The MGX framework for microbial community analysis

Zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften an der Technischen Fakultät der Universität Bielefeld vorgelegte Dissertation

von

Sebastian Jaenicke

5. Juni 2019

Sebastian Jaenicke
Bahnhofstraße 103
35440 Linden
sjaenick@CeBiTec.Uni-Bielefeld.DE

Supervisors: Prof. Dr. Jens Stoye
Prof. Dr. Alexander Goesmann

Gedruckt auf alterungsbeständigem Papier (ISO 9706).

Contents

1

Introduction

For the microbial ecologist, what can be cultured is the basis of his conception of what exists. This is exactly like learning about animals from visiting zoos.

– Carl R. Woese, Microbiology in transition

1.1 Preface

It is assumed that more than 99% of all microbiota cannot be cultured under laboratory conditions (Streit and Schmitz, 2004), and only culture-independent approaches like metagenomics enable us to study their genetic information, gain access to their metabolic potential and better understand interactions within microbial communities. The importance of these microbiota cannot be understated, as they

are known to play an important role not only in natural ecosystems, but also within ourselves, where they contribute to the health of the host:

"Complex microbial communities shape the dynamics of various environments, ranging from the mammalian gastrointestinal tract to the soil. Advances in DNA sequencing technologies and data analysis have provided drastic improvements in microbiome analyses, for example, in taxonomic resolution, false discovery rate control and other properties, over earlier methods" (Knight *et al.*, 2018).

The MGX framework for metagenome analysis has been developed exactly for this purpose, facilitating capabilities for large-scale data processing and allowing to rapidly evaluate and adopt novel methodological developments. MGX enables researchers to benefit from the most recent bioinformatics methods, thus allowing them to analyze and interpret their metagenome datasets in order to gain valuable new insights into the composition and functioning of interacting microbial life.

1.2 Structure of this document

Chapter 2 introduces the basic terminology of different approaches for the analysis of microbial communities, detailing their individual advantages as well as their shortcomings. The chapter then provides an overview of the methods and tools that are relevant for metagenome data generation, analysis and interpretation. Key developments in sequencing techniques are presented, outlining basic biological methodology, performance indicators and advantages as well as inherent weaknesses of the individual approaches. Also, their relevance to metagenomics is outlined briefly. Afterwards, the most prominent bioinformatics approaches for the analysis of metagenome data are introduced. Starting with early methods that were transferred from the genomics field, the technical and algorithmic advances over time that lead to improved and more target-oriented methods are presented in detail. Apart from individual software packages, the major integrated platforms for metagenome data analysis are introduced. The chapter closes with an interim conclusion which sums up the achievements reached so far and also outlines the major shortcomings of existing approaches for microbial community data analysis.

Consequently, Chapter 3 opens with a definition of the main objectives that were identified for the design and implementation of the MGX framework for the analysis of unassembled metagenome data. The chapter presents key decisions and their rationale in the design of the software, detailing the different features and capabilities that were implemented.

Chapter 4 presents the design and implementation of a taxonomic classification workflow that was developed for use within MGX. The workflow was created as

a combination of preexisting tools, and according to the evaluation that was conducted employing *in silico* datasets, provides superior performance in comparison to standalone algorithms. Also, additional improvements that were implemented after the initial publication of the workflow are described.

Chapter 5 sums up the achievements that were accomplished and how the availability of the MGX software contributed to advance the field of metagenome analysis and interpretation. For a project this size, and the time that was required to implement it, it is natural that some aspects turned out to be addressed suboptimally, biological developments were predicted wrongly, or new technical advances evolved that were not foreseen during the initial design phase. Thus, this chapter also identifies those parts that retrospectively were not solved satisfactorily as well as lessons learned from the feedback of users of the application. As a future extension of the MGX framework is already planned, the chapter hence closes with an outlook outlining possible starting points for new components that would contribute to further improve the application.

2

Background

. . . knowledge of sequences could contribute much to our understanding of living matter.

– Frederick Sanger

2.1 Microbial community analysis

In 1998, J. Handelsman coined the term "metagenomics" (Handelsman *et al.*, 1998) for the study of microbial communities based on their genetic material. Ever since, metagenomics has not only evolved into a popular term, but also into one that is often attached to any kind of study related to microbial community research. Many publications are misattributed as metagenomics studies even though the term doesn't apply to the content at all; various studies identify themselves as being of metagenomic nature as long as at least some kind of next-generation sequencing

technique is applied to evaluate the properties of microbial communities. It is thus important to define a basic terminology that will be used throughout this document, which follows the vocabulary proposed by Marchesi and Ravel (2015). Of course, other methods are also at hand that allow to study the properties of microbial communities without prior culturing, *e.g.* metaproteomics, which aims at the identification and characterization of proteins obtained from a certain habitat; however, due to the scope of this thesis, only sequence-based approaches are presented.

2.1.1 Metataxonomics

Metataxonomics, sometimes also denoted as metabarcoding or metagenetics[1], describes the approach where specific genes of interest are targeted for amplification and sequencing. Highly conserved regions are required to design appropriate primer pairs; in addition, the distance between the respective primer binding sites has to be chosen with respect to the selected sequencing strategy, *i.e.* the method has to be able to span the complete distance between the primers.

The most popular metataxonomics target is the 16S rRNA gene as an established taxonomic marker for prokaryotes due to the presence of both highly conserved as well as highly variable regions, and Sanger as well as next-generation sequencing technologies have successfully been applied for metataxonomics studies based on the 16S rRNA gene. 27F/1492R (Lane, 1991) is a popular example for a set of universal primers targeting the 16S rRNA gene with a distance of approximately 1,464 bp in *Escherichia coli*, allowing sequencing of the entire amplicon with a Sanger paired-end strategy. For next-generation technologies like Illumina, the V3/V4 hypervariable region of the 16S gene represents the most common target, with primer pairs like S-D-Bact-0341-b-S-17/S-D-Bact-0785-a-A-21 being used (Klindworth *et al.*, 2013); the resulting amplicon library with an average fragment length of 460 bp is suitable for sequencing with an Illumina 2×300 bp paired-end protocol.

While third-generation sequencing platforms like PacBio SMRT allow obtaining long fragments, their error rate is so far insufficient for amplicon sequencing, where downstream bioinformatics analysis requires high sequence quality. Even with approaches like circular consensus sequencing (CCS), where multiple passes over the template are performed to obtain improved accuracy, the error rate of the consensus sequence still remains too high.

For eukaryotes, the 18S rRNA gene or the ITS[2] region represent established targets for metataxonomics. Sometimes, other marker genes are also used to restrict the study to certain taxa of interest with known metabolic capabilities, *e.g. mcrA*

[1] `http://www.opiniomics.org/youre-probably-not-doing-metagenomics/`
[2] internal transcribed spacer

(Kröber *et al.*, 2009; Wilkins *et al.*, 2015), which encodes a subunit of the methyl-coenzyme-M reductase gene involved in methanogenesis and is commonly found in anaerobic digesters and biogas fermenters. Marker-gene based approaches are unsuitable for viruses, which lack conserved genes such as 16S or 18S rRNA. Even for established taxonomic markers such as 16S, a large fraction of the microbial community might be missed (Eloe-Fadrosh *et al.*, 2016a).

The main advantage of metataxonomic studies is that the target gene is already known, and a high resolution may be obtained with relatively small sequencing effort. However, this also means that the method is limited to taxonomic and phylogenetic profiling only. On the other hand, several disadvantages are known to exist: Primers do not have equal affinity for all possible targets and thus consequently introduce a bias during the PCR amplification. Also, the target gene may occur more than once within an organism, thus artificially inflating its relative abundance. For an appropriate normalization, extensive information about gene copy numbers for all organisms would be required, which typically can only be obtained for genomes that have already been sequenced before.

Bioinformatic analysis of amplicon datasets is widely applied to deduce the taxonomic composition of microbial communities, and several popular tools like mothur (Schloss *et al.*, 2009) or QIIME (Caporaso *et al.*, 2010) are available to support data analysis and interpretation. Taxonomic classification of sequences, however, is highly dependent on the availability of appropriate reference databases, and while these are available for 16S, 18S and ITS sequences, the assignment of amplicons for other target genes often remains problematic.

Apart from direct sequence classification based on *e.g.* a taxonomic assignment derived from the results of a BLAST homology search (Rubio *et al.*, 2014; Lori *et al.*, 2018), clustering approaches are often employed in order to subdivide a dataset into operational taxonomic units, short OTUs (Sneath and Sokal, 1963), which are computed based on sequence similarity and intended to represent a taxonomic group. OTUs can either be generated *de novo*, *i.e.* all sequences within a dataset are clustered among themselves, or in a "closed-reference" approach, where sequences are clustered against a reference database and reads without a match are excluded from subsequent analyses.

The OTU approach has repeatedly been called into question, as OTUs are often misinterpreted as microbial species, and sequencing artifacts may result in overestimation of true microbial diversity (Kunin *et al.*, 2010). In addition, grouping sequences into OTUs possibly hides valuable information: "OTUs underutilize the quality of modern sequencing by precluding the possibility of resolving fine-scale variation" (Callahan *et al.*, 2016). Also, the 97% identity threshold that is often used to define species-specific OTUs and was already proposed back in 1994 (Stackebrandt and Goebel, 1994) has been criticized: "If the goal of OTUs is to

approximate species, then the canonical 97% threshold is far from optimal for all clustering algorithms and should be increased to at least 99%." (Edgar, 2018)

A different approach has been proposed in the form of amplicon sequence variants (ASVs), which offer improved resolution "without imposing the arbitrary dissimilarity thresholds that define molecular OTUs" (Callahan *et al.*, 2017). As per the authors, "the improvements in reusability, reproducibility and comprehensiveness are sufficiently great that ASVs should replace OTUs as the standard unit of marker-gene analysis and reporting."

2.1.2 Metagenomics

In the study that initially defined the metagenomics term, Handelsman *et al.* (1998) emphasize the immense potential of microbes inhabiting soil and the possibility to study the genetic material of unculturable organisms in order to obtain access to novel natural compounds:

> "A new frontier of science is emerging that unites biology and chemistry – the exploration of natural products from previously uncultured soil microorganisms. The approach involves directly accessing the genomes of soil organisms that cannot be, or have not been, cultured by isolating their DNA, cloning it into culturable organisms and screening the resultant clones for the production of new chemicals." (Handelsman *et al.*, 1998)

Of course, since next-generation sequencing approaches were not yet available, their original definition still includes the cloning step that was necessary for Sanger-based sequencing strategies. Nowadays, this step is no longer required, and metagenomics is commonly understood as a culture-independent approach to gain admittance to the total gene pool of microorganisms residing in an ecosystem.

As sequencing is conducted in an undirected manner, no prior assumptions (such as presence/absence of certain marker genes) are made, and DNA viruses or phages are sequenced with equal precedence as prokaryotic and eukaryotic DNA fragments.

In a typical metagenomics study, whole genomic DNA is thus extracted from a natural or synthetic environment and subjected to sequencing. There are different options for the computational analysis of such data (Section 2.3), which, after preceding quality control, most commonly revolve around determining the relative abundance of the different organisms present in an environment (taxonomic profiling) as well as the identification of genes that occur in them (functional analysis). Above all, additional analyses are highly dependent on the actual aim of the study: an investigation of clinical samples might focus on pathogens or antibiotic resistance

genes, while a survey aimed at the discovery of novel enzymes for biotechnological exploitation would rather require specialized databases to detect and characterize possible targets. Finally, statistical evaluation is typically performed, for example to determine whether sufficient coverage is provided by the available sequence data, compute biodiversity indices which describe the compositional complexity of the identified microbiota, or to identify significant differences between multiple datasets.

The computational analysis of metagenome datasets can either be performed by direct analysis of the raw, unassembled sequence data, or metagenome assembly might be attempted. Both approaches have their strengths and weaknesses, which will be presented in the following two sections. The preferable course of action should be chosen taking into account the study's specific goals as well as the characteristics of the available sequence data. In both cases, "metagenomics circumvents the unculturability and genomic diversity of most microbes, the biggest roadblocks to advances in clinical and environmental microbiology" (Anne and Ann, 2007).

2.1.2.1 Read-based metagenomics

Read-based metagenome analysis allows to directly infer the taxonomic composition and metabolic potential of the microbial community in a given environment. Unlike marker-based approaches, metagenomics avoids possible biases induced by PCR amplification, and even while a higher sequencing effort is required in order to obtain sufficient amounts of data, the undirected sequencing approach provides more information about the microbiota.

Nowadays, most metagenome studies are based on the analysis of raw sequence data, and several user-friendly applications are readily available to aid the scientist in data handling, analysis and interpretation (Section 2.4). Most of these solutions have been custom-tailored for the analysis of the microbial fraction of a metagenome, but more specialized tools and databases are also available targeting viral (Rampelli *et al.*, 2016; Skewes-Cox *et al.*, 2014), fungal (Donovan *et al.*, 2018), or phage (Jurtz *et al.*, 2016) sequences.

However, the information content of short DNA reads is limited, and even with the most recent tools, a large fraction of a dataset typically remains unassigned. Also, these short DNA fragments still contain sequencing errors, another aspect contributing to wrong classification results as well as an artificial overestimation of microbial diversity.

2.1.2.2 Metagenome assembly

In contrast to read-based metagenomics, if community complexity is low or data volume large enough, metagenomic assembly becomes a viable option. The approach allows to obtain larger contiguous DNA stretches, and thus full-length genes, partial and sometimes even complete genomes might be recovered. Also, sequencing errors that might still be present in the raw data are to a certain degree corrected by the assembly process.

Several DNA assemblers have been either adapted or specifically developed for the assembly of metagenomes, among them metaSPAdes (Nurk *et al.*, 2017), MEGAHIT (Li *et al.*, 2016) or Ray Meta (Boisvert *et al.*, 2012). Taxonomic binning software such as MetaBAT (Kang *et al.*, 2015) or GroopM (Imelfort *et al.*, 2014) relies on coverage information as well as intrinsic sequence properties to subdivide the assembled contigs into "bins" hopefully representing taxonomic entities, and the basic principles of genome annotation may be applied for gene prediction and functional annotation of these potential draft genomes.

However, metagenome assembly is not only computationally demanding, but also a highly challenging task, as the different organismal abundances result in varying coverage in contrast to single-genome assembly. It also induces new possible biases: Depending on the stringency of the assembly software, the genomes of highly abundant species might either not be assembled at all due to intra-species variations, or these variations might be collapsed into a consensus of several different strains, thus hiding possibly valuable information. At the same time, genomes of organisms present in low amounts are unlikely to be assembled at all due to lack of coverage. Finally, genes and DNA stretches conserved in more than one organism promote possible misassemblies, resulting in chimeric contigs.

Since unassembled DNA reads are typically no longer considered after the metagenome assembly step, these effects contribute to shifts in taxonomic and functional profiles, even if varying coverage has been taken into account and addressed.

As described by Westbrook *et al.* (2017), "...researchers often turn to metagenome assembly and subsequent annotation, which has profound shortcomings, such as chimeric assembly of closely related sequences, strong bias toward abundant organisms, and substantial human and computer resource requirements".

2.1.3 Metatranscriptomics

In contrast to metagenomics, which aims to identify the microbes that are present within a certain environment and to capture their genetic potential, metatranscrip-

tomics is focused on quantifying actual gene expression levels and genetic diversity. Also, metatranscriptomics allows to perform differential gene expression analysis between different habitats or the same habitat under different conditions, *e.g.* to identify biomarkers that might indicate a change of external conditions or the presence of possibly harmful substances.

In a characteristic metatranscriptomics study, sequencing of community RNA is conducted after reverse transciption into cDNA[3]. As the largest fraction of a transcriptome consists of ribosomal RNA, different wet-lab approaches are applied in order to increase the yield of gene-coding RNA available for downstream bioinformatics analysis (Griffith *et al.*, 2015). For this purpose, kits are available from several manufacturers, which employ strategies like poly(A) capturing to enrich eukaryotic mRNA or rRNA depletion based on oligomers complementary to highly conserved subregions of ribosomal RNA genes.

While this approach greatly reduces the amount of ribosomal RNA in a sample, this process is not species-agnostic and causes a biased composition of the leftover rRNA. Thus, while *e.g.* taxonomic assignments of 16S rRNA fragments recovered from a metatranscriptome may be used to predict the presence or absence of individual species, corresponding taxonomic abundance profiles deduced from this data would be massively distorted.

Transcript assignment itself is also a difficult task, as appropriate reference sequences are typically not available in public databases, and transcripts originating from highly conserved genes might still align to the reference genome of a completely different species. Different approaches have been suggested, among them the parallel sequencing, assembly and annotation of a metagenome obtained from the same community in order to establish a reference gene catalog of the microbial community. However, this strategy is highly dependent on the complexity of the microbial community, and large sequencing effort might be necessary to obtain sufficient community coverage. Also, the quality of the assembly has a significant influence on downstream statistical evaluation, and misassemblies as well as chimeric contigs contribute to a distortion of results.

In single organism transcriptomics studies, the obtained sequence data is aligned to a corresponding reference genome in order to identify the correct source gene of the transcript. Afterwards, transcript counts are normalized using *e.g.* RPKM (Mortazavi *et al.*, 2008), FPKM (Trapnell *et al.*, 2010), or TPM (Li and Dewey, 2011) values in order to allow comparison between different conditions. These methods cannot directly be transferred towards metatranscriptomics, as appropriate reference genomes will not be available in most cases; also, gene expression levels cannot be directly compared between organisms, as this would induce a bias by favoring organisms with higher rates of transcription.

[3]complementary DNA

A different approach involves the de-novo assembly of the obtained metatranscriptome(s), which avoids the problems arising from the lack of appropriate reference genomes. In an independent study, it could be shown that metatranscriptome assembly using *e.g.* Trinity (Grabherr *et al.*, 2011) or MetaVelvet (Namiki *et al.*, 2012) significantly improved the number of transcript reads that could be functionally annotated (Celaj *et al.*, 2014). However, it should be noted that even with the best assembly generated in the study, only 50.3% of all sequences could be assigned to a source transcript. While this represents a major improvement over the analysis of unassembled sequences, which resulted in an assignment rate of only 15.5%, a large fraction of the dataset remained unannotated.

In a recent publication (Klingenberg and Meinicke, 2017), the authors suggested a novel approach involving a taxonomic binning step followed by taxon-specific scaling in order to obtain a normalized gene expression profile that respects different source organisms.

2.2 Sequencing technologies

In the 1970s, Ray Wu for the first time was able to determine the nucleotide sequence of a DNA fragment employing a chromatography-based approach (Wu, 1972). Ever since, DNA sequencing has become one of the building blocks of molecular biology. For several decades, the predominant Sanger sequencing method did not undergo any major changes, apart from minor improvements and increased automation. This finally changed in 2005, when the first next-generation sequencing technique became widely available and allowed to obtain unprecedented amounts of genetic information at reasonable cost (Figure 2.1). Nowadays, DNA sequencing is a readily available process performed in laboratories around the world, and new genome sequences are determined each day in a routine manner. These advances have enabled scientists to shift their attention away from individual genomes towards the sequence-based study of related organisms (comparative genomics), and finally, towards microbial community research. This section presents the technological progress in DNA sequencing over time, providing a description of the underlying biological approaches and potential shortcomings which need to be taken into account when processing and interpreting DNA sequence data. While all major sequencing technologies are introduced, those that never gained widespread adoption such as SOLiD[4] have been omitted for brevity.

[4]Sequencing by Oligonucleotide Ligation and Detection

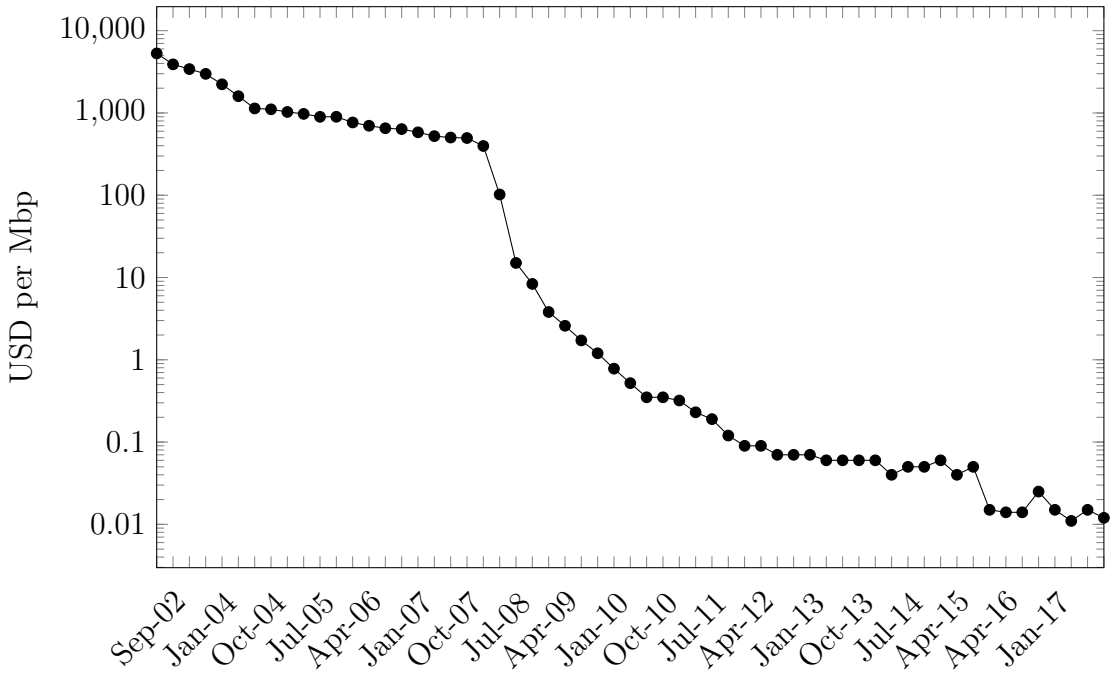


Figure 2.1: Decline of DNA sequencing costs. The advent of the next generation sequencing methods dramatically reduced the costs for DNA sequencing. Data source: `https://www.genome.gov/pages/der/Sequencing_Costs_Table_July_2017.xlsx`

2.2.1 Low-throughput sequencing

In 1977, Allan Maxam and Walter Gilbert published the first widely applied[5] approach for DNA sequencing which became commonly known as chemical sequencing and relied on chemical modification and subsequent cleavage at specific nucleotides (Maxam and Gilbert, 1977). The method became quite popular for a short period of time, but was quickly disbanded in favor of the chain termination method proposed by Sanger (Sanger *et al.*, 1977), which required fewer radioactive substances and less toxic chemicals (Karger and Guttman, 2009) while at the same time delivering comparable output (Maxam-Gilbert: "at least 100 bases"; Sanger: "15 to about 200 nucleotides").

The method requires prior amplification of the DNA molecule to be sequenced, which can be performed either by in-vitro PCR[6] or by cloning of the target fragment into a plasmid of a bacterial host. Bacterial cloning provides lower error rates than PCR (natural DNA replication: 1 in 10 billion; PCR: 1 in 1 million) and makes it

[5] not counting Sanger's and Coulson's plus-minus sequencing approach (Sanger and Coulson, 1975), which was published just two years beforehand.

[6] polymerase chain reaction

possible to distinguish *e.g.* between different variants of genes present in multiple copies, but requires a higher initial effort.

For Sanger-based sequencing, the sample consisting of an amplified single-stranded template DNA is divided into four equal parts, which are deposited onto different lanes of a slab gel together with DNA polymerase, standard deoxynucleotides as well as a small fraction of dideoxynucleotides (ddNTPs; approx. 1%) labeled with radioactive phosphorus. For each of the four samples, only one of the ddNTPs is added. During synthesis of the second strand, the DNA polymerase either incorporates a standard nucleotide, after which further extension can continue, or a ddNTP, which prohibits further strand elongation. Afterwards, gel electrophoresis is used to separate the DNA fragments based on their molecular weight, and an autoradiography is created to make the fragment locations visible. Based on the position of the DNA band and the samples lane, which identified the corresponding ddNTP, the complete DNA sequence of the fragment can be read off the autoradiography image.

Within short time, Sanger's method evolved into the de-facto standard for DNA sequencing, and over the years several technical advances helped automation and contributed to increased throughput; fluorescent dye labeling of dideoxynucleotides (ddNTPs) replaced the initially used radioactive compounds and capillary-based containment of the sequencing reaction instead of polyacrylamide slab gels allowed to perform multiple sequencing reactions in parallel.

For over 20 years, Sanger sequencing represented the prevailing method to determine the DNA sequence of an organism and was used to establish the genome sequences of many different bacterial (*e.g.* Fleischmann *et al.*, 1995; Blattner *et al.*, 1997), archaeal (Bult *et al.*, 1996) and eukaryotic organisms (Goffeau *et al.*, 1996) as well as the first draft versions of the human genome (Venter *et al.*, 2001; Consortium *et al.*, 2001).

Sequences obtained with the Sanger technique represent the consensus of millions of individual reactions and hence are of very high quality. However, the process is still prone to several sources of error: "Methods based on bacterial cloning and Sanger-chemistry sequencing were subject to many coverage-reducing biases, notably at GC extremes, palindromes, inverted repeats, and sequences toxic to the bacterial host" (Ross *et al.*, 2013).

Nowadays, modern Sanger-based capillary sequencers such as the $3730xl$ DNA Analyzer marketed by Applied Biosystems™ still remain in widespread use; with support for up to 384 sequencing reactions to be performed in parallel, they are able to deliver read lengths up to 1,100 bp at a high quality and low per-reaction cost. These sequencers are mostly utilized in certain application areas where only

low throughput is required, *e.g.* in molecular diagnostics for HLA[7] typing, pathogen detection, or sequence validation.

First metagenomic studies based on Sanger sequencing required high effort and cost for comparably low output. Several studies successfully employed Sanger sequencing in order to generate metagenome datasets and study *e.g.* human gut microbiota (Gill *et al.*, 2006) or microbial communities occurring in the Sargasso Sea (Venter *et al.*, 2004; Yooseph *et al.*, 2007). Nonetheless, while these studies discovered large numbers of novel organisms, coverage by the obtained sequence data did not suffice to adequately represent the complete microbial communities.

2.2.2 Next-generation sequencing

2.2.2.1 Pyrosequencing

The advent of the next-generation sequencing era began in 2005, when 454 Life Sciences presented the Genome Sequencer GS20 instrument, which relied on the so-called pyrosequencing technology (Ronaghi *et al.*, 1996); within just two years after market introduction, pyrosequencing initiated a dramatic drop of sequencing costs. The method builds upon a DNA polymerase synthesizing the second strand of single-stranded and previously immobilized template DNA fragments ("sequencing-by-synthesis principle"), using nucleoside triphosphates (dNTPs) as a substrate, which leads to the release of pyrophosphate ions proportional to the number of incorporated nucleotides. This pyrophosphate is then being used to drive an enzymatic cascade consisting of luciferase and ATP sulfurylase, ultimately resulting in photon emission, which is detected by an optics system (Nyrén, 1987; Margulies *et al.*, 2005).

As single photons would still be difficult to detect, not one, but several million identical copies of the template DNA are required in order to obtain a reliable signal; these copies are generated during sequencing library preparation using an emulsion PCR (emPCR) protocol, where the short template fragments are ligated to beads and amplified within aqueous droplets, thus ideally ensuring that each bead will carry clonal copies of only one single DNA molecule.

Unprecedented parallelism is achieved by performing the strand synthesis within the etched wells of a glass slide, the PicoTiterPlate™ (PTP); these wells serve as small reaction vessels, and a single PTP bears over 1.6 million of them. The wells are loaded with the DNA-carrying beads, depositing exactly one bead into each well. The sequencing system itself operates in cycles, alternating between one of the four nucleotides being flowed over the PTP in a fixed order, with intermittent washing steps to remove leftover nucleotides and reagent residue from the preceding

[7] Human Leukocyte Antigen

cycles. For each nucleotide flow, a CCD[8] camera system takes an image of the whole slide, and during downstream image processing, the nucleotide sequences for the fragments in each well are determined based on light intensity measurements.

As strand synthesis is not blocked after incorporation of a complementary nucleoside, multiple subsequent identical residues (homopolymers) of the DNA template are synthesized within just one flow, and the number of incorporations has to be derived from the light intensity signal. This poses a problem as homopolymer length gradually increases, since the exact number of nucleotides can no longer reliably be determined and accuracy deteriorates. Also, the emulsion PCR amplification has been shown to be a major source of error; firstly, multiple beads can be captured within one droplet of the emulsion, and a DNA fragment is thus amplified across the surface of several beads, ultimately resulting in duplicate reads when the beads are deposited in different wells of the PicoTiterPlate (Gomez-Alvarez *et al.*, 2009). A different issue is related to the GC content of the DNA template, as GC-rich DNA tends to form stable secondary structures (Frey *et al.*, 2008) which prevent template amplification during the emulsion PCR phase. This problem was also shown to exist for the 454 technology (*e.g.* Jaenicke *et al.*, 2011; Schwientek *et al.*, 2012) and was partially addressed by a modification of the PCR reagents with an emPCR additive that was later identified to be trehalose (Schwientek *et al.*, 2011). Slowing down the PCR reaction, trehalose inhibits the self-annealing of the DNA during the emPCR step, allowing to amplify even GC-rich template DNA.

Obtainable read length is limited by residue buildup during the sequencing run, and accuracy decreases over cycles due to noise caused by either signal carry-forward (insufficient washing between flows) or incomplete extension (insufficient amount of nucleosides during incorporation). The instrument software partially addresses these issues *e.g.* by the CAFIE (carry forward and incomplete extension) filter, which either shortens or completely discards reads based on signal quality.

The first pyrosequencing system, the Genome Sequencer GS 20, was able to generate 200,000 reads with an average length of 100 bp; an improved version, the GS FLX instrument, delivered up to 600,000 reads with lengths up to 400–500 bp. The GS FLX Titanium upgrade, finally, used titanium-coated glass slides to reduce crosstalk between the wells; combined with other improvements, the system could provide more than a million reads, while at the same time obtainable read lengths could be slightly increased, as well. Aiming at the diagnostics market, Roche/454 also developed a smaller benchtop sequencer, the GS Junior, offering approximately one tenth the output of the GS FLX system (see also Table 2.2).

For metagenomics applications, the 454 pyrosequencing technology was groundbreaking, allowing to sequence sufficient amounts of genetic material at a reasonable cost for the first time. This, for example, enabled scientists to study some of the

[8] Charge-coupled device

more complex microbial environments such as biogas fermenters (Krause *et al.*, 2008), endogenous microbes of an ancient mammoth (Poinar *et al.*, 2006), or marine viral communities (Breitbart *et al.*, 2002). Also, several approaches have been proposed in order to mitigate the impact of the inherent problems of the technology, *e.g.* tools for sequence data correction (Quince *et al.*, 2009), duplicate removal (Niu *et al.*, 2010), frameshift-aware protein classification tools (Zhang and Sun, 2011), or filtering steps intended to lessen the bias inflicted by uneven GC representation (Jaenicke *et al.*, 2011).

2.2.2.2 IonTorrent

The underlying principle of the IonTorrent technology is identical to that of pyrosequencing, but relies on unmodified dNTPs (deoxyribonucleoside triphosphates) as a substrate (Rothberg *et al.*, 2011). As with the pyrosequencing approach, short DNA fragments are amplified using an emulsion PCR step and the actual sequencing reaction is performed by synthesis of the second strand. As incorporation of a nucleotide releases a hydrogen ion (H^+), the pH value of the surrounding solution is slightly lowered; the sequencing reaction is performed in microscopic wells on top of a CMOS[9] semiconductor chip containing ISFETs[10], which are able to detect the pH shift as a change in voltage.

It is especially this use of semiconductor technology that promoted use of the IonTorrent PGM benchtop sequencer, as existing semiconductor factories could be used for chip manufacturing, lowering the operating cost of the instrument. The device was marketed as a direct competitor to the 454 GS Junior system, offering better throughput as well as reduced sample preparation time at lower cost. Also, future throughput improvements would easily be obtainable, as novel enhancements in semiconductor development would directly benefit capacity and accuracy of the sequencing chip. Three different types of sequencing chips are available for the PGM sequencer, the 314™, 316™ and 318™ series, offering an overall output between 30 Mbp to 2 Gbp at an individual read length of up to 400 bp. In addition, IonTorrent also released the Proton sequencer, which delivers between 60 and 80 million reads up to 200 bp for a total output of 10 Gbp using the Ion PI™ chip.

The IonTorrent technology exhibits the same disadvantages as 454 pyrosequencing, especially its susceptibility to homopolymer-related errors. However, the omission of the enzymatic cascade, the use of unlabeled nucleosides and the direct measurement of an electrical current instead of an expensive optics-based signal detection system contributed to a slightly increased base calling accuracy in comparison to pyrosequencing-based approaches. At the same time, however, semiconductor se-

[9] Complementary metal-oxide-semiconductor
[10] Ion-sensitive field-effect transistors

quencing was shown to be more prone to incorrect resolval of homopolymer stretches and frameshift-related errors (insertions and deletions; indels) than pyrosequencing (Loman *et al.*, 2012). Due to their comparable error properties, the same tools that were developed for pyrosequencing data could also be applied to IonTorrent-based metagenome datasets.

2.2.2.3 Illumina

In 2006, Solexa, which was later acquired by Illumina, released the Genome Analyzer (GA) IIx, a sequencing platform which employed a different variant of the "sequencing-by-synthesis" approach.

Short DNA fragments with ligated adapters are bound to an oligo-coated glass surface, the flow cell, and a step called solid-phase bridge amplification (Kawashima *et al.*, 2012) is performed, generating dense colonies ("clusters") of approximately 1,000 identical template copies in close vicinity to each other. On the flow cell surface, this amplification method produces between 100 and 200 million spatially separated clusters.

The sequencing run is then performed in cycles, during which a DNA polymerase synthesizes the second strand using reversible terminator bases (RT bases) as a substrate (Canard and Sarfati, 1994); these RT bases, 3′-O-azidomethyl-2′-deoxynucleoside triphosphates, are flowed over the glass slide, but unlike pyro- or semiconductor sequencing, which both alternate between the four different nucleotides, all nucleotides are supplied at the same time. The RT bases prevent elongation of the second DNA strand after the incorporation of only one nucleotide; two lasers then excite the fluorophores that are attached to the RT bases, an image of the flow cell is taken and for each cluster, the exact base is finally identified based on the emitted wavelengths of the different fluorophores. After the synthesis of one base, an intermittent step is required to cleave off the fluorescent dyes, again permitting strand elongation in the subsequent cycle. Finally, the DNA sequence for the template fragments can be derived from the fluorophore signal sequence for each cluster. Due to the cycle-based nature of the method, all sequences obtained within one sequencing run exhibit the same length.

The Illumina technology revolutionized the sequencing market with unprecedented basecall accuracy, high output and the near absence of indel errors compared to previously released platforms (Loman *et al.*, 2012). Even their first device, the Genome Analyzer IIx system, was able to provide more than 130 million sequences per run with a read length of 35 bp. Illumina was able to continuously improve the method, with the currently marketed chemistry allowing to perform sequencing runs up to 2 times 300 bp (paired-end). At the same time, cluster density on the flow cell could also be increased, contributing to enhanced overall output.

Different sequencing devices were released to address varying customer needs, ranging from medium-output benchtop sequencers (MiniSeq, MiSeq, NextSeq series) to large production-scale machines like the HiSeq and NovaSeq systems offering up to 6 Tbp of sequencing information per run. Illumina's devices are the currently most widespread and commercially successful sequencing technology, offering solutions for almost all high-throughput sequencing applications at low cost.

Substitution miscalls are the main source of error for reads originating from the Illumina sequencing method. As only one base is synthesized per cycle, almost no indel errors are present within Illumina datasets, but basecall accuracy deteriorates with increasing cycle numbers due to reagent residue buildup and phasing (incomplete fluorophore/terminator cleavage) or prephasing (incorporation of non-terminating nucleotides) artifacts. As these sources of noise are (to a certain degree) reflected by the quality values that are provided for each basecall, quality-based trimming of sequence data is the recommended and typically sufficient procedure for Illumina data preprocessing. A variety of tools is available for quality evaluation (Andrews *et al.*, 2010; Ewels *et al.*, 2016) and trimming (Bolger *et al.*, 2014; Schmieder and Edwards, 2011) purposes.

Illumina-based sequencing enabled scientists to obtain and analyze the largest metagenome datasets so far (Table 2.1), allowing them to gain deep insights into highly complex communities such as cow rumen (Hess *et al.*, 2011), the human gut microbiome (Huttenhower *et al.*, 2012), or microbiota occurring in natural soils (Vogel *et al.*, 2009).

Table 2.1: Metagenome sequencing effort. The availability of next-generation sequencing platforms and their continuous enhancement resulted in increasing dataset sizes, thus facilitating studies of even complex microbial communities.

year	study target	platform	size
2006	lean/obese mice (Turnbaugh *et al.*, 2006)	454	160 Mbp
2008	biogas fermenter (Schlüter *et al.*, 2008)	454	142 Mbp
2011	permafrost soil (Mackelprang *et al.*, 2011)	Illumina	39.8 Gbp
2011	cow rumen (Hess *et al.*, 2011)	Illumina	268 Gbp
2014	Human Microbiome Project (HMP; PRJNA48479)	Illumina	19.5 Tbp
2015	human gut (Bäckhed *et al.*, 2015)	Illumina	3.6 Tbp
2016	Tara Oceans protist (bioproject PRJEB4352)	Illumina	25.5 Tbp
2018	air metagenome (unpublished; PRJNA421162)	Illumina	5.5 Tbp

2.2.3 Third generation sequencing

2.2.3.1 Pacific Biosciences

Single molecule real time (SMRT) sequencing is a third-generation DNA sequencing technique developed by Pacific Biosciences ("PacBio"; Eid *et al.*, 2008). The approach is based on a DNA polymerase synthesizing the second strand of one single DNA template fragment which is monitored in real time. For this, a single DNA polymerase enzyme is immobilized at the bottom of a tiny hole in a 100 nm aluminum film deposited on top of a glass slide (Korlach *et al.*, 2008); the hole, a so-called zero-mode wave guide (ZMW), has a diameter smaller than the wavelength of visible light (390–700 nm), hence acting as a confinement and ensuring light is absorbed and cannot escape the ZMW.

Template double-stranded DNA is converted into a circularized, single-stranded loop by ligating hairpin adapters to both ends of the molecule. The circular DNA reaches the DNA polymerase at the bottom of a ZMW by diffusion, and replication is initiated after binding to one of the hairpin adapters. Strand synthesis uses nucleotides with fluorescent dye molecules attached to the phosphate chain (phospholinked nucleotides) as a substrate, and during incorporation, the fluorescent dye is released (Rhoads and Au, 2015).

A laser continuously shines into the ZMW from below of the glass slide, illuminating only its bottom region which contains the DNA polymerase, and excites the cleaved fluorophore before it diffuses out of the ZMW. The system thus detects single incorporation events as light pulses, and the correct nucleotide is identified based on the fluorophore's emission spectrum. As this process takes place in real time, variations in incorporation time (inter-pulse durations; IPDs) can also be utilized to detect epigenetic base modifications of the DNA template such as methylation.

Sequencing can be continued as long as the DNA polymerase is functional, generating a continuous long read (CLR); due to the circularization of the template DNA, smaller input fragments will be repeatedly sequenced in multiple passes, and the resulting long CLR can later be split into several subreads, which can be overlapped and merged into one shorter sequence with higher basecall accuracy (CCS; circular consensus).

For the PacBio RS II instrument, the sequencing operation is performed on SMRT cells which contain 150,000 ZMWs, and up to 75,000 ZMWs generate a sequencing read between 10,000 and 15,000 kbp during a run for a total output of approx. 1 Gbp. Its successor, the Sequel system, is able to operate SMRT cells with up to one million ZMWs. In a typical run, the RS II instrument will generate 50,000–75,000 reads on average, while the Sequel system provides approximately ten-fold output.

The main advantage of the technology is the ability to obtain very long reads, even if total output is quite low in comparison to previously established sequencers such as the instruments marketed by Illumina. As the sequencing approach directly monitors the activity of the DNA polymerase acting on a single template molecule, the technology also avoids possible biases introduced by amplification PCR steps. However, the technology is especially prone to insertion and deletion errors, with an overall basecall accuracy for single-pass CLRs of approximately 80–85% (Koren *et al.*, 2012); CCS reads offer higher accuracy, but at the cost of greatly reduced sequence lengths depending on the number of passes.

Thus, high coverage is required for applications like genome assembly in order to obtain a genome of reasonable quality; also, hybrid strategies employing *e.g.* a combination of Illumina and PacBio sequence data have been proposed, with Illumina data providing high accuracy and PacBio contributing sufficiently long reads to allow either scaffolding or the resolution of complex repeat structures (Rupp *et al.*, 2018; Miller *et al.*, 2017). Finally, different software packages have been published that perform either intrinsic (Salmela *et al.*, 2016) or hybrid (Hackl *et al.*, 2014) error correction of PacBio sequence data.

For microbial community analysis, the high error rate makes sequence data generated on PacBio instruments rather unsuitable for applications where appropriate accuracy is required; studies have already shown a poor community representation of PacBio data when used *e.g.* for 16S rRNA amplicon sequencing (Whon *et al.*, 2018); the same is valid for the interpretation of unassembled metagenome data, as the high error rates negatively affect commonly used approaches for taxonomic and function characterization. For metagenome assembly, however, long PacBio reads have already successfully been used to improve overall assembly quality of Illumina metagenome datasets (Frank *et al.*, 2016).

2.2.3.2 Oxford Nanopore

Employing nanopore technology is an alternative method to produce long-read sequence data from single DNA molecules. The sequencing instruments from Oxford Nanopore Technologies implement this approach with flow cells that contain an array of microscaffolds, and each of them is covered with an electrically resistant polymer membrane containing one protein nanopore.

The technique requires double-stranded template DNA molecules with adapters ligated to both ends and a motor enzyme bound to the 5′ end of one adapter (Jain *et al.*, 2016). During the sequencing process, a voltage is passed across the membrane, and thus, through the nanopore; the current attracts the negatively charged DNA and allows it to pass from the cis-side to the trans-side of the flow

cell, while a sensor on an ASIC[11] continuously monitors voltage changes across the polymer membrane. As unhindered pore traversal would occur too fast ($2 \cdot 10^6$ to $10 \cdot 10^6$ bp/s), this process is throttled by the motor enzyme, which limits pore throughput to approximately 450 bp/s (de Lannoy *et al.*, 2017). The DNA sequence of the template is derived based on the duration, mean amplitude, and variance of the voltage shifts during the passage of the different nucleotides.

Different sequencing protocols are available; for so-called 1D reads, only one strand of the template is passed through the nanopore, resulting in high throughput but lowered accuracy; for 2D reads with increased accuracy, a hairpin adapter is ligated to one end of the template, thus "connecting" the strands, and both of them are sequenced in succession; a consensus sequence with higher accuracy can then be computed merging the information from both strands. With $1D^2$, the successor of the 2D chemistry, both strands are passed through the nanopore independently; while the first strand passes through the pore, the complement strand remains attached to the membrane surface and later follows through the pore once it is unoccupied.

In contrast to PacBio, the sequencing approach by Oxford Nanopore does not allow to perform multiple passes over a single template molecule (CCS; circular consensus), as the DNA is physically located at the opposite trans-side of the polymer membrane after passing through the nanopore. However, like PacBio, Oxford Nanopore also allows for direct identification of methylation while sequencing, as methylation "causes a minute, but detectable, shift in current of a few picoamps" (Sedlazeck *et al.*, 2018).

A special and unique property of the technique is that it also allows to directly sequence RNA molecules without prior reverse transcription into cDNA (Garalde *et al.*, 2018), thus avoiding potential biases caused by the cDNA conversion step (Liu and Graber, 2006).

The Oxford Nanopore MinION is the smallest sequencing device manufactured so far, closely resembling a USB memory stick in size and weighing only 90 g; the corresponding flow cell features 512 nanopores. The device is able to deliver up to one million reads with lengths up to 100,000 bp for a total output of 10–20 Gbp. Due to its portability, it has already successfully been used for sequencing in the jungle (Pomerantz *et al.*, 2018), in the arctic (Johnson *et al.*, 2017), onboard the International Space Station (ISS) and in hotel rooms[12]. The GridION X5 instrument is a larger benchtop version and able to accommodate up to five MinION flow cells for each run; the device also incorporates an Intel-based compute server running Ubuntu Linux for analysis purposes. Finally, the PromethION is a modular benchtop

[11] Application-Specific Integrated Circuit

[12] `http://lab.loman.net/2016/07/30/nanopore-r9-data-release/`; NB: use of coffee as a heat source

sequencer, with a capacity of up to 48 flow cells comprising 3,000 sequencing channels each.

Similar to PacBio, the high error rate for long reads obtained with Oxford Nanopore instruments is dominated by insertions and deletions (indels). With several changes to both chemistry and bioinformatics software, the initial accuracy of 66% could be increased up to 92% (Ip *et al.*, 2015). Basecalling does not occur for individual nucleotides, but is instead performed for short overlapping k-mers (Jain *et al.*, 2016) between 3 and 6 bp, and a bias affecting certain k-mers, among them homopolymers, has been reported (Ashton *et al.*, 2015). In addition, at least one study reported the presence of modest though nonnegligible amounts of chimeric sequences for 1D reads generated on a MinION device (White *et al.*, 2017).

Interestingly, the Oxford Nanopore technology has already successfully been evaluated for low complexity synthetic metagenome data interpretation despite its high susceptibility to sequencing errors (Brown *et al.*, 2017). Methods employing k-mer-based schemes for taxonomic classification performed best, while MG-RAST, which relies on gene prediction and sequence alignment, "generally showed the lowest rate of correct taxonomic assignment". The authors did not report results for functional assignments; however, as functional classification pipelines almost exclusively rely on sequence homology searches, it can be assumed that unassembled Oxford Nanopore data would be rather unsuitable for such a purpose[13].

For metagenome assembly, on the other side, the long reads provide beneficial contextual information, contributing to improved overall assembly quality, especially when combined with high-quality data obtained with other methods.

[13]author's opinion

Table 2.2: Characteristics of selected sequencing machines. Average performance indicators of first-, second- and third-generation sequencing instruments.

Manufacturer	Device	Read length	Output reads	Output
ABI	3730*xl*	900 bp	384	3 Mbp
Roche/454	GS 20	100 bp	200,000	20 Mbp
	GS FLX	400–500 bp	600,000	300 Mbp
	GS FLX Titanium	600–700 bp	>1,000,000	700 Mbp
	GS Junior	700 bp	100,000	35 Mbp
IonTorrent	PGM	400 bp	4–5 million	1.2–2 Gbp
	Proton	200 bp	60–80 million	10 Gbp
Illumina	GA IIx	2×75 bp	138–168 million	25 Gbp
	MiSeq	2×300 bp	44–50 million	15 Gbp
	HiSeq 4000	2×150 bp	5 billion	1500 Gbp
	NextSeq 550	2×150 bp	800 million	120 Gbp
	NovaSeq 6000	2×150 bp	16–20 billion	6 Tbp
Pacific Biosciences	RS II	10–15,000 bp	50,000	0.5–1 Gbp
	Sequel	10–15,000 bp	500,000	5–10 Gbp
Oxford Nanopore	MinION	up to 100,000 bp	up to 1 million	10–20 Gbp
	GridION X5	up to 100,000 bp	up to 5 million	50–100 Gbp
	PromethION 48	up to 100,000 bp	up to 48 million	7.5 Tbp

2.3 Bioinformatics tools for the analysis of microbial community data

This section introduces some of the most commonly used or recent bioinformatics tools that have been used or specifically developed for metagenome analysis. The range of existing tools is too broad to cover all of them, thus only a subset of tools is described; these are presented in chronological order in order to demonstrate the technological advances over time, providing a brief outline of the algorithmic approach implemented by each tool as well as an assessment of its utility.

Also, only tools that can be installed and run locally were selected, excluding web-based services like *e.g.* CoMet (Lingner *et al.*, 2011) or One Codex (Minot *et al.*, 2015) that some might consider unsuitable to process confidential unpublished data.

Explicitly, tools relevant to data preprocessing and quality control are not within the scope of this section, as these are established standard procedures in the preparation of sequencing data for downstream analyses and not specific to metagenomics.

Generally, there are two prevailing approaches in metagenome data analysis:

1. **Alignment-based** tools like BLAST (Camacho *et al.*, 2009) or HMMER (Eddy, 2010) rely on sequence homology searches in order to deduce the taxonomic origin of a DNA fragment or predict the function of a suspected gene fragment. Widely used in genomics research, alignment-based methods were historically applied to metagenome analysis as no dedicated metagenome analysis tools existed yet. Nowadays, the large computational overhead inflicted by traditional local alignment-based methods like BLAST has contributed to their demise for purposes of taxonomic classification, but they remain popular for the functional characterization of metagenome sequences due to a lack of suitable alternatives. More recent approaches like Kaiju (Menzel *et al.*, 2016) or Centrifuge (Kim *et al.*, 2016) embrace technological advances and modern indexing schemes like the FM-index (Ferragina and Manzini, 2000) in order to provide a major speedup.

2. Alignment-free or **composition-based** methods like MetaCV (Liu *et al.*, 2012), TETRA (Teeling *et al.*, 2004) or PhyloPythia (McHardy *et al.*, 2007) attempt to infer information based on the intrinsic properties of DNA fragments, relying on characteristics like GC content, polynucleotide frequencies, or codon usage. Composition-based algorithms are frequently applied for taxonomic classification of metagenome sequences due to their superior throughput, but are rarely used for functional analysis.

With both approaches, metagenome sequences are typically compared to a database of either nucleotide or amino acid reference sequences or properties derived from such sequences of known origin. While some classification tools solely operate in nucleotide space, like *e.g.* Kraken (Wood and Salzberg, 2014), amino acid-based comparisons benefit from the higher degree of conservation on the protein level, thus contributing to improved accuracy, while at the same time offering increased resilience against certain types of sequencing errors due to the redundancy of the genetic code. On the other hand, only a fraction of metagenome sequences comprises actual gene fragments, thus this approach inevitably also results in a decreased sensitivity.

The alignment-based approach addresses the major shortcoming of k-mer based tools, the choice of the parameter k, which directly controls sensitivity as well as precision of the search. Chosen too large, no matching k-mers between the input sequence and the database will be found, either due to short metagenome fragments or sequencing errors, as well as for evolutionarily more distant input sequences.

On the other hand, if k is selected too small, too many matches will be returned, resulting in either more false positive classifications or assignment on a less specific taxonomic rank.

Interestingly, a lot of progress has been made in tools that address the taxonomic classification problem, and nowadays a scientist is able to choose from a wide range of tools that fulfill this purpose. Also, metagenomics studies have led to the discovery of new genera (Shivani *et al.*, 2017), families (Pillonel *et al.*, 2018), and sometimes even phyla (Eloe-Fadrosh *et al.*, 2016b; Parks *et al.*, 2017), and thus significantly contributed to an extension of the known tree of life.
On the other side, however, functional metagenome characterization remains a challenging task – a large number of proteins and protein functions remain unknown (Gilbert *et al.*, 2008), while at the same time existing sequence databases are to a certain degree biased towards genes found in organisms that were already cultivated, which is not really helping the cause. Due to the lack of suitable alternatives, scientists have once again resorted to the established methods from the genomics field, and apart from some minor exceptions (Meinicke, 2014), sequence similarity searches remain the prevailing approach for the functional characterization of metagenomes.

2.3.1 BLAST and derived methods (LCA, SOrt-ITEMS, CARMA3)

2.3.1.1 BLAST

Identifying homology between a sequence and possibly matching candidate sequences contained in a database has always been a central aspect in bioinformatic sequence analysis. In metagenomics, many algorithms perform taxonomic classification as well as functional analysis based on results of a homology search.

Originally published in 1990, the Basic Local Alignment Search Tool (Altschul *et al.*, 1990), BLAST, has since evolved into the de facto standard tool for sequence comparisons. Different modes of operation are available, allowing for the comparison of nucleotide and protein query sequences with both nucleotide and protein sequence databases, internally translating the sequences where required or explicitly requested.

To perform a sequence homology search, BLAST divides the query sequence into overlapping segments ("words") of fixed length, employing a database index to identify target sequences containing at least two of these words (exact matching) in correct order. Matching word pairs ("High Scoring Pairs", HSPs) are used as seeds and subsequently extended; then, two or more HSPs are joined into a longer

alignment using the Smith-Waterman algorithm for local alignment. This approach actually makes BLAST a heuristic algorithm, *i.e.* it does not guarantee to produce all alignments fulfilling the specified criteria (reduced sensitivity); however, an exact algorithm would be computationally far more expensive and was therefore deemed impractical for everyday use. Fidelity of an alignment generated by BLAST is indicated by a bit score S' and an E-Value E:

$$S' = \frac{\lambda S - \ln K}{\ln 2} \tag{2.1}$$

$$E = Kmne^{-\lambda S} \tag{2.2}$$

Initially, a raw score S is computed for a given alignment, which considers matching and non-matching positions as well as gaps. Different scoring systems may be used, ranging from simple edit distance to more sophisticated approaches taking into account the chemical properties of the corresponding amino acids. The bit score (Equation 2.1) is a rescaled version of the raw score and also incorporates information about the scoring system (λ) and the search space size (K). It serves as an estimate of the size of the search space required to yield an alignment with the very same raw score, allowing to evaluate whether the corresponding alignment is actually relevant or might have occurred just by chance. The E-Value (Equation 2.2) finally indicates the number of alignments expected to occur by chance with either the same or a better raw score. It is computed taking into account additional parameters representing the size of the query sequence (m) as well as size of the database (n).

Since its inception, different new features have been added to the BLAST program, allowing to filter out regions of low complexity or to provide composition-based scoring. Apart from the reference implementation, other versions like WU-BLAST (now AB-BLAST) or CS-BLAST have been developed, which improved or modified the original algorithm in certain areas.

In Camacho *et al.* (2009), the BLAST+ framework was announced by the NCBI, a major rewrite of the original BLAST application incorporating several improvements and new features. Introducing chunked processing of large sequences, BLAST+ was able to achieve a reduction of both run time as well as memory usage. More recently, several adaptations of the BLAST algorithm have been published, which aim to improve overall performance by employing specialized hardware to accelerate at least parts of the algorithm: GPU-BLAST (Vouzis and Sahinidis, 2011) and CUDA-BLAST make use of modern graphics processing units (GPUs), while others employ programmable hardware (Field-Programmable Gate Arrays, FPGAs) to speed up their implementation. A commercial supplier, TimeLogic[14] (Carlsbad, CA), mar-

[14] `http://www.timelogic.com/`

kets Tera-BLAST™ for their Xilinx-based DeCypher systems offering considerable speedup over the original implementation especially for protein database searches.

2.3.1.2 Use of BLAST in metagenome analysis

In metagenomics studies, BLAST has been widely applied for both the taxonomic analysis as well as functional assignment of metagenomic reads. As BLAST does not directly emit taxonomic classifications, different methods were developed to infer the taxonomic origin of a DNA fragment given the results of a homology search.

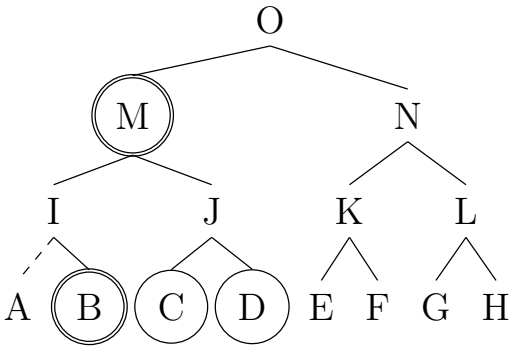


Figure 2.2: Illustration of taxonomic assignment strategies. Given a sequence of taxonomic origin A (which is absent from the database) and a list of homology results containing taxa B, C and D, the *"best-hit"* approach yields a wrong assignment to taxon B, while the *"lowest-common-ancestor"* algorithm generates an overly conservative assignment to taxon M.

Initially, quite naïve approaches have been used, where the taxonomic descent of a metagenomic sequence was simply determined to be identical to the biological source of the corresponding database sequence, provided that certain criteria (sequence identity, alignment length, E-Value threshold) were met. This *"best-hit"* approach (Figure 2.2), however, has shown to be quite problematic, as many of the species found in metagenome datasets were never analyzed beforehand and are thus not represented in sequence databases. Consequently, a large number of wrong assignments are the outcome of this method, typically resulting in too specific classifications.

Later on, Huson *et al.* (2007) proposed the *"lowest-common-ancestor"* (LCA) approach, which incorporates not just the best, but all BLAST database hits passing certain quality criteria. Herein, a metagenome sequence is assigned to the taxonomic entry which represents the lowest common ancestor of all database sequences returned for this query sequence (Figure 2.2). While this method greatly improves

onto the "best-hit" approach and reflects ambiguity by producing a less specific assignment, its main disadvantage is a negligence of alignment qualities – all database hits are considered equal, and no attempt is made to infer the classification using *e.g.* a weighted consideration of taxa. To some extent, the impact of this problem has been addressed by an additional filtering step, which relates the sequence identity of a given alignment to that of the best BLAST hit and removes all hits that do not fall within a certain range.

The SOrt-ITEMS (Sequence ORTholog based approach for binning and Improved Taxonomic Estimation of Metagenomic Sequences; Haque *et al.*, 2009) algorithm further improved upon this approach by also considering certain alignment parameters (length, bit score, identity) in order to determine the most specific taxonomic rank that may be assigned for a given alignment. SOrt-ITEMS also introduced an additional reciprocal BLAST search step of the highest scoring hit against a database containing all hit sequences as well as the initial query sequence. The final taxonomic assignment was then derived as the lowest common ancestor of all hits preceding the query sequence in the list of hits ordered by similarity. The SOrt-ITEMS algorithm offers increased precision in comparison to previous approaches, as the reciprocal BLAST search step was used to reduce the number of possible false positive predictions.

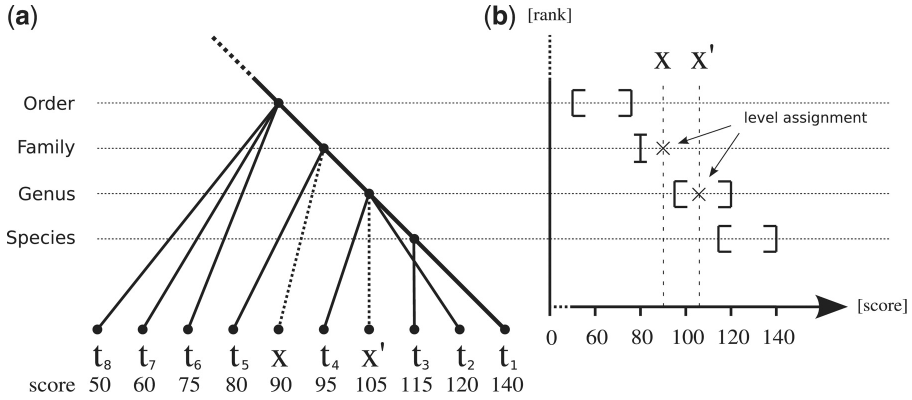


Figure 2.3: CARMA3 algorithm. Reciprocal BLAST hits are projected onto the taxonomic lineage and bit score intervals are computed to determine the taxonomic rank that should be used to classify the query sequence (Image source: Gerlach and Stoye, 2011, License: CC BY-NC 3.0).

CARMA3 (Gerlach and Stoye, 2011) is another algorithm extending the idea of employing a reciprocal BLAST step to further increase the precision over previous approaches. For this, CARMA3 internally creates a projection of the obtained reciprocal BLAST hits onto the taxonomic lineage of the query sequence in order to determine bit score intervals for each taxonomic rank. The final rank used to classify the query sequence is chosen based on the bit score it achieved in the reciprocal

search (Figure 2.3). In case no appropriate reciprocal hits are detected, CARMA3 falls back to a fixed set of predefined thresholds to determine the taxonomic rank.

In addition to the reciprocal BLAST search, CARMA3 also provides a variant based on HMMER3 (Eddy, 2010) employing the Pfam (Bateman *et al.*, 2004) database. As hmmscan does not directly support the alignment of DNA sequences to protein HMM models, CARMA3 performs the HMM search using the six-frame-translated query sequences. This approach lacks the capability to identify possible frameshifts and thus results in a decrease of both sensitivity as well as accuracy. However, a noteworthy property of this variant is the ability to derive Gene Ontology (GO) identifiers (Ashburner *et al.*, 2000) from the matched Pfam models.

2.3.2 Accelerated sequence homology search

With the increasing output of next-generation sequencing technologies, the application of BLAST for metagenome analysis purposes became more and more impractical. Not only did this development lead to larger metagenome datasets, but also resulted in an exponential growth of sequence databases that were commonly used for metagenome analysis. Several tools have been published to address this issue, all targeting to improve the throughput of the sequence alignment task (Table 2.3).

Table 2.3: Relative speed of BLAST and comparable homology search tools. In recent years, several methods have been published providing a magnitude of performance improvements over the original BLAST algorithm while retaining comparable sensitivity. Numbers were obtained from the relevant publications and are not necessarily comparable among each other.

Algorithm	Year	Relative speed
BLAST+	2009	1
RAPSearch	2011	90
RAPSearch 2	2012	180–270
GHOSTX	2014	150–160
GHOSTZ	2014	185–261
DIAMOND	2014	2,000–20,000
PALADIN	2017	8,000

RAPSearch (Ye *et al.*, 2011) and its successor, RAPSearch 2 (Zhao *et al.*, 2012), were among the first tools released for this purpose. RAPSearch uses the same seed-extend technique already present in BLAST, but employs variable-length seeds as well as a reduced amino acid alphabet of only ten symbols to achieve a 90-fold speedup compared to BLAST. RAPSearch used a suffix array (Manber and Myers, 1993) to index the sequence database, which resulted in rather high memory requirements; for RAPSearch 2, the indexing scheme was exchanged for a collision-free hash table, thereby significantly reducing memory usage while scoring an additional 2- to 3-fold speedup. Due to the usage of a reduced alphabet, both RAPSearch and RAPSearch 2 are restricted to the application in conjunction with protein sequence databases. For the very same reason, only the BLOSUM62 (Henikoff and Henikoff, 1992) scoring matrix is supported (Suzuki *et al.*, 2014a). According to the authors, both tools miss less than 5% of database hits in comparison to BLAST.

GHOSTX (Suzuki *et al.*, 2014a) is another approach employing the seed-extension algorithm to identify promising alignment candidates. GHOSTX uses suffix arrays for both query as well as database sequences to accelerate the identification of seeds and their ungapped extension, which is the most time-consuming step of the original BLAST algorithm. The high memory demand caused by the use of suffix arrays was partly mitigated by dividing the sequence database into equally sized chunks, which are processed sequentially during database searches. Like RAPSearch, GHOSTX uses variable-length seeds, extending them until a match score is exceeded. In their evaluation study, the authors demonstrated a significant speedup (150–160 times) compared to BLAST and still a modest acceleration (1.5 times) in comparison to RAPSearch 2, but with improved sensitivity.

GHOSTZ (Suzuki *et al.*, 2014b), published shortly after GHOSTX, was able to further accelerate the homology search by clustering the database into subsequences. GHOSTZ applies a method named similarity filtering where potential seeds are selected from the clusters' representative sequence and a lower distance bound between query subsequence and cluster member sequences is established as the distance between query subsequence and the corresponding subsequence of the cluster's representative sequence. Seeds exceeding a certain threshold are discarded, and only the remainder of seeds are passed on to the next step, which performs ungapped extension. While this filtering approach decreases sensitivity, GHOSTZ achieves a further acceleration of the original GHOSTX tool, with a reported 185–261-fold increase compared to BLAST.

DIAMOND (Buchfink *et al.*, 2014) makes use of spaced seeds (*i.e.* longer seeds with only some positions considered) and a reduced amino acid alphabet to speed up the detection of possible alignment candidates without reducing sensitivity. A set of four different seed shapes was chosen, and DIAMOND employs a double-indexing scheme for both query and database sequences in order to accelerate the seed identification step. Different modes of operation are supported, and while

the default parameters achieve a 20,000-fold speedup in comparison to BLAST, the more sensitive mode based on 16 seed shapes still accomplishes a 2,000-fold acceleration. To a certain extent, the memory usage can be adapted by the user, as DIAMOND processes query and database sequences in blocks of fixed size; thus, it is feasible to use DIAMOND even on modest hardware, but at the cost of a moderate slowdown.

PALADIN (Westbrook *et al.*, 2017) is a modification of the popular Burrows-Wheeler Aligner (Li and Durbin, 2009) which was adopted to provide mapping capabilities in protein space. PALADIN incorporates an internal ORF (Open Reading Frame) detection filter, which discards DNA fragments with stop codons in all six reading frames. The remainder of the sequences is translated in all possible frames and passed on to the alignment phase, which only retains and reports the result for the highest scoring frame, while possibly occurring additional protein alignments are dropped. The alignment phase is based on the BWA-MEM (Li, 2013) algorithm with some modifications (alphabet size, structure processing) to support protein sequence alignment. In their benchmark study, the authors were able to demonstrate a 8,000-fold performance increase of PALADIN in comparison to BLAST and a 7-fold speedup with regard to DIAMOND.

2.3.3 HMMER

The popular HMMER software (Eddy, 2010) follows a different approach to determine sequence homology. Unlike BLAST, which uses a database of completely unrelated sequences, HMMER databases contain a collection of profile Hidden Markov Models (pHMMs; Eddy, 2004). Hidden Markov Models have historically been widely applied in various areas, *e.g.* speech and text recognition, signal processing, or computer vision. In molecular biology, HMMs are used for gene prediction (Besemer and Borodovsky, 2005; Pedersen and Hein, 2003) or sequence homology search. In the latter case, which is implemented in the HMMER package, profile Hidden Markov Models (pHMMs, Figure 2.4) are employed to represent sets of conserved homologous sequences with equal biological function. Heuristic profile methods have already been suggested previously (Gribskov *et al.*, 1987, 1990), which directly converted the observed residue frequency into a position-specific scoring scheme (Durbin *et al.*, 1998). In contrast to these methods, profile HMMs offer the advantage of having a formal probabilistic basis, and statistical methods are applied to estimate true residue frequency instead of relying on observed frequencies only. They represent a system with an unobserved (hidden) internal state that is changing at random, while at the same time emitting symbols that can be observed from the outside.

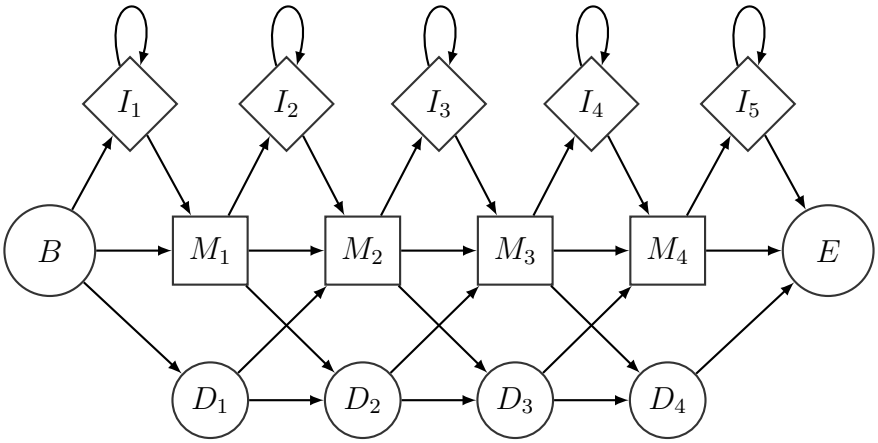


Figure 2.4: Topology of a Profile Hidden Markov Model. Starting with the initial state B, a series of state transitions allows to reach the end state E. The intermediate hidden states represent residue matches (M), insertions (I), and deletions (D).

Similar to profile methods, pHMMs are constructed based on multiple sequence alignments and are able to reflect position-specific conservation of nucleotides or amino acid residues (Figure 2.5) as well as position-sensitive gap scores; in addition, they are able to incorporate information about the likelihoods of different observations considering *e.g.* the total number of sequences partaking in a multiple sequence alignment.

Historically, HMMER used to be around 100-fold slower than BLAST (Eddy, 2011), but with a complete rewrite that was released as HMMER 3 and incorporated several algorithmic optimizations such as the multiple segment Viterbi (MSV) heuristic, it was able to reach processing speeds comparable to BLAST[15], while at the same time substantially increasing sensitivity over HMMER 2.

When applied to metagenome analysis, a major disadvantage of the HMMER package is the lack of a mode of operation that allows to directly compare nucleotide query sequences to models based on an amino acid alphabet, similar to what blastx provides for local sequence alignments. As this situation frequently occurs during the functional annotation of metagenome sequences, it is typically addressed by either creating all six possible protein translations (*e.g.* Boulund *et al.*, 2012; Walsh *et al.*, 2017) of the query sequence (accompanied with the corresponding computational overhead) or by applying gene prediction programs such as FragGeneScan+ (Kim *et al.*, 2015) or MetaGeneMark (Zhu *et al.*, 2010) which are capable of handling short, incomplete, and error-prone gene fragments.

[15]only valid for protein sequences; DNA searches use a different approach called SSV ("Single ungapped Segment Viterbi")

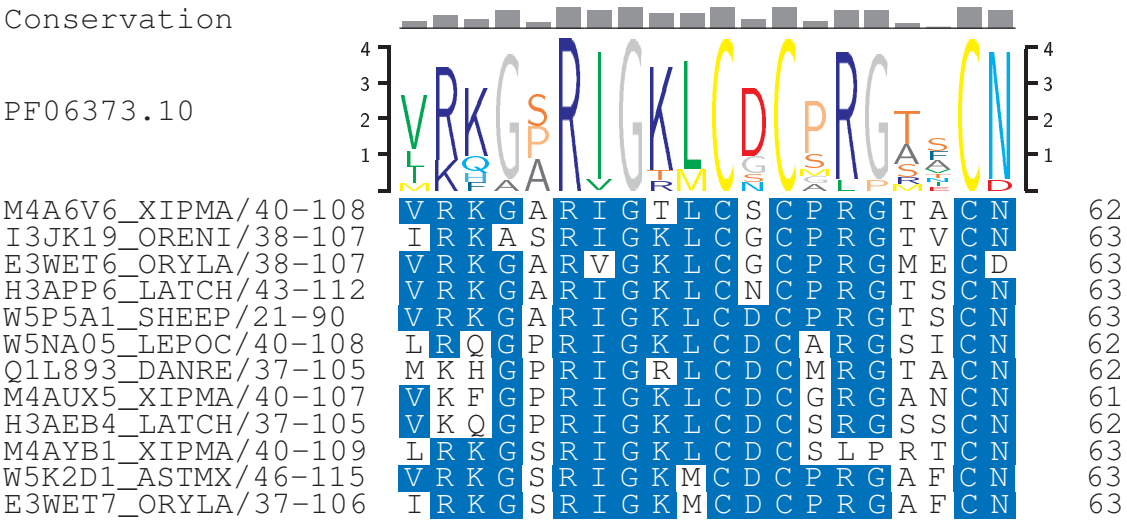


Figure 2.5: Sequence logo. Sequence logo for residues 43–62 of the Pfam protein domain PF06373.10 (Cocaine and amphetamine regulated transcript protein (CART)). Based on a multiple alignment of sequences (bottom), the derived sequence logo depicts position-specific residue frequencies as well as the amount of conservation.

2.3.4 RDP Classifier

The Ribosomal Database Project II (RDP; Cole *et al.*, 2013) maintains aligned and quality-controlled databases comprising bacterial and archaeal 16S rRNA and fungal 28S rRNA sequences as well as several tools for quality control, taxonomic classification and taxonomy-independent analysis of user-provided metataxonomics datasets.

Among these analysis tools is the RDP Classifier (Wang *et al.*, 2007), a naïve Bayesian classifier that assigns sequences to the Bergey taxonomy (Garrity *et al.*, 2004). The classifier is distributed with several different training sets, allowing taxonomic assignment of either 16S, 28S or ITS fragments down to the genus level. Classification is achieved using models that were trained with 8 bp oligomer frequencies, and a minimum fragment length of at least 50 bp is required; also, a confidence estimate is provided for each assignment based on 100 bootstrap iterations. While the RDP Classifier is typically applied for the analysis of metataxonomics data, it can also be employed in the scope of metagenomics projects for the taxonomic classification of 16S, 28S or ITS fragments after a preceding step that identifies and extracts reads bearing a fragment of the respective gene. For this filtering step, a sequence homology search versus the RDP database or a tool like SortMeRNA

(Kopylova *et al.*, 2012) may be used. An advantage of this approach is the highly discriminative power of the 16S rRNA gene as a taxonomic marker, however, this is opposed by the fact that only a very small fraction of metagenomic reads actually comprises a sufficiently long fragment of the ribosomal marker gene (McHardy and Rigoutsos, 2007).

2.3.5 MetaPhlAn

MetaPhlAn (Metagenomic Phylogenetic Analysis; Segata *et al.*, 2012) is a tool for the taxonomic profiling of metagenome datasets based on a set of clade-specific marker genes. Using 2,887 published microbial genomes obtained from IMG (Markowitz *et al.*, 2011), the authors extracted a subset of genes they identified to be most suitable to discriminate between different taxa. With the resulting reference database comprising approximately 400,000 marker genes, MetaPhlAn is able to distinguish between 1,221 different species. On average, 231 marker sequences are provided for each species, while the remainder of genes serves for taxonomic assignment to less specific ranks.

For each of the marker genes, the MetaPhlAn database contains the corresponding nucleotide sequence, and the actual taxonomic classification step is performed by mapping metagenome reads to the database using either blastn or Bowtie 2 (Langmead and Salzberg, 2012). As MetaPhlAn relies on a comparatively small database of reference sequences, it is able to process even large datasets in a very short time frame, while at the same time retaining an excellent precision.

MetaPhlAn 2 (Truong *et al.*, 2015) is an updated version of the MetaPhlAn tool. Its database has been extended to contain approximately one million marker gene sequences derived from 17,000 different genomes. While the original MetaPhlAn database was restricted to microbial taxa, MetaPhlAn 2 also incorporates eukaryotic as well as viral marker genes. In addition, a new feature was introduced allowing to perform strain level characterization for selected species and track identified strains across samples. Support for blastn was dropped, and MetaPhlAn 2 solely relies on Bowtie 2 for the mapping step.

As its predecessor, MetaPhlAn 2 offers high precision; however, due to the marker-based approach, it achieves a very low sensitivity, as only a minuscule fraction of input sequences contains a corresponding marker gene fragment and can successfully be assigned to a taxon.

2.3.6 Kraken

The Kraken sequence classification program (Wood and Salzberg, 2014) is one of the first software packages explicitly designed for metagenomics data analysis. Unlike previously used approaches that almost exclusively relied on sequence alignment (*e.g.* Huson *et al.*, 2007; Gerlach and Stoye, 2011), Kraken employs exact matches of short k-mers against a database of user-provided reference sequences. During the construction phase of a Kraken database, the reference sequences are processed and each k-mer is stored together with the taxon representing the lowest common ancestor of all genomes this k-mer occurs in. While Kraken uses a predefined k-mer length of 31 bp, this default setting can be overwritten by the user. However, this approach also implies that changes to the desired k-mer length require a complete rebuild of the database.

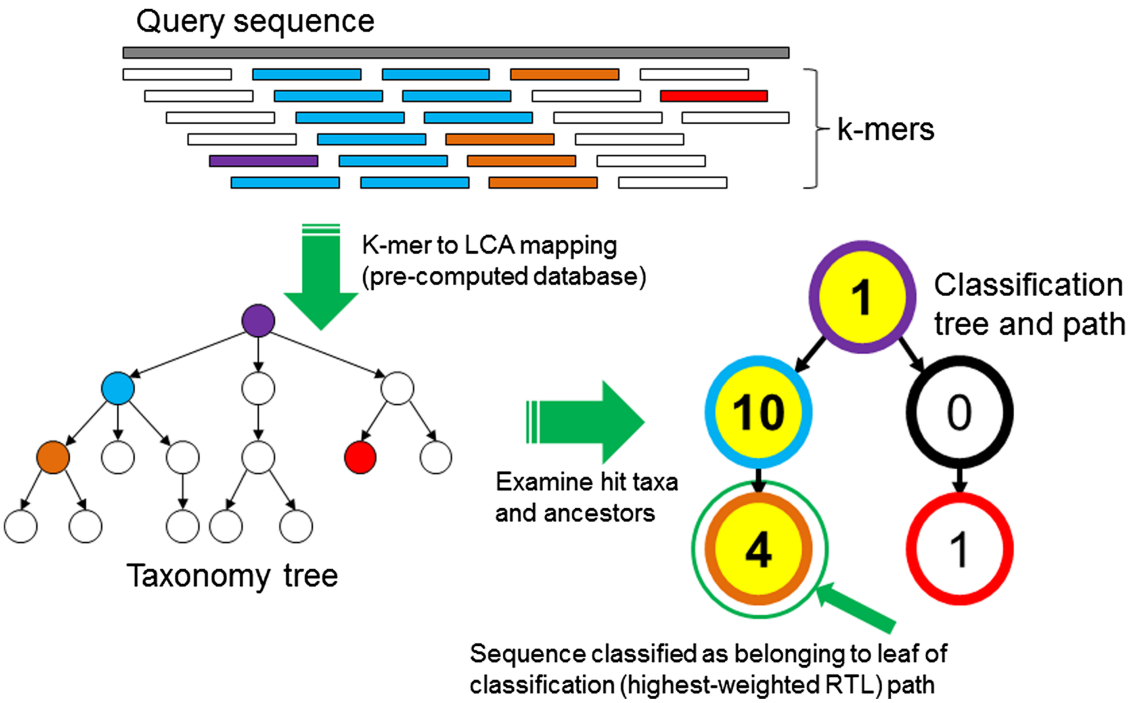


Figure 2.6: Kraken sequence classification. A weighted taxonomy subtree based on occurring k-mers is created and the query sequence assigned to the leaf of the highest-scoring root-to-leaf (RTL) path (Image source: Wood and Salzberg, 2014, License: CC BY 2.0).

For sequence classification, the query sequence is split into overlapping k-mers and each k-mer is mapped to its corresponding taxon; a taxonomy subtree is then constructed containing these taxa together with the frequency of their occurrence. Each root-to-leaf (RTL) path is weighted adding the number of k-mer occurrences, and finally, the sequence is assigned to the leaf taxon with the largest RTL path weight (Figure 2.6).

As Kraken avoids the computationally expensive sequence alignment step, it is able to process metagenomic sequences at a rate by far exceeding that of other taxonomic classification pipelines; according to the authors, Kraken was able to process 100 bp reads over 900 times faster than MegaBLAST, while still achieving sensitivity and precision "very close to that of MegaBLAST".

Seed-Kraken (Břinda *et al.*, 2015) is a minor Kraken variant which relies on spaced seeds instead of contiguous k-mers. Spaced seeds have initially been proposed within the context of sequence homology search (*e.g.* Ma *et al.*, 2002); a considerably increased sensitivity could be demonstrated when the authors compared spaced seeds to consecutive seeds as used for the initial exact matching phase of the BLAST (Altschul *et al.*, 1990) program. Even though the method itself is alignment-free, spaced seeds can also be interpreted as gapless alignments. The concept was later transferred to machine learning for the classification of string data, as well (Onodera and Shibuya, 2013); applied to the taxonomic classification problem as frequently encountered within metagenomics, Seed-Kraken consistently showed increased sensitivity and precision compared to the original Kraken implementation.

KrakenHLL (Breitwieser and Salzberg, 2018) is another Kraken variant that employs the HyperLogLog (Meunier *et al.*, 2007) cardinality estimation algorithm in order to determine the number of unique k-mers covered by metagenome data for each taxon. For taxa that are present in the studied microbial community, a uniform distribution of k-mers across the genome can be assumed, and a low number of unique k-mers is often a hint for either contamination or sequences originating from low-complexity genomic subregions. KrakenHLL thus addresses this possible cause for false positive taxonomic assignments, and by pruning taxa with a low number of unique k-mers allows to improve the overall precision.

Finally, Kraken has also been suggested for functional metagenome analysis instead of taxonomic classification, *e.g.* for the detection of antibiotic resistance genes (`https://github.com/fbreitwieser/card-krakendb`).

2.3.7 Kraken 2

For the recently released Kraken 2 software, no manuscript has yet been published describing its implementation details and performance. However, from the corresponding documentation and the source code, it is already known that Kraken 2 no longer stores a database consisting of k-mers with associated taxa; instead, the k-mers have been replaced by their minimizers, which are the lexicographically smallest short ℓ-mers within a k-mer (*i.e.* $\ell \leq k$), thus significantly reducing database size.

A database built from the viral, archaeal and bacterial complete genomes obtained from NCBI RefSeq (Pruitt *et al.*, 2006) occupies 178 GB disk space when built for Kraken (1.x), but only 27 GB for Kraken 2; at the same time, results obtained from Kraken 2 remain comparable to those generated by the first version.

Also, Kraken 2 "utilizes spaced seeds in the storage and querying of minimizers to improve classification accuracy"[16]. Finally, Kraken 2 also allows to create databases derived from protein sequences, while the original version only supported nucleotide data. For this, a reduced amino acid alphabet is used and database matches from six-frame translated query sequences are internally combined into a single classification result.

2.3.8 Kaiju

Kaiju (Menzel *et al.*, 2016) performs taxonomic classification of metagenomic datasets based on alignments of metagenome reads against a database of protein sequences. For this, an indexing scheme based on the Burrows-Wheeler transform (Burrows and Wheeler, 1994) and an FM-index (Ferragina and Manzini, 2000) is used to identify maximum exact matches (MEMs) between the six-frame-translated metagenome reads and the reference database containing genes with known taxonomic origin; for evolutionarily more distant sequences, Kaiju also offers a greedy mode where these exact matches are subsequently extended at the left end only allowing for a certain number of substitution errors.

Kaiju finally assigns query sequences to the source taxon of their corresponding database hit, if only one hit is found, or to the lowest common ancestor in case several hits fulfill the minimum required match length.

For the benchmark data analyzed in the manuscript describing Kaiju, the authors report "higher sensitivity and similar precision compared with current k-mer-based classifiers" (Menzel *et al.*, 2016).

2.3.9 Centrifuge

Centrifuge (Kim *et al.*, 2016) follows a similar approach as Kaiju, making use of the Burrows-Wheeler transform and the FM-index for memory-efficiency and fast database searches. For this, Centrifuge relies on a data structure adapted from the popular Bowtie 2 (Langmead and Salzberg, 2012) read-mapping software, which was developed within the same group.

[16] `https://github.com/DerrickWood/kraken2/blob/master/docs/MANUAL.markdown`

Centrifuge significantly reduces the size of the reference database, employing an iterative compression/deduplication scheme. Within a group of related genomes, a k-mer-based approach is applied in order to identify the two most closely related sequences, and subsequences of the second genome that already appear in the first one with at least 99% sequence identity are pruned before the remaining sequence data is added to the index. Iteratively executed, this means that only sufficiently "novel" information is added to the index; on the other hand, this also means the order in which reference sequences are added directly influences the final contents of the index. Employing this compression scheme, the authors were able to achieve a space reduction of almost 40% for a reference set of approximately 4,300 bacterial and archaeal genomes.

For classification, the FM-index is used to identify short exact matches between the query sequence and the database, which serve as seeds and are subsequently extended as far as possible. The taxonomic origins for all database hits are then scored, taking into account the number of exact matches as well as their respective lengths. Centrifuge is able to assign up to five different taxa to each query sequence; if more taxa are detected, they are iteratively merged, replacing the largest subgroup of taxa with their lowest common ancestor.

Within their evaluation based on a synthetic metagenome generated from complete genomes contained in the NCBI RefSeq (Pruitt *et al.*, 2006) database, the authors compared Centrifuge to Kraken as well as MegaBLAST; even though MegaBLAST showed the highest sensitivity as well as precision among all tools on both genus and species level, its low throughput still makes it unsuitable for routine metagenome analysis. Centrifuge demonstrated better sensitivity than Kraken on both taxonomic ranks at the cost of a slight decrease in precision, but while Kraken processed sequences almost twice as fast as Centrifuge, the latter one required only a fraction of the memory that was occupied by Kraken.

2.4 Metagenome analysis platforms

Most bioinformatics tools are nowadays released as command line tools for the Linux operating system; also, significant compute resources are often necessary, especially for metagenome analysis. In order to make these tools and applications available to less technically proficient scientists or to users without access to adequately potent compute resources, several metagenome analysis platforms have been developed. These are typically accessible via a web-based interface or a GUI and offered free of charge to the scientific community. There are two prevalent service types – offerings intended to promote use of a single tool, such as WebCARMA (Gerlach *et al.*, 2009) or the PhyloPythiaS web server (Patil *et al.*, 2012), and fully integrated solutions that provide a complete pipeline covering taxonomic, functional and statistical

aspects of metagenome characterization. Within this section, only the most popular examples of the latter category will be presented.

2.4.1 MG-RAST

MG-RAST (Metagenomic Rapid Annotations using Subsystems Technology; Meyer *et al.*, 2008), first released in 2008, is by far the most widely used web-based application for the analysis of metagenome datasets. The software features a pipeline suitable for assembled as well as unassembled metagenomes which incorporates quality control, contamination removal, taxonomic classification and functional analysis.

After initial metadata and sequence upload, quality control is performed, which includes adapter removal, barcode demultiplexing, dereplication and optional removal of host contamination based on a predefined list of reference sequences. DRISEE (Keegan *et al.*, 2012) is used to establish a dataset-specific measurement of internal sequencing error based on the examination of near-duplicate reads.

Taxonomic profiles within MG-RAST are computed using an LCA-based approach based on both ribosomal as well as protein coding sequences. Detection of ribosomal RNAs previously relied on sequence alignment versus a reduced database (M5rna) generated from SILVA (Pruesse *et al.*, 2007), Greengenes (DeSantis *et al.*, 2006) and RDP (Cole *et al.*, 2013), but this step was recently replaced by the SortMeRNA (Kopylova *et al.*, 2012, 2014) tool for performance reasons[17]. Identified rRNAs are subsequently clustered (at 97% sequence identity) and finally assigned to taxa using BLAT versus the M5rna database.

For protein-level analysis, FragGeneScan (Rho *et al.*, 2010) is used to predict gene fragments in metagenomic sequence data; predicted genes are then annotated based on sequence homology results generated with sBLAT, a parallelized version of the BLAT (Kent, 2002) tool. BLAT is approximately 50 times faster than BLAST, however, its search sensitivity is reportedly much lower than that of BLAST (Suzuki *et al.*, 2014a), and more distant homologues are likely to be missed. For the database search step, the MG-RAST developers have implemented their own non-redundant protein database, the M5nr (Wilke *et al.*, 2012), which incorporates sequence and annotation data from a variety of different sources such as UniProt (Wu *et al.*, 2016), KEGG (Kanehisa *et al.*, 2014), or SEED (Overbeek *et al.*, 2005) (Table 2.4).

[17] `http://blog.mg-rast.org/2017/05/mg-rast-pipeline-402-released-with-new.html`

Table 2.4: Sources for MG-RAST databases. The MG-RAST project maintains two distinct databases for protein (M5nr) as well as ribosomal fragment (M5rna) classification. (Modified after Wilke *et al.* (2012)).

Database	Source	Description
M5nr	GO	Gene Ontology
	IMG	Integrated Microbial Genomes
	KEGG	Kyoto Encyclopedia of Genes and Genomes
	NCBI	NCBI RefSeq & GenBank
	SEED	The SEED Project
	eggNOG	Evolutionary Genealogy of Genes: Non-supervised Orthologous Groups
	PATRIC	Pathosystems Resource Integration Center
	UniProt	UniProt Knowledgebase
M5rna	RDP	Ribosomal Database Project
	SILVA	Aligned ribosomal RNA sequence data
	Greengenes	Chimera-Checked 16S rRNA Gene Database

In addition to taxonomic profiles and functional assignments, quality control reports, species-based rarefaction curves and biodiversity indices are available. All results are offered via a web-based interface and can also be downloaded in a variety of file formats; additionally, MG-RAST provides a REST[18]-based API that allows programmatic access to projects, metadata, and analysis results (Wilke *et al.*, 2015).

As MG-RAST is a heavily used service, processing typically takes several weeks even for small datasets; however, users may choose to make their metagenomes publicly accessible right away, which is rewarded with higher job prioritization and thus faster turnaround times by the system.

Even though MG-RAST is not an official INSDC (Cochrane *et al.*, 2015) repository, many publications from the field provide MG-RAST project identifiers instead of depositing data into one of the established sequence archives such as the NCBI SRA (Leinonen *et al.*, 2010b).

[18]Representational State Transfer

2.4.2 IMG/M

IMG/M (Integrated Microbial Genomes and Microbiomes; Markowitz *et al.*, 2012) is a web-based[19] application initially created for the annotation of genomes sequenced within the Joint Genome Institute (JGI). The system was later extended in order to provide capabilities to process metagenome data, as well, and the MGAP[20] pipeline for isolate genomes was adapted to build MAP, the Metagenome Annotation Pipeline.

MAP (Huntemann *et al.*, 2016) performs data preprocessing (quality control) including quality- as well as sequence complexity-based filtering; in addition, a dereplication step is applied to 454 pyrosequencing data. For metagenome annotation, CRISPR[21] elements, ribosomal RNAs, and tRNAs are predicted using a sophisticated pipeline built from PILER-CR, CRT, tRNAscan-SE and HMMER 3. For protein coding genes, GeneMark.hmm, MetaGeneAnnotator, Prodigal and FragGeneScan are used.

Taxonomic assignment of sequences is performed based on a *lowest-common-ancestor* approach using USEARCH (Edgar, 2010), and functional analysis consists of annotations derived from database comparisons with COG, KEGG, Pfam and the in-house IMG NR database. In addition, users can manually define metagenome bins, *e.g.* based on taxonomic assignments of assembled contigs, and use these as subsets for more detailed analyses.

Since 2017, submissions outside of the JGI have been restricted to preassembled data, and read-based metagenome analysis is no longer offered as a public service (Chen *et al.*, 2017).

2.4.3 EBI MGnify

MGnify (Hunter *et al.*, 2014; Mitchell *et al.*, 2017) is a service for metagenome analysis provided by the EBI (European Bioinformatics Institute) in tight collaboration with the EBI ENA (European Nucleotide Archive; Leinonen *et al.*, 2010a). Unlike other applications, a direct upload of metagenome datasets is not supported; instead, it is necessary to deposit the sequence data and associated metadata into the ENA archive.

Initial preprocessing steps include trimming of low-quality bases with Trimmomatic (Bolger *et al.*, 2014) and removal of adapter sequences; optionally, overlapping

[19] `https://img.jgi.doe.gov/m/`
[20] Microbial Genome Annotation Pipeline
[21] Clustered Regularly Interspaced Short Palindromic Repeats

paired-end sequences can be merged with the SeqPrep[22] program. The system requires a minimal read length of 100 bp, and shorter sequences are automatically pruned from each dataset.

For taxonomic analysis, non-coding RNAs (ncRNAs) are identified based on corresponding covariance models from the Rfam database (Nawrocki *et al.*, 2014) with the Infernal (Nawrocki and Eddy, 2013) package. While no further processing is performed for tRNAs, RNase P and signal recognition particle RNAs, ribosomal RNAs are used to create taxonomic profiles using the MAPseq (Matias Rodrigues *et al.*, 2017) program, which maps the sequences to the SILVA database; classification is available for both prokaryotic organisms as well as eukaryotes.

For functional profiling, genes and gene fragments are predicted using either Prodigal (Hyatt *et al.*, 2010) for assembled metagenome contigs or FragGeneScan (Rho *et al.*, 2010) for unassembled short reads. Predicted genes are analyzed with InterProScan (Zdobnov and Apweiler, 2001) using only a reduced subset of the available databases (Pfam, TIGRFAMs, PRINTS, ProSitePatterns, and Gene3d); this subset was chosen both for performance reasons as well as due to the ability to accurately detect and assign features to incomplete gene fragments. Based on the detected InterPro matches, GO terms are assigned from a reduced GO ontology list.

The MGnify web site[23] provides an overview of each dataset together with statistics describing the quality control results as well as interactive charts for the taxonomic and functional analysis results. Taxonomic classifications, functional assignments and sequences grouped by assigned category (tRNA, rRNA, predicted CDS) are also available for download in CSV, JSON and FASTA formats. A REST-based API is also offered for programmatic access, allowing to retrieve metadata as well as analysis results for each dataset.

2.4.4 CyVerse

CyVerse (Merchant *et al.*, 2016), formerly known as iPlant Collaborative, is a virtual infrastructure for data management and analysis. The CyVerse DE (Discovery Environment) is a web-based environment offering a wide range of bioinformatics tools for sequence analysis in the form of packaged "Apps". Even while the application is not primarily targeted at metagenomics, at least some software packages are provided for data preprocessing and microbial community analysis, among them FastQC, MetaPhyler, QIIME, or Centrifuge.

[22] `https://github.com/jstjohn/SeqPrep`
[23] `https://www.ebi.ac.uk/metagenomics/`

The associated SciApps component provides complete workflows for data analysis, but the current offering mostly revolves around genome annotation, RNA-Seq and ChIP-Seq analysis, and there are no predefined analysis pipelines available for metagenome processing. Thus, only the Apps within CyVerse DE can currently be used for microbial community analysis, and users of the infrastructure have to know beforehand which software is suitable for their purpose.

While CyVerse offers appropriate compute resources even for large-scale data analysis, no visualization options are available for metagenome interpretation, and the included software packages provide textual output only.

2.4.5 MEGAN

MEGAN (MEtaGenome ANalyzer; Huson *et al.*, 2007) is a desktop application for the visualization of homology-based metagenome analysis results. The software is implemented in Java and distributed as a standalone tool for Mac OS, Windows and Linux, but an add-on component, MeganServer, allows to share data between several installations.

MEGAN does not include quality control or preprocessing capabilities, and these tasks must be performed by the user outside of the application. Also, MEGAN features no own metagenome analysis pipeline; instead, BLAST-compatible output has to be provided in order to use MEGAN, thus requiring significant compute resources; in the meanwhile, this issue has been partially mitigated with the DIAMOND sequence alignment program, which greatly reduces computational effort, but still requires appropriate hardware and a Linux-based operating system to execute the analysis portion.

Taxonomic annotations within MEGAN were initially created in an LCA-based manner, while several mapping files can be downloaded and installed in order to provide functional result mappings to InterPro, eggNOG, SEED, and partially, an old legacy KEGG release from 2011. With the current version, MEGAN CE (Community Edition), the LCA method has been replaced by a weighted LCA approach (Huson *et al.*, 2016).

While MEGAN does not support whole metagenome assembly, an interesting feature is a gene-centric assembler, which allows to manually extract metagenome sequences annotated with certain genes of interest and subsequently assemble them in order to possibly recover corresponding full-length sequences.

A premium commercial version of MEGAN 6, called Ultimate Edition, offers "additional features, tools and support" according to the MEGAN homepage, among

them support for mappings to current KEGG pathways for users that already own a paid KEGG subscription.

2.5 Preliminary conclusions

Ever since scientists have begun to study microbial communities, technological advances in both sequencing methods as well as the set of tools available for successful data interpretation have largely contributed to improve our understanding of these interacting systems. Even more, in many cases these developments were the crucial key prerequisites that for the first time enabled us to access these valuable resources at all and gain important novel insights. Some natural environments have been shown to be inhabited by highly complex microbial communities, for which even current metagenome sizes do not provide sufficient coverage; *e.g.* soil:

> "Based on our analysis, we propose that the sequencing depth required to provide comprehensive coverage of soil metagenomes should be increased by an order of magnitude, to ∼100 Gbp. This is a function of the extreme taxonomic heterogeneity of soil microbial communities ..." (Van der Walt *et al.*, 2017).

Especially the advent of next-generation sequencing resulted in a dramatic decrease of sequencing costs, without which a large number of studies would not have been possible. Ever since, unprecedented amounts of sequence data have been generated, and while data interpretation yielded a large number of novel findings, there were also new types of error discovered and new biases which needed to be addressed.

With third-generation sequence data, the next round of adaptations and novel opportunities has arrived – the ability to sequence long DNA stretches favors metagenome assembly, while the decrease in basecall accuracy at the same time rather obstructs established read-based approaches. As a consequence, new approaches are needed to successfully analyze these metagenome datasets.

Bioinformatics tools rapidly conquered the new and evolving metagenomics field, and while initially tools typically used for genome analysis were employed, a researcher is nowadays able to choose from a wide range of software packages that were specifically developed for the processing and analysis of microbial community data. Within a short time frame, a large number of methodological advances have been made; but as the corresponding implementations are almost exclusively provided as command-line tools for the Linux operating system, they are thus only accessible to tech-savvy people with the corresponding knowledge. In addition, the computational effort required for metagenome analysis still remains quite large de-

spite these developments, and without access to appropriate compute infrastructure, a timely data analysis is not attainable.

Platforms like IMG/M and MG-RAST avoid this burden, providing easily accessible web interfaces in combination with sizeable compute resources. However, these applications trade ease of use for limited customizability – only a single analysis pipeline is offered allowing to execute a rather generic and often insufficient analysis, parameters can not be adapted, and visualization/charting capabilities are typically restricted to certain result types. Also, these pipelines are rarely updated and thus often rely on outdated tools (*e.g.* FragGeneScan instead of the far more recent FragGeneScan+; sequence alignment and best-hit annotation instead of faster and more flexible tools like Kraken) or software components with known deficiencies (Eren *et al.*, 2013). Finally, they do not allow to include own data sources (*e.g.* limited predefined set of reference genomes for host contamination removal) or lack the ability to define custom analysis pipelines to address less common use cases, such as the processing of eukaryotic data.

These static, "one-size-fits-all" pipelines have clearly been designed in order to cover the most common use cases, and while the overall throughput and quality of results is quite impressive, they do not suffice to address specific analysis needs where specialized sequence databases are required, or provide suboptimal performance when exposed to less common sequence data types (Brown *et al.*, 2017).

3

MGX: An advanced framework for microbial community analysis

The first 90 percent of the code accounts for the first 90 percent of the development time. The remaining 10 percent of the code accounts for the other 90 percent of the development time.

– Tom Cargill, Bell Laboratories

This chapter describes the design and implementation of the MGX framework for metagenome analysis and its accompanying components. Contents are partially based on the original publication describing the MGX framework (Jaenicke *et al.*, 2018) without further explicit attribution.

3.1 Objectives

Based on the interim conclusions given in Section 2.5, it is apparent that metagenome data analysis is a fast-moving field that requires frequent adaptation to keep up with novel developments in sequencing technologies as well as latest methodological advancements in data analysis. The MGX framework has been designed in order to address the shortcomings of other applications for metagenome analysis currently in existence and to provide an environment that can easily be adjusted to future developments.

On several occasions (Su *et al.*, 2011; Zakrzewski *et al.*, 2013), newly developed platforms soon fell into obsolescence once it became clear that neither were they able to cope with current sequence data volumes nor did their algorithmic approach provide sufficient scalability, disallowing an adaptation to changed external conditions. Hence, modular approaches are preferable over fixed systems, as they allow to dynamically improve or exchange components when better tools are available or sequencing strategies change.

The primary target of the MGX software is the storage and analysis of unassembled metagenome sequence data; however, possible future extensions have already been taken into account, and support for *e.g.* metataxonomics or metagenome assembly is attainable without major structural changes to the underlying data model.

Consequently, the main objectives for the development of the MGX framework have been defined as follows:

Data and metadata storage. For each dataset, the sequence data ought to be stored in conjunction with corresponding metadata detailing origin and treatments applied for data generation (also see Section 3.3), thus facilitating the repetition of an experiment.

Fast adoption of novel tool developments. Newly developed tools should be provided as readily available analysis pipelines as soon as possible in order to allow users to benefit from improved methods, or to use the tool most suitable for their type of data.

Single sequence resolution. It is desirable to retain all analysis results in a manner that allows to identify and extract subsets of the available sequencing data based on arbitrary criteria. Therefore, results should mandatorily be traceable to the individual input sequences.

Abstraction to allow use without bioinformatics expertise. Most bioinformatics tools are prevalently invoked using the Linux/UNIX command-line and

therefore typically inaccessible to users without at least basic Linux expertise; UI (user interface) components should provide a sufficient degree of abstraction to enable users to configure and execute tools without Linux proficiency.

Provisioning of compute resources. Considerable resources are still necessary to run recent metagenome analysis tools; due to the required hardware investments and associated maintenance costs, compute infrastructure used by MGX should be provided centrally and without imposing any costs on (academic) users.

Multiple specialized workflows for different tasks. Instead of one central pipeline for taxonomic as well as functional metagenome analysis, specialized and modular workflows should be provided, thus permitting users to restrict their selection to those aspects they are interested in; also, this approach eases future adaptations and improvements.

Parameter customization. All workflows should allow to adapt certain parameters (such as cutoff values or scores), where applicable. Nonetheless, each possible parameter should still provide a predefined value, which serves as a sensible default and can be used as a starting point for possible future refinements.

Dynamic visualization options. Instead of fixed visualizations for specific results, a dedicated component should automatically determine whether a user-selected result type and a certain mode of presentation are compatible and offer only appropriate combinations.

Comparative analysis and statistics. MGX should provide different analysis types allowing to interactively perform comparisons between several datasets as well as execute statistical evaluation using state-of-art methods such as compositional data analysis.

Ability to implement and execute own workflows. For advanced users or scientists with highly specific analysis needs, it should be possible to draft and implement own analysis workflows, which are subsequently scheduled and executed on MGX-provided infrastructure.

Possibility to include own databases and reference genomes. Users should be able to upload and include own datasources for metagenome analysis, *e.g.* specialized sequence databases or unpublished reference genomes. Thus, it is desirable to allow the inclusion of arbitrary data and provide predefined workflow templates for the most frequently used data formats, such as *e.g.* FASTA-based sequence collections or HMM models.

Modular design to allow adaptation to novel developments. A modular software design greatly reduces the required effort needed to exchange individual components as long as the interface remains constant.

API provisioning. For programmatic access and automated data analysis, an API (Application Programming Interface) should be provided together with an appropriate library for command-line usage.

3.2 System architecture

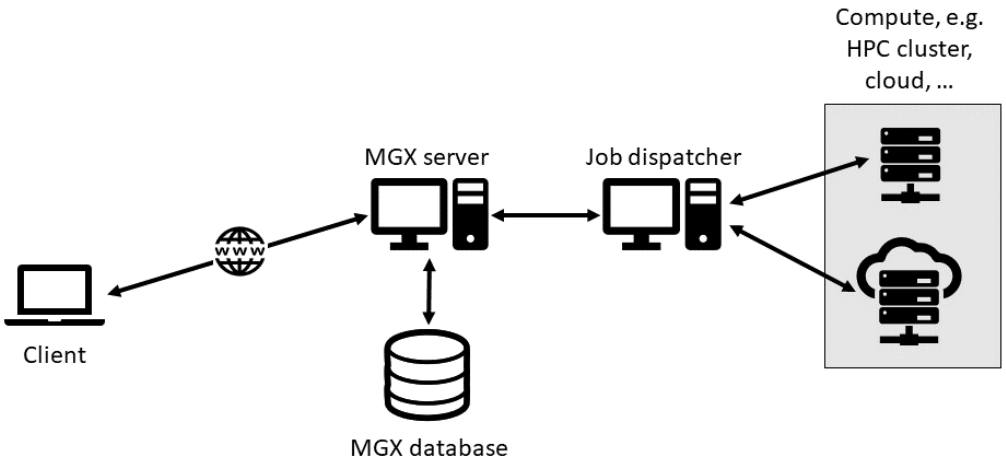


Figure 3.1: MGX system architecture. MGX has been implemented as a distributed application, where multiple clients connect to one or several server instances. All project data is securely and centrally stored within the MGX database, and compute resources are provisioned for the timely execution of analysis pipelines.

MGX has been designed as a distributed application following the traditional and established client/server model (Figure 3.1). The server provides centralized data storage and access to compute resources, while the clients take a mere steering role, requesting data creation, retrieval, modification and deletion as well as initiating the execution of analysis tasks.

An associated job dispatcher takes care of prioritizing and scheduling analysis tasks, manages pre- and postprocessing steps and can temporarily queue jobs, *e.g.* during planned hardware maintenance of the compute infrastructure. This separation of concerns contributes to enhanced scalability; also, the dispatch mechanism

is agnostic towards the job type, allowing potential re-use for other applications apart from MGX, as well.

Mostly for portability reasons, Java was chosen for the implementation of the various components such as the command-line client, the graphical user interface, as well as the server and dispatcher codebase; the GUI makes use of the NetBeans Platform, a modular rich client framework with support for an own proprietary module format (NBM; NetBeans Module) as well as the standardized OSGi[1] specification. The MGX server and dispatcher have been implemented as Java EE[2] applications; the Java EE framework readily supports context and dependency injection, messaging services, concurrency and also provides an API for the development of REST-based web services. Oracle™ GlassFish, the reference implementation of the Java EE specification, was chosen as the application server to deploy the MGX components. Finally, each MGX server manages an associated Rserve (Urbanek, 2003) instance, which is employed to perform statistical evaluation of analysis results.

3.3 The importance of metadata

A variety of different sources have pointed out the importance of metadata, *e.g.* Field *et al.* (2008), which should always accompany a dataset; even while exact reproduction of a study is often impossible, the metadata is intended to provide sufficient information about a dataset's provenance and processing steps that were performed to create it that a repetition of the experiment is made possible.

A proposed list of parameters that should be recorded for metagenome sequence data is provided by Anne and Ann (2007):

"
- Detailed, three-dimensional geographic location of the sample, including depth (for water sampling) or height (for land and air samples).
- The general features of the environment of the sample, such as ocean, soil, mine, human, or insect.
- Specific features of the sample site, such as chemical data (pH, salinity, and so on), physical data (temperature, incident light, and so on), time when the sample was taken, and host condition, diet, and habitat.
- Method of sampling, size of sample, and sample preparation."

[1] Open Services Gateway initiative
[2] Java Enterprise Edition

The Genomic Standards Consortium[3] (GSC) is a community-driven initiative striving to establish standards for genomic datasets in order to ease interoperability, future data discovery and integration. The consortium has published several specifications, detailing, among other standards, the minimum information about genomic sequences (MIGS; Field *et al.*, 2008), marker genes (MIMARKS), or any (MIxS; Yilmaz *et al.*, 2011) sequence. For metagenomic data, the corresponding GSC checklists propose a set of basic metadata items to be recorded, and 15 extensions called "environmental packages" are currently available that suggest habitat-specific additional data to be collected for, among others, soil, water, artificial and host-associated metagenomes as well as datasets from the built environment.

This important aspect was incorporated into the design of the MGX data model, which – following the recommendations given by Anne and Ann (2007) – associates each dataset with corresponding metadata describing the habitat, sampling procedures, DNA extraction protocols as well as the sequencing technique that was applied to generate it (Figure B.1). Metadata is always stored in combination with the sequence data, and MGX enforces metadata entry before allowing to upload any actual sequences. While some metadata properties are inevitably stored as free text, MGX provides a set of ontology categories and terms, which are used to provide a predefined vocabulary for certain fields, *e.g.* the chosen DNA extraction method, sequencing platform, and sequencing strategy. Where possible, user-provided values are checked for their validity, *e.g.* valid ranges for SI units.

3.4 Server

The implementation of the MGX server follows the classical three-tier architecture (Figure 3.2); business logic operates on and modifies data residing within the storage backend or initiates external processing steps. All relevant interactions with the system are exposed via the presentation layer, which takes care of authentication, authorization, message routing and object marshalling/unmarshalling.

During operation, an MGX client connects to one or several MGX application servers, which provide centralized storage and compute resources for all hosted projects, managing project access and resource assignment. All calculations and the execution of analysis pipelines are therefore performed not on the client, but on the server and an associated compute cluster, thus eliminating the need to establish and maintain a local compute infrastructure. The server also features dedicated storage resources allocated to each project, allowing users to upload own data sources to be included into analysis pipelines, *e.g.* own sequence collections, databases or unpublished reference genomes.

[3] `http://gensc.org/`

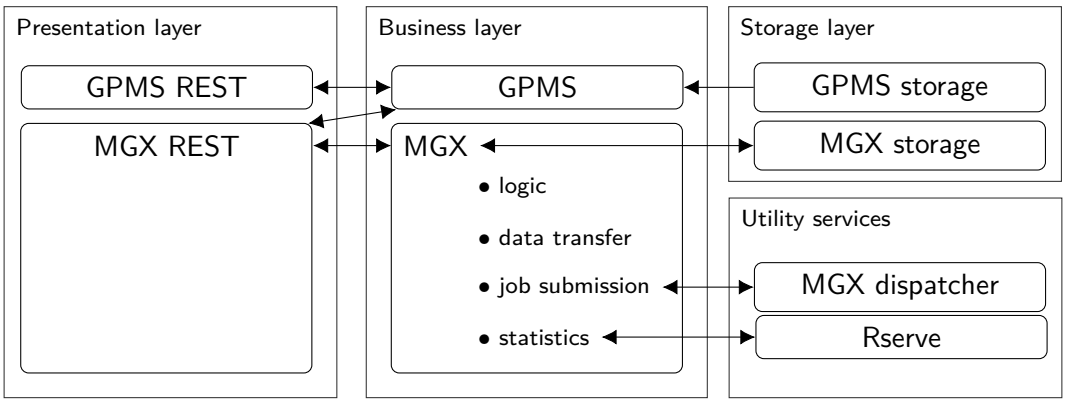


Figure 3.2: MGX server components. The implementation of the MGX server components follows a three-tier architectural model; all relevant interactions with the system are implemented within the business layer, which operates on stored data or triggers external processing steps. Access to the system is provided via the presentation layer, which is offered as a REST-based interface.

Services offered by an MGX server are exposed via REST (Table 3.1), a typically HTTP-based interface offering access to predefined operations. REST itself is a stateless protocol, *i.e.* states and state transitions reside only in the client, and being HTTP-based, allows to make use of existing proxy and caching infrastructure.

Table 3.1: REST interface examples. REST calls allow easy access to data stored within an MGX project. For modifying operations (create/update), the new desired entity state is transmitted as the request payload.

HTTP method	Request URI	Description
PUT	/Job/create	create new analysis job
GET	/Habitat/fetch/1	retrieve habitat with ID 1
POST	/Sample/update	update sample data
DELETE	/SeqRun/delete/5	delete sequencing run with ID 5

The execution of analysis workflows (Subsection 3.4.2) is not handled by the MGX server itself; instead, job submission, validation, execution as well as pre- and postprocessing steps are transparently forwarded to an associated job dispatcher; thus, further compartmentalization is achieved, which enhances system security and resilience as the server does not require direct access to compute resources, neither is it essential to expose straight access to the compute cluster.

Finally, the MGX server also includes the necessary infrastructure for a variety of statistical calculations; the implementation makes use of the popular R statistical environment (R Core Team, 2014) via an Rserve (Urbanek, 2003) instance, providing *e.g.* clustering and dimensionality reduction computations (PCA, NMDS). For security reasons, the Rserve instance may not be used directly and access is managed by a separate Java EE module.

3.4.1 Data model and sequence storage

For each MGX project, various types of data need to be stored, which all differ in their inherent structure as well as desired access pattern (*e.g.* sequential, random). For the storage of sequences and metadata, MGX makes use of relational as well as file-based storage systems; depending on intended use, a mapping for each data type has been developed that assigns a type to a corresponding storage backend (Table 3.2). Most of the data is stored within a relational database management system (RDBMS), which allows efficient data creation, querying and modification while maintaining relational integrity. Also, metagenome analysis results are stored within the relational database, unless a specialized file format is available that warrants deviation from this decision, such as *e.g.* the SAM/BAM file format for read mapping data. Finally, for user-provided data to be used within analysis pipelines, no structure known *a priori* can be assumed, as users may choose to design arbitrary analysis workflows at their own discretion.

Table 3.2: Storage structure. Different types of data need to be stored with each MGX project; depending on data structure, access pattern and cost of different storage systems, a mapping to file- and database-based storage backends has been developed.

data type	structured	backend	description
metadata	x	RDBMS	metadata
sequence data	x	RDBMS/file-based	nucleotide sequences
genomic	x	RDBMS	reference genome data
analysis jobs	x	RDBMS	workflows
analysis results	x	RDBMS	analysis results
read mapping	x	file-based	reference alignment
user data	–	file-based	user-supplied arbitrary data

Based on a requirements analysis taking into account the necessary database operations, the PostgreSQL[4] open source database system was selected for the storage of relational data, as it offers excellent performance and also supports several advanced features such as recursive Common Table Expressions (CTEs) that aid in efficient result retrieval. For user-supplied data, file-based storage is provided for each MGX project via a network-accessible file system such as NFS or CephFS.

3.4.1.1 Sequence storage

The output of next-generation sequencing machines is typically delivered in one of several standardized data formats, most prominently the text-based FASTA format for sequence data and the FASTQ format, which offers combined storage of sequencing reads with associated basecall accuracy information ("quality values"). However, these formats are neither space-efficient nor do they allow fast access to individual sequences; on the other hand, RDBMS-based storage is generally considered quite costly and thus rather unsuitable for the storage of large amounts of nucleotide data. Hence, only sequence names with an associated 64-bit unsigned integer identifier are persistently saved in the MGX project database (Figure B.2), while the actual sequence data is stored using proprietary file formats that provide lossless data encoding. Two different implementations were developed for use in conjunction with MGX:

CSF (Compact sequence format). The CSF format is used for storage of encoded nucleotide data; even though current sequencing technologies only emit standard nucleotide codes (A, T, G, C and N to express ambiguity), CSF uses 4-bit encoding, thus offering full support for all possible IUPAC[5] nucleotide codes, with the individual records separated by NUL bytes (`'\0'`). An additional indexing scheme mapping sequence identifiers (`uint64_t`) to file offsets (`uint64_t`) can be used for fast random access to individual sequences.

CSQF (CSF with qualities). The CSQF format is an extension of the CSF file format and in addition allows to deposit sequence quality information; nucleotide data and quality values for each sequence are stored in conjunction, and sequence qualities are represented using a variable bit length encoding. Based on the minimum and maximum quality values for any given sequence, the whole range of possible values in between can be encoded in $\lceil \log_2(max - min + 1) \rceil$ bits per value; two additional bytes are needed to store the number of bits per value and the base offset (min) that needs to be added to each encoded value.

[4] `https://www.postgresql.org/`
[5] International Union of Pure and Applied Chemistry

After initial sequence import into an MGX project, the corresponding storage file is immutable and never changed; within the project database, a discard flag is provided that allows to exclude individual sequences from all processing steps. This mechanism is provided to *e.g.* address host contamination within a dataset, and an analysis workflow can be used to conditionally set the flag for sequences that match a user-choosable reference genome; setting the flag will not only exclude the corresponding sequence from all future pipeline invocations, but also prune existing results that were generated by previous workflows.

3.4.1.2 Analysis result storage

Based on the initial requirement to support user-developed analysis workflows arises the need for a highly flexible model to represent and store different analysis result types. A static result model for the storage of taxonomic classifications and functional assignments does not fulfill this requirement in a sufficient manner, as it cannot be safely assumed that all possibly occurring results can be represented within this model. Also, the result model should not exhibit any dependencies against external data sources that are subject to change over time, *e.g.* database identifiers or taxonomy data provided by the NCBI. Instead, it is preferable to fully capture the inner structure and relationships of individual result entities among each other and store them in a self-describing way.

For this purpose, a dynamic result model has been developed (Figure 3.3) that allows a representation of analysis results based on their intrinsic properties:

Attributes represent a single piece of not otherwise specified information and may exist as either unconnected values ("basic attribute") or embedded into a tree structure ("hierarchical attribute") together with other attributes. Possible value examples are thus "*42*" or "*Escherichia*", which explicitly lack any indication of *e.g.* the measured unit or category.

Each attribute references an **AttributeType**, which not only adds a descriptive label to the attribute, but also further indicates the attributes' structure and domain (continuous or discrete values) to represent numerical or categorical data types. Example: "*GC content*" (basic, numerical) or "*NCBI genus*" (hierarchical, discrete).

Observations model analysis results as associations between a DNA sequence and a certain attribute. Thus, results are retained on individual sequence level, permitting users to search for and export sequences based on arbitrary criteria. As observations in addition provide base positions specifying the subregion that is described by an attribute, sub-sequence resolution is achieved.

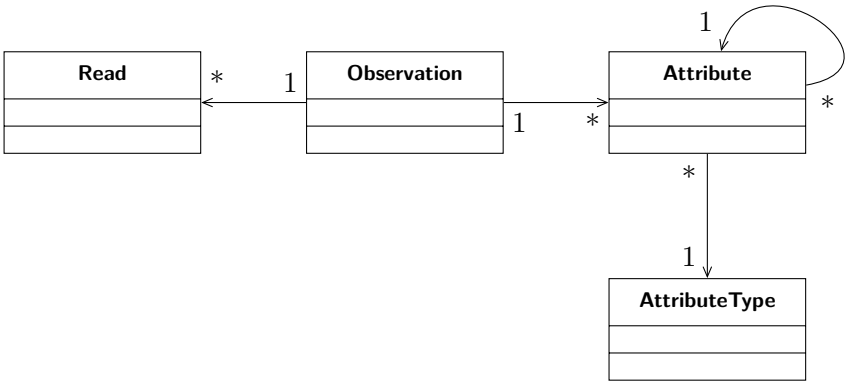


Figure 3.3: MGX result model. An Observation serves as an association between a metagenome sequence (Read) and an Attribute, while the AttributeType referenced by each Attribute further indicates the properties of the attribute such as its structure or value type. Hierarchical attributes are used to represent data such as taxonomic classifications.

Based on this result model, it is possible to not only represent current taxonomic and functional assignment results (Table 3.3), but also to denote other possible future result types. Also, the complete result structure is saved to the MGX database, thus avoiding dependencies against external resources which might be subject to change. However, this also implies that the observation table tends to become quite large, as especially taxonomic results require a separate observation to be created for each taxonomic rank.

Table 3.3: Attribute examples. The combination of Attribute and AttributeType is used to model different analysis results. Basic and hierarchical attributes are backed by attribute types, which not only add a label (*e.g.* "*Pfam domain*", "*Phylum*") to the data, but also indicate the character of the annotated values, thus allowing to determine valid operations; numerical attribute types are used for quantifiable values, while discrete attribute types denominate categorical data.

	numerical	discrete
basic	GC, length	COG, EC numbers
hierarchical	–	tax. classifications

The time required to obtain results for complete sequencing runs from this result model therefore represents a major shortcoming; even though PostgreSQL supports recursive join queries, which are an indispensable requirement for efficient retrieval

of hierarchical data, the collection of all results for a sequencing run and a requested result type still causes a non-neglectable delay, as an expensive SQL JOIN operation involving the large observation table is required in addition to subsequent grouping and aggregation. Hence, the model was extended and an additional database table stores precomputed result assignment counts for all attributes; these aggregated attribute counts are only computed once during the post-processing phase of completed analysis jobs. Hereby, joining the observation table can be avoided for interactive queries and instead, the precomputed attributecount table is used, which is significantly smaller and thus greatly reduces database response time.

Currently, one single exception exists from this general approach which is related to the mapping of metagenome sequences to reference genomes for fragment recruitments; as a performant file format already exists for the storage of read-mapping results, indexed BAM files are used to store this kind of information. Within the project database, only a small mapping entity is retained, which references the corresponding analysis job as well as the target genome and metagenome dataset that were used (Figure B.4).

3.4.1.3 Job infrastructure

Analysis jobs within MGX are implemented as workflows (Subsection 3.4.2) for the Conveyor workflow engine, and all predefined analysis pipelines are hosted in a central public repository with typically preset but customizable parameters. Each MGX server periodically updates its local copy of workflow definitions, and users may either choose to import a predefined pipeline into their project or provide their own custom method implemented as a Conveyor workflow.

Within the project database, the corresponding entities for the management and execution of analysis tasks are provided (Figure B.5), allowing access to status information as well as chosen parameters for each job.

After creation, each job initially needs to be validated; in this step, the job dispatcher checks validity of the workflow definition and verifies any user-supplied parameters (Figure 3.4). Subsequently, a job is either directly scheduled for execution or can remain queued in the internal dispatcher queue, *e.g.* during compute cluster maintenance or when insufficient resources are available.

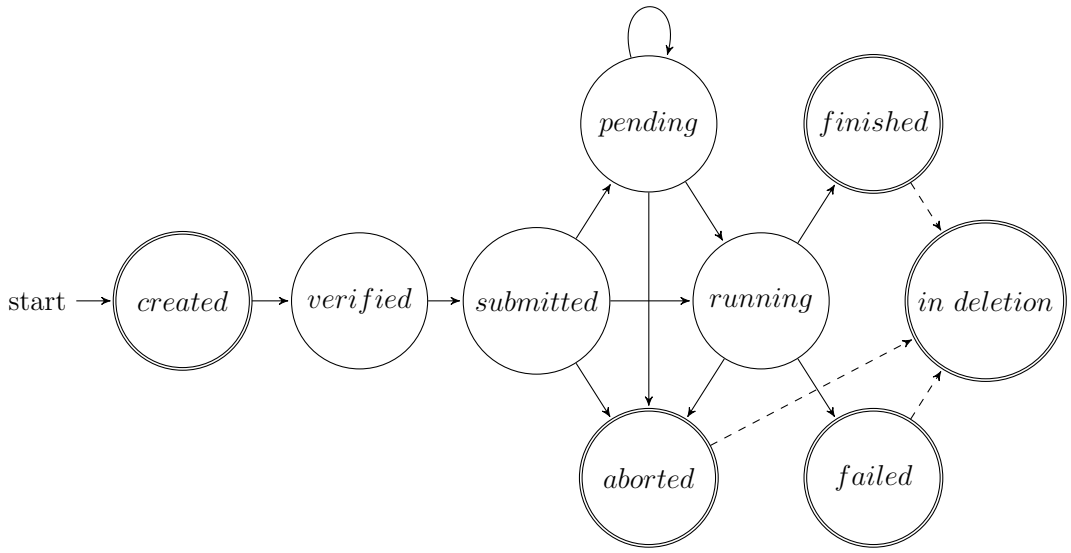


Figure 3.4: Job state transition diagram. Newly created jobs need to pass a verification step which validates parameter values before being forwarded to the MGX dispatcher. After submission, a job might be started at once or remain queued until sufficient resources become available.

3.4.2 Workflow-based analysis

Workflow systems like Galaxy (Goecks *et al.*, 2010), GenePattern (Reich *et al.*, 2006) or Conveyor (Linke *et al.*, 2011) pose an interesting alternative to custom programming: Mainly specific to a certain application domain, they provide data processing capabilities in the form of small tasks; more complex analysis workflows are then devised by connecting these tasks into a pipeline or directed graph. Programming knowledge is not required, as a graphical user interface is provided to implement an analysis pipeline. The resulting workflow definition can be published along with its results, allowing for easy reproducibility of methods as well as serving as a self-documenting description.

Furthermore, the building blocks of a workflow can easily be exchanged once an improved method becomes available: BLAST (Altschul *et al.*, 1990), for example, has been one of the most popular tools for database searches, despite its rather large computational overhead. Recently developed alternatives like GHOSTX (Suzuki *et al.*, 2014a) or DIAMOND (Buchfink *et al.*, 2014) offer considerable acceleration compared to the original BLAST algorithm while retaining similar sensitivity. Employing a workflow engine, these alternatives can effortlessly be introduced as replacements for the BLAST program, an inevitable requirement to keep pace with the continuously growing output of next-generation sequencing machines.

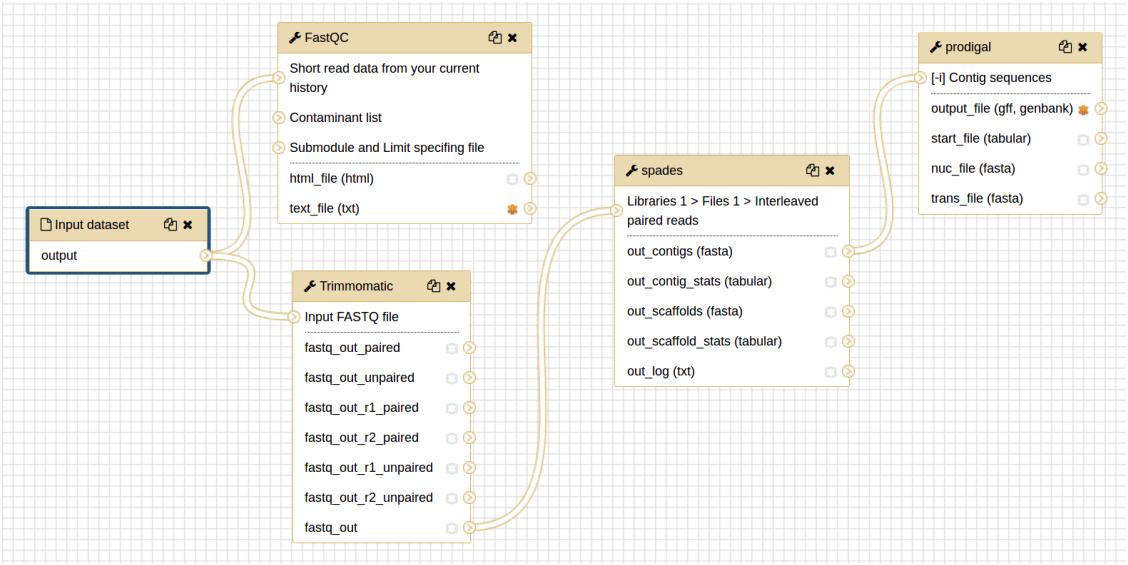


Figure 3.5: Galaxy workflow canvas. The web-based galaxy workflow editor allows to easily design analysis workflows based on predefined tools. Using just five nodes, a simple pipeline performing quality control, genome assembly and gene prediction can be devised.

However, adequate compute resources are still required, and workflow engines typically lack in both data management capabilities as well as support for appropriate visualizations, making them unsuitable as a sole means for metagenome analysis, unless they are used as an integral component of a larger software platform.

From the available workflow engines with a focus on bioinformatics, the popular Galaxy platform (Figure 3.5; Goecks *et al.*, 2010) provides an API for tool execution, but the software seems to be mostly intended for web-based usage. The Taverna (Oinn *et al.*, 2004) workflow management system is also intended for bioinformatics data processing, but was not considered for MGX due to data confidentiality issues, as it distributes data to external web services for analysis. Conveyor, a workflow engine developed at Bielefeld University (and thus with easily obtainable support), provides an advanced type system and integrated support for local job execution as well as distribution to DRMAA[6]-based compute clusters. Conveyor is based on the .NET platform, which provides access to a wide range of supported programming languages, and is also easily extensible, as new functionalities can be integrated in the form of plugins. Conveyor plugins are already available for a large collection of bioinformatics tools from the genomics field, but the system lacks support for metagenome data analysis. Nonetheless, it was decided to use the Conveyor workflow engine for MGX and contribute the missing plugins for commonly used metagenome processing tools such as Kraken, Centrifuge or GHOSTX to the Conveyor codebase (Table 3.4).

[6]Distributed Resource Management Application API

Table 3.4: Contributed Conveyor plugins. A large variety of tools for microbial community analysis were implemented as plugins and contributed to the Conveyor workflow engine.

Task	Plugin	Description
Tax. classification	Conveyor.MetaCV	MetaCV
	Conveyor.MetaBin	MetaBin
	Conveyor.Kraken	Kraken, Kraken 2
	Conveyor.MetaPhyler	MetaPhyler
	Conveyor.MetaPhlAn	MetaPhlAn
	Conveyor.MetaPhlAn2	MetaPhlAn 2
	Conveyor.Kaiju	Kaiju
	Conveyor.Centrifuge	Centrifuge
Gene function	Conveyor.UProC	UProC
Reference mapping	Conveyor.ReadMapping.FR-HIT	FR-HIT
	Conveyor.ReadMapping.Bowtie	Bowtie 2
	Conveyor.ReadMapping.BlastToSAM	MagicBLAST
	Conveyor.ReadMapping.FR-HIT	FR-HIT
	Conveyor.Minimap2	Minimap2
Homology search	Conveyor.Blast.RAPSearch2	RAPSearch 2
	Conveyor.Blast.GHOSTX	GHOSTX, GHOSTZ
	Conveyor.Blast.DIAMOND	DIAMOND
Databases	Conveyor.Database.FunGene	FunGene
	Conveyor.Blast.ClusterMine360	ClusterMine 360
Gene prediction	Conveyor.FragGeneScan	FragGeneScan+
	Conveyor.MetaGeneMark	MetaGeneMark
Metataxonomics	Conveyor.Qiime[1]	QIIME
	Conveyor.Mothur[1]	Mothur

[1]The QIIME and Mothur plugins were implemented by Patrick Blumenkamp during his employment as a student programmer under my supervision.

Workflows for Conveyor are implemented using a graphical user interface, the Conveyor Designer application; capabilities for reading, processing, and writing of bioinformatics data are implemented in the form of nodes, which denote individual processing steps. As Conveyor installations might differ in their set of available plugins, initially a "plugin dump" needs to be imported into the Designer, which

Listing 3.1: Conveyor node parameterization example. The RDPClassifier node has one fixed (chunkSize) as well as two user-adaptable parameters (trainset, threshold); all configuration items provide preset default parameters.

```
<node id="1" type="Conveyor.RDPClassifier.RDPClassifier`1">
  <configuration_items>
    <configuration_item name="chunkSize" value="500"/>
    <configuration_item name="trainset" user_name="training set"
       user_description="training set type" value="16srrna"/>
    <configuration_item name="threshold" user_name="confidence
       cutoff" user_description="Classification confidence cutoff"
        value="0.7"/>
  </configuration_items>
  <typeParameters>
    <type name="Conveyor.BioinformaticsTypes.SimpleDNASequence"/>
  </typeParameters>
</node>
```

contains a list of all available nodes and data types known to this instance. To create a novel workflow, the Conveyor Designer is then used to place and configure the required node types, which are finally connected among each other into a directed graph. The completed workflow is saved in an XML-based file format, which can be imported into an MGX project. During the actual execution, data passes along the connected nodes and is transformed in transit. A step-by-step guide for the implementation of Conveyor workflows for use within MGX is given in Appendix A.

Conveyor distinguishes between fixed and user-adaptable node configuration items, and those that should be exposed for possible manual refinement can be annotated with a short name (user_name) and a corresponding description (user_description) providing additional information, *e.g.* a short explanation (Listing 3.1). As graph definitions are stored based on XML, it is easy to parse and extract the relevant configuration fields within MGX and offer them to a user for review and possible modification.

3.4.2.1 Conveyor.MGX

The Conveyor workflow engine provides a wide range of nodes for bioinformatics data processing, among them a large variety of input nodes that are able to parse and extract data from common bioinformatics file formats such as FASTA, FASTQ, GenBank or EMBL. However, all these nodes operate on files and are neither able to access the proprietary sequence container format used by MGX nor does Conveyor

support reading data from or writing to relational databases. Therefore, a dedicated plugin has been implemented that serves as an interface between Conveyor and MGX, allowing sequence retrieval and storage of analysis results in MGX projects; the plugin also provides the appropriate means to access files uploaded by the user, thus making it possible to incorporate custom data, for example sequence databases maintained within a research group, into an analysis pipeline.

- The GetMGXJob node is the main entry point for workflow execution within MGX (Figure 3.6). By convention, exactly one instance of this node is required for each pipeline definition, and the MGX framework provides all required node configuration properties at runtime via an external configuration file. The node emits a single MGXJob instance, which provides contextual information such as access to the MGX project database, sequence data and files uploaded by a user.

- The GetSequences node provides access to the individual sequences of the sequencing run for a job. It requires an MGXJob instance for operation and emits MGXSequence instances (Figure 3.6); the node also verifies the discard flag for each metagenome sequence, omitting instances where the flag has been set. The GetQSequences variant has the same characteristics, but outputs sequences with associated quality information (MGXQSequence).

- The CreateAttributeType and CreateAttribute nodes are responsible for the creation of the appropriate data types within Conveyor (Figures 3.7, 3.8); these data types, however, are not directly persisted, as subsequent filtering steps might still occur. For hierarchical attributes, the CreateHierarchicalAttribute node is available (Figure 3.9).

- The AnnotateAttribute node typically represents the final step of an MGX workflow. The node persists attributes and attribute types to the database and creates the corresponding observations for a subregion of the input sequence. A variant, the AnnotateSequence node, creates observations spanning the complete length of the MGXSequence argument and thus does not require input links for the start and stop coordinates.

- Additional utility nodes provide access to reference genomes stored within MGX (GetMGXReference), implement preprocessing capabilities for certain user-provided data types (*e.g.* formatting of HMM databases; MGXAAHMM-Database, MGXDNAHMMDatabase), handle the discard flag or create mapping entities within the project database (CreateMapping).

Figure 3.6: Application context and sequence access. The GetMGXJob and GetSequences nodes represent the initial steps required to inject sequences stored within MGX projects into the Conveyor workflow system as MGXSequence instances for processing and annotation.

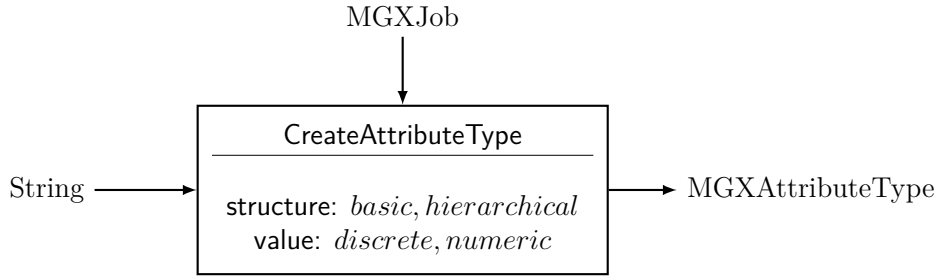


Figure 3.7: The CreateAttributeType node uses textual data to create attribute types; additional node configuration options are provided in order to define attribute structure as well as value type. Each attribute type also references the currently active analysis job, which provides access to the corresponding MGX project database as well as contextual information.

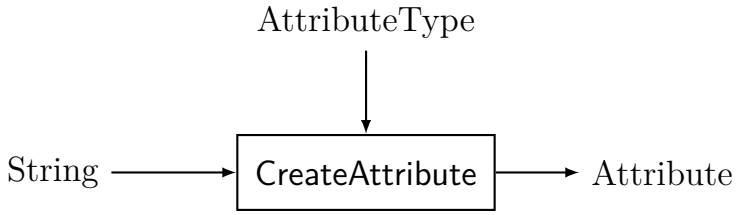


Figure 3.8: The CreateAttribute node has two input connectors and one output endpoint. An AttributeType is required for each Attribute, and its value is derived from an input string.

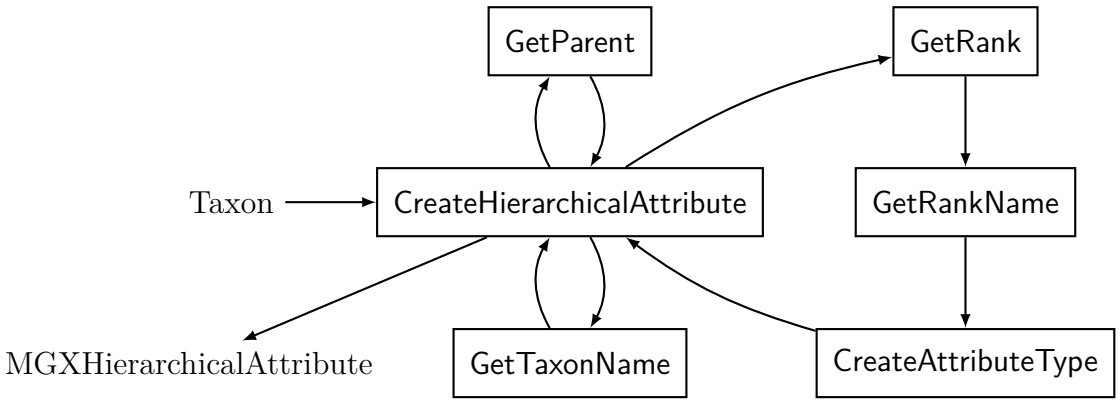


Figure 3.9: The CreateHierarchicalAttribute node: The node requires three external loops to convert a taxon into a hierarchical attribute: In clockwise order, the first loop maps a taxon to its parent, the second loop generates the attribute type for each element, and the third loop creates the respective attribute value, *i.e.* the scientific taxon name. The generated AttributeType is emitted from the node and passed on *e.g.* to an AnnotateAttribute instance.

3.4.3 Data serialization

For data transfers between client and server, entities from the MGX data model need to be serialized in an efficient manner; traditionally, RPC-, SOAP- and REST- based services often consume and produce structured text-based encoding formats such as XML[7] or JSON[8] for object marshalling; binary encodings yield more compact messages, thus reducing message size while at the same time improving decoding performance. Modern serialization formats such as Apache Thrift, Avro, or Google Protocol Buffers offer space-efficient encoding as well as high-performance serialization and deserialization. Typically, a formal Interface Description Language (IDL) is used to define a schema containing the required data types (Listing 3.2), and an IDL compiler creates the corresponding code and data transfer objects (DTOs).

For MGX, Google's Protocol Buffers format was chosen for performance reasons as well as small size of encoded entities. The protoc compiler supports code generation for a wide range of programming languages, among them C++, C#, Go, Java, and Python, and also allows to define RPC-based services. The communication between client and the MGX server is in addition mandatorily encrypted using the standardized SSL (Secure Sockets Layer) protocol, ensuring confidentiality of unpublished data and protecting the integrity of login credentials.

[7] Extensible Markup Language
[8] JavaScript Object Notation

Listing 3.2: Protocol Buffers IDL definition. A text-based schema is used to define a structured message entity.

```
message HabitatDTO {
  optional uint64 id = 1;
  required string name = 2;
  required double gps_latitude = 3;
  required double gps_longitude = 4;
  required int32 altitude = 5;
  required string biome = 6;
  optional string description = 7;
}
```

3.4.4 Security and access control

For user and project management, the GPMS (General Project Management System) system developed at the CeBiTec (Center for Biotechnology, Bielefeld University) is employed. GPMS serves as an established project management infrastructure, managing single sign-on (SSO) for all applications offered by the CeBiTec; the software has also been deployed at the Bioinformatics and Systems Biology group at Justus-Liebig-University Gießen and is currently in use for GenDB (Meyer *et al.*, 2003) as well as EDGAR (Blom *et al.*, 2016). As GPMS has so far only been provided for Perl-based applications, a Java implementation with support for MySQL- and LDAP-based data storage backends was implemented.

GPMS provides RBAC (Role-based access control) capabilities to applications that employ it, and each application is free to define own rights and roles to constrict access to certain operations. Within MGX, three different user roles have been defined:

1. **Guest:** The guest role represents the lowest access level, providing read-only access to MGX projects, and while existing data and analysis results may be retrieved, no new entities are allowed to be created. Also, the role does not permit deletion of any data.

2. **User:** The user role represents the most frequently assigned role within MGX, allowing full access to data and metadata as well as the execution of analysis workflows.

3. **Admin:** The admin role inherits all rights already provided via the user role; in addition, admins possess project management privileges and can grant or revoke membership rights to/from additional users.

Listing 3.3: Programmatic MGX access. The MGX client library enables simple programmatic access to data stored on remote MGX servers. After establishing a server connection, the defined habitats are obtained, and for each habitat, possibly existing samples are retrieved, displayed and purged from the project.

```
MGXDTOMaster master = getMaster("juser", "SeCrEt", "MGX_Demo");

Iterator<HabitatDTO> hit = master.Habitat().fetchall();
while (hit != null && hit.hasNext()) {
    HabitatDTO h = hit.next();
    System.out.println(h.getName());
    Iterator<SampleDTO> sit = master.Sample().byHabitat(h.getId());
    while (sit != null && sit.hasNext()) {
        SampleDTO sample = sit.next();
        System.out.println(sample.getMaterial());
        master.Sample().delete(sample.getId());
    }
}
```

3.5 MGX client library

All the required functionality to remotely access the contents of an MGX project has been wrapped in a Java client library that provides programmatic access to GPMS-based projects in general and implements the required REST calls specific to MGX. Using this client library, existing metadata, data, and analysis results can be retrieved from MGX projects (Listing 3.3); also, new entities and analysis jobs can be created, thus allowing to automate routine data processing with MGX. Finally, the MGX client library provides access to certain statistical functions that are implemented via the R statistical environment (R Core Team, 2014) and offered by each MGX server.

3.6 Graphical user interface

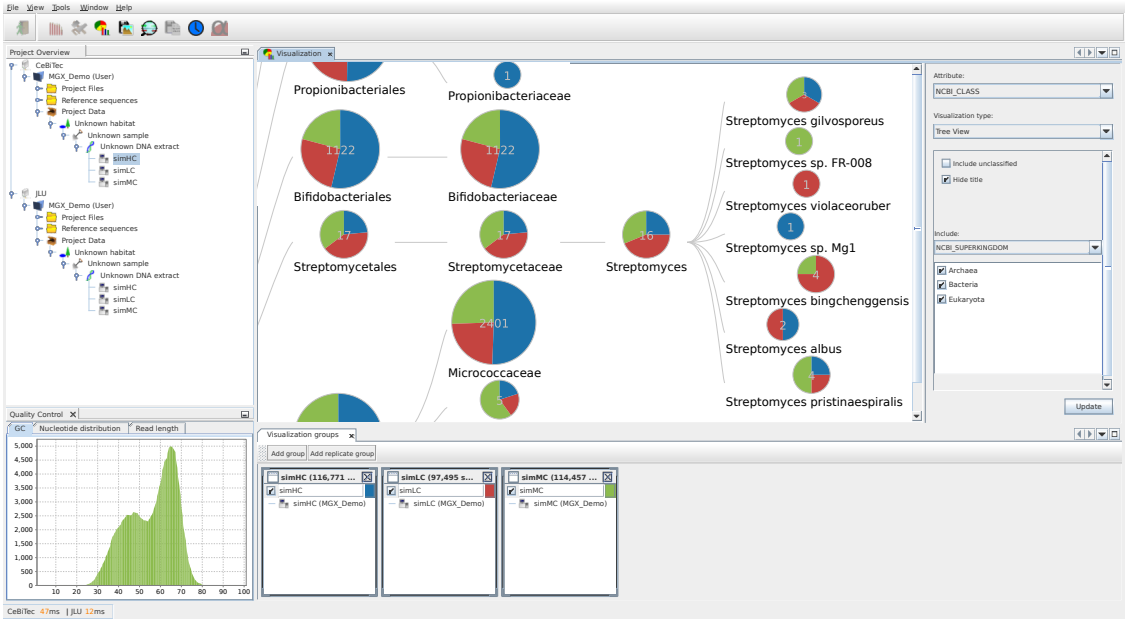


Figure 3.10: The MGX application client. Shown are the project explorer window (top left), quality control reports for the currently selected sequencing run (bottom left), and a hierarchical tree chart (center) displaying three groups, which are defined at the bottom. Toolbar buttons provides convenient access to the various components of the application.

The MGX graphical user interface (Figure 3.10) has been built as an interactive Java application available for all major operating systems such as Microsoft® Windows™, Apple macOS®, or Linux. The user interface has been implemented using the NetBeans Platform, a modular framework for Rich Client application development. The graphical user interface (GUI) assists the user in all common tasks, featuring convenient wizard-driven acquisition and validation of metadata, sequence data import and analysis workflow execution. After the initial sequence data upload, researchers can already inspect quality control reports for their metagenome datasets and are able to select one or several analysis pipelines, review and adapt existing parameters, and finally schedule the analysis jobs for execution. Upon completion, results can be retrieved and evaluated.

For this, the GUI features a rich set of different visualization modules, enabling researchers to interactively explore analysis results for their metagenome datasets, generate high-quality charts, or export results to *e.g.* Microsoft® Excel. Based on the abstract modeling of results, the client application is able to automatically select suitable visualization types and offer postprocessing operations such as normaliza-

tion or filtering. For all visualizations, MGX allows users to freely define groups, thus allowing to combine and compare datasets across project or server boundaries without the need to re-execute classification pipelines. Various statistical methods are provided to investigate community complexity and coverage, as well as to identify determining factors in comparison between several metagenomes: rarefaction analysis allows researchers to estimate whether the amount of sequence data suffices to draw valid conclusions, biodiversity indices provide intrinsic measurements of community complexity, and several methods such as PCA, PCoA, M/A plots, or clustering can be utilized to interpret data in a comparative approach. Reference mappings, the alignment of metagenome sequences to reference genomes of known origin, are another noteworthy feature and allow the creation of fragment recruitment plots based on public as well as user-supplied reference genomes.

The GUI is able to connect to several servers in parallel, facilitating easy scalability as more server instances are deployed, and researchers may even choose to operate their own MGX server with dedicated compute resources.

3.6.1 Project Structure

From within the user interface, the different sections for data storage (Table 3.2) provided by each MGX project are presented as three different parts (Figure 3.11): User-provided files for intended use within analysis pipelines can be uploaded into the "Project Files" section; annotated reference genomes are stored within the "Reference Sequences" section and may serve as targets for the mapping of metagenome sequences or for the creation of fragment recruitment plots. These reference genomes can either be uploaded by the user (in EMBL, GenBank or FASTA format), or they can be imported from a dedicated repository of published archaeal and bacterial genomes, which is hosted by each MGX server. Finally, the "Project Data" section is used to store project metadata as well as actual metagenome sequence data.

Figure 3.11: Project structure. Each MGX project is divided into three different parts; from top to bottom, file storage is offered for arbitrary data that should be used within analysis pipelines. Reference sequences (including annotation data, if available) and project data containing metadata as well as sequence datasets are also stored in different sections.

3.6.2 Data Import

In order to process metagenome sequence data with MGX, the first required step is the definition of the corresponding metadata entities before importing any sequences. Metadata acquisition is supported by user-friendly wizards, which aid the user and validate the entered metadata parameters, where possible. Initially, the properties of the targeted habitat need to be provided, including its location and the biome type (Figure 3.12). Afterwards, other wizards allow to record the sampling and DNA extraction procedures in a similar manner (Figure 3.13). Once all required metadata items have been recorded, sequence data import is started; the upload wizard supports all commonly used sequence formats (SFF[9] for 454/IonTorrent data, FASTA, FASTQ, as well as their gzip-compressed variants) and sequence data is transmitted in chunks to the corresponding server.

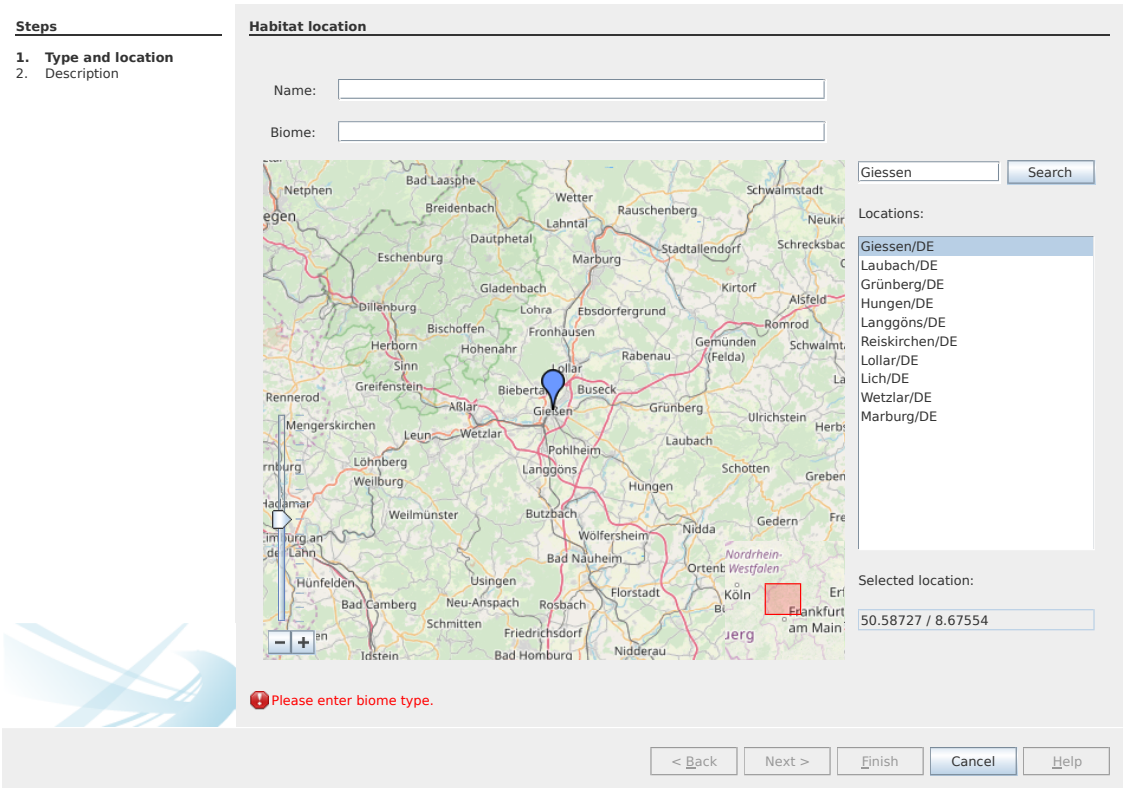


Figure 3.12: Habitat metadata acquisition. Comfortable wizards within MGX assist the user in metadata entry and validation; the habitat wizard collects the geographical location of a habitat as well as the biome type.

[9]Standard Flowgram Format

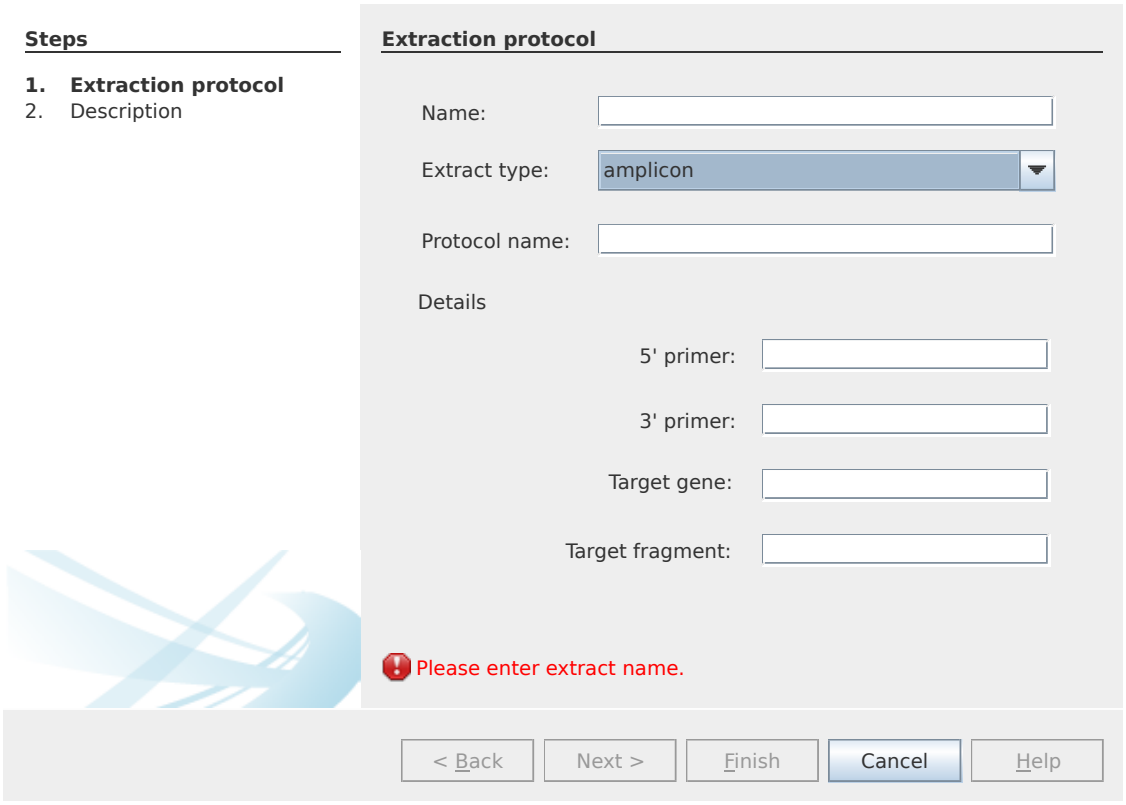
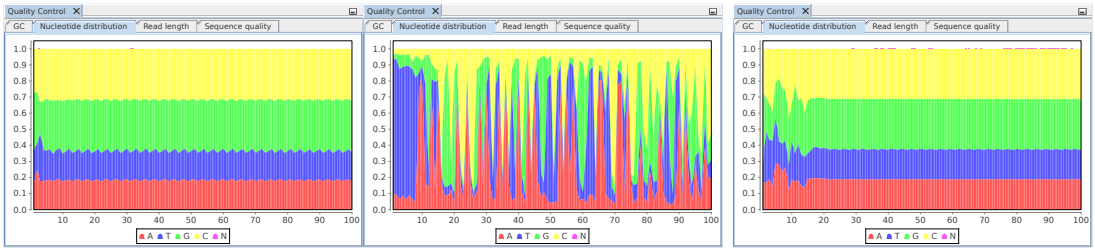


Figure 3.13: DNA extraction procedures. The DNA extract wizard captures metadata describing lab procedures undertaken to extract DNA from a sample. Depending on the approach (metataxonomics study, metagenome, or metatranscriptome), target genes, primer sequences and additional treatments such as ribosomal depletion of metatranscriptomes are also recorded.

3.6.3 Quality Control

During initial sequence data import, a set of different quality control reports is simultaneously generated. Given the sheer magnitude of different sequencing strategies, it is difficult to provide a preprocessing step that satisfactorily addresses all possible combinations that might occur. Thus, MGX at present creates quality control reports for uploaded datasets and while host sequences or contamination can be pruned from an existing dataset at a later point in time, no additional processing steps are performed. Demultiplexing, merging of overlapping read pairs, adapter removal and dereplication are tasks so far left to the users, as these are expected to be most familiar with their own datasets and have access to the correct barcode and adapter sequences, which are typically provided by the sequencing facility upon request.

Currently, MGX creates GC content and sequence length distribution charts as well as a position-specific residue frequency plot of the first 100 bp (Figure 3.14). For datasets with corresponding quality score information, an additional statistic with mean and average quality score as well as standard deviation per base position is generated, as well (Figure 3.15). Based on these reports, MGX users are able to assess the quality of their datasets and also identify possible artifacts such as incomplete adapter removal before executing any actual analysis jobs.



(a) High-GC data **(b)** Amplicon data **(c)** Adapter residue

Figure 3.14: Nucleotide distribution examples. The leftmost panel displays an almost even nucleotide distribution over the first 100 base pairs of a metagenome dataset with high GC content. For metataxonomics data (mid), conserved and less conserved positions are clearly visible from the plot. The plot to the right shows the position-specific nucleotide frequency for another metagenome dataset, and the distinct variation pattern at the 5' end indicates incomplete removal of adapter/barcode sequences.

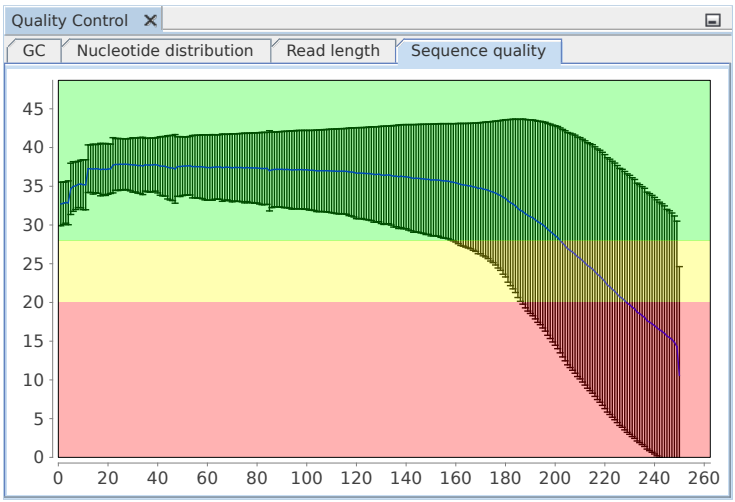


Figure 3.15: Quality control. For sequence data imported with associated basecall quality information, an additional statistic detailing the quality score distribution is provided. The corresponding chart displays position-specific mean and standard deviation values for the encountered quality scores.

3.6.4 Job Execution

Execution of analysis workflows is also initiated via a separate wizard; as MGX already offers a wide range of workflows addressing different topics, users are able to i.) select a new workflow from the repository provided by each server (Figure 3.16), ii.) choose a pipeline that has previously been used within a project or iii.) upload an own workflow definition compatible with the Conveyor workflow engine.

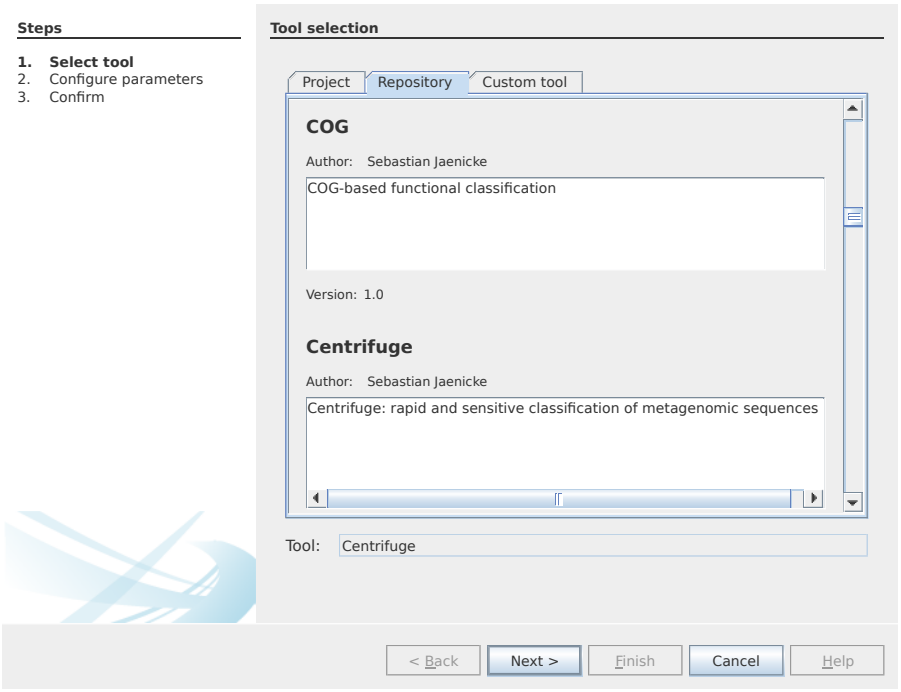


Figure 3.16: Analysis selection. The MGX repository allows users to choose from a wide range of different analysis workflows.

The XML definition file for the chosen workflow is automatically processed, and subsequent configuration steps of the wizard allow to adapt and review parameters that have been marked for user customization (Figure 3.17). Depending on the parameter type, the wizard offers appropriate alternatives (*e.g.* reference genomes present in an MGX project) and validates conformity of user-provided values with the corresponding data types. Once completed, the wizard deposits a corresponding job entity into the project database and submits the job for execution.

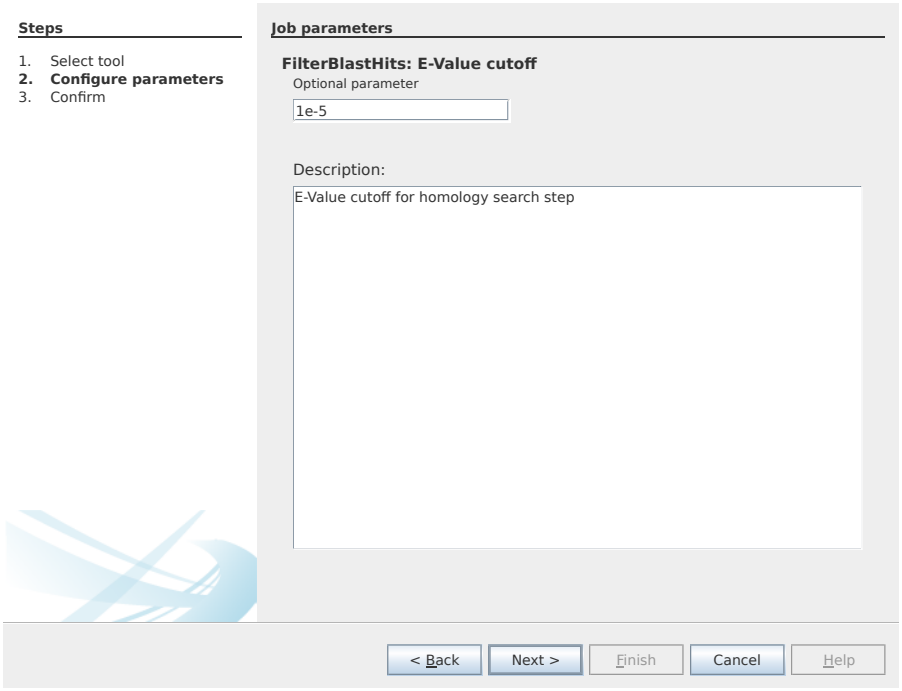


Figure 3.17: Job parameterization. Before a job is scheduled for execution, the wizard allows to review and adapt different configuration parameters.

Job progress and status can be examined using a different GUI component, the Job Monitor (Figure 3.18). This component provides a tabular view of all analysis tasks that have been created for one or several selected sequencing runs, together with their parameters as well as execution status. For completed jobs, the runtime is also indicated in a tooltip, and in case an error occurred, the component allows to retrieve and inspect the corresponding log file and re-schedule the task.

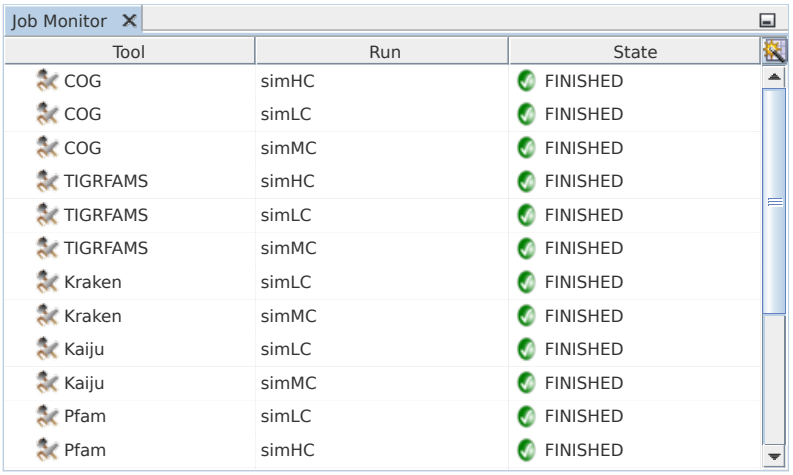


Figure 3.18: Job surveillance. The Job Monitor component provides status information about the analysis tasks for one or several sequencing runs. For each job, a tooltip provides detailed information, including configured job parameters and runtime.

3.6.5 Visualization and Reporting

Once the computation of an analysis pipeline has finished, results are ready to be retrieved and inspected. MGX supports the handling of analysis results even if the corresponding sequencing datasets are located on different MGX servers. Result visualization within MGX is not based on individual sequencing runs; instead, runs need to be arranged in groups. All groups are managed by one central instance, the VGroupManager (Figure 3.19), which controls the creation of new groups as well as the addition and removal of datasets. As soon as a user of the application selects a certain result type, or sequencing runs are added to or removed from a group, the VGroupManager initiates retrieval of analysis results from the corresponding MGX server, which is performed in parallel for fast response times; also, the VGroupManager maintains an internal cache of previously obtained results. For the graphical representation of analysis results, MGX provides a dedicated reporting and visualization component which actually consists of two separate windows (Figure 3.20).

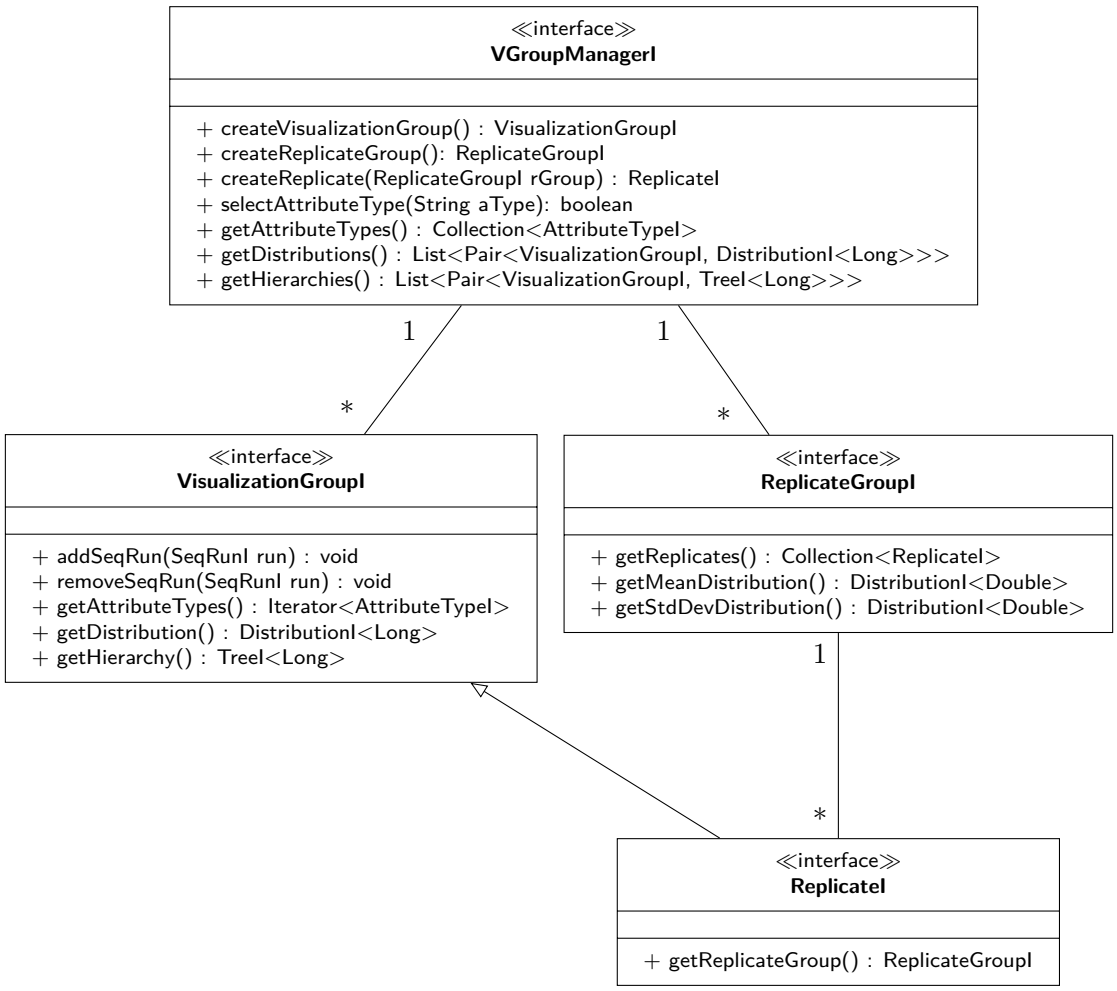


Figure 3.19: Data group management components. A VGroupManagerI implementation represents the central resource for the management of data groups. Sequencing runs can be assigned to either visualization groups, which aggregate results, or to replicates within replicate groups, which allow further statistical evaluation. The class diagram depicts a shortened selection of all available interface methods for clarity.

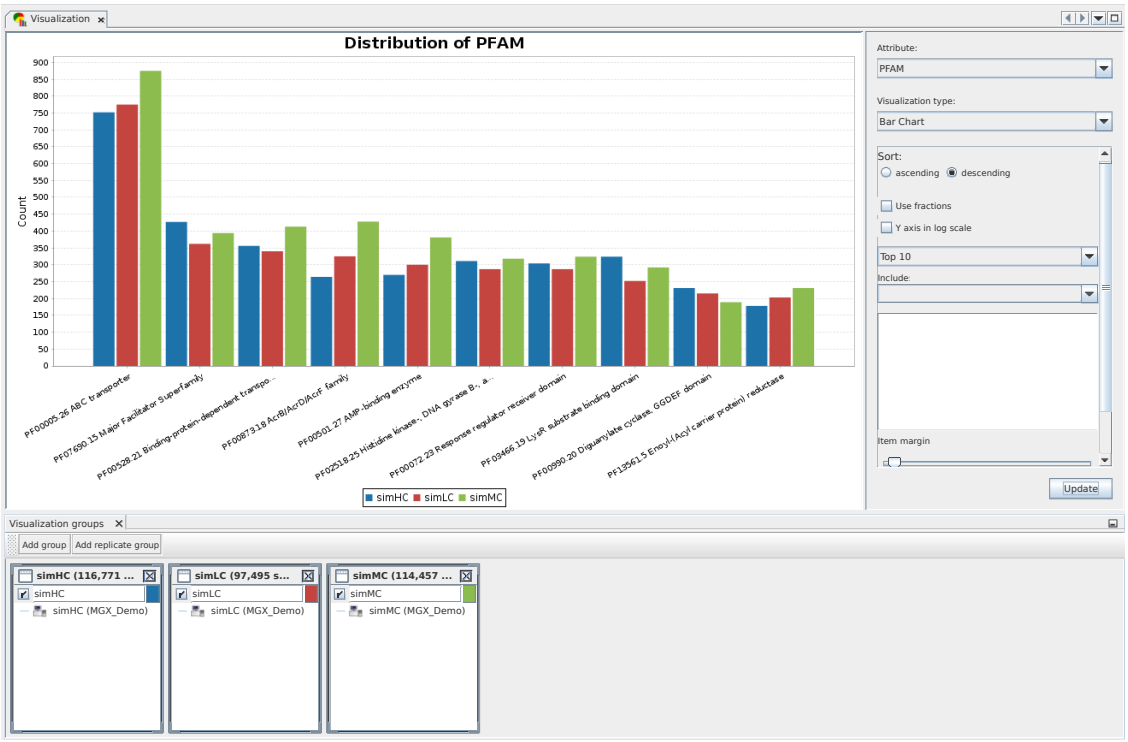


Figure 3.20: Visualization module. GUI components of the visualization module: The top window comprises the main display area (center) currently showing a bar chart and allows to select result and visualization type (top right); in addition, chart-specific customization options are available on the right hand side. On the bottom, the Group Window is used to create and define data groups.

3.6.5.1 Data groups

Data groups are provided as a container for one or several sequencing runs, for example multiple metagenomes obtained from the same sample, or the distinct output files from a paired-end sequencing approach. Each group may be assigned a custom name, its display color can be chosen by the user, and it may be temporarily disabled. A sequencing run can be a member of several different groups, but it is not possible to add it to a group more than once. Currently, two distinct group types are supported:

Basic **visualization groups** are purely additive, *i.e.* they aggregate results across all datasets contained therein. One or several sequencing runs can be placed into a visualization group and be displayed together, *e.g.* paired-end datasets or multiple sequencing runs generated using the same library.

Replicate groups are used to handle biological and technical replicates, and cannot directly contain any sequencing data; instead, individual replicates need to be defined within a replicate group and runs are added to these replicates. Thus, replicates are additive and exhibit the same properties as visualization groups, but as they are contained in replicate groups constituting an additional hierarchy level, they allow to access certain additional features such as statistical barcharts or volcano plots.

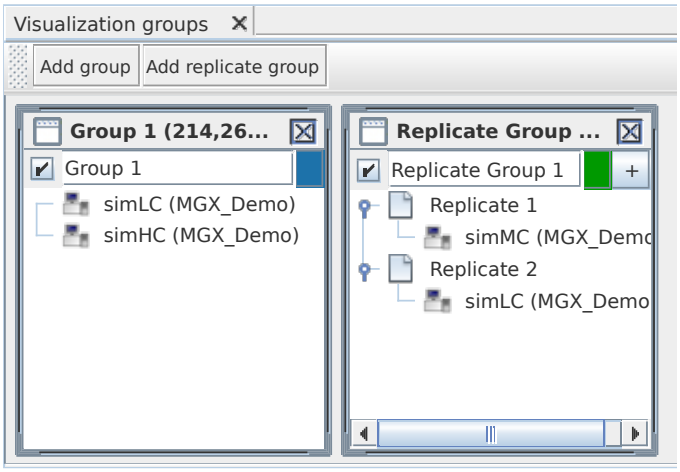


Figure 3.21: Group definition. MGX supports different group types for result reporting; here, a basic group with two sequencing runs is shown (left) together with a replicate group containing two replicates (right). For both group types, a name as well as a chart color can be assigned by the user; also, groups can be temporarily disabled to exclude them from a visualization.

The Group Window is the corresponding UI component to create and modify the different group types (Figure 3.21). Sequencing runs are added to the individual groups using "Drag and Drop"; they do not have to reside within the same MGX project, but can originate from different projects or even different servers. In case a certain result is provided by more than one analysis job, *e.g.* different taxonomic classification approaches, the user is presented with a dialog inquiring to choose between the possible alternatives. Hence, results can not only be compared for different datasets, but also for one metagenome analyzed with different methodological approaches.

3.6.5.2 Visualizations

Different visualizations are provided in the form of plugins which can be dynamically added to the GUI application, and implementations are automatically detected

by the framework. For this, all visualizations implement the ViewerI interface (Figure 3.22); a ViewerI implementation is not necessarily a graphical depiction of the data itself, but rather an abstract concept and an instance can also provide a tabular view of the results or implement some kind of statistical method that postprocesses the data and creates a corresponding statistical chart.

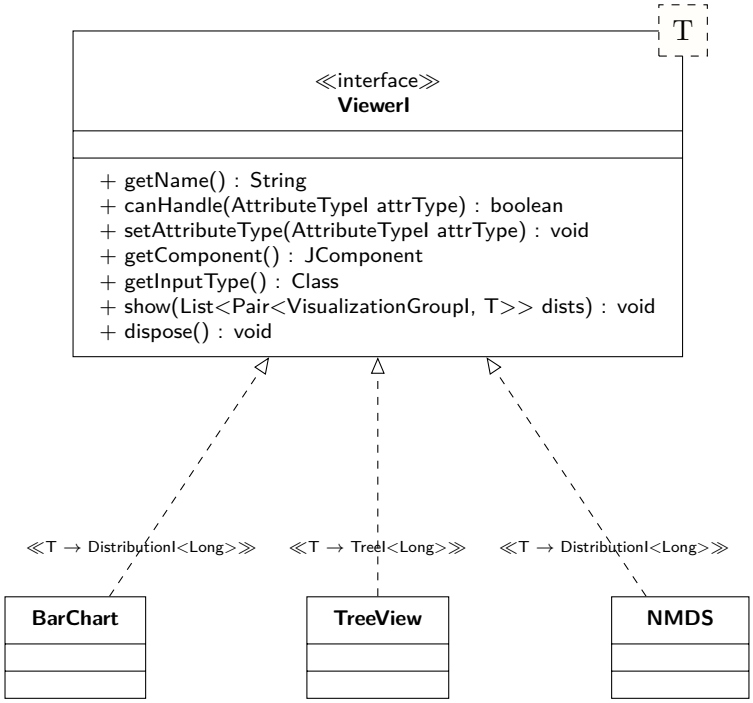


Figure 3.22: The ViewerI interface. Visualizations as well as statistical methods within MGX are provided as implementations of the ViewerI interface.

ViewerI implementations can indicate their ability to handle a certain result based on the intrinsic properties of the corresponding AttributeTypes as well as additional criteria (Table 3.5). Bar charts (Figure 3.23), for example, are suitable for categorical data and can process basic as well as hierarchical data, while other visualizations might be restricted to hierarchical data only (Figure 3.24) or depend on a fixed number of defined groups (*e.g.* visualizations that support comparative evaluation of exactly two groups). In rare cases, viewers are available only for certain AttributeType values – the KEGG[10] pathway viewer (Figure 3.25), for example, is available only for annotated EC[11] numbers.

[10]Kyoto Encyclopedia of Genes and Genomes
[11]Enzyme Commission

Table 3.5: Example ViewerI restrictions. Data viewers can indicate their ability to display certain data types based on the characteristics of the corresponding AttributeType as well as additional criteria. B: basic; H: hierarchical; N: numeric; D: discrete.

Chart type	Restriction
XY Plot	$(B \vee H) \wedge N$
Bar Chart	$(B \vee H) \wedge D$
Tree Viewer	$H \wedge D$
M/A Plot	$(B \vee H) \wedge D \wedge numGroups == 2$
KEGG pathway	$B \wedge D \wedge AttributeType == \text{“}EC\ number''$

The main Visualization Window is provided for chart selection, customization and display; it obtains a list of available analysis result types from the VGroupManager and offers them to the user. Based on the desired result type, the display component inquires all available visualizations and downstream analysis components whether they support handling a certain result, narrowing down the selection that is offered to the user. Also, valid postprocessing operations such as sorting or filtering can be automatically determined in this manner.

Depending on the selected combination of attribute type and viewer, the individual groups provide analysis results as instances implementing either the DistributionI or TreeI interface for “flat” as well as hierarchical results. The generated charts can be exported in a variety of standard image formats such as PNG, JPEG or SVG; tabular data is saved to TSV/CSV format, while other visualizations in addition offer more specific export capabilities, *e.g.* the result of a hierarchical clustering which can be stored in Newick tree format.

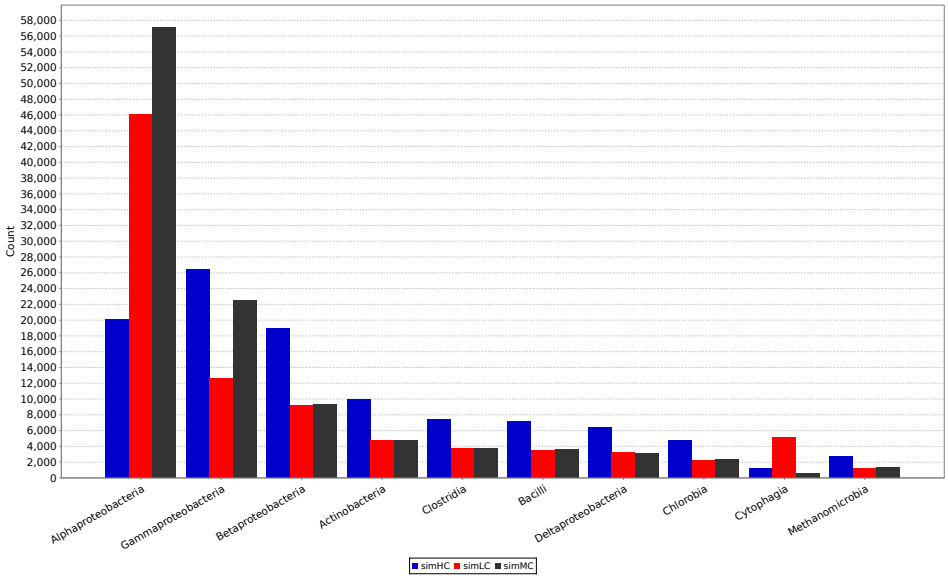


Figure 3.23: Result visualization. The bar chart is the most simple diagram type for categorical data; the respective customization component (not shown) provides various means allowing to normalize, transform and filter the underlying data.

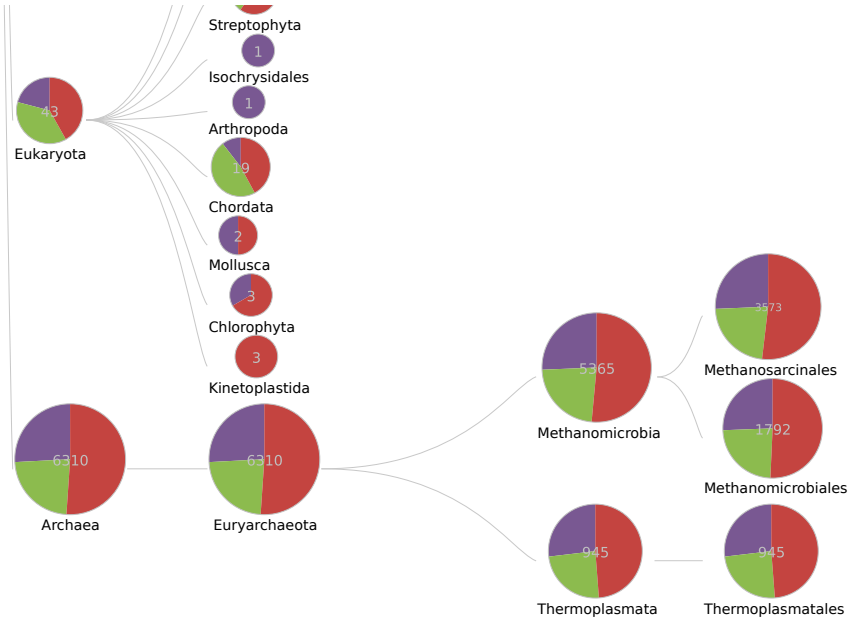


Figure 3.24: Tree visualization. For hierarchical data, the tree viewer allows interactive navigation and node expansion; therefore, even large hierarchies can comfortably be explored. The viewer supports one or several data groups, and the number of assigned sequences is reflected by the size of each node.

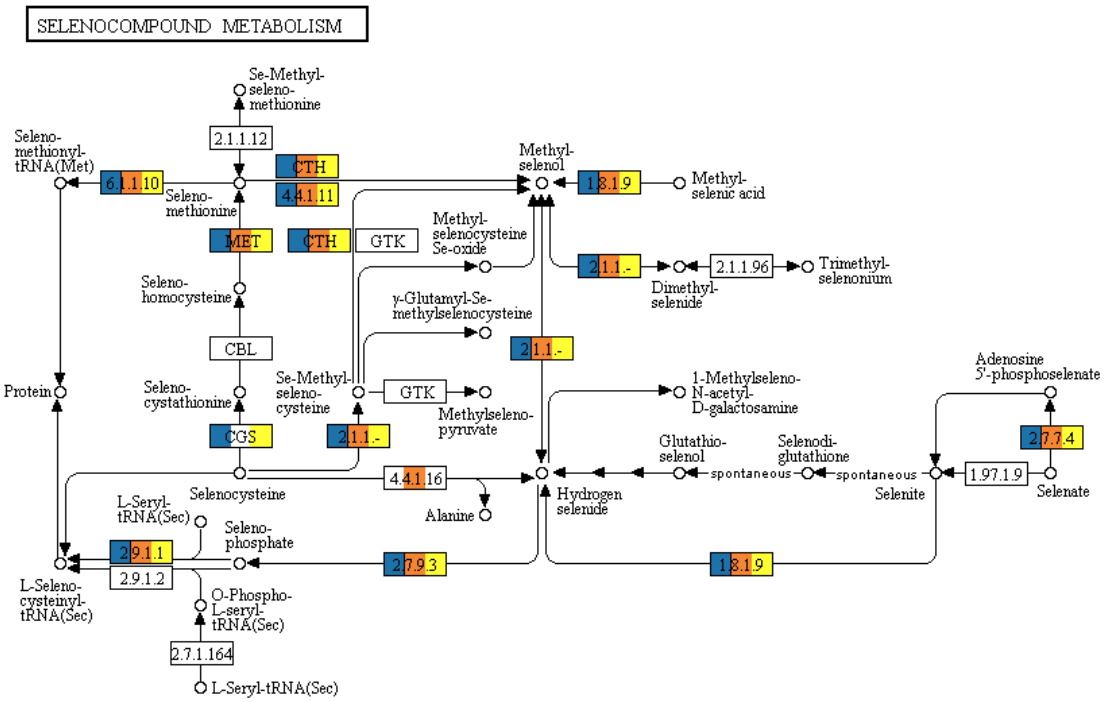


Figure 3.25: Pathway mapping. For certain result types, highly specific visualizations are implemented within MGX; EC numbers, for example, can be mapped to corresponding KEGG pathways. The colors correspond to three different visualization groups and indicate presence/absence of an annotated EC number within each dataset. A tooltip provides additional information such as the name for each EC number and the number of assigned sequences from each group.

3.6.6 Sequence Export

As analysis results are stored with subsequence resolution, MGX also allows to export metagenome subsets based on user-specified criteria. This enables users to selectively extract sequence data for additional processing, *e.g.* sequences with a common taxonomic classification might be exported in order to attempt reconstruction of the corresponding genome, or sequences with identical functional assignment (Figure 3.26) might be obtained to perform gene-centric assembly (Wang *et al.*, 2015; Li *et al.*, 2017) and subsequent detection of genetic variants. Also, such a subset might again be analyzed within MGX, *e.g.* to perform a closer investigation into the functional potential of just one taxonomic group or, *vice versa*, an analysis of the different taxa that comprise a certain gene.

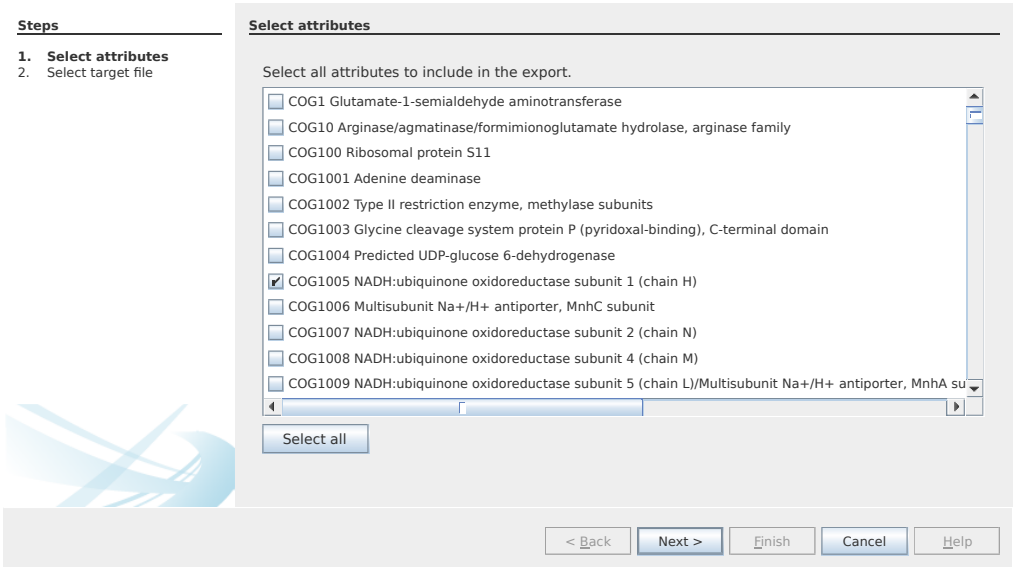


Figure 3.26: Sequence export. MGX retains analysis results with subsequence resolution, thus allowing to selectively export subsets of the data for more focused additional processing.

3.6.7 Search

As all analysis results are retained on the individual sequence level, MGX allows to inspect results with single sequence resolution. The corresponding Search Component (Figure 3.27) enables users to search for arbitrary terms within the annotated Attributes and inspect matching sequences. For each sequence, the desired Attribute is displayed together with all other annotations that refer to this sequence; hereby, analysis results are enriched with additional context, such as *e.g.* taxonomic classifications for a certain gene fragment. Also, results from a single analysis workflow can be independently confirmed by other methods that yielded the same result when a functional annotation is supported by different database matches. Finally, the Search Component allows to obtain the original nucleotide sequence for further processing outside of MGX.

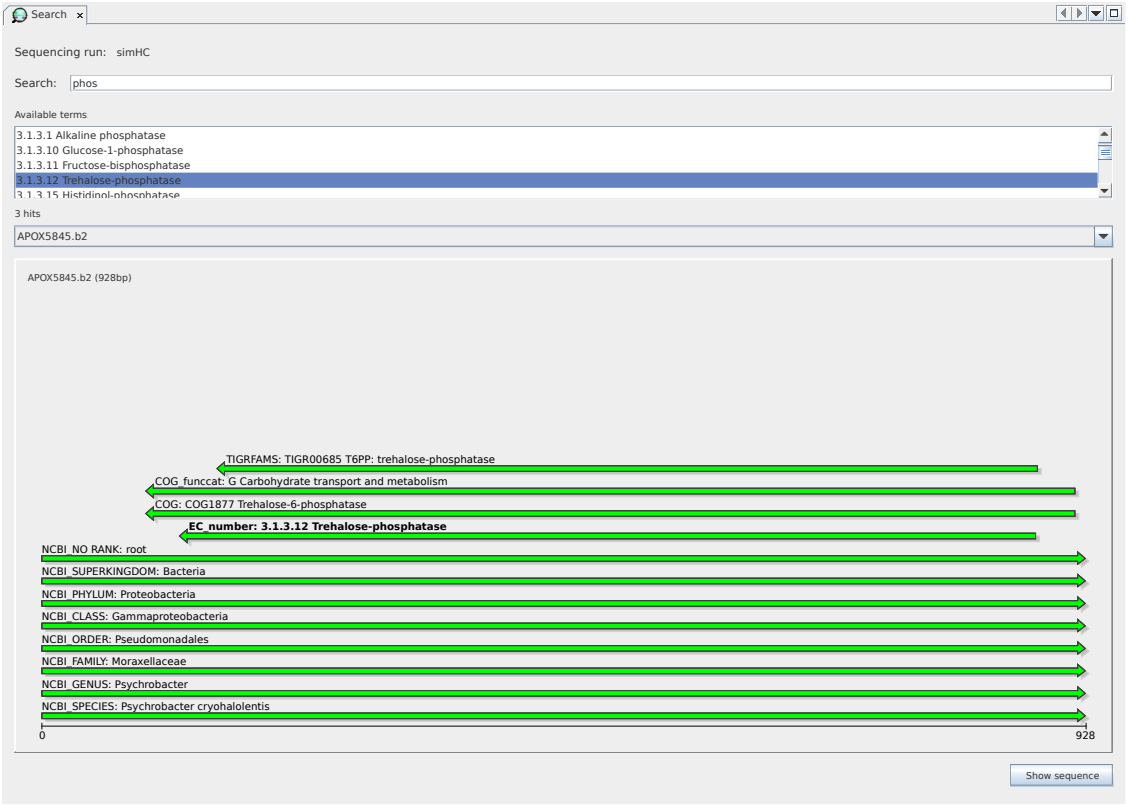


Figure 3.27: Attribute Search. MGX allows to trace annotations down to the individual sequence level, allowing the comparison between different annotation strategies as well as providing additional contextual information. Here, a metagenome sequence assigned to *Psychrobacter cryohalolentis* carries a trehalose-phosphatase fragment, which is independently supported by three different analysis methods (TIGRFAMs, COG, and an EC number).

3.6.8 Statistical data interpretation

In microbial community analysis, *alpha diversity measures* are used to describe the richness of organisms within a single sample and the evenness of their individual abundances, whereas *beta diversity measures* are applied to denote the degree of similarity or dissimilarity between different samples. Biodiversity indices and richness estimates are typical representatives for alpha diversity measures, while the UniFrac distance (Lozupone *et al.*, 2011), Bray-Curtis dissimilarities or the Sørensen index are popular beta diversity measures. All these, however, do not account for the *compositional nature* of microbial community sequencing data; phylogenetic ILR (PhILR; Silverman *et al.*, 2017) is a recently proposed compositional replacement that avoids this shortcoming. Typically, these distance or similarity metrics are

being employed with various statistical evaluation approaches such as volcano plots, PCA or for clustering purposes.

3.6.8.1 Biodiversity indices

Several biodiversity indices have found widespread application as alpha diversity measurements especially in ecology, among them the Shannon index (Shannon, 1948), the Simpson index (Simpson, 1949), and the ACE (Chao and Lee, 1992) and Chao1 (Chao, 1984) species richness estimates. The Shannon index H' represents the information entropy within the assigned sample and is defined as

$$H' = -\sum_{i=1}^{S_{obs}} p_i \ln p_i, \tag{3.1}$$

where p_i denotes the relative abundance of sequences assigned to species i, and S_{obs} is the number of all observed species. The Shannon Evenness E is derived from this value and defined as H' divided by H_{max} ($= \ln S_{obs}$), where H_{max} represents the highest possible Shannon index value. Compared to the Shannon index, the Shannon Evenness has the advantage of a predictable range (0 to 1), which facilitates easier comparability. The Simpson index D is defined as

$$D = 1 - \sum_{i=1}^{S} \frac{n_i(n_i - 1)}{n(n-1)}, \tag{3.2}$$

where S is the number of different categories, n_i the number of entities belonging to category i and n is the total number of entities (Simpson, 1949). The Chao1 richness estimator S_{Chao1} is calculated as

$$S_{Chao1} = S_{obs} + \frac{n_1^2}{2n_2}, \tag{3.3}$$

where n_1 and n_2 denote the number of singletons and doubletons, *i.e.* entities occurring only once or twice, respectively. Finally, the ACE (abundance-based coverage estimator) S_{ACE} is defined as

$$S_{ACE} = S_{abund} + \frac{S_{rare}}{C_{ACE}} + \frac{F_1}{C_{ACE}}\gamma^2_{ACE}, \tag{3.4}$$

where

$$C_{ACE} = 1 - \frac{F_1}{N_{rare}}, \tag{3.5}$$

$$\gamma^2_{ACE} = \max\left[\frac{S_{rare}\sum_{i=1}^{10} i(i-1)F_i}{C_{ACE}(N_{rare})(N_{rare}-1)} - 1, 0\right], \tag{3.6}$$

and

$$N_{rare} = \sum_{i=1}^{10} iF_i. \tag{3.7}$$

Here, S_{rare} denotes the number of rare entities (≤ 10), while S_{abund} is the number of species considered as abundant (> 10) and F_i is the number of species with exactly i sequences assigned. The cutoff value κ that is used to divide the data into rare and abundant groups has initially been defined as 10, but alternative cutoffs have also been proposed, *e.g.* $\kappa = \max(10, n/S_{obs})$ (Chao and Chiu, 2012).

For the various biodiversity indices, it is important to be aware of their calculation and their inherent weaknesses, as *e.g.* Chao1 and ACE are known to underestimate richness for small samples (Hughes *et al.*, 2001). In addition, the γ^2_{ACE} CV (coefficient of variation) has been reported to underestimate "for species-rich and highly heterogeneous assemblages. In such cases, a modified estimator, ACE-1, was derived" (Chao and Chiu, 2014). Also, their special handling of rare entities needs to be accounted for, *e.g.* the role of singletons and doubletons of the Chao1 richness estimator or the κ cutoff of the ACE coverage estimator.

The application of established biodiversity indices is not necessarily restricted to taxonomic assignments; they may also be employed as measures for the diversity of other entities such as *e.g.* annotated genes or gene fragments within metagenomic datasets. In either case, interpretation of an index often remains difficult, as noted for the Shannon index by Hill *et al.* (2003):

> "Crucially, "the difficulty with this statistic is to understand its meaning". This seems to be due to H' being a measure but "not in any way a probability" of the difficulty in predicting the identity of the next bacterial clone. As a result, discussions are typically limited to simply pointing out that a particular sample had the highest H' and hence appears to be the most diverse. But what would it mean if the H' of a hypothetical soil sample fell from 4.5 to 4.1, and would it be a cause for concern?"

Within MGX, biodiversity indices are accessible via the Biodiversity Window (Figure 3.28). The component automatically obtains the currently selected visualization group as well as AttributeType and computes the most commonly used alpha diversity measures. These are available as implementations of the StatisticI interface (Listing 3.4), and additional implementations are easily added to the application.

Listing 3.4: The StatisticI interface. Supported alpha diversity indices are provided as implementations of the StatisticI interface.

```
public interface StatisticI {

    /**
     * @return name of the implemented statistic
     */
    public String getName();

    /**
     *
     * @param distribution attribute distribution for selected group
     * @return pre-formatted String with computed value
     */
    public String measure(DistributionI<Long> distribution);

}
```

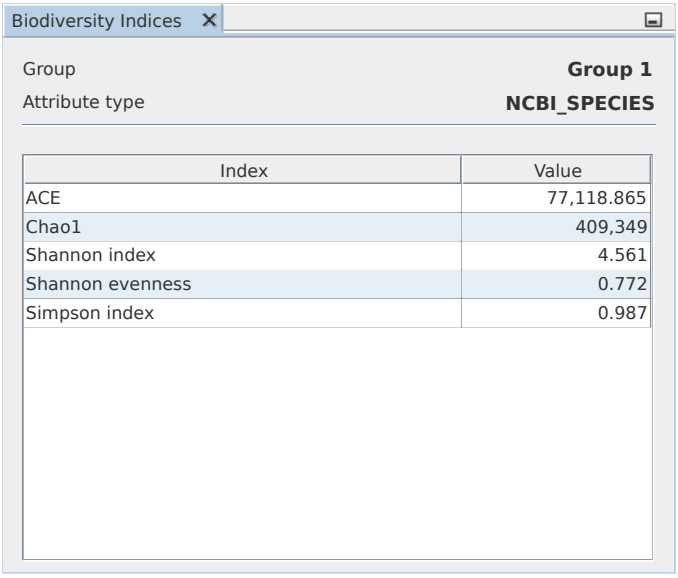


Figure 3.28: Biodiversity Indices. For statistical evaluation purposes, the Biodiversity Window provides various alpha diversity measures that are commonly used to estimate ecological diversity.

3.6.8.2 Compositional data

With a fixed upper bound for the number of sequences they can produce, data obtained from sequencing instruments has to be considered as compositional data (Quinn *et al.*, 2018b). This represents an important difference from *e.g.* ecology,

where co-existence of different groups as well as absolute abundances can be assessed. By counting the different species, their relative abundances as well as the individual population sizes are determined.

In studies employing high-throughput sequencing (HTS), however, absolute abundances cannot be determined, as the capacity of the sequencing device limits the total number of entities that can be counted, and an increase of the abundance of one species in an environment indispensably is required to reduce the relative abundance of others (Gloor *et al.*, 2017). Hence, assessment of the microbial composition within an environment by HTS will yield equal results for sparsely as well as densely populated areas as long as their relative compositions are equal, and there exists no means to distinguish between the two based on the data. "Thus, the total read count observed in a HTS run is a fixed-size, random sample of the relative abundance of the molecules in the underlying ecosystem. Moreover, the count can not be related to the absolute number of molecules in the input sample" (Gloor *et al.*, 2017). While a single dataset is able to represent the relative proportion of entities within a sample, it does not allow to deduce the total number of bacteria or DNA molecules present in a certain habitat. This is especially important when comparing multiple datasets, as a reduction of the relative abundance of one species does not necessarily impose that it was displaced by another one – instead, the total abundance of organisms might just have increased.

Compositional data are vectors where the individual components denote the relative magnitude in relation to the other contents of the vector; the total sum of the compositional vector is related to the data generation procedure, *i.e.* the size of the sequencing library, and does not indicate actual abundance (Soneson and Delorenzi, 2013). Hence, the individual components of the vector can not be interpreted isolated from each other, as their meaning is provided only by the context. The sample space of compositional data is called the simplex $\mathcal{S}^D$ and defined as

$$\mathcal{S}^D = \{[x_1, x_2, \ldots, x_D] \mid x_j > 0; j = 1, 2, \ldots, D; \sum_{j=1}^{D} x_j = c\}, \tag{3.8}$$

where c is an arbitrary constant and typically depends on the measured unit, *e.g.* the number of sequences. As the simplex does not represent an Euclidean space, many statistical metrics typically applied in order to *e.g.* determine correlations or distances cannot directly be employed. In *Compositional Data Analysis* (CoDA), it is therefore necessary to either exclusively apply simplex-aware methods or to perform an initial transformation of the compositional data vector. The Aitchison distance $d^2(x, y)_a$ is the distance in the simplex and defined as

$$d^2(x,y)_a = \frac{1}{2D}\sum_{i=1}^{D}\sum_{j=1}^{D}(\ln\frac{x_i}{x_j} - \ln\frac{y_i}{y_j})^2 \tag{3.9}$$

It denotes the equivalent of Euclidean distance measurement in compositional simplex space. For data transformation, several approaches have been proposed, among them the additive log-ratio transformation (alr) which standardizes the components of the vector x with regard to a reference feature x_D:

$$\mathrm{alr(x)} = \left[\ln\tfrac{x_1}{x_D}; \ldots; \ln\tfrac{x_{D-1}}{x_D}\right]. \tag{3.10}$$

The centered log-ratio transform (clr) avoids the need to select a reference element and uses the geometric mean $g(\mathrm{x}) = \sqrt[D]{x_1 \cdot x_2 \cdot \cdots \cdot x_D}$ of the vector instead, *i.e.*

$$\mathrm{clr(x)} = \left[\ln\tfrac{x_1}{g(x)}; \ldots; \ln\tfrac{x_D}{g(x)}\right], \tag{3.11}$$

while the isometric log-ratio transform (ilr; Egozcue *et al.*, 2003) is based on the clr transform, but preserves all metric properties:

$$\mathrm{ilr(x)} = \mathrm{clr(x)} \cdot V \tag{3.12}$$

Here, V is a matrix fulfilling $V \cdot V^t = I_{D-1}$, with I_{D-1} being the $D-1$-dimensional identity matrix.

A major issue for the statistical analysis of compositional data arises from the introduction of zeros into a dataset, which commonly and inevitably occurs *e.g.* during comparative analysis methods when an entity is not detected in one of the samples. "Before handling zeros, the analyst must first consider the nature of the zeros. There exist three types of zeros: (1) rounding, also called sampling, where the feature exists in the sample below the detection limit, (2) count, where the feature exists in the sample, but counting is not exhaustive enough to see it at least once, and (3) essential, where the feature does not exist in the sample at all. The approach to zero handling depends on the nature of the zeros. [...] Since there is no general methodology for dealing with essential zeros within a strict CoDA framework, we assume that any feature present in at least one sample could appear in another sample if sequenced with infinite depth, and thus treat all NGS zeros as 'count zeros'" (Quinn *et al.*, 2018a). Several strategies have been suggested to cope with this situation, most prominently *feature removal*, where features containing zeros are excluded from the analysis, and *feature modification*, which replaces all occurrences of zeros with a small value. As the replacement of zeros must not

modify the sum of all entities, non-zero values need to be adapted, as well. MGX employs the additive replacement strategy for zeros that was proposed by Aitchison (1986). Briefly, a composition $x \in \mathcal{S}^D$ containing one or several zeros is converted into a composition $y \in \mathcal{S}^D$ that does not contain zeros by replacing the individual values by

$$y_i = \begin{cases} \frac{\delta(Z+1)(D-Z)}{D^2}, & \text{if } x_i = 0, \\ x_i - \frac{\delta(Z+1)Z}{D^2}, & \text{if } x_i > 0, \end{cases} \tag{3.13}$$

where D denotes the total number of elements in the composition, Z the number of zeros, and a δ a small positive value[12] (Martín-Fernández *et al.*, 2003).

3.6.8.3 Rarefaction

Rarefaction is a technique often employed to assess coverage of a microbial community based on the analysis results of a dataset; rarefaction analysis is therefore conducted to determine whether the conducted sequencing effort is sufficient or if additional sequencing data is required before valid conclusions might be drawn. The method itself originates in ecology and is widely applied to microbial community data, even though it is not capable of handling compositional data and should thus be avoided to compare multiple samples. A rarefaction curve is created by repeatedly subsampling a dataset at different sizes and observing the number of different entities, *e.g.* taxonomic assignments, that still occur in the reduced ("rarefied") subset. Plotting the subset size on the abscissa versus the number of entities on the ordinate axis then yields the rarefaction curve (Figure 3.29), and the slope of the curve denotes the rate at which new entities are being discovered. The rarefaction curve increases monotonically to a maximum, which is reached at the point representing the size of the complete dataset and the number of different entities contained therein.

[12] MGX uses R's builtin value .Machine$double.eps as a pseudocount, which is computed as the smallest positive floating-point number x such that $1 + x \neq 1$.

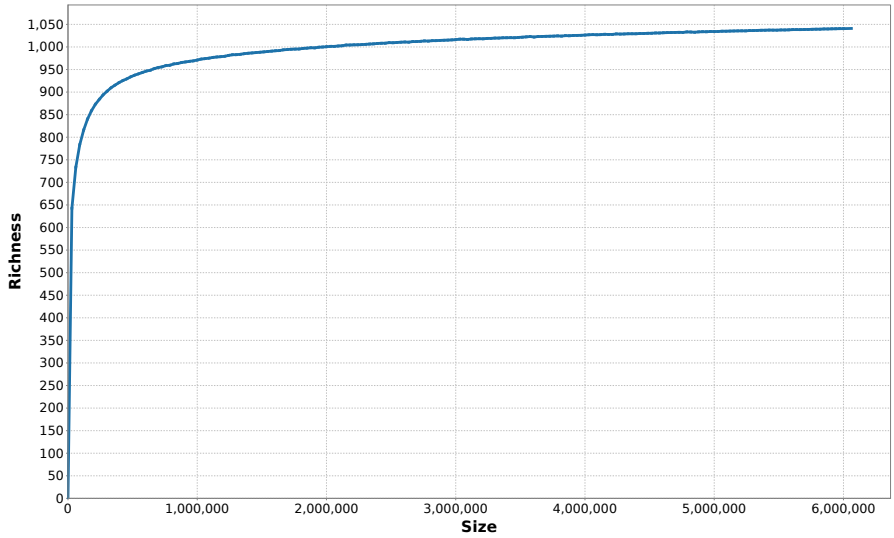


Figure 3.29: Rarefaction curve. A rarefaction curve is created by randomly subsampling the data at various depths and plotting the subsample size versus the number of distinct groups still present in the drawn sample. The depicted curve has almost reached saturation, indicating sufficient sequencing effort.

The rarefaction implementation initially used by MGX was based on the vegan R package and computed by the MGX server; for performance reasons, this approach has since been revised and now relies on a Java version of the algorithm that was presented in the rarefaction toolkit (RTK; Saary *et al.*, 2017). The algorithm achieves a significant speedup over the original implementation by reusing the shuffled feature vector as long as possible instead of permuting it each time before drawing a subset. Additionally, the Java version was adapted to use the faster Xorshift64* RNG[13] (Marsaglia, 2003) instead of the Mersenne Twister RNG (MT19937; Matsumoto and Nishimura, 1998) used by the RTK.

3.6.8.4 M/A plot and Volcano plot

The M/A plot is a log-scaled variant of the Bland-Altman plot (Bland and Altman, 1986) and commonly used to visually identify differences between two samples. This plot type has found wide application especially for the interpretation of microarray data, but also for certain high-throughput sequencing application such as RNA-Seq, or metagenomics. An M/A plot is generated by plotting the log-scaled mean abundance (intensity) versus the log-scaled ratio (difference) of the different entities encountered in the two datasets (Figure 3.30). The Volcano plot is similar to the M/A plot, but plots the fold-change versus the significance as determined by

[13]Random Number Generator

a statistical test. Therefore, the computation of Volcano plots typically requires replicates.

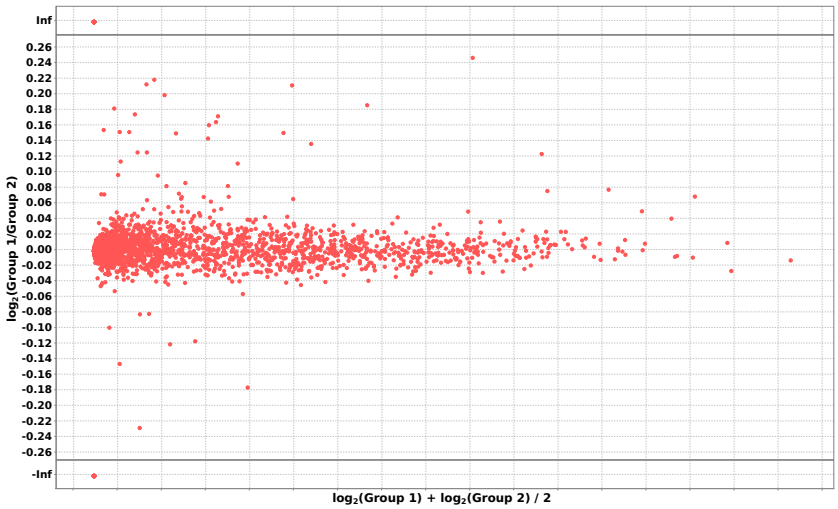


Figure 3.30: M/A plot. The M/A plot can be employed to visually identify differences between two datasets. The higher the total share of an entity, the further to the right it is displayed; the greater the difference in abundance between the two groups, the more to the top or bottom the data point is displayed.

3.6.8.5 Ordination plots

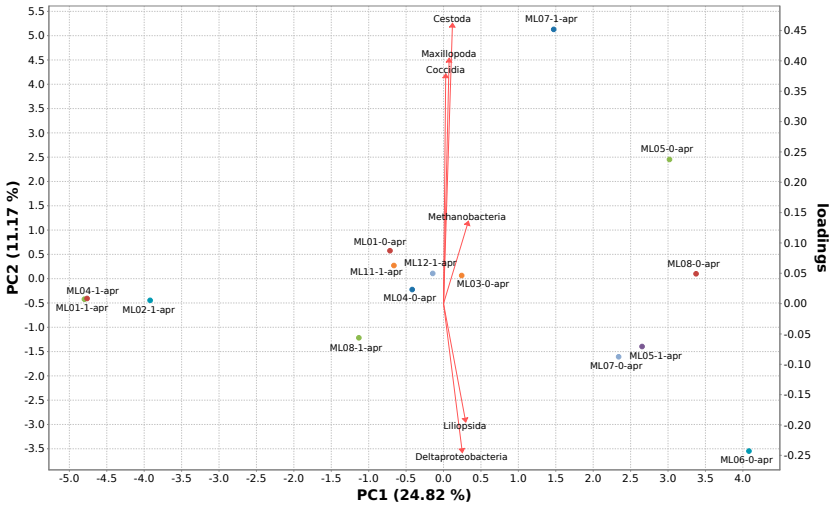


Figure 3.31: Principal Component Analysis. PCA plot of several microbial community datasets; in addition to the datasets, the most relevant eigenvectors are depicted as vectors. Axis labels also include information about the amount of variation.

Principal Component Analysis (PCA; Figure 3.31) and Principal Coordinates Analysis (PCoA), also known as Multidimensional Scaling (MDS), are statistical approaches to visualize similarities and dissimilarities of several datasets. These analysis techniques are applied in order to obtain a graphical representation of beta diversity between datasets and to identify possible inherent structures within the data that might not be obvious. PCA is a dimensionality reduction technique that transforms a large number of possibly correlated values into a smaller amount of uncorrelated values, which are computed as linear combinations of the input data.

PCoA/MDS is a similar approach, but while PCA is used with a similarity matrix, PCoA requires a dissimilarity matrix; in the special case where the matrix is created using Euclidean distances, PCoA is identical to PCA. Also, some potential problems arise "with PCoA if the selected distance is not metric, because some eigenvalues may be negative and then, the graphical representation will not perform properly" (Calle, 2019). To mitigate this issue, NMDS (Non-metric Multidimensional Scaling; Figure 3.32), which uses rank orders, can be used instead. NMDS, as a result, is a much more flexible technique that is able to handle a variety of types of data. The MGX implementations of PCA and NMDS are based on the R prcomp() and metaMDS() functions (vegan package); both were adapted to use the Aitchison distance after additive replacement of zeros.

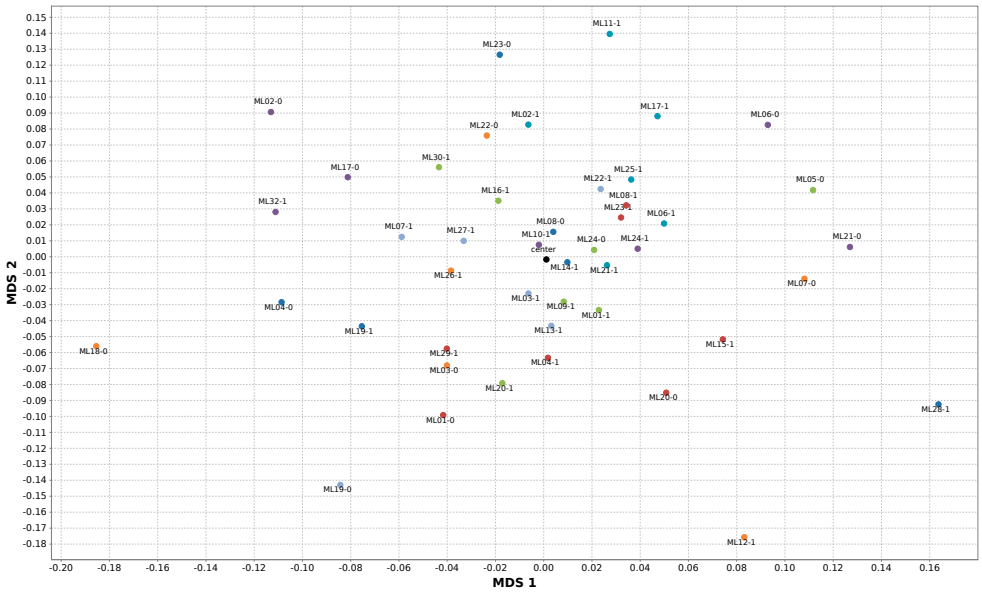


Figure 3.32: Non-metric Multidimensional Scaling (NMDS). NMDS plot of microbial community data based on taxonomic assignments at rank "family". Dissimilarities were computed using the Aitchison distance with additive replacement of zeros.

3.6.8.6 Hierarchical clustering

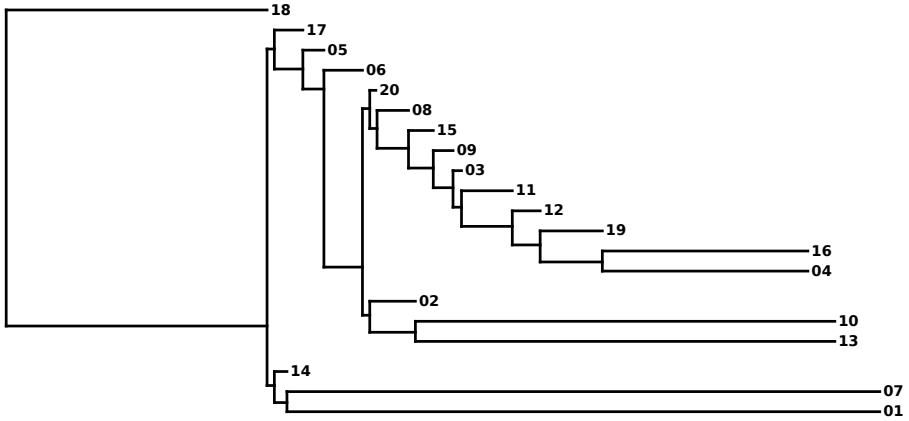


Figure 3.33: Hierarchical clustering. The dendrogram plot is provided to visualize the outcome of a hierarchical clustering of multiple datasets based on metagenome analysis results.

Cluster analysis is a process performed in order to partition several datasets into groups, thereby allowing to unveil hidden patterns in the data; "dissimilarities and similarities are assessed based on the attribute values describing the objects and often involve distance measures. Clustering as a data mining tool has its roots in many application areas such as biology, ..." (Han *et al.*, 2011). With hierarchical clustering, dissimilarities between data groups are used to arrange them into a tree structure (dendrogram). For the clustering of metagenome datasets, agglomerative clustering is mostly employed; this bottom-up approach places each dataset within its own cluster, and the hierarchical structure is obtained by iteratively identifying and merging the two most similar clusters. Alternatively, a divisive approach (top-down) can be applied, where all datasets are initially placed in just one cluster, which is then iteratively split until a termination condition is met, *e.g.* the number of desired target clusters.

Within MGX, hierarchical clustering is provided via the server-side R installation and based on the dist() and hclust() functions provided by the stats package; accordingly, the user is free to select between different methods for distance measurement and agglomeration. However, due to the compositional nature of analysis results, the Aitchison distance (Section 3.6.8.2) is the suggested default distance metric; the Aitchison distance implementation used for clustering within MGX is based on the isometric log-ratio transform (ilr) after additive replacement of zeros. Clustering is performed based on obtained analysis results, hence allowing to use *e.g.* taxonomic or functional profiles for different datasets as an input. The MGX GUI implements a graphical dendrogram viewer for clustering results (Figure 3.33); in addition, the outcome may also be exported to a file in Newick format.

3.6.9 Reference mapping

Reference mapping, the alignment of metagenomic or metatranscriptomic short sequences to annotated reference genomes, represents another feature provided by the MGX application. These mappings are commonly performed in order to identify conserved genomic regions, genetic expression patterns, or to validate taxonomic classifications. In addition, aligned metagenome sequences might also be extracted from a dataset for either subsequent variant detection (for closely related genomes) or even subjected to *de novo* assembly, where such a selective enrichment procedure represents an adequate method to reduce the risk of misassemblies, which commonly occur during metagenome assembly.

Figure 3.34: Reference mapping. The Mapping Window showing alignment results for a metatranscriptome dataset mapped to the reference genome of one of the dominant organisms. From top to bottom, the component displays a) navigation and coverage histogram, b) currently selected interval and c) aligned DNA sequences for the interval. Color coding of the aligned sequences encodes relative identity with regard to the reference.

Different analysis pipelines employing popular and established read mapping tools such as Bowtie 2, FR-HIT, or MagicBLAST are provided within MGX; each server hosts a repository of public and annotated reference genomes obtained from NCBI

GenBank, or the desired reference sequence can be imported by the user in EMBL, GenBank, or FASTA format, thus allowing to incorporate unpublished confidential genomes.

The result of a mapping analysis is typically inspected visually, and MGX currently offers two distinct plot types for this purpose: The alignment plot (Figure 3.34) displays individual sequences aligned to the reference genome, thus enabling easy identification of coverage information; the fragment recruitment plot (Figure 3.35) displays aggregated relative identity of aligned reads, allowing to infer the degree of sequence conservation. The graphical representation can be exported in a variety of image formats; also, the complete mapping result is available for download in BAM format and can be employed for additional downstream analyses.

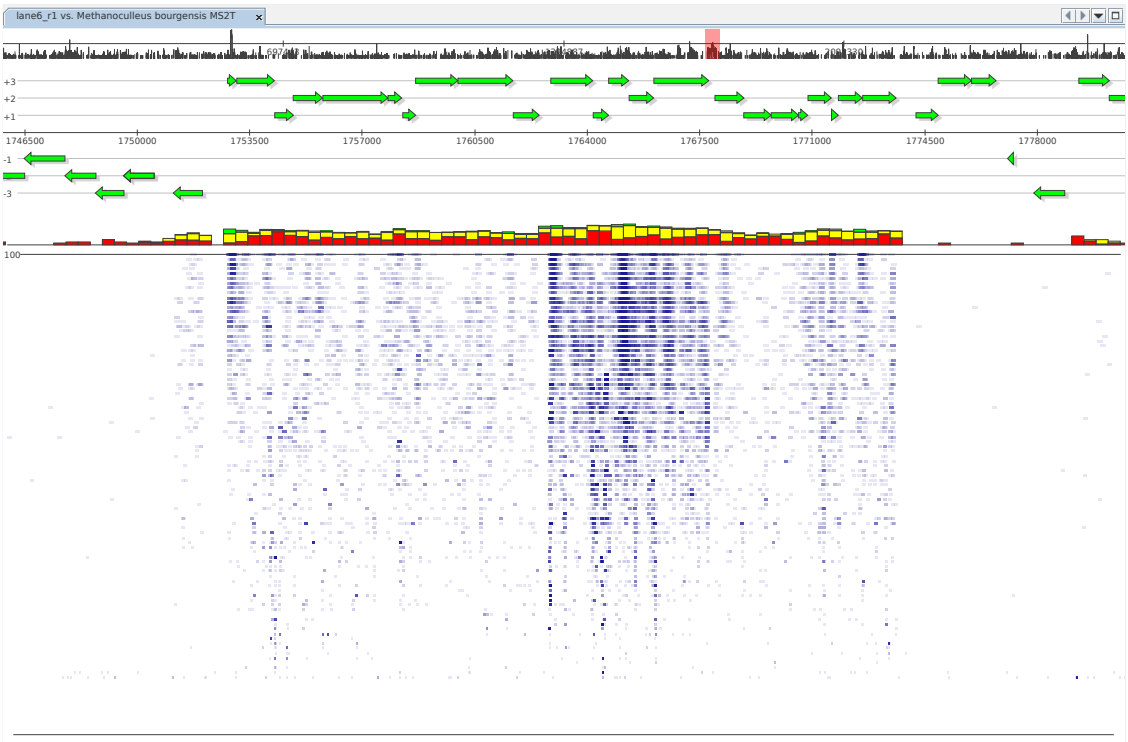


Figure 3.35: Fragment recruitment. An alternate visualization mode for reference mappings is the fragment recruitment plot, here displaying the same genomic subregion as the previous figure. This view mode features the fragment recruitment plot itself and additionally provides stacked bars summarizing mapping identity within reference intervals, grouped into low (red), medium (yellow; $\geq$ 75%) and high (green; $\geq$ 97% sequence identity) quality mappings.

3.6.10 Fidelity assessment of workflows

Within the course of a M.Sc. thesis conducted by Patrick Blumenkamp under my supervision, a novel MGX module, the Evaluation Component, was developed. This module allows to compare different analysis workflows among each other or to a selected reference annotation. Various performance indicators (Table 3.6) are computed and can be inspected; also, the overall runtime of different analysis workflows can easily be compared (Figure 3.36). While the Venn diagram viewer is able to display the overlap between up to four different analysis jobs, it does not consider differing relative abundances (Figure 3.37); if these should be taken into account, either a distance measurement such as the Aitchison distance or the weighted UniFrac distance can be applied to compute a distance matrix or a Quantification accuracy plot might be generated to compare two analysis results (Figure 3.38). For the import of external annotation data, for example a predefined gold standard, the proprietary MGS file format (Listing 3.5) was also developed. MGS is a text-based format offering single-read resolution which contains the necessary information in order to create the corresponding attributes and attribute types within an MGX project. Lines starting with the keyword READ identify the metagenomic sequence to be annotated, and sequence names are required to be unique within a dataset. All subsequent lines are required to start with the AT tag and contain the value of the attribute type, the value of the desired attribute, 0-based coordinates for the targeted subregion of the sequence, the required attribute characteristics (hierarchical, discrete, ...), and a final column which serves as a discriminator between multiple hierarchies.

Table 3.6: Performance indicators measured by the evaluation component. All parameters are computed with regard to a reference annotation, which can be a predefined standard of truth or assignment results obtained from any other analysis tool.

Indicator
True positive
False positive
False negative
True negative
Sensitivity
Specificity
Precision
Negative predictive value
False positive rate
False negative rate
False discovery rate
Accuracy
F1 score

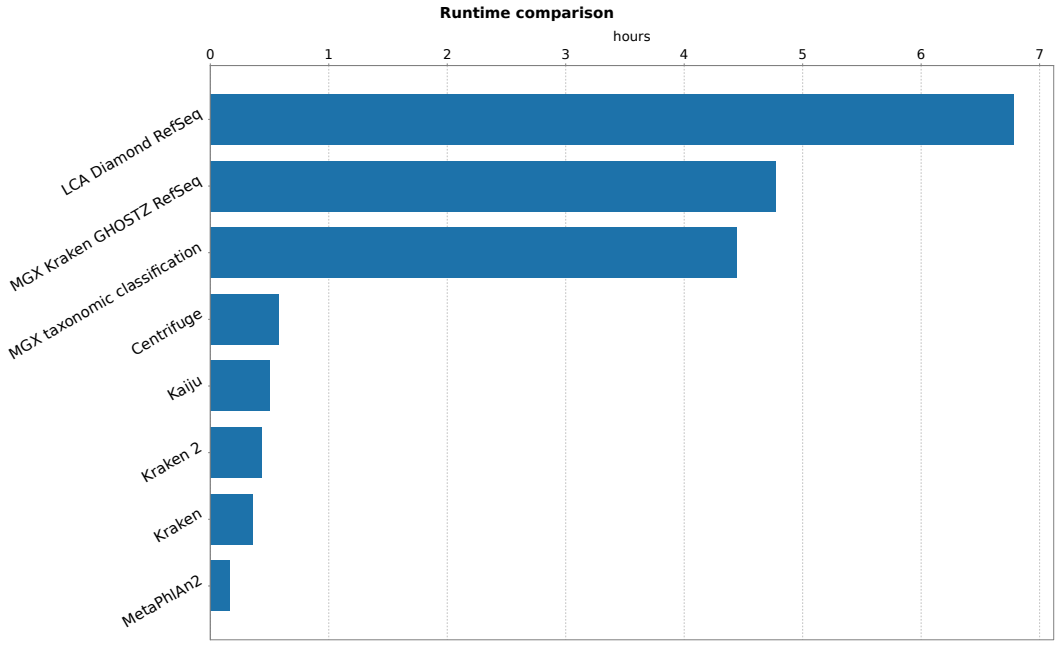


Figure 3.36: Runtime comparison. The Evaluation Component allows to assess overall runtime required for different types of analysis workflows.

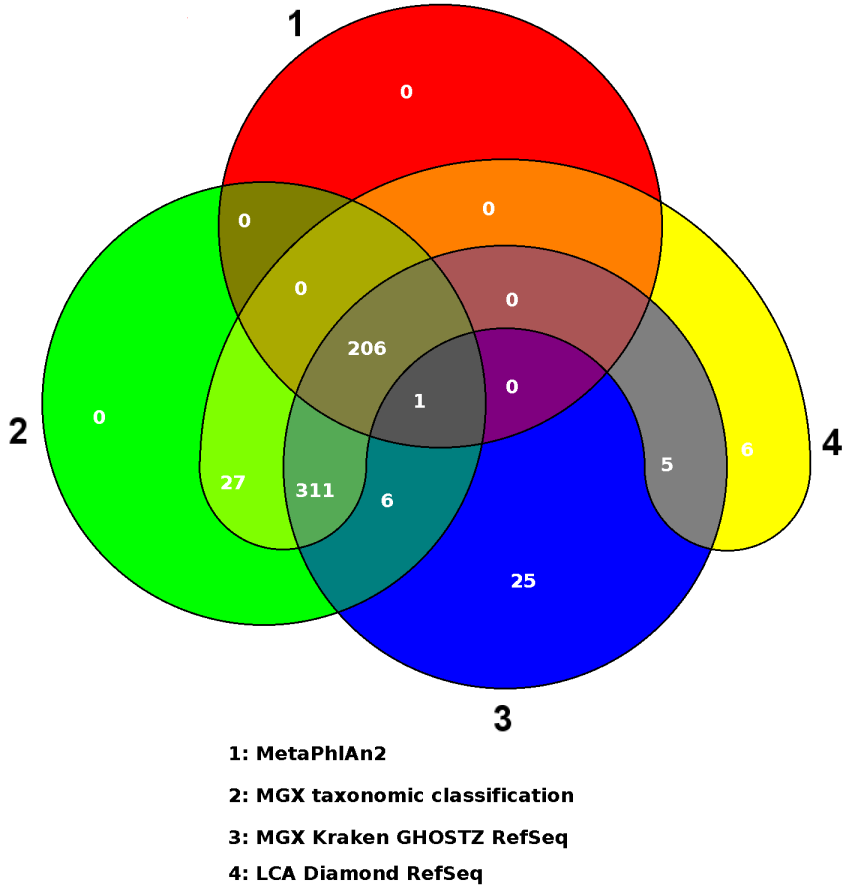


Figure 3.37: Workflow comparison. Based on the Venn diagram viewer, results originating from up to four different workflows can be compared. The Venn diagram, however, does not take relative abundances into account, and only the overlap of generated Attributes is depicted in the chart.

Listing 3.5: MGS file format. The text-based MGS format can be used to import an external annotation for a metagenome dataset, *e.g.* a reference annotation. The excerpt represents the taxonomic assignment for a single sequence named 1005039_407 onto the major taxonomic ranks, which are provided as attributes (AT) with hierarchical (H) and discrete (D) attribute type. Additional columns indicate the subregion to be annotated.

```
READ    1005039_407
AT      NCBI_NO RANK       Root                    0    99    HD    1
AT      NCBI_SUPERKINGDOM  Bacteria                0    99    HD    1
AT      NCBI_PHYLUM        Armatimonadetes         0    99    HD    1
AT      NCBI_CLASS         Fimbriimonadia          0    99    HD    1
AT      NCBI_ORDER         Fimbriimonadales        0    99    HD    1
AT      NCBI_FAMILY        Fimbriimonadaceae       0    99    HD    1
AT      NCBI_GENUS         Fimbriimonas            0    99    HD    1
AT      NCBI_SPECIES       Fimbriimonas ginsengisoli 0  99    HD    1
```

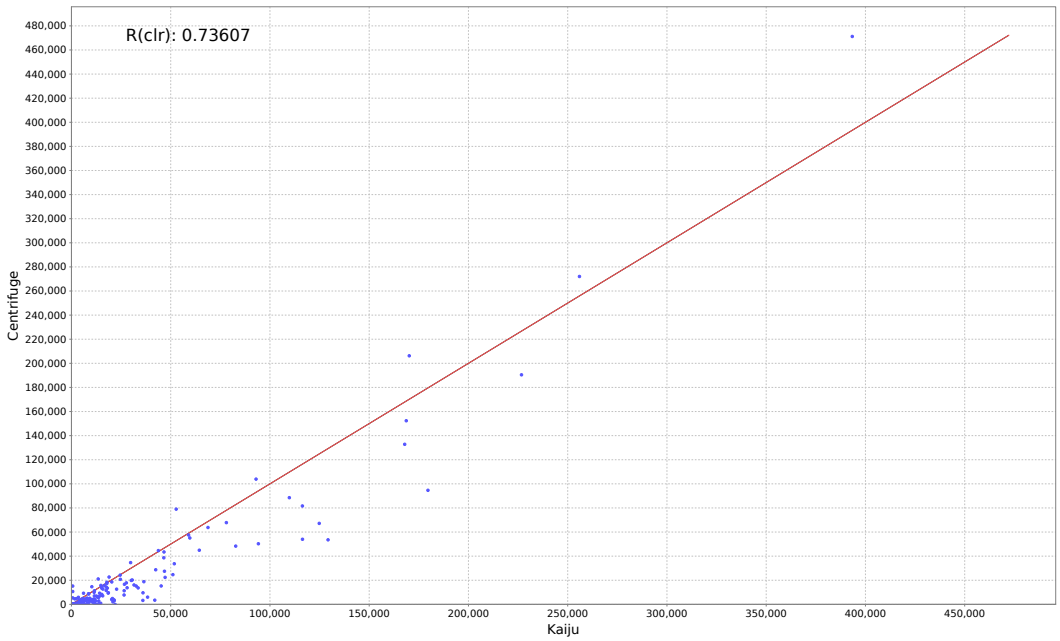


Figure 3.38: Quantification accuracy. A scatter plot generated for the results of two different workflows depicts the relative abundance of different classification results. Also, Pearson's correlation coefficient is automatically computed after a clr-transformation of the input data.

3.7 Currently available pipelines

Table 3.7: Taxonomic classification. Taxonomic assignment pipelines currently implemented within MGX.

Name	Approach/Description
LCA	LCA taxonomic classification compatible to MEGAN
16S-Pipeline	16S rRNA gene fragment classification
MetaPhyler	Marker-based assignment
MetaCV	Composition-based classification
Kraken	k-mer-based taxonomic assignment
Kraken 2	minimizer-based taxonomic classification
MetaPhlAn	Marker-based assignment
MetaPhlAn 2	Marker-based assignment
Kaiju	Taxonomic assignment based on maximum exact matches
Centrifuge	Rapid and sensitive classification of metagenomic sequences
MGX default	Taxonomic classification based on Kraken and GHOSTZ

The set of available workflows aimed at taxonomic classification of metagenomes (Table 3.7) reflects, to a certain degree, the development and continuous improvement of the MGX framework over time. Initially, methods applied for the analysis of two 454 pyrosequencing-based metagenomes originating from a biogas fermenter plant (Jaenicke *et al.*, 2011) were re-implemented as Conveyor-based workflows and integrated into the application. The "16S-Pipeline", for example, is a Conveyor workflow version of a Perl script that was used in the aforementioned study. It executes an initial sequence homology search in order to identify fragments carrying partial 16S rRNA genes; this is followed by a taxonomic classification step based on the RDP Classifier (Section 2.3.4; Wang *et al.*, 2007). Additional, more recent methods were added over time following internal evaluation and often based on requests by MGX users asking for a particular tool to be integrated.

Table 3.8: Functional assignment. The functional analysis pipelines currently implemented within MGX employ the major established databases in order to ascertain the functional characteristics of a metagenome.

Name	Approach/Description
EggNOG	Assignment to COG groups and categories
KEGG pathways	KEGG pathway mapping based on EC numbers annotated using the SwissProt database
Pfam	Identification of protein families in the Pfam database
TIGRFAMs	Protein family alignment vs. the TIGRFAMs database
ClusterMine360	Screening for PKS/NRPS domains
dbCAN	Identification of carbohydrate-active enzymes
FunGene	Functional analysis based on FunGene functional genes

Functional characterization of unassembled datasets (Table 3.8) is typically performed based on the results of a homology search after a preceding step performing gene fragment identification employing FragGeneScan+ (Kim *et al.*, 2015). Workflows targeting different sequence and HMM databases are provided covering the most frequently used functional characterization tasks. Due to a large interest especially in the analysis of potential genes involved in antibiotic resistance and possible virulence, a distinct set of workflows employing the most commonly used databases in this field has been implemented for this purpose (Table 3.9).

Table 3.9: Antimicrobial resistance determination. Several antimicrobial resistance pipelines provided within MGX employ the most popular databases available for this purpose.

Name	Approach/Description
MVirDB	Antibiotic resistance genes, toxins or virulence factors based on alignment vs. the MVirDB database
ARDB	Antibiotic resistance gene screening based on the ARDB database
BacMet	Antibacterial biocide- and metal-resistance gene annotation
CARD	Resistance gene annotation using the CARD database
ARG-ANNOT	Antibiotic Resistance Gene-ANNOTation

Table 3.10: Reference alignment. Several analysis pipelines based on established short- as well as long-read alignment algorithms are currently offered within MGX.

Name	Approach/Description
Bowtie	Alignment to reference genomes based on the Bowtie 2 aligner
FR-HIT	Fragment recruitment employing FR-HIT
MagicBLAST	Fragment recruitment using MagicBLAST
Minimap 2	Long-read fragment recruitment

For the creation of fragment recruitment plots, workflows based on three different short-read aligners are currently available within MGX (Table 3.10). An additional workflow employs Minimap 2 (Li, 2018) and is capable of handling third-generation sequencing data such as long reads obtained from either PacBio or Oxford Nanopore sequencing instruments.

Table 3.11: Metataxonomics pipelines. Overview of metataxonomic analysis workflows. Support for the analysis of amplicon sequence data within MGX is currently restricted to taxonomic classification employing different approaches.

Name	Approach/Description
mothur	16S rRNA classification using Wang's method
RDP	Bayesian 16S rRNA and fungal ITS amplicon classification
QIIME	16S rRNA amplicon classification via uclust or SortMeRNA

As MGX has primarily been designed for the analysis of metagenome and metatranscriptome datasets, support for the processing of metataxonomics data is currently rather limited. Nonetheless, several analysis workflows for the taxonomic classification of these types of data were implemented based on collaborator requests (Table 3.11). Both mothur and QIIME are popular tools for this task and were integrated by Patrick Blumenkamp, while the RDP workflow is a variant of the "16S-Pipeline" for metagenome analysis without the initial homology search step.

Table 3.12: Pipeline templates. Analysis pipeline templates currently provided by MGX. These templates need additional user-provided input data such as own sequence databases or reference genomes.

Name	Approach/Description
BestHit-Blast	Annotation of best homology hit employing user-provided nucleotide or amino acid database
BestHit-HMM	Best HMMer hit annotation employing user-provided HMM amino acid models
Discard-Host	Contamination removal based on user-provided reference genomes

Finally, an additional set of predefined workflows is provided for users that want to integrate own datasources into their analysis. These workflows (Table 3.12) can be applied in conjunction with own HMM- or sequence-based databases, reference sequences or may just serve as a starting point when implementing an own custom workflow.

4

Evaluation and benchmarking of taxonomic classification approaches for environmental DNA sequences

Look deep into nature, and then you will understand everything better.

– Albert Einstein

Inferring the taxonomic composition of a microbial community is one of the central tasks in metagenome analysis, and several tools are available to accomplish this task. This chapter presents the design and implementation of a taxonomic classification workflow for MGX. The corresponding performance evaluation demonstrates that the implemented pipeline is able to outperform existing solutions. Also, subsequent improvements that were performed after the original publication are described. Contents of this chapter are partially based on the manuscript covering the MGX

framework (Jaenicke *et al.*, 2018), where the presented taxonomic classification pipeline was initially described.

4.1 Taxonomic classification approaches

Taxonomic profiling of microbial communities is one of the most important aspects in metagenome data analysis; hence, a sophisticated taxonomic classification workflow should also be provided within MGX in addition to the variety of published tools. However, as the implementation of a novel classification system that exceeds the performance of existing solutions is a cumbersome undertaking, a different approach was chosen: Employing the components that were already available and integrated into the Conveyor workflow engine within the course of this thesis, an improved pipeline was implemented. For this, a combination of established and well-tested classifiers was elected based on discriminative power, attainable throughput and availability as a Linux command-line tool.

Still, if more than one tool shall be used to analyze a dataset, it is necessary to decide on a way how the individual and sometimes even contradicting predictions should be handled:

- If tools are to be run in parallel, an additional component is required that will merge multiple results; this can be implemented as either a majority-based decision with equally weighted or differentially scored tools, or results can be combined using an LCA (*lowest-common-ancestor*) or LCA* (Hanson *et al.*, 2016) approach.

- If tools are run in succession as a pipeline with several annotation stages, an optimal order has to be determined in which the tools should be arranged. For this, no additional components are required, as each tool's input will only be the fraction of the dataset that was not already classified by the preceding steps. In a simplistic approach, tools are sorted based on the number of false positive predictions they produce in ascending order, and additional tools are added to the end of the pipeline as long as they contribute to an overall improvement. However, it is also important to keep runtime considerations in mind, and it might still be worthy to deviate from this order and either omit individual tools or let faster tools take precedence over slower ones. Following this strategy, scalability and overall throughput is improved, even if slightly suboptimal results are achieved.

- A combination of both approaches.

4.2 MGX classification workflow

The taxonomic analysis workflow itself was designed as a combination of the popular Kraken classifier enhanced by a *lowest-common-ancestor* stage employing DIAMOND (Buchfink *et al.*, 2014) and the RefSeq protein database (Figure 4.1). This approach was chosen considering the high throughput of Kraken and the sensitivity offered by both approaches. Additionally, the combination of a k-mer- with an alignment-based algorithm can assumed to be beneficial as a general-purpose workflow that is applied to both next-generation as well as third-generation sequencing datasets.

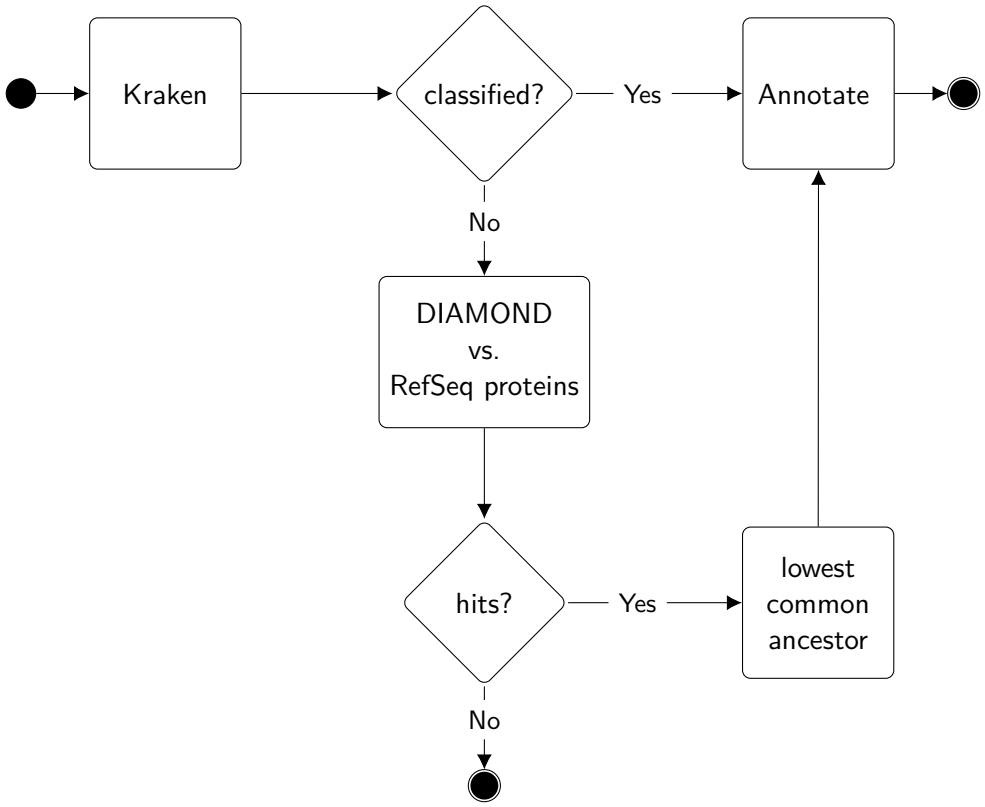


Figure 4.1: Taxonomic classification workflow. Classification of environmental DNA sequences is performed employing two different tools; initially, a k-mer-based classification algorithm (Kraken) is applied, and for sequences that remain unassigned, a sequence homology search based on DIAMOND and the RefSeq protein database is conducted. If homologous proteins are found in the database, the taxonomic origin of a sequence is deduced as the lowest common ancestor of all database hits.

4.2.1 Generation of benchmark datasets

The performance of the resulting pipeline was evaluated using two synthetic metagenome datasets mimicking scenarios where the source organism was present or absent from the underlying taxonomic classification database. Both artificial benchmark datasets were created using the Mason read simulator (Holtgrewe, 2010) with a read length of 100 bp, an Illumina-specific error profile and a mismatch probability of 3 percent (–seq-technology illumina –illumina-prob-mismatch 0.03).

In the first test scenario (organism present in database), a reference database of complete archaeal and bacterial genomes restricted to one genome per species was created based on NCBI RefSeq. From this set of 2,672 finished genomes, 5,000 reads were sampled from each genome and the resulting artificial metagenome (13,360,000 reads) containing only fragments of known taxonomic origin was subsequently analyzed.

The second experiment (clade exclusion, organism not present in database) was performed to evaluate the classification performance based on sequences not contained in the database of each tool. For this, an additional synthetic metagenome dataset based on NCBI GenBank was compiled, including only genomes where the annotated genus was present, but the species absent in NCBI RefSeq. From this set of 2,376 genomes, an artificial metagenome comprising 11,880,000 reads was obtained as described above.

4.2.2 Performance evaluation

For comparison, several existing taxonomic classification tools were selected based on recency and reported frequency of use according to literature: Kraken (Wood and Salzberg, 2014), Kaiju (Menzel *et al.*, 2016), Centrifuge (Kim *et al.*, 2016) and MetaPhlAn 2 (Truong *et al.*, 2015). All tools were run with their respective default settings, with databases generated in May 2017.

The performance evaluation of all tools was conducted within MGX using the Evaluation Component (Section 3.6.10), which allows to compare tool results either among each other or to a previously imported reference annotation ("gold standard"). Classification performance was evaluated at the genus level (Table 4.1).

In the first experiment, the classification performance using simulated reads generated from the NCBI RefSeq genomes database was assessed; as NCBI RefSeq is also the origin of sequences used to initially create the classification databases for Kraken, Centrifuge and Kaiju, a very high precision in excess of 98% was observed for all tools, with Kraken and MGX showing the highest overall precision

(both 99.84%) followed by Centrifuge (99.58%), MetaPhlAn 2 (98.30%) and Kaiju (98.05%).

Table 4.1: Taxonomic classification performance on genus level for benchmark datasets. All tools achieve high precision on the RefSeq-derived metagenome (a), as the source organisms are already included in the relevant classification databases. For the GenBank-based metagenome containing only species not present in the tools databases (b), MetaPhlAn 2 offers high precision but only a very low sensitivity (0.78%), followed by the MGX-provided default pipeline, which ranks highest in sensitivity, accuracy as well as F1 score. TP: True Positives; FP: False Positives; FN: False Negatives; numbers in bold denote best results.

(a) RefSeq-derived synthetic metagenome

	Kraken	Kaiju	Centrifuge	MetaPhlAn 2	MGX
TP	12,059,412	9,329,288	12,611,380	414,943	12,566,362
FP	18,748	185,899	53,092	7,171	20,698
FN	1,281,840	3,844,813	695,528	12,937,886	772,940
Sensitivity	0.9039	0.7082	**0.9477**	0.0311	0.9421
Precision	**0.9984**	0.9805	0.9958	0.9830	**0.9984**
Accuracy	0.9027	0.6983	**0.9440**	0.0311	0.9406
F1 score	0.9488	0.8224	**0.9712**	0.0602	0.9694

(b) GenBank-derived synthetic metagenome

	Kraken	Kaiju	Centrifuge	MetaPhlAn 2	MGX
TP	1,851,436	2,592,655	2,175,122	92,383	3,976,270
FP	398,899	1,230,445	864,989	10,378	734,389
FN	9,629,665	8,056,900	8,839,889	11,777,239	7,169,341
Sensitivity	0.1613	0.2435	0.1975	0.0078	**0.3568**
Precision	0.8227	0.6782	0.7155	**0.8990**	0.8441
Accuracy	0.1558	0.2182	0.1831	0.0078	**0.3347**
F1 score	0.2697	0.3583	0.3095	0.0154	**0.5015**

In the second experiment, the artificial metagenome derived from NCBI GenBank was used, which solely contains genomes from genera where the species was not present in the classification databases of the individual tools (clade exclusion). In this experiment, MetaPhlAn 2 achieved the highest precision (89.90%) of all tools, but with a very low sensitivity of only 0.78% due to the fact that MetaPhlan 2 is relying on a small set of marker genes. The MGX classification pipeline not only showed the second highest precision (88.41%), but also the highest sensitivity (35.68%), accuracy (33.47%) and F1 score (50.15%) of all evaluated tools (Table 4.1).

4.2.3 Conclusion

While all standalone tools already provide valuable results and correctly assign a large number of sequences, the performance of the taxonomic analysis pipeline as implemented in MGX is either equivalent to or even exceeding that of comparable tools. However, the majority of sequences remained unclassified by all of the tools, showing there is still a lot of room for future improvement in metagenomic sequence analysis.

Even without the development of a novel algorithm, the performance that was achieved clearly demonstrates that a sophisticated combination of existing methods can be superior to these methods applied individually in order to reach a certain goal. The obtained results emphasize the strengths of the workflow-based approach and as time progresses, the workflow in its current form can easily be adapted by adding new components or replacing existing ones with improved alternatives.

4.3 Sequence alignment evaluation

Runtime considerations are an important aspect for metagenome analysis apart from the quality of the results generated by a certain algorithm. Often, the performance values provided in the corresponding publications of the different tools available for sequence alignment are not directly comparable, as different hardware, target databases, or only a subset of available tools have been assessed. Also, many researchers naturally tend to report only that part of results that most favors their own tool, while drawbacks of the individual tools are reported less enthusiastically. Thus, an independent evaluation was performed in order to assess the runtime of the different tools as well as their ability to identify the same best database hit.

4.3.1 Runtime measurement

In the first experiment, runtime measurements were taken to estimate the performance and scalability of various commonly applied tools during the alignment task of nucleotide sequences using the SwissProt database (comprising 554,860 protein sequences) as a target. All tools (Table 4.3) were executed using one single thread, and query sequences were randomly subsampled from the benchmark dataset generated in the previous section (artificial 100 bp sequences derived from NCBI RefSeq; Section 4.2.1) using fastq-sample[1]; also, the target databases were copied into shared memory (/dev/shm) in order to avoid any performance impact caused by disk I/O.

Table 4.3: Alignment tools. For benchmarking purposes, various tools for sequence homology search were compared.

Algorithm	Version
BLAST+	2.7.1+
PALADIN	1.4.4
DIAMOND	0.9.10
RAPSearch	1.02
RAPSearch 2	2.24
GHOSTX	1.3.6
GHOSTZ	1.0.2

The experiment was performed on a HPE (Hewlett-Packard Enterprise) ProLiant DL380 server machine with two physical Intel® Xeon® Gold 6126 CPUs (12 cores each) with hyperthreading enabled and 128 GB of system memory. Runtime and memory consumption were measured with the UNIX time command; several iterations were performed for each input size and results averaged.

[1]fastq-tools 0.8; `https://github.com/dcjones/fastq-tools`

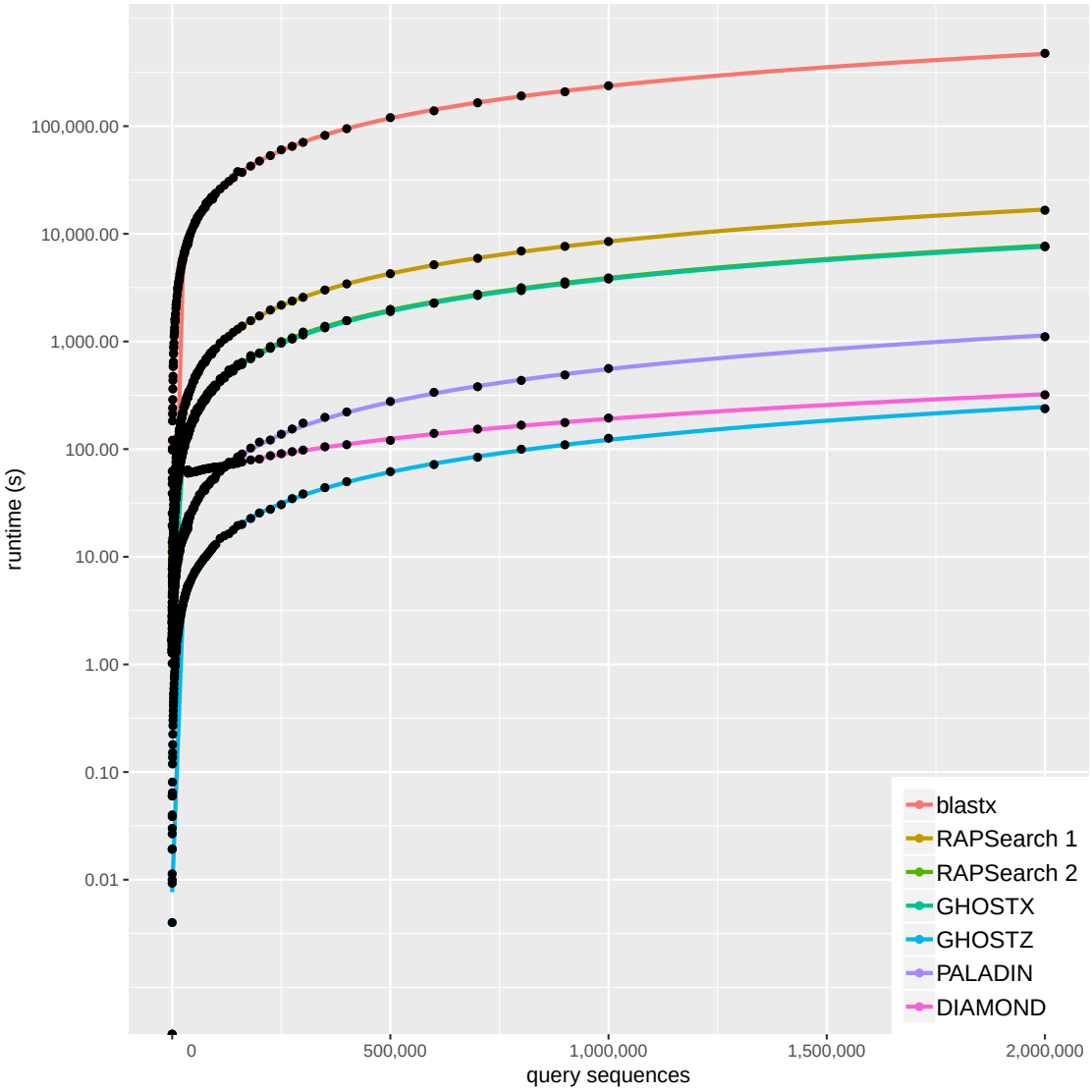


Figure 4.2: Runtime of alignment programs. For all tested input sizes, GHOSTZ exhibits the best performance. Even for larger input chunks (data not shown), GHOSTZ still outperforms its closest competitor, DIAMOND.

As expected, the runtime of all evaluated programs demonstrated linear scalability with regard to the number of query sequences used (Figure 4.2), and more recent algorithms showed improved runtime compared to older ones, *e.g.* RAPSearch 2 was notably faster than RAPSearch 1 and GHOSTZ clearly outperformed GHOSTX. An interesting observation was made for the runtime of the DIAMOND aligner, where runtime unexpectedly more than doubled once a certain number of query sequences needed to be processed: 35,751 queries required 23.5 secs to align, while 35,752 queries took 57.7 seconds wallclock time (Figure 4.3). According to Benjamin

Buchfink, DIAMOND's author, this is caused by DIAMOND employing a query-indexing scheme for small numbers of input sequences, while larger amounts of query sequences cause a switch to a double-indexing approach.[2] In the test scenario, this switch already occurs for a rather unfavorably low number of inputs; however, DIAMOND allows to manually specify the desired indexing scheme as a command line parameter instead of relying on a potentially suboptimal automatism.

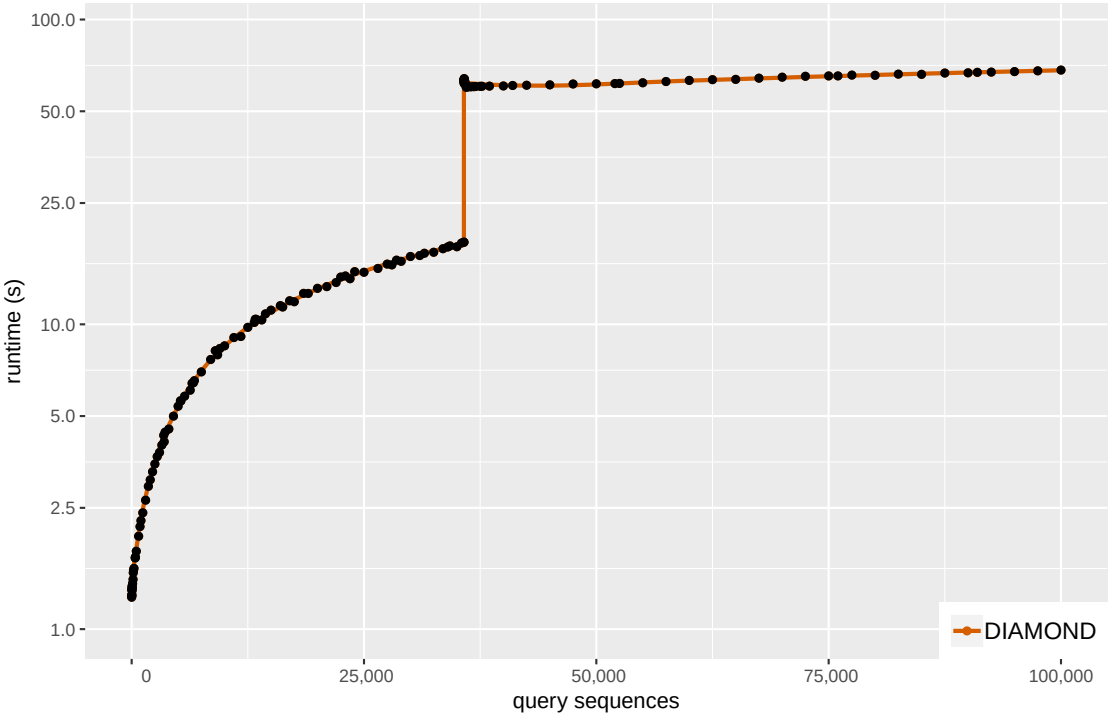


Figure 4.3: DIAMOND runtime (Excerpt from Figure 4.2). Runtime of the DIAMOND aligner more than doubles between 35,751 and 35,752 query sequences due to a switch between different algorithms: For small query files, DIAMOND employs a query-indexing approach, while a double-indexing algorithm is used for larger inputs.

With a speedup of more than 1,900 compared to BLAST+ (for 500,000 query sequences; Table 4.4), GHOSTZ provides superior performance which by far exceeds that of all other tools for the tested conditions. Interestingly, this no longer holds true for larger reference databases: An identical test performed with the RefSeq protein database shows a performance advantage in favor of DIAMOND, which needed 10,739.0 seconds to align 100,000 query sequences, while GHOSTZ required 17,919.1 seconds for the same amount of input data; the difference becomes even more apparent for larger numbers of inputs – DIAMOND processed 500,000 queries in just 11,504.6 seconds, while GHOSTZ took 85,848.6 seconds. The deviation from

[2]personal communication

the results reported in the respective literature (Table 2.3) as well as the observed behavior of DIAMOND for differently-sized inputs stress the importance to perform own experiments that most closely resemble the intended use case.

Table 4.4: Speedup over BLAST+. Of all compared programs, GHOSTZ by far outperforms all other sequence homology search tools regardless of the number of query sequences. Subscripted numbers denote the number of query sequences.

Algorithm	$\text{Speedup}_{50,000}$	$\text{Speedup}_{500,000}$	$\text{Speedup}_{1,000,000}$
BLAST+	1.0	1.0	1.0
RAPSearch	28.1	28.2	27.9
RAPSearch 2	60.1	60.5	60.6
GHOSTX	64.6	63.3	62.3
PALADIN	425.9	433.2	422.62
DIAMOND	196.2	963.8	1220.1
GHOSTZ	1727.1	1943.5	1880.3

4.3.2 Accuracy evaluation

In the second experiment, an evaluation of result quality was performed. In this scenario, the functional analysis of a metagenome (or metatranscriptome) dataset was assessed; even while a variety of different algorithmic approaches are available for taxonomic classification, functional profiling almost exclusively relies on sequence homology searches to identify matching protein fragments, and traditionally, BLAST is being employed for this purpose. As no read simulator is currently able to generate artificial sequences that purposefully overlap with annotated genes, generation of an appropriate gold standard dataset for functional profiling was not possible in this case; instead, the ability of the different tools to reproduce the results of the original BLAST algorithm was measured.

Applying the MGX framework, workflows were designed that perform the functional annotation of a synthetic metagenome dataset derived from public NCBI RefSeq genomes based on the best hit of a sequence homology search versus the SwissProt protein database using the different alignment algorithms. PALADIN was excluded from this experiment, as it does not emit an E-Value fidelity estimate and thus cannot be used with comparable cutoff settings; all workflows were configured to use an E-Value cutoff of 1×10^{-5} and executed within MGX. The evaluation component within MGX was then employed to directly compare the results of dif-

ferent tools; the annotation generated by the workflow based on BLAST+ was used as the standard of truth.

Table 4.5: Best-hit accuracy. Among all compared tools, GHOSTZ is able to generate the results most closely resembling those obtained with the original BLAST+ program. TP: True Positives; FP: False Positives; FN: False Negatives; r_{clr}: Pearson correlation after centered log-ratio transform. Numbers in bold denote best results.

	RAPSearch	DIAMOND	GHOSTZ	GHOSTX	RapSearch 2
TP	30,676	1,241,922	1,530,031	1,432,905	33,033
FP	15,591	629,879	440,255	544,666	15,000
FN	2,640,251	903,584	767,492	760,889	2,639,415
TN	10,673,482	10,584,615	10,622,222	10,621,540	10,672,552
Sensitivity	0.0115	0.5788	**0.6659**	0.6532	0.0124
Specificity	0.9985	0.9438	0.9602	0.9512	**0.9986**
Precision	0.6630	0.6635	**0.7766**	0.7246	0.6877
Accuracy	0.8012	0.8852	**0.9096**	0.9023	0.8013
F1 score	0.0226	0.6183	**0.7170**	0.6870	0.0243
Pearson's r	0.2409	0.7698	**0.9206**	0.9171	0.3143
Pearson's r_{clr}	0.0952	0.6569	**0.7332**	0.7222	0.1398

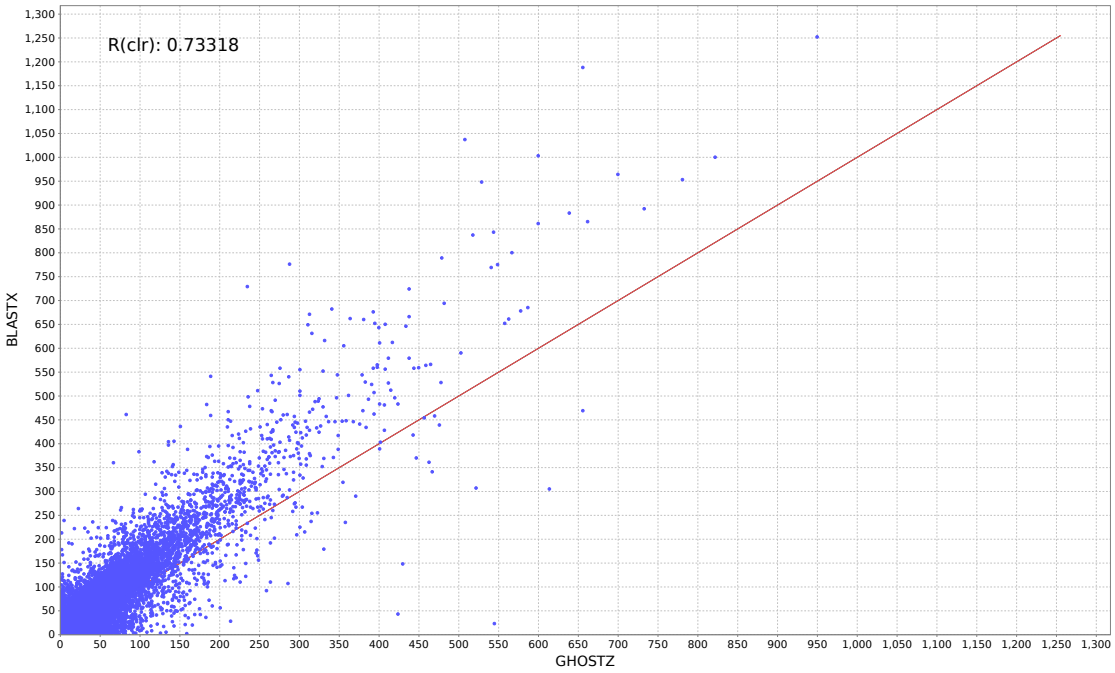


Figure 4.4: MGX accuracy plot. Results obtained with GHOSTZ exhibit the highest Pearson correlation coefficient ($r_{clr} = 0.7332$) with the reference assignment computed with BLAST+. Each data point represents a SwissProt database hit and its position reflects the number of sequences assigned by BLAST and GHOSTZ, respectively. Based on the plot, a general trend towards less hits being reported by GHOSTZ can be observed and suggests additional future experiments.

Interestingly, both RAPSearch and RAPSearch 2 demonstrated the highest specificity in this experiment, but at the same time, they only showed very low sensitivity ($< 1.5\%$; Table 4.5). DIAMOND produced mediocre results, and GHOSTZ showed the highest sensitivity, precision, accuracy (Figure 4.4), as well as F1 score. Also, the functional profile obtained with GHOSTZ showed the highest Pearson correlation coefficient[3] (0.7332) when compared with the BLAST-based analysis, followed by GHOSTX (0.7222) and DIAMOND (0.6569).

4.3.3 Refinement of the taxonomic classification workflow

The conducted experiments demonstrate GHOSTZ to be capable of providing excellent performance and also yielding results most closely resembling those obtained with the original BLAST+ program. However, due to restrictions of available read simulation programs, this fact alone does not suffice to recommend GHOSTZ for

[3]Pearson correlation was computed after centered log-ratio (clr) transformation.

metagenome analysis, as the best hit obtained with BLAST+ not necessarily has to be in accordance with the correct source organism or protein-coding gene.

Even though DIAMOND demonstrated slightly better runtime when applied with the RefSeq protein database, GHOSTZ was introduced into the taxonomic classification workflow that was provided with the original manuscript describing the MGX framework (Section 4.1) as an alternative for the DIAMOND program. Subsequently, the evaluation was repeated for the DIAMOND- as well as GHOSTZ-based variants using the original benchmark datasets.

Table 4.6: Taxonomic classification performance on genus level for benchmark datasets. Replacement of DIAMOND with GHOSTZ in the taxonomic classification workflow used by MGX resulted in improved overall performance. Numbers are not directly comparable to those given in Table 4.1 due to more recent database versions. TP: True Positives; FP: False Positives; FN: False Negatives; numbers in bold denote best results.

(a) RefSeq-derived metagenome

	DIAMOND variant	GHOSTZ variant
TP	12,320,504	12,393,601
FP	178,382	179,896
FN	861,114	786,503
Sensitivity	0.9347	**0.9403**
Precision	**0.9857**	**0.9857**
Accuracy	0.9222	**0.9277**
F1 score	0.9595	**0.9625**

(b) GenBank-derived metagenome

	DIAMOND variant	GHOSTZ variant
TP	3,528,413	3,764,555
FP	657,601	683,152
FN	7,693,986	7,432,293
Sensitivity	0.3144	**0.3362**
Precision	0.8429	**0.8464**
Accuracy	0.2970	**0.3169**
F1 score	0.4580	**0.4813**

The obtained findings document the superiority of the GHOSTZ-based variant of the taxonomic classification workflow in terms of assignment quality, resulting in improved sensitivity, precision and accuracy (Table 4.6). The presented methodology clearly illustrates the advantages of the MGX framework for the continuous improvement of analysis pipelines: The workflow-based approach allows to easily replace individual pipeline components, and the evaluation module facilitates rapid assessment of the enhancements that were possibly achieved by the modification. As a consequence, the improved GHOSTZ-based variant has since been promoted and currently represents the recommended analysis workflow for the taxonomic characterization of metagenome and metatranscriptome datasets within the MGX framework.

5

Discussion and Outlook

Scientists have become the bearers of the torch of discovery in our quest for knowledge.

– Stephen Hawking

5.1 Discussion

For decades, studying the DNA of microorganisms required the creation of a lab culture followed by (low-throughput) sequencing with traditional Sanger-based capillary sequencing instruments – a tedious and also cost-intensive task. However, it is widely known that the majority of microorganisms cannot be cultured under laboratory conditions (Streit and Schmitz, 2004), and hence this approach greatly restricted which type of organisms could be studied by researchers around the world. Without the need for prior generation of a laboratory culture, the advent of next-

generation sequencing opened up a window to access an enormous variety of novel genetic information that was previously unattainable. These advances have largely contributed to the emergence of a new research field called metagenomics, where DNA originating from microbial communities is obtained directly from the habitat they are living in, sequenced and finally analyzed. These communities are often quite diverse, and without the large output of modern sequencing instruments, the required resolution typically could not be achieved. Metagenomics is a fascinating topic and has evolved into a field offering a large number of opportunities, especially with a focus on the understanding of nature as well as permitting access to, for example, novel enzymes and compounds with potential applications in biotechnology as well as medicine.

MGX is a novel solution for the management and analysis of such metagenome sequence datasets. Developed as a flexible and extensible client/server framework, it provides all the required means for successful storage, processing and interpretation of environmental sequence data. MGX features a comprehensive set of adaptable workflows required for taxonomic and functional metagenome analysis, combined with an intuitive and easy-to-use graphical user interface offering customizable result visualizations. At the same time, MGX allows to include own data sources such as unpublished reference genomes or sequence databases and devise custom analysis pipelines, thus enabling researchers to perform basic as well as highly specific analyses within a single application.

With MGX, researchers gain access to a large collection of distinct analysis capabilities accomplishing taxonomic classification as well as functional assignment of environmental sequence data. These are implemented as workflows for the Conveyor workflow engine, which offers type safety and allows transparent distribution of resource-intensive tasks to high-performance compute clusters. Workflow systems facilitate easy exchange of individual components once improved approaches become available. This is also valid for MGX – pipelines that depend on sequence homology searches were initially offered with BLAST, which was eventually replaced by faster methods such as GHOSTX, then DIAMOND, and finally, GHOSTZ, while still providing the same functionality. Apart from rather generic approaches such as the assignment of sequences to COG groups or EC numbers, MGX also features highly specialized workflows aimed at microbial resistance screening or the identification of gene fragments involved in the synthesis of secondary metabolites. For prevalent organisms, users of the MGX framework may choose to perform a fragment recruitment of metagenome sequences versus a published or private reference genome; for this purpose, several workflows tailored for different types of sequence data are included. The workflow-based approach has proven especially advantageous for the fast inclusion of new tools, and most of them were integrated within a short timeframe after they were published.

Within MGX, sequence data is always retained together with appropriate metadata, and properties describing the provenience of each dataset accompanied with a description of applied sampling and sequencing library preparation procedures are mandatorily inquired from the user. Hereby, potential treatment differences are easily identified and the source of a potential adapter contamination can be traced back to the applied sequencing protocol. Also, the availability of the corresponding metadata eases the submission process of a dataset to one of the public nucleotide archives such as the NCBI SRA (short read archive), which is nowadays required by most journals before a publication is accepted.

The unique result model that was developed for MGX allows to represent arbitrary results and fully captures their intrinsic characteristics, therefore making them independent of external data sources such as taxonomy databases that are subject to change over time. All resulting annotations are retained with sub-sequence resolution in the MGX project database, hence enabling users to trace results back to the individual sequence level and for example create data subsets based on discretionary criteria.

The MGX graphical user interface addresses end users, offering user-friendly and wizard-driven data entry complemented with a wide range of convenient high-quality visualizations that are dynamically offered depending on the desired analysis type; these visualizations comprise basic data plotting capabilities as well as advanced statistical evaluation methods. Currently, the majority of all novel algorithms for metagenome data analysis are released as command-line tools for the Linux operating system; employing the MGX GUI, users gain access to these advanced tools, and it is not necessary for them to acquire profound expertise in using the Linux command line, as control over the parameterization and execution of analysis workflows is easily achieved. Also, all workflows within MGX already come with sensible default parameters that were defined based on own experience, and while users are able to adapt these settings as they like, this is not mandatory. After analysis completion, MGX facilitates the means for interactive exploration of taxonomic and functional assignments, allows to determine community coverage, compute common biodiversity indices, or perform various statistical evaluation methods. With its modular design, the MGX application is easily extensible, and new visualizations or statistical approaches are frequently added based on user requests.

For advanced users or those who wish to include MGX into their routine data analysis tasks, the MGX library is provided and allows to easily automate all required steps via the REST API exposed by each MGX server. Also, no local compute resources need to be provisioned, as MGX comprises the necessary compute infrastructure and its maintenance and operation is currently secured by the de.NBI network[1].

[1] German Network for Bioinformatics Infrastructure

Some aspects relevant to sequence data analysis and metagenomics in particular have not sufficiently and satisfactorily been addressed in the past. At the moment, none of the major platforms for metagenome data analysis addresses the compositional nature of the sequencing data generated in microbial community studies; while an experienced statistician will be capable to manually address this aspect by custom programming, the majority of users is rather unlikely to question the results that they're offered by a ready-made software. Here, it is up to the software developers that provide data analysis capabilities to also steer their users towards the correct methods for data interpretation. MGX provides a wide range of different statistical analysis types for biodiversity estimation, dimensionality reduction (PCA, NMDS), or clustering, that allow to assess and compare the different properties of environmental DNA datasets. Where necessary, MGX suggests the use of appropriate functions for distance measurement such as the Aitchison distance, or applies suitable data transformations before invoking the desired method.

However, MGX does not only offer the required infrastructure for metagenome analysis, but also the means that allow to assess the performance of newly designed analysis types. The Evaluation Component is provided in order to evaluate accuracy and runtime of different approaches, and analysis results can be compared either among each other or with regard to a predefined reference annotation. While this feature is rather unlikely to be used by the typical MGX user, it is a worthwhile addition to MGX and frequently used to evaluate novel workflows before making them publicly available; also, developers of new algorithms will find this feature useful for the assessment of their own approaches.

The taxonomic classification pipeline that was implemented for MGX demonstrates the advantages of the chosen workflow-based approach; with comparably little effort, a pipeline was designed as a combination of previously existing tools which exceeds the performance of the tools applied individually and provides superior accuracy when compared to established taxonomic assignment algorithms. Built from an initial Kraken step with a subsequent *lowest-common-ancestor* assignment based on the RefSeq protein database, the pipeline currently serves as the recommended workflow for taxonomic profiling within MGX. As new and improved methods become available, the pipeline can easily be extended or individual components can be replaced. The outcome of the performed runtime evaluation also greatly stresses the importance of assessing published tools in their intended use case scenario, as becomes obvious when comparing the published speedups of popular homology search tools to those obtained from own experiments. However, it is important that even if runtime is an important aspect especially for the large data volumes of current metagenomes, a slower approach might still yield better results and should therefore be preferred. Of course, this tradeoff is only acceptable to a certain extent, and tools like BLAST are nowadays deemed unsuitable for metagenome analysis due to runtime considerations.

Read-based metagenomics relies on the information content contained in short, error-prone sequences, and even while recent developments provide improved classification accuracy or faster analysis times, the amount of available information within such a short fragment is limited. It is hence especially important to apply the most recent state-of-the-art methods for metagenome data analysis in order to provide scientists with the best obtainable results for their studies. Particularly third-generation sequencing data poses a significant challenge due to the high error rate that is currently inherent to these technologies, and alignment-based approaches as implemented by MG-RAST or IMG/M perform suboptimally for these types of data.

MGX is a viable and promising alternative to the various existing tools and platforms for metagenomics. It offers a large selection of the most recent methods combined with a rich set of visualizations and postprocessing options, while other applications provide less sophisticated approaches and sometimes even employ tools with known shortcomings.

With jobs being distributed to large infrastructures like XSEDE, the MG-RAST platform (Meyer *et al.*, 2008) achieves an impressive throughput and is in high demand; despite its popularity, the application however partially relies on outdated tools, improvable algorithmic approaches and provides only very basic functionality for the processing of individual datasets. Furthermore, statistical interpretation and comparative analysis need to be performed mostly outside of MG-RAST. Also, some aspects of the functional analysis pipeline rely on quite antiquated databases; the annotation of EC numbers, for example, is based on the latest publicly available version of the KEGG database, which dates back to 2011, when KEGG underwent a license change and has been offered on a paid subscription basis ever since. IMG/M follows a similar approach to MG-RAST, and both solely rely upon the *lowest-common-ancestor* approach for taxonomic profiling. Also, with the restriction to assembled metagenome submissions, users will need significant compute resources in order to assemble their datasets prior to uploading them to IMG/M. On the other hand, the MG-RAST platform is in high demand, and it will typically take at least several weeks before analysis results may be inspected. Both platforms provide a predefined default analysis pipeline which does not allow any additional customization; as far as MG-RAST is concerned, results can at least be filtered based on the obtained E-Values to a certain extent.

Apart from MGX, none of the currently existing platforms offers such an extensible range of different and highly specialized analysis pipelines. Also, none of them allows advanced users to include own data sources or to devise and implement custom workflows. Features such as fragment recruitment or the wide range of supported statistical methods are currently unique to MGX and not present in any of the other web-based platforms such as MG-RAST or IMG/M. In terms of offered features, MEGAN 6 comes closest to MGX, but as a standalone desktop application,

significant local compute resources are required. Finally, MGX can easily be scaled horizontally, as a single GUI instance is able to connect to multiple MGX servers in parallel and perform *e.g.* a comparison of datasets that reside in geographically separated servers. Hence, MGX fulfills all the requirements that were stated initially (Section 3.1), and the rich set of features provided is at present unrivaled in its extent. While established metagenomics platforms provide predefined static analysis pipelines, no system offers the high degree of flexibility and customizability available in MGX. Workflow systems like Galaxy, on the other hand, can be employed to implement own analysis pipelines, but lack in offered visualizations or downstream statistical evaluation. Also, – apart from custom programming – no other framework provides such a large number of different and highly specific analysis workflows or allows to devise own custom-tailored pipelines.

The biggest advantage of the MGX application is also one of its limitations: Fast development cycles and quick adoption of new methods to a certain degree hinder comparability of results obtained with different tool or database versions. For the two MGX instances currently hosted at the Bioinformatics and Systems Biology group at JLU Gießen and the Center for Biotechnology at Bielefeld University, tool versions and databases are kept in sync, but this obviously cannot be ensured for future servers operated by third parties. The only alternative, however, would have been a development model based on "fixed point in time"-releases, and tools as well as databases must not be updated until a new release is made. Even then, results from different release versions would not be comparable, at least unless old results would be discarded and recomputed based on the new *status quo*, a computationally intensive effort.

Another issue is mostly caused by the freedom of choice offered by MGX; on several occasions, users did not actively choose a pipeline for some intended purpose, but instead executed all analysis pipelines offering a certain type of result and presumably chose the result which offered best support for their hypothesis. Obviously, this problem is present with other solutions as well, as users might upload and analyze their data with both MG-RAST and IMG/M, and only retain those results which best match their expectations. During the development of the MGX framework it became apparent that there is a clear demand for the education of users; not only is it necessary to assist in data analysis, but also in project planning, selection of an appropriate sequencing and analysis strategy and matching of intended data volume to their study aims. Within de.NBI, this deficiency has already been addressed by means of a metagenomics workshop offered by the BiGi service center[2] on a yearly basis; in the scope of this course, topics like sequence data quality control, 16S rRNA amplicon analysis, metagenome analysis employing MGX as well as metagenome assembly are covered. Also, consulting services are provided, and scientists are encouraged to contact the de.NBI service center in

[2]Bielefeld-Gießen Resource Center for Microbial Bioinformatics

the early planning phase of their studies to discuss the optimal strategies for their scientific questions.

Right from the start and long before its publication, MGX was successfully employed in various metagenome studies and has allowed biologists to benefit from modern means of metagenome data interpretation, which is also apparent from several published studies that relied on MGX. Also, these early bird users provided valuable feedback and suggestions for improvements that were subsequently integrated. The value of MGX to the scientific community is demonstrated by more than ten finished studies and an even larger number of ongoing projects. Currently, users of 69 different MGX projects have deposited datasets comprising more than 7.5 billion individual sequences in total, and at present the largest project contains more than 550 million sequences.

5.1.1 Completed studies

The following studies were successfully completed and relied upon MGX for the analysis of sequence data obtained from a variety of different habitats such as biogas fermenters, soil, or aqueous biotopes.

1. Barbara Klippel, Kerstin Sahm, Alexander Basner, Sigrid Wiebusch, Patrick John, Ute Lorenz, Anke Peters, Fumiyoshi Abe, Kyoma Takahashi, Olaf Kaiser, Alexander Goesmann, Sebastian Jaenicke, Ralf Grote, Koki Horikoshi, and Garabed Antranikian.

   Carbohydrate-active enzymes identified by metagenomic analysis of deep-sea sediment bacteria.

   Extremophiles, 18(5):853–863, 2014

2. Vímac Nolla-Ardèvol, Marc Strous, and Halina Elisabeth Tegetmeyer.

   Anaerobic digestion of the microalga Spirulina at extreme alkaline conditions: biogas production, metagenome, and metatranscriptome.

   Frontiers in Microbiology, 6:597, 2015

3. Vera Ortseifen, Yvonne Stolze, Irena Maus, Alexander Sczyrba, Andreas Bremges, Stefan P Albaum, Sebastian Jaenicke, Jochen Fracowiak, Alfred Pühler, and Andreas Schlüter.

   An integrated metagenome and-proteome analysis of the microbial community residing in a biogas production plant.

Journal of Biotechnology, 231:268–279, 2016

4. María C Martini, Daniel Wibberg, Mauricio Lozano, Gonzalo Torres Tejerizo, Francisco J Albicoro, Sebastian Jaenicke, Jan Dirk Van Elsas, Alejandro Petroni, M Pilar Garcillán-Barcia, Fernando De La Cruz, Andreas Schlüter, Alfred Pühler, Mariano Pistorio, Antonio Lagares, and Maria F Del Papa.

   Genomics of high molecular weight plasmids isolated from an on-farm biopurification system.

   Scientific Reports, 6:28284, 2016

5. Thanh Van Nguyen, Daniel Wibberg, Kai Battenberg, Jochen Blom, Brian Vanden Heuvel, Alison M Berry, Jörn Kalinowski, and Katharina Pawlowski.

   An assemblage of Frankia Cluster II strains from California contains the canonical nod genes and also the sulfotransferase gene nodH.

   BMC Genomics, 17(1):796, 2016

6. Irena Maus, Daniela E Koeck, Katharina G Cibis, Sarah Hahnke, Yong S Kim, Thomas Langer, Jana Kreubel, Marcel Erhard, Andreas Bremges, Sandra Off, Yvonne Stolze, Sebastian Jaenicke, Alexander Goesmann, Alexander Sczyrba, Paul Scherer, Helmut König, Wolfgang H Schwarz, Vladimir V Zverlov, Wolfgang Liebl, Alfred Pühler, Andreas Schlüter, and Michael Klocke.

   Unraveling the microbiome of a thermophilic biogas plant by metagenome and metatranscriptome analysis complemented by characterization of bacterial and archaeal isolates.

   Biotechnology for Biofuels, 9(1):171, 2016

7. Zala Schmautz, Andreas Graber, Sebastian Jaenicke, Alexander Goesmann, Ranka Junge, and Theo HM Smits.

   Microbial diversity in different compartments of an aquaponics system.

   Archives of microbiology, 199(4):613–620, 2017

8. Martina Lori, Sarah Symanczik, Paul Mäder, Norah Efosa, Sebastian Jaenicke, Franz Buegger, Simon Tresch, Alexander Goesmann, and Andreas Gattinger.

   Distinct nitrogen provisioning from organic amendments in soil as influenced by farming system and water regime.

Frontiers in Environmental Science, 6:40, 2018

9. Bhaskar Reddy, Jitendra Pandey, and Suresh Kumar Dubey.

   Assessment of environmental gene tags linked with carbohydrate metabolism and chemolithotrophy associated microbial community in River Ganga.

   Gene, 704:31–41, 2019

10. Johanna Nelkner, Christian Henke, Timo Wentong Lin, Wiebke Pätzold, Julia Hassa, Sebastian Jaenicke, Rita Grosch, Alfred Pühler, Alexander Sczyrba, and Andreas Schlüter.

    Effect of long-term farming practices on agricultural soil microbiome members represented by metagenomically assembled genomes (MAGs) and their predicted plant-beneficial genes.

    Genes, 10(6), 2019

11. Stefanie P Gläser, Meysam Taghinasab, Jafargholi Imani, Sebastian Jaenicke, Jens Steinbrenner, Gerald Moser, Peter Kämpfer, Christoph Müller, and Karl-Heinz Kogel.

    Phylogenetic diversity of fungal endophytes dominating the roots of two privileged plant in a permanent grassland exposed to elevated CO2 and increased surface temperature.

    Submitted.

12. Martina Lori, Gabin Piton, Sarah Symanczik, Nicolas Legay, Lijbert Brussaard, Sebastian Jaenicke, Eduardo Nascimento, Filipa Reis, Paulo Sousa, Paul Mäder, Andreas Gattinger, Jean-Christophe Cleément, and Arnaud Foulquier.

    The impact of ecological intensive management on proteolytic microbial communities is central for nitrogen-related ecosystem service provisioning under altered rain regimes: a cross country experiment.

    In preparation.

5.2 Outlook

While MGX already represents a complete solution for the analysis of unassembled metagenome sequence data, there are already several ideas how the framework

might be extended in the future. This section outlines some of these aspects, giving potential implementation hints and ideas where possible.

5.2.1 Preprocessing and quality control

So far, MGX features no component for the preprocessing of sequence datasets, as the usage of different sequencing technologies, protocols, multiplexing strategies and the variety of chemistry-specific adapter sequences for each platform has shown to be quite diverse. Instead, users are asked to inquire specifics of the employed protocols from their sequencing provider and perform appropriate dataset preparation. MGX does however provide the user with several reports that allow to deduce the necessity of additional preprocessing; nonetheless, these aspects are not fully addressed within the framework and it is currently up to the user to perform the required steps. For the future, it is thus recommended to establish appropriate infrastructure within MGX, which should not only perform quality-based trimming and removal of adapter sequences, but also incorporate additional advanced techniques such as the self-correction of sequence datasets obtained by third-generation sequencing platforms. Based on the deposited metadata, a preprocessing pipeline would be able to automatically determine adapters specific to a certain sequencing platform and perform *e.g.* deduplication selectively for technologies susceptible to artificial duplicates, which are commonly encountered within 454 pyrosequencing and IonTorrent datasets. Nonetheless, barcodes used for sample multiplexing would still need to be provided by the user.

5.2.2 Metagenome assembly

Within the last years, the metagenomics field has seen great advances and a lot of methodological improvement. Nowadays, metagenome assembly has begun to become a feasible approach, even though it remains to be a field that requires careful attention to detail (Ayling *et al.*, 2019). Especially third-generation sequencing technologies with their long obtainable read lengths have fueled the incentive for metagenome assembly, even though these types of data will typically need to be complemented with high-accuracy short-read sequences due to their high error rate; in combination, however, these long reads greatly ease the recovery of larger contiguous genome fractions from environmental sequence data.

With a platform like MGX, which hides a lot of the inherent complexity from the user, but still allows to control almost all aspects of the data analysis process (if desired), support for the assembly and subsequent annotation of metagenome datasets would constitute a worthwhile addition. Here, it would be especially

important to provide a rather conservative approach and educate users towards potential biases and artifacts in order to avoid data misinterpretation.

Support for metagenome assembly would also open the road to an adoption of additional methods established in *e.g.* comparative genomics, such as the determination of core, dispensable and pan-(meta)genomes of functioning microbial communities in their natural environments; so far, this direction has mostly been hindered by the lack of full-length environmental genes. Also, environmental communities represent a valuable resource for the discovery of novel natural compounds such as secondary metabolites with antimicrobial activity. As these compounds are often produced by enzymes coded within large gene clusters and their structure cannot reliably be determined without access to the complete sequence of the cluster, identification of such candidates is difficult if not impossible without prior assembly.

Recently, an approach termed co-assembly has become a popular method where several metagenome datasets from closely related habitats are subjected to joint assembly. This approach offers the advantage of increased community coverage provided that the samples are similar enough, and therefore allows to recover larger fractions of the individual genomes. For MGX, it would hence be desirable to implement appropriate analysis workflows for metagenome assembly followed by subsequent taxonomic binning and annotation. While some of these aspects remain specific to metagenomics, a lot of the work has already been done, for example in the genomics field, which offers established tools for high-quality automated annotation of assembled sequences. In order to facilitate metagenome co-assembly, a modification of the underlying job model would be required, as it is currently limited to the processing of individual datasets. Within the user interface, new modules need to be implemented that enable scientists to assess overall assembly quality metrics, inspect taxonomic bins for consistency and completeness, and query annotation results for genes of interest. Some of these aspects such as the reference genome browser from the Mapping Viewer module, however, are readily implemented within MGX and might be reused for the visualization of assembled metagenomic contigs.

5.2.3 Selection of data processing engine

At the time the MGX framework's components were designed, the Conveyor workflow engine was chosen as the sole sequence data processing means based on several criteria: (i) its advanced type system, (ii) transparent distribution of analysis tasks to a DRMAA-based compute cluster, (iii) easy inclusion of new metagenomics processing tools and (iv) availability of a graphical interface enabling the creation of novel workflows without programming knowledge as a prerequisite. However, several shortcomings were also identified, most prominently a lack of resilience. While

Conveyor is able to automatically reschedule failed tasks to a certain extent, several possible causes for job abortion remain unaddressed, resulting in cancellation of workflow processing. As Conveyor does not support checkpointing, this scenario requires re-running the complete analysis workflow *ab initio*, thus computationally challenging steps will possibly have to be recomputed even if just a rather simple processing step produced an error.

Also, Conveyor did not attract much attention, resulting in a lack of community adoption and external contributions of new functionality. While Conveyor still remains in use and Conveyor-based workflows are actively used *e.g.* within the EDGAR platform for comparative genomics (Blom *et al.*, 2009, 2016) or for the initial processing steps of sequence data generated at the CeBiTec sequencing center, it has failed to gain momentum and attract external attention. Currently, the Conveyor workflow engine is mostly unknown to the outside world and should probably be considered unmaintained, as its sole creator is no longer actively developing it.

In the meanwhile, several potential alternatives have been published, *e.g.* the Nextflow DSL[3] (Di Tommaso *et al.*, 2017) for computational data analysis or Snakemake (Köster and Rahmann, 2012), a Python-based workflow engine with a rule-based syntax closely resembling that of the popular Unix make command. Another alternative is proposed by several community efforts such as Bioboxes (Belmann *et al.*, 2015) or Biocontainers (da Veiga Leprevost *et al.*, 2017), which both aim at the containerization of bioinformatics tools using Docker-based containers. A lack of standardized data interchange formats, however, has hindered widespread adoption of these initiatives so far. The Common Workflow Language Specification (CWL; Amstutz *et al.*, 2016) promotes an abstract means to describe analysis workflows that is agnostic and thus independent of a specific implementation. So far, several established workflow engines have started to adopt the CWL specification, among them Galaxy and Taverna (Oinn *et al.*, 2004). Apart from cwltool, the CWL reference implementation, workflow engines like Arvados[4] or Cromwell[5] support the execution of CWL-based workflows on compute clusters as well as cloud-based infrastructures.

Cloud computing has seen a large rise especially in bioinformatics, and the European ELIXIR organization[6] and its German node, the de.NBI network[7], provide extensive cloud computing resources to the scientific community free of charge. For future developments of the MGX framework, it is therefore desirable to make use of these resources and to extend the analysis capabilities by either performing a migration of the currently provided analysis pipelines to a different data processing

[3] Domain-Specific Language
[4] https://arvados.org/
[5] https://github.com/broadinstitute/cromwell
[6] http://elixir-europe.org
[7] http://www.denbi.de

engine, implementing cloud support for the Conveyor workflow engine or adding support for an additional workflow system to MGX that is able to deploy tasks to cloud-based resources.

In its current form, distribution of MGX workflows to cloud-based resources is prohibited as the execution of Conveyor-based pipelines requires direct access to the PostgreSQL-based project database as well as sequence storage in order to retrieve metagenome sequence data and to create MGX database entities for the representation of analysis results. To enable the use of remote compute resources, it is therefore necessary to provision the means that allow automated data retrieval and annotation of results in a secure manner. As far as sequence data access is concerned, this can be achieved via the existing MGX dispatcher infrastructure, which would need to deposit the sequence data in cloud storage such as Amazon S3 prior to scheduling the actual workflow; also, S3 access credentials would need to be forwarded to the corresponding workflow execution mechanism. As an alternative that avoids data duplication, streaming of sequences from the MGX dispatcher to the workflow executor might be employed. For the creation of analysis results, namely attributes, attribute types and observations, a REST interface is already in place, which was implemented for the import of MGS-based reference annotations (Section 3.6.10). However, this interface is currently only exposed to authenticated users; for automated processing, a modification of the job model would be necessary: Assuming the MGX server would assign a dedicated and secret API key to each job, this generated token could then be used as a substitute for the access credentials provided by a human user.

The next required part would consist of the packaging of bioinformatics tools and databases used by MGX in either a VM image or a supported container format such as Docker or Singularity (Kurtzer *et al.*, 2017). While this does not pose a problem for the bioinformatics tools since MGX already relies on automated software builds, the sheer size of the employed databases prohibits packaging them in a container format. S3-based storage is also unsuitable for these types of data, as it performs badly for random-access operations, which are frequently encountered for indexed databases. Hence, a different approach would need to be implemented for the storage and distribution of sequence databases *e.g.* to the different de.NBI cloud locations. Here, either an implementation-specific approach such as Docker volumes might be employed, or each cloud provider would be required to transparently provide access to a shared storage system containing the needed databases.

For a solution based on CWL, all bioinformatics tools currently used within MGX would need to be integrated and corresponding CWL descriptions defined; as CWL, and its implementations such as Nextflow, Snakemake etc. lack a proper type system and all of them use files as the means to transport data between nodes, this part necessitates the definition of common data interchange formats between nodes, which would require quite some effort.

From a mere technical perspective, it would therefore be preferable to obtain novel funding for the maintenance and extension of the Conveyor workflow engine. While currently not "cloud-aware", the (compute-intensive) parts that are at the moment distributed to DRMAA-based infrastructure could with overseeable effort be modified to employ *e.g.* Kubernetes as a platform for data analysis. Kubernetes already offers a job specification to run batch workloads; hence, it would only be necessary to implement a binding to the Kubernetes API within Conveyor and convert the parts that currently rely on the availability of a shared network filesystem to S3 storage.

5.2.4 Containerization of MGX components

In order to ease the adoption of MGX in institutions which already own a sufficiently sized IT environment or for the deployment of private MGX servers in cloud environments, it is also desirable to provide prebuilt container images of the different MGX server components (server, dispatcher, database). This would facilitate the application of MGX for users who either do not wish to establish a private project on one of the existing MGX servers, or in corporate environments where data protection standards might not permit distribution of data to outside locations. As analysis results within MGX can be compared across server boundaries, this would allow to easily operate a private MGX instance but still enable users to compare findings obtained for own datasets to taxonomic or functional profiles of metagenomes deposited on other servers. Apart from containerization itself, this would also include the provisioning of orchestration templates for different types of cloud frameworks such as OpenStack HEAT[8] or Kubernetes-based deployments.

5.2.5 Standardized functional analysis

The advantages of MGX for the design of novel analysis workflows have already been outlined previously (Chapter 4). To complement the newly developed pipeline for taxonomic classification, future work on MGX should also strive to implement a default pipeline for the functional assignment of metagenome sequences; this pipeline should cover at least the most prominent approaches for functional categorization and therefore include analysis steps assigning COG functional categories, COG groups, Pfam domains and EC numbers. As metadata is available within MGX, some parts of such a modular pipeline could also be enabled conditionally, for example an annotation step employing the MarDB database (Klemetsen *et al.*, 2017) which contains marine microbial genomes, which would be executed only for datasets derived from water-borne habitats.

[8] `https://docs.openstack.org/heat/`

With standardized workflows addressing both taxonomic as well as functional assignment tasks, a dedicated reporting component could then be implemented within the MGX user interface, which would provide adequate assignment metrics, a predefined set of default charts and statistics to the user. Such an offering would be particularly relevant to studies that do not follow a distinct hypothesis but instead aim at deciphering the general taxonomic composition and genetic repertoire occurring within a certain community.

5.2.6 Public metagenome resource

A standardized operating procedure for both taxonomic as well as functional classification would also open up the possibility to establish the automated retrieval and processing of metagenomes deposited in the short read archive (SRA) collections operated by both the NCBI and EBI. To a certain extent similar to the approach currently chosen by EBI MGnify, such an effort could then be offered as a public database to the scientific community. For these sheer amounts of data, however, it will no longer be possible to retain all the information within a project's SQL database. There are two viable options to address this issue:

Firstly, a new MGX project subtype could be devised for static projects; this subtype would not offer single-sequence resolution and only retain aggregated analysis results. Herewith, a significant reduction of the project database could be achieved, as neither sequences nor the observation table would need to be stored permanently. However, this approach would also prohibit the execution of additional analysis workflows and require a recomputation of the project, or the extraction of data subsets, and such projects would primarily serve as a precomputed resource of public metagenomes. Even with the slightly reduced subset of available features, interoperability with "full scale" MGX projects would still be retained as far as comparative evaluation of results and statistics are concerned.

Another approach would outsource the read and observation tables to a different storage type. Columnar storage formats such as *e.g.* Parquet[9] or ORC[10] offer efficient access and could be evaluated for this purpose. However, with the distribution of these data to several storage backends, relational integrity could no longer be ensured by the RDBMS itself. Also, some algebraic operations such as joins that are currently performed by the SQL database engine would require a different approach.

[9] `https://parquet.apache.org`
[10] `https://orc.apache.org`

5.2.7 Conclusion

Even though MGX represents a solution addressing all major aspects for the processing of unassembled metagenome data, these possible extension points would further extend the portfolio of offered analysis types and therefore additionally improve the value of the MGX framework to the scientific community. Especially metagenome assembly has gained popularity due to the availability of third-generation sequencing techniques. These are able to provide long contiguous sequences, therefore greatly reducing the risk of chimeric assemblies. Also, the capabilities to process very large metagenomes would significantly benefit from the inclusion of cloud-based compute resources into MGX. Such datasets are nowadays produced by large initiatives focussing on *e.g.* marine environments, soil, or the Human Microbiome Project (HMP), and while MGX is already capable of handling such data volumes, additional compute resources would further reduce the time required for taxonomic or functional profiling. These resources are already offered free of charge by initiatives like ELIXIR or de.NBI, just waiting to be put to use. Nevertheless, even now MGX already represents a highly flexible metagenome processing solution and, with its large variety of supported analysis, has the potential to significantly ease the processing and interpretation of such data in practice.

APPENDIX A

Implementation of custom analysis pipelines

Analysis tools provided by the MGX platform are implemented as workflows for the Conveyor (Linke *et al.*, 2011) workflow engine. Within Conveyor, tools are provided as so-called "nodes", which resemble individual processing steps and which are used to implement novel analysis methods by simply arranging and connecting them into a larger workflow. Conveyor includes plugins providing typical bioinformatics tools like BLAST or HMMER, but within the scope of this thesis dedicated plugins have been developed that are aimed at metagenome analysis, like MetaCV, Kraken, or MetaPhlAn, which all perform taxonomic classification.

A dedicated Conveyor plugin provides access to MGX data structures, thereby enabling the analysis of metagenomes stored in the MGX system with processing tools provided by Conveyor itself. While workflow definitions are stored in an XML-based format, a graphical user interface, the Conveyor Designer (Figure A.1), enables users to implement new analysis workflows in an intuitive manner by simply placing and connecting nodes.

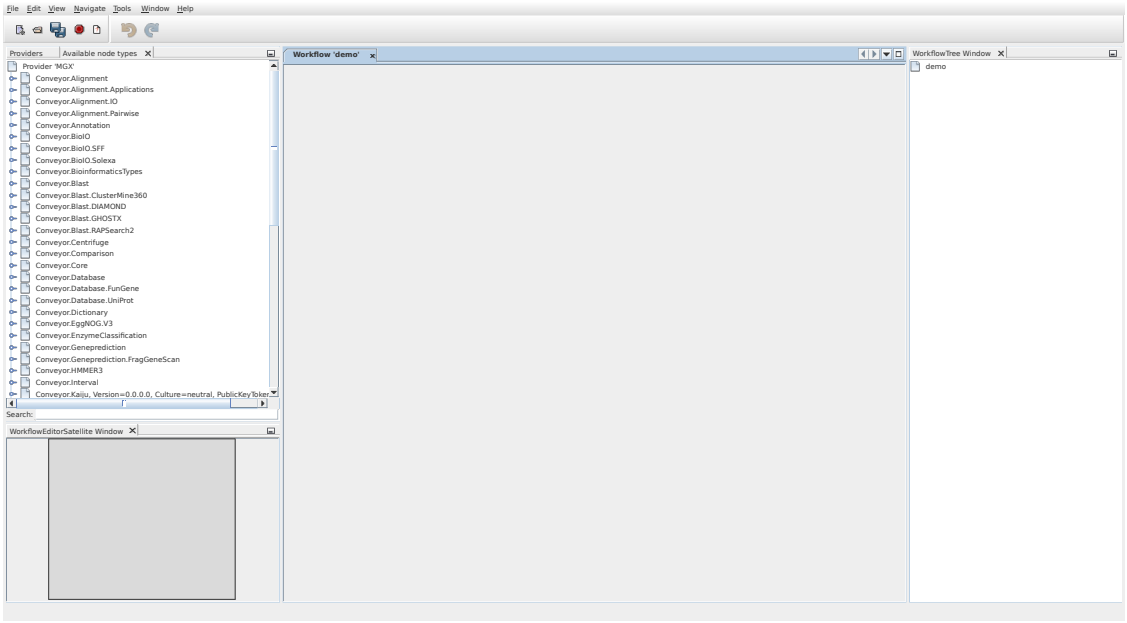


Figure A.1: Conveyor Designer The Conveyor Designer application allows easy and user-friendly development of custom analysis algorithms in a graphical way.

Giving a thorough introduction to Conveyor is beyond the scope of this document; the documentation describing Conveyor itself and the Conveyor Designer in particular can be found at the Conveyor web site[1].

A.1 Setting up the Conveyor workflow Designer

In order to implement a custom workflow, the Conveyor Designer needs to be configured with a definition of available Conveyor plugins and node types. This is achieved by importing a plugin dump file, which contains a list of data types and nodes provided by a Conveyor installation.

To use the Designer to implement a workflow for the MGX framework, a corresponding plugin dump file can be obtained from within MGX via the context menu of each project (Figure A.2).

[1] `http://www.uni-giessen.de/fbz/fb08/Inst/bioinformatik/software/Conveyor`

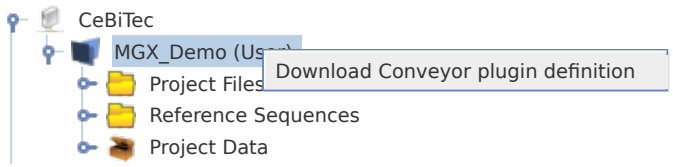


Figure A.2: Obtaining a plugin dump file. A plugin dump file for use with the Conveyor Designer can be obtained from within MGX by right-clicking on the project name.

Afterwards, a new provider needs to be defined in the Designer application (via right-click on "Available providers"). After specifying "Plugin dump file" (Figure A.3) as the type of plugin set, the file generated by MGX can be imported; once the plugin dump file has been successfully imported, new workflows can be implemented. Initially starting with an empty sheet, nodes can be dragged from the list of all available nodes offered by a provider (on the left) and placed onto the sheet. Node connections are created by clicking on a node, keeping the mouse button pressed and releasing it over the desired target node, thus creating the link; in ambiguous cases, *e.g.* for nodes with several unconnected inputs/outputs, a dialog allows to select the desired connection endpoint. Nodes may also require node-specific configuration, which can be edited from the nodes' context menu. A red border around a node indicates missing configuration items or connections.

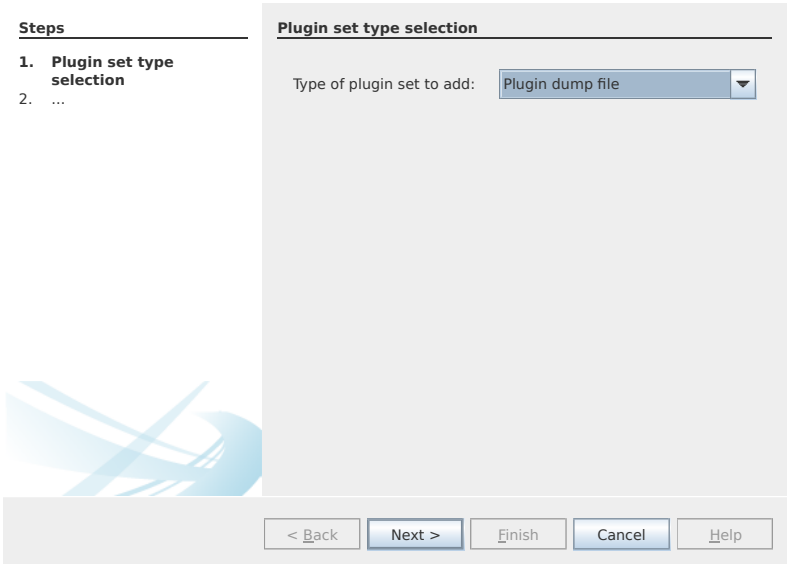


Figure A.3: Adding a new provider. Importing a plugin dump file into the Conveyor Designer.

A.2 Basic workflow requirements

In order to design custom Conveyor workflows for later usage within the MGX platform, there are several constraints to be met which will be described in more detail. First of all, a dedicated GetMGXJob node (Figure A.4) has to be present as part of the workflow; in addition, this node has to be assigned the name "**mgx**" from within the node configuration dialog. During execution of a pipeline within MGX, this node is configured via an external configuration file, providing required information about a job's context, like *e.g.* access to the project-specific database and associated storage systems.

Figure A.4: GetMGXJob. The GetMGXJob node provides necessary context for executing a workflow within MGX, such as database access. By convention, this node has to be named "**mgx**".

Access to metagenome DNA sequences is provided via the GetSequences node, which will provide all metagenome sequences for a sequencing run stored within MGX, filtering out those for which the "discard" flag has already been set. As pipelines are always executed for one single analysis job, this node needs to be connected to the GetMGXJob node (Figure A.5). Figure A.6 shows a minimal working example of a Conveyor-based pipeline for use within MGX. Once executed, the pipeline would unconditionally set the "discard" flag for all sequences.

Figure A.5: GetSequences. The GetSequences node is used to obtain metagenome sequence data from within MGX; it has one input that needs to be connected to the GetMGXJob node.

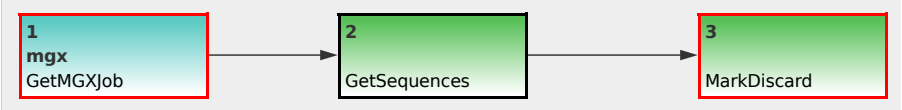


Figure A.6: Minimal example. A minimal working example of a pipeline developed for MGX, which unconditionally sets the "discard" flag for all sequences.

A.3 Annotating metagenome sequences

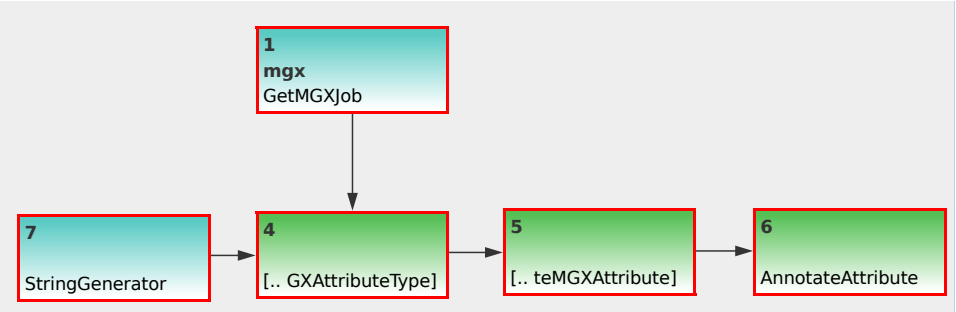


Figure A.7: Metagenome annotation. Basic workflow template to illustrate sequence annotation. A StringGenerator is used to generate a label for the attribute type (CreateMGXAttributeType), which also requires job context information. The attribute type is required to create attributes, thus the node is connected to the CreateMGXAttribute node. Finally, the annotation data can be persisted to the project database (AnnotateAttribute node).

A.3.1 Performing basic sequence annotation

Annotation of metagenome sequences requires an "attribute type" and an "attribute". As an example, the step-wise implementation of a pipeline for the annotation of GC content within metagenome sequences is demonstrated. A StringGenerator node configured to generate the string "GC" is used to create a label for the necessary attribute type. As GC content is indicated by a number, the CreateMGXAttributeType node is appropriately configured to emit a **basic** (*i.e.* not hierarchical) as well as **numerical** "attribute type" (Figure A.8).

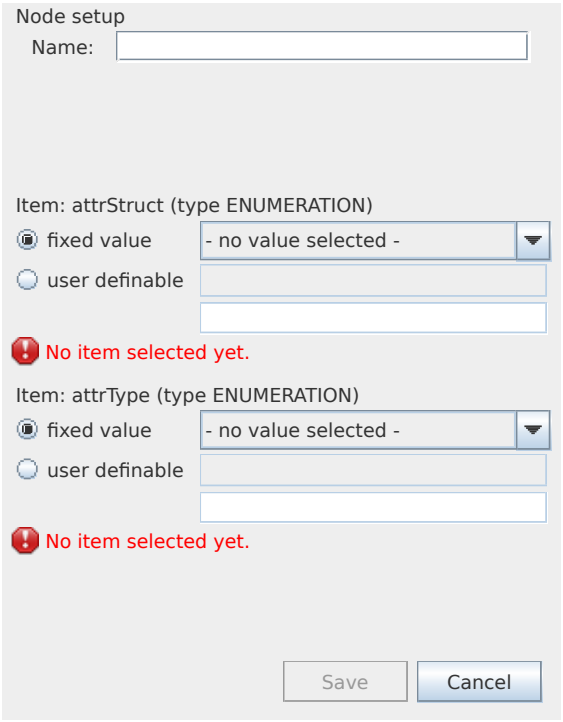


Figure A.8: Defining an attribute type. Within the configuration dialog for the CreateMGXAttributeType node, structure and type of the generated attribute values are defined.

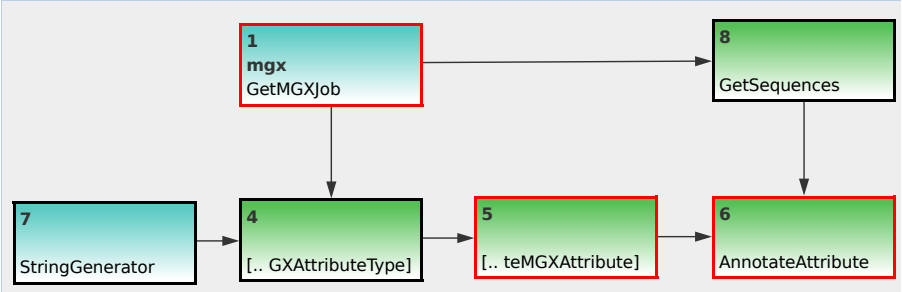


Figure A.9: Metagenome annotation. Incomplete example; extending upon Figure A.7, the GetSequences node will provide the necessary metagenome sequences to be annotated. Still, there is no actual analysis specified.

In a second step, the GetSequences node is used to obtain access to the individual metagenome sequences; as MGX annotates sequences individually, a connection between the GetSequences and AnnotateAttribute nodes is required (Figure A.9). Subsequently, the actual analysis is implemented, which is provided by the GCContent node. It will process all sequences and emit the corresponding GC content value for each of them. To convert these values into appropriate "attributes", an

"attribute type" is required for each value; therefore, a Repeat node is inserted between nodes 4 and 5 (Figure A.10).

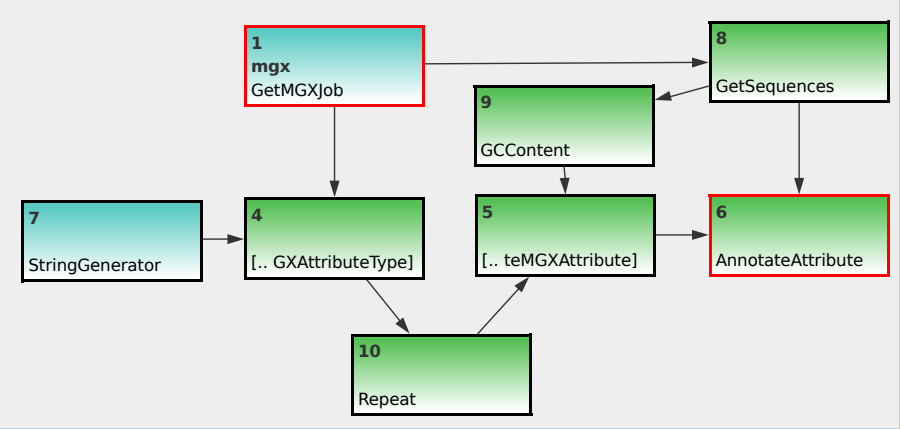


Figure A.10: Metagenome annotation. Step 2: The GCContent node represents the actual analysis step; it is used to determine the GC content of a DNA sequence, which will then be converted into an "attribute". Since an "attribute type" is required for each "attribute", a Repeat node is inserted between nodes 4 and 5.

Finally, as an annotation always refers to only a part of a sequence, the corresponding start and end coordinates need to be provided; since GC content is a property referring to the full sequence, an ULongGenerator node configured to emit 0 (MGX uses 0-based coordinates) is used to generate the start coordinate; this node needs to be connected to a Repeat node to generate a series of 0s.

The end coordinate can be created based on the sequence length, with 1 subtracted, obtained through the GetLength and MinusOne nodes (Figure A.11).

While the workflow is now fully implemented, the GetMGXJob node still retains its red border due to missing configuration; this, however, can be ignored, as appropriate configuration will be provided by the MGX framework at runtime.

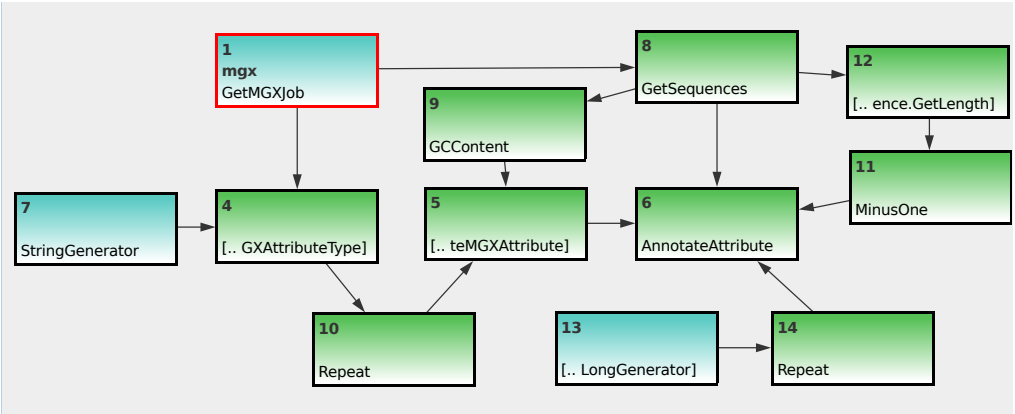


Figure A.11: Metagenome annotation. Completing the workflow: the ULongGenerator and GetLength nodes are added to specify coordinates for the subregion of the DNA sequence described by the "attribute"; the start coordinate is simply repeated, while 1 is subtracted from the sequence length due to 0-based coordinates.

A.3.2 Annotation of hierarchical attributes

Annotation of hierarchical attributes requires a little more effort. The CreateHierarchicalMGXAttribute node is used to obtain the inner structure of the hierarchy in a bottom-up approach; it contains several loops which will be explained in more detail.

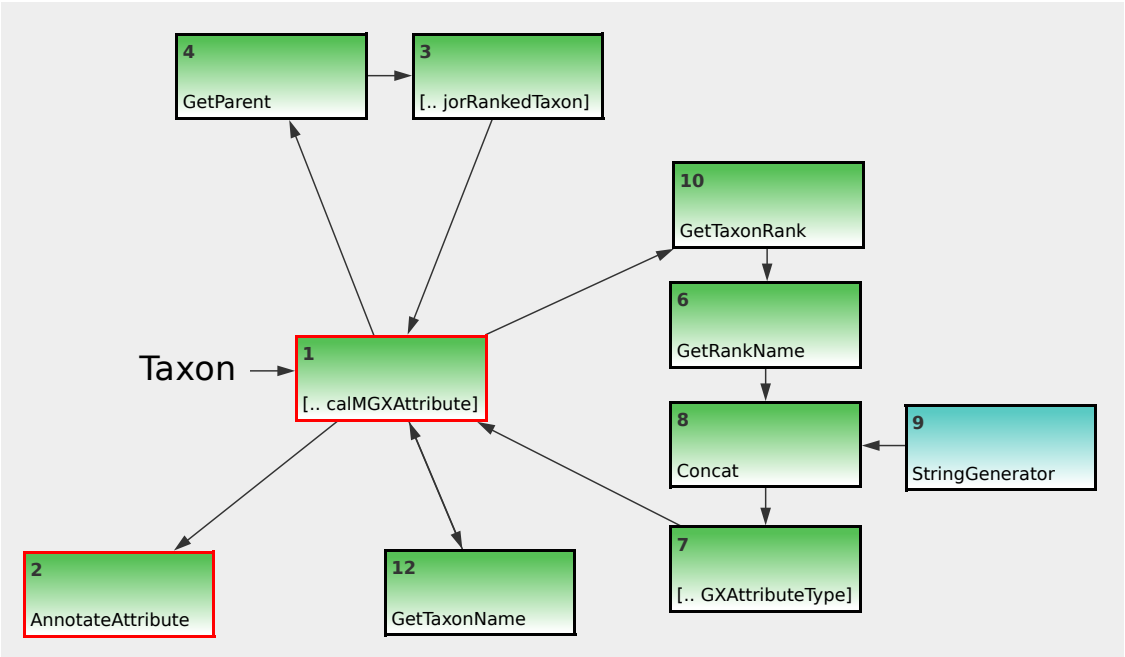


Figure A.12: Hierarchical attributes. The CreateHierarchicalMGXAttribute node requires three loops (note double-ended arrow on third loop between nodes 1 and 12) to create the internal structure of the hierarchy. Several connections were removed from the figure for illustrative purposes.

A single object, *e.g.* a NCBI taxon generated by the Kraken (Wood and Salzberg, 2014) classifier, is provided as the initial input into the node (Figure A.12). The first loop is required to obtain the objects' parent object, thus defining the hierarchy. In this example it is implemented using the GetParent and GetMajorRankedTaxon nodes, making sure only the major taxonomic ranks (superkingdom, phylum, class, order, family, genus, species) are included.

The second loop is used to obtain the corresponding attribute type for an object: it operates on the initial taxon as well as its parents obtained by the first loop. The GetTaxonRank and GetRankName nodes provide the corresponding rank's name, *e.g.* "class"; the StringGenerator and Concat nodes are then used to create the attribute type: "NCBI_class". This value is used to create the corresponding attribute type employing the CreateMGXAttributeType node, which is returned into the CreateHierarchicalMGXAttribute node.

The third and final loop is used to map a data object to its name, which is used to create the attribute's value; it is built up using the GetTaxonName node, which delivers its output back into the node.

Thus, the three loops might also be termed as **Get parent**, **Get AttributeType for object** and **Generate value**.

The CreateHierarchicalMGXAttribute node emits a hierarchical MGXAttribute for the initial data object, with the corresponding AttributeType provided by loop 2 and the MGXAttribute's value obtained using loop 3. Internally, loop 1 is used repetitively until the root node is reached, with all intermediary results passing through loops 2 and 3, thus generating a single path of hierarchical attributes within the taxonomic tree. The output of the CreateHierarchicalMGXAttribute is connected to the AnnotateAttribute node as already shown in the previous example.

For brevity's sake, several connections are hidden within the image, which have already been explained in the previous section: the CreateMGXAttributeType node needs an incoming connection providing a MGXJob, and the AnnotateAttribute node requires additional connections providing the sequence to be annotated and start/stop coordinates for the subregion which is described by the annotation.

A.4 Workflow import into MGX

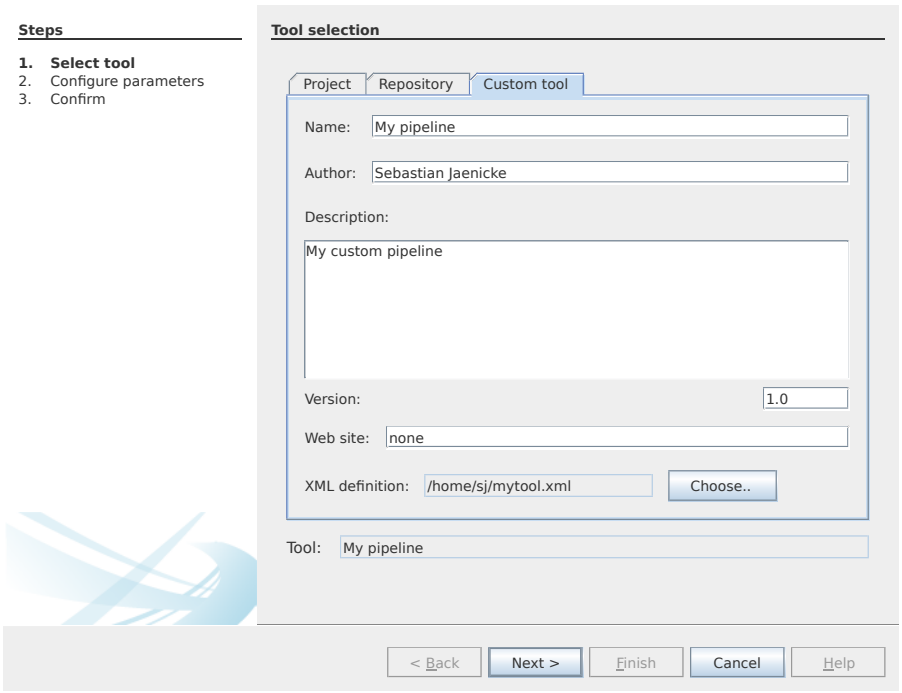


Figure A.13: Import a new workflow. The finished workflow is imported into an MGX project via the analysis wizard, which allows to upload the pipeline definition and assign a reasonable name.

Once the workflow has been fully implemented, it can be saved to a workflow definition file in an XML-based format. Within MGX, the execution of analysis jobs is

controlled via a comfortable wizard that allows selection as well as parameterization of the desired workflow. The wizard provides access to the server-specific repository of predefined workflows, but also allows to import and execute the Conveyor workflow (Figure A.13).

APPENDIX B

MGX database model

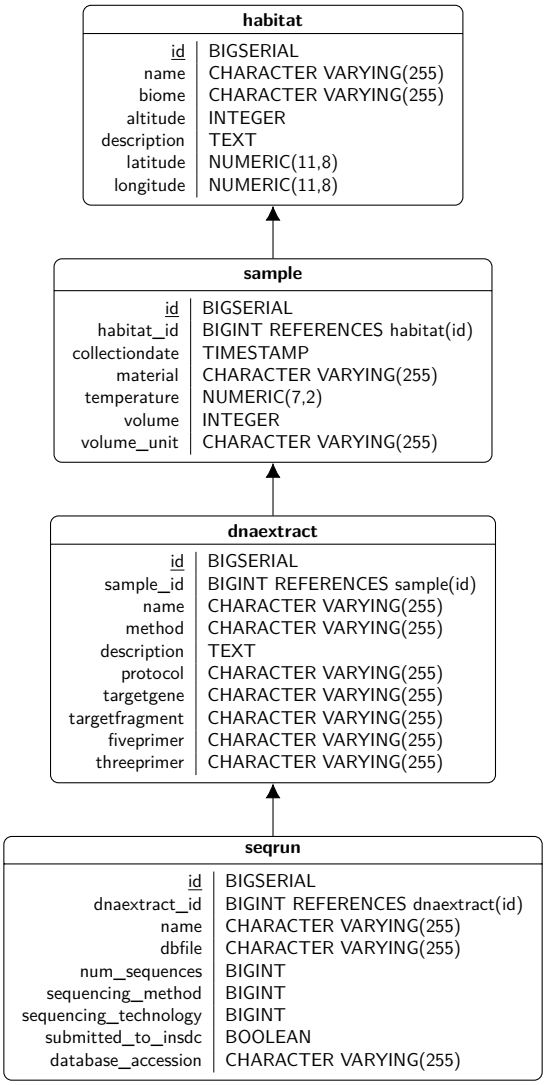


Figure B.1: MGX metadata model. MGX collects metadata detailing the origin of a dataset together with additional information describing the sampling process, DNA extraction procedures, as well as the sequencing approach that was employed. Some constraints such as uniqueness or nullability of fields have been omitted for brevity.

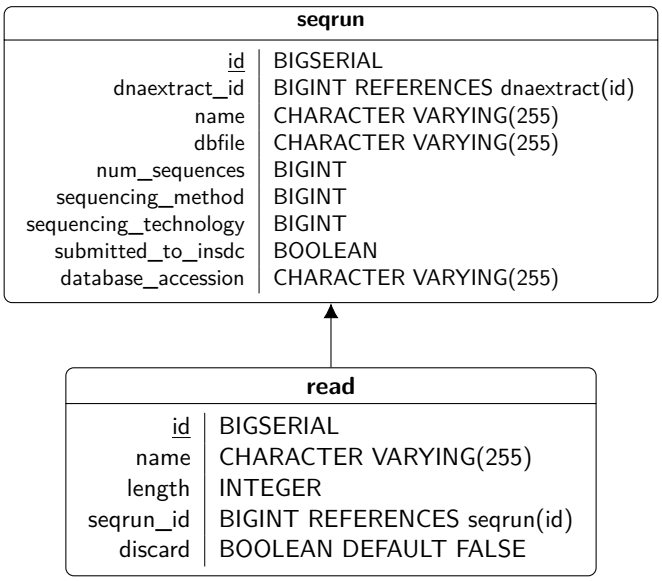


Figure B.2: MGX sequence storage. Only sequence names are stored to the relational project database; the actual nucleotide data for all reads belonging to a sequencing run is contained in a separate indexed storage file (dbfile field of the seqrun table).

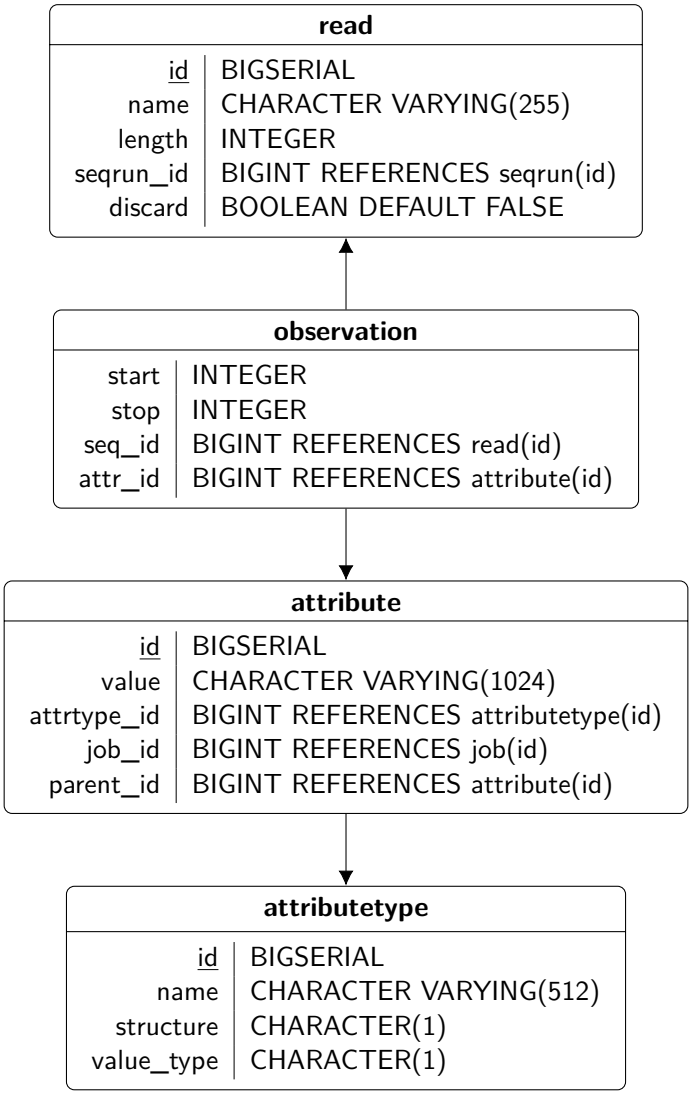


Figure B.3: MGX result model. An observation creates an association between a metagenome sequence and an attribute, while the attributetype referenced by each attribute further indicates the properties of the attribute such as its structure or value type. As the observation also includes the relevant base positions within the read (start and stop fields), subsequence resolution is achieved.

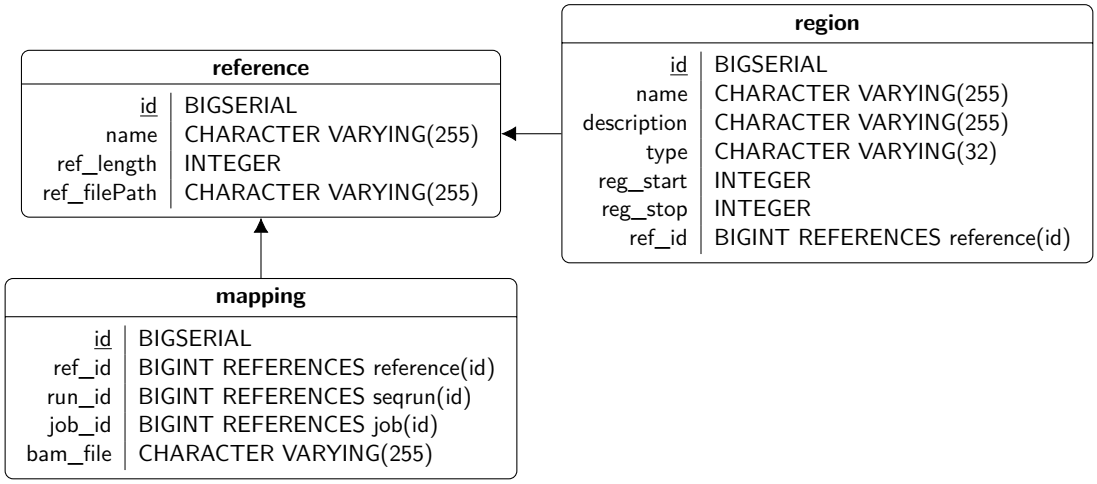


Figure B.4: MGX database model for reference genome alignment. While a mapping entity is created in the database for each reference mapping job, the corresponding data resides within an external file in BAM format.

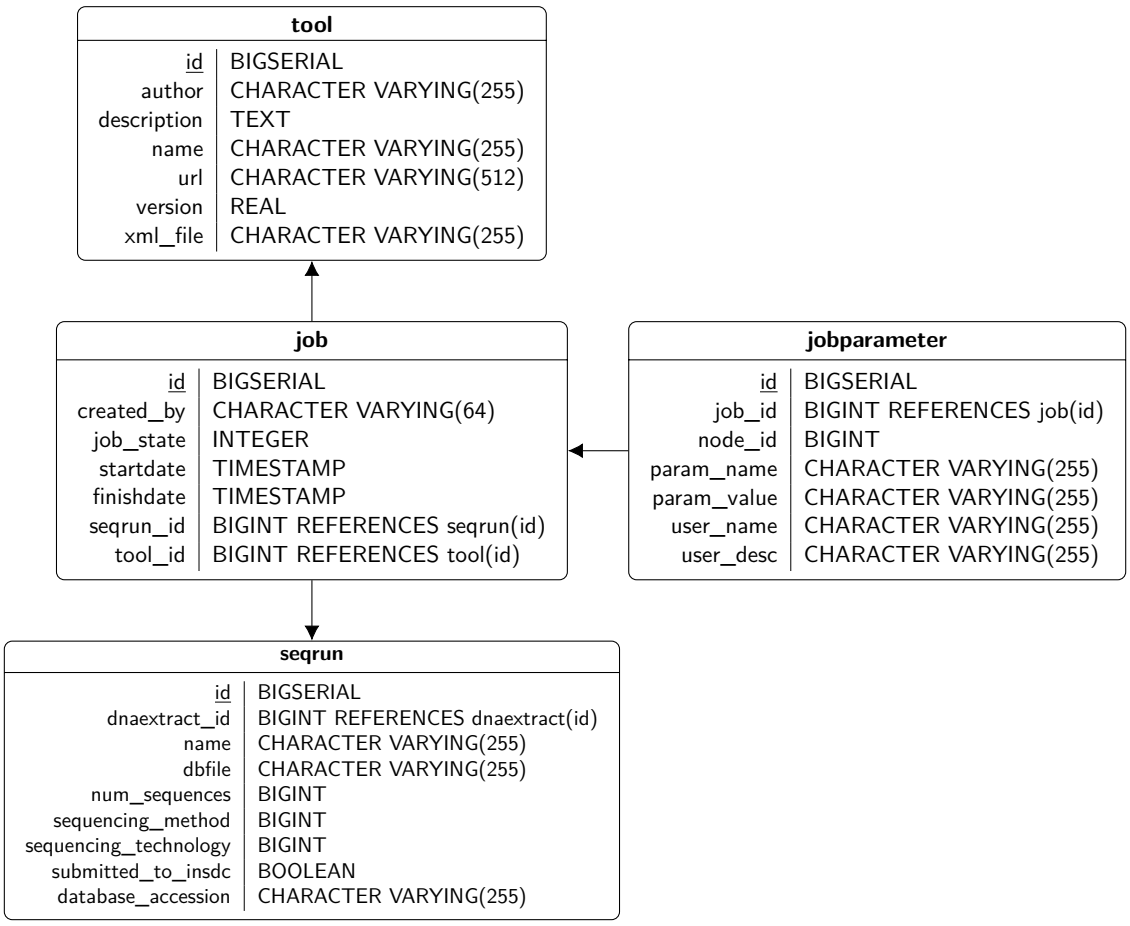


Figure B.5: MGX job model. Analysis jobs are executed for each sequencing run separately; the corresponding job parameters for the selected workflow are stored within the project database.

Bibliography

John Aitchison. The statistical analysis of compositional data. *Monographs on Statistics and Applied Probability. London: Chapman and Hall Ltd.*, 1986.

Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3): 403–410, 1990.

Peter Amstutz, Michael R Crusoe, Nebojša Tijanić, Brad Chapman, John Chilton, Michael Heuer, Andrey Kartashov, Dan Leehr, Hervé Ménager, Maya Nedeljkovich, *et al.* Common Workflow Language, v1.0. Specification, Common Workflow Language working group, 2016.

Simon Andrews *et al.* FastQC: a quality control tool for high throughput sequence data. 2010. URL `https://www.bioinformatics.babraham.ac.uk/projects/fastqc/`.

J Anne and R Ann. *The new science of metagenomics: revealing the secrets of our microbial planet*. National Academies Press, 2007.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, *et al.* Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

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

Martin Ayling, Matthew D Clark, and Richard M Leggett. New approaches for metagenome assembly with short reads. *Briefings in Bioinformatics*, 2019.

Fredrik Bäckhed, Josefine Roswall, Yangqing Peng, Qiang Feng, Huijue Jia, Petia Kovatcheva-Datchary, Yin Li, Yan Xia, Hailiang Xie, Huanzi Zhong, *et al.* Dynamics and stabilization of the human gut microbiome during the first year of life. *Cell host & microbe*, 17(5):690–703, 2015.

Alex Bateman, Lachlan Coin, Richard Durbin, Robert D Finn, Volker Hollich, Sam Griffiths-Jones, Ajay Khanna, Mhairi Marshall, Simon Moxon, Erik LL Sonnhammer, *et al.* The Pfam protein families database. *Nucleic Acids Research*, 32(suppl 1):D138–D141, 2004.

Peter Belmann, Johannes Dröge, Andreas Bremges, Alice C McHardy, Alexander Sczyrba, and Michael D Barton. Bioboxes: standardised containers for interchangeable bioinformatics software. *GigaScience*, 4(1):47, 2015.

John Besemer and Mark Borodovsky. GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses. *Nucleic Acids Research*, 33(suppl_2): W451–W454, 2005.

J Martin Bland and Douglas G Altman. Statistical methods for assessing agreement between two methods of clinical measurement. *The Lancet*, 327(8476):307–310, 1986.

Frederick R Blattner, Guy Plunkett, Craig A Bloch, Nicole T Perna, Valerie Burland, Monica Riley, Julio Collado-Vides, Jeremy D Glasner, Christopher K Rode, George F Mayhew, *et al.* The complete genome sequence of Escherichia coli K-12. *Science*, 277(5331):1453–1462, 1997.

Jochen Blom, Stefan P Albaum, Daniel Doppmeier, Alfred Pühler, Frank-Jörg Vorhölter, Martha Zakrzewski, and Alexander Goesmann. EDGAR: a software framework for the comparative analysis of prokaryotic genomes. *BMC Bioinformatics*, 10(1):154, 2009.

Jochen Blom, Julian Kreis, Sebastian Spänig, Tobias Juhre, Claire Bertelli, Corinna Ernst, and Alexander Goesmann. EDGAR 2.0: an enhanced software platform for comparative gene content analyses. *Nucleic Acids Research*, 44(W1):W22–W28, 2016.

Sébastien Boisvert, Frédéric Raymond, Élénie Godzaridis, François Laviolette, and Jacques Corbeil. Ray Meta: scalable de novo metagenome assembly and profiling. *Genome Biology*, 13(12):R122, 2012.

Anthony M Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 2014.

Fredrik Boulund, Anna Johnning, Mariana Buongermino Pereira, DG Joakim Larsson, and Erik Kristiansson. A novel method to discover fluoroquinolone antibiotic resistance (qnr) genes in fragmented nucleotide sequences. *BMC Genomics*, 13 (1):695, 2012.

Mya Breitbart, Peter Salamon, Bjarne Andresen, Joseph M Mahaffy, Anca M Segall, David Mead, Farooq Azam, and Forest Rohwer. Genomic analysis of uncultured marine viral communities. *Proceedings of the National Academy of Sciences*, 99 (22):14250–14255, 2002.

Florian P Breitwieser and Steven L Salzberg. KrakenHLL: Confident and fast metagenomics classification using unique k-mer counts. *bioRxiv*, page 262956, 2018.

Karel Břinda, Maciej Sykulski, and Gregory Kucherov. Spaced seeds improve k-mer-based metagenomic classification. *Bioinformatics*, 31(22):3584–3592, 2015.

Bonnie L Brown, Mick Watson, Samuel S Minot, Maria C Rivera, and Rima B Franklin. MinION™ nanopore sequencing of environmental metagenomes: a synthetic approach. *GigaScience*, 6(3):1–10, 2017.

Benjamin Buchfink, Chao Xie, and Daniel H Huson. Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, 2014.

Carol J Bult, Owen White, Gary J Olsen, Lixin Zhou, Robert D Fleischmann, Granger G Sutton, Judith A Blake, Lisa M FitzGerald, Rebecca A Clayton, Jeannine D Gocayne, *et al.* Complete genome sequence of the methanogenic archaeon, Methanococcus jannaschii. *Science*, 273(5278):1058–1073, 1996.

Michael Burrows and David J Wheeler. A block-sorting lossless data compression algorithm. In *SRC Research Report 124*. Digital, 1994.

Benjamin J Callahan, Paul J McMurdie, Michael J Rosen, Andrew W Han, Amy Jo A Johnson, and Susan P Holmes. DADA2: high-resolution sample inference from Illumina amplicon data. *Nature Methods*, 13(7):581, 2016.

Benjamin J Callahan, Paul J McMurdie, and Susan P Holmes. Exact sequence variants should replace operational taxonomic units in marker gene data analysis. *The ISME Journal*, page 113597, 2017.

M Luz Calle. Statistical analysis of metagenomics data. *Genomics Inform*, 17(1): e6–, 2019.

Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. BLAST+: architecture and applications. *BMC Bioinformatics*, 10(1):1, 2009.

Bruno Canard and Robert S Sarfati. DNA polymerase fluorescent substrates with reversible 3′-tags. *Gene*, 148(1):1–6, 1994.

J Gregory Caporaso, Justin Kuczynski, Jesse Stombaugh, Kyle Bittinger, Frederic D Bushman, Elizabeth K Costello, Noah Fierer, Antonio Gonzalez Peña, Julia K Goodrich, Jeffrey I Gordon, *et al.* QIIME allows analysis of high-throughput community sequencing data. *Nature Methods*, 7(5):335–336, 2010.

Albi Celaj, Janet Markle, Jayne Danska, and John Parkinson. Comparison of assembly algorithms for improving rate of metatranscriptomic functional annotation. *Microbiome*, 2(1):39, 2014.

Anne Chao. Nonparametric estimation of the number of classes in a population. *Scandinavian Journal of statistics*, 11:265–270, 1984.

Anne Chao and Chun-Huo Chiu. Estimation of species richness and shared species richness. *Methods and Applications of Statistics in the Atmospheric and Earth Sciences*, pages 76–111, 2012.

Anne Chao and Chun-Huo Chiu. Species Richness: Estimation and Comparison. *Wiley StatsRef: Statistics Reference Online*, pages 1–26, 2014.

Anne Chao and Shen-Ming Lee. Estimating the number of classes via sample coverage. *Journal of the American statistical Association*, 87(417):210–217, 1992.

I-Min A Chen, Victor M Markowitz, Ken Chu, Krishna Palaniappan, Ernest Szeto, Manoj Pillay, Anna Ratner, Jinghua Huang, Evan Andersen, Marcel Huntemann, *et al.* IMG/M: integrated genome and metagenome comparative data analysis system. *Nucleic Acids Research*, 45(D1):D507–D516, 2017.

Guy Cochrane, Ilene Karsch-Mizrachi, Toshihisa Takagi, and International Nucleotide Sequence Database Collaboration. The international nucleotide sequence database collaboration. *Nucleic Acids Research*, 44(D1):D48–D50, 2015.

James R Cole, Qiong Wang, Jordan A Fish, Benli Chai, Donna M McGarrell, Yanni Sun, C Titus Brown, Andrea Porras-Alfaro, Cheryl R Kuske, and James M Tiedje. Ribosomal Database Project: data and tools for high throughput rRNA analysis. *Nucleic Acids Research*, 42(D1):D633–D642, 2013.

International Human Genome Sequencing Consortium *et al.* Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860, 2001.

Felipe da Veiga Leprevost, Björn A Grüning, Saulo Alves Aflitos, Hannes L Röst, Julian Uszkoreit, Harald Barsnes, Marc Vaudel, Pablo Moreno, Laurent Gatto, Jonas Weber, *et al.* BioContainers: an open-source and community-driven framework for software standardization. *Bioinformatics*, page btx192, 2017.

Carlos de Lannoy, Dick de Ridder, and Judith Risse. A sequencer coming of age: de novo genome assembly using MinION reads. *F1000Research*, 6, 2017.

Todd Z DeSantis, Philip Hugenholtz, Neils Larsen, Mark Rojas, Eoin L Brodie, Keith Keller, Thomas Huber, Daniel Dalevi, Ping Hu, and Gary L Andersen. Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Applied and Environmental Microbiology*, 72(7):5069–5072, 2006.

Paolo Di Tommaso, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. Nextflow enables reproducible computational workflows. *Nature Biotechnology*, 35(4):316–319, 2017.

Paul D Donovan, Gabriel Gonzalez, Desmond G Higgins, Geraldine Butler, and Kimihito Ito. Identification of fungi in shotgun metagenomics datasets. *PloS ONE*, 13(2):e0192898, 2018.

Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.

Sean R Eddy. What is a hidden Markov model? *Nature Biotechnology*, 22(10): 1315–1316, 2004.

Sean R Eddy. HMMER3: a new generation of sequence homology search software. *URL: http://hmmer.janelia.org*, 2010.

Sean R Eddy. Accelerated profile HMM searches. *PLoS Computational Biology*, 7 (10):e1002195, 2011.

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

Robert C Edgar. Updating the 97% identity threshold for 16S ribosomal RNA OTUs. *Bioinformatics*, pages 1367–4803, 2018.

Juan José Egozcue, Vera Pawlowsky-Glahn, Glòria Mateu-Figueras, and Carles Barcelo-Vidal. Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, 35(3):279–300, 2003.

John Eid, Adrian Fehr, Jeremy Gray, Khai Luong, John Lyle, Geoff Otto, Paul Peluso, David Rank, Primo Baybayan, Brad Bettman, *et al.* Real-time DNA sequencing from single polymerase molecules. *Science*, 2008.

Emiley A Eloe-Fadrosh, Natalia N Ivanova, Tanja Woyke, and Nikos C Kyrpides. Metagenomics uncovers gaps in amplicon-based detection of microbial diversity. *Nature Microbiology*, 1:15032, 2016a.

Emiley A Eloe-Fadrosh, David Paez-Espino, Jessica Jarett, Peter F Dunfield, Brian P Hedlund, Anne E Dekas, Stephen E Grasby, Allyson L Brady, Hailiang Dong, Brandon R Briggs, *et al.* Global metagenomic survey reveals a new bacterial candidate phylum in geothermal springs. *Nature Communications*, 7: 10476, 2016b.

A Murat Eren, Hilary G Morrison, Susan M Huse, and Mitchell L Sogin. DRISEE overestimates errors in metagenomic sequencing data. *Briefings in Bioinformatics*, 15(5):783–787, 2013.

Philip Ewels, Måns Magnusson, Sverker Lundin, and Max Käller. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, 32(19):3047–3048, 2016.

Paolo Ferragina and Giovanni Manzini. Opportunistic data structures with applications. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 390–398. IEEE, 2000.

Dawn Field, George Garrity, Tanya Gray, Norman Morrison, Jeremy Selengut, Peter Sterk, Tatiana Tatusova, Nicholas Thomson, Michael J Allen, Samuel V Angiuoli, *et al.* The minimum information about a genome sequence (MIGS) specification. *Nature Biotechnology*, 26(5):541, 2008.

Robert D Fleischmann, Mark D Adams, Owen White, Rebecca A Clayton, Ewen F Kirkness, Anthony R Kerlavage, Carol J Bult, Jean-Francois Tomb, Brian A Dougherty, Joseph M Merrick, *et al.* Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. *Science*, 269(5223):496–512, 1995.

Jeremy A Frank, Yao Pan, Ave Tooming-Klunderud, Vincent GH Eijsink, Alice C McHardy, Alexander J Nederbragt, and Phillip B Pope. Improved metagenome assemblies and taxonomic binning using long-read circular consensus sequence data. *Scientific Reports*, 6:25373, 2016.

Ulrich H Frey, Hagen S Bachmann, Jürgen Peters, and Winfried Siffert. PCR-amplification of GC-rich regions: 'slowdown PCR'. *Nature Protocols*, 3(8):1312, 2008.

Daniel R Garalde, Elizabeth A Snell, Daniel Jachimowicz, Botond Sipos, Joseph H Lloyd, Mark Bruce, Nadia Pantic, Tigist Admassu, Phillip James, Anthony Warland, *et al.* Highly parallel direct RNA sequencing on an array of nanopores. *Nature Methods*, 15(3):201, 2018.

George M Garrity, Julia A Bell, and Timothy G Lilburn. Taxonomic outline of the prokaryotes. Bergey's manual of systematic bacteriology. *Springer, New York, Berlin, Heidelberg*, 2004.

Wolfgang Gerlach and Jens Stoye. Taxonomic classification of metagenomic shotgun sequences with CARMA3. *Nucleic Acids Research*, 39(14):e91–e91, 2011.

Wolfgang Gerlach, Sebastian Jünemann, Felix Tille, Alexander Goesmann, and Jens Stoye. WebCARMA: a web application for the functional and taxonomic classification of unassembled metagenomic reads. *BMC Bioinformatics*, 10(1): 430, 2009.

Jack A Gilbert, Dawn Field, Ying Huang, Rob Edwards, Weizhong Li, Paul Gilna, and Ian Joint. Detection of large numbers of novel sequences in the metatranscriptomes of complex marine microbial communities. *PloS ONE*, 3(8):e3042, 2008.

Steven R Gill, Mihai Pop, Robert T DeBoy, Paul B Eckburg, Peter J Turnbaugh, Buck S Samuel, Jeffrey I Gordon, David A Relman, Claire M Fraser-Liggett, and Karen E Nelson. Metagenomic analysis of the human distal gut microbiome. *Science*, 312(5778):1355–1359, 2006.

Stefanie P Gläser, Meysam Taghinasab, Jafargholi Imani, Sebastian Jaenicke, Jens Steinbrenner, Gerald Moser, Peter Kämpfer, Christoph Müller, and Karl-Heinz Kogel. Phylogenetic diversity of fungal endophytes dominating the roots of two privileged plant in a permanent grassland exposed to elevated CO2 and increased surface temperature. Submitted.

Gregory B Gloor, Jean M Macklaim, Vera Pawlowsky-Glahn, and Juan J Egozcue. Microbiome datasets are compositional: and this is not optional. *Frontiers in Microbiology*, 8:2224, 2017.

Jeremy Goecks, Anton Nekrutenko, James Taylor, *et al.* Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology*, 11(8):R86, 2010.

André Goffeau, Bart G Barrell, Howard Bussey, RW Davis, Bernard Dujon, Heinz Feldmann, Francis Galibert, JD Hoheisel, Cr Jacq, Michael Johnston, *et al.* Life with 6000 genes. *Science*, 274(5287):546–567, 1996.

Vicente Gomez-Alvarez, Tracy K Teal, and Thomas M Schmidt. Systematic artifacts in metagenomes from complex microbial communities. *The ISME Journal*, 3(11): 1314, 2009.

Manfred G Grabherr, Brian J Haas, Moran Yassour, Joshua Z Levin, Dawn A Thompson, Ido Amit, Xian Adiconis, Lin Fan, Raktima Raychowdhury, Qiandong Zeng, *et al.* Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology*, 29(7):644–652, 2011.

Michael Gribskov, Andrew D McLachlan, and David Eisenberg. Profile analysis: detection of distantly related proteins. *Proceedings of the National Academy of Sciences*, 84(13):4355–4358, 1987.

Michael Gribskov, Roland Lüthy, and David Eisenberg. Profile analysis. *Methods in Enzymology*, 183:146, 1990.

Malachi Griffith, Jason R Walker, Nicholas C Spies, Benjamin J Ainscough, and Obi L Griffith. Informatics for RNA sequencing: a web resource for analysis on the cloud. *PLoS Computational Biology*, 11(8):e1004393, 2015.

Thomas Hackl, Rainer Hedrich, Jörg Schultz, and Frank Förster. proovread: large-scale high-accuracy PacBio correction through iterative short read consensus. *Bioinformatics*, 30(21):3004–3011, 2014.

Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: Concepts and techniques*. Elsevier, 2011.

Jo Handelsman, Michelle R Rondon, Sean F Brady, Jon Clardy, and Robert M Goodman. Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products. *Chemistry & Biology*, 5(10):R245–R249, 1998.

Niels W Hanson, Kishori M Konwar, and Steven J Hallam. LCA*: an entropy-based measure for taxonomic assignment within assembled metagenomes. *Bioinformatics*, 32(23):3535–3542, 2016.

M Monzoorul Haque, Tarini Shankar Ghosh, Dinakar Komanduri, and Sharmila S Mande. SOrt-ITEMS: Sequence orthology based approach for improved taxonomic estimation of metagenomic sequences. *Bioinformatics*, 25(14):1722–1730, 2009.

Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

Matthias Hess, Alexander Sczyrba, Rob Egan, Tae-Wan Kim, Harshal Chokhawala, Gary Schroth, Shujun Luo, Douglas S Clark, Feng Chen, Tao Zhang, *et al.* Metagenomic discovery of biomass-degrading genes and genomes from cow rumen. *Science*, 331(6016):463–467, 2011.

Tom CJ Hill, Kerry A Walsh, James A Harris, and Bruce F Moffett. Using ecological diversity measures with bacterial communities. *FEMS Microbiology Ecology*, 43 (1):1–11, 2003.

Manuel Holtgrewe. Mason–a read simulator for second generation sequencing data. *Technical Report FU Berlin*, 2010.

Jennifer B Hughes, Jessica J Hellmann, Taylor H Ricketts, and Brendan JM Bohannan. Counting the uncountable: statistical approaches to estimating microbial diversity. *Appl. Environ. Microbiol.*, 67(10):4399–4406, 2001.

Marcel Huntemann, Natalia N Ivanova, Konstantinos Mavromatis, H James Tripp, David Paez-Espino, Kristin Tennessen, Krishnaveni Palaniappan, Ernest Szeto, Manoj Pillay, I-Min A Chen, *et al.* The standard operating procedure of the DOE-JGI Metagenome Annotation Pipeline (MAP v. 4). *Standards in Genomic Sciences*, 11(1):17, 2016.

Sarah Hunter, Matthew Corbett, Hubert Denise, Matthew Fraser, Alejandra Gonzalez-Beltran, Christopher Hunter, Philip Jones, Rasko Leinonen, Craig McAnulla, Eamonn Maguire, *et al.* EBI metagenomics-a new resource for the analysis and archiving of metagenomic data. *Nucleic Acids Research*, 42(D1): D600–D606, 2014.

Daniel H Huson, Alexander F Auch, Ji Qi, and Stephan C Schuster. MEGAN analysis of metagenomic data. *Genome Research*, 17(3):377–386, 2007.

Daniel H Huson, Sina Beier, Isabell Flade, Anna Górska, Mohamed El-Hadidi, Suparna Mitra, Hans-Joachim Ruscheweyh, and Rewati Tappu. MEGAN community edition-interactive exploration and analysis of large-scale microbiome sequencing data. *PLoS Computational Biology*, 12(6):e1004957, 2016.

Curtis Huttenhower, Dirk Gevers, Rob Knight, Sahar Abubucker, Jonathan H Badger, Asif T Chinwalla, Heather H Creasy, Ashlee M Earl, Michael G FitzGerald, Robert S Fulton, *et al.* Structure, function and diversity of the healthy human microbiome. *Nature*, 486(7402):207, 2012.

Doug Hyatt, Gwo-Liang Chen, Philip F LoCascio, Miriam L Land, Frank W Larimer, and Loren J Hauser. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, 11(1):119, 2010.

Michael Imelfort, Donovan Parks, Ben J Woodcroft, Paul Dennis, Philip Hugenholtz, and Gene W Tyson. GroopM: an automated tool for the recovery of population genomes from related metagenomes. *PeerJ*, 2:e603, 2014.

Camilla LC Ip, Matthew Loose, John R Tyson, Mariateresa de Cesare, Bonnie L Brown, Miten Jain, Richard M Leggett, David A Eccles, Vadim Zalunin, John M Urban, *et al.* MinION Analysis and Reference Consortium: Phase 1 data release and analysis. *F1000Research*, 4, 2015.

Sebastian Jaenicke, Christina Ander, Thomas Bekel, Regina Bisdorf, Marcus Dröge, Karl-Heinz Gartemann, Sebastian Jünemann, Olaf Kaiser, Lutz Krause, Felix Tille, *et al.* Comparative and joint analysis of two metagenomic datasets from a biogas fermenter obtained by 454-pyrosequencing. *PLoS ONE*, 6(1):e14519, 2011.

Sebastian Jaenicke, Stefan P. Albaum, Patrick Blumenkamp, Burkhard Linke, Jens Stoye, and Alexander Goesmann. Flexible metagenome analysis using the MGX framework. *Microbiome*, 6(1):76, Apr 2018. ISSN 2049-2618. doi: 10.1186/s40168-018-0460-1.

Miten Jain, Hugh E Olsen, Benedict Paten, and Mark Akeson. The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community. *Genome Biology*, 17(1):239, 2016.

Sarah S Johnson, Elena Zaikova, David S Goerlitz, Yu Bai, and Scott W Tighe. Real-time DNA sequencing in the Antarctic dry valleys using the Oxford Nanopore sequencer. *Journal of Biomolecular Techniques: JBT*, 28(1):2, 2017.

Vanessa Isabell Jurtz, Julia Villarroel, Ole Lund, Mette Voldby Larsen, and Morten Nielsen. MetaPhinder-Identifying Bacteriophage Sequences in Metagenomic Data Sets. *PloS ONE*, 11(9):e0163111, 2016.

Minoru Kanehisa, Susumu Goto, Yoko Sato, Masayuki Kawashima, Miho Furumichi, and Mao Tanabe. Data, information, knowledge and principle: back to metabolism in KEGG. *Nucleic Acids Research*, 42(D1):D199–D205, 2014.

Dongwan D Kang, Jeff Froula, Rob Egan, and Zhong Wang. MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. *PeerJ*, 3:e1165, 2015.

Barry L Karger and Andras Guttman. DNA sequencing by Capillary Electrophoresis. *Electrophoresis*, 30(S1), 2009.

Eric H Kawashima, Laurent Farinelli, and Pascal Mayer. Method of nucleic acid amplification, March 27 2012. US Patent 8,143,008.

Kevin P Keegan, William L Trimble, Jared Wilkening, Andreas Wilke, Travis Harrison, Mark D'Souza, and Folker Meyer. A platform-independent method for detecting errors in metagenomic sequencing data: DRISEE. *PLoS Computational Biology*, 8(6):e1002541, 2012.

W James Kent. BLAT - The BLAST-like alignment tool. *Genome Research*, 12(4): 656–664, 2002.

Daehwan Kim, Li Song, Florian P Breitwieser, and Steven L Salzberg. Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Research*, 26(12):1721–1729, 2016.

Dongjae Kim, Aria S Hahn, Shang-Ju Wu, Niels W Hanson, Kishori M Konwar, and Steven J Hallam. FragGeneScan+: high-throughput short-read gene prediction. In *2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology*, pages 1–7, 2015.

Terje Klemetsen, Inge A Raknes, Juan Fu, Alexander Agafonov, Sudhagar V Balasundaram, Giacomo Tartari, Espen Robertsen, and Nils P Willassen. The MAR databases: development and implementation of databases specific for marine metagenomics. *Nucleic Acids Research*, 46(D1):D692–D699, 2017.

Anna Klindworth, Elmar Pruesse, Timmy Schweer, Jörg Peplies, Christian Quast, Matthias Horn, and Frank Oliver Glöckner. Evaluation of general 16S ribosomal RNA gene PCR primers for classical and next-generation sequencing-based diversity studies. *Nucleic Acids Research*, 41(1):e1–e1, 2013.

Heiner Klingenberg and Peter Meinicke. How To Normalize Metatranscriptomic Count Data For Differential Expression Analysis. *PeerJ*, page 134650, 2017. doi: 10.7717/peerj.3859.

Barbara Klippel, Kerstin Sahm, Alexander Basner, Sigrid Wiebusch, Patrick John, Ute Lorenz, Anke Peters, Fumiyoshi Abe, Kyoma Takahashi, Olaf Kaiser, Alexander Goesmann, Sebastian Jaenicke, Ralf Grote, Koki Horikoshi, and Garabed

Antranikian. Carbohydrate-active enzymes identified by metagenomic analysis of deep-sea sediment bacteria. *Extremophiles*, 18(5):853–863, 2014.

Rob Knight, Alison Vrbanac, Bryn C Taylor, Alexander Aksenov, Chris Callewaert, Justine Debelius, Antonio Gonzalez, Tomasz Kosciolek, Laura-Isobel McCall, Daniel McDonald, *et al.* Best practices for analysing microbiomes. *Nature Reviews Microbiology*, page 1, 2018.

Evguenia Kopylova, Laurent Noé, and Héléne Touzet. SortMeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. *Bioinformatics*, 28 (24):3211–3217, 2012.

Evguenia Kopylova, Laurent Noé, Pierre Pericard, Mikael Salson, and Hélène Touzet. SortMeRNA 2: ribosomal RNA classification for taxonomic assignation. In *Workshop on recent computational advances in metagenomics, ECCB 2014*, 2014.

Sergey Koren, Michael C Schatz, Brian P Walenz, Jeffrey Martin, Jason T Howard, Ganeshkumar Ganapathy, Zhong Wang, David A Rasko, W Richard McCombie, Erich D Jarvis, *et al.* Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nature Biotechnology*, 30(7):693, 2012.

Jonas Korlach, Patrick J Marks, Ronald L Cicero, Jeremy J Gray, Devon L Murphy, Daniel B Roitman, Thang T Pham, Geoff A Otto, Mathieu Foquet, and Stephen W Turner. Selective aluminum passivation for targeted immobilization of single DNA polymerase molecules in zero-mode waveguide nanostructures. *Proceedings of the National Academy of Sciences*, 105(4):1176–1181, 2008.

Johannes Köster and Sven Rahmann. SnakemakeâĂŤa scalable bioinformatics workflow engine. *Bioinformatics*, 28(19):2520–2522, 2012.

Lutz Krause, Naryttza N Diaz, Robert A Edwards, Karl-Heinz Gartemann, Holger Krömeke, Heiko Neuweger, Alfred Pühler, Kai J Runte, Andreas Schlüter, Jens Stoye, *et al.* Taxonomic composition and gene content of a methane-producing microbial community isolated from a biogas reactor. *Journal of Biotechnology*, 136(1-2):91–101, 2008.

Magdalena Kröber, Thomas Bekel, Naryttza N Diaz, Alexander Goesmann, Sebastian Jaenicke, Lutz Krause, Dimitri Miller, Kai J Runte, Prisca Viehöver, Alfred Pühler, *et al.* Phylogenetic characterization of a biogas plant microbial community integrating clone library 16S-rDNA sequences and metagenome sequence data obtained by 454-pyrosequencing. *Journal of Biotechnology*, 142(1):38–49, 2009.

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

Gregory M Kurtzer, Vanessa Sochat, and Michael W Bauer. Singularity: Scientific containers for mobility of compute. *PloS ONE*, 12(5):e0177459, 2017.

David J Lane. 16S/23S rRNA sequencing. *Nucleic acid techniques in bacterial systematics*, 1991.

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

Rasko Leinonen, Ruth Akhtar, Ewan Birney, Lawrence Bower, Ana Cerdeno-Tárraga, Ying Cheng, Iain Cleland, Nadeem Faruque, Neil Goodgame, Richard Gibson, *et al.* The European nucleotide archive. *Nucleic Acids Research*, 39 (suppl_1):D28–D31, 2010a.

Rasko Leinonen, Hideaki Sugawara, Martin Shumway, and International Nucleotide Sequence Database Collaboration. The sequence read archive. *Nucleic Acids Research*, 39(suppl_1):D19–D21, 2010b.

Bo Li and Colin N Dewey. Rsem: accurate transcript quantification from rna-seq data with or without a reference genome. *BMC bioinformatics*, 12(1):323, 2011.

Dinghua Li, Ruibang Luo, Chi-Man Liu, Chi-Ming Leung, Hing-Fung Ting, Kunihiko Sadakane, Hiroshi Yamashita, and Tak-Wah Lam. MEGAHIT v1. 0: A fast and scalable metagenome assembler driven by advanced methodologies and community practices. *Methods*, 102:3–11, 2016.

Dinghua Li, Yukun Huang, Chi-Ming Leung, Ruibang Luo, Hing-Fung Ting, and Tak-Wah Lam. MegaGTA: a sensitive and accurate metagenomic gene-targeted assembler using iterative de Bruijn graphs. *BMC Bioinformatics*, 18(12):408, 2017.

Heng Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv preprint arXiv:1303.3997*, 2013.

Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 2018.

Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

Thomas Lingner, Kathrin Petra Aßhauer, Fabian Schreiber, and Peter Meinicke. CoMet–a web server for comparative functional profiling of metagenomes. *Nucleic Acids Research*, 39(suppl_2):W518–W523, 2011.

Burkhard Linke, Robert Giegerich, and Alexander Goesmann. Conveyor: a workflow engine for bioinformatic analyses. *Bioinformatics*, 27(7):903–911, 2011.

Donglin Liu and Joel H Graber. Quantitative comparison of EST libraries requires compensation for systematic biases in cDNA generation. *BMC Bioinformatics*, 7 (1):77, 2006.

Jiemeng Liu, Haifeng Wang, Hongxing Yang, Yizhe Zhang, Jinfeng Wang, Fangqing Zhao, and Ji Qi. Composition-based classification of short metagenomic sequences elucidates the landscapes of taxonomic and functional enrichment of microorganisms. *Nucleic Acids Research*, 41(1):gks828, 2012.

Nicholas J Loman, Raju V Misra, Timothy J Dallman, Chrystala Constantinidou, Saheer E Gharbia, John Wain, and Mark J Pallen. Performance comparison of benchtop high-throughput sequencing platforms. *Nature Biotechnology*, 30(5): 434, 2012.

Martina Lori, Gabin Piton, Sarah Symanczik, Nicolas Legay, Lijbert Brussaard, Sebastian Jaenicke, Eduardo Nascimento, Filipa Reis, Paulo Sousa, Paul Mäder, Andreas Gattinger, Jean-Christophe Cleément, and Arnaud Foulquier. The impact of ecological intensive management on proteolytic microbial communities is central for nitrogen-related ecosystem service provisioning under altered rain regimes: a cross country experiment. In preparation.

Martina Lori, Sarah Symanczik, Paul Mäder, Norah Efosa, Sebastian Jaenicke, Franz Buegger, Simon Tresch, Alexander Goesmann, and Andreas Gattinger. Distinct nitrogen provisioning from organic amendments in soil as influenced by farming system and water regime. *Frontiers in Environmental Science*, 6:40, 2018.

Catherine Lozupone, Manuel E Lladser, Dan Knights, Jesse Stombaugh, and Rob Knight. UniFrac: an effective distance metric for microbial community comparison. *The ISME Journal*, 5(2):169, 2011.

Bin Ma, John Tromp, and Ming Li. PatternHunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440–445, 2002.

Rachel Mackelprang, Mark P Waldrop, Kristen M DeAngelis, Maude M David, Krystle L Chavarria, Steven J Blazewicz, Edward M Rubin, and Janet K Jansson. Metagenomic analysis of a permafrost microbial community reveals a rapid response to thaw. *Nature*, 480(7377):368, 2011.

Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *siam Journal on Computing*, 22(5):935–948, 1993.

Julian R. Marchesi and Jacques Ravel. The vocabulary of microbiome research: a proposal. *Microbiome*, 3(1):31, 2015.

Marcel Margulies, Michael Egholm, William E. Altman, Said Attiya, Joel S. Bader, Lisa A. Bemben, Jan Berka, Michael S. Braverman, Yi-Ju Chen, Zhoutao Chen,

Scott B. Dewell, Lei Du, Joseph M. Fierro, Xavier V. Gomes, Brian C. Godwin, Wen He, Scott Helgesen, Chun He Ho, Gerard P. Irzyk, Szilveszter C. Jando, Maria L.I. Alenquer, Thomas P. Jarvie, Kshama B. Jirage, Jong-Bum Kim, James R. Knight, Janna R. Lanza, John H. Leamon, Steven M. Lefkowitz, Ming Lei, Jing Li, Kenton L. Lohman, Hong Lu, Vinod M. Makhijani, Keith E. McDade, Michael P. McKenna, Eugene W. Myers, Elizabeth Nickerson, John R. Nobile, Ramona Plant, Bernard P. Puc, Michael T. Ronan, George T. Roth, Gary J. Sarkis, Jan F. Simons, John W. Simpson, Maithreyan Srinivasan, Karrie R. Tartaro, Alexander Tomasz, Kari A. Vogt, Greg A. Volkmer, Shally H. Wang, Yong Wang, Michael P. Weiner, Pengguang Yu, Richard F. Begley, and Jonathan M. Rothberg. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, 2005.

Victor M Markowitz, I-Min A Chen, Krishna Palaniappan, Ken Chu, Ernest Szeto, Yuri Grechkin, Anna Ratner, Biju Jacob, Jinghua Huang, Peter Williams, *et al.* IMG: the integrated microbial genomes database and comparative analysis system. *Nucleic Acids Research*, 40(D1):D115–D122, 2011.

Victor M Markowitz, I-Min A Chen, Ken Chu, Ernest Szeto, Krishna Palaniappan, Yuri Grechkin, Anna Ratner, Biju Jacob, Amrita Pati, Marcel Huntemann, *et al.* IMG/M: the integrated metagenome data management and comparative analysis system. *Nucleic Acids Research*, 40(D1):D123–D129, 2012.

George Marsaglia. Xorshift RNGs. *Journal of Statistical Software*, 8(14):1–6, 2003.

Josep A Martín-Fernández, Carles Barceló-Vidal, and Vera Pawlowsky-Glahn. Dealing with zeros and missing values in compositional data sets using nonparametric imputation. *Mathematical Geology*, 35(3):253–278, 2003.

María C Martini, Daniel Wibberg, Mauricio Lozano, Gonzalo Torres Tejerizo, Francisco J Albicoro, Sebastian Jaenicke, Jan Dirk Van Elsas, Alejandro Petroni, M Pilar Garcillán-Barcia, Fernando De La Cruz, Andreas Schlüter, Alfred Pühler, Mariano Pistorio, Antonio Lagares, and Maria F Del Papa. Genomics of high molecular weight plasmids isolated from an on-farm biopurification system. *Scientific Reports*, 6:28284, 2016.

João F Matias Rodrigues, Thomas SB Schmidt, Janko Tackmann, and Christian von Mering. MAPseq: highly efficient k-mer search with confidence estimates, for rRNA sequence analysis. *Bioinformatics*, 33(23):3808–3810, 2017.

Makoto Matsumoto and Takuji Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1):3–30, 1998.

Irena Maus, Daniela E Koeck, Katharina G Cibis, Sarah Hahnke, Yong S Kim, Thomas Langer, Jana Kreubel, Marcel Erhard, Andreas Bremges, Sandra Off,

Yvonne Stolze, Sebastian Jaenicke, Alexander Goesmann, Alexander Sczyrba, Paul Scherer, Helmut König, Wolfgang H Schwarz, Vladimir V Zverlov, Wolfgang Liebl, Alfred Pühler, Andreas Schlüter, and Michael Klocke. Unraveling the microbiome of a thermophilic biogas plant by metagenome and metatranscriptome analysis complemented by characterization of bacterial and archaeal isolates. *Biotechnology for Biofuels*, 9(1):171, 2016.

Allan M Maxam and Walter Gilbert. A new method for sequencing DNA. *Proceedings of the National Academy of Sciences*, 74(2):560–564, 1977.

Alice C McHardy and Isidore Rigoutsos. What's in the mix: phylogenetic classification of metagenome sequence samples. *Current opinion in microbiology*, 10(5): 499–503, 2007.

Alice C McHardy, Hector Garcia Martin, Aristotelis Tsirigos, Philip Hugenholtz, and Isidore Rigoutsos. Accurate phylogenetic classification of variable-length DNA fragments. *Nature Methods*, 4(1):63–72, 2007.

Peter Meinicke. UProC: tools for ultra-fast protein domain classification. *Bioinformatics*, 31(9):1382–1388, 2014.

Peter Menzel, Kim Lee Ng, and Anders Krogh. Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nature Communications*, 7:11257–11257, 2016.

Nirav Merchant, Eric Lyons, Stephen Goff, Matthew Vaughn, Doreen Ware, David Micklos, and Parker Antin. The iPlant collaborative: cyberinfrastructure for enabling data to discovery for the life sciences. *PLoS Biology*, 14(1):e1002342, 2016.

Frédéric Meunier, Olivier Gandouet, Éric Fusy, and Philippe Flajolet. HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm. *Discrete Mathematics & Theoretical Computer Science*, 2007.

Folker Meyer, Alexander Goesmann, Alice C McHardy, Daniela Bartels, Thomas Bekel, Jörn Clausen, Jörn Kalinowski, Burkhard Linke, Oliver Rupp, Robert Giegerich, *et al.* GenDB-an open source genome annotation system for prokaryote genomes. *Nucleic Acids Research*, 31(8):2187–2195, 2003.

Folker Meyer, Daniel Paarmann, Mark D'Souza, Robert Olson, Elizabeth M Glass, Michael Kubal, Tobias Paczian, A Rodriguez, Rick Stevens, Andreas Wilke, *et al.* The metagenomics RAST server–a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics*, 9(1):386, 2008.

Jason R Miller, Peng Zhou, Joann Mudge, James Gurtowski, Hayan Lee, Thiruvarangan Ramaraj, Brian P Walenz, Junqi Liu, Robert M Stupar, Roxanne Denny, *et al.* Hybrid assembly with long and short reads improves discovery of gene family expansions. *BMC Genomics*, 18(1):541, 2017.

Samuel S Minot, Niklas Krumm, and Nicholas B Greenfield. One Codex: A sensitive and accurate data platform for genomic microbial identification. *bioRxiv*, page 027607, 2015.

Alex L Mitchell, Maxim Scheremetjew, Hubert Denise, Simon Potter, Aleksandra Tarkowska, Matloob Qureshi, Gustavo A Salazar, Sebastien Pesseat, Miguel A Boland, Fiona MI Hunter, *et al.* EBI Metagenomics in 2017: enriching the analysis of microbial communities, from sequence reads to assemblies. *Nucleic Acids Research*, 46(D1):D726–D735, 2017.

Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, 5(7):621–628, 2008.

Toshiaki Namiki, Tsuyoshi Hachiya, Hideaki Tanaka, and Yasubumi Sakakibara. MetaVelvet: an extension of Velvet assembler to de novo metagenome assembly from short sequence reads. *Nucleic Acids Research*, 40(20):e155–e155, 2012.

Eric P Nawrocki and Sean R Eddy. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics*, 29(22):2933–2935, 2013.

Eric P Nawrocki, Sarah W Burge, Alex Bateman, Jennifer Daub, Ruth Y Eberhardt, Sean R Eddy, Evan W Floden, Paul P Gardner, Thomas A Jones, John Tate, *et al.* Rfam 12.0: updates to the RNA families database. *Nucleic Acids Research*, 43(D1):D130–D137, 2014.

Johanna Nelkner, Christian Henke, Timo Wentong Lin, Wiebke Pätzold, Julia Hassa, Sebastian Jaenicke, Rita Grosch, Alfred Pühler, Alexander Sczyrba, and Andreas Schlüter. Effect of long-term farming practices on agricultural soil microbiome members represented by metagenomically assembled genomes (MAGs) and their predicted plant-beneficial genes. *Genes*, 10(6), 2019.

Beifang Niu, Limin Fu, Shulei Sun, and Weizhong Li. Artificial and natural duplicates in pyrosequencing reads of metagenomic data. *BMC Bioinformatics*, 11(1): 187, 2010.

Vímac Nolla-Ardèvol, Marc Strous, and Halina Elisabeth Tegetmeyer. Anaerobic digestion of the microalga Spirulina at extreme alkaline conditions: biogas production, metagenome, and metatranscriptome. *Frontiers in Microbiology*, 6:597, 2015.

Sergey Nurk, Dmitry Meleshko, Anton Korobeynikov, and Pavel A Pevzner. metaSPAdes: a new versatile metagenomic assembler. *Genome Research*, 27(5):824–834, 2017.

Pål Nyrén. Enzymatic method for continuous monitoring of DNA polymerase activity. *Analytical Biochemistry*, 167(2):235–238, 1987.

Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R Pocock, Anil Wipat, *et al.* Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, 2004.

Taku Onodera and Tetsuo Shibuya. The gapped spectrum kernel for support vector machines. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 1–15. Springer, 2013.

Vera Ortseifen, Yvonne Stolze, Irena Maus, Alexander Sczyrba, Andreas Bremges, Stefan P Albaum, Sebastian Jaenicke, Jochen Fracowiak, Alfred Pühler, and Andreas Schlüter. An integrated metagenome and-proteome analysis of the microbial community residing in a biogas production plant. *Journal of Biotechnology*, 231: 268–279, 2016.

Ross Overbeek, Tadhg Begley, Ralph M Butler, Jomuna V Choudhuri, Han-Yu Chuang, Matthew Cohoon, Valérie de Crécy-Lagard, Naryttza Diaz, Terry Disz, Robert Edwards, *et al.* The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. *Nucleic Acids Research*, 33(17): 5691–5702, 2005.

Donovan H Parks, Christian Rinke, Maria Chuvochina, Pierre-Alain Chaumeil, Ben J Woodcroft, Paul N Evans, Philip Hugenholtz, and Gene W Tyson. Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. *Nature Microbiology*, 2(11):1533, 2017.

Kaustubh Raosaheb Patil, Linus Roune, and Alice Carolyn McHardy. The PhyloPythiaS web server for taxonomic assignment of metagenome sequences. *PloS ONE*, 7(6):e38581, 2012.

Jakob Skou Pedersen and Jotun Hein. Gene finding with a hidden Markov model of genome structure and evolution. *Bioinformatics*, 19(2):219–227, 2003.

Trestan Pillonel, Claire Bertelli, and Gilbert Greub. Environmental Metagenomic Assemblies Reveal Seven New Highly Divergent Chlamydial Lineages and Hallmarks of a Conserved Intracellular Lifestyle. *Frontiers in Microbiology*, 9:79, 2018.

Hendrik N Poinar, Carsten Schwarz, Ji Qi, Beth Shapiro, Ross DE MacPhee, Bernard Buigues, Alexei Tikhonov, Daniel H Huson, Lynn P Tomsho, Alexander Auch, *et al.* Metagenomics to paleogenomics: large-scale sequencing of mammoth DNA. *Science*, 311(5759):392–394, 2006.

Aaron Pomerantz, Nicolás Peñafiel, Alejandro Arteaga, Lucas Bustamante, Frank Pichardo, Luis A Coloma, César L Barrio-Amorós, David Salazar-Valenzuela, and Stefan Prost. Real-time DNA barcoding in a rainforest using nanopore

sequencing: opportunities for rapid biodiversity assessments and local capacity building. *GigaScience*, 7(4):giy033, 2018.

Elmar Pruesse, Christian Quast, Katrin Knittel, Bernhard M Fuchs, Wolfgang Ludwig, Jörg Peplies, and Frank Oliver Glöckner. SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. *Nucleic Acids Research*, 35(21):7188–7196, 2007.

Kim D Pruitt, Tatiana Tatusova, and Donna R Maglott. NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research*, 35(suppl_1):D61–D65, 2006.

Christopher Quince, Anders Lanzén, Thomas P Curtis, Russell J Davenport, Neil Hall, Ian M Head, L Fiona Read, and William T Sloan. Accurate determination of microbial diversity from 454 pyrosequencing data. *Nature Methods*, 6(9):639, 2009.

Thomas P Quinn, Ionas Erb, Greg Gloor, Cedric Notredame, Mark F Richardson, and Tamsyn M Crowley. A field guide for the compositional analysis of any-omics data. *bioRxiv*, page 484766, 2018a.

Thomas P Quinn, Ionas Erb, Mark F Richardson, and Tamsyn M Crowley. Understanding sequencing data as compositions: an outlook and review. *Bioinformatics*, 34(16):2870–2878, 2018b.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014. URL `http://www.R-project.org/`.

Simone Rampelli, Matteo Soverini, Silvia Turroni, Sara Quercia, Elena Biagi, Patrizia Brigidi, and Marco Candela. ViromeScan: a new tool for metagenomic viral community profiling. *BMC Genomics*, 17(1):165, 2016.

Bhaskar Reddy, Jitendra Pandey, and Suresh Kumar Dubey. Assessment of environmental gene tags linked with carbohydrate metabolism and chemolithotrophy associated microbial community in River Ganga. *Gene*, 704:31–41, 2019.

Michael Reich, Ted Liefeld, Joshua Gould, Jim Lerner, Pablo Tamayo, and Jill P Mesirov. GenePattern 2.0. *Nature Genetics*, 38(5):500, 2006.

Mina Rho, Haixu Tang, and Yuzhen Ye. FragGeneScan: predicting genes in short and error-prone reads. *Nucleic Acids Research*, 38(20):e191–e191, 2010.

Anthony Rhoads and Kin Fai Au. PacBio sequencing and its applications. *Genomics, Proteomics & Bioinformatics*, 13(5):278–289, 2015.

Mostafa Ronaghi, Samer Karamohamed, Bertil Pettersson, Mathias Uhlén, and Pål Nyrén. Real-time DNA sequencing using detection of pyrophosphate release. *Analytical Biochemistry*, 242(1):84–89, 1996.

Michael G Ross, Carsten Russ, Maura Costello, Andrew Hollinger, Niall J Lennon, Ryan Hegarty, Chad Nusbaum, and David B Jaffe. Characterizing and measuring bias in sequence data. *Genome Biology*, 14(5):R51, 2013.

Jonathan M Rothberg, Wolfgang Hinz, Todd M Rearick, Jonathan Schultz, William Mileski, Mel Davey, John H Leamon, Kim Johnson, Mark J Milgrew, Matthew Edwards, *et al.* An integrated semiconductor device enabling non-optical genome sequencing. *Nature*, 475(7356):348, 2011.

Raquel Rubio, Anna Jofré, Belén Martín, Teresa Aymerich, and Margarita Garriga. Characterization of lactic acid bacteria isolated from infant faeces as potential probiotic starter cultures for fermented sausages. *Food Microbiology*, 38:303–311, 2014.

Oliver Rupp, Madolyn L MacDonald, Shangzhong Li, Heena Dhiman, Shawn Polson, Sven Griep, Kelley Heffner, Inmaculada Hernandez, Karina Brinkrolf, Vaibhav Jadhav, *et al.* A reference genome of the Chinese hamster based on a hybrid assembly strategy. *Biotechnology and Bioengineering*, 115(8):2087–2100, 2018.

Paul Saary, Kristoffer Forslund, Peer Bork, and Falk Hildebrand. RTK: efficient rarefaction analysis of large datasets. *Bioinformatics*, 33(16):2594–2595, 2017.

Leena Salmela, Riku Walve, Eric Rivals, and Esko Ukkonen. Accurate self-correction of errors in long reads using de Bruijn graphs. *Bioinformatics*, 33(6):799–806, 2016.

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

Frederick Sanger, Steven Nicklen, and Alan R Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467, 1977.

Patrick D Schloss, Sarah L Westcott, Thomas Ryabin, Justine R Hall, Martin Hartmann, Emily B Hollister, Ryan A Lesniewski, Brian B Oakley, Donovan H Parks, Courtney J Robinson, *et al.* Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology*, 75(23):7537–7541, 2009.

Andreas Schlüter, Thomas Bekel, Naryttza N Diaz, Michael Dondrup, Rudolf Eichenlaub, Karl-Heinz Gartemann, Irene Krahn, Lutz Krause, Holger Krömeke,

Olaf Kruse, *et al.* The metagenome of a biogas-producing microbial community of a production-scale biogas plant fermenter analysed by the 454-pyrosequencing technology. *Journal of Biotechnology*, 136(1-2):77–90, 2008.

Zala Schmautz, Andreas Graber, Sebastian Jaenicke, Alexander Goesmann, Ranka Junge, and Theo HM Smits. Microbial diversity in different compartments of an aquaponics system. *Archives of microbiology*, 199(4):613–620, 2017.

Robert Schmieder and Robert Edwards. Quality control and preprocessing of metagenomic datasets. *Bioinformatics*, 27(6):863–864, 2011.

Patrick Schwientek, Rafael Szczepanowski, Christian Rückert, Jens Stoye, and Alfred Pühler. Sequencing of high G+C microbial genomes using the ultrafast pyrosequencing technology. *Journal of Biotechnology*, 155(1):68–77, 2011.

Patrick Schwientek, Rafael Szczepanowski, Christian Rückert, Jörn Kalinowski, Andreas Klein, Klaus Selber, Udo F Wehmeier, Jens Stoye, and Alfred Pühler. The complete genome sequence of the acarbose producer Actinoplanes sp. SE50/110. *BMC Genomics*, 13(1):112, 2012.

Fritz J Sedlazeck, Hayan Lee, Charlotte A Darby, and Michael C Schatz. Piercing the dark matter: bioinformatics of long-range sequencing and mapping. *Nature Reviews Genetics*, page 1, 2018.

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

Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

Yadav Shivani, Yadav Subhash, Ch Sasikala, and Ch V Ramana. Characterisation of a newly isolated member of a candidatus lineage, *Marispirochaeta aestuarii* gen. nov., sp. nov. *International Journal of Systematic and Evolutionary Microbiology*, 67(10):3929–3936, 2017.

Justin D Silverman, Alex D Washburne, Sayan Mukherjee, and Lawrence A David. A phylogenetic transform enhances analysis of compositional microbiota data. *Elife*, 6:e21887, 2017.

Edward H Simpson. Measurement of Diversity. *Nature*, 163(4148):688, 1949.

Peter Skewes-Cox, Thomas J Sharpton, Katherine S Pollard, and Joseph L DeRisi. Profile hidden Markov models for the detection of viruses within metagenomic sequence data. *PloS ONE*, 9(8):e105067, 2014.

Peter HA Sneath and Robert R Sokal. Principles of numerical taxonomy. *San Francisco and London I*, 963, 1963.

Charlotte Soneson and Mauro Delorenzi. A comparison of methods for differential expression analysis of RNA-seq data. *BMC Bioinformatics*, 14(1):91, 2013.

Erko Stackebrandt and Brett M Goebel. Taxonomic note: a place for DNA-DNA reassociation and 16S rRNA sequence analysis in the present species definition in bacteriology. *International Journal of Systematic and Evolutionary Microbiology*, 44(4):846–849, 1994.

Wolfgang R Streit and Ruth A Schmitz. Metagenomics–the key to the uncultured microbes. *Current opinion in microbiology*, 7(5):492–498, 2004.

Chien-Hao Su, Ming-Tsung Hsu, Tse-Yi Wang, Sufeng Chiang, Jen-Hao Cheng, Francis C Weng, Cheng-Yan Kao, Daryi Wang, and Huai-Kuang Tsai. MetaABC-an integrated metagenomics platform for data adjustment, binning and clustering. *Bioinformatics*, 27(16):2298–2299, 2011.

Shuji Suzuki, Masanori Kakuta, Takashi Ishida, and Yutaka Akiyama. GHOSTX: An Improved Sequence Homology Search Algorithm Using a Query Suffix Array and a Database Suffix Array. *PloS ONE*, 9(8):e103833, 2014a.

Shuji Suzuki, Masanori Kakuta, Takashi Ishida, and Yutaka Akiyama. Faster sequence homology searches by clustering subsequences. *Bioinformatics*, 31(8): 1183–1190, 2014b.

Hanno Teeling, Jost Waldmann, Thierry Lombardot, Margarete Bauer, and Frank Oliver Glöckner. TETRA: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in DNA sequences. *BMC Bioinformatics*, 5(1):163, 2004.

Cole Trapnell, Brian A Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Marijke J Van Baren, Steven L Salzberg, Barbara J Wold, and Lior Pachter. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, 28(5):511–515, 2010.

Duy Tin Truong, Eric A Franzosa, Timothy L Tickle, Matthias Scholz, George Weingart, Edoardo Pasolli, Adrian Tett, Curtis Huttenhower, and Nicola Segata. MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nature Methods*, 12 (10):902–903, 2015.

Peter J Turnbaugh, Ruth E Ley, Michael A Mahowald, Vincent Magrini, Elaine R Mardis, and Jeffrey I Gordon. An obesity-associated gut microbiome with increased capacity for energy harvest. *Nature*, 444(7122):1027–131, 2006.

Simon Urbanek. Rserve–A Fast Way to Provide R Functionality to Applications. In *Proc. of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003), ISSN 1609-395X, Eds.: Kurt Hornik, Friedrich Leisch & Achim Zeileis*. Citeseer, 2003.

Andries Johannes Van der Walt, Marc Warwick Van Goethem, Jean-Baptiste Ramond, Thulani Peter Makhalanyane, Oleg Reva, and Don Arthur Cowan. Assembling metagenomes, one community at a time. *BMC Genomics*, 18(1):521, 2017.

Thanh Van Nguyen, Daniel Wibberg, Kai Battenberg, Jochen Blom, Brian Vanden Heuvel, Alison M Berry, Jörn Kalinowski, and Katharina Pawlowski. An assemblage of Frankia Cluster II strains from California contains the canonical nod genes and also the sulfotransferase gene nodH. *BMC Genomics*, 17(1):796, 2016.

J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, *et al.* The sequence of the human genome. *Science*, 291(5507):1304–1351, 2001.

J Craig Venter, Karin Remington, John F Heidelberg, Aaron L Halpern, Doug Rusch, Jonathan A Eisen, Dongying Wu, Ian Paulsen, Karen E Nelson, William Nelson, *et al.* Environmental genome shotgun sequencing of the Sargasso Sea. *Science*, 304(5667):66–74, 2004.

Timothy Vogel, Pascal Simonet, Janet Jansson, Penny Hirsch, James Tiedje, Jan Van Elsas, Mark Bailey, Renaud Nalin, and Laurent Philippot. TerraGenome: a consortium for the sequencing of a soil metagenome. *Nature Reviews Microbiology*, 7(4):252, 2009.

Panagiotis D Vouzis and Nikolaos V Sahinidis. GPU-BLAST: using graphics processors to accelerate protein sequence alignment. *Bioinformatics*, 27(2):182–188, 2011.

Calum J Walsh, Caitriona M Guinane, Paul W O'Toole, and Paul D Cotter. A Profile Hidden Markov Model to investigate the distribution and frequency of LanB-encoding lantibiotic modification genes in the human oral and gut microbiome. *PeerJ*, 5:e3254, 2017.

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

Qiong Wang, Jordan A Fish, Mariah Gilman, Yanni Sun, C Titus Brown, James M Tiedje, and James R Cole. Xander: employing a novel method for efficient gene-targeted metagenomic assembly. *Microbiome*, 3(1):32, 2015.

Anthony Westbrook, Jordan Ramsdell, Taruna Schuelke, Louisa Normington, R Daniel Bergeron, W Kelley Thomas, and Matthew D MacManes. PALADIN: protein alignment for functional profiling whole metagenome shotgun data. *Bioinformatics*, 33(10):1473–1478, 2017.

Ruby White, Christophe Pellefigues, Franca Ronchese, Olivier Lamiable, and David Eccles. Investigation of chimeric reads using the MinION. *F1000Research*, 6: 631–631, 2017.

Tae Woong Whon, Won-Hyong Chung, Mi Young Lim, Eun-Ji Song, Pil Soo Kim, Dong-Wook Hyun, Na-Ri Shin, Jin-Woo Bae, and Young-Do Nam. The effects of sequencing platforms on phylogenetic resolution in 16S rRNA gene profiling of human feces. *Scientific data*, 5, 2018.

Andreas Wilke, Travis Harrison, Jared Wilkening, Dawn Field, Elizabeth M Glass, Nikos Kyrpides, Konstantinos Mavrommatis, and Folker Meyer. The M5nr: a novel non-redundant database containing protein sequences and annotations from multiple sources and associated tools. *BMC Bioinformatics*, 13(1):141, 2012.

Andreas Wilke, Jared Bischof, Travis Harrison, Tom Brettin, Mark D'Souza, Wolfgang Gerlach, Hunter Matthews, Tobias Paczian, Jared Wilkening, Elizabeth M Glass, *et al.* A RESTful API for accessing microbial community data for MG-RAST. *PLoS Computational Biology*, 11(1):e1004008, 2015.

David Wilkins, Xiao-Ying Lu, Zhiyong Shen, Jiapeng Chen, and Patrick KH Lee. Pyrosequencing of mcrA and archaeal 16S rRNA genes reveals diversity and substrate preferences of methanogen communities in anaerobic digesters. *Applied and Environmental Microbiology*, 81(2):604–613, 2015.

Derrick Wood and Steven Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, 15(3):R46, 2014.

Cathy H Wu, UniProt Consortium, *et al.* UniProt: the universal protein knowledgebase. *Nucleic Acids Research*, 45(D1):D158–D169, 2016.

Ray Wu. Nucleotide sequence analysis of DNA. *Nature New Biology*, 236(68):198, 1972.

Yuzhen Ye, Jeong-Hyeon Choi, and Haixu Tang. RAPSearch: a fast protein similarity search tool for short reads. *BMC Bioinformatics*, 12(1):159, 2011.

Pelin Yilmaz, Renzo Kottmann, Dawn Field, Rob Knight, James R Cole, Linda Amaral-Zettler, Jack A Gilbert, Ilene Karsch-Mizrachi, Anjanette Johnston, Guy Cochrane, *et al.* Minimum information about a marker gene sequence (MIMARKS) and minimum information about any (x) sequence (MIxS) specifications. *Nature Biotechnology*, 29(5):415, 2011.

Shibu Yooseph, Granger Sutton, Douglas B Rusch, Aaron L Halpern, Shannon J Williamson, Karin Remington, Jonathan A Eisen, Karla B Heidelberg, Gerard Manning, Weizhong Li, *et al.* The Sorcerer II Global Ocean Sampling expedition: expanding the universe of protein families. *PLoS Biology*, 5(3):e16, 2007.

Martha Zakrzewski, Thomas Bekel, Christina Ander, Alfred Pühler, Oliver Rupp, Jens Stoye, Andreas Schlüter, and Alexander Goesmann. MetaSAMS-a novel software platform for taxonomic classification, functional annotation and comparative analysis of metagenome datasets. *Journal of Biotechnology*, 167(2):156–165, 2013.

Evgeni M Zdobnov and Rolf Apweiler. InterProScan–an integration platform for the signature-recognition methods in InterPro. *Bioinformatics*, 17(9):847–848, 2001.

Yuan Zhang and Yanni Sun. HMM-FRAME: accurate protein domain classification for metagenomic sequences containing frameshift errors. *BMC Bioinformatics*, 12(1):198, 2011.

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

Wenhan Zhu, Alexandre Lomsadze, and Mark Borodovsky. Ab initio gene identification in metagenomic sequences. *Nucleic Acids Research*, 38(12):e132–e132, 2010.

Acknowledgements

Alexander Goesmann and Jens Stoye are acknowledged for the opportunity and supervision of this PhD thesis. Without their expert advice and the continuous financial funding, this work would not have been possible. I'm also grateful to Peter Belmann and Patrick Blumenkamp for the contribution of software components which they developed during their B.Sc. and M.Sc. theses under my supervision. My colleagues Jochen Blom and Burkhard Linke are thanked for advice, collaboration on many different projects and many fruitful and sometimes heated discussions.

Gießen, June 2019

Sebastian Jaenicke

ERKLÄRUNG

Ich, Sebastian Jaenicke, erkläre hiermit, daß ich die Dissertation selbständig erarbeitet und keine anderen als die in der Dissertation angegebenen Hilfsmittel benutzt habe.

Gießen, den 5. Juni 2019

Sebastian Jaenicke