RMI, Hessian, Burlap, Httpinvoker, WebService comparison. `http://www.javawebdevelop.com/364546/`. [Online; accessed 18-May-2014].

Xie, W., Wang, F., Guo, L., Chen, Z., Sievert, S. M., Meng, J., Huang, G., Li, Y., Yan, Q., Wu, S., Wang, X., Chen, S., He, G., Xiao, X., and Xu, A. (2011). Comparative metagenomics of microbial communities inhabiting deep-sea hydrothermal vent chimneys with contrasting chemistries. *The ISME Journal*, 5(3):414–426.

Yilmaz, P., Kottmann, R., Field, D., Knight, R., Cole, J. R., Amaral-Zettler, L., Gilbert, J. A., Karsch-Mizrachi, I., Johnston, A., Cochrane, G., Vaughan, R., Hunter, C., Park, J., Morrison, N., Rocca-Serra, P., Sterk, P., Arumugam, M., Bailey, M., Baumgartner, L., Birren, B. W., Blaser, M. J., Bonazzi, V., Booth, T., Bork, P., Bushman, F. D., Buttigieg, P. L., Chain, P. S. G., Charlson, E., Costello, E. K., Huot-Creasy, H., Dawyndt, P., DeSantis, T., Fierer, N., Fuhrman, J. A., Gallery, R. E., Gevers, D., Gibbs, R. A., Gil, I. S., Gonzalez, A., Gordon, J. I., Guralnick, R., Hankeln, W., Highlander, S., Hugenholtz, P., Jansson, J., Kau, A. L., Kelley, S. T., Kennedy, J., Knights, D., Koren, O., Kuczynski, J., Kyrpides, N., Larsen, R., Lauber, C. L., Legg, T., Ley, R. E., Lozupone, C. A., Ludwig, W., Lyons, D., Maguire, E., Methe, B. A., Meyer, F., Muegge, B., Nakielny, S., Nelson, K. E., Nemergut, D., Neufeld, J. D., Newbold, L. K., Oliver, A. E., Pace, N. R., Palanisamy, G., Peplies, J., Petrosino, J., Proctor, L., Pruesse, E., Quast, C., Raes, J., Ratnasingham, S., Ravel, J., Relman, D. A., Assunta-Sansone, S., Schloss, P. D., Schriml, L., Sinha, R., Smith, M. I., Sodergren, E., Spor, A., Stombaugh, J., Tiedje, J. M., Ward, D. V., Weinstock, G. M., Wendel, D., White, O., Whiteley, A., Wilke, A., Wortman, J. R., Yatsunenko, T., and Glockner, F. O. (2011). Minimum information about a marker gene sequence (MIMARKS) and minimum information about any (x) sequence (MIxS) specifications. *Nature Biotechnology*, 29(5):415–420.

Yilmaz, P., Parfrey, L. W., Yarza, P., Gerken, J., Pruesse, E., Quast, C., Schweer, T., Peplies, J., Ludwig, W., and Gloeckner, F. O. (2013). The SILVA and “All-species Living Tree Project (LTP)” taxonomic frameworks. *Nucleic Acids Research*.

Yooseph, S., Sutton, G., Rusch, D. B., Halpern, A. L., Williamson, S. J., Remington, K., Eisen, J. A., Heidelberg, K. B., Manning, G., Li, W., Jaroszewski, L., Cieplak, P., Miller, C. S., Li, H., Mashiyama, S. T., Joachimiak, M. P., van Belle, C., Chandonia, J.-M., Soergel, D. A., Zhai, Y., Natarajan, K., Lee, S., Raphael, B. J., Bafna, V., Friedman, R., Brenner, S. E., Godzik, A., Eisenberg, D., Dixon, J. E., Taylor, S. S., Strausberg, R. L., Frazier, M., and Venter, J. C. (2007). The Sorcerer II Global Ocean Sampling Expedition: Expanding the universe of protein families. *PLoS Biology*, 5(3):e16.

Zhao, Y., Tang, H., and Ye, Y. (2012). RAPSearch2: a fast and memory-efficient protein similarity search tool for next-generation sequencing data. *Bioinformatics*, 28(1):125–126.