Metadata-driven computational (meta)genomics

A practical machine learning approach

Madis Rumming

Minu emale.

A vast amount of bacterial and archaeal genomic sequences have been generated in the past decade through single cell sequencing and in particular binning of metagenomic sequences, but a detailed characterization of the functional features and observable phenotypes of such novel genomes is mostly unknown and thus missing. Machine learning models are trained on previously annotated organisms in relation to the mentioned traits and can be used for the characterization of so far undiscovered novel microbial organisms. The metadata is also used to enrich microbial community profiles with this kind of information, and a client-side webtool has been developed for comparative visualizations of these profiles.

The manuscript of this doctorate thesis was handed in at the Faculty of Technology, Bielefeld University, in the summer term 2017. The disputation was succesfully held in the summer term 2017. For publication purposes the manuscript has been slightly modified.

Printed on non-ageing paper – ISO 9706

Acknowledgement

I would kindly like to thank my supervisor Dr. Alexander Sczyrba, who was always supporting my research and gave me great advice for conducting in-silico metagenome research aside those presented in this thesis. Without his great support I would not have been able to finish my doctorate studies. I also thank my second referee Prof. Dr. Leonid Chindelevitch for his excellent remarks and criticism of the thesis manuscript.

Being part of the DFG funded german-canadian international research training group 1906/1 *Computational Methods for the Analysis of the Diversity and Dynamics of Genomes* was fundamental to my success during my doctorate studies. They provided me with funding, but more important with fruitful and intense discussion with my DiDy fellows also. Being a visiting researcher at the Simon Frasier University in Burnaby, Canada, and the DOE Joint Genome Institute in Walnut Creek, USA, as well as at Illumina Inc., Cambridge, Great Britain, would not have been possible without DiDy. I also thank the DFG, Prof. Dr. Robert Giegerich, Prof. Dr. Arndt Borkhardt, and Dr. Alexander Sczyrba for funding during my doctorate studies. Big cheers go out to my research groups Computational Metagenomics as well as Practical Computer Science for great scientific and fun discussions, especially with Cédric Saule and my office mate Jan Krüger. It was a real pleasure to work with you guys over all the years! My wife Lena Gumpert supported and inspired me all the time. She always reminded me to think outside the box – Ma armastan sind!

Finally, I have to thank my mother Karin Rumming to whom I dedicate this work.

Bielefeld, March 2018
Madis Rumming

Contents

1 Introduction

Traditional ecology investigates diversity in the macro universe, whereas microbial ecology is focused on a microscopic level. The main driver of microbial ecology, i. e., population studies targeting the understanding and interpretation of manifold microbial communities and their functional repertoire, is nowadays metagenomic studies, the analysis of a biotope's genome at population and functional level.

The vast amount of genomic sequences processed within the field of metagenomics is accompanied by annotations of the sample, describing the habitat as the source of the biological sample. This kind of secondary information is also called metadata, which is also available for particular organisms, describing their functional metabolic capabilities and observable phenotypes. The focus of this thesis is to evaluate the utilization of metadata in order to characterize novel, so far uncultivable bacterial and archaeal organisms and thus gain a better understanding of their lifestyle. Furthermore, it is of interest how the practical use of a basic microbial taxonomic community profile can be extended utilizing metadata.

Thesis Structure This thesis begins with a general introduction to metagenomics, community profiling, their application for functional annotation of reconstructed organisms (culture free technique), single cell sequencing, and leads to conclusions about the analytical and computational challenges involved. The following is structured into four parts, where the first part, 'MetaStone – Foundation for Metagenomic Storage of novel entities' from page 27 onwards, describes the *MetaStone* system acting as the data basis and computational foundation of the phenotype prediction tool *PhenoPointer* and the metagenome visualization platform *MVIZ*. The second part's scope is the prediction of bacterial and archaeal phenotype, 'PP – PhenoPointer' from page 45 onwards. *MVIZ* is introduced in the third part, namely 'MVIZ – Metagenome VIZualition', and starts at page 109. The thesis ends with the 'From genotype over phenotype to function-driven metagenomics' on page 129, presenting a summary of the results and a concluding discussion, with future prospects for the application of metadata in metagenomics.

2 Metagenomics, an extension to traditional ecology

Observational study of the various faunal and floral species residing side by side as a community in an ecosystem and the systematic investigation of the whole complex was first defined as *ecology* by Ernst Haeckel in 1866 (Haeckel 1866). The definition has been considerably refined since 1866 by adding methodologies and formulating statistical measures to capture shifts in communities or to describe in an abstract way the richness of species, as Edward H. Simpson did in 1949 with the eponymous *Simpson index* (Simpson 1949), that represents the probability of randomly selecting two individuals of the same species from a habitat. The distinction between the different localization scopes of habitats and the examined species richness by means of scale, was ordered by Robert H. Whittaker into the first commonly accepted hierarchy in α-, β-, and γ-diversity in 1960 (Fisher et al. 1943; R. H. Whittaker 1960) and crucially refined by Martin L. Cody in 1975 (Cody 1975). Since Whittaker's proposal of his diversity hierarchy a more common term has been used in the natural sciences, namely the study of (bio)diversity. However, diversity is not limited to the investigation of natural habitats by measuring total organism abundances and species richness, because nowadays it is also possible to determine genetic diversity and thus the functional potential of a habitat depending on the community inhabiting it. Thus it can be said that the theories and experimental strategies of diversity are still evolving in ecology (R. J. Whittaker et al. 2001) and thriving in microbiology with the help of bioinformatics, namely here computational metagenomics.

Metagenomics, the analysis of a biotope's genome, is a conflation of the terms *meta-analysis*, taken from statistics and *genomics*. Whereas genome sequencing as isolates was limited to single shotgun-library Sanger sequencing (Sanger and Coulson 1975; Sanger, Nicklen et al. 1977), second generation sequencing, such as Roche 454 (Margulies et al. 2005) and Illumina GAIIX (Imelfort et al. 2009), enabled the sequencing of multiple organisms in a single metagenome shotgun sequencing run (untargeted sequencing), boosting the generation of genomic sequences to be analyzed and promoting the shift from single sequencing genomics to community-based metagenomics (Wooley et al. 2010). Nowadays, massive multi-parallel next generation sequencing (NGS) techniques, often described as third generation NGS, (Illumina HiSeqXTen (Telenti et al. 2016)) and real-time extra long read sequencing (Oxford Nanopore (Edwards et al. 2016), PacBio SMRT (Huo et al. 2015)) are driving the move away from tractable computational analysis of generated sequences (Z. D. Stephens et al. 2015). For marker gene-based community profiling, 16s rRNA gene sequencing (Yarza et al. 2014; Srinivasan et al. 2015) (amplicon sequencing) is the means of choice, where a set of variable regions within the 16s rRNA gene is amplified and further sequenced. The nine variable regions are highly conserved on species level and can thus be used for taxonomic classification (Amann et al. 1995). Metaganomic studies often involve techniques from other omic-related research fields and combine these, such as the analysis of the transcriptome of microbial community (metatranscriptomics) through RNA-sequencing (Bashiardes et al.

2016; Bikel et al. 2015) or the analysis of the metabolites (metabolomics) with techniques such as mass spectrometry or nuclear magnetic resonance spectroscopy (Johnson et al. 2016).

Through metagenomics it is not only possible to characterize microbial communities and detect relationships between the inhabiting organisms living within a habitat, but also so far unknown and non-cultivable microbial organisms can be directly identified and further described. These non-cultivable organisms form the *microbial dark matter* (Rondon et al. 2000; Baker et al. 2013), to which approximately 99% of all bacteria and archaea belong, so their poorly described taxonomical assignment has to be investigated. These organisms are non-cultivable, because their optimal growth conditions and required nutrients are unknown. Exploration of this dark matter is a promising research field, helping to understanding the influence of microbial communities on the environment in different habitats all around the world, whereby as a side effect the microbial taxonomic tree will be restructured (Hugenholtz et al. 2016) as new taxa are identified. To discover new so far non-cultivable organisms, metagenomic shotgun sequencing in combination with 16s rRNA sequencing can be applied, as performed by Banfield and colleagues in their thousand genomes study related to an aquifer system (Anantharaman et al. 2016), where they detected 47 new lineages on phylum level. Another preliminary approach prior to potential whole genome shotgun (WGS) sequencing of isolates was performed by Edwards and colleagues through setting up enrichment cultures of benzene-degrading organisms (F. Luo et al. 2016). This technique is assigned to the field of *Culturomics*, described later on.

Based on NGS, single cell sequencing became very popular for specifically sequencing the genome of a single organism from the microbial dark matter (Rinke, Schwientek et al. 2013; Hedlund et al. 2014). This sequencing technique requires some prior preparation, starting with cell sorting for picking the desired cell through to amplification of the DNA via multiple displacement amplification (Ishoey et al. 2008; Rinke, Lee et al. 2014). Recent studies shed some light into the microbial dark matter, like that of Bruno and colleagues, who investigated drinking water treatment plants (Bruno et al. 2017) and found bacteria belonging to the microbial dark matter in potable drinking water. Single cell sequencing is crucial for understanding the role of a bacterial or archaeal organism in a microbial community by coupling sequencing-based functional features and phenotypic annotations (Woyke et al. 2015), yielding a big picture in combination with additional analysis techniques at community level. Unfortunately, phenotypic annotation (traits) and metabolic features require considerable manual work, as described later on.

As stated earlier, the microbial dark matter consists of organisms that have so far been non-cultivable due to unknown optimal environmental growth conditions and lack of knowledge of their preferred nutrients. Therefore the research field of *Culturomics* is seeing a revival in microbiology for assessing the microbial dark matter (Kambouris et al. 2017; Dickson 2016). The strategy is comparable to a brute force search of optimal cultivation conditions, where a batch of different culture media are assessed as a test series under different environmental conditions (temperature, oxygen levels, humidity), as defined by Lagier and colleagues as a common protocol for clinical culturomics studies (Lagier, Edouard et al. 2015). Lagier also applied his protocol as a proof-of-concept study for cultivating organisms inhabiting the human gut (Lagier, Hugon et al. 2015).

2.1 Metagenome studies and practical implications for our every day life

One of the first notable studies that can be considered metagenomics on a large scale was the Sorcerer II global ocean sampling expedition initiated by Craig Venter in 2003, with a pilot sampling experiment at sea, starting in the Sargasso Sea (Venter et al. 2004) and continuing a year later for a two-year expedition in the Atlantic and Pacific Ocean (Rusch et al. 2007). The goal was to investigate the diversity of marine-inhabited microbial communities, but the subsequent analysis of the sequencing data was not limited to microbial planctonic organisms; viral sequences were also collected during the journey and analyzed later (Williamson et al. 2008). Since then, many more large-scale metagenome projects have been initiated and have produced a great deal of substantial data relevant to preventing and curing microbial-related diseases, to understanding the effects and influences of the environment in relation to climate change, how the benefit from such environmental studies for the generation of sustainable energy production and bioremediation of polluted areas, and also how take advantage of microbiota in molecular biology on an industrial scale.

The human body harbors trillions of microbiota on and in its skin, gut and even internal organs, where the total wet weight of these is of up to 2 kg (Van de Wiele et al. 2016). To chart the microbial communities in different locations, such as nasal and oral cavities, gastrointestinal and urogenital tract, and skin on inner elbow/behind the ear, the Human Microbiome Project (HMP) was initiated (Human Microbiome Jumpstart Reference Strains Consortium et al. 2010). For this purpose, more than 5700 samples were collected from 240+ human adults, to sequence and perform further functional as well as comparative analysis of the sampled microbial communities living in and on the human body. 16s rRNA-based sequencing was utilized to assess the microbial community structures residing in the abovementioned locations for each human subject, and another 560 samples were sequenced with whole metagenome shotgun (WMGS) sequencing. Another emphasis lay on the development of a reference set of 3000 isolate genomes, where so far 1500+ could be cultured and sequenced. These sequenced organisms are available via *IMG/HMP M*[1], an online resource for the integrated analysis of microbial genomes. A project solely focusing on the human gut is the Metagenomics of the human intestinal tract (MetaHIT) (Dusko Ehrlich et al. 2010; Qin et al. 2010), a collaborative research initiative mainly founded by the European Community. The digestion of nutrients is highly abundant from the intestinal microbiome of bacteria and fungi (Kirschner et al. 2015; Bode 2015) and also has a tremendous positive effect to the immune system (Round et al. 2009). But the microbiome is also related to cancer and has therefore been the subject of several clinical studies such as of correlation to liver cancer (Ezzat et al. 2014) or colorectal cancer (Vogtmann et al. 2016). The HMP and MetaHIT initiatives have substantially enabled these studies, confirming that metagenomics will play an essential role on future medical applications and diagnosis techniques (Mulcahy-O'Grady et al. 2016).

[1] *IMG/HMP M* – Integrated Microbial Genomes for the Human Microbiome Project. `https://img.jgi.doe.gov/cgi-bin/imgm_hmp/main.cgi`

Apart from HMP and MetaHIT for gathering human body-related microbiomes, data analysis and collation projects have been established, providing useful collections and tools for performing comparative metagenomic and metatranscriptomic studies, such as the Human Oral Microbiome Database (HOMP) (T. Chen et al. 2010; Dewhirst, T. Chen et al. 2010) and the Integrate Microbial Genomes & Metagenomes (*IMG/M*) (Markowitz et al. 2014; I.-M. A. Chen, Markowitz, Chu, Palaniappan et al. 2017). Such data collections are not limited to human body sites; a comparable study on felines has been published in which 246 full length 16s rRNA genes could be assembled from a total of 20 subjects, allowing specification of an oral reference taxonomy set for felines (Dewhirst, Klein et al. 2015). Clinical metagenomic studies also enable the definition of possible bio markers in cancer development, as performed in a collaboration with medical researchers from Heinrich-Heine Universität Düsseldorf, Birgit Henrich and some other colleagues (Henrich et al. 2014), where 16s rRNA community profiles of healthy subjects were compared to those from patients with fanconi anemia, a chromosal strand-break disease, with symptoms comparable to leukamia. Two of the fanconi anemia patients hosted a tumor in the oral cavity, on the left side of the tongue to be exact. The one available for swabbing fanconi anemia patients was swabbed on four locations (left side of tongue (tumor), right side of the tongue, and the two opposing sides on the gingiva), whereas the healthy subjects were sampled on the left side of the tongue. In this study it could be shown that *Mycoplasama salivarium* was the primary colonizer of both tumors with an abundance of $> 98\%$ compared to all other organisms found in this sample. Thus it could be said, that *M. salivarium* can be seen as a promising biomarker candidate for a developing oral cancer squamous cell cancer.

Apart from animals as host-related metagenomic studies, environmental field studies are essential to identify climate-driven changes to the environment. The different microbial consortium present on ice and the open sea in the arctic zone may be a key indicator of how and why these environments differ in their capacity to degrade hydrocarbon from the microbial perspective (Yergeau et al. 2017). Climate change may also influence airborne pathogen caused infections, and so screening of airborne microbial parameters is essential for the prevention of such outbreaks (Leuken et al. 2016). These studies can also be applied to livestock animals and foodborne pathogens (Hellberg et al. 2016). Another aspect is the emission of greenhouse gases, such as methane and carbon mono/dioxide, leading to a drastic acceleration of climate change. Cows as livestock animals produce tons of methane and so study of the functioning of their rumen-inhabited microbiomes as performed by Hess and colleagues is of great importance (Hess et al. 2011). Their discoveries offer significant insights into the potential production of biofuels (Parisutham et al. 2014) such as methane gas (biogas) produced in industrial scale anaerobic digesters by microbia (Stolze et al. 2016; Ortseifen et al. 2016), e. g. animal manure and sugar-rich carbohydrates (maize) act as substrates for bacterial and archaeal organisms, thereafter fermentation, primarily archaeal, produces methane. Ongoing studies are addressing how to improve the amount of produced biogas or even enable it (B. Yang et al. 2017; Lebuhn et al. 2014). As a side effect, nutrient rich fertilizer is produced that can be spread directly onto the field for further agricultural use. Other studies have pointed out that digesters can also be used for decontamination of animal wastes such as manure, but may also create new risks, when handling is not properly performed (Manyi-Loh et al. 2013). These include enrichment of antibiotics and metals (zinc, copper) used in nutrients for livestock, but possible gene-transfer of antibiotic resistance genes is also a major risk, especially when the fertilizer is spread on arable land. This specific issue was addressed in a student's work, to develop a pipeline for the detection of β-lactamase genes in metagenomic data, presented at the '3rd International Symposium on the environmental Dimension of Antibiotic Resistance' in 2015 (Osterholz et al. 2015).

Collection and analysis of environmental microbiomes is mostly driven by two initiatives, namely TerraGenome (Vogel et al. 2009) and the Earth Microbiome Project (Gilbert et al. 2014), to discover taxa belonging to the microbial dark matter and characterize the different environmental microbial communities by structure and also functionally by metabolic features. Based on the gathered data, studies related to agriculturally used soils have been enabled (Bevivino et al. 2014; Stempfhuber et al. 2015) and also the imminent effects of climate change like drought can be studied (Acosta-Martinez et al. 2014), especially related to maintaining supplies of staple foods such as wheat (Timmusk et al. 2014) or soybeans (Mendes et al. 2014) and basic nutrients for field plants in general (Pii et al. 2016; Stempfhuber et al. 2015). In biotechnology efficient production of enzymes and chemical molecules are important for further processing in pharmaceutical and food industry (Coughlan et al. 2015; Molinari 2010), but also the efficient catalysis of chemical reactions (Vandamme et al. 2005). Additives such as enzymes are used widely in nutrients utilized in animal feed industry and are thus of special interest of the industry (Choct 2006; Selle et al. 2007). Another application of metagenomics is the investigation of efficient bioremediation of polluted environments with the help of complex microbial communities and their metabolic capabilities. As mentioned earlier, antibiotics can be released into the wild via fertilizers from biogas producing digesters, but it has also been detected that certain microbia are capable of degrading veterinary antibiotics (Alexandrino et al. 2017) and even organic pollutants (Wang et al. 2017). Inorganic pollutants released into the wild from oil spills can be degraded by gamma-irradiated microbia (VanMensel et al. 2017). In addition, microbial organisms can be used to bioremediate highly toxic chemicals (Prasad 2014; Shuib et al. 2016) and also biological weapons (Stuart et al. 2005).

2.2 Setup of a metagenomic experiment

As with most explorative biological experiments, metagenomic experiments begin with sample picking of an environmental habitat and collection of data to describe the samples, such as of environmental conditions (temperature, humidity, pH), descriptions of the host-substrate under examination (type of substrate such as skin or soil, geographic location, altitude) and, if applicable in the case of an engineered environment, current process parameters (flow control, concentration and type of nutrients/metabolites). This sample description data is not used directly during the metagenomic analysis and is therefore secondary information, also called metadata. Metadata becomes important when comparing several metagenomic datasets with each other or drawing conclusions from observations from the analysis of the metabolic features or the metatranscriptome. Specifications have been drawn up, for instance by the Genome Standards Consortium (GSC) (Field et al. 2014) as the Minimum Information About a Marker Gene Sequence (MIMARKS) and Minimum Information about any (x) sequence (MIxS) (Yilmaz, Kottmann et al. 2011) and have been adapted by several sequence repositories such as the European Nucleotide

Archive (ENA) (Leinonen et al. 2011; Toribio et al. 2017) and NCBI's BioSample (Federhen et al. 2014). Metadata annotations and recommended fields are available for genomic and metagenomic data, where for the latter specially designed metadata catalogues (Kopf et al. 2015; Kyrpides, Woyke et al. 2014) have been implemented and also extended with categorized data structures such as Environmental Ontology (ENVO) (Buttigieg, Morrison et al. 2013). Only through these data acquisition procedures is further replicability and comparability guaranteed, and is as a consequence a necessity for good scientific practice (Knight et al. 2012). A more detailed explanation of metadata in general and the captured metadata in genomics and metagenomics is given in Chapter '3 – Materials and Methods' from page 27 onwards.

After collection of samples the environmental sample is processed in the lab for purification and sequencing library preparation, followed by sequencing on a sequencing machine and thereafter bioinformatic processing starts. In the case of a WMGS sequencing experiment, community profiling via an additional 16s rRNA gene-based sequencing is helpful for estimating the taxonomical composition, to derive the diversity of the environmental sample, and how abundant the expected taxa in the WMGS sequencing will be. Also, biological and technical replicates are a good control measure for ensuring the integrity of the sequencing results, whereby it is easier to identify erroneous results due to contamination occurring during sample picking and further preparation in the lab (Lupan et al. 2013; Mukherjee, Huntemann et al. 2015; Brooks et al. 2015). In the following bioinformatic processing of marker gene-based sequencing experiments and WMGS sequencing experiments are further described and tools for performing the described tasks are stated where applicable. It is assumed that raw sequencing reads have been produced on current Illumina sequencing machines.

Processing of marker gene-based sequencing experiments First, quality control (QC) is performed on the raw sequencing reads, e. g. checking for presence and possible trimming of primer and linker sequences utilizing trimmomatic (Bolger et al. 2014). This tool is also capable of checking for sequencing quality based on Phred scores and filtering accordingly. For visual inspection of sequencing quality, fastqc (Andrews 2010) can be used. In the case of overlapping sequencing reads (paired-end), these have to be merged. Trimmomatic is capable of this, but flash (Magoč et al. 2011) is an alternative tool specifically designed for merging paired-end reads incorporating read correction (nucleotides/length of merged read pairs) and Phred-based quality filtering. Community profiling is performed by clustering the reads of all sequenced input samples into operational taxonomic units (OTU) with a minimal sequence identity of 97%, where an OTU is defined as a group of taxonomically related organisms, or originally defined as the things being studied (Sokal et al. 1963; Sneath et al. 1973). The threshold of 97% identity is said to cluster those sequences together (Hughes et al. 2001) that belong to the same taxon at species-level, but this definition is dubious, because some taxa are separable on a sequence identity of 95%, whereas others can only be divided into OTUs at *species* level at a higher threshold of 99% (Koeppel et al. 2013), meaning that an artificial split is introduced (95%) or that an OTU is defining a grouping at *genus* level (99%). After clustering into OTUs, the sequence amount of contributing samples can be computed, resulting in an OTU profile specific for each input sample. Taxonomic assignment of OTUs is accomplished via alignment of an OTU-representative sequence against a marker gene reference database. In the case of a microbial community consisting of bacteria and archaea, suitable reference databases are GreenGenes (DeSantis, Hugenholtz et al. 2006; DeSantis, Dubosarskiy et al. 2003) and SILVA (Yilmaz, Parfrey et al. 2014; Quast et al. 2013), whereby the former is a hand curated but slightly outdated database of 16s rRNA reference gene sequences (M. W. Gray et al. 1984), and the latter is an automatically quality controlled 16s rRNA reference database that contains an almost up-to-date collection of known 16s rRNA genes. For a fungal marker gene-based experiment the set of sequenced marker sequences is changed, whereby the internal transcribed spacer (ITS) region is species specific (Schoch et al. 2012). The UNITE database acts in this case as the provider of reference data sets to compare to (Kõljalg et al. 2013).

There are a lot of pipelines and tool packages to perform the required tasks of OTU clustering and taxonomic assignment of OTU representative sequences, such as mothur (Schloss et al. 2009) and QIIME (Caporaso et al. 2010) to mention two of the most popular. In the following, the three basic OTU clustering approaches of QIIME will be partially outlined. QIIME offers three different OTU clustering approaches: de novo, closed-reference, and open-reference. The *de novo* approach attempts to cluster all input sequences into OTUs resulting in quantities of possible OTUs with a large amount of singletons containing only a single sequence. The opposite approach is the *closed-reference* clustering approach, where only those sequences that have a hit in the reference database are clustered to OTUs. A composition of both approaches is the *open-reference* OTU clustering pipeline, where closed-reference clustering is performed first and sequences without a perfect hit against the reference database are subsequently clustered. Further clustering is divided into three phases, whereby in the first phase the remaining sequences are subsampled and clustered de novo to form a new reference data set, following a closed reference OTU clustering of the left-over sequences against the new reference data set, to form high quality de novo-clustered OTUs. In the last and optional third phase, the remaining non-alignable sequences are simply clustered de novo. The open-reference approach is preferred over the two other approaches because the result combines the benefits of the two other basic approaches yielding an optimal resolution of OTUs among different taxonomic levels.

After OTU clustering and taxonomic assignment of the OTUs, taxonomic trees with abundances can be drawn per sample to investigate the distribution of organisms among all input samples to identify a core genome or to detect sample specific organisms. In addition, comparative statistical analyses of diversity can be performed, likewise for a set of samples representing time series or the comparison of control groups against groups of samples as the object under investigation. The limitation of a marker gene-based study is that it is only capable of describing the current composition of a microbial population, due to its descriptive way of observing the community structure. It is not capable of performing deeper functional investigations such as metabolic capabilities or the detection of mutations in an organism. Nevertheless, in combination with a WMGS experiment it gives a researcher clues about the expected composition of the sampled environment.

The workflow described here is implemented as a pyFlow[2] performing all steps starting from QC over read merging and automatic QIIME invocation locally on a PC or in parallel executed in a compute cluster environment: `https://github.com/mrumming/PyFlows`.

[2] *pyFlow* – Python workflow engine. `http://illumina.github.io/pyflow/`

Processing of whole metagenome shotgun sequencing experiments In the following an exemplary workflow of a WMGS experiment is described, but not in such detail as the previous part. The focus will lie on the possibilities of reconstructing genomes contained in the sequence sample as well as the analysis of the functional capabilities encoded in the metagenomic sequences. The analysis of WMGS sequencing starts, like marker gene-based experiments, with the QC. In the case of paired-end reads, Trimmomatic will do the work of dismissing low-quality reads and splitting the input reads as paired and unpaired sequence libraries, which can be further processed by a sequence assembler such as megahit (Li, C.-M. Liu et al. 2015; Li, R. Luo et al. 2016). During the assembly process, the assembler tries to reconstruct the genomic content based on the input sequences, resulting in a collection of contigs (partial assemblies of a genome). Based on these contigs, further analyses are applied for gene calling (prodigal (Hyatt et al. 2010)), homologue search (diamond (Buchfink et al. 2015)) of predicted genes/proteins against a reference database (NCBI nr database (NCBI Resource Coordinators 2017)), and computation of the taxonomical assignment of the contigs via the lowest common ancestor (LCA) approach (MEGAN6 (Huson et al. 2007)) based on the taxonomies of the genes that have been called on the contigs. For estimating the abundance of a contig, single read mapping onto the assembled contig can be performed (bbmap (DOE Joint Genome Institute 2014), FR-Hit (Niu et al. 2011)). The combination of the LCAs and the mapped reads per contig in relation to the total read count allows construction of an abundance profile with taxonomic assignments that is less precise than a marker gene-based sequencing one.

Further on, genome binning can be performed to combine assembled contigs into genomic bins representing a specific taxa (metabat (Kang et al. 2015)). Of course, such an initially artificial genome could be partial, not representing a 100% complete genome, or include sequencing reads originating from other foreign taxa, and so contamination detection and completeness (CheckM (Parks et al. 2015)) checks have to performed. If a genome bin meets the requirement of a high quality genome bin (completeness $> 80\%$, contamination $< 5\% - 10\%$), a subsequent assembly of the sequencing reads contributing to the contigs on which the genome bin has been constructed can be performed for refinement (SPAdes (Bankevich et al. 2012)). Genome binning can yield high quality genomic reconstructions out of the metagenome, not fully comparable to those generated by isolate genomes, but nevertheless offers a great opportunity to investigate the microbial dark matter.

For functional profiling of known genomes, assembled genome bins, as well as contigs and their enzymes contained on the metagenome, several reference database and data sets describing metabolic functional units and pathways can be utilized. The first step has already been performed through prediction of genes on contigs and homology search of the encoded proteins against a reference database. The results of the alignment against the NCBI nr database for each gene give an informative view about the contig and its genes. For computational and analytical reasons, the scope has to be adjusted to the style of analyses that are to be performed, thus analysis-specific annotations are needed, using a controlled vocabulary.

The protein sequences can be searched against generalizing reference databases that group proteins together either by function or sequence similarity. Clusters of Orthologous Groups (COG) database (Tatusov et al. 1997; Galperin et al. 2015) categorizes proteins by function, e. g. metabolism pathways involved, cellular processes or structures, or response mechanisms to environmental stimuli. The functional grouping of protein domains based on their sequence is performed within the Pfam database (Finn, Coggill et al. 2016). Both databases are suitable for analyzing the functional content in a broader perspective, and can be used for mapping onto pathway databases or onto terms of a domain-specific ontology. The latter would be a translation to the terms of the Gene Ontology (GO) (Ashburner et al. 2000), enabling precise semantic mapping of proteins to functions and concepts modelled in the ontology. The Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa 1996; Kanehisa et al. 2017) and MetaCyc (Caspi et al. 2014) database provide a rich set of metabolic pathways, specific on primary and secondary metabolism with contained genes, reactions, responsible enzymes for reactions, and substrates as well as produced metabolites. As a common practice, the biological researcher is interested in certain metabolic pathways, so prior knowledge about potential metabolic reactions is often necessary, coming from observations of the environment. But such reactions can be performed by a set of pathways, so the mapping of annotated genes onto a given set of pathways and further inspection of the expected coverage is an elaborate and computing-intensive task. After mapping, it is necessary to identify which taxa are responsible for certain reactions on metagenomic scale in a pathway, or which enzymes in a reaction cycle are over- or underexpressed. Often the desired coverage level is not achievable when the set of taxa mapped onto a pathway is restricted, and so screening of alternative pathways performing the reaction of interest is needed. Special purpose database are focused on certain enzymatic families or a certain catalyzed reaction performed by enzymes and can also be used for functional profiling. Examples are dbCAN (Yin et al. 2012) as a comprehensive catalogue of carbohydrate-active enzymes, ARDB (B. Liu et al. 2009) collects information and protein models for antibiotic resistance genes, and CyanoLyase (Bretaudeau et al. 2013) collects information about phycobilin lyases that play a major role in light-harvesting systems of cyanobacteria and red algae.

2.3 Finding the missing links

One major aspect of metagenomics is its ability to grasp and explore the functional and metabolic features of a community inhabiting an environment, to understand how the organisms are organized and also to evaluate how the findings may be compared to other studies or even applied in real-world applications such as bioremediation or clinical diagnosis procedures. In addition, it is possible to analyze non-cultivable organisms to explore the microbial dark matter. The ability to reconstruct complete genomes is a culture-free technique, by using the sequencing reads assembled as contigs and binning them. The question now is, how to consider the functional profile and environmental cultivation conditions of a non-cultivable organism represented as a genome bin. To characterize a bin, the process of functional profiling takes place, involving labor-intensive manual screening of metabolic pathway coverage in a step-by-step manner. This process is not limited to artificially assembled genomes: isolate genomes and single cell assemblages can also benefit from this approach.

To solve this issue and speed up the process of functional profiling, the given data base is used to translate the information about predicted genes and their encoded protein sequences to a controlled vocabulary and to use these features as input of a previously trained supervised machine learning classification model. A classification machine learner is capable of detecting hidden patterns in a data space (features) and assigning a class label as a prediction to a hitherto unseen input sample. The constraint of *supervised learning* means that the machine learner is modelled on a given set of training samples for which the real classification is known. In the case of a microbial organism a classification label would be a set of evaluable concrete instances of microbial traits and observable phenotypes, such as temperature range for optimal growth, the shapes of a cell, or encoded metabolic pathways. This kind of metadata is given for a relatively small set of bacterial and archaeal organisms and is available through the *IMG/M* system. The controlled vocabulary of features on which to train must fulfill the prerequisite of being at the right level of granularity (level of information), so as to be neither too overgeneralizing nor too specific. For representing the right level of information, families of functional protein domains would satisfy this need, such as organism Pfam abundance profiles. They are also beneficial because the computational effort is minimal. *PhenoPointer* implements this procedure for a set of microbial traits and observable phenotypes and is introduced in Part II of this thesis. For prediction of functional capabilities and observable phenotypes of a novel organism, specialized black box classification models are often trained for a single prediction purpose. One tool to compare to would be Traitar by Weimann and colleagues (Anantharaman et al. 2016), where trait specific support vector machines are trained and used for classification. This tool is used as a competitive tool for comparison of classification performance. It must be also stated that machine learning methods are not limited to classification purposes and are thus widely used in applications focused on metagenomics for contamination detection (ACDC (Lux et al. 2016)), genome binning (MetaBAT (Kang et al. 2015), PhyloPythia (McHardy et al. 2006)), or placement of sequences in a phylogenetic tree (pplacer (Matsen et al. 2010)).

Besides a hypothesis-driven approach, the remaining question is: Which other metabolic features are hidden in the metagenome, and which of these are worth looking at in detail? One solution would be to perform a brute force approach and try to map all predicted proteins onto pathways and filter the results afterwards by taxa. This is a very computing- and manual elaborate-intensive task with no guarantee of success. Another solution would be to look at the whole metagenome at a higher abstraction level of traits or observable phenotypes – looking at community profiles per sample combined with descriptive metadata. An implementation of such an approach is *MVIZ*, that takes community profiles as input and enriches these with metadata related to microbial traits or observable phenotypes that summarize the characteristics of a microbial community in an intuitively accessible way. By looking at the metadata-enriched community profiles the researcher is guided to formulate new hypotheses and is able to decide which hitherto unknown potential pathways to examine more deeply. The decision-making process would thus be accelerated and workload minimized. With this procedure of observing the potential on a trait/ phenotypic level, errors in the metadata annotations of samples are detectable. The same holds for 16s rRNA-based community profiles, which are a solely descriptive approach, but their field of application can be extended through enrichment with organism-specific metadata. For valuable interpretations, the metadata should cover characteristics related to concrete states of the microbial community such as temperature range, correlated diseases, or primary energy and carbon source. PICRUST does this by matching a 16s rRNA sequence against a taxonomic reference database and performing an ancestral state reconstruction to infer functional capabilities (Langille et al. 2013). Since the significance of reliable predictions on a 16s rRNA basis is highly questionable, the process can be accelerated by direct mapping of taxons from the input profile onto known taxa in a reference database by their identifier as performed in *MVIZ*. *MVIZ* is introduced in Part III of this thesis.

Part I

MetaStone – Foundation for Metagenomic Storage of novel entities

3 Materials and Methods

When performing studies in the field of computational biology researchers are primarily interested in the results and the derived outcome of certain tools, developed for a particular domain of application. This kind of data is called primary data. But what is often neglected is the vast amount of secondary information created during the runtime of such bioinformatics tools, the metadata.

Metadata is generated alongside primary data and often contains useful additional information about the primary data itself. Metadata can be defined as (Merriam-Webster, Incorporated 2017):

> « [Metadata is] data that provides information about other data»

With increasing available sequencing data as seen in the past decade (Stevens 2013), metadata is important not only for scientific data managing but also for analyzing such huge amounts of information, as described by Gray and colleagues (J. Gray et al. 2005). As an information scientist, Pomerantz conducts intensive research in the field of information processing and distinguishes between different categories of metadata (Pomerantz 2015):

Descriptive Metadata Details about underlying information of primary data, i. e. information about certain characteristics or creation details of primary data, which is helpful for result interpretation or in technical applications for maintenance scheduling

Administrative Metadata Secondary information helping administrative tasks, i. e. as status reports during bioinformatics pipeline runs or in technical applications for access control and status of resources

Structural Metadata Description/Specification of a container format for structuring data, i. e. XHTML format specification (World Wide Web Consortium 2000)

In *MetaStone*, the code and data foundation for *PhenoPointer* and *MVIZ*, descriptive metadata is used to describe and characterize bacterial and archaeal genomes as secondary information. One type of metadata is the categorical mapping to certain environmental states describing preferred growth conditions such as optimal temperature range and salinity. Another categorical mapping specifies developed phenotypes regarding metabolic features as carbon and energy source for bacterial growth, the correlation to host-related diseases, or more general characteristics, e. g. Gram staining, sporulation, and cell shape. In addition, predicted protein domains (Finn, Coggill et al. 2016) per genome and their abundance is included in *MetaStone*.

For this work, it is crucial to emphasize that in the scope of *PhenoPointer* and *MVIZ* a shift of view on how to look at the data takes place; a shift from secondary to primary information occurs, meaning that metadata is no longer merely secondary information.

Metadata serves in *PhenoPointer* as primary data for training machine learners on Pfam abundance profiles to predict the mentioned genomic characteristics. In the case of *MVIZ* this takes place in a similar way, where the genomic characteristics are used to generate metadata-enriched metagenomic community profiles as easy feasible comprehensive visualizations.

3.1 Metadata in metagenomics

Especially in the field of metagenomics, additional information about sampling projects and their contained samples are a necessity, because without any underlying data about the samples no comparisons can reasonably be made to other metagenomes or even to samples of the same metagenome (National Research Council US Committee 2007; Knight et al. 2012): Metadata are the foundation of (comparative) metagenomics. The captured metadata of a metagenome might describe e. g., the source of the biological sample and the conditions during sample picking.

Different standards have been developed to collect metadata in a standardized format for free text and also for using controlled vocabularies to guarantee machine processability (Field et al. 2014), and genomic sequence databases often require metadata annotation before publishing data sets. Three most common standards or recommendations are the Environment Ontology (EnvO) (Buttigieg, Morrison et al. 2013; Buttigieg, Pafilis et al. 2016), used in the European Nucleotide Archive, BioSample/BioProject at the NCBI (Federhen et al. 2014), and the minimum information about a marker gene sequence (MIMARKS)/ minimum information about any sequence (MIxS) specifications by the Genomic Standards Consortium (GSC) (Yilmaz, Kottmann et al. 2011; Field et al. 2014).

Enabling comparative metagenomic studies requires a centralized curated metadata repository with high quality and accurate metadata. In the best case, a repository should not only administer its own sets of metadata and submitted ones, but should also incorporate data sets from external sources. The metadata should at least be checked for inconsistency and mappable attributes should preferably be translated and transformed into a controlled vocabulary for computational purposes. One example of such a repository satisfying the mentioned aspects is the Genomes OnLine Database (*GOLD*) (Kyrpides 1999; Mukherjee, Stamatis et al. 2017), developed and hosted at the DOE Joint Genome Institute (JGI). *GOLD* contains data about sequencing projects and correlated metadata concerning ecosystem, habitat, or place of isolation on metagenomic level and organism-specific data fields as lineage (Federhen 2012), phenotypes, or biotic relationship to other genomes. The metadata is obtained from several sources such as user-submitted, NCBI's BioProject and BioSample system, and in-house sequencing projects processed at the JGI itself. To guarantee accuracy and consistency of the imported data, the input is checked manually and semi-automatically before being imported and made publicly available. Some of the metadata is categorized in a taxonomy-like data structure to control level of granularity, as in modelling the ecosystem for aquatic habitats: coastal water, open sea, coupled with different depths as surface water, deep sea, or hydrothermal vents.

GOLD, with its metadata about sequencing project, metagenomic and genomic samples is the primary data service provider for the Integrated Microbial Genome with Microbiome Sample system (*IMG/M*), which, like *GOLD*, is also hosted and developed at the JGI. The *IMG/M* system is the interconnection between metadata with biological DNA/RNA sequences, results of analysis pipelines and annotations of archaeal, bacterial, eukaryotic and viral genomes from cultured systems, single cell genomes (SCG) and genomes from metagenomes (I.-M. A. Chen, Markowitz, Chu, Palaniappan et al. 2017; Markowitz et al. 2014). All imported sequences run through the IMG annotation pipeline before being published (Huntemann et al. 2015) and the results of each tool performed during the pipeline run are available through the *IMG/M* data warehouse. This pipeline includes such steps as feature detection of CRISPR elements (Bland et al. 2007), gene prediction utilizing Prodigal (Hyatt et al. 2010) and further processing with HMMER (Finn, Clements et al. 2015) for protein domain comparison to Pfam-A v.29 (Finn, Coggill et al. 2016). As of 23 April 2017, functionally annotated and including metadata from *GOLD*, the following numbers of genome and metagenomes are available through *IMG/M*:

Bacteria	51415
Archaea	1199
Eukarya	222
Plasmids	1193
Viruses	6230
Genome Fragments	1192
Metagenome	5880
Cell Enrichments	507
Single Particle Sorts	1
Metatranscriptome	1446

3.2 Common base of input data

IMG/M with its incorporated metadata available for bacterial and archaeal genomes, is the primary data source for *MetaStone* and the dependent applications *PhenoPointer* and *MVIZ*. The data warehouse was chosen because it contains a vast variety of quality controlled data sets coupled with high quality metadata from *GOLD*, which come with a controlled vocabulary for certain metadata categories.

Metadata information about bacteria and archaea has been imported into *MetaStone*. The time point for data extraction from *IMG/M*, which is used as the data foundation for *PhenoPointer*, was March 2016. The data available in the current release of *MetaStone* is taken from *IMG/M* as of 24 April 2017 and is used without restrictions. In the case of *PhenoPointer*, the newly uploaded genomes are used as the final test set for measuring the predictive performance of the trained machine learners. As further data Pfam abundance profiles have been extracted form *IMG/M* using the `Compare Genomes -> Abundance Profiles` functionality for every genome and stored in *MetaStone* for use within *PhenoPointer*. Table 3.1 shows the number of entries for both time points of data processed and imported into *MetaStone*.

Domain	**March 2016**	**April 2017**
Bacteria	34188	51415
Archaea	675	1199
In total:	34863	52614

Table 3.1: ***IMG/M* data sets imported into *MetaStone*** All data sets include metatdata and Pfam abundance profiles for each genome.

The chosen metadata categories reflect the biological truth of the characterized genomes at the right level of granularity for computational and human perceptional reason, in order not to over-generalize or not be too fine to create a particular instance of every metadata entry per genome. The same holds for the more technical and sample related metadata categories. Table 3.2 gives an overview of the selected *GOLD* metadata categories. The summary highlights those categories, which are used as phenotypical metadata used for prediction matters in *PhenoPointer*. It also marks metadata categories which come with a controlled vocabulary or which needed some manual normalization of values before being imported into the database. For categories with predictive means, the classification problem (Hastie et al. 2009) is given as:

binary Classification of a data set into two *distinct groups*, i. e. labeling as True/ False or positive/ negative

multiclass Classification of a data set into *one* particular class out of a set of classes

multilabel Classification of a data set into *one or more* classes out of a set of classes

3.3 Common base of code, distinct application focus

As on the data side, *PhenoPointer* and *MVIZ* do also rely on the same codebase as *MetaStone* implements the data back end system. *MetaStone* is a python application taking advantage of the pythonic Django web framework (Django Software Foundation 2016) for persistence and user interaction via a CLI (Command Line Interface). *PhenoPointer* uses the genomic Pfam abundance profiles as features for training machine learners to predict organism-related phenotypes given as metadata annotations from *GOLD*. *MVIZ* visualizes user-uploaded metagenomic community profiles previously enriched with *GOLD* metadata within *MetaStone* and takes advantage of predicted organism metadata as an optional visualization feature.

A data keeping backbone must fulfill basic demands e. g., fast execution time of data retrieval, pipeline invocation, and a simplistic user-centric client. In addition, new data must be loaded, processed and stored in the database in the manner of the established ETL (Extract, Transform, Load) approach. For this purpose, the pythonic web framework Django was chosen with its ORM (Object Related Mapping) capabilities to not only handle data management but also to create the data representing models and setting up database structure automatically. For direct user interaction, a CLI with the support of self-defined commands is included. Because Django was

Metadata Category	Description	Used for ML	Problem Class	Manual Normalization	Controlled Vocabulary
Habitat	Unordered description of place of isolation				
Sample Body Site	Animals/Plants as host: sampling site				✓
Sample Body Subsite	Animals/Plants as host: sampling site, subcategorization				✓
Sequencing Method	Employed Sequencing platform (i. e. 454, illumina, PacBio)				
Status	Genome analysis status				✓
Type Strain	Status being a type strain				✓
Uncultured Type	Source of sequences (i. e. single cell, genome bin, synthetic)				✓
Ecosystem	1st level of categorization of sampling site as ecosystem				✓
Ecosystem Category	2nd level of categorization of sampling site as ecosystem				✓
Ecosystem Type	3rd level of categorization of sampling site as ecosystem				✓
Ecosystem Subtype	4th level of categorization of sampling site as ecosystem				✓
Relevance	Industrial/biological application scope of the genome				
Specific Ecosystem	Organism's Light adaptivity				✓
Biotic Relationships	Organism is free living or symbiotic	✓	binary	✓	✓
Cell Arrangement	Kinds of organization of cells in the biofilm	✓	multilabel		✓
Cell Shape	The cell shape of the organism	✓	multiclass	✓	✓
Diseases	Host-correlated or caused diseases by an organism infection	✓	multilabel		
Energy Source	Microbial metabolism classification of obtaining energy	✓	multilabel		✓
Gram Staining	Organism is Gram positive or Gram negative	✓	binary		✓
Metabolism	Genome-encoded metabolic features and substrate degradation	✓	multilabel		✓
Motility	Organism's motility capability	✓	multiclass	✓	✓
Oxygen Requirement	Type of ozygen requirement for growth	✓	multiclass		✓
Phenotype	Observable expressed phenotype and pathogenicity	✓	multilabel		✓
Salinity	Salinity of the environment for optimal growth	✓	multiclass	✓	✓
Sporulation	Organism is sporulating or not	✓	binary	✓	✓
Temperature Range	Optimal temperature range for growth	✓	multiclass	✓	✓

Table 3.2: ***GOLD* metadata categories used in *MetaStone*** Metadata categories are sorted into four groups, whereas the fourth one describes phenotypic and metabolic feature and is used in *Pheno Pointer* for predicting these on novel organisms for characterization. Problem class designates what kind of underlying classification problem is to be solved by the machine learner. Manual normalization is only applied to metadata categories used in *PhenoPointer*.

primarily designed as a web framework, it is possible for a developmental user to build a web UI for his or her own needs, but for this work the key aspect of functionality was the CLI.

Any Django application can be extended through its object-oriented implementation fashion, resulting in easy customization of the underlying data model automatically adjusting the backbone PostgreSQL (The PostgreSQL Global Development Group 2016) RDBMS[3], adding new functionalities to the backbone systems for data processing and extending the CLI by defining new workflows and pipeline calls.

The bundled software release comes with pre-trained ML classifiers within *PhenoPointer*, so for this software part no RDBMS is required, but for training new classifiers on an updated organism data set a database is essential. The same holds for *MVIZ*, since it is an HTML+JavaScript application, no direct database binding is needed as long as the input data fulfills the format specification as defined in Chapter 9.1 – 'JSON output format specification' on page 113. The *MetaStone* back end is needed to perform metadata-enrichment on input community profiles.

Detailed information about *PhenoPointer* can be found starting at page 45 and for *MVIZ* from page 109 onwards.

[3] *RDBMS* – Relational Database Management System

4 Implementation – MetaStone

This chapter will introduce the common code basis and further implementation details of it for both tools, *PhenoPointer* as well as *MVIZ*. First, the desired functionality of a data hosting backbone system will be illustrated and subsequently it will be explained how the requirements of such a data warehouse can be fulfilled.

The full source code is released under the BSD 3-clause license and can be accessed from GitHub `https://github.com/mrumming/MetaStonePhenoPointer/`.

4.1 Basic Django setup and project structure

A Django project consists of a basic setup directory and directories for any Django application belonging to the top-level project. Figure 4.1 on page 33 shows the general structure of *Meta Stone* and its accompanying applications *PhenoPointer* and *MVIZ*.

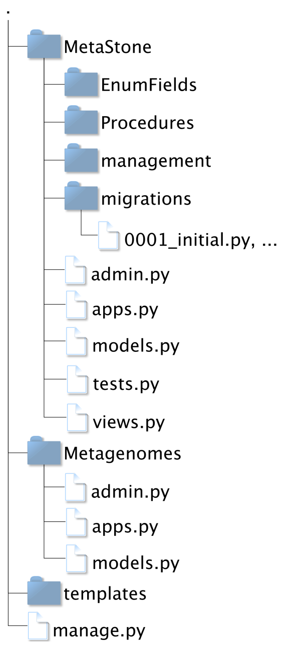


Figure 4.1: **Extended Django project directory** General structure of a Django project, with security and connectivity settings under *./Metagenomes/*, web UI definitions under *./templates/*, and application logic in *./MetaStone/*.

General configuration for connectivity with the database, security settings and probable UIs[4] is defined in the python files located under *./Metagenomes/*, while implementing web UI related files are located under *./templates/*.

The main *MetaStone* back end and the logics of its dependent *PhenoPointer* and *MVIZ* software are located under *./MetaStone/*. Methods for importing and exporting data from and to *Meta Stone* are located under *./MetaStone/Procedures/* and mappings of genomic metadata are stored under *./MetaStone/EnumFields/*.

The project contains a management python script *./manage.py* helpful for any developmental task such as initial database setup, altering and updating the database through the ORM back end system, starting the development web server, and setting up new applications. Additional CLI feature can be added to the program by extending the management console, located in *./MetaStone/management/*.

Database setup A PostgreSQL 9.5 RDBMS with its standard setup is sufficient for uses with Django. A database for *MetaStone* needs to be created and a database user must be assigned as the owner of this specific database for adding and altering tables, their contained data and PostgreSQL sequences. This user must not have super user privileges, because of the danger of possible security flaws. For the database in use the hstore[5] extension has to be installed. This extension is needed for retaining data representing key-value pairs, in this case storing Pfam domains and their abundance in genomes. Django uses its own PostgreSQL back end for accessing the database, so the psycopg2[6] python library must be installed.

4.2 The data model for persistence

The main data stored in *MetaStone* is of types of bacterial and archaeal genomes and their related metadata name, taxonomy, their encoded Pfams and abundances per protein family/domain, and most importantly correlated metadata of categorized values belonging to four top-level metadata groups:

Ecosystem detailed ecological information about the ecosystem from which the biological sample was picked

Sampling site categorical data about the picking site from which the environmental sample containing the genome was taken

Sequencing information about the sequencing method, genome assembly status, and possible type strain declaration

Species related detailed categorized data about expressed phenotypes and metabolic features

[4] *UI* – User interface

[5] *hstore* – `https://www.postgresql.org/docs/current/static/hstore.html`, accessed 07 April 2017

[6] *psycopg2* – `http://initd.org/psycopg/`, accessed 07 April 2017

The exported data from *IMG/M* contains this particular kind of data and needs to be modelled in *MetaStone* for further processing in *PhenoPointer* and *MVIZ*. The Django framework provides a dynamic database-abstraction API to access modelled objects and its fields for CRUD[7] operations. The ETL process is described in section 4.3 on page 38.

Divided into several files, the initial input data is packaged into three different categories of possible input file formats:

Genome cart Export of a genome cart from *IMG/M* including all available primary data fields such as `taxon_oid` or `Genome Name / Sample Name`, phylogeny related fields such as `Phylum`, `Class` or `Genus`, and metadata fields such as `Specific Ecosystem`, `Sporulation` or `Temperature Range`.

Pfam data Initial Pfam data with Pfam identifiers, a corresponding short name and a full name describing the protein family

Pfam to genome Export of all genomes and their contained Pfams including their abundances in the genome

Storing and processing a genome name and its phylogeny is straightforward, because their nomenclature is distinct and well-defined. However, associated secondary data of a genome needs special attention as the data space of such metadata can be very heterogeneous. This type of information needs to be represented as simplified abstract metadata per category for computational reasons, e.g. `Temperature Range` would be the category and valid entities would be `Mesophile`, `Hyperthermophile`, or `Psychrotolerant`, thus all possible entities of a category need to be modelled as well to gain a controlled vocabulary. In addition to the three mentioned formats, a fourth input file format must be stored in *MetaStone* for further processing in *MVIZ*, one which represents a community profile of a metagenome packed in the BIOM[8] (McDonald et al. 2012) format.

ORMs representing the mentioned primary and secondary data are modelled in Django through defining classes of type `django.db.models.Model` filled with class variables of type `django.db.models.fields.Field`. The principle of defining object relational mappings in Django is that a model class represents a database table, its class variables of type `Field` represent database table fields and an instance of a model represents an individual table record. In *MetaStone* this modelling is implemented in *./MetaStone/models.py.* Figure 4.2 shows the Django model dependency as a class diagram for the defined input data and further extensions for *Pheno Pointer* and *MVIZ*.

The model of type `Genome` acts as the main entity for all associated genomic metadata and is used as a part of a user-uploaded metagenome represented in `User_Metagenome_Sample_Genome`. The `taxon_oid` field of `Genome` is unique for every stored genome and acts as the primary key for all correlated models via a one-to-one relationship, except for inferred metadata and metagenomic samples, where many-to-one relationships are present. The taxonomical lineage is modelled in `GenomeLineage`, with fields for taxonomic levels starting at *Domain* over *Phylum*, *Class*, *Order*,

[7] *CRUD* – Create, read, update, delete. Basic functions of a persistent storage system.

[8] *BIOM* – Biological Observation Matrix. General purpose file format for storing biological samples by observation contingency tables.

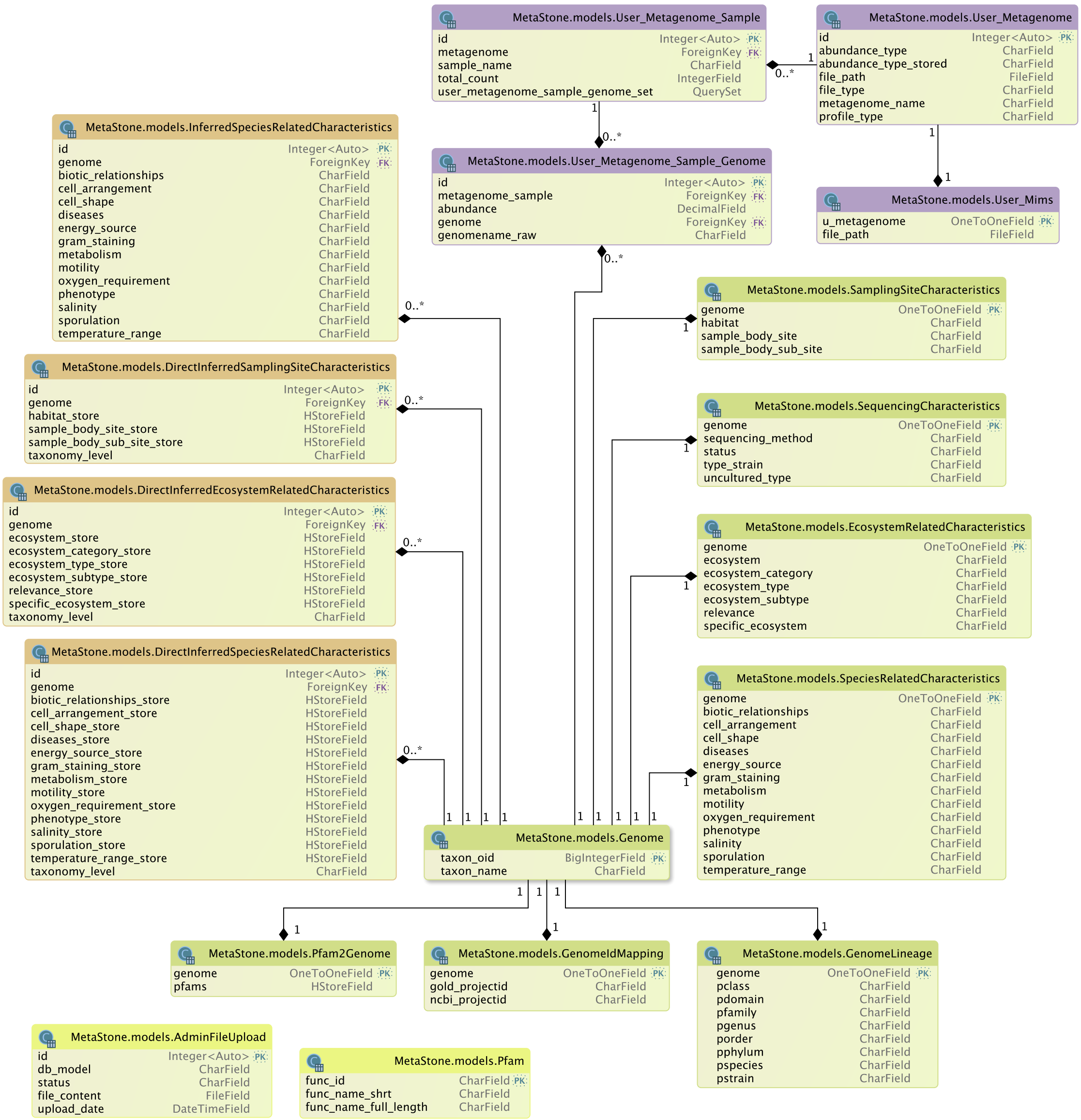

MetaStone.models.User_Metagenome_Sample
id Integer<Auto> PK
metagenome ForeignKey FK
sample_name CharField
total_count IntegerField
user_metagenome_sample_genome_set QuerySet
MetaStone.models.User_Metagenome
id Integer<Auto> PK
abundance_type CharField
abundance_type_stored CharField
file_path FileField
file_type CharField
metagenome_name CharField
profile_type CharField
MetaStone.models.User_Metagenome_Sample_Genome
id Integer<Auto> PK
metagenome_sample ForeignKey FK
abundance DecimalField
genome ForeignKey FK
genomename_raw CharField
MetaStone.models.User_Mims
u_metagenome OneToOneField PK
file_path FileField
MetaStone.models.InferredSpeciesRelatedCharacteristics
id Integer<Auto> PK
genome ForeignKey FK
biotic_relationships CharField
cell_arrangement CharField
cell_shape CharField
diseases CharField
energy_source CharField
gram_staining CharField
metabolism CharField
motility CharField
oxygen_requirement CharField
phenotype CharField
salinity CharField
sporulation CharField
temperature_range CharField
MetaStone.models.DirectInferredSamplingSiteCharacteristics
id Integer<Auto> PK
genome ForeignKey FK
habitat_store HStoreField
sample_body_site_store HStoreField
sample_body_sub_site_store HStoreField
taxonomy_level CharField
MetaStone.models.DirectInferredEcosystemRelatedCharacteristics
id Integer<Auto> PK
genome ForeignKey FK
ecosystem_store HStoreField
ecosystem_category_store HStoreField
ecosystem_type_store HStoreField
ecosystem_subtype_store HStoreField
relevance_store HStoreField
specific_ecosystem_store HStoreField
taxonomy_level CharField
MetaStone.models.DirectInferredSpeciesRelatedCharacteristics
id Integer<Auto> PK
genome ForeignKey FK
biotic_relationships_store HStoreField
cell_arrangement_store HStoreField
cell_shape_store HStoreField
diseases_store HStoreField
energy_source_store HStoreField
gram_staining_store HStoreField
metabolism_store HStoreField
motility_store HStoreField
oxygen_requirement_store HStoreField
phenotype_store HStoreField
salinity_store HStoreField
sporulation_store HStoreField
temperature_range_store HStoreField
taxonomy_level CharField
MetaStone.models.SamplingSiteCharacteristics
genome OneToOneField PK
habitat CharField
sample_body_site CharField
sample_body_sub_site CharField
MetaStone.models.SequencingCharacteristics
genome OneToOneField PK
sequencing_method CharField
status CharField
type_strain CharField
uncultured_type CharField
MetaStone.models.EcosystemRelatedCharacteristics
genome OneToOneField PK
ecosystem CharField
ecosystem_category CharField
ecosystem_type CharField
ecosystem_subtype CharField
relevance CharField
specific_ecosystem CharField
MetaStone.models.SpeciesRelatedCharacteristics
genome OneToOneField PK
biotic_relationships CharField
cell_arrangement CharField
cell_shape CharField
diseases CharField
energy_source CharField
gram_staining CharField
metabolism CharField
motility CharField
oxygen_requirement CharField
phenotype CharField
salinity CharField
sporulation CharField
temperature_range CharField
MetaStone.models.Genome
taxon_oid BigIntegerField PK
taxon_name CharField
MetaStone.models.Pfam2Genome
genome OneToOneField PK
pfams HStoreField
MetaStone.models.GenomeIdMapping
genome OneToOneField PK
gold_projectid CharField
ncbi_projectid CharField
MetaStone.models.GenomeLineage
genome OneToOneField PK
pclass CharField
pdomain CharField
pfamily CharField
pgenus CharField
porder CharField
pphylum CharField
pspecies CharField
pstrain CharField
MetaStone.models.AdminFileUpload
id Integer<Auto> PK
db_model CharField
status CharField
file_content FileField
upload_date DateTimeField
MetaStone.models.Pfam
func_id CharField PK
func_name_shrt CharField
func_name_full_length CharField
1
0..*
1
1
0..*
0..*
0..*
0..*
0..*
0..*
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1

Figure 4.2: **DB schematics of *MetaStone*** Yellow classes depict housekeeping database tables for internal usage. Green classes are in use for storage of genomes and their related metadata, whereas derived metadata generated using ML-methods or direct inference are shown in orange colour. User uploaded metagenomes/community profiles are stored in classes marked purple.

Family, *Genus*, and ending in *Species* with *Strain*, with the restriction for the latter one whether it is available or not, otherwise *Strain* stays empty. General information about protein families and domains are stored in the `Pfam` model, where the ID of a Pfam entry is represented as `func_id` and acts as the primary key. Additional fields of this model are a short name of an entry and a longer more descriptive one. Pfam motifs, the existence and abundance of Pfams in a particular genome, are stored in `Pfam2Genome` utilizing an `HStoreField` for every `Genome`, where `func_id` from the `Pfam` model acts as the key and the abundance of this distinct protein family/domain is stored as the value.

Genome related metadata are sorted into four groups as mentioned above in 4.2 on page 34. The partitioning follows a pre-grouping of metadata categories, which reflects the meaning of the grouped categories and their underlying labels[9].

1st level metadata For storing more general information about sampling location of a recorded genome and its sequencing, the classes `SamplingSiteCharacteristics` and `SequencingCharacteristics` are modelled, while a more detailed but categorical view about sampling location is stored in `EcosystemRelatedCharacteristics`, so that comparison to other records is more or less effortlessly possible. The model of type `SpeciesRelatedCharacteristics` records with its fields the metabolic features and phenotypes of a genome persistently and is the underlying data source for *PhenoPointer*, whereas *MVIZ* takes advantage of all four metadata groups. Each class consists of a one-to-one relation field to `Genome` and a `CharField` for every category named accordingly to its underlying representation of meaning. For modelling it is not important to take into account whether a category is limited to exactly one label or an undefined number of possible labels per entity; in the latter case the entry for a multilabel category would be a comma-separated list of possible labels in contrast to single valued categories.

2nd level/inferred metadata Equivalent models exist for direct inferred labels of metadata groups except for sequencing related information, because no reasonable meaningful data can be generated out of this metadata. The models are prefixed with `DirectInferred`. Directly inferred means that no ML-methods are applied to predict the labels, but the means of subsumed information per taxonomic level are computed. The sole exception is the categories of type `InferredSpeciesRelatedCharacteristics`, which are predicted with ML-methods used in *PhenoPointer*. Inferred metadata are useful for filling gaps in the data used in *MVIZ* for visualization purposes. Directly inferred metadata is stored in an `HStoreField`, where the key represents the label and the value reflects the support of the label from those genomes from whence this type of information was derived. The ML-predicted values for metabolic features and phenotypes are stored in the same fashion as primary metadata as `CharFields`, because the output of machine learners depends on the category and its labels on which the machine learner has been trained. Therefore, the result set's cardinality equals the one from the input set, so it is a single label, a set of labels, or nothing, if no reasonable prediction with sufficient support by the underlying ML-method could be made.

[9] *Label* – Possible value of a metadata category

For storing single or multiple end-user uploaded metagenomic community profiles, the model `User_Metagenome` is the initial database table where the parsed input is consistently serialized. This model contains a field for application-side file path of the input file, whereas all subsequent fields are filled automatically during the ETL-process. The field `abundance_type` contains information about the given values in the input, whether the values denote absolute or relative values. The type of stored values is specified in `abundance_type_stored`. For each sample encoded in the input, an instance of type `User_Metagenome_Sample` is created with a particular `sample_name` extracted from the input file. Each genome existing in *MetaStone* and mappable is stored in the metagenomic sample-genome bridging model `User_Metagenome_Sample_Genome` including the abundance inside the sample.

For administration purposes, the class `AdminFileUpload` models the requirements for file upload management and future importing of admin uploaded data. This model records the correlated Django models needing to be updated as work packages, for which the uploaded input file contains data. In addition, the model stores the ETL-status of the uploaded files as one of three types:

uploaded uploaded to the server, but left untouched

processing the ETL-process for specified Django models, stored in `db_model`, is in progress

finished the designated ETL-process is finished

4.3 Basic Workflows and Pipelines

Importing, exporting, and manipulation of data has to be performed in a well-defined and reproducible way. In *MetaStone* these tasks are distributed to different implementing scripts responsible for getting data into *MetaStone*, exporting metadata-enriched metagenome community profiles, performing cross-validation on machine learners or performing ML-based phenotyping. The only manual task that has to be performed beforehand is the manual normalization of six fields in the IMG exported metadata.

Although the high standard of quality controlled *GOLD* metadata, there still exist some erroneous data fields that have to be normalized manually before being imported into *MetaStone*. These data fields are described in Table 3.2 on page 31. The occurrence of errors in the quality controlled metadata is very sparse, i. e. single punctuation characters used as delimiters have been left over, or values have been introduced as singletons, which makes the underlying data basis in the context of further ML-processing more diverse without adding any useful information, and thus were identified as erroneous outliers or false data.

Biotic Relationships 12% of the total data is annotated with this metadata, where 2 entries are labeled as "Free-living, endophytic"; normalized to "Free living"

Cell Shape Punctuation character "," was left over at the end of one entry, so the character was deleted

Motility Three variations of a value denoting the same meaning existent: "Non motile", "Non-motile", "Nonmotile". Normalized to "Nonmotile"

Salinity Two erroneous entries removed: "2-10% NaCl", "Undefined"

Sporulation Two variations of a value denoting the same meaning existent:
"Non-sporulating", "Nonsporulating". Normalized to "Nonsporulating"

Temperature Range Removal of four erroneous entries:
"30 C", "25?C", "37?C", "to 93 C"

ETL-scripts and coupled enumerations of model fields for workflow control are shown in Figure 4.3. The main ETL-workflow is defined in *./MetaStone/Procedure/Helper/MaintenanceToolkit.py* as `process_all_imports(*args)`, which scans first all entities in the `AdminFileUpload` model for retrieving all models that need to be updated, and creates, depending on the defined models, a specific instance of `UploadModel`, on that finally the function `process()` is called. The latter is a generic function for all deriving ETL-tasks and is defined for previously mentioned database model representing genomes and 1st level metadata in a factory pattern manner. Implementing classes can be found in *./MetaStone/Procedures/Imports/*. This ETL-routine is controlled through tasks defined as work packages in *./MetaStone/EnumFields/DatabaseModels/Internal/DbModelMapping.py*, where a work package links to a list of models and also to the ID field in the input data, that is used as primary key for each entity.

Importing genomes and 1st level metadata Genome related metadata consists of extracted data about genomes retrieved from *IMG/M* as TSV[10] files. A record in such files is divided to various values, which are consistently ordered and their meaning assigned by the header of the input file. In an ETL-workflow, the values have to be mapped to certain fields in the models. For this purpose, each metadata category model for genomes, the genome itself and its lineage model has a key-value mapping included, to automatically assign the parsed values per record to its related field. An example of this mapping is given in Listing 4.1 on page 39 for the class `SpeciesRelatedCharacteristics.`, with the key representing fields in the input file and the value assigning the correspondent model field.

```
img_name = {"Biotic Relationships":"biotic_relationships", "Metabolism":"metabolism",
  "Cell Arrangement":"cell_arrangement", "Diseases":"diseases", "Motility":"motility",
  "Sporulation":"sporulation", "Gram Staining":"gram_staining", "Cell Shape":"cell_shape",
  "Salinity":"salinity", "Phenotype":"phenotype", "Energy Source":"energy_source",
  "Oxygen Requirement":"oxygen_requirement", "Temperature Range":"temperature_range"}
```

Listing 4.1: Mapping of IMG identifiers to persistent Django model fields for 1st level metadata of phenotypes and metabolic features.

The mappings are used to be evaluated during runtime of the `process()` function call in a `for` loop as `setattr(dbmodel, dbmodel.img_name[key], normalized_value)`, where the first argument is the model representation itself, the second argument denotes the table field of the model, and the third argument is the value to persist. Valid values for model fields are defined in *./MetaStone/EnumFields/Database*

[10] *TSV* – tab-separated values. Textual file format for storing data in a tabular format, separated with tab characters as delimiters.

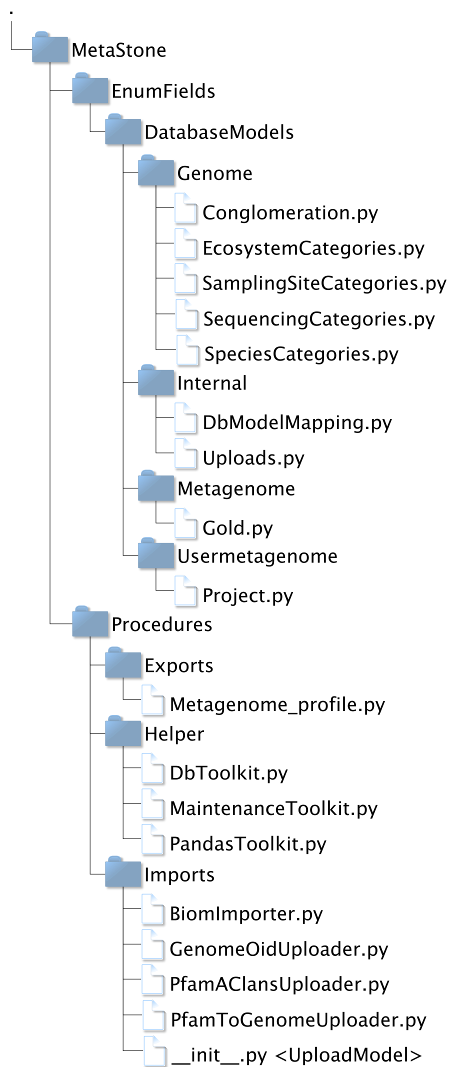


Figure 4.3: **ETL code and helper modules** ETL control modules are located under *./MetaStone/Procedures/Helper/*, where subsequent control modules are located under *./MetaStone/Procedures/Imports/*. Internal guidance enumerations for automatic recognition and control flow guidance are given the corresponding modules in *./MetaStone/EnumFields/DataBaseModels/* and related subdirectories.

*Models/Genome/*Categories.py* and automatically checked in Django through restricting entries of fields in the model itself: `motility = models.CharField (choices =SpeciesCategories.MOTILITY, default="Unknown", max_length=255)`. Listing 4.2 gives an example of valid choices definition for the given code snippet of the field `motility`.

```
MOTILITY = (("Unknown", "Unknown"), ("Nonmotile", "Nonmotile"),
  ("Chemotactic", "Chemotactic"), ("Motile", "Motile"))
```

Listing 4.2: Valid values for storing the 1st level metadata *Motility* in the model SpeciesRelatedCharacteristics.

Importing protein families/domains and genomic profiles Before importing Pfam profiles of genomes, the model `MetaStone.models.Pfam` has to be filled. For this purpose a list of all Pfam-A v.29[11] is loaded into the database utilizing *./MetaStone/Procedures/Imports/PfamAClansUploader.py* importer. After this prerequisite step, Pfam profiles per genomes can be imported by the responsible `UploadModel`, implemented as *./MetaStone/Procedures/Imports/PfamToGenomeUploader.py*. The underlying data structure is a python dictionary, which is directly stored in the model `MetaStone.models.Pfam2Genome` as an `HStoreField`, where the Pfam identifier is used as the key and the abundance in a particular genome is used as the value. The stored dictionary is a sparse one in the sense that clans are stored, iff the abundance is > 0.

4.4 Accessing entities and exporting data sets

To access data stored in *MetaStone*, no SQL[12] need be written because of the Django abstraction API available in every Django model. To load lazily a specific entity of a model, the `get()`-function has to be called up on the model manager to construct query set: `um = User_Metagenome.objects.get(pk=id_of_requested_entity)`. Values of database table fields can directly accessed through pythons attribute handling of instantiated objects: `um.profile_type`. For ease of access, records of genomes and their related phenotypes with Pfam abundances are stored internally during ML-processing as pandas[13] data frames. The method to call can be found in the module *./MetaStone/Procedures/MachineLearning/Tools/Transformer.py* as `genomes_characteristics_to_dict()`. For interactive direct data access with query sets, the Django shell can be used from command line: `./manage shell`

4.5 Software packaging and the CLI

Software developed with Django as the persistence framework comes with an extendible CLI to enable administrative and even potent end user functionality for standalone-like behavior as known from other command line tools.

[11] *Pfam-A v.29, released November 2015* – `ftp://ftp.ebi.ac.uk/pub/databases/Pfam/releases/Pfam29.0/Pfam-A.clans.tsv.gz`, accessed 03 April 2016

[12] *SQL* – Structured Query Language, programming language to access, alter and create data in a RDBMS

[13] *pandas* – Python Data Analysis Library. `http://pandas.pydata.org/`, Accessed: 20 April 2017

To extend to existing management console tasks, an extension as has to be made and stored as a python script like file. The extension is implemented as a class inherited from `django.core.management.base.BaseCommand` and stored under the *./MetaStone/management/* directory. The name of an implementing file is used as an alias of the newly available command for direct access via the management script. Implemented are:

pandas Export certain genome characteristics to a pickled pandas data frame

enrich Enrich a metagenomic community profile with metadata for use within *MVIZ*

xcross Perform stratified cross-validation

To run the `xcross` workflow and see, what the exact parameters are, a user must run the following command: `./manage help xcross`

For installation purposes of *MetaStone* and the two dependent tools *MVIZ* and *PhenoPointer*, a python3 virtual environment has to be setup and all libraries requirements as outlined installed in it.

- psycopg2 – 2.7.*
- scikit_learn – 0.18.1
- ipython – 5.*
- scipy – 0.19.0
- numpy – 1.12.*
- Django – 1.9.4-1.9.*
- pandas – 0.19.*
- biom-format – 2.1.*

This setup is sufficient for *PhenoPointer* with pre-trained machine learners, but for *MetaStone*, *MVIZ* and cross-validation workflow within *PhenoPointer* a suitable RDBMS, preferably PostgreSQL ≥ 9.2, is required.

Part II

PP – PhenoPointer

Little is known about the metabolic features and the encrypted phenotypes within novel bacterial and archaeal genomes defined as isolate genomes, detected in metagenomes as assembled genome bins or as a single cell in a single cell study respectively. These genomes are today mostly detected through Metagenomic SOP (standard of procedures) pipelines involving the assembly of NGS reads to reconstruct concealed organisms in fractions (contigs) or partially complete (genome bins), followed by gene calling and a homology search against a reference data base using blast, functional profiling of proteins as protein families (Pfam), or clusters of orthologous groups of proteins (COG), and subsequent reconstruction of metabolic pathways.

Deriving the phenotype of novel assembled partial or complete genomes is a so far labor-intensive manual task. *PhenoPointer*, a phenotyping tool for bacterial and archaeal genomes, solves this problem by taking Pfam functional annotation as input and reliably predicting the ecological and metabolic phenotypes as the output, thereby building the missing link from genotype to phenotype through ML-based classification models.

This part of the thesis first gives an introduction to the fundamentals of supervised learning in ML and elucidates the classification methods that have been evaluated as predictive phenotyping models based on Pfam abundance profiles in a statistically-sound experiment, followed by describing the extensions set on-top of the *MetaStone* code base. In Chapter 7 the evaluation and validation results are presented for each phenotype category, discussed in detail, and further improvements presented. After the presentation of the prediction performances a real world data set of 22 organisms is processed and the predictions of *PhenoPointer* are compared to a promising competitor phenotype prediction tool. This part closes with a final discussion of *Pheno Pointer*.

5 Machine learning-based classification

In computer science, machine learning (ML) is a research field in which computers are assumed to have the ability to learn without being explicitly programmed (Samuel 1959), emphasizing the process of learning, and to use this gained knowledge afterwards. ML methods are used to explore data sets for detection of hidden patterns and structures, to learn these and make practical use of them. Once a data space has been visited, new knowledge about their origin can be extracted and correlations between different data sets may be deduced, i. e. knowledge is acquired. Of course, the exploration of this newly generated data needs domain experts to interpret new findings, but they are the foundation of scientific data analysis. They help to formulate new hypotheses and define relationships in areas which have not been investigated so far because the practical tools to do so had not hitherto existed.

ML is a general term that unifies diverse techniques used in data analytics such as clustering of data (Sander et al. 1998), regression analysis (Efron et al. 2004), topic modelling in text mining (Blei et al. 2003), and image analysis (De La Calleja et al. 2004), to mention but a few. Related research fields are data mining and artificial intelligence. This chapter covers the classification of data, i. e. the categorization of new data identified by use of a predictive machine learner model. This model needs to be trained first before being applied to the new data sets. During the learning step, training samples are processed for detecting and extrapolating patterns in the underlying distribution of data points of a sample to generate a model for making classification on other data (Russell et al. 2010; Bishop 2006). The kth data point of samples is called a feature and the classification target is a single label or a set of labels, where a label represents an abstract class of information. The set of possible classes can be pre-defined or is unknown. The outcome of the training process is a prediction model, henceforth the requested machine learner/classifier.

As a loosely and for this work more than sufficient formal notation:

> An input X is a matrix and its components can directly be accessed by subscripts. Observed values, a single sample in our case, are denoted as lower case, therefore the ith sample in X is written as x_i, where x is a vector. The length of a vector $x_i \in \mathbb{R}^p$ is the number of available features p, so the all values of the kth feature can be assessed with X_k^T. G (for group) denotes qualitative output, in our case class labels, so a single label g_i corresponds to x_i, and g is written in bold and $\boldsymbol{g_i} \in G$ denotes a vector of labels. The classification goal can be defined as taking X as input for making good predictions on G, resulting in predictions $\hat{G}$ with elements from the set $\mathcal{G}$ associated with G, hence the whole set of N samples is $x_i, g_i, i = 1, \dots, N$, thus training a machine learner is equivalent to finding a suitable mapping function $f : X \mapsto G$.

The task of learning can be categorized into three algorithmic methodologies in general (Hastie et al. 2009):

- semi-supervised learning
- unsupervised learning
- supervised learning

The first category is semi-supervised learning methods, where only a small amount of samples is labeled and the majority are not, but both types sets are used during the training phase. This kind of classification algorithm is not used in this work, thus elucidating the methodology in more detail is out of scope. For the second category, no label of samples is known (or wanted) and is subject to the learning process, whereas for the third the labelling of samples is known and used during training for later predictive usage.

One of the biggest problems that can occur during ML-driven data analysis, is that some learning methods can be biased in classification scenarios by a high amount of dimensionality of the input feature space. This is an example of many possible manifestations of the so called *curse of dimensionality* (Bellman 1961). The sampling density is proportional to $N^{1/p}$, with N as the sample size and p representing the dimensionality of a sample (number of features). For $N_1 = 100$ and $p = 1$ it is fairly clear that the sampling density is quite dense. But with increasing p, the amount of samples needed to train a suitable classifier with comparable density is much higher; For the same $N = 100$ and $p = 10$ this would be $N_{10} = 100^{10}$. The spread of samples in the high dimensional feature space is sparse and training for gathering the real distribution of samples is much harder as a result. This does not mean that no satisfactory predictive performance can be achieved with such methods, but in this case a ML method must also generalize well non-locally, thus in regions not covered in the feature space.

5.1 Unsupervised learning

Unsupervised learning techniques rely solely on the input feature space, so no specific learning has been specified in advance. Thus the task is to identify concealed patterns in the feature space and to deduce a possible model of the input samples in a meaningful way. As an examples pattern recognition or feature learning can be named (Berkhin 2006). For this work feature learning is relevant e. g., to reduce dimensionality p of the input feature space and construct new features in a much lower dimensional space instead, so the curse of dimensionality can be circumvented. A well-known technique of dimensionality reduction is Principal Component Analysis (PCA) (Peason 1901; Jolliffe 2002), part of the field of multivariate analysis techniques (Mardia et al. 1980), so the statistical analysis of a set of statistical outcome variables with a minimum cardinality of ≥ 2 at the same time. PCA transforms a set of input data and its values e. g., through singular value decomposition (Golub et al. 1970), into a set of linearly uncorrelated variables named Principal Components (PC). The PCs represent the underlying variances of the input data space with decreasing variance explained i. e., the first PC carries the most variance and the last PC the least. Dimensionality reduction is accomplished by taking all those PCs into consideration until a minimal ratio of variance is reached by summarizing the variances of the chosen PCs instead of using the features of the input data directly for further ML analysis.

5.2 Supervised learning

In contrast to unsupervised learning, the labels G of samples X are used as prior knowledge for training a model in supervised learning for classifying a so-far-unknown data sample $\hat{x}$, i. e. assigning a label $\hat{g}$. Those methods training a classification model with an assumption on the distribution of data follow a parametric learning approach, and methods without any restrictions of the underlying data distribution are correspondingly non-parametric.

Some non-parametric supervised learning methods, especially, can suffer from the curse of dimensionality. To overcome this specific burden of high dimensionality of feature space p, one might apply dimensionality reduction methods prior to training or use parametric supervised learning techniques, where hard assumptions are being made. For such techniques the parameters of a machine learner must be chosen carefully to fit the underlying model in the unknown feature space to generalize in a meaningful way respecting the real distribution of samples, called generalization. As explained in Section 5.1, PCA can be applied beforehand to the input sample for dimensionality reduction.

The evaluated machine learners in this work belong to the methodological class of supervised learners. The chosen learners reflect classical and well-known approaches, but are still used nowadays where no domain-specific sophisticated ML method must be used. As initially introduced in Section 3.2, a classification task can be categorized into three problem classes as binary, multiclass, and multilabel. A binary classification problem is formally characterized by $|\mathcal{G}| = 2$ and g, $\hat{g}$ are scalar values. For multiclass classification the same definition applies to g and $\hat{g}$, but the set of class labels is larger, hence $|\mathcal{G}| \geq 2$. It can be said, that binary classification is a sub-problem of multiclass classification, so ML methods capable of solving multiclass problems can also be applied to binary problems, but not vice versa. In multilabel classification, the label set has a minimum size of 2, so $|\mathcal{G}| \geq 2$, and $\boldsymbol{g}$, $\hat{\boldsymbol{g}}$ are vectors, so they are defined as $\boldsymbol{g} = (l_1, \ldots, l_n)$, $\hat{\boldsymbol{g}} = (l_1, \ldots, l_{\hat{n}})$, $l \in \mathcal{G}$. Thus it can be said, that multiclass classification is a subcategory of multilabel classification and ML methods capable of solving multilabel classification problems might be applied to multiclass problems, but again not vice versa as in the case of multiclass and binary classification problems.

Gaussian/ multinomial Naïve Bayes Naïve Bayes classifiers are based on Bayes' theorem (Bayes et al. 1763) and are the simplest form of Bayesian networks. They can be used for binary and multiclass classification (Han et al. 2011). This approach is called naïve, because it combines Bayes' theorem with the assumption of conditional independence between values in the feature space in relation to the assigned label. This assumption is in contrast to many real-world settings, but the naïve Bayes performs quite well even on data not fulfilling the assumption and often better than more sophisticated approaches (H. Zhang 2005). The assumed underlying distribution of the features contributing to an assignable label is called event model, and thus all naïve Bayes methods are parametric ML approaches.

For the Gaussian naïves Bayes classifier, a normal distribution of continuous features is assumed. Discrete features are used in the multinomial naïve Bayes approach, where a multinomial distribution of features is assumed. The probability estimates can be zero, if features are not present for certain labels. To counteract this, a Laplace or Lidstone smoothing expressed as variable α can be optionally added onto the priors.

***k*-Nearest Neighbors** The idea behind the non-parametric neighbors-approach is an instance-based learning – not trying to fit a model for generalization but to classify on a majority vote on locally positioned k neighbors stored during the learning step (Fix et al. 1951; Cover et al. 1967). It can be used for multiclass and multilabel classification.

One of the most important criteria for predictive performance is the proper selection of the amount of neighbors k to take into account for classification, because a low value of k can increase the effect of noise and a high value of k can suppress noise, but also weaken the classification boundaries. A weight function can also be applied to the voting function, where all weights are uniformly distributed over all neighbors, or distance-based, where close neighbors get higher weights than those further away. Another parameter to tune is the distance metric, where one might choose the Manhattan or the Euclidian distance.

Decision Tree A decision tree is a non-parametric machine learner used for multiclass and multilabel classification (Breiman et al. 1984), where internal nodes represent branching decision nodes constrained to features and paths inside the tree lead to leaves representing class labels as the classification target. The step-wise construction of a decision tree is based on setting simple decision rules inferred from the training data on decision nodes and applying a statistical measure of which feature is the best splitting factor and how to perform the further separation of the input training data. Prediction performance is limited by the topology of the tree, so the depth (amount of used features used for training) and breadth are limiting factors and describe its complexity.

The advantages of a decision tree as a machine learner are its relatively low classification cost related to time consumption, and that it is human interpretable, since tree structures can be easily visualized and are, unlike other ML methods, white box models. Furthermore, the trained model's reliability can be validated by statistical tests. As disadvantages it must be pointed out that single decision trees tend to be over-complex in certain scenarios and do not perform well in terms of the generalization error, if the trained model had learned the incorporated error in the data but not the real distribution of it (Bramer 2016); this is called overfitting. In addition to the previous disadvantage, slight changes in the training data tend to generate entirely different decision trees (James et al. 2013).

To overcome these hindrances, one can apply ensemble learning techniques through training a set of weak learners resulting in one strong classifier with very good predictive performance and feasible generalization over the data space compared to the basing machine learner (Dasarathy et al. 1979; Hansen et al. 1990). In this work the Decision Tree described here is used as the base classifier for all ensemble methods except for the gradient tree boosting approach, where a regression tree is used, hence leaves carry real numbers instead of labels. It must be said that the choice of the base classifier is not limited to the Decision Tree (L. Yang 2011), again except for gradient tree boosting. Ensemble methods can be distinguished in general by two families of methodology as boosting methods on the one hand and averaging methods on the other. Boosting describes a technique of resampling data during an iterative training process of an ensemble of base classifiers whose individual predictions are combined by a voting function to deduce the final classification (Schapire 1990). Averaging methods follow the principle of training multiple classifiers independently and choosing the average of their predictions as the final classification results. Ensemble methods belong to the class of metalearners i. e., the primary task is not to build a model on the input directly but to learn how the performance during training is achieved and thus can be tuned to maximum prediction performance.

5.2.0.1 Adaptive Boosting

One boosting ensemble method is the adaptive boosting (AdaBoost) approach, which can be used for binary (Freund et al. 1995) and multiclass classification tasks (Zhu et al. 2009). AdaBoost starts with a finite set $\mathcal{C}$ of M base classifiers c as weak classifiers and enhances (*boosts*) these in an iterative manner one by one. Initial training starts with choosing random samples for the first base classifier c_1 and applying weights to the samples, which are $1/N$ in the first round. After the training phase the weights of samples are altered individually in such a way that sample weights of misclassified samples are increased and of correctly predicted samples decreased. In the following round, boosting takes place by choosing random samples for training in concordance to the updated weights of the round before and training the new classifier c_2. This adaptive boosting procedure continues, until training of classifier c_M has been finished. For classification purposes with this ensemble $\mathcal{C}$, a weighted majority voting is applied onto the set of individual predictions $\hat{G} = \{\hat{g}_1, \hat{g}_2, \ldots, \hat{g}_N\}$ to get the final classification $\hat{g}$.

With this iterative learning technique, individual base classifiers are more focused on the hard cases to predict and the ensemble of the boosted base classifiers has a higher predictive strength than any individual one. In this work a decision tree was used as the underlying base classifier, which had been boosted as an ensemble to maximize prediction performance and minimize possible side effects in relation to a standalone decision tree.

Forest of randomized trees Another example of an ensemble method is the Forest of randomized trees (Random Forest) approach suitable for multiclass and multilabel classification using decision trees as weak base classifiers (Breiman 2001). Since this method is an averaging ensemble method it trains an ensemble of independent base classifiers in contrast to AdaBoost, and final classification decision follows on an averaging procedure of predictions made by the base classifiers.

Given a finite set $\mathcal{C}$ of M base classifiers c_i, training is performed for each base classifier c_i given an individual set of bootstrapped samples $\mathcal{S}$ (randomly drawn with replacement) out of the training sample set X, where $|\mathcal{S}| = n$, $X = (x_{i,j}) \in \mathbb{R}^{m \times n}$. The training of the base classifiers is performed in the same manner as for standard decision trees with the exception that a random subset of features is used for decision making on how to perform the split on a decision node, in lieu of all features. This means that, as an effect of this randomness, a single base classifier's bias can be marginally increased, but thanks to averaging over all predictions $\hat{G}$, the total variance of the ensemble method drops more in relation to the gained bias, producing a better model for prediction as a whole (Ho 1998).

Gradient Tree Boosting Gradient tree boosting is an ensemble method applicable for binary and multiclass classification (Friedman 2001). As stated earlier, a regression tree is used as the base classifier and is trained in such an additive way that the prediction error is minimized through gradient descent in each boosting step.

The principle of boosting to a loss function is the following: For each label $g \in \mathcal{G}$ to be predicted, a predictive model $F(x) = \hat{y}$ is trained, where y is a real number. F consists of a finite set $\mathcal{C}$ of M base classifiers c_i, where iteratively classifiers c_i are added one to another to minimize the loss function. In each boosting step, a new instance of the base classifier denoted as $h(x)$ is added to correct the prediction error that has occurred in the step before, naïvely defined as $y - F(x)$, also named residuals. The additive definition of a boosted optimal F can be written as $F_m(x) = F_{m-1}(x) + h(x) = y$, thus by adding an optimal classifier h the prediction error $h(x) = y - F_{m-1}(x)$ is minimized. The main goal is to find an approximation of the function F, therefore a loss function, in this example the square loss, is needed to represent the error: $L(y, F(x)) = \frac{(y-F(x))^2}{2}$ is the target function that is going to be minimized during each boosting step.

The term *gradient boosting* is derived from the fact that while boosting the ensemble classifier, a gradient descent $\theta_i := \theta_{i-1} - \rho \frac{\partial J}{\partial \theta_{i-1}}$ takes place: Over all N samples X, the loss function has to be minimized, hence $J = \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i))$.

So J can be partially derived as followed:

$$\frac{\partial J}{\partial F_{m-1}(x_i)} = \frac{\partial \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)} = \frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)} = F_{m-1}(x_i) - y_i$$

Accordingly residuals can be formulated as negative gradients

$$y_i - F_{m-1}(x_i) = -\frac{\partial J}{\partial F_{m-1}(x_i)}$$

This can finally be put in the additive definition of F mentioned beforehand:

$$\begin{aligned} F_m(x) &= F_{m-1}(x) + h(x) \\ F_m(x) &= F_{m-1}(x) + y - F_{m-1}(x) \\ F_m(x) &= F_{m-1}(x) - \frac{\partial J}{\partial F_{m-1}(x)}, \text{ which equals} \\ \theta_i &:= \theta_{i-1} - \rho \frac{\partial J}{\partial \theta_{i-1}} \end{aligned}$$

This procedure is performed for each class label separately, thus the final ensemble classifier will consist of $|\mathcal{G}| * M$ base classifiers, hence the final classification $\hat{g}$ is the one with highest scoring probability of an ensemble $\mathcal{C}$, apart from binary classification, where only one ensembled model is induced since $|\mathcal{G}| = 2$.

Support Vector Classifier A Support Vector Classifier (SVC) utilizes a single Support Vector Machine (SVM) (Vapnik et al. 1963) for binary classification, but to solve a multiclass classification problem multiple SVMs are trained in a pairwise class manner, whereas for each pair of classes a single SVM is trained. In the multiclass case, each single SVM predicts a possible classification and the final classification is selected following the max-wins strategy as the SVC output.

The strategy of a linear SVM is that it tries to find a $p - 1$ dimensional hyperplane $\vec{\omega}$ as the decision boundary for separating data points into two distinct classes as labeled with $|\mathcal{G}| = 2$. During the training phase, the hyperplane is selected as the separator of the data points, that maximizes the distance between the data points of the two classes, names as the maximum-margin hyperplane. The higher the margin between the two sets is, the lower the generalization error is as a matter of fact. In a 3-dimensional feature space, the hyperplane would be a $2D$ plane, and in 2-dimensional space a $1D$ line, respectively, as shown in Figure 5.1. A sample x is in essence a vector $\vec{x}$ in a p-dimensional space, thus internally the computational effort of the optimization problem of finding a maximum-margin hyperplane is performed using the dot product of two vectors. Those vectors that lie on the two margins of the hyperplane are the so-called support vectors and are the ones describing the hyperplane. One of the most important parameters is C, which is the regularization term used for determining the margins properly and thereby controlling fit to the data and preventing possible overfitting. This term is used in linear and non-linear SVMs.

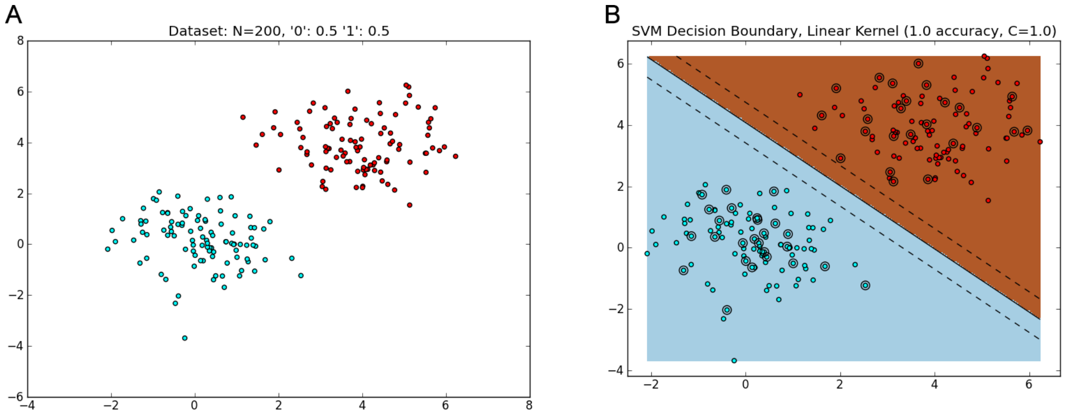


Figure 5.1: **Linear SVC for binary classification** **(A)** 200 samples are distributed in a $\mathbb{R}^2$ space in two cluster of equal size, where $g =$ "red" is set for the first cluster and $g =$ "blue" for the second one. **(B)** The solid line is the hyperplane (decision boundary) computed by a linear SVM, the dashed lines are the margins of it. the circled data points have been used as the test set. It can be seen, that the deduced hyperplane separates both clusters for a perfect separation from (A).

This strategy is not applicable to those data sets which are not linearly separable with a $p-1$-dimensional hyperplane, so mapping the data points to a higher dimensional space may be a solution to this problem, because a reasonable split may be found in a much higher dimensional space. As an example of an inseparable $2D$ feature space, there are data points at the center labeled as $g =$ "blue" and orbited by data points labeled as $g =$ "red". In this case no clear separation using a linear SVM with a line can be achieved. The solution here was to boost the dimensions to $3D$ by transforming X to X' with the transformation $\phi([x_1, x_2]) = [x_1, x_2, x_1^2 + x_2^2]$ first, and then to find a separating hyperplane in the $2D$ space secondly, see Figure 5.2. This method is non-linear, because $\vec{\omega}$ cannot be projected back onto the original p-dimensional space X.

Since for a prior transformation of $\mathbb{R}^p$ to $\mathbb{R}^{p+k}$ utilizing $\phi(x) = x'$ and further analysis the runtime and space requirement for a k-degree polynomial extension of the original feature space of size p would yield $\mathcal{O}(p^k)$, direct transformation and hyperplane determination in the boosted dimensional space becomes intractable. To solve this issue, a kernel function $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{p+k}$ can be used, which implicitly solves the dot product in $\mathbb{R}^{p+k}$ without transforming explicitly x to x'. Replacing the dot product by a kernel function is called the Kernel Trick (Boser et al. 1992). Suitable kernel functions are for instance polynomial, sigmoid, or even linear. For classification purposes a data point can be classified using the kernel K for implicit transformation and the deduced hyperplane $\vec{\omega}$.

5.3 Evaluation of machine learners

As diverse as the many application fields of ML are, as various are the different techniques for supervised learning described here. Additionally the ML methods applied on data sets sourced from the same domain with identical feature sets may differ in correlation to the target classes to be predicted. In summary, it can be stated that it is a challenging and work intensive task to evaluate suitable classification models for a particular data set with regard to generalization over the given feature space and in means of prediction performance.

It is important not to learn and test to the same set of data, because otherwise the trained model would reflect the mapping almost on a $1:1$ basis, so unseen data can not be analyzed satisfactorily since the trained classification model is overfitted and not generalizing at all. To evaluate the prediction performance of candidate classification models properly, the initial training data X, for which the class labels G are known, should be split into a training set X_{train} and a test set X_{test}. Training is performed on the training set and the predictive performance is measured by performing predictions on the test set, whose results $\hat{G}$ are compared against the real values G_{test}. The ML model and its parameters are tweaked and optimized in several rounds, until the classifier gives satisfactory prediction results (measured with evaluation metrics like accuracy). But it has to be noted that with such a model training procedure knowledge about the underlying data distribution and its class label relationship could seep through the optimization directly into the model until a nearly perfect fit is achieved, so no generalization can be assumed anymore and the potential of simulating unseen real-world settings dissipated (Rao et al. 2013). The solution here is to separate the data into three parts with an additional validation set that stays

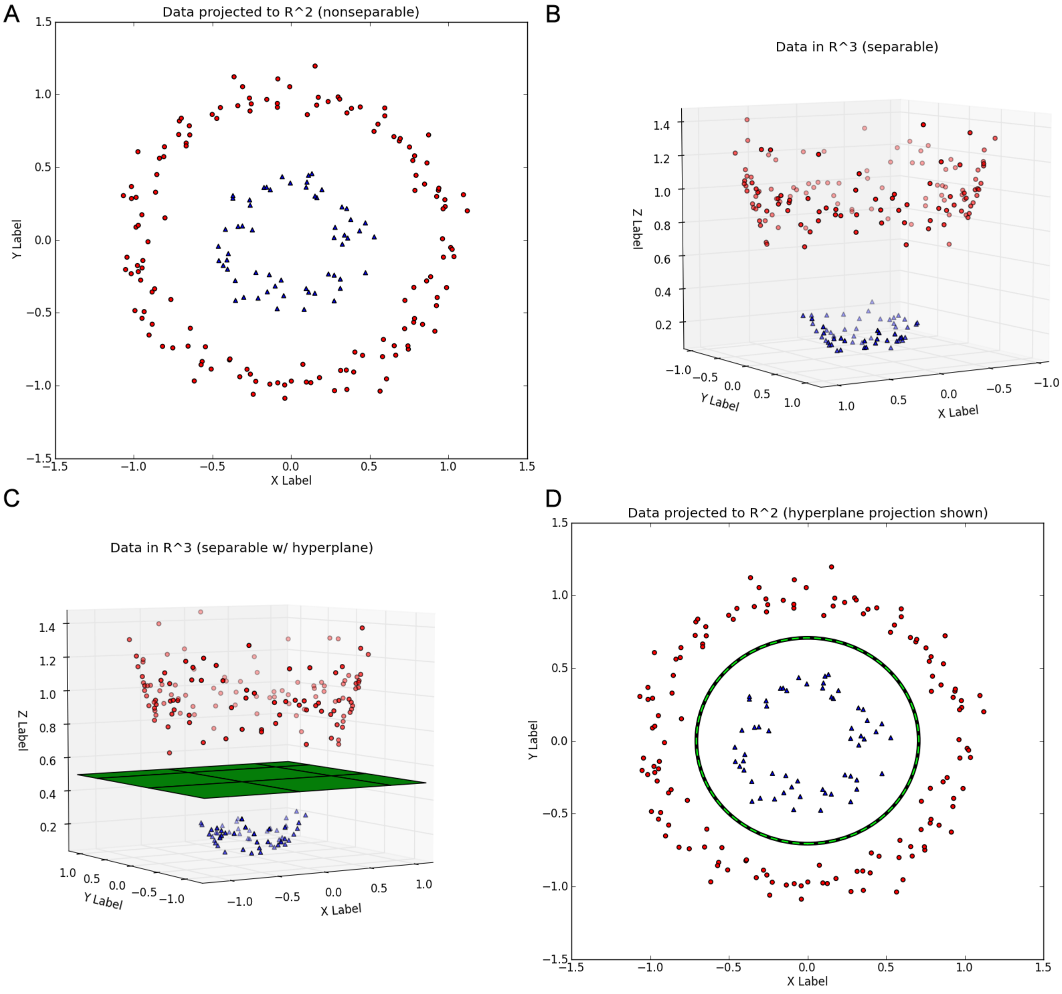


Figure 5.2: **Non-linear SVC for binary classification** **(A)** Samples are distributed in a $\mathbb{R}^2$ space in two cluster, where the inner circle is labelled as $g =$ "blue" and as $g =$ "red" for the outer orbit. The data points cannot be clearly separated with a linear SVM. **(B)** Transformation of data points shown in (A) with $\phi([x_1, x_2]) = [x_1, x_2, x_1^2 + x_2^2]$. **(C)** The green plane is the hyperplane $\vec{\omega}$ (decision boundary) computed by a non-linear SVM, that could be found for separating the two clusters in the boosted space (B). **(D)** The shown decision boundary is a non-linear projection of $\vec{\omega}$ onto the original space $\mathbb{R}^2$.

untouched for optimizing the model until the final prediction performance should be measured. A final classification model is thus trained on a small fraction of the initial data set without the test and validation sets, but high quality data is too expensive to be consumed in such a wasteful way.

For maximizing usage of the training data and guaranteeing generalization of a candidate ML model, cross validation is a commonly accepted way for evaluating a classification model (Kohavi 1995; Seni et al. 2010). This statistical technique is based on partitioning the input date set into k folds of equal size and during an iterative process of k rounds, where model training is performed on $k-1$ folds, i. e. a subset of the input data, and the leftover fold is used as the test set, which is equal to a validation set. Figure 5.3 gives an illustrative overview of this approach for a 3-fold cross validation process. Thus it can be said that training data is distinct from validation data and again, as described earlier, to inspect the fitting performance of the classification model in each iteration the model prediction and classification performance is measured with evaluation metrics. With this method no additional evaluation with distinct test sets and evaluation sets is needed due to the iterative methodology and averaging over the evaluation measurements. Hence, a decision about the final model and its parameters to use can be made. The final classifier is afterwards built upon the whole input data set. For initial test studies, the selection of k can be quite small, so e. g., one can perform a split into three partitions. If a cross validation procedure for selecting a model and its parameters for a real-world application should be performed, it has been established to perform a 10-fold cross validation study (McLachlan et al. 2004) ensuring a statistically sound setup.

If not all data points X in an input data set are equally assigned to class labels from $\mathcal{G}$, the cross validation procedure should take account of this. An extension to the k-fold method is the stratified k-fold cross validation, where the fractions of class labels of the total input samples are reflected in each fold. This approach can be performed in binary and multiclass classification studies, but not in the case of multilabel experiments.

For evaluating the prediction performance of an ML learner, the ratios of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions are not sufficient, to assess the predictive power in means of generalization and overfitting. What can, however, be performed with these values is the formulation of a confusion matrix (Stehman 1997a), where the values are plotted in a comparison table. This table can be further used for identifying structural misclassification for certain class labels by inspecting the FP and FN samples in detail. If there is an observable trend regarding certain samples belonging to one particular class this model should be dismissed. Especially in cases where the mean values of evaluation measures over all k folds are showing a high statistical variance, it might be worth building a confusion matrix for each fold.

In the following, the evaluation measures used in this work are explained in more detail (Olson et al. 2008). Since all measures rely on rates of TP, TN, FP, and FN per class label, and there exist three classification scenarios, the measures have to be averaged. If not stated in another way, the averaging is for binary and multiclass classification *weighted*, so the average score of each class label is weighted by the support of it within G (number of samples assigned with a particular class label). In cases of multilabel classification, each sample is a single instance to be

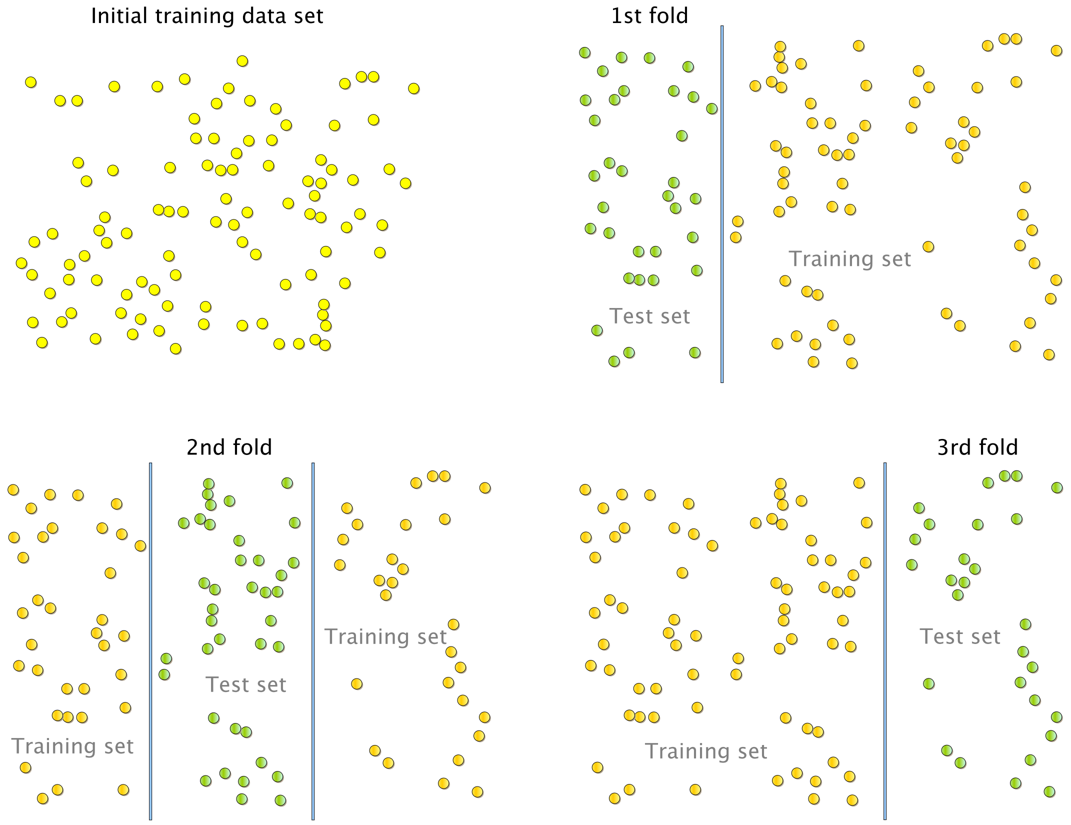


Figure 5.3: ***k*-fold cross validation** In a 3-fold cross validation setting, the initial data set is split into three folds of equal size. In the first round, fold two and three are used for training and the model is tested against the first. In the second round, fold one and three are used for training, the second fold is the test set. The third and final round finishes the 3-fold cross validation by training with the first and second fold, testing against the third fold.

measured and the final score is the average over all individual samples. The optimal value of a scoring function is 1 and the worst 0.

Precision The precision (PRC) measure describes the ability of assigning only those samples to a class, that really belong to it. It can be seen as a relevance measure in relation to the error rate.

$$\mathrm{PRC} = \frac{\sum TP}{\sum TP + \sum FP}$$

Recall A recall (RCL) is characterizing the ability to reflect the real distribution of assigned class labels, i. e. it accounts for missed assignments (FN).

$$\mathrm{RCL} = \frac{\sum TP}{\sum TP + \sum FN}$$

F1 score In a classification scenario only the correct assignment of labels to samples is of interest; TP are of relevance so to speak. PRC and RCL describe this predictive ability of a model well, thus accounting for missed classifications or falsely classified as a particular class member. To find a balance between PRC and RCL scores and combine them, the harmonic mean of these can be calculated: the F1 score.

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{PRC} \cdot \mathrm{RCL}}{\mathrm{PRC} + \mathrm{RCL}}$$

Accuracy The measure of correct predictions is the accuracy (ACC) score. This score is not weighted, because no distinction between binary, multiclass, and multilabel classification problems occurs: a prediction $\hat{g}$ is only realized as TP, iff $g = \hat{g}$ holds.

$$\mathrm{ACC} = \frac{1}{|X|} \cdot \sum_{i=1}^{|X|} 1(\hat{g}_i = g_i)$$

5.4 Recap and comparison of ML methods

Naïve Bayes works well on simple problems, quite often much better than more sophisticated methods even with a small amount of training data, but the outcome in large data sets is questionable. As a contrary method, the k-Nearest Neighbour approach could be considered, where no assumption at all is made about the distribution of the data and no correlation of features is assumed. Its approach is one of the simplest one could imagine: See who is in the neighbourhood and perform a majority vote. As simple as this method is, good predictions can be made. A drawback is that its most important parameter k (amount of neighbours) must be wisely chosen if one is to get good prediction performance. The larger the parameter the better it suppresses noisy data points, but overlooks nuances in the data. If a too small k is chosen there is no generalization effect any more. To trade off between exclusion of noisy data and detecting nuances in the data space, it is necessary to choose a weighted distance to look for neighbours. The closer a neighbour, the more weight it gets in the majority decision function. If some instances of data points with class labels $g =$ "*special one*" are distributed within the area of the feature space of more abundant data points labeled with $g =$ "*overall common class*", then these cannot be detected well with the previous methods. The same holds for a linear SVC and a SVC utilizing a linear kernel, since it tries to find a separating hyperplane for linear separation of data points in the feature space. In this case, a non-linear SVC with a suitable kernel function may be applied to the data set and its parameters may be optimized to achieve maximum prediction performance. The advantage of SVMs is that they are very effective in high dimensional features spaces, even if there are more features than samples to train on. Nevertheless it must be said that despite their memory efficiency and the fast execution time for classification using an already trained SVC, the computational time for training may be quite high in relation to other ML methods. Furthermore it can be very hard to comprehend how a prediction was made, because a non-linear SVM is a black box model in terms of transparency of the underlying model.

Another non-linear ML approach is the decision tree, where a prediction is made based on possible values of features. In a manner of speaking, it describes, through its internal structuring, the data it has been trained on and thus the whole decision making process can be surveyed. Since they make no assumptions about the distribution of the features correlated to the assigned labels of a data sample, prediction models of this kind are very flexible and can thus provide promising prediction performance. The traceability of predictions can be easily made from a visualization of the underlying tree model, and even be verified by statistical tests (Barlow et al. 2001). One major drawback is that it might tend to overfit, so that the model has a perfect fit on the training data set without making any error (or only marginal ones), but does not generalize well and is not suited for making predictions on hitherto unseen data, which is one primary goal of ML. One solution is to shorten the tree by setting suitable parameters of the model during training, such as tree depth and minimal amount of samples needed in a leaf node. Some other tweakable parameters for training a decision tree influence how the split on a decision node is performed, so that the measure used to evaluate the quality of a split (Gini impurity or entropy for information gain) and the maximum amount of features for considering the applied measure on.

SVC classification models can generate very good prediction results, but require considerable tweaking of the parameters. By contrast, AdaBoost gives quite comparable results to an almost perfectly trained SVC, but with less optimization efforts. On the other hand, AdaBoost is very sensitive to outliers and data sets containing erroneous data samples, but it is less prone to overfitting than other ML methods. Random forest avoids this issue by using a lot of decision trees, averaged during prediction making, which also improves the accuracy of the model. Accuracy is the fraction of correctly classified samples of a test set, for which the correct labeling is known and which the model can be tested against. A bias towards dominating classes out of $\mathcal{G}$ can be counteracted by boosting a decision tree as performed in AdaBoost and gradient tree boosting by compensating misclassifications in each boosting step. In forests of random decision trees the bias of a single particular tree is slightly higher due to the random split procedure, but as mentioned in Section 5.2.0.1, the variance of the overall model also decreases in relation to a single decision tree-based prediction model. The characteristic of instability owing to a slight modification in a training data set resulting in completely different trees can be alleviated by using decision trees as base classifiers within ensembled models. The gradient tree boosting method is conceptually robust against outliers, but one has to be aware of the time consumed during training, because the boosting is performed sequentially and thus can not be parallelized. The same holds for AdaBoost, but not for the random forest, where each decision tree of the ensemble is trained independently.

6 *PhenoPointer* – Principles and Implementation

As defined in Chapter 3, in the scope of *PhenoPointer* a switch from secondary to primary information takes place: the goal is to train machine learners on Pfam abundance profile of metadata annotated bacterial and archaeal genomes to reliably predict expressed phenotypes in novel hitherto unknown prokaryotic organisms. The information was gathered in March 2016 from the *IMG/M* system, which incorporates additional metadata from *GOLD*. Since the assignment of metadata class labels to particular samples (a bacterial or archaeal genome) is known, in this work are supervised learning methods as candidate classifiers used for final model selection, described in detail in Chapter 5.2 from page 49 onwards. An individual ML learner is trained for each phenotype category as shown in Table 3.2 on page 31. The decision making process, as the selection of the best ML classifier per phenotype, is guided by a 10-fold cross validation process per phenotype, which is a stratified one for binary and multiclass classification problems and a standard cross validation for multilabel classifications. After a classification method with its probable parameters has been identified, additional measurements are assessed for contaminated and degenerated genomes. As a final validation, the prediction performance is evaluated against all bacterial and archaeal genomes, that have been added to *IMG/M* in the period March 2016 to April 2017 as a validation set. The *IMG/M* genomes are quasi final assemblies, where the analysis is completed, and the available annotations represent the biological truth. Each validated classifier is thus a strict one, representing a strongly-performing generalizing ML learner.

6.1 Features and classification targets

In *PhenoPointer* the feature space of a sample X represents the abundance of protein domains and families as Pfam v.29 entities in a genome i. e., the first feature stands for the amount of detected '`PF00001 -- 7 transmembrane receptor (rhodopsinfamily)`'[14] domains in a genome and the last one for the amount of detected '`PF17203 -- Single cache domain 3`'[15] domain in a genome, with a total of 16295 possible protein domains. The prediction target set $\mathcal{G}$ and classification problem class differs per phenotype category; furthermore, the amount of training samples and contained protein domains differs, because the metadata labeling is not available for every prokaryotic genome and moreover not every protein domain is encoded in each genome. Table 6.1 gives an overview of the phenotype categories, their problem class, and their density in relation to the total amount of available genomes in the March 2016 release of *IMG/M* (*M data set*) and newly uploaded genomes into *IMG/M* in the period March 2016 to April 2017 (*A data set*). The last column shows the total amount of annotated genomes and cardinality of $\mathcal{G}$ in the pooled data set (*M+A data set*).

[14] *PF00001* – `http://pfam.xfam.org/family/PF00001`

[15] *PF17203* – `http://pfam.xfam.org/family/PF17203`

Phenotype Category	#Pfam motifs	M: Density	M: #Classes	A: Density	A: #Classes	M+A: Density	M+A: #Classes
In total	16295	100 % (34862)	–	100 % (18328)	–	100 % (52614)	–
Biotic Relationships	10291	19.99 %	2	0.75 %	2	12.45 %	2
Gram Staining	10820	47.61 %	2	4.62 %	2	32.09 %	2
Sporulation	10175	18.07 %	2	0.93 %	2	11.63 %	2
Cell Shape	10388	23.72 %	25	1.32 %	11	15.21 %	25
Motility	10326	22.14 %	3	1.13 %	3	14.15 %	3
Oxygen Requirement	10407	24.51 %	7	2.55 %	7	16.19 %	7
Salinity	8362	1.33 %	4	0.11 %	3	0.89 %	4
Temperature Range	10472	24.54 %	7	1.87 %	6	15.73 %	7
Cell Arrangement	9832	12.23 %	9	0.63 %	7	7.77 %	9
Diseases	9031	11.09 %	332	0.3 %	36	6.87 %	333
Energy Source	9675	8.14 %	30	0.95 %	17	5.72 %	30
Metabolism	9261	3.56 %	138	0.22 %	26	2.32 %	139
Phenotype	9954	12.76 %	132	0.48 %	18	7.83 %	132

Table 6.1: **Phenotype categories for predictive usage in *PhenoPointer*** Initial training is the M data set, which is extracted from *IMG/M* as of March 2016. The A data set is the validation data set and consists of newly populated genomic records in *IMG/M* as of April 2017 from March 2016 on. The ML classifiers utilizing this distinct data sets form the strict phenotype predictors, whereas the M+A data represents the set used within *MVIZ* for metadata-enrichment of microbial community profiles.

The stated number of possible Pfam motifs is the intersection of protein domains over all genomes that have been labeled according to the described phenotype. From the total of 52614 genomes stored in *MetaStone*, ∼ 18068 genomes are assigned to a phenotype category of relevance and thus these genomes represent the total training set for evaluation suitable ML methods for phenotype prediction. The naming of the phenotype categories are taken on a 1 : 1-basis from *GOLD*, *IMG/M* respectively, thus it must be clarified that the category *Phenotype* is a particular metadata category and not a summary of all given categories.

6.2 Strict classification models

As introduced at the beginning of this chapter, the main target is to construct a particular classification model per phenotype category utilizing the M data set for training and the A data set for auxiliary final validation.

The set of validated prediction models has been evaluated with 22 supplementary high quality annotated complete genomes, analyzed in-house at CeBiTec (Center for Biotechnoogy), Bielefeld University. As additional performance measurements, evaluation metrics have been computed for the final prediction models on degenerated and contaminated sample sets, where cross-validation is performed utilizing unaltered samples, but the feature set of validation samples has been modified in each fold.

Statistics on degenerated genomes A degenerated genome is an incomplete genome, which arises as a result of a genome binning assembly or a single cell sequencing experiment. Three levels of degeneration (5%, 10%, 20%) mimic incomplete genome bins, where randomly the specified fraction of annotated Pfam motifs in the abundance profile of validation samples has been set to 0. With this approach, an interpretation of the influence of missing features on the prediction performances of the final classification models per phenotype can be feasibly made.

Statistics on contaminated genomes For the analysis of possibly contaminated genomes, as an outcome of an impure genome bin or contaminated genome assembly, random Pfam abundances in the validation set have been altered in such a way that their counts were increased in a range of $1, \ldots, 10$. The fraction of altered Pfam motifs is split into four levels (3%, 10%, 15%, 20%).

6.3 Cross-validation workflow

The procedure of cross-validated phenotype-specific ML classifiers is divided into four iterative steps for the strict classifiers. The first step computes several classification model candidates for every phenotype category with different methodology related parameter setups, and measures their prediction performances following the 10-fold cross validation scheme. In the second step, selection of the best-performing classification model, the statistics of the candidate models for each phenotype category, are gathered and compared based on the four evaluation scores, introduced in 5.3 from page 54 onwards. If a single classifier outperforms in all of the four metrics, then this ML method will be chosen with its particular parameter set as the classification method for final classifier built. If two or more methods perform equally well, then the statistics on degenerated and contaminated genomes are added to the evaluation process and their comparison yields the final method and parameter set.

Selection of the best-performing ML model is followed by training the conclusive classifier for each phenotype. In this third step, the model selection and its set of parameters is frozen, thus the classifier is trained upon the whole input training data set with all samples, resulting in a well-generalizing machine learner that can be delivered to the end user. For the strict set of classifiers, a terminal validation against the A data set is followed as the fourth step. If this evaluation shows satisfactory results, no further consultation of other ML candidates need be performed. In the case of questionable results, a confusion matrix can be used to identify possible flaws to finally decide whether to discard this classifier and select the second best scoring method from cross validation as the final classification model. The final classification model set is evaluated with 22 manually functionally annotated genomes as an example of a real-world application.

6.4 Extensions to the *MetaStone* code base

The *PhenoPointer* related extensions made to *MetaStone* for ML capabilities, such as classifier implementing classes, modules and control workflows are listed in Figure 6.1 on page 64.

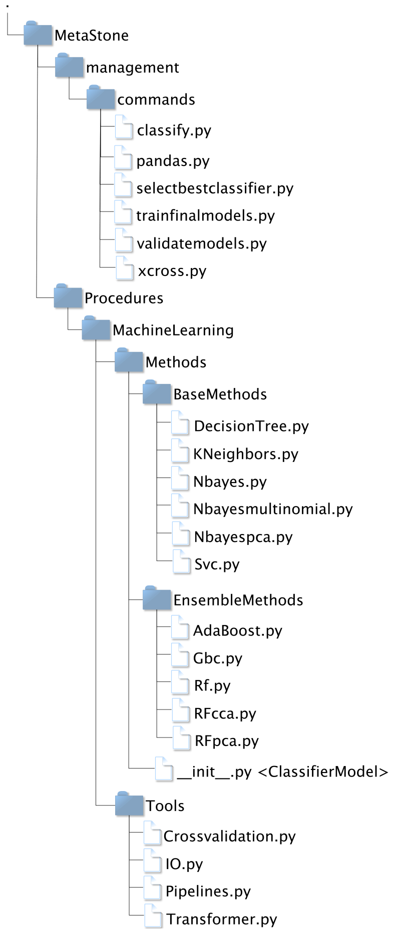


Figure 6.1: ***PhenoPointer* code extension to *MetaStone*** ML implementing classes are located under *./MetaStone/Procedures/MachineLearning/Methods/*, where non-ensemble learners are located under the subdirectory *BaseMethods/* and ensemble Methods under *EnsembleMethods*. Workflow control modules and service modules are located under *./MetaStone/Procedures/Machine Learning/Tools*. CLI command implementing classes are defined under *./MetaStone/management/ commands/*.

ML method implementing classes are located under *./MetaStone/Procedures/MachineLearning/ Methods/* and related subdirectories, whereas helper modules and control workflows are located under *./MetaStone/Procedures/MachineLearning/Tools/*. The implementation of the ML algorithms itself is not part of this work, thus *PhenoPointer* relies on the ML methods from *scikit-learn – Machine Learning in Python* (Pedregosa et al. 2011) library and uses functionalities and concepts from *pandas – Python Data Analysis Library* (Stehman 1997b) for internal data structures.

The evaluated ML methods, as described in 5.2 from page 49 onwards, are implemented in an identical fashion as they implement a common interface for classifier training, perform phenotype prediction on a sample, and perform cross validation returning a fixed pre-defined set of evaluation scores. To achieve this, they are all inherited from `class ClassifierModel(object, metaclass=abc.ABCMeta)`, implemented in *./MetaStone/ Procedures/MachineLearning/Methods/__ini__.py*, in a factory design pattern and each ML class registers itself in the superclass, thus a fully automated cross-validation process can easily be implemented. A ML class carries a particular instance of an ML method from scikit-learn as a class field, so the model related parameters are those used in the scikit-learn library. Model related parameters are passed during `train()` function invocation as a generic `**kwargs`[16] parameter, are dynamically unpacked, and passed to the scikit-learn implementation as constructor parameters.

Genome sets can be loaded and exported from *MetaStone* as a `pandas.DataFrame`, so contacting the data base for data delivery before any cross validation run is avoided. This useful because the phenotype data sets used for ML training will not change and are processed several times. The implementing function `genomes_characteristics_to_dict()` can be found in the module *./MetaStone/Procedures/MachineLearning/Tools/Transformer.py*, which is used in a wrapper CLI command for writing a gzip'ed pickled `DataFrame` object onto local storage for later usage. The exported file can easily be imported in any continuative *PhenoPointer* instance due to the python object serialization capabilities as 'pickled' objects.

The encapsulation cross validation helper module is in *./MetaStone/Procedures/MachineLearning/Tools/Crossvalidation.py*, which loads the training data directly from the database or loads previously persisted `DataFrames`from any location in the local file system, instantiates a particular `ClassifierModel` and starts the k-fold cross validation by calling the function `cross_validate_stratifiedkfold()` in the particular `ClassifierModel` inherited instance.

For automation purposes an enclosing module is implemented in *./MetaStone/Procedures/MachineLearning/Tools/Pipelines.py*, which parses a configuration file in the INI file format[17] and starts the defined cross validation workflow over specified phenotype data sets for a series of ML methods and parameter setups. The ML related parameters are given in the INI configuration as key-value pairs, which are packed in a dictionary to pass this as described earlier to the constructor of a scikit-learn ML learner implementation. The output of a cross validation experiment run are individual textual output files containing the evaluation scores, where the precise setup for a run is encoded in the file name. Encoded are the ML method, related parameters and values of it as key-value pairs, the utilized data set and the level of genome completeness/contamination.

Since various combinations of parameters for a certain machine learner are possible, the value of a key-value mapping defined in the INI configuration can be a list of elements, thus multiple cross validation runs are performed in such a case and need to be packed as distinct dictionaries respectively. Assume two parameters p_1 and p_2 for a ML method M are defined as $p_1 = \{1, 5\}$ and $p_2 = \{\text{'log2'}, \text{'sqrt'}, \text{'auto'}\}$, this yields in six cross validation runs for this specific ML method:

$$|p_1 \times p_2| = |\{(1, \text{'log2'}), (1, \text{'sqrt'}), (1, \text{'auto'}), (5, \text{'log2'}), (5, \text{'sqrt'}), (5, \text{'auto'})\}|$$

$$|p_1 \times p_2| = 6$$

[16] ***kwargs* – Dictionary of key-value pairs, where keys become during invocation of the function named parameters assigned with the designated value

[17] *INI file format* – Simple plain text configuration file format, structured in sections followed by key-value pairs, where value can be a single entry or a list of entries. `https://docs.python.org/3.4/library/configparser.html`

The total amount of cross validation runs is the product of the cardinality of phenotype work packages $\mathcal{W}$ to train on, and the sum of the cardinalities of Cartesian products of parameters $\mathcal{P}_m$ for each evaluated ML method $\mathcal{M}$:

$$|\mathcal{W}| \cdot \sum_{m \in \mathcal{M}} \left(\prod_{p \in \mathcal{P}_m} |p| \right)$$

6.4.1 Ini file specification

For ease of use, the `xcross()` function in the *Pipelines.py* module performs multiple cross validation runs in various data sets with several ML methods, which are defined in configuration file, formatted as an INI file. The structure of a *PhenoPointer* compatible INI file is divided into three. The first part is a single section named `[setup]` for specifying more general parameters like classifiers to evaluate, the amount of folds, completeness/contamination level of a genome, and instruction for sample sets (work package) to test on. The second part consists of definitions about the work packages – the pickled data frames, their paths and classification problem class. In the third part, listings of ML methods with sets of parameters are defined. Listing 6.1 on page 68 shows an example of a cross validation setup defined in INI format. For loading a previously pickled `pandas.DataFrame` it is assumed that its directory is located in the local file system at the same level as *./MetaStone*, i. e. *./dataframes*.

[setup]
All six following entries are mandatory, because they define the general outline of the cross validation experiment to perform. The order of elements is not important.

classifier Comma-separated list of machine learner setups. A particular entry is used as a section name later on in the INI configuration file.

classifier_kfold The amount of folds to divide the data set into. The value must be an integer in float notation.

dataframe_base_dir Base path to the directory, where further subdirectories for separate data frame packages are located.

dataframe_suffix The suffix of a pickled and gzip'ed data frame

dataframe_package Comma-separated list data frame work packages. A particular entry is used as a section name later on in the INI configuration file.

completeness Integer value specifying the completeness of each sample in the validation set. 100 stands for 100% complete genomes, where a value < 100 sets randomly Pfam abundances to 0 until the specified fraction of completeness is reached, based on the amount of identified protein domains found in a particular genome. A value > 100 alters Pfam abundances by increasing random Pfam abundances, until the amount of desired contamination level is reached, i. e. 105 results in 5% of Pfam abundances being randomly set between 1 and ≤ 10.

[data frame work package – *dataframepackage*]

The second part of a cross validation configuration file consists of sections for grouping data frames with sample sets of the same classification problem class. Names of the sections defining dataframe packages are those previously denoted in `[setup]-> dataframe_package`.

type Classification problem class. Can be one of the following values: *binary*, *multiclass*, or *multilabel*.

subdir Subdirectory name, where the pickled data frames are stored. This entry will be prefixed with the entry from `[entry]-> dataframe_base_dir`.

dataframes List of pickled data frames, where each entry stands for a single file.

The combination of `[setup]-> dataframe_base_dir`, `subdir`, one list entry from `dataframes` and `[setup] -> dataframe_suffix` results in the path pointing to the desired data frame, i./,e. *./dataframe/binary/gram_staining.pck.gz*.

[ML methods – *classifier*]

The last part defines the ML models to evaluate. The name of each section is taken from the list in `[setup]-> classifier`, where the entries in each section are used to define the different combinations of parameters and their settings.

All entries come in tuples, since the parsing mechanism in python of INI files interprets each value as plain `string`, thus explicit type casting must be performed. The naming of entries follows the pattern that the prefix followed by `_type` sets the desired data type to cast the value of the parameter to, whereas the solely prefix defines the name of the parameter passed to the underlying ML class with its value respectively. The parsed tuples are passed as the previously introduced `**kwargs`.

name_type Always `str`

name Name of the `ClassifierModel` implementing class, with which it has registered itself in the model factory. Valid values are

- **dt** Decision Tree
- **kn** *k*-Nearest Neighbors Classifier
- **nbayes** Gaussian naïve Bayes
- **nbayesmultinomial** Multinomial naïve Bayes
- **nbayespca** Gaussian naïve Bayes with previous dimensionality reduction utilizing PCA
- **svc** Support Vector Classifier
- **adaboost** AdaBoost
- **gbc** Gradient Tree Boosting
- **rf** Forest of randomized trees
- **rfpca** Forest of randomized trees with previous dimensionality reduction utilizing PCA

*parameter***_type** Target of type casting for the value of *parameter*

parameter The value of the *parameter*.

pre_*parameter*_type Target of type casting for the value of *parameter* used in ML methods, which perform initial dimensionality reduction

pre_*parameter* The value of the *parameter*

```
[setup]
classifier=rfpcamle,gbc
classifier_kfold=10.
dataframe_base_dir=dataframes/
dataframe_suffix=.pck.gz
dataframe_package=multiclass,binary,multilabel
completeness=100

[multilabel]
type=multilabel
subdir=multilabel
dataframes=cell_arrangement_both,diseases_both,energy_source_both,metabolism_both,phenotype_both
[multiclass]
type=multiclass
subdir=multiclass
dataframes=cell_shape_both,oxygen_requirement_both,salinity_both,temperature_range_both
[binary]
type=binary
subdir=binary
dataframes=biotic_relationships_both,gram_staining_both,motility_both,sporulation_both

[rfpcamle]
name_type=str
name=rfpca
n_jobs_type=int
n_jobs=40
n_estimators_type=int
n_estimators=50,100,200,400,600
criterion_type=str
criterion=entropy,gini
pre_n_components_type=str
pre_n_components=mle
[gbc]
name_type=str
name=gbc
loss_type=str
loss=deviance,exponential
n_estimators_type=int
n_estimators=10,50,100,200,400,600
max_depth_type=int
max_depth=1,3,5,7,10,20,50,100
max_features_type=str
max_features=auto,sqrt,log2
warm_start_type=bool
warm_start=False,True
```

Listing 6.1: INI configuration file for testing all 13 phenotype data sets with gradient boosting classifier and Random Forest classifier with previous dimensionality reduction utilizing PCA.

6.4.2 CLI commands

In addition to the previously defined functionalities of *PhenoPointer* – exporting sample sets as `pandas.DataFrame`, performing cross validation, and phenotype prediction for a novel prokaryotic organism – further CLI commands have been implemented. These include such tasks as selecting best-performing classification model after cross validation experiments, building the final phenotype classification models for end user usage, and validating final classifiers against a validation data set. In the following, *PhenoPointer* CLI commands are explained in detail.

pandas To export data sets from *MetaStone*, e. g. genome related metadata, this command loads a bulk of samples from the data base and persists the generated `pandas.DataFrame` as a gzip'ed pickle file. To specify which data sets to export, a data base model must be stated as well as the desired metadata table field. Valid values are those from the DB ORM model as shown in Figure 4.2 on page 36. Reasonable arguments for the DB model and related table fields are as following:

SamplingSiteCharacteristics `habitat`, `sample_body_site`, `sample_body_sub_site`

SequencingCharacteristics `sequencing_method`, `status`, `type_strain`, `uncultured_type`

EcosystemRelatedCharacteristics `ecosystem`, `ecosystem_category`, `ecosystem_type`, `ecosystem_subtype`, `relevance`, `specific_ecosystem`

SpeciesRelatedCharacteristics `biotic_relationships`, `cell_arrangement`, `cell_shape` , `diseases`, `energy_source`, `gram_staining`, `metabolism`, `motility`, `oxygen_requirement`, `phenotype`, `salinity`, `sporulation`, `temperature_range`

xcross Performs a cross validation experiment. Takes as input a INI configuration file and performs the designated experimental tasks as specified in an iterative approach. For parallelization purposes multiple instances have to be instantiated.

selectbestclassifier Takes as input a path to a directory of cross validation output files to generate statistics about the best-performing ML method(s) for a phenotype data set.

trainfinalmodels Takes as input an INI file, where for each phenotype a single ML methods with fixed parameters and `dataframe_package` is defined. Based on this, a final classification model is trained on the whole input data set and persisted in the local file system.

validatemodels Validates final classification models against a validation set.

classify This command is the end user entry point for performing phenotype prediction on novel genomes. It supports as input an HMMer output file format of a previously Pfam v.29 annotated genome or a FASTA file containing contig(s). The latter will be processed with prodigal for gene detection and further annotated with pfam_scan.pl utilizing HMMer with Pfam v.29 as reference motif set. The output is a text file containing key-value pairs of phenotype categories and their predicted value.

6.5 Final phenotype prediction models

A typical workflow for generation phenotype predictors can be built based on the CLI commands introduced in 6.4.2. Frist, generate pickled data frames utilizing `pandas`, then define experiments in a INI configuration and pass this configuration to the `xcross` command. After several cross validation runs under different settings, select automatically the best-performing ML methods for a specific task utilizing the `selectbestclassifier` command. Finally train consecutively final prediction models with `trainfinalmodels` and validate these via `validatemodels`.

The final classifiers can be used after this procedure by the end user utilizing the `classify` command for novel genomes that need to be annotated with probable phenotypes. As the `classify` task takes readily assigned Pfam motifs for a genome or a FASTA file as input, the potential target audience is enlarged, since basic knowledge of a command line interface is needed. Via defining easy-to-use callable CLI tasks, even an integration into a larger computational environment such as clusters and/or web services can be made. This is preferable especially in cases where mainly unannotated genomes are used for input of *PhenoPointer* in the form of 'raw' biological sequences such as assemblies or a single contig.

Pre-trained models can be downloaded from the GitHub repository, available under `https://github/mrumming/ppmodels/`. These files must be placed under *./classifiers/*, where *.* is the base directory of the *MetaStone*/*PhenoPointer* installation directory. Additional files for filling *MetaStone* with genomes, metadata, and Pfam information are provided, so the *M+A data set* and auxiliary files needed to populate the database.

7 *PhenoPointer* – 13 classifiers for phenotype prediction

All ML methods as described in Chapter 5.2 have been trained and evaluated with different model-specific parameter setups in a 10-Fold cross validation manner. The whole line-up of methods and parameter sets was tested for each metadata category describing microbial phenotypes, also called *traits*, and finally validated against a validation set (see '6.2 – Strict classification models' on page 62 for reference), named as data set A. When a single method outperformed all other methods in the four given evaluation metrics (details in Chapter 5.3), this method was chosen as the final classification model. If no clear best-performing method could be found, than the decision was based on the evaluation scores for degenerated genomes used for testing on three different levels of 95%, 90%, and 80% completeness. For the final prediction models of microbial phenotypes, the predictive performance for contaminated genomes was computed on levels of 3%, 10%, 15%, and 20% contaminated genomes. In the following the classification models are explained in detail and discussed.

Experiments have shown that dimensionality reduction utilizing PCA prior to training in cross-validation resulted in no increase of prediction performance in any of the thirteen phenotype categories. Dimensionality reduction was performed for all ML methods with several parameter sets (based on the ML method where applicable) on different levels of expressed variance through selecting subsets of principal components until the desired variance level has reached. Because of this behavior of the ML methods, 10-fold cross-validations experiments were performed on the whole set of available features per phenotype category.

For binary and multiclass classification problems an explanatory confusion matrix is provided for better understanding of misclassifications and the 'hard' cases to predict. In the end of this chapter a short note about runtime and memory consumption is given. Boxplots showing prediction performances of the 10-fold cross validation runs for each phenotype are given in the Appendix from page 159 onwards, as well as settings of ML methods and their parameters sets from page 167 onwards.

7.1 Biotic Relationships

The distinction as to whether a microbe is a symbiont living aside a host organism or a free-living organism is made with this phenotype prediction model. Studies related to symbiotic microbes have shown the importance of mutual symbionts for human health, such as the influence of the gut microbiome influencing the host metabolism (Devaraj et al. 2013; Lozupone et al. 2012) or acne associated microbes (Fitz-Gibbon et al. 2013). These studies benefited from the vast amount of data generated during the experimental phase of the Human Microbiome Project (HMP) (Human Microbiome Jumpstart Reference Strains Consortium et al. 2010) to chart the human body and characterize its inhabited microbial communities. Mutual symbiosis can also ensure survival capabilities of host organisms living in hostile environments. *Riftia pachyptila*, a deep sea tube worm, lives near hydrothermal vents and is dependent on thioautotrophic microbes (Barnes et al. 1992; Stewart et al. 2006). Endosymbiotic organisms can also be mentioned, where these are living inside an organism's tissue. A well-known example are certain *Rhizobiaceae* that enable optimal growth of plants by nitrogen-fixation directly in root nodules (Laguerre et al. 2007; Alami et al. 2000; Wdowiak-Wróbel et al. 2017). It must be noted, however, that no subdivision of symbioses with respect to mutualism, commensalism, or parasitism takes place.

GBC as classification model The classification problem class of *Biotic Relationships* is a binary one, because the classification target set consists of exactly two values $\mathcal{G} = \{\text{'Free living'}, \text{'Symbiotic'}\}$. Gradient tree boosting was identified during the 10-fold cross-validation run as the best-performing ML method for this phenotype, whereby three different parameter setups of GBC had almost identical validation scores. All three candidate models had in common that boosting was performed in 600 rounds. They also used the square root of the amount of possible features for decision making on how many features should be taken into consideration to evaluate the perfect split in an internal node of the underlying tree. Candidate model #1 and #2 used the deviance function[18] as the loss function, whereas candidate model #3 used the exponential loss function, which resembles a behavior likely to AdaBoost. The maximum depth of the underlying tree is set in candidate #1 and #3 to 10, whereas it is set to 20 in candidate #2. Through inspection of validation scores of the three models on $80\%, 90\%, \text{and} 95\%$ genome completeness of the test set, candidate #1 outperformed the other models for 90% and 95% completeness and was thus selected as the final classification model on *Biotic Relationships*. For 80% completeness no model performed best in relation to the other candidates. Table 7.1 shows the validation results of the final classification model of *Biotic Relationships* for degenerated, contaminated and validation against data set A.

Remarks and discussion As shown in Table 7.1, the classification model shows overall excellent prediction results for complete and incomplete genomes. With increasing values of simulated contamination, the predictive performance is very good for 3% and 10% levels of contamination and still good for 15%. A huge drop occurs for a contaminated genome that carries 20% of erroneous features. Thus it can be said that this predictive model generalizes very well and can be applied to degenerated and even slightly contaminated genomes, retrieving very good predictions on hitherto unseen genomes, depending on the level of contamination. The sample set used for evaluation shows comparable results for this classifier.

[18] Comparable to deviance in logistic regression.

Genome completeness/ contamination		Precision	Recall	f1	Accuracy
Incomplete genomes	80%	0.9584	0.9598	0.9566	0.9598
	90%	0.9626	0.9638	0.9612	0.9638
	95%	0.9651	0.9661	0.9635	0.9661
	100%	0.9653	0.9666	0.9641	0.9666
Contaminated genomes	100% + 3% contamination	0.9081	0.8286	0.8539	0.8286
	100% + 10% contamination	0.8832	0.1669	0.1317	0.1669
	100% + 15% contamination	0.7242	0.1128	0.0275	0.1128
	100% + 20% contamination	0.0121	0.1099	0.0218	0.1099
Validation		0.9581	0.9559	0.9533	0.9559

Table 7.1: **Evaluation scores – Biotic Relationships** GBC – max_depth: 10, max_features: sqrt(#features), boosting rounds: 600, loss-function: deviance.

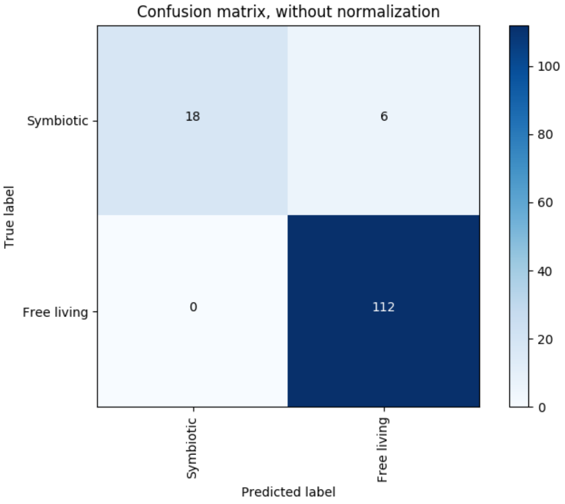


Figure 7.1: **Confusion matrix (validation) – Biotic Relationships**

7.2 Cell Shape

The shape of microbial cells is commonly used for taxonomic determination of microbes and come in various forms such as rod-shaped, vibrio-shaped, or filament-shaped cells. It is possible to taxonomically differentiate mutants from related bacterial strains (Cabeen et al. 2005). For instance, mutants of the rod-shaped bacteria *Bacillus subtilis* (Honeyman et al. 1989) and *Escherichia coli* (Doi et al. 1988) show a spherical or helical form. For the *E. coli K12* strain a spherical mutant was identified, where the shape inhibited directly the binding of β-lactams, hence this mutant was resistant to this specific kind of antibiotics (Spratt 1975).

GBC as classification model The maintainers of *GOLD* partition the shape of a bacterial or archaeal in 25 different shapes in total, but is distinct for a genomic entry, so that the classification of cell shapes of an organism becomes a multiclass classification problem. Two different setups of a gradient tree boosting classifier showed initially better results than all other evaluated methods. Both candidate classification models have been constrained, so that the maximum depth of the underlying weak learner was limited to 7, the maximum amount of features taken into consideration for the best split were the square root of all available features, and both utilize the deviance loss-function to be minimized. The candidates differ in the amount of boosting rounds, whereas the first one utilized 400 rounds and the other one 600. The second candidate model outperformed the other one on 80% and 90% genome completeness and was thus selected as the final classification model for predicting a *Cell Shape*.

Genome completeness/ contamination		Precision	Recall	f1	Accuracy
Incomplete genomes	80%	0.9011	0.9155	0.9026	0.9155
	90%	0.9170	0.9278	0.9182	0.9278
	95%	0.9203	0.9303	0.9219	0.9303
	100%	0.9219	0.9313	0.9234	0.9313
Contaminated genomes	100% + 3% contamination	0.8644	0.8478	0.8474	0.8478
	100% + 10% contamination	0.7496	0.3521	0.2965	0.3521
	100% + 15% contamination	0.6599	0.2410	0.1301	0.2410
	100% + 20% contamination	0.4492	0.2118	0.1009	0.2118
Validation		0.8722	0.8917	0.8752	0.8917

Table 7.2: **Evaluation scores – Cell Shape** GBC – max_depth: 7, max_features: sqrt(#features), boosting rounds: 600, loss-function: deviance.

Remarks and discussion Table 7.1 shows the validation results of the final classification model of *Cell Shape*. The selected final gradient boosting classifier shows very good prediction results for complete, incomplete and slightly contaminated genomes in all four evaluation metrics. The very high precision scores in contaminated genomes in comparison to the low recall scores suggest, that less predictions are made, but if they are made, they are still acceptable at a contamination of 15%. The validation shows quite good results for this classifier, thus the trained model is reflecting the real distribution of features in correlation to the prediction targets. Figure 7.2 shows the confusion for the true and predicted labels of the validation set. The diagonal shows, how accurately predicted labels are distributed.

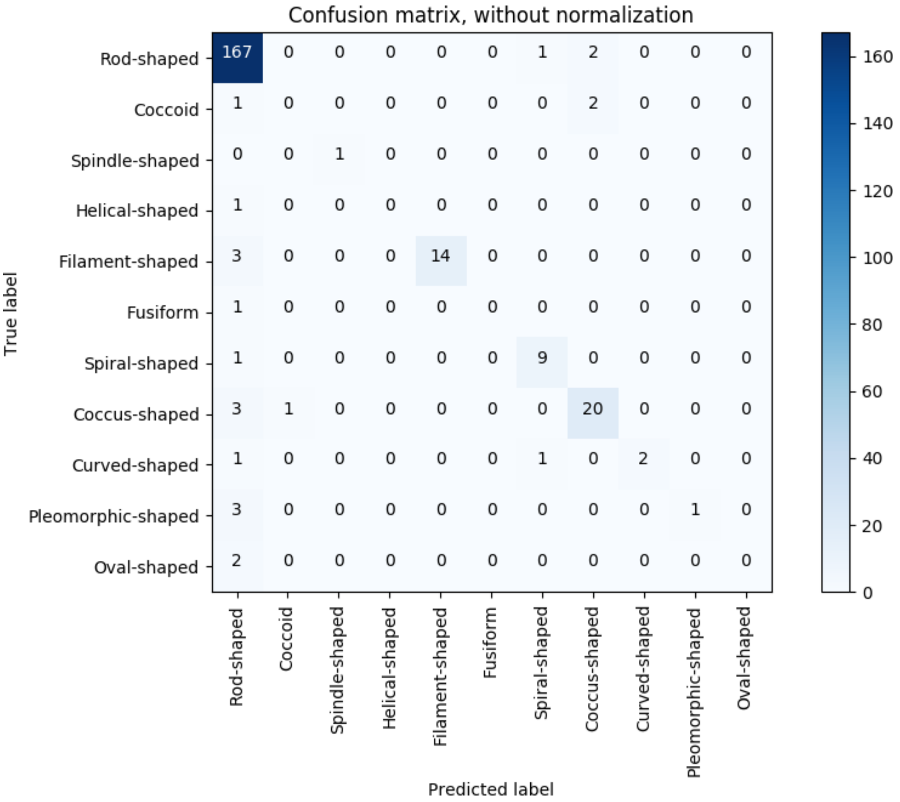


Figure 7.2: **Confusion matrix (validation) – Cell Shape**

The problematic classification targets are `pleomorphic` and `coccoid` labels, but for the latter class two out of three available validation samples have been classified as `coccus-shaped`. Since `coccoid` and `coccus-shaped` describe biologically the same characteristic, a merger of these two classes is beneficial for the whole prediction problem. It must be also noted, that the validation contained 11 out of 25 available classes to be predicted.

7.3 Cell Arrangement

As the shape of a cell, their arrangement is used for taxonomic classification (Madigan et al. 2014). The arrangement can be as `single`, `paired`, `vibrio`, or `tetrads`, to name only a few. It is also possible, that an organism can occur in several different arrangement, depending on the environment or during motility phases (Peruani et al. 2012). In addition to that, their arrangement is often related to their shape, so some coccoid-shaped microbes can also form three-dimensional structures.

Random Forest as classification model The problem class of predicting the *Cell Arrangement* is a multilabel case, because the formation of organisms is not fixed and can change during the lifetime of a bacterial and archaeal cell culture. As candidate models, two Random Forests and one k-Neighbors model could be identified, and were further investigated during cross-validation experiments on degenerated genomes. Both Random Forests, candidate model #1 and candidate model #2, used the entropy-based evaluation function for the decision on how to perform a split in the trees. Candidate model #1 was of size 35 and utilizing bootstrapping strategy for initial feature selection, whereas candidate model #2 was of size 200 and did not used bootstrapping, thus leaving the whole feature set unchanged. Candidate model #3 was a k-Neighbors model that used a distance-based weighting of neighbors, which are set to 5 for decision making of how to classify a sample. Random Forest candidate #1 underperformed in all subsequent incomplete genome cross-validation runs compared to candidate #2, leaving the large Random Forest model as the only competitor of the k-Neigbors approach. Both remaining candidate models performed comparably in that neither outperformed the other in a single setting in all four evaluation scores. The k-Neigbors had for 80% and 90% completeness levels, minimally better f1-scores than the Random Forest model[19], but in terms of accuracy the Random Forest classifier performed better in 95% and 90% complete genomes[20]. In this case, accuracy weighs much more than the f1-score, thus the Random Forest was selected as the final classification model for the category *Cell Arrangement*.

Genome completeness/ contamination		Precision	Recall	f1	Accuracy
Incomplete genomes	80%	0.9406	0.9089	0.9165	0.8489
	90%	0.9485	0.9266	0.9305	0.8707
	95%	0.9506	0.9320	0.9343	0.8764
	100%	0.9501	0.9360	0.9365	0.8794
Contaminated genomes	100% + 3% contamination	0.9495	0.9315	0.9336	0.8757
	100% + 10% contamination	0.9424	0.9179	0.9226	0.8576
	100% + 15% contamination	0.9336	0.9058	0.9107	0.8322
	100% + 20% contamination	0.9243	0.8945	0.8988	0.8043
Validation		0.7299	0.9569	0.8032	0.5172

Table 7.3: **Evaluation scores – Cell Arrangement** Random Forest – no bootstrapping of samples, estimators: 200, split_measure: entropy

Remarks and discussion In relation to the relatively hard case of multilabel classification, the prediction model for the category *Cell Arrangement* performs very well in all cross-validation scenarios of incomplete and contaminated genomes, as shown in Table 7.3. For the validation set, recall is quite high but precision is lacking, thus the model is, frankly, rather "talkative". This also explains the relatively low accuracy score of 51.72% precise predictions of the real labeling, but this does not mean that this classifier is a bad one. The model predicts a little too much, which

[19] *f1-Score* – 0.9326 (k-Neighbors) compared to 0.9285(Random Forest)

[20] *Accuracy* – 86.65% and 86.23% (k-Neighbors) compared to 87.47% and 86.78%(Random Forest)

is better than predicting completely nothing or just wrong. Details about correctly predicted labels of the validation set are given in Table 7.4, where misclassifications are given in red.

#Samples	True annotations	Predicted annotations
1	V-shaped	Singles
1	V-shaped	V-shaped, Singles
4	Singles	Singles, Pairs, Chains
47	Singles	Singles
23	Singles	Singles, Pairs
5	Singles	Singles, Clusters
5	Singles	Singles, Chains
7	Chains	Chains
1	Clusters	Singles
2	Filaments	–
1	Filaments	Chains
4	Filaments	Filaments
9	Pairs	Pairs,Chains
1	Pairs	Pairs, Singles, Chains
4	Pairs	Pairs, Singles
2	Pairs	Pairs

Table 7.4: **Classification summary – Cell Arrangement** Correct predictions are written black, incorrect red.

7.4 Energy Source

Separation of microbes can be performed by exploiting their strategy of obtaining carbon and energy for life and growth (Madigan et al. 2014). The first decision is whether an organism requires light (*photo-*) or chemical compounds (*chemo-*) as the primary energy source for further metabolization of substrates. Depending on the processed chemical compounds or materials as substrates, a further division can be made into processing organic compounds (*organo-*) or inorganic compounds (*litho-*). The last subdivision is based on the source of carbon, which can be organic compounds (*hetero-*) or carbon dioxide (*auto-*). The combination of the three sources of energy, substrates and carbon can be used to specifically define the primary energy source of an organism, e. g. *Thermosulfidibacter takaii* is a chemolithotrophic bacteria found in a hydrothermal field (Nunoura et al. 2008). Some bacteria and archaea are not limited to one metabolic model, meaning they can switch between different ways of obtaining energy depending on environmental conditions such as the presence or absence of carbon, or between substrates processed. These organisms are *facultative* ones, as in the case of *Cellulomonas uda*, which is a facultative anaerobe organism (Poulsen et al. 2016). In contrast to facultative organisms, *obligate* organisms are dependent on the succeeding source or compounds definition e. g., an *obligate aerob* cannot survive in anaerobic environments. The nomenclature used for this predictor is extended with some specific assignments of energy obtaining strategies, e. g. *diazotroph* characterizes nitrogen fixing organisms (Doroshenko et al. 2007), or *oligotrophic* organisms are capable of living in environments offering substrates only at very low concentration level (Choi et al. 2013).

K-Neighbors as classification model After 10-fold cross-validation testing against complete genome sets, the best-performing classification method in all four evaluation metrics was a k-Neighbors model that uses at least two neighbors for decision making on classification output and utilizes a distance-weighted manhattan metric. The scores of the final classifier are shown in Table 7.5, with a corrected generalized scoring for the validation data set explained in the following subsection.

Genome completeness/ contamination		Precision	Recall	f1	Accuracy
Incomplete genomes	80%	0.8017	0.8039	0.8003	0.7785
	90%	0.8094	0.8127	0.8085	0.7859
	95%	0.8071	0.8095	0.8059	0.7845
	100%	0.8159	0.8170	0.8134	0.7876
Contaminated genomes	100% + 3% contamination	0.8076	0.8108	0.8069	0.7859
	100% + 10% contamination	0.8090	0.8112	0.8078	0.7869
	100% + 15% contamination	0.8055	0.8085	0.8045	0.7813
	100% + 20% contamination	0.8084	0.8104	0.8071	0.7859
Validation		0.5676	0.5765	0.5706	0.5588
Generalized validation		0.6324	0.6412	0.6367	0.6291

Table 7.5: **Evaluation scores – Energy Source** K-Neighbors – weights: distance, minkowski_p: 1, neighbors: 2

Remarks and discussion Finding a good predictor for the phenotype category *Energy Source* is a difficult matter, because the underlying problem belongs to the category of multilabel classification. The label set consists of 30 possible labels, where each combination of labels is a valid prediction, thus the theoretical solution space contains $2^{30} = 1,073,741,824$ possible elements. Despite the fact that the final classifier performs consistently well on the test data set in all evaluation scenarios, the validation scores might be surprising. By looking at the assigned annotations (the "real" values) and the predicted values as shown below in more detail, the high level of generality attracts attention e. g., a lot of samples are simply labeled as `Heterotroph`, so describing the primary carbon source and not going into detail about the primary energy or electron source. 48 of such organisms are labeled as `heterotroph` (carbon source) and have been predicted as `chemoorganotroph` (energy/ electron source), thus it cannot be said that this is false in a biological sense, but in ML-logics it is clearly a false prediction. That is why these predictions are marked in red. Predictions where the prediction is more specific than the original labeling or vice versa are marked in yellow e. g., 4 genomes are labeled as `photoautotroph` and the predicted label is `phototroph`, or 2 organisms are labeled as `heterotroph` and the predicted label is an extension of it, `chemoheterotroph`. These generalizations and specifications are thus correct predictions and the corrected generalized validation scores shown in the last row in Table 7.5 as *'Generalized Validation'*. Details about generalized values are shown in Table 7.6, where these are highlighted in yellow.

Improvement of the classification process To improve the predictor, a separation of the contained labels into three distinct phenotype categories, a single one for energy/ electron/ carbon source, and training distinct ML classifiers might offer a way to solve the subdomain-specific classification issues.

Another more reasonable and applicable approach is the setup of a cascading classifier – a stage-by stage procedure where independent predictions are made in each step and passed to the following classifier as secondary information, thereby refining the information content of the decision making process. This proposed method is comparable to a walk through a rooted directed non-cyclic graph representing an ontology, starting at the root going down to the leaves, while information granularity increases concomitantly. This process is guided by separately trained machine learners, thus the final prediction is a concatenation of all the prediction modes independently. The difference to the principle suggested first is training independent ML classifiers so that in a staged approach the subsequent ML method is filled with additional information by the previous ML method, which can be either feature importances or additional newly created features, but this is part a future release of *PhenoPointer*.

7.5 Gram Staining

Gram staining is used to determine whether a prokaryotic cell wall consists of a single thick multilayered peptidoglycan sheath and an inbound cytoplasmic membrane or whether the cell wall is constructed of multiple thin layers stacked upon the inner cytoplasmic membrane (Cabeen et al. 2005). The first ones are gram-positive organisms, since during the gram staining test the

#Samples	True annotations	Predicted annotations
1	Phototroph	Phototroph, Photosynthetic
3	Photoheterotroph	Photoheterotroph
1	Chemolithoautotroph	Chemolithoautotroph
3	Chemolithoautotroph	Chemoorganoheterotroph
1	Facultative autotroph	Lithotroph
1	Heterotroph	–
48	Heterotroph	Chemoorganotroph
65	Heterotroph	Heterotroph
2	Heterotroph	Chemoheterotroph
1	Heterotroph	Heterotroph, Chemoheterotroph
1	Heterotroph	Heterotroph, Chemoorganotroph
1	Chemoorganotroph	Heterotroph
18	Chemoorganotroph	Chemoorganotroph
1	Chemoorganotroph	Chemoorganoheterotroph
1	Chemoorganotroph	Chemolithotroph
1	Chemoorganoheterotroph	Heterotroph
1	Chemoorganoheterotroph	Chemoorganotroph
1	Chemolithotroph	Chemolithotroph
1	Organotroph	Organotroph
1	Methanotroph	Methylotroph
2	Autotroph	Autotroph
1	Autotroph	Lithotroph, Oligotroph
1	Autotroph	Lithotroph, Oligotroph, Chemolithotroph
1	Mixotroph	Mixotroph
1	Diazotroph	Heterotroph
1	Chemoheterotroph	Heterotroph
2	Chemoheterotroph	Chemoorganotroph
1	Chemoheterotroph	Diazotroph
1	Chemoheterotroph	Chemoheterotroph
2	Lithotroph	Lithotroph
4	Photoautotroph	Phototroph

Table 7.6: **Classification summary – Energy Source** Correct predictions are written black, incorrect red, specialization/ generalization yellow.

dye colors the cells to show a positive result, whereas the latter ones are gram-negative as no color change can be observed. The relevance of this cell wall property is that different antibiotics must be applied against an infection of gram-negative and gram-positive bacteria (Singh et al. 2017; Hautala et al. 2005). Not all prokaryotes can be classified as gram-negative and gram-positive organisms, because some show variable behavior related to the gram-staining test and some are gram-indeterminate such as acidfast organisms (Madison 2009), but this is not covered by underlying *GOLD* annotations; therefore the classifier is not capable of classifying organisms as gram-indeterminate or gram-variable.

Random Forest as classification model For classifier selection of *Gram Staining*, a clear best-performing ML method could be found: a random forest classifier utilizing 50 decision trees as base weak learners and the gini function for evaluating the best split among the feature set on an internal node. This classifier with its chosen parameters was best in all four evaluation metrics for a complete genome. The evaluation scores are shown in Table 7.7.

Genome completeness/ contamination		Precision	Recall	f1	Accuracy
Incomplete genomes	80%	0.9806	0.9805	0.9805	0.9805
	90%	0.9812	0.9811	0.9811	0.9811
	95%	0.9817	0.9816	0.9816	0.9816
	100%	0.9837	0.9836	0.9836	0.9836
Contaminated genomes	100% + 3% contamination	0.9815	0.9814	0.9815	0.9814
	100% + 10% contamination	0.8927	0.8871	0.8878	0.8871
	100% + 15% contamination	0.6893	0.6586	0.6642	0.6586
	100% + 20% contamination	0.5131	0.4741	0.4808	0.4741
Validation		0.9882	0.9881	0.9880	0.9880

Table 7.7: **Evaluation scores – Gram Staining** Random Forest – no bootstrapping of samples, estimators: 50, split_measure: gini

Remarks and discussion The chosen classification model generalizes very well, as can be seen from the scores testing the classifier against the validation data set A. The predictor shows excellent predictive performance for partial genomes and contaminated genomes. Performance decreases slightly at a contamination rate of 10% and drops to 51% at a contamination level of 20%. This classifier can be used for incomplete genomes as well as slightly contaminated ones. The absence of a predicted value is not equivalent to gram-indeterminate or gram-variability, because it must be assumed that some organisms are not labeled although their gram-staining is known, but it might be a slight indication towards indeterminability or variability.

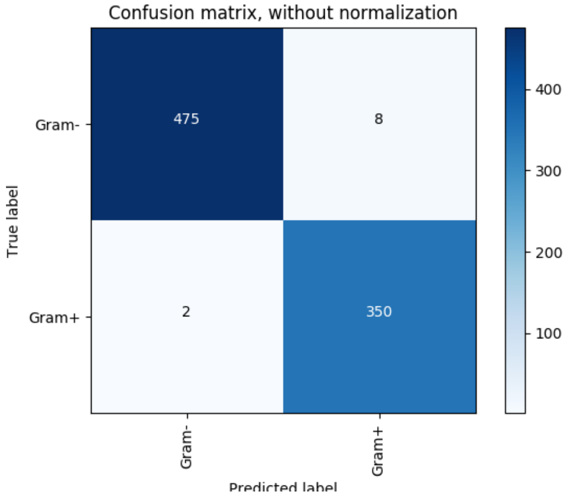


Figure 7.3: **Confusion matrix (validation) – Gram Staining**

7.6 Sporulation

To counteract environmental stress, such as lack of nutrients for energy production, high temperatures and changes in salinity, sporulation is a survival technique solely adopted by bacteria, e. g. as performed by *Bacillus subtilis* (Burbulys et al. 1991; Palop et al. 1999). It has been discovered that under nutrient stress sporulating *B. subtilis* cannibalizes other *B. subtilis* cells by sending out lysis-inducing factors (González-Pastor et al. 2003) inhibiting the forming of spores. This study has been performed in a pure *B. subtilis* culture, but in a mixed culture they prefer predation of cells belonging to other taxa rather than cells of their own kind (Nandy et al. 2007). This classifier predicts whether an organism form spores, but not under which specific environmental conditions.

Random Forest as classification model The final classification model for *Sporulation* could be identified after the 10-fold cross-validation on complete genomes, where a random forest was the best-performing model in all four evaluation metrics compared to all other methods and parameter setups. The forest has a size of 200 underlying trees and measures the split on features on an internal with the gini impurity function.

Remarks and discussion Generalization on hitherto unseen data sets is given, as it can be seen from validation scores. The random forest performs very well on partial genomes and even on strongly contaminated genomes. At a contamination level of 20% predictions are less probable (see recall), but if a prediction is given, it is almost correct as precision indicates.

Genome completeness/ contamination		Precision	Recall	f1	Accuracy
Incomplete genomes	80%	0.9797	0.9798	0.9796	0.9798
	90%	0.9815	0.9816	0.9815	0.9816
	95%	0.9830	0.9830	0.9830	0.9830
	100%	0.9838	0.9838	0.9838	0.9838
Contaminated genomes	100% + 3% contamination	0.9831	0.9830	0.9830	0.9830
	100% + 10% contamination	0.9694	0.9662	0.9670	0.9662
	100% + 15% contamination	0.9134	0.8368	0.8551	0.8368
	100% + 20% contamination	0.8845	0.6959	0.7356	0.6959
Validation		0.9824	0.9821	0.9822	0.9821

Table 7.8: **Evaluation scores – Sporulation** Random Forest – no bootstrapping of samples, estimators: 200, split_measure: gini

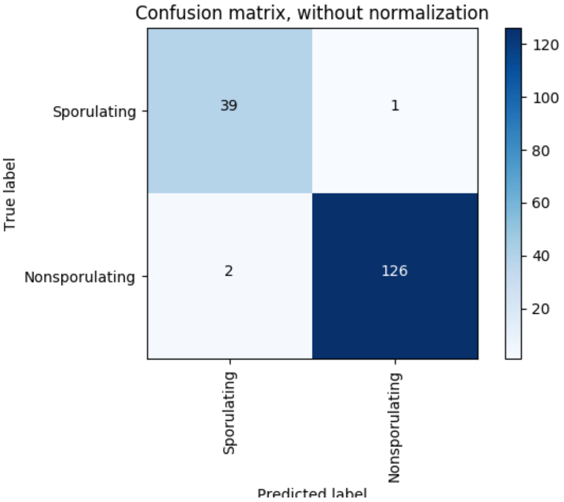


Figure 7.4: **Confusion matrix (validation) – Sporulation**

7.7 Metabolism

As mentioned in Section 7.4, microbiotic organisms are capable of utilizing different sources of energy and carbon, but the underlying metabolic (functional) features are described in detail within this phenotype category. Example phenotype entries are for energy `Pyrrolizidine alkaloids metabolizer`, `Hydrocarbon-oxidizing`, `Iron oxidizer`, `Saccharolytic`, `Polymer degrader`, and `Sulfate reducer`. In addition, metabolite production is also contained as annotation data on which predictions are possible, e. g. `Pristinamycin producer`, `Methanogen`, `Succinic acid production`, and `Stores polyhydroxybutyrate`. Information about metabolic features is also useful for practical application in biotechnology (Osorio-Lozada et al. 2008; Pandey et al. 2007) and bioremediation (Al-Mailem et al. 2017; Matturro et al. 2016) and supported by phenotype entries such as `Oil degrading` and `Hexachlorocyclohexane degrader`.

K-Neighbors as classification model During the 10-fold cross-validation experiments on complete genomes, a k-Neighbor model gave the best scores in all four evaluation metris and was thus selected as the final classifier for *Metabolism*. As the k-Neigbors utilized in the multilabel classification problem class, this model used a distance-weighted manhattan metric to evaluate at least 2 neighbors for final decision making on how to classify an input sample. Because the scores of the validation data set did not look promising enough, as shown in Table 7.11, the 2nd placed models (one Decision Tree, two Random Forests setups) were also tested for contaminated and incomplete genomes for verification purposes, but the three models performed even worse than the k-Neighbors model. All three alternative models were unable to assess the underlying structures and patterns in the feature space and had to be discarded due to lack of generalization.

Genome completeness/ contamination		Precision	Recall	f1	Accuracy
Incomplete genomes	80%	0.6235	0.6252	0.6145	0.5459
	90%	0.6225	0.6263	0.6138	0.5403
	95%	0.6233	0.6282	0.6150	0.5419
	100%	0.6426	0.6436	0.6327	0.5644
Contaminated genomes	100% + 3% contamination	0.6250	0.6314	0.6172	0.5435
	100% + 10% contamination	0.6271	0.6309	0.6180	0.5451
	100% + 15% contamination	0.6289	0.6344	0.6205	0.5491
	100% + 20% contamination	0.6249	0.6303	0.6165	0.5435
Validation		0.4189	0.5385	0.4530	0.3333
Generalized validation		0.5598	0.6667	0.6086	0.4872

Table 7.9: **Evaluation scores – Metabolism** K-Neighbors – leaf_size: 30, weights: distance, minkowski_p: 1, neighbors: 2

Remarks and discussion Training a multilabel classification model to predict the 138 possible labels, with a theoretical amount of 2^{138} label set combinations, is the hardest category within

PhenoPointer to perform ML, because only 1242 annotated had been available for training. Nevertheless a classifier could be found that performed comparably well in all test scenarios of degenerated and contaminated genomes, yielding a very robust classifier. By looking at the generalized validation scores, it can be seen that the recall score is slightly higher than on the training set. So it can be assumed that it generalizes in set boundaries of expressivity and is not overfitting, as the constant evaluation scores demonstrate. This assumption is supported by looking at the detailed generalized predictions made on the validation set, as shown in . For instance, samples annotated with `Thiosulfate reducer` are relatively rare in the training set (1.037%) and the validation sample has been predicted as a `Sulfur reducer`, thus at least a generalized term has been predicted. The same holds for label `Dehalogeniation`, which is equivalent to the predicted label `Dechlorinates Tetrachloroethene`. The summary of all predictions made on the validation set are shown in Table 7.10.

Improvement of classification process The captured metadata within this category has grown over the years since *GOLD*, the data basis of annotation of *IMG/M* and thus of *PhenoPointer*, was released. It captures all terms related to the metabolism of an organism in an internally unrelated way. It mixes up annotations for primary substrates, produced metabolites, metabolic functions and pathways, and use cases for practical applications in the laboratory or industrial environment. Furthermore, the category of metabolites could be further split into two disparate sets, where one contains all produced antibiotics and the other the residual labels.

To achieve this goal, the set of labels has to be strictly screened for relevant terms, following a split of the normalized label set into five distinct metabolism-related categories to finally select suitable ML methods for classification purposes on each of the five subdivided categories.

Another approach takes the opposite direction: to keep the training sample set as it is and change the training and classification process as whole rather than splitting the complete data basis to formulate new metabolism-related categories. Prior to training a supervised ML method, a unsupervised correlation analysis might be of help to identify correlated terms contributing to a prediction target label. After this step, a class weighting matrix can be derived from the unsupervised learning that can be passed into the ML method for training. Combination with an unsupervised ML technique e. g., Frequent Pattern Mining (Aggarwal et al. 2014), a much more robust classifier can be constructed yielding better prediction results.

7.8 Motility

Motility determines whether an organism is capable of moving itself by utilizing energy or whether it stays non-motile at its origin spot. The motility of bacteria and archaea can be divided into several ways of motility such as flagellar-mediated (Macnab 1996; Stock et al. 1996) swarm motility (Harshey 1994), pili-mediated twitching motility (Henrichsen 1983), or gliding motility, which still remains a field of intensive research about the manifold underlying cellular processes in detail (McBride 2001). As for biotic relationships, no further subdivision of motility is provided in the annotation data in *GOLD*, thus the classifier identifies organisms as motile or non-motile, but

#Samples	True annotations	Predicted annotations
2	Nitrate reducer	Nitrate reducer
1	Hydrogen production	Nitrogen fixation
1	Acetogen	Acetogen
1	Acetogen	Iron reducer
1	Sulfate reducer	Sulfate reducer, Thiosulfate reducer
1	Solvent producer	Gelatin hydrolysis negative
1	Nitrogen fixation	Nitrogen fixation, Carbon dioxide fixation
5	Nitrogen fixation	Nitrogen fixation
1	Hydrogenotrophic	Nitrate reducer, Chlorate-reducer, Fatty-acid-oxidizer
2	Methanogen	Methanogen
2	Nitrogen fixer- aerobic	Nitrogen fixer- aerobic
2	Nitrogen fixer- aerobic	Nitrogen fixation
2	Cellulose degrader	Cellulose degrader, Ethanol production, Ethanogenic
1	Fatty-acid-oxidizer	Chlorate-reducer, Fatty-acid-oxidizer
1	Fatty-acid-oxidizer	Nitrate reducer
2	Fatty-acid-oxidizer	Nitrate reducer, Chlorate-reducer, Fatty-acid-oxidizer
1	Fatty-acid-oxidizer	Nitrogen fixation
1	Carbon dioxide fixation	Acetogen, Carbon dioxide fixation
1	Dehalogenation	Dechlorinates Tetrachloroethene
1	Sulfur metabolizing	Acetate oxidizer
1	Sulfur metabolizing	Iron oxidizer
1	Iron reducer	Iron reducer
1	Iron reducer	Acetogenic
1	Saccharolytic	Cellulose degrader, Ethanol production, Acetate producer
1	Chitin degradation	Sulfur respiration
1	Prototrophic	Homofermentative
1	Thiosulfate reducer	Sulfur reducer
1	Sulfur oxidizer	Nitrogen fixation
1	Polymer degrader	Nitrogen fixation

Table 7.10: **Classification summary – Metabolism** Correct predictions are written black, incorrect red, specialization/ generalization yellow.

in combination with annotations (`Polar flagella`, `Gliding`) from the *Phenotype* classification category, see Section 7.10 on page 89, the specific type of motility can be deduced.

Random Forest as classification model As in the case of *Gram Staining* and *Sporulation*, a clear winner as best-performing classification model could be found in the 10-fold cross-validation on complete genomes. The identified ML method is a random forest consisting of 600 decision trees and utilizing the entropy function, describing the information gain, as the evaluation function for division of features for a split on internal nodes. The evaluation scores among all contaminated and incomplete genome data sets is shown in Table 7.11.

Genome completeness/ contamination		Precision	Recall	f1	Accuracy
Incomplete genomes	80%	0.9357	0.9364	0.9358	0.9364
	90%	0.9374	0.9381	0.9375	0.9381
	95%	0.9402	0.9408	0.9402	0.9408
	100%	0.9400	0.9405	0.9400	0.9405
Contaminated genomes	100% + 3% contamination	0.9394	0.9399	0.9394	0.9399
	100% + 10% contamination	0.9371	0.9373	0.9367	0.9373
	100% + 15% contamination	0.9350	0.9348	0.9342	0.9348
	100% + 20% contamination	0.9274	0.9260	0.9253	0.9260
Validation		0.9131	0.9175	0.9152	0.9175

Table 7.11: **Evaluation scores – Motility** Random Forest – no bootstrapping of samples, estimators: 600, split_measure: entropy

Remarks and discussion The selected random forest classifier shows constantly very good predictive performance in all evaluation scenarios, where all metrics show scores > 90%. The results of the validation against data set A supports this assumption. Figure 7.5 shows the confusion matrix of the true and predicted labels of the validation set. Samples labeled as `chemotactic` are a hard case to predict because a total of 9 training samples are labeled respectively, whereas > 3700 training samples are labeled as `nonmotile` and > 3900 as `motile`. It must be monitored as to whether this class makes sense if no additional organisms are labeled accordingly in the future to enforce predictive strength over this specific class.

7.9 Oxygen Requirement

Separate annotations about oxygen requirement of organisms, apart from the primary energy source nomenclatures in 7.4 – Energy Source, is given with this predictive model. It contains only information about `aerobic`, `anaerobic`, `microaerophilic`, `facultative` and `obligate` lifestyles of microorganisms.

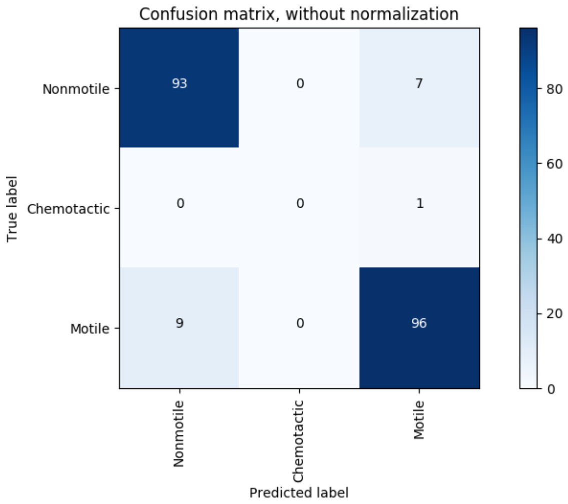


Figure 7.5: **Confusion matrix (validation) – Motility**

Random Forest as classification model In the case of *Oxygen Requirement* as the prediction target, a multiclass classification problem needs to be solved. Two random forest classifiers emerged from the cross-validation comparison, one utilizing 20 base estimators in combination with the entropy function for split evaluation among available features and another utilizing 35 base estimators with gini function as split evaluation measure. The first one performed best for 90% complete genomes, whereas the second performed best on completeness levels of 80% and 95%, and therefore the second was used for training the final classifier.

Genome completeness/ contamination		Precision	Recall	f1	Accuracy
Incomplete genomes	80%	0.8254	0.8346	0.8179	0.8346
	90%	0.8450	0.8489	0.8368	0.8489
	95%	0.8516	0.8542	0.8439	0.8542
	100%	0.8708	0.8724	0.8652	0.8724
Contaminated genomes	100% + 3% contamination	0.8492	0.8508	0.8416	0.8508
	100% + 10% contamination	0.7981	0.8012	0.7919	0.8012
	100% + 15% contamination	0.6968	0.7006	0.6912	0.7006
	100% + 20% contamination	0.6064	0.6121	0.6007	0.6121
Validation		0.9055	0.8983	0.8963	0.8983

Table 7.12: **Evaluation scores – Oxygen Requirement** Random Forest – no bootstrapping of samples, estimators: 35, split_measure: gini

Remarks and discussion As can be seen from Table 7.12, the final classifies showed good prediction performances on partially complete genomes and still reasonably good results on higher contamination levels of 15% and above. Interestingly, the classifier performs better on the validation set than on the initial training samples (see Figure 7.6), which confirms the assumption that this model generalizes very well.

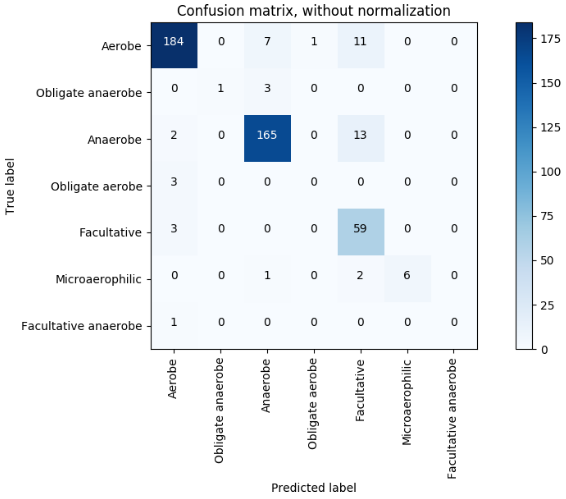


Figure 7.6: **Confusion matrix (validation) – Oxygen Requirement**

7.10 Phenotype

The *Phenotype* category is a conglomeration of organisms' specific microbial traits, whose entries explain phenotypes in general or that do not fit in any other of the twelve predictive phenotype categories. Accumulated traits are the preferred pH-value of an organism (Hankinson et al. 1988), e. g. `Acidophile`, `Alkaliphile`, or `Neutrophilic`, resistance and reduction of metals (Dopson et al. 2003; Wright et al. 2016), distinction between pathogens and non-pathogenic organisms (Rossi et al. 2013), antibiotic resistance (Poole 2001), or growing speed, to mention only a few entries of subcategories gathered in this category.

K-Neighbors as classification model As for classifications made in the category *Energy Source*, a k-Neighbors model could be directly found in the cross-validation experiments on complete genomes as the best-performing machine learner among all other methods and setups for predicting labels belonging to the more general category names as *Phenotype*. The parameters of the ML method are identical to that used in *Energy Source*, i. e. a distance-weighted manhattan metric for evaluating neighbors in the surrounding area in the feature space, where at least 2 neighbors are taken into consideration for decision making on how to classify a hitherto unseen sample.

Remarks and discussion The classification problem class of training and predicting on the *Phenotype* category is very hard, as it is a multilabel problem where the theoretical prediction target space consists of 2^{132} sets of labels. The selected method for final training on the training data results in a robust classification model that performs well on all levels of contamination and completeness. The generalization aspect is supported by the validation scores, where the generalized validation, as explained in Section 7.4 on page 79, shows better results than on complete genomes of the training set used in the 10-fold cross-validation experiment. This effect possibly emerges from the relatively small subset of labels used for annotation of the organisms contained in the validation set, where 18 out of 132 potential prediction target labels have been

Genome completeness/ contamination		Precision	Recall	f1	Accuracy
Incomplete genomes	80%	0.7723	0.7678	0.7625	0.7029
	90%	0.7724	0.7669	0.7625	0.7056
	95%	0.7716	0.7665	0.7620	0.7065
	100%	0.7890	0.7882	0.7824	0.7327
Contaminated genomes	100% + 3% contamination	0.7706	0.7654	0.7608	0.7049
	100% + 10% contamination	0.7728	0.7661	0.7625	0.7079
	100% + 15% contamination	0.7722	0.7662	0.7622	0.7088
	100% + 20% contamination	0.7714	0.7650	0.7609	0.7040
Validation		0.7529	0.7931	0.7662	0.7126
Generalized validation		0.7989	0.8391	0.8185	0.7586

Table 7.13: **Evaluation scores – Phenotype** K-Neighbors – weights: distance, minkowski_p: 1, neighbors: 2

used. But it can still be said that the model generalizes fairly pretty well (see Table 7.13 and Table 7.14), because rare labels, such as `Polar flagella` and `Saprophyte`, have been successfully predicted by the classifier.

Improvement of the classification process Since this metadata category has evolved over the last years in *GOLD*, some of the information gathered can be outsourced into its own category for classification. One example would be the distinction between pathogens and non-pathogens, because for these labels a reasonable large number of organisms are explicitly annotated and can thus be extracted from the category *Phenotype*. The same holds for catalase activity, and probably for antibiotics resistance. The latter is dependent on an external data source to enlarge the knowledge of *MetaStone*. The CARD (Comprehensive Antibiotic Resistance Database) (McArthur et al. 2013; Jia et al. 2017) can be seen as a foundation for labeling the *MetaStone* genomes accordingly, but this involves downloading the genomic sequences of the *IMG/M* genomes, the main data source of *MetaStone*, and then further processing with the CARD models.

7.11 Salinity

Microbial growth and vitality is strongly correlated with levels of salinity, describing the osmotic pressure as an environmental condition (Wood 2015). Salinity measures are not limited to marine environments; soil can also carry high concentrations of salt (Dion et al. 2008). The nomenclature of halotolerance encompasses several partially overlapping levels of salinity concentrations an organism can withstand, e. g. *halophilic* microbial organisms favor very high salinity concentrations (Ventosa et al. 1998; Oren 2002), *halotolerant* organisms prefer high salinity (Dahal et al. 2017) for optimal growth but are not dependent on high salt concentrations for survival, whereas *euryhaline* organisms are capable of living in a wide range of salinity concentrations (Alvensleben et al. 2013). In the scientific and industrial context, bioreactors used in biotechnology can benefit

#Samples	True annotations	Predicted annotations
2	Symbiont	Symbiont
1	Symbiont	Non-Pathogen
1	Acidophile	Acidophile
2	Acidophile	Catalase positive
1	Acidophile	Alkaliphile
1	Acetic-acid	Acetic-acid
2	Thermoacidophile	Acidophile
1	Gliding	Gliding, Catalase negative
9	Non-Pathogen	Non-Pathogen
1	Non-Pathogen	Pathogen
1	Non-Pathogen	Pathogen, Highly Virulent
6	Catalase negative	Catalase negative
2	Catalase negative	Non-Pathogen
1	Catalase negative	Catalase negative, Alpha-hemolytic
2	Catalase negative	Pathogen
1	Metal-mobilizing	Alkaliphile
1	Opportunistic Pathogen	Pathogen, MDR (Multi-drug resistant)
1	Opportunistic Pathogen	Opportunistic Pathogen, Catalase positive
1	Polar flagella	Polar flagella, Pathogen
1	Saprophyte	Saprophyte, Non-Pathogen
1	Non-Haemolytic	Non-Haemolytic, Catalase negative
1	Pathogen	Pathogen, Alpha-hemolytic
1	Pathogen	Beta-hemolytic, Catalase negative
35	Pathogen	Pathogen
1	Catalase positive	Non-Pathogen
1	Catalase positive	Pathogen
8	Catalase positive	Catalase positive
1	Biofilm	Catalase positive

Table 7.14: **Classification summary – Phenotype** Correct predictions are written black, incorrect red, specialization/ generalization yellow.

from *extremophiles*, because under such environmental conditions i. e., high salinity concentration, the stability of produced biomolecules is increased and production running costs can be reduced because unpurified basal mediums can be used (Margesin et al. 2001).

Random Forest as classification model During cross-validation one candidate ML method with two different setups could be the identifier: a random forest classifier utilizing the gini measure for evaluation of how to perform splits on internal nodes among the given feature space with two different sizes, one with 100 decision tress and the other one with 200. The smaller forest outperformed the larger one on completeness levels of 90% and 95% and showed greater stability in shown accuracy. The forest consisting of 200 trees drops in accuracy from 83.64% instantly to 81.17% and from 81% to 79% in the case of the f1-score on a genome completeness of 95% and remains on these levels on further degeneration of test genomes even for 80% completeness, whereas the decrease in accuracy and the f1-score for the smaller tree is not that steep. Because of the overall better performance and smoothness of prediction loss, the smaller tree was chosen as the final classifier for *Salinity*.

Genome completeness/ contamination		Precision	Recall	f1	Accuracy
Incomplete genomes	80%	0.7927	0.8159	0.7787	0.8159
	90%	0.7847	0.8205	0.7883	0.8205
	95%	0.7903	0.8204	0.7884	0.8204
	100%	0.8137	0.8342	0.8111	0.8342
Contaminated genomes	100% + 3% contamination	0.7961	0.8249	0.7999	0.8249
	100% + 10% contamination	0.7718	0.7950	0.7783	0.7950
	100% + 15% contamination	0.7470	0.7195	0.7124	0.7195
	100% + 20% contamination	0.7298	0.6550	0.6532	0.6550
Validation		0.8403	0.8824	0.8533	0.8824

Table 7.15: **Evaluation scores – Salinity** Random Forest – no bootstrapping of samples, estimators: 100, split_measure: gini

Remarks and discussion In general the chosen ML classifier shows good prediction performance, with little influence of missing features for partially complete genomes, as shown in Table 7.15. The same observation holds for contaminated genomes for levels of up to 10% contamination. The model is generalizing, as can be seen from validation scores, where it actually performs better than on complete genomes of the training set. This might arise from the fact that the validation set is quite small, with only 14 genomes available. Misclassification of the single available sample labeled as `euryhaline` as `halophile` sounds reasonable in that sense, that this organism tolerates different levels of salinity. One label of total available class is missing in the validation set, `stenohaline`; for this class no assertion related to generalization can be made. In the training set, `stenohaline` and `euryhalione` are the least populated classes, as a consequence these two classes are hard cases to predict and should be monitored in future releases of *PhenoPointer*, if newly annotated genomes are available through *IMG/M*.

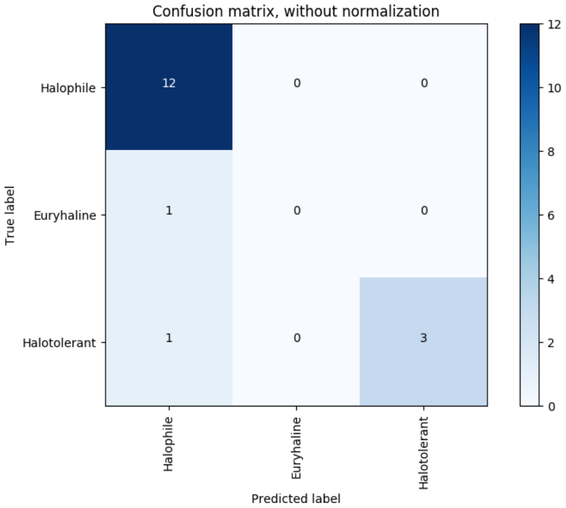


Figure 7.7: **Confusion matrix (validation) – Salinity**

7.12 Temperature Range

Besides human-habitable environments which are mesophilic, bacteria and archaea can actually flourish in environments with very low and extremely high temperatures. Such extremophiles are adapted to a certain temperature range, possessing special membranes (Siliakus et al. 2017; Schouten et al. 2013) and temperature-adapted enzymes (D'Amico et al. 2006; Laksanalamai et al. 2004). The psychrophilic organisms *Moritella profunda* and *Moritella abyssi* posses maximum growth rates at temperatures of 2°C for *M. profunda* and 4°C for *M. abyssi* (Xu et al. 2003), whereas organisms adapted to temperatures above 40°C are thermophils and those with optimal growth rates above 80°C are hyoperthermophils (Antranikian et al. 2017).

GBC as classification model For the phenotype category of *Temperature Range* for optimal growth, several candidate ML methods could be identified during cross-validation on complete genomes used as test sets. Candidate #1 is a Gradient Boosting Classifier minimizing the deviance loss function in 600 rounds of boosting, trees as base weak learners with a maximum depth of 7 levels and utilizing at maximum $\log_2$ of available features for decision making on how to perform a splits on internal tree nodes. One k-Neighbors Classifier could also be identified as a candidate model, where the number of neighbors was set to 5 and utilizing the distance-based weight function for samples contribution to the final label decision. In addition, two different Random Forest classifiers could also be identified, but both candidate models could be discarded in the cross-validation runs on degenerated genomes because they underperformed on all completeness levels in comparison to the other candidate models. The GBC-based approach outperformed the remaining k-Neighbors classifier in the case of 90% complete genomes and gave comparable results in the two other completeness levels.

Remarks and discussion The cross-validation results of the selected GBC model are almost constant across all levels of contamination and degeneration. In addition, the scores of the val-

Genome completeness/ contamination		Precision	Recall	f1	Accuracy
Incomplete genomes	80%	0.9380	0.9563	0.9438	0.9563
	90%	0.9498	0.9621	0.9527	0.9621
	95%	0.9522	0.9628	0.9540	0.9628
	100%	0.9553	0.9640	0.9558	0.9640
Contaminated genomes	100% + 3% contamination	0.9529	0.9631	0.9547	0.9631
	100% + 10% contamination	0.9490	0.9596	0.9516	0.9596
	100% + 15% contamination	0.9377	0.9446	0.9384	0.9446
	100% + 20% contamination	0.9300	0.9163	0.9200	0.9163
Validation		0.9338	0.9397	0.9238	0.9397

Table 7.16: **Evaluation scores – Temperature Range** GBC – max_depth: 7, max_features: $\log_2$(#features), boosting rounds: 400, loss-function: deviance.

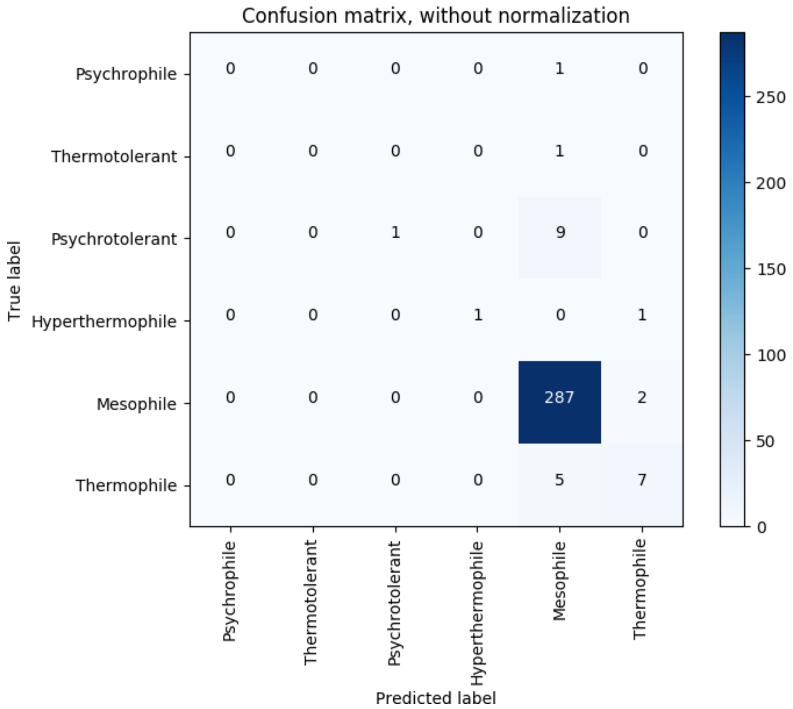


Figure 7.8: **Confusion matrix (validation) – Temperature Range**

idation set show that this model generalizes pretty well. Because of the overall consistent scores, the final *Temperature Range* classifier is very robust when confronted with noisy and fragmented feature sets.

7.13 Diseases

Bacterial infections often cause or correlate with primary diseases of eurykaryotes, or accompanying viral infections as a secondary disease and subsequently causing a primary one. Tuberculosis is one of the best-known bacterial diseases of humans and animals caused by *Mycobacterium tuberculosis* (Gomez et al. 2004). A severe illness can also be caused by passive disease-carrying parasites like ticks, causing meningitis *Neisseria meningitidis* (D. S. Stephens et al. 2007) or African tick-bite fever *Rickettsia africae* (Portillo et al. 2007). But microbial pathogens are not only limited to invertebrates and insects; they can also affect plants, such as a wilt disease caused by *Ralstonia solanacearum* (Patil et al. 2017) in more than 200 cultivated plants such as tomatoes or potatoes. As a matter of fact the range of correlated diseases is not limited to one, thus multiple diseases can be predicted with this classifier.

K-Neighbors as classification model Three candidate models, two different Random Forest classifiers and one k-Neighbors, have qualified as final classifiers for predictors in the category *Diseases* upon all other models and parameter setups. The k-Neighbors model was trained with the same setup, as for the majority of multiclass classifiers, by looking at least at 2 neighbors using a distance-weighted manhattan score as a decision metric for selecting a final prediction on an input sample, where the Random Forest differed only in their size. The first had a size of 100 decision trees as weak base learners and the second had a size of 600 trees, but both had the entropy-based split decision function.

In the partial completeness test cases, no model performed best at levels of 90% and 95%. In the 80% test case the k-Neighbors model outperformed both Random Forests in all four evaluation metrics, but by just a small margin. Because this result does not seem reliable enough, the evaluation scores based on the validation set were screened as a safety precaution measure, and here the k-Neighbors again outperformed both Random Forest models, thus the k-Neighbors model confirmed its performance and was finally selected as the classification model for *Diseases*. The evaluation scores are shown in Table 7.17.

Remarks and discussion Although the classification problem is the hardest within *Pheno Pointer*, the selected model performs quite well. The possible classification target space consists of 2^{332}, caused by the more or less uncontrolled vocabulary of the metadata annotations of this category within *GOLD*. Despite the high complexity of prediction targets and their combination as a classification result, the validation scores for contaminated and degenerated genomes are consistent over all levels and imply a satisfactory predictive performance of the final classifier. The shown generalized scores on the validation data set follow the same methodology as described in Section 7.4 on page 79, resulting in validation scores as if the vocabulary had been

Genome completeness/ contamination		Precision	Recall	f1	Accuracy
Incomplete genomes	80%	0.7486	0.7505	0.7367	0.6422
	90%	0.7554	0.7581	0.7443	0.6515
	95%	0.7537	0.7592	0.7444	0.6520
	100%	0.8145	0.8177	0.8063	0.7423
Contaminated genomes	100% + 3% contamination	0.7561	0.7613	0.7465	0.6546
	100% + 10% contamination	0.7570	0.7623	0.7477	0.6567
	100% + 15% contamination	0.7533	0.7592	0.7441	0.6518
	100% + 20% contamination	0.7539	0.7577	0.7438	0.6526
Validation		0.5567	0.7091	0.5982	0.4545
Generalized validation		0.7442	0.8545	0.7956	0.6364

Table 7.17: **Evaluation scores – Diseases** K-Neighbors – leaf_size: 30, weights: distance, minkowski_p: 1, neighbors: 2

a controlled one. Based on the generalized validation scores, the performance on unseen data sets, novel genomes, is adequate in relation to the cross-validation results and thus the model can generalize in the set boundaries of limited expressibility of the underlying prediction target space, as described in the following subsection 7.13. The only criticism that can be made on the final classifier for predicting annotations on the category *Diseases* is that it is not accurate enough, as the accuracy scores show. More than 85% of true annotations could be recalled, but unfortunately in combination with some false positive results, so the researcher should take the predictions of this phenotype classifier more as suggestions rather than as results representing the ground-truth. It should be also mentioned that out of the label set (size of 332), on which the classifier has been trained, only 36 were available in the validation set. The prediction results on the validation set are shown in Table 7.18.

Improvement of the classification process To improve predictions made on this phenotype category, the metadata annotation basis needs to be modified; the granularity of the vocabulary is too fine on the one hand and the overall umbrella of terminology used for contained labels is too unspecific on the other hand.

The terms used in this category mirror the domain of expertise of the researchers by using their own nomenclature to annotate the organisms e. g., some seldom-used medical terms and other more common ones such as 'Endocarditis' vs. 'Heart disease' and 'Gastroenteritis' vs. 'Gastrointestinal disease'. The kind of information is also widespread in the sense of domain-specificity e. g., the metadata annotations are a mixture of terms describing a particular disease, disease implicating symptoms, and general observations related to a disease, like 'Plague-like illness' or 'Food poisoning'. Another aspect of mixed domains is that the described host-associated diseases are an intermix of different taxa i. e., *Animalia* and *Plantea*.

To implement these changes, the vocabulary used in *GOLD* has to be translated into three consistent vocabularies at taxonomy level, where each one reflects the affected host, accordingly defined as *Homo sapiens* (Species), *Animalia* (Kingdom) excluding *Homo s.*, and *Plantea* (Kingdom). For this purpose, terms have to be subdivided into the three distinct sets of host-related diseases as described. In each host-related vocabulary synonymous terms have to be consolidated and further structured into hierarchical categories, where a differentiation is made according to affected organs, induced symptoms, and determination of disease-causing effects, like `Food poisoning` or `Tick-borne`. The hierarchical structure here described is comparable to a host-specific ontology of diseases, allowing the use of a staged classification approach similar to that described in '7.4 – Energy Source, Improvement of the classification process' on page 79 for separate training of host-specific disease classification models. The need of such ontologies has been realized by the scientific community and accordingly implemented, such as the *Disease Ontology* for human-related diseases (Schriml et al. 2012; Kibbe et al. 2015), the *Animal Disease Ontology* focussed on animals (Faure et al. 2011), and the *Plant Disease Ontology* handling plant diseases (Walls et al. 2012). These ontologies are used in biomedical data analysis and drug discovery repositories, especially on human-related diseases (Sarntivijai et al. 2016; Peng et al. 2017; Ursu et al. 2017).

The subdivision of *GOLD* annotations, categorization, mapping of consolidated terms onto a given ontology, and possible extension with new so far missing but valuable terms requires expert domain knowledge – medical researchers, veterinarians, plant physiologists – specializing in the field of microbial-induced and microbial-related diseases. A permanent screening process of metadata annotations and ontology mappings is crucial for a high quality data set and ML classification models based on these, especially in application fields with a low error tolerance level, such as human-related disease risk assessment in medicine.

7.14 Runtime and memory consumption

The memory consumption during a cross-validation experiment and final training of a classifier depends on the selected ML method and the phenotype category. In case of the phenotype category, *Gram Staining* as the largest set consumes up to 3GB of memory, whereas *Salinity* as the smallest set takes up to 130MB of memory. Related to the ML method it can be said, the Random Forest and k-Neighbors approach may take up to 12GB memory temporarily and mark the upper memory consumption of all evaluated ML classification methods.

Runtime depends strongly on the selected ML method and chosen set of parameters. Whereas Random Forests can be trained in parallel and performs quite well on a multi-core machine, a gradient boosting classifiers performs sequential boosting and thus does not scale. The expected runtime of gradient boosting classifiers with 600 boosting rounds may take ~ 24 hours. The same holds for support vector machine-based classifiers, where runtime for training can take several days depending on the phenotype category and selected kernel.

Classification is performed quite fast; after Pfam prediction of an input organism (2 min.), the additional runtime for making phenotype predictions is $10-15$ seconds. The memory consumption is also quite small at a max of ~ 1.5 GB.

#Samples	True annotations	Predicted annotations
1	Food poisoning	Food poisoning
1	Food poisoning	Food poisoning, Cholera, Diarrhea
6	Mastitis	Diarrhea
1	Mastitis	Food poisoning
1	Mastitis	Opportunistic infection, Cepacia syndrome
1	Gas gangrene	Botulism
1	Opportunistic infection	Opportunistic infection
2	Periodontal infection	Periodontal infection
1	Nosocomial infection	Nosocomial infection
2	Leptospirosis	Leptospirosis
1	Plant rot	Pepper spot
1	Plant rot	Wilting disease, Tuber rot, Ring rot
1	Lung infection	Pneumonia, Legionellosis
1	Tularemia	Tularemia
1	Sotto disease	Sotto disease
1	Keratoconjunctivitis	Keratoconjunctivitis
1	Brucellosis	Brucellosis, Infectious abortions, Fever
1	Ulcer	Ulcer, Gastric inflammation
1	Respiratory disease	Respiratory infection, Genital infection
1	Respiratory disease	Respiratory infection, Urogenital infection
1	Diarrhea	Diarrhea, Peritonitis, Colitis
1	Diarrhea	Gastroenteritis
1	Bacterial spot	Bacterial spot
1	Respiratory infection	Pneumonia
1	Pneumonia	Pneumonia, Meningitis, Otitis media
1	Pneumonia	Pneumonia, Bacteremia, Melioidosis
2	Pneumonia	Pneumonia
1	Urinary tract infection	Urinary tract infection, Pneumonia
2	Gastroenteritis	Gastroenteritis
1	Gastroenteritis	Gastroenteritis, Food poisoning
1	Septicemia	Septicemia, Meningitis, Food poisoning, Listeriosis, Abortion
1	Septicemia	Septicemia, Meningitis, Pneumonia, Mastitis
1	Septicemia	Septicemia, Meningitis
1	Septicemia	Septicemia, Pneumonia
2	Toxic-shock syndrome	Toxic-shock syndrome, Staphylococcal scarlet fever
4	Tuberculosis	Tuberculosis
1	Soft tissue lesions	Soft tissue lesions
5	Gonorrhea	Gonorrhea

Table 7.18: **Classification summary – Diseases** Correct predictions are written black, incorrect red, specialization/ generalization yellow.

8 *PhenoPointer* – Application on a real world data set and comparison to a competitor

To test the prediction performances of *PhenoPointer* on real world data sets not contained in *IMG/M*, 22 isolate genomes identified in several biogas producing anaerobic digesters were characterized by their microbial traits and observable phenotype described in the 13 phenotype categories. These 22 organisms might have been available in *IMG/M* in December 2016, when the comparison described here was performed, but were not annotated respectively in the selected metadata categories used for classification targets. Thus they may have been annotated in the data set A that was used for validation in the previous chapter, but this does not influence the results of this comparison because the evaluation of predictions was performed manually by an external sovereign domain expert. As a competitive tool, *Traitar* (Weimann et al. 2016) was executed with its sets of machine learner classifiers on the genomes and is compared against *PhenoPointer*.

The 22 organisms to be characterized by both tools were sampled from mesophilic and thermophilic biogas producing anaerobic digesters. Each organism's isolate genome was assessed through WGS amplicon sequencing and have been phenotypically and functionally described in almost full detail. The isolates' genome sequences were published in a time frame from 2012 to 2016. Tests were performed as wet lab experiments to detect gram staining, catalase activity, or sporulation capability. Table 8.1 gives an overview of the characterized organisms, related publications, EBI accession ids, and temperature range of the sampled digester. A publication reference is given where available.

The authors of *Traitar* speak exclusively of traits to be predicted, since the classification target classes are focused on utilization of substrates for gaining energy and carbon, antibiotic susceptibility, proteolysis, and some observable phenotypes. In comparison to *PhenoPointer*, the main goal is to predict consumed substrates and metabolic reactions utilizing these, whereas in *Pheno Pointer* a broader view into phenotypes and metabolic features is presented. As features, *Traitar* also employs Pfam abundance profiles, on which the classification is performed. It features a very interesting approach of presenting candidate protein families associated with a predicted trait.

Traitar utilizes exclusively SVMs in a binary classification fashion for each trait, and even here binary states are predicted independently, so there exist two distinct trait predictors for gram staining: one responsible for predicting gram negative, another for gram positive. This might support the classification of gram variable or indeterminate organisms, but remains to be elucidated. Another example for ambiguous predictions that might cause confusion is the style of predicting cell shapes, where only two shapes can be predicted independently in contrast to *PhenoPointer*, where 25 cell shapes are distinguished and only one of these is the predicted class

Organsism	**Publication**	**EBI accesion id**	
Clostridium bornimense M2/40^{T}	(Tomazetto, Hahnke, Koeck et al. 2016)	HG917868, HG917869	mesophilic
Proteiniborus sp. DW1[1]	(Maus, Koeck et al. 2016)	FMDO01000001-FMDO01000062	
Sporanaerobacter sp. PP17-6a	(Maus, Koeck et al. 2016)	FMIF01000001-FMIF01000053	
Peptoniphilus sp. ING2-D1G	(Tomazetto, Hahnke, Maus et al. 2014)	LM997412	
Propionispora sp. 2/2-37	(Koeck, Maus et al. 2016b)	CYSP01000001-CYSP01000043	
Proteiniphilum saccharofermentans M3/6^{T}	(Maus, Koeck et al. 2016)	LT605205	
Fermentimonas caenicola ING2-E5B^{T}	(Hahnke et al. 2016)	LN515532	
Petrimonas mucosa ING2-E5A^{T}	(Maus, Koeck et al. 2016)	LT608328	
Methanobacterium formicicum MF^{T}	(Maus, Stantscheff et al. 2014)	LN515531	
Methanobacterium formicicum Mb9	(Maus, Koeck et al. 2016)	LN734822	
Methanobacterium sp. Mb1	(Maus, Wibberg, Stantscheff, K. Cibis et al. 2013)	HG425166	
Methanobacterium congolense Buetzberg	(Maus, Koeck et al. 2016)	LT607757, LT607757	
Methanoculleus bourgensis MS2^{T}	(Maus, Wibberg, Stantscheff, Eikmeyer et al. 2012)	HE964772	
Methanoculleus chikugoensis L21-II-0	(Maus, Koeck et al. 2016)	FMID01000001-FMID01000070	
Clostridium cellulosi DG5	(Koeck, Wibberg, Maus, Winkler, Albersmeier, Zverlov, Liebl et al. 2014)	LM995447	thermophilic
Clostridium sp. N3C	(Maus, Koeck et al. 2016)	FMJL01000001-FMJL01000109	
Ruminiclostridium thermocellum BC1	(Maus, Koeck et al. 2016)	CBQO010000001-CBQO010000139	
Herbinix hemicellulosilytica T3/55^{T}	(Koeck, Maus et al. 2015)	CVTD020000001-CVTD020000035	
Herbinix luporum SD1D^{T}	(Koeck, Maus et al. 2016a)	LN879430	
Bacillus thermoamylovorans 1A1	(Koeck, Wibberg, Maus, Winkler, Albersmeier, Zverlov, Pühler et al. 2014)	CCRF01000001-CCRF01000106	
Defluviitoga tunisiensis L3	(Maus, K. G. Cibis et al. 2015)	LN824141	
Methanothermobacter wolfeii SIV6	(Maus, Koeck et al. 2016)	LT608329	

Table 8.1: **22 isolate genomes for comparison of *PhenoPointer* against *Traitar*** The prediction performance of *PhenoPointer* has been evaluated on 22 isolated genomes representing a real world data set and compared to predictions of *Traitar*.

label. *Traitar* is capable of making predictions on 67 phenotypes, but this number of predictable traits is accounted for by the binary classification nature of the applied SVM, so the real amount of predictable phenotypes is lower. At a first glance the four phenotypes describing the oxygen requirement can be summarized, as well as the two classifiers for gram staining, the two classifier describing the cell shape, and finally the two classifiers cell arrangement as clusters and pairs/ chains. 57 singleton traits sounds more reasonable. Training and evaluation has been performed following a 10-fold cross validation principle on 234 annotated bacterial organisms contained in the Global Infectious Disease and Epidemiology Online Network (GIDEON) database (Berger 2005), a commercial closed-source software product, and so no evaluation of the annotated genomes used within *Traitar* and screening of the contained metadata fields could be performed. A classifier consists of five independent SVMs as an ensemble, where the final classification output is given by a majority decision function over all five predictions.

8.1 Comparison of *PhenoPointer* and *Traitar*

Directly comparable are five traits/ observable phenotypes, that are common on both tools: *Gram Staining*, *Sporulation*, *Motility*, *Oxygen Requirement*, *Cell Arrangement*, and *Cell Shape*. *Traitar* is not capable of making predictions about *Biotic Relationships*, *Temperature Range*, *Energy Source*, and *Salinity*. The phenotype category *Diseases* has been omitted, because there is no reference in the literature known about the organisms upon which the comparison is made.

A fragmented comparison can be made on the categories *Phenotype* and *Metabolism*, because these categories are, in the case of *PhenoPointer*, multilabel classification problems, whereas in *Traitar* specific pathways and reactions are modelled as particular binary classifiers. In the following only those traits are taken into consideration from *Traitar*, that have a corresponding label in one of the two phenotype categories in *PhenoPointer*. Table 8.1 shows the comparison results of the comparable phenotype categories, where green fields mark correct predictions, red marks misclassifications, blue marks undeterminates (*Gram Staining*), and grey unknown true annotations. Figure 8.2 depicts the non-mappable phenotypes of *Traitar* and the evaluation of the predictions made. In this figure those phenotypes are left out where no classification resulted in a prediction in at least one organism e. g., *Cell Arrangement*, although it is capable of doing so.

In the case of directly comparable phenotypes, *PhenoPointer* shows a better performance overall than *Traitar*. Predictions on *Cell Shape* and *Oxygen Requirement* show a perfect match to the real values (100% correctly predicted), whereas *Traitar* made four mismatches on *Oxygen Requirement* and six mismatches on *Cell Shape*. For prediction in the category *Motility* both tools performed equally, misclassifying the same organisms. Classifications made in the categories *Sporulation* and *Gram Staining* by *PhenoPointer* (two mismatches, or three respectively) are better than those from *Traitar* (three mismatches, or four respectively). On *Cell Arrangemnt*, *Traitar* made no prediction at all and *PhenoPointer* again had a perfect match over all annotated organisms of 100% correctly predicted labels.

	Sporanaerobacter sp. PP17-6a	*Methanobacterium congolense Buetzberg*	*Methanobacterium formicicum MF^T*	*Proteiniborus sp. DW1^1*	*Ruminiclostridium thermocellum BC1*	*Bacillus thermoamylovorans 1A1*	*Clostridium cellulosi DG5*	*Petrimonas mucosa ING2-E5A^T*	*Methanoculleus bourgensis MS2^T*	*Clostridium bornimense M2/40^T*	*Proteiniphilum saccharofermentans M3/6^T*	*Defluviitoga tunisiensis L3*	*Herbinix hemicellulosilytica T3/55^T*	*Clostridium sp. N3C*	*Methanobacterium sp. Mb1*	*Methanoculleus chikugoensis L21-II-0*	*Fermentimonas caenicola ING2-E5B^T*	*Propionispora sp. 2/2-37*	*Herbinix luporum SD1D^T*	*Methanobacterium formicicum Mb9*	*Peptoniphilus sp. ING2-D1G*	*Methanothermobacter wolfeii SIV6*
Metabolism	**P** T	P ·	P ·	· ·	P **T**	P ·	P **T**	P T	P ·	**P** ·	P T	**P** ·	P **T**	P **T**	P ·	P ·	P T	· ·	P **T**	P ·	**P** T	P ·
Salinity	· ·	· ·	· ·	· ·	· ·	· ·	**P** ·	· ·	· ·	· ·	· ·	P **T**	· ·	· ·	· ·	· ·	· ·	· ·	· ·	· ·	· ·	· ·
Phenotype	· ·	P ·	P ·	· ·	P ·	P T	P ·	**P** ·	P ·	**P** ·	**P** T	P ·	· ·	P ·	**P** ·	P ·	P T	P T	P ·	**P** ·	P T	P ·
Cell Shape	P **T**	P **T**	P **T**	P T	P T	P T	P T	P T	P **T**	P T	P T	P T	P T	P T	P T	P **T**	P T	P **T**	P T	P T	P **T**	P T
Cell Arrangement	· ·	· ·	P ·	· ·	P ·	· ·	P ·	P ·	P ·	P ·	P ·	P ·	P ·	· ·	P ·	P ·	P ·	· ·	P ·	P ·	P ·	P ·
Oxygen Requirement	P T	P T	P **T**	P T	P T	P **T**	P T	P T	P T	P T	P T	P T	P T	P T	P ·	P T	P T	P T	P T	P **T**	P T	P T
Energy Source	P ·	· ·	· ·	· ·	· ·	P ·	P ·	P ·	· ·	P ·	P ·	P ·	P ·	· ·	· ·	· ·	P ·	P ·	P ·	· ·	P ·	· ·
Temperature Range	P ·	P ·	P ·	P ·	P ·	**P** ·	P ·	P ·	P ·	P ·	P ·	P ·	**P** ·	**P** ·	P T	P ·	P ·	P ·	**P** ·	P ·	P ·	P ·
Biotic Relationships	· ·	P ·	P ·	· ·	P ·	· ·	P ·	P ·	P ·	P ·	P ·	P ·	P ·	· ·	P ·	P ·	P ·	· ·	P ·	P ·	P ·	P ·
Motility	P T	P T	P T	P T	P T	P T	P T	P T	P T	· ·	P T	**P** **T**	P T	P T	· ·	· ·	P T	P T	**P** **T**	P T	P T	P T
Sporulating	P T	P T	P T	P T	P T	P T	P T	P T	P T	P **T**	P T	P T	**P** **T**	P T	· ·	P T	P T	P T	**P** **T**	P T	P T	P T
Gram Staining	P T	**P** T	**P** T	– –	P T	P T	**P** **T**	P T	P **T**	– –	P T	P T	P T	– –	· ·	P **T**	P T	P **T**	P T	· ·	P T	· ·

P: True prediction (Phenopointer); T: True prediction (Traitar); **P** / **T** : False prediction (Phenopointer/Traitar); · : Ambiguous/ unknown; – : Gram indeterminate

Figure 8.1: **Prediction comparison matrix of mappable phenotype categories: *Pheno Pointer* vs. *Traitar***

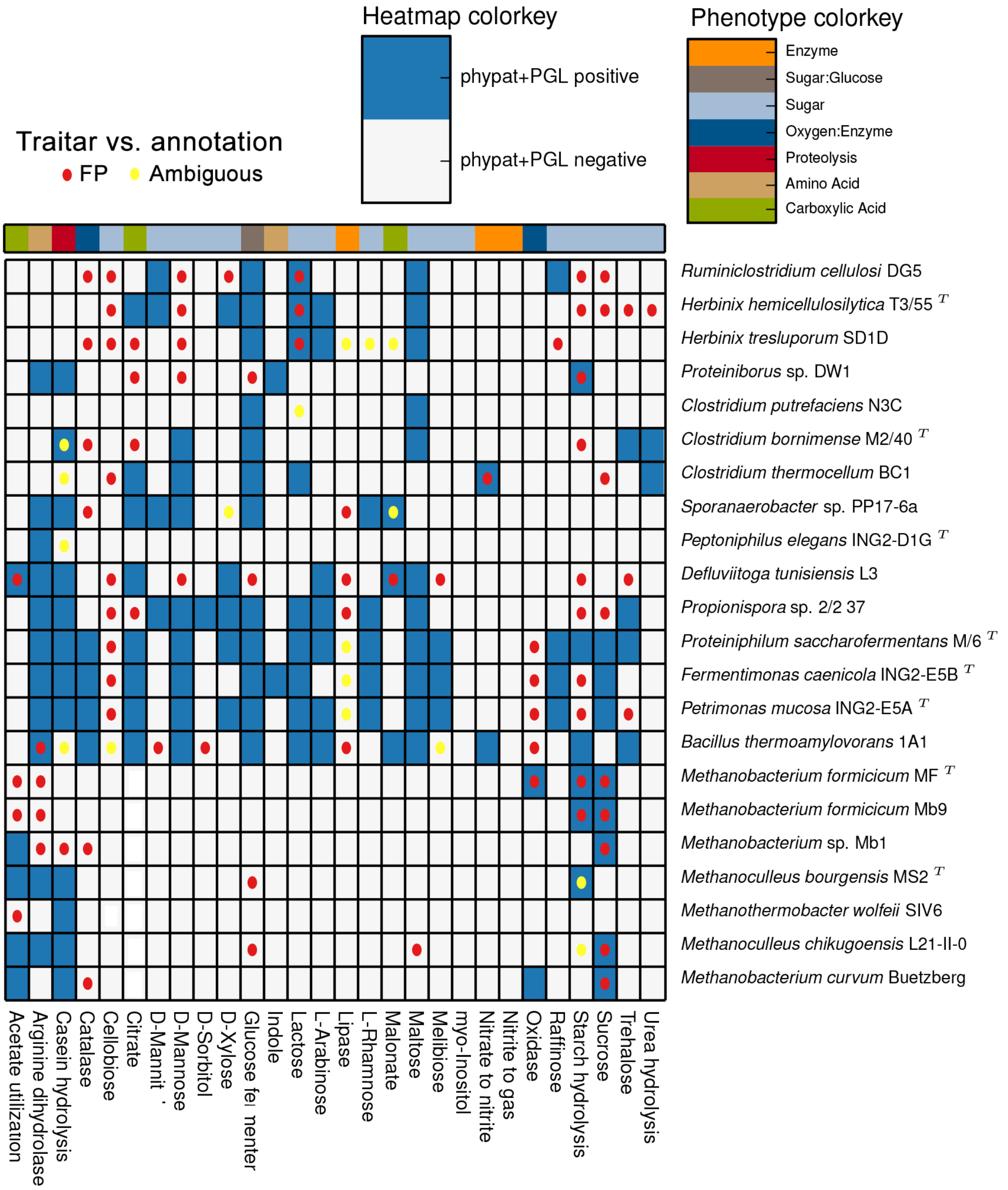


Figure 8.2: **Predictions by *Traitar* of unmappable phenotypes** Manual control of predicted phenotypes by *Traitar*, that could not be mapped on phenotype categories of *MetaStone*, blue marks a prediction made, white marks no made prediction, red circles mark FP predictions as evaluated by a sovereign domain expert, yellow circles mark ambiguous references in literature/ wet lab experiments.

In the four phenotype categories, where only *PhenoPointer* is capable of making predictions, *PhenoPointer* again showed very good prediction performance, with only one misclassification in the category *Salinity* (13 correctly predicted) and four misclassifications in the category *Temperature Range* (18 correctly predicted), The categories *Energy Source* and *Biotic Relationships* were predicted with a perfect match over all annotated organisms.

The last two categories are the ones with fragmented comparability, because *Metabolism* and *Phenotype* group together possible singleton phenotypic classifiers of *Traitar*. In the case of *Metabolism*, *Traitar* did not find six metabolic reactions that have been predicted by *Pheno Pointer*, but did not predict two class labels that had been accidently predicted by *PhenoPointer*. For ten predictions made by *PhenoPointer Traitar* had no correspondent classifier, such as 'Methanogen', 'Sulfur oxidiazr' and 'Iron reducer'. From 20 predictions made by *PhenoPointer* in this category, four are misclassifications while 16 were correctly classified, e. g., 'Cellulose degrader' or 'Lactose fermenting'. The only common labels within the category *Phenotype*, are those related to catalase activity, where *Traitar* predicted correctly four out of four, whereas *PhenoPointer* misclassified one sample. The other labels within this category that were correctly predicted by *PhenoPointer* are more general ones like 'non-pathogen'.

8.2 Conclusion

A direct comparison of *PhenoPointer* and *Traitar* is hard to draw, because *Traitar* is specialized for predicting primarily consumed substrates for energy production and not metabolic pathways or observable phenotypes. This may explain why additional general observable phenotypes that are present in *PhenoPointer* are missing. Without looking at any prediction performance measures, *PhenoPointer* is theoretically more suitable for characterizing cultivation conditions in order to reproduce preferred environmental conditions for optimal growth. By looking at the amount of correctly predicted observable phenotypes, *PhenoPointer* performs much better in a real world setting than the validation scores might suggest. Looking at the few comparable phenotypes, *PhenoPointer* clearly outperforms *Traitar* in all phenotype categories describing observable phenotypes. Interestingly, the same holds for predictions of metabolic features, where *PhenoPointer* again performs much better on this real world data set than the validation or generalized validation measures would suggest, with a relatively low false positive rate, and also better than *Traitar*. In comparison to *Traitar* the predictive strength is comparable for mappable prediction class labels, or in some cases even better, as seen in *Herbinix luporum SD1D*^{T} and *Ruminiclostridium thermocellum BC1*, both correctly classified as 'cellulose degrader', or *Clostridium cellulosi DG5* correctly classified as 'cellulose degrader', 'biomass degrader', and 'nitrogen producer'.

The only possible remaining benefits of *Traitar* are thus the theoretical predictive strength of consumed substrates for energy production, as proposed in the publication. The knowledge about primary energy sources, such as specific sugar and glucose molecules and acids, is of special interest for molecular biologists. But the reality of *Traitar* on this real world sample set looks quite devastating, as the amount of misclassified nutrients shows (see Figure 8.2 for details) with a misclassification ratio of nearly 50%. *PhenoPointer* has no specific phenotype classifier specialized on nutrients, but if the amount of organisms within *IMG/M* are accordingly annotated, one may think of training specialized ML classifiers for this purpose. Nevertheless, some nutrients have been successfully predicted, such as 'Xylan degrader' in *Proteiniphilum saccharofermentans M3/*6^T or 'Xylose fermenting' in *Bacillus thermoamylovorans 1A1*, but also some misclassifications appeared e. g., two organisms have been misclassified as reducing sulfur.

Because the GIDEON database could not be screened, no proper conclusion can be made about the underlying data basis of *Traitar*. One major issue seems to be that the amount of training samples is quite low and no coverage is shown in the *Traitar* publication about the annotated organisms related to a specific phenotype. Another issue might be that training has been only performed on bacterial organisms and not on archaeal organisms as well. Some pre-studies related to this thesis showed that training on a mixed set of organisms of the two domains, bacteria and archaea, results in slightly better prediction performance.

Part III

MVIZ – Metagenome VIZualition

Analysis of marker gene-based studies is often limited to determining the taxonomic structure of microbial communities, whereby diversity studies or observation of community shifts due to stress response reaction can be performed. What remains unclear, however, is what possible molecular reactions are concealed in the microbial community i. e. observable phenotypes and metabolic features are missing in time series and community shift analyses.

MVIZ tries to elucidate metabolic capabilities and phenotypical characteristics with the help of organism-related metadata through enrichment of a community profile with this type of information. A move from taxonomy level-based community analyses to phenotypic analyses takes place, allowing observation of shifts over time not only at taxonomic level but also at trait level. This also holds for large-scale studies, where hundreds of subjects are screened and a core functional trait and observable phenotype catalogue can be derived.

This part of the thesis first explains the implementation details of *MVIZ* and offers a guide of how to perform a study within *MVIZ*. Afterwards, a study is explained and exemplarily analyzed with *MVIZ*, to explore the metabolic features of biogas producing digesters and to demonstrate the capabilities of metadata-enrichment studies. This part closes with a discussion of *MVIZ*.

9 *MVIZ* – Principles and Implementation

The main feature of *MVIZ* is the visualization of community profiles, such as WMGS-based community profiles or 16s rRNA-based taxonomy profiles. The visualization relies on metadata-enriched data sets, although a visual comparison of several communities based on their characteristics and expressed phenotypes is possible. *MetaStone* will serve, as in the case of *Pheno Pointer*, as the data back end system for input data processing and data retrieval for the enrichment process with metadata. The added data to the input are the categorical metadata as shown preciously in Table 3.2 on page 31. After accomplishing data preparation through functionalities in *MetaStone*, the enriched data set can be put into the web application *MVIZ*. *MVIZ* is a client-side web application, which is implemented with AngularJS (Google Inc. 2010) for explorative data analysis in the context of visual comparison of metadata-enriched community profiles.

9.1 Metadata Enrichment of metagenomic community profiles

One output of a 16s rRNA-based sequencing project is a community profile, describing the prokaryotic organisms living in a specific environment, where the initial sample has been picked. A common output file format for such community profiles is the BIOM format (McDonald et al. 2012), which is a comprehensive format specifically for storing organisms and their abundances as a community profile. A BIOM file is capable of storing multiple samples and thus a collection of various samples rather than a single sample. The taxonomic classification of a genome entry in a sample is given as a human readable taxon name. Parsing a BIOM file with embedded metadata is not simple, since its identifiers for additional data fields can come in various fashions, and a generic output file format of a metadata-enriched community profile must be defined. Therefore, a JSON (Bray 2014) file with a specific structure is used as the output, loaded into *MVIZ* for visualization purposes.

Metadata levels of confidence The metadata used for enrichment that is applied to the input data set during the pipeline run can be considered as both trusted and untrusted metadata; the confidence level is explained by the source of metadata generation. The metadata imported from *IMG/M*, relying on the annotations made within the *GOLD* system representing quality controlled metadata, have a high level of trust. Within *PhenoPointer* predictions of hitherto partially annotated organisms regarding their phenotypes have been made and stored in *MetaStone* as additional metadata, this set of ML classified phenotypic metadata are considered as untrusted. Untrusted does not mean that predicted phenotypes must be considered as false, but such annotations must always be treated with caution. The predictions regarding an organism's phenotype are persistently stored in the ORM model `MetaStone.models.InferredSpeciesRelatedCharacteristics`.

Another approach to filling gaps in the metadata annotations of organisms stored in *MetaStone* is to infer missing metadata, directly based on the taxonomy level of related organisms. This approach is performed for non-phenotypic metadata categories, except those belonging to the group of sequencing related metadata. For genomes, where a particular metadata category is not annotated, the taxonomic tree is traversed in a bottom-up fashion until organisms can be found at a taxonomic level that are annotated with labels belonging to the missing metadata category. The labels are gathered in a dictionary and persistently stored in *MetaStone* in one of the responsible ORM models `MetaStone.models.DirectInferredEcosystemRelatedCharacteristics` and `MetaStone.models.DirectInferredSamplingSiteRelatedCharacteristics`.

Metadata enrichment pipeline For metadata enrichment of taxonomically assigned organisms in a sequenced environmental sample collection, the BIOM file must be initially parsed to map each genomic taxon name onto the genomes stored in *MetaStone* for further processing. For mappable taxon names the metadata is gathered from the database, and finally the output file in a JSON file format with metadata annotated genomes embedded in the community profiles is delivered to the end user as the result of this pipeline. The implementing python modules are shown in Figure 9.1.

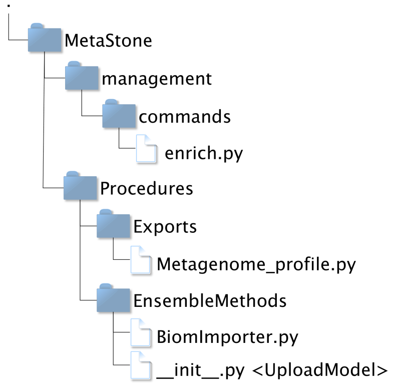


Figure 9.1: ***MVIZ* code extension to *MetaStone*** The CLI command for automated metadata-enrichment of metagenomic community profiles is implemented in *enrich.py* under *./MetaStone/-management/commands/*. The workflow parses first the BIOM input file in the module *BiomImporter.py*, located under *./MetaStone/Procedures/Imports/*, following by enriching the profiles with metadata and final export as a JSON-based output file in `Metagenome_profile`, located under *./MetaStone/Procedures/Exports/*.

After calling the `enrich` command via the CLI and passing the path to the input BIOM file as a parameter, the user can optionally have the samples and their community profile permanently loaded into *MetaStone* for later collation of several community sets as one data set. During parsing of the input BIOM file, each single genome taxon name from input is tried first to map as a whole against all genomes stored in *MetaStone*. This reference information is stored in the field `taxon_name` of the ORM object `MetaStone.models.Genome`. If no direct database hit can be found then the taxon name is split into its *genus* and *species* parts and is used in a filter query on the corresponding fields in the ORM model `MetaStone.models.GenomeLineage`. For further details about the DB schematics see Figure 4.2 on page 36 and for a description see section 4.2 'The data model for persistence' from page 34 onwards. If a taxon name cannot be mapped by either of the two mapping mechanisms, the organism is considered as non-mappable to known genomes and its abundance accumulated in a mock genome per individual sample. The internal data structure for processing samples and their inhabited organisms reflects the same structure as in a BIOM file with additional metadata about the metagenome for internal use, shown in Figure 9.2.

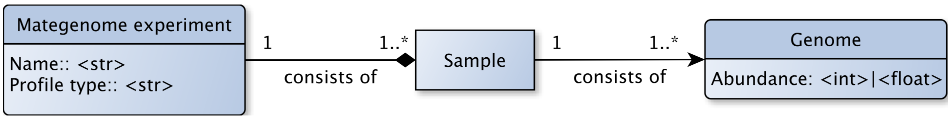


Figure 9.2: **Relationship of a metagenomic experiment and its community profiles** An experimental setup of for comparing metagenomes is modelled internally in *MetaStone*, that an experiment consists of one or more samples, where each sample set consists of a set of genomes, which carry their abundances inside a particular samples as internal fields.

Following the mapping of genomes onto *MetaStone*, the community profiles can be enriched with additional metadata covering trusted and untrusted annotations as described in section 9.1. These categories are grouped in four groups also used for the DB models, which were *Ecosystem*, *Sampling Site*, *Sequencing*, and *Species related*, where the latter one represents phenotypes of an organism. For enriching the input data set with this metadata categories, the metadata on trusted and untrusted level related to the mapped genomes is loaded from the database and added to the output data structure, a JSON file. The values for non-mappable genomes, thus the accumulated mock genome, are set to `Unmapped` for trusted and untrusted metadata.

JSON output format specification The contents of a *MVIZ* compatible JSON file are divided into two main parts. The first part contains information about all organisms that are community members in at least one sample of the overall sample set. The second part contains information about a metagenome community profiling project as extracted from the input BIOM file in combination with the described samples and their organisms with abundances. Listing 9.1 shows an example of a metadata-enriched community profile.

In the genome metadata annotation part, the `taxon_oid` of a genome entry in *MetaStone* is in the JSON file as its primary identifier. Metadata is stated per genome entry and for each metadata category two values are given: the first one is the trusted *IMG/M* annotation and the second value is the inferred one. The mock genome is identified by the ID `0` and all metadata annotation are designated as `Unmapped`, i. e. `"Motility":["Unmapped", "Unmapped"]`, thus the fractions of these organisms can easily be determined in the metadata charts in *MVIZ*.

The part containing the information about a metagenomic experiment and the composition of its related samples concerning the inhabited organisms, lists the organisms and their abundances for each sample. Each organism is identified by its `taxon_oid`, thus metadata annotations of genomes can be held separately and need be stored only once.

```
{
  "genomes": {
    "2504756063": {
      "Ecosystem related characteristics": {
        "Ecosystem": ["Host-associated", "Unknown"],
        "Ecosystem Category": ["Insecta", "Insecta"],
        "Ecosystem Subtype": ["", "Mine"],
        ...
    }
    "0": {
      "Ecosystem related characteristics": {
        "Ecosystem": ["Unmapped", "Unmapped"],
        "Ecosystem Category": ["Unmapped", "Unmapped"],
        ...

...
  "metagenomes": {
    "MockData": {
      "name": "MockData",
      "profile_type": "16s_rRNA",
      "samples": {
        "Yvy1.1.GR1m": {
          "0": 1024,
          "2504756063": 1,
          "2506520040": 1,
          "2509276047": 13,
          ...
        },
        "Yvy2.2.TL2n": {
          "0": 42,
          "2504756063": 1,
          "2511231163": 54,
          "2512047001": 1,

...
}
```

Listing 9.1: Snippet of a JSON file describing a metadata-enriched metagenomic community profile. The genome with the id `0` is the container for the mock genome, individually accumulated per sample.

9.2 WebUI for interactive Visualization

For the purpose of comparative visualization of metadata-enriched metagenome sample sets, *MVIZ* has been developed, an interactive client-side web application written purely in JavaScript utilizing the AngularJS (Green et al. 2013) open-source web application framework maintained by Google Inc. (Google Inc. 2010). The reason for choosing AngularJS was that the underlying conceptual design focuses primarily in the development of single-page applications and that the framework has its strength in data manipulation through a bidirectional binding between the view (WebUI) component and the data containing model. This means that changes in the WebUI related to the selection of visualized data directly affect the model in the back end and vice versa; changes in the data model instantly affect the WebUI. Because of these two aspects it is possible to implement a data manipulating and visualization application that runs in any current web browser. The program logic runs directly on the user side and the required program code and libraries can be stored locally at the user side or be deposited at a remote location accessible in the network or the internet. It must be said that manipulating data in *MVIZ* is related to selection of data to be displayed in the WebUI and not for metadata-enrichment of metagenomic data sets, which is implemented independently in *MetaStone* as described in the previous section 9.1 "Metadata Enrichment of metagenomic community profiles".

For loading data into *MVIZ*, the input JSON should be loaded directly into the web application without any further pre-processing other than the previous enrichment process with *MetaStone*. It is also desirable that compressed JSON files can be used with *MVIZ*, and so ZIP (Katz 2007) compressed input data should also be processable.

The user should have the ability to select samples and group them as a subset of all samples contained in the input. In addition, it is a necessity to have separate views for each subset of grouped samples in order to compare them. It should also be possible to select which metadata category to display, and also which metadata labels to show. The visualized metadata is split into two different switchable views: an accumulation of all samples grouped together, and of all samples visualized individually. The visualization technique should be as a pie chart or as bar chart. For a bar chart, the columns must be aligned over all shown groups of selected sample subsets. Metadata labels of low abundance will be masked and grouped together automatically in the UI for reasons of clarity and comprehensibility of the displayed charts.

The main programmatic concept of data flow and control is achieved in AngularJS applications by singleton `service`-objects, that offer globally functional and data manipulation services in the scope of an AngularJS application across all components. Communication between different `service` objects is accomplished via an internal notification broadcasting service that interconnects dependent application components, such as the data model or the UI. Before some explanatory data flow charts are shown, the most important `service` objects in *MVIZ* are briefly explained.

msg.service Extension to the internal messaging service and view components (directives), so that these can communicate following the principles of observer design pattern.

data.service Loads input data into the model, where input is a JSON file that can be compressed with ZIP

metadata.service Processes the input data and converts it into the final data structure. For computational reasons the taxon oids of an organism in a sample are substituted by the genome related metadata for direct access.

collection.service Logic for grouping samples into subsets. This service is also responsible for storing the ordering of views in the GUI of sample sets.

characteristic.service Main service for managing available metadata groups and subsequent categories.

label.service Controls and stores selected metadata classes for visualization and current selected samples.

threshold.service Intermediate controller for storing masked metadata labels and by the user dismissed metadata labels.

pref.service Manages changes in thresholds and metadata labels, controls interactively masking of low abundant labels.

graph.service Visualization logics of pie charts and bar charts. This service is dependent on other services like `label.service` or `threshold.service`.

In Figure 9.3 on page 117 data flow diagrams are shown from loading and parsing a JSON input file up to finally plotting the charts. After loading and potential decompression of the input file via `data.service`, the data is further processed in `metadata.service`. Data describing the metagenomic data set and its samples are loaded in the UI for further grouping of samples by the user. After grouping samples by the user, the `collection.service` send a `collectionservice.sample.added` message to all listening components and service objects. The use has now the ability to select which metadata group and subsequently which metadata category to visualize, processed by `characteristic.service`. After this step the chart is ready to be displayed to the user, but beforehand the `label.service` collects available metadata labels and the `threshold.service` determines which labels are should be masked and which are displayed. Finally, the `graph.service` collates data from the services and presents them to user via the WebUI chart component.

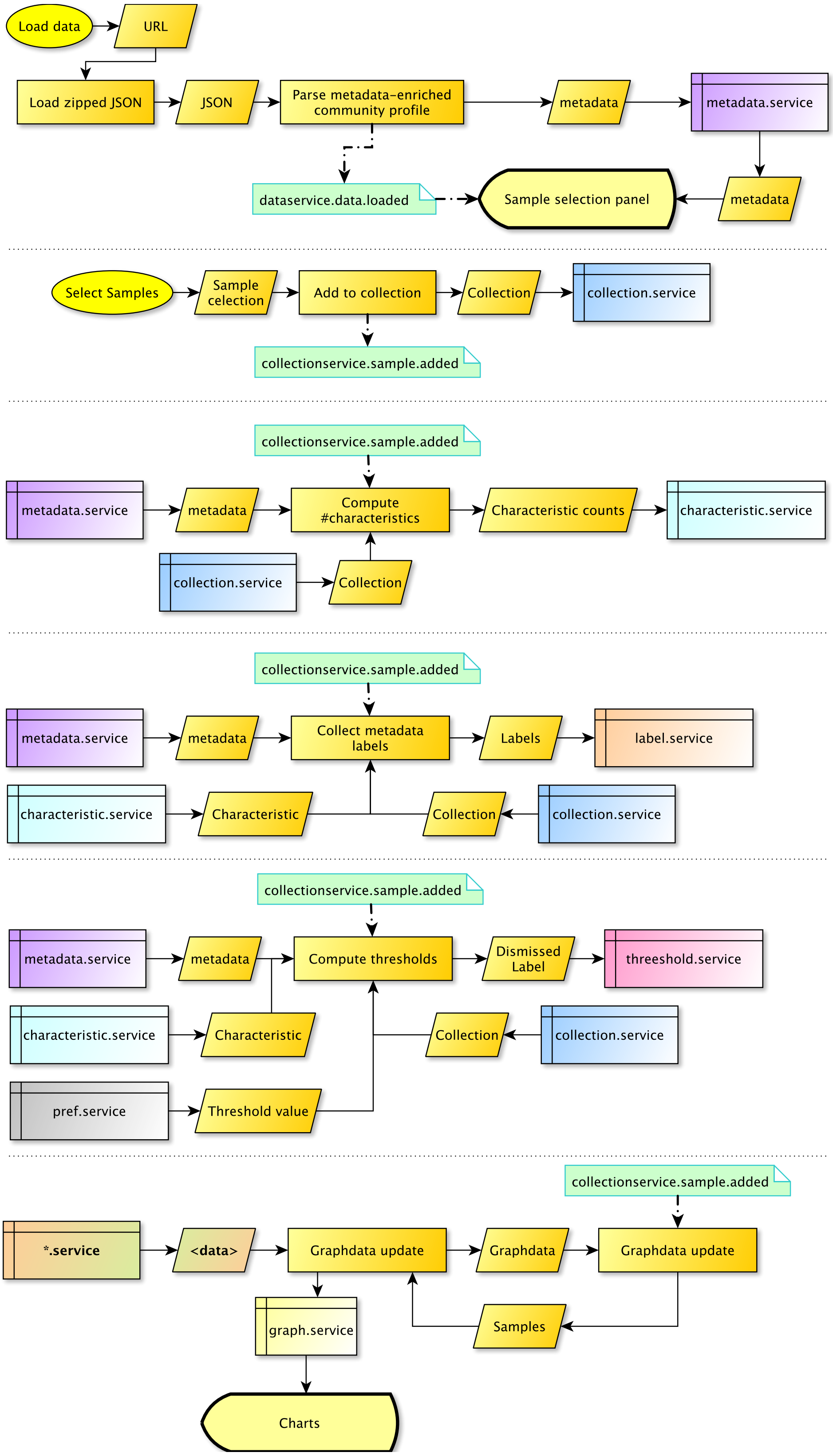
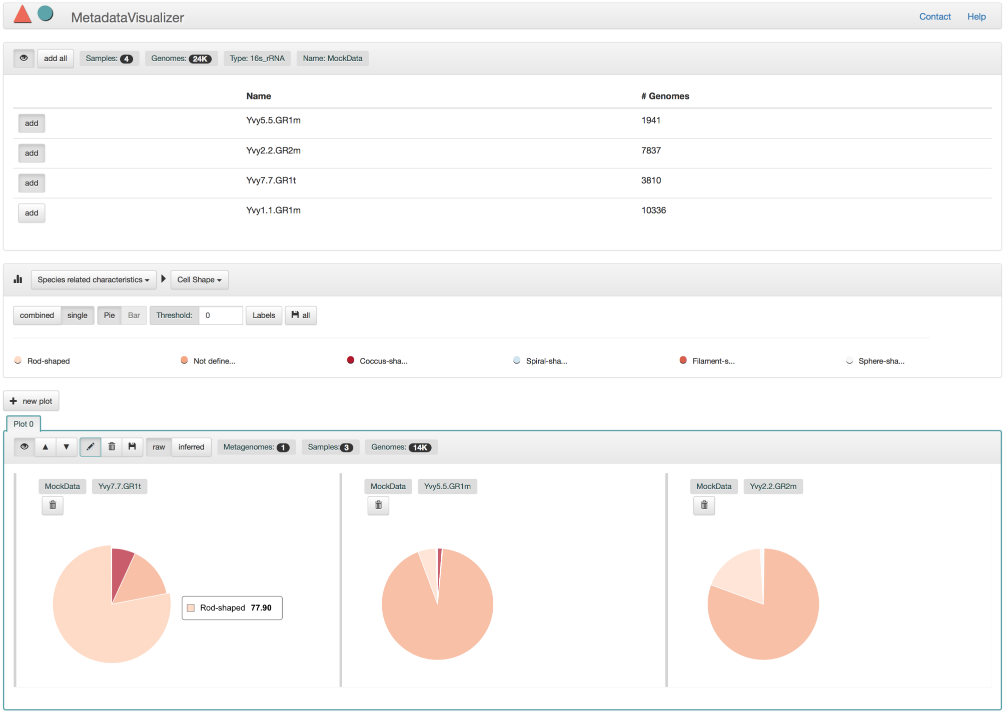


Figure 9.3: **Data flow in *MVIZ*** Starting at the top of the flow chart, data is loaded to *MVIZ* and after sample selection a `collectionservice.sample.added` is broadcasted. Subsequent collation of data is initiated by individual components resulting in displaying the desired charts.

9.3 How to: *MVIZ*

MVIZ comes packed within *MetaStone* and is also deposited at GitHub as a solitary download. It can be checked out from `https://github.com/mrumming/MetagenomeVIZualizer/` and is released under the BSD 3-clause license. Please note that the application has been tested in Firefox[21] and Safari[22] browsers for full functionality. The Chrome browser with default security settings is not supported, since loading of locally stored files is prohibited.

For loading a metadata-enriched metagenomic community profile into *MVIZ*, the *index.html* must be opened in a browser and the URL suffixed with the parameter `data`. The value of `data` is the path to the JSON document, previously enriched with metadata by *MetaStone*. If *MVIZ* is deposited at a remote location and the JSON file is located under a local home directory, the URL would look like `https://yoururl.com/mviz/index.html?subdirInYourHome/meta1.json`. Figure 9.4 shows a screen shot of a loaded metagenome data set that has been previously enriched with metadata.



	Name	# Genomes
add	Yvy5.5.GR1m	1941
add	Yvy2.2.GR2m	7837
add	Yvy7.7.GR1t	3810
add	Yvy1.1.GR1m	10336

Figure 9.4: **Comparative pie charts in *MVIZ*** An example metadata-enriched marker gene-based on 16s rRNA sequencing data, named as *MockData*, consisting of three samples loaded in *MVIZ*. All three samples are grouped together and the selected visualized metadata labels belong to the metadata group *Species related characteristics* and describe their *Cell Shape* (the selected metadata category). The generated interactive chart shows the abundances of the metadata annotation for each sample separately as pie charts.

MVIZ is divided into three components: the samples selection tab in the top for grouping samples into groups for direct comparison, the preferences tab in the middle for selection of displayed

[21] *Firefox* – `https://www.mozilla.org/firefox/`
[22] *Safari* – `https://www.apple.com/safari/`

metadata categories and switching between pie chart and bar charts, and the chart panel in the bottom displaying the comparison visualization plots.

Sample selection & Preferences tab To plot charts characterizing samples based on the metadata of their contained genomes, samples have to be added via the *add* button in the corresponding metagenome sample summary list. To show all samples belonging to a metagenome experiment, the eye icon is selected. After grouping of samples, the metadata group and subdivided metadata category to be visualized is selected in the preference tab. If multiple samples have been grouped together, an accumulative plot is shown, so all abundances metadata of genome are shown in an accumulated chart. In the preference tab it is possible to switch the visualization to single charts per sample in that sample group. It is also possible to switch from here between compact pie charts and a more comparable bar chart. A bar chart is common for comparing several sample groups, because the columns are aligned in each plot for ease of perception. One property of grouped samples is that each group has its own visualization plot tab in the chart panel. The legend of currently available metadata labels is shown below the main preferences tab. These labels are switchable for blending in and blanking out correlated metadata labels.

Adding new sample groups Between the legend and chart panel, new visualization plot tabs can be added via the 'new plot' button. The scope of selected samples switches to the new panel, which is currently empty, thus a new samples group can now be added to *MVIZ*. Thereafter the additional chart is filled with data and presented to the user. To add or remove samples belonging to a sample group, the pencil icon in the correlated visualization plot is selected, and changes in the sample selection tab promptly influence the displayed chart.

Chart panel for displaying visualizations of comparison plots As mentioned previously, for each sample group an individual chart tab is presented in the WebUI. The ordering of tabs can be changed via the arrow icons, blanked out via the eye icon, and completely removed by clicking the trash can symbol. To switch between trusted and untrusted metadata annotations, the buttons *raw* and *inferred* respectively can be selected. Labels including abundance percentages explaining the metadata labels in the charts are switched off as default, these can be switched on in the preferences tab with the *labels* button. Nevertheless, an explanatory overlay is shown if the mouse cursor is moved over a certain segment of a chart. By clicking on the tab name in the upper-left corner of a chart tab, the name of this chart can be changed.

Exporting charts By clicking the *disk* symbol in the preferences tab, all displayed charts are exported as a plain single HTML page including the legend of metadata labels. If only a particular chart should be exported, the disk symbol in the related chart tab exports the plot solely the correlated metadata labels legend.

10 *MVIZ* – Metadata-enrichment of a community profile

In this example of a metadata-enrichment study performed on a 16s rRNA-based community profile, the opportunities for metadata-assisted profiling will be shown and further applications based upon this study introduced. The real world community profiles are taken from the study 'Identification and genome reconstruction of abundant distinct taxa in microbiomes from one thermophilic and three mesophilic production-scale biogas plants.' (Stolze et al. 2016) published in 2016, in which four biogas producing anaerobic digesters (BGP1–BGP4) were functionally analyzed. The study utilized 16s rRNA marker gene sequencing (two biological replicates) and WMGS sequencing for all four biogas plants individually. The plots visualize the metadata-enrichment diagrams of the two biological samples as a single combined plot.

Three biogas plants (BGP1–3) were operated under mesophilic conditions and the fourth (BGP4) under thermophilic conditions. Substrates to be fermented by the bacterial and archaeal community consisted of maize silage, sugar beet, manure from poultry, pigs, or cattle, and grass, whereby each fermenter was fed with an individual composition of these and also with in different proportions e. g., BGP3 was fed with maize silage (67%) and pig manure (33%), whereas BGP4 was fed with maize silage (60%), pig manure (10%) and additionally with grass 30%. The biogas plants differed in their yields of produced biogas, including different proportions of methane gas. This information was collected as process-related metadata for each biogas plant individually, and is shown in Table 10.1.

Figure 10.1 shows the metadata-enriched community profile, where the optimal temperature range of the inhabited organisms is drawn as a bar chart. The profile shows that the process annotations related to the process temperature are consistent with the shown profile: $BGP1 - 3$ contains no organisms annotated as thermophilic, whereas in BGP4 the community consists of $\sim 8\%$ thermophilic organisms.

While the study was being conducted BGP4 was annotated as being fed only with plant based substrates, so no manure was added to the reactor system. The metabolism profile as shown in Figure 10.2, does show a relatively high abundance of sugar and xylose fermenters, but also a high abundance of organisms capable of reducing sulfur and triosulfate, as well as nitrate reducers. The BGP4 process metadata thus made a suspicious impression, and so they were validated and an error in the fed substrates was detected. It emerged that BGP4 had also been fed with manure, explaining why the microbiome is capable of reducing sulfur/ thiosulfate.

In comparing BGP4 to the other biogas plants that had been fed with much higher amounts of manure, the abundance of organisms capable of cellulose degradation is much lower, but this could be explained by the much higher abundant organisms fermenting xylose and the slightly

Parameters	**Optimal range**	**BGP1**	**BGP2**	**BGP3**	**BGP4**
pH	6.8-8.0	7.7	7.8	7.53	7.8
VOA (mg/l)	2050-6500	4876	5093	3391	3300
TIC (mgCaCO$_3$/l)	8500-15 000	11 040	15 928	14 714	11 600
VOA/TIC	0.11-0.6	0.45	0.32	0.23	0.28
NH4-N (g/kg)	1.2-4.0	1.9	2.32	3.15	n. d.
HAC-eg (g HAceg/l)	1.3-1.9	2.03	0.40	n. d.	0.57
Temperature (°C)	n. d.	40 (mesophilic)	40 (mesophilic)	40 (mesophilic)	50 (thermophilic)
Fed stubstrates (%)	n. d.	Maize silage (45), sugar beet (22), poultry manure (33)	Maize silage (50), grass (10), poultry/ pig/ cattle manure (40)	Maize silage (67), pig manure (33)	Maize silage (60), grass (30), pig manure (10)
Retention time (days)	n. d.	92	74	81	28
Biogas yield (l/kg oDM)	n. d.	609.87	664.5	528.5	658.11
% Methane	n. d.	49.60	52.24	52.4	56

VOA Volatile organic acids; TIC Total inorganic carbon; oDM Organic dry matter; n. d. No data

Table 10.1: **Process-related metadata of four biogas plants** Physico-chemical characteristics and fed substrates of the four different biogas plants analyzed and optimal ranges some of process parameters.

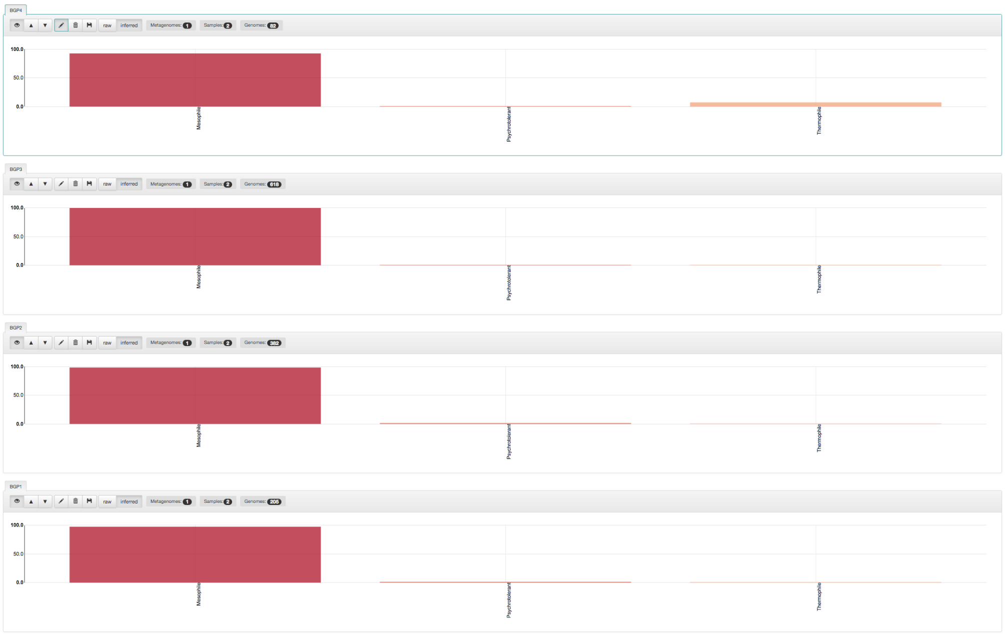


Figure 10.1: ***Temperature Range*** **profiles of four biogas plants** BGP4 shows the highest amount of thermophilic organisms as expected by the process-related metadata annotation. BGP1 and BGP2 show a slightly increased number of psychrophilic organisms.

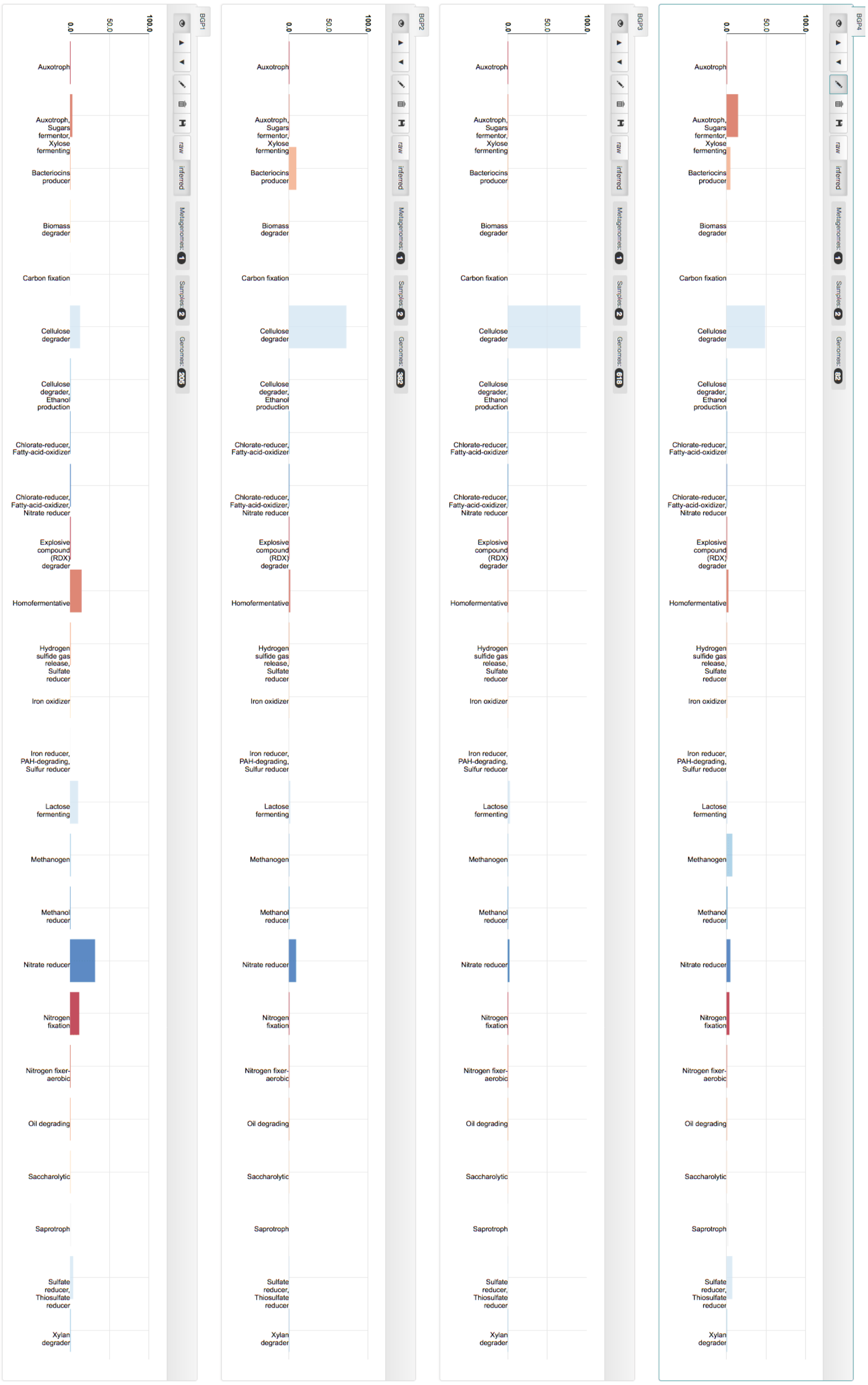


Figure 10.2: ***Metabolism* profiles of four biogas plants** Metabolic capabilities of the four analyzed biogas producing anaerobic digesters.

increased fraction of organisms capable of xylan degradation. BGP1 had the lowest amount of cellulose degrading organisms, but the amount of homofermentatives and lactose fermenters is markedly increased, along with nitrate reducers. The core functional phenotypes, functions shared among over all samples and thus characterizing this microbial habitat, are nitrate reduction and cellulose.

Comparing displayed trusted and untrusted metadata i. e., annotations from *GOLD* in the first case and with predicted annotations replacing missing values in the second case, shows that the information gain is tremendous, as shown in Figure 10.3. The dependence on oxygen of the inhabited organisms is shown on the left of the plot, whereas for the energy source only the untrusted metadata pie charts are shown. BGP2 showed, at trusted level, only a low amount of aerobes, increased based on the predictions of unlabeled organisms. In BGP3 almost all unannotated organisms are classified as anaerobes and in BGP1 and BGP4 the gains are almost balanced over all displayed requirement levels. Interestingly, BGP4 has the largest amount of organisms classified as facultatives.

On the right hand side, the primary source of obtaining carbon and energy is shown at untrusted metadata level, because the matchable organisms in *MetaStone* were not annotated with metadata related to the energy source. BGP1/3/4 profiles look fine, but BGP2 shows a high ratio of photosynthetic organisms, a very unusual observation for anaerobic digesters where the content matter is kept in total darkness. After identification of the correlated organisms it was revealed that the prediction was erroneous for this specific highly abundant and mappable organism. The real nature of it would be chemoorganotrophic.

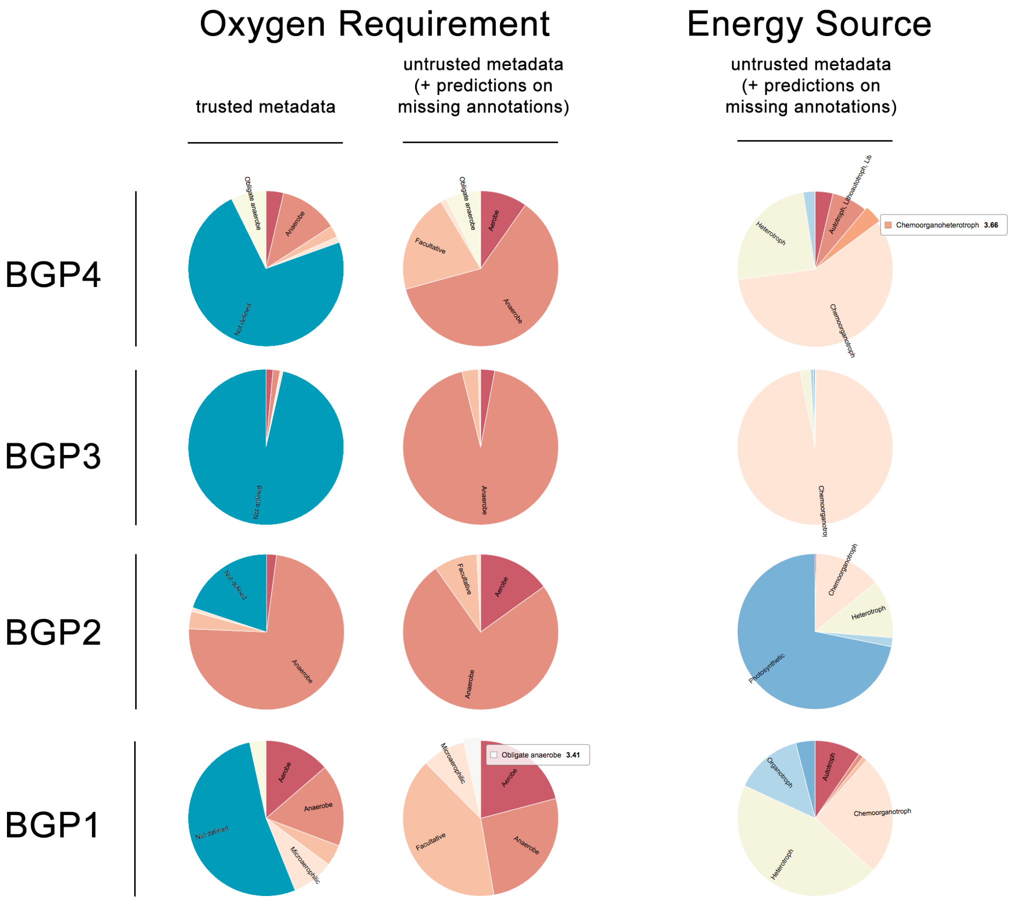


Figure 10.3: ***Oxygen Requirement* and *Energy Source* profiles of four biogas plants** On the left metadata annotations are shown for *Oxygen Requirement* for trusted and untrusted metadata, on the right untrusted metadata annotations are shown for *Energy Source*.

Part IV

From genotype over phenotype to function-driven metagenomics

11 Review of *PhenoPointer* and *MVIZ*

Genome binning is an undirected approach of generating artificial genomes guided by the methodology of the binning algorithm applied, generating partially complete and possibly contaminated genomes. There is a need for guidance to identify candidates to be further analyzed, derived from their potential metabolic and phenotypic profile. For this purpose *PhenoPointer* was developed, enabling fast and accurate phenotypic characterization of novel organisms. The predictable phenotypes include observable phenotypes such as structural properties (gram staining, cell shape, cell arrangement) over microbial traits describing their lifestyle for gathering carbon and energy, encoded metabolic pathways, antibiotic resistances or synthesis capability, through to consumed substrates and produced metabolites, grouped into 13 phenotype categories. The easy to use CLI accepts hmmer output against Pfam v.29 or the genome sequences as FASTA format as complete genome or as multiple contigs. It does not require any programming knowledge; the only step needed is to initially define the path to the Pfam resource directory.

The grouping of the phenotypes follow the metadata annotation categories of organisms in *IMG/M*, so a reference check in this much larger data mart is possible. The predictive strength of all 13 ML classifiers has been shown, even on heavily contaminated and incomplete genomes, confirmed by a validation data set of organisms not used for selecting the best performing ML model. Further improvements of the classification models in future releases and possible current drawbacks have been stated in the discussion sections, particularly for each phenotype category in Chapter '7 – *PhenoPointer* – 13 classifiers for phenotype prediction'.

It should also be mentioned if a prediction on a specific phenotype had not resulted in an assigned label to an input sample, that this prediction is not an excluding criteria, as it could be a missed classification (multilabel classification problem) or an undecidable problem for a specific input (multiclass/ binary classification problem).

Although *PhenoPointer* gives, in general, more of an overview about encoded metabolic features, specializations of substrates can also be predicted, as shown in the comparison of *PhenoPointer* against *Traitar*, the only known tool, with a comparable set of microbial traits and observable phenotypes as classification targets. The results suggest that *PhenoPointer* performs better than *Traitar*, which is true for some phenotype categories, but a precise comparison is hard to achieve because of the different prediction focus. The proposed specializations of classification models in Chapter 7 through subdividing the phenotype categories *Metabolism* and *Phenotype* may lead to better prediction results in the scope of utilized substrates. It would be extremely interesting to see how *PhenoPointer* will perform in such classification scenarios in comparison to *Traitar*, just because of the relatively poor prediction performance shown by *Traitar* on substrates, which reflects the main application focus of *Traitar*.

With *PhenoPointer* organisms belonging to the microbial dark matter can be directly analyzed to predict their preferred environmental conditions for optimal growth including their lifestyle in terms of primary sources of carbon and energy, enabling successful cultivation. If no such cultivation is desired, the prediction made on the metabolic features can be used for identifying substantial enzymatic reactions and metabolic features, thus limiting the set of possible KEGG pathways to map protein sequences to and manually inspect the coverage of a pathway. In addition, novel and so far unexpected functional capabilities can be detected, acting as new hypotheses.

Besides these advantages of functional profiling, *PhenoPointer* is suitable for minimizing the effort needed to perform culturomics studies by narrowing down the set of possible cultivation parameters in terms of environmental conditions for optimal growth and giving hints to possible encoded primary energy and carbon source and even utilized substrates.

While evaluating how to set the time frame to define the validation set from *IMG/M*, actually defined as all annotated new genomes within *IMG/M* from March 2016 till April 2017, an alarming trend could be observed even in a well curated metadata carrying data mart such as *IMG/M*: Although 18328 organisms have been added to the database, only a minimal fraction of these are annotated with organism-related metadata describing their lifestyle as environmental parameters and functional capabilities (shown in Table 6.1 on page 62). The most annotated phenotype category is *Gram Staining* with 835 annotated organisms, followed by *Oxygen Requirement* with 462 annotated organisms and *Temperature Range* with 315 annotated organisms. Comparing the fractions of annotated organisms to the total amount of organisms in the March 2016 and the April 2017 release of *IMG/M*, this is a drop from 47.61% to 32.09% of genomes annotated with metadata about *Gram Staining* and a drop from 24.54% down to 15.73% for *Temperature Range*. This may have been caused by an inflationary generation of sequences without any further documentation or analysis; only a taxonomic assignment was performed before uploading and publishing the genome, or a slight idleness in documentation, but this assumption would be very harsh.

To overcome this big issue of uncharacterized organisms in metadata-dependent data marts such as *IMG/M* and *GOLD*, *PhenoPointer* can easily be applied to these genomic sequences for filling the annotational gap from genotype to phenotype, allowing further functional studies.

The benefit of a metadata-enrichment study with *MVIZ* is that it extends the scope of application of a marker gene-based analysis by switching the view from a taxonomy profile to observable phenotypes, meaning that these can be directly assessed without any further computational and wet lab work. The only limitation is that the metadata-enriched profile visualizes only those organisms that are known. It may also be the case that these organisms are not entirely annotated, but the missing metadata labels can be internally computed by *PhenoPointer* and stored in *MetaStone*, as described in Chapter '4.2 – The data model for persistence' from page 34 onwards, so this is no longer an issue. An example of this was shown in Figure 10.3, but it was also shown that working with untrusted metadata may also introduce conflicting annotations, so further checking is advised.

As shown, errors in study-related metadata about the sampling environments can be detected, averting false research conclusions or misguided further research. It can be assessed, based on a shown metabolic profile, whether further on-going WMGS sequencing should be performed if a marker gene study was performed as a pilot study to evaluate the potential of a microbial community.

Functional profiling in terms of preferred lifestyles and environmental conditions can easily be performed, as shown in Figure 10.2, enabling explorative definition of core functional phenotypes and core metabolic pathways. The identification of shared and distinct metadata annotations over a set of samples allows one to draw conclusions about the inhabited environments, because the presented annotations of metadata-enriched profiles represent the true nature of the sample in relation to their substrates and environmental conditions. Inspection of shared and distinct phenotypes on a community level basis enables the formulation of new hypotheses if an unexpected observation can be made and thus represents a novelty especially when comparing samples originating from different environments. This kind of profiling is useful when investigating a time series of samples, for observing the adaptions and adjustments performed by a microbial community as (stress) response reactions to changed environmental conditions, because in general the taxonomic profile including abundances had so far been the only observable reference, but now the view can change to phenotypic and trait shifts in time.

This type of analysis is not limited to marker gene-based studies; WMGS sequencing-derived community profiles can also be enriched with metadata, allowing fast and accurate screening of concealed microbial traits and phenotypes within known organisms. Initial studies (not shown in this thesis) prior to *MVIZ* indicated that a metadata-enrichment can potentially result in very viable conclusions by looking at a WMGS metadata-enriched community profile. This is useful when a specific metabolic reaction is expected to occur in the microbial community, and is also verified by an enriched community profile, so no overwhelming outcome via a further functional pathway study is guaranteed anymore, becsause the main drivers have already been identified.

MVIZ is currently not capable of visualizing organisms belonging to unknown taxa. PICRUST (Langille et al. 2013) can perform functional profiling on the basis of 16s rRNA marker genes by aligning a 16s rRNA marker gene sequence against a taxonomic reference database and performing an ancestral state reconstruction until an annotated ancestor can be identified. A similar kind of approach is used for metadata-enrichment within *MetaStone* for missing annotation metadata labels, as described in Chapter 9.1.

12 Summary

The aim of this thesis was to evaluate the use of metadata in the research field of metagenomics and to define use cases that lead to practical applications relying on metadata as the primary source of information. The outcomes have been *PhenoPointer* and *MVIZ*, both utilizing metadata-descriptive traits and observable phenotypes of microbial organisms. The bridging element between these two applications is *MetaStone*, which functions as the main data repository for storing and accessing organisms, their phenotypic metadata and Pfam abundance profiles. The organisms and their correlated metadata as well as the Pfam abundance profile have been exported from *IMG/M*.

One of the main issues of metagenomics is the composition of microbial community structures, their lifestyles, and how they influence each other in direct relation to their environment. To answer this question, the discovery of enzymatic reaction and thus reconstruction of metabolic pathways is essential. Apart from this, metagenomics enables the cultivation-free reconstruction of hitherto unknown microbial organisms belonging to the microbial dark matter. A leading scientific program to investigate this microbial dark matter is the Genomic Encyclopaedia of Bacteria and Archaea (GEBA) project, where hundreds of new microbial organisms have already been discovered with more to come (Kyrpides, Woyke et al. 2014; Kyrpides, Hugenholtz et al. 2014). Thus an efficient and scalable way to characterize these novel organisms on a phenotypic way is needed. It could be shown in this thesis that organisms newly published in *IMG/M* within a period of 13 months were poorly described, as be seen from the organism related metadata fields describing phenotypes.

To address these demands *PhenoPointer* was developed to predict microbial traits and observable phenotypes quickly and accurately, so the above-mentioned issues can be addressed in a time- and cost efficient manner. The methodology behind *PhenoPointer* is that the phenotypes are categorized beforehand into 13 categories, and a particular ML classification model trained for each category. Several classification methods have been evaluated in 10-fold cross validation experiments with different parameter setups separately for each phenotype category, resulting in a set of strong classification models. *PhenoPointer* was very successfully tested on 22 hitherto unseen organisms and often predicted their phenotypes with 100% precision. Comparing these results with those from the validation set, *PhenoPointer* performed surprisingly better than expected and validated. The only known competitive tool to *PhenoPointer*, *Traitar*, is specialized on the prediction of utilized substrates and was clearly outperformed, even in its own specialized prediction cases.

This tool can thus help annotate novel organisms in an accurate, comprehensive and diversified way, covering the most important parameters needed for estimating optimal growth conditions for cultivation in the laboratory, limiting the effort of brute force approaches (Lagier, Edouard et al. 2015), particularly in the field of clinical studies (Mulcahy-O'Grady et al. 2016; Balasopoulou et al. 2016) and health care (Berendonk et al. 2015).

The maintainers of *IMG/M* have used a rule-based predictor for some microbial traits (I.-M. A. Chen, Markowitz, Chu, Anderson et al. 2013), but it is questionable whether this tool is still used for annotation purposes, because of the high number of new organisms added ($\sim$ 18000) to *IMG/M* in a period of thirteen months, only a few of which have been annotated. *PhenoPointer* may help in this case, following further manual control.

Another use case of organism related metadata is the enrichment of microbial community profiles to enable comparisons of multiple samples not only by their taxonomic composition but also by the aggregated traits and observable phenotypes. This also makes marker gene-based sequences studies possible, to define a core set of metabolic pathways, nutrients, and enzymatic functions (Litchman et al. 2015). *MVIZ* is the answer to this demand through enriching community profiles with organism related metadata. It also takes advantage of *PhenoPointer* and its unique property of making trustworthy predictions on bacterial and archaeal traits as well as observable phenotypes, filling the gaps of missing metadata annotations from *IMG/M*.

The influence on marker gene sequencing studies has been illustrated with the case study of four biogas plants, where a core set of functions shared among all biogas plants was defined, but also a correction on vital project metadata was discovered and corrected.

The combination of both applications can lead to function-driven metagenomics (Woyke et al. 2015), because the limiting factor is no longer the data but workforce and expertise. The demand for such tools is still there (Fierer et al. 2014; Hanemaaijer et al. 2015), and increasing as new upcoming analytical technologies emerge.

Metadata & future prospect Predictions representing the real nature of an organism hinge on the question of the underlying controlled vocabulary, as could be shown in the phenotype *Diseases*, *Metabolism*, and *Phenotype*. The predictive performance in these categories was not bad, but not as good as the more specialized categories such as *Cell Shape*, so a detailed manual control of the represented values by domain experts is needed. After manual control of the metadata values, a subsequent split of some categories can result in finer granularity and the generation of new functional phenotype categories.

Adding new data sources as features or prediction targets is not feasible because the feature space should be kept as specific as needed and as general as possible, and for prediction targets high quality reference data is needed. The Comprehensive Antibiotic Resistance Database (CARD) (Jia et al. 2017) has been evaluated during the implementation phase of *PhenoPointer*, but at that time point CARD contained only $\sim$ 1000 reference sequences and only a few were assigned to the same antibiotic term, so CARD was discarded.

A more promising extension would be the application of metadata-based clustering and classification at metagenome level. Within such a study, clusters of core organisms would be easily identifiable, and metagenomes could be clustered by their microbial composition. A method to perform such clustering would be document based clustering (topic modelling) (Blei et al. 2003) as already performed in metagenomics for sequence clustering (R. Zhang et al. 2015; Gkanogiannis et al. 2016), but with taxonomical community profiles rather than sequencing data as input. Again, *IMG/M* and *GOLD* may act as the primary data source for metadata because, as for bacterial and archaeal organisms, metagenomes are annotated with sample-descriptive metadata as a controlled vocabulary (Mukherjee, Stamatis et al. 2017; Field et al. 2014).

Bibliography

Acosta-Martinez, V., J. Cotton, T. Gardner et al. (Dec. 2014). 'Predominant bacterial and fungal assemblages in agricultural soils during a record drought/heat wave and linkages to enzyme activities of biogeochemical cycling'. In: *Applied Soil Ecology* 84, pp. 69–82.

Aggarwal, C. C., M. A. Bhuiyan and M. Al Hasan (2014). 'Frequent Pattern Mining Algorithms: A Survey'. In: *Frequent Pattern Mining.* Ed. by C. C. Aggarwal and J. Han. Springer International Publishing, pp. 19–64.

Alami, Y., W. Achouak, C. Marol et al. (Aug. 2000). 'Rhizosphere soil aggregation and plant growth promotion of sunflowers by an exopolysaccharide-producing Rhizobium sp. strain isolated from sunflower roots.' In: *Applied and Environmental Microbiology* 66.8, pp. 3393–3398.

Alexandrino, D. A. M., A. P. Mucha, C. M. R. Almeida et al. (Mar. 2017). 'Biodegradation of the veterinary antibiotics enrofloxacin and ceftiofur and associated microbial community dynamics.' In: *The Science of the total environment* 581-582, pp. 359–368.

Alvensleben, N. von, K. Stookey, M. Magnusson et al. (2013). 'Salinity tolerance of Picochlorum atomus and the use of salinity for contamination control by the freshwater cyanobacterium Pseudanabaena limnetica.' In: *PloS one* 8.5, e63569.

Amann, R. I., W. Ludwig and K. H. Schleifer (Mar. 1995). 'Phylogenetic identification and in situ detection of individual microbial cells without cultivation.' In: *Microbiology and Molecular Biology Reviews* 59.1, pp. 143–169.

Anantharaman, K., C. T. Brown, L. A. Hug et al. (Oct. 2016). 'Thousands of microbial genomes shed light on interconnected biogeochemical processes in an aquifer system'. In: *Nature Communications* 7, p. 13219.

Andrews, S. (2010). *FastQC: a quality control tool for high throughput sequence data.* URL: `https://www.bioinformatics.babraham.ac.uk/projects/fastqc/`.

Antranikian, G., M. Suleiman, C. Schäfers et al. (May 2017). 'Diversity of bacteria and archaea from two shallow marine hydrothermal vents from Vulcano Island.' In: *Extremophiles: life under extreme conditions* 1.1, pp. 1–10.

Ashburner, M., C. A. Ball, J. A. Blake et al. (May 2000). 'Gene Ontology: tool for the unification of biology'. In: *Nature Genetics* 25.1, pp. 25–29.

Baker, B. J. and G. J. Dick (Sept. 2013). 'Omic approaches in microbial ecology: charting the unknown'. In: *Microbe Magazine* 8.9, pp. 353–360.

Balasopoulou, A., G. P. Patrinos and T. Katsila (2016). 'Pharmacometabolomics Informs Viromics toward Precision Medicine'. In: *Frontiers in pharmacology* 7, p. 411.

Bankevich, A., S. Nurk, D. Antipov et al. (May 2012). 'SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing.' In: *Journal of computational biology : a journal of computational molecular cell biology* 19.5, pp. 455–477.

Barlow, S. T. and P. Neville (2001). 'Case Study: Visualization for Decision Tree Analysis in Data Mining.' In: *infovis*, pp. 149–152.

Barnes, M., A. D. Ansell and R. N. Gibson (1992). 'The biology of hydrothermal vent animals: physiology, biochemistry, and autotrophic symbioses'. In: *Oceanogr. Mar. Biol. Annu. Rev* 30, pp. 337–441.

Bashiardes, S., G. Zilberman-Schapira and E. Elinav (2016). 'Use of Metatranscriptomics in Microbiome Research'. In: *Bioinformatics and Biology Insights* 10, pp. 19–25.

Bayes, T. and R. Price (1763). ‘An essay towards solving a problem in the doctrine of chances’. In: *Philosophical Transactions of the Royal Society of London* 53, pp. 370–418.

Bellman, R. E. (1961). *Adaptive Control Processes.* A Guided Tour. Princeton University Press.

Berendonk, T. U., C. M. Manaia and C. Merlin (2015). ‘Tackling antibiotic resistance: the environmental framework’. In: *Nature Reviews Microbiology* 13.5, pp. 310–317.

Berger, S. A. (2005). ‘GIDEON: a comprehensive Web-based resource for geographic medicine’. In: *International Journal of Health Geographics* 4.1, p. 10.

Berkhin, P. (2006). ‘A Survey of Clustering Data Mining Techniques’. In: *Grouping Multidimensional Data.* Springer-Verlag, pp. 25–71.

Bevivino, A., P. Paganin, G. Bacci et al. (2014). ‘Soil Bacterial Community Response to Differences in Agricultural Management along with Seasonal Changes in a Mediterranean Region’. In: *PloS one* 9.8, e105515.

Bikel, S., A. Valdez-Lara, F. Cornejo-Granados et al. (2015). ‘Combining metagenomics, metatranscriptomics and viromics to explore novel microbial interactions: towards a systems-level understanding of human microbiome’. In: *Computational and Structural Biotechnology Journal* 13, pp. 390–401.

Bishop, C. M. (Aug. 2006). *Pattern Recognition and Machine Learning.* Springer.

Bland, C., T. L. Ramsey, F. Sabree et al. (June 2007). ‘CRISPR recognition tool (CRT): a tool for automatic detection of clustered regularly interspaced palindromic repeats.’ In: *BMC bioinformatics* 8.209.

Blei, D. M., A. Y. Ng and M. I. Jordan (2003). ‘Latent Dirichlet Allocation’. In: *Journal of Machine Learning Research* 3.Jan, pp. 993–1022.

Bode, H. B. (Sept. 2015). ‘The Microbes inside Us and the Race for Colibactin’. In: *Angewandte Chemie International Edition* 54.36, pp. 10408–10411.

Bolger, A. M., M. Lohse and B. Usadel (Aug. 2014). ‘Trimmomatic: a flexible trimmer for Illumina sequence data’. In: *Bioinformatics* 30.15, pp. 2114–2120.

Boser, B. E., I. M. Guyon and V. N. Vapnik (July 1992). ‘A training algorithm for optimal margin classifiers’. In: *COLT ’92 Proceedings of the fifth annual workshop on Computational learning theory.* ACM, pp. 144–152.

Bramer, M. (2016). *Principles of Data Mining.* London: Springer.

Bray, T. (2014). *The JavaScript Object Notation (JSON) Data Interchange Format.* URL: `https://tools.ietf.org/pdf/rfc7159.pdf`.

Breiman, L. (2001). ‘Random Forests’. In: *Machine learning* 45.1, pp. 5–32.

Breiman, L., J. Friedman, C. J. Stone et al. (1984). *Classification and regression trees.* CRC Press.

Bretaudeau, A., F. Coste, F. Humily et al. (Jan. 2013). ‘CyanoLyase: a database of phycobilin lyase sequences, motifs and functions.’ In: *Nucleic acids research* 41.Database issue, pp. D396–401.

Brooks, J. P., D. J. Edwards, M. D. Harwich et al. (Mar. 2015). ‘The truth about metagenomics: quantifying and counteracting bias in 16S rRNA studies.’ In: *BMC microbiology* 15.1.

Bruno, A., A. Sandionigi, E. Rizzi et al. (Mar. 2017). ‘Exploring the under-investigated "microbial dark matter" of drinking water treatment plants.’ In: *Scientific reports* 7.

Buchfink, B., C. Xie and D. H. Huson (Jan. 2015). ‘Fast and sensitive protein alignment using DIAMOND’. In: *Nature Methods* 12.1, pp. 59–60.

Burbulys, D., K. A. Trach and J. A. Hoch (Feb. 1991). ‘Initiation of sporulation in B. subtilis is controlled by a multicomponent phosphorelay’. In: *Cell* 64.3, pp. 545–552.

Buttigieg, P. L., N. Morrison, B. Smith et al. (Dec. 2013). ‘The environment ontology: contextualising biological and biomedical entities’. In: *Journal of biomedical semantics* 4.1.

Buttigieg, P. L., E. Pafilis, S. E. Lewis et al. (Sept. 2016). ‘The environment ontology in 2016: bridging domains with increased scope, semantic density, and interoperation.’ In: *Journal of biomedical semantics* 7.1.

Cabeen, M. T. and C. Jacobs-Wagner (Aug. 2005). ‘Bacterial cell shape.’ In: *Nature reviews. Microbiology* 3.8, pp. 601–610.

Caporaso, J. G., J. Kuczynski, J. Stombaugh et al. (May 2010). ‘QIIME allows analysis of high-throughput community sequencing data’. In: *Nature Methods* 7.5, pp. 335–336.

Caspi, R., T. Altman, R. Billington et al. (Jan. 2014). ‘The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases.’ In: *Nucleic acids research* 42.Database issue, pp. D459–D471.

Chen, I.-M. A., V. M. Markowitz, K. Chu, I. Anderson et al. (Dec. 2013). ‘Improving Microbial Genome Annotations in an Integrated Database Context’. In: *PloS one* 8.2, e54859.

Chen, I.-M. A., V. M. Markowitz, K. Chu, K. Palaniappan et al. (Jan. 2017). ‘IMG/M: integrated genome and metagenome comparative data analysis system.’ In: *Nucleic acids research* 45.D1, pp. D507–D516.

Chen, T., W.-H. Yu, J. Izard et al. (July 2010). ‘The Human Oral Microbiome Database: a web accessible resource for investigating oral microbe taxonomic and genomic information.’ In: *Database: the journal of biological databases and curation* 2010.

Choct, M. (Mar. 2006). ‘Enzymes for the feed industry: past, present and future’. In: *World’s Poultry Science Journal* 62.1, pp. 5–16.

Choi, E. J., H. M. Jin, S. H. Lee et al. (Jan. 2013). ‘Comparative genomic analysis and benzene, toluene, ethylbenzene, and o-, m-, and p-xylene (BTEX) degradation pathways of Pseudoxanthomonas spadix BD-a59.’ In: *Applied and Environmental Microbiology* 79.2, pp. 663–671.

Cody, M. L. (1975). ‘Towards a theory of continental species diversities: bird distributions over Mediterranean habitat gradients’. In: *Ecology and evolution of communities* 545, pp. 214–257.

Coughlan, L. M., P. D. Cotter, C. Hill et al. (June 2015). ‘Biotechnological applications of functional metagenomics in the food and pharmaceutical industries’. In: *Frontiers in microbiology* 6, p. 672.

Cover, T. and P. Hart (1967). ‘Nearest neighbor pattern classification’. In: *IEEE transactions on information theory* 13.1, pp. 21–27.

D’Amico, S., T. Collins, J. C. Marx et al. (Apr. 2006). ‘Psychrophilic microorganisms: challenges for life’. In: *EMBO reports* 7.4, pp. 385–389.

Dahal, R. H., D. K. Chaudhary and J. Kim (Feb. 2017). ‘Acinetobacter halotolerans sp. nov., a novel halotolerant, alkalitolerant, and hydrocarbon degrading bacterium, isolated from soil.’ In: *Archives of microbiology*, pp. 1–10.

Dasarathy, B. V. and B. V. Sheela (1979). ‘A composite classifier system design: Concepts and methodology’. In: *Proceedings of the IEEE* 67.5, pp. 708–713.

De La Calleja, J. and O. Fuentes (Mar. 2004). ‘Machine learning and image analysis for morphological galaxy classification’. In: *Monthly Notices of the Royal Astronomical Society* 349.1, pp. 87–93.

DeSantis, T. Z., I. Dubosarskiy, S. R. Murray et al. (Aug. 2003). ‘Comprehensive aligned sequence construction for automated design of effective probes (CASCADE-P) using 16S rDNA.’ In: *Bioinformatics* 19.12, pp. 1461–1468.

DeSantis, T. Z., P. Hugenholtz, N. Larsen et al. (July 2006). ‘Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB.’ In: *Applied and Environmental Microbiology* 72.7, pp. 5069–5072.

Devaraj, S., P. Hemarajata and J. Versalovic (Apr. 2013). ‘The human gut microbiome and body metabolism: implications for obesity and diabetes.’ In: *Clinical chemistry* 59.4, pp. 617–628.

Dewhirst, F. E., T. Chen, J. Izard et al. (Oct. 2010). ‘The human oral microbiome.’ In: *Journal of Bacteriology* 192.19, pp. 5002–5017.

Dewhirst, F. E., E. A. Klein, M.-L. Bennett et al. (Feb. 2015). 'The feline oral microbiome: a provisional 16S rRNA gene based taxonomy with full-length reference sequences.' In: *Veterinary microbiology* 175.2-4, pp. 294–303.

Dickson, I. (Jan. 2016). 'Gut microbiota: Culturomics: illuminating microbial dark matter.' In: *Nature reviews. Gastroenterology & hepatology* 14.1, p. 3.

Dion, P. and C. S. Nautiyal, eds. (2008). *Microbiology of Extreme Soils.* Vol. 13. Soil Biology. Berlin, Heidelberg: Springer.

Django Software Foundation (2016). *Django (Version 1.9).* URL: `https://djangoproject.com`.

DOE Joint Genome Institute (2014). *BBTools - DOE Joint Genome Institute.* URL: `http://jgi.doe.gov/data-and-tools/bbtools/`.

Doi, M., M. Wachi, F. Ishino et al. (Oct. 1988). 'Determinations of the DNA sequence of the mreB gene and of the gene products of the mre region that function in formation of the rod shape of Escherichia coli cells.' In: *Journal of Bacteriology* 170.10, pp. 4619–4624.

Dopson, M., C. Baker-Austin, P. R. Koppineedi et al. (Aug. 2003). 'Growth in sulfidic mineral environments: metal resistance mechanisms in acidophilic micro-organisms'. In: *Microbiology* 149.8, pp. 1959–1970.

Doroshenko, E. V., E. S. Bulygina, E. M. Spiridonova et al. (Jan. 2007). 'Isolation and characterization of nitrogen-fixing bacteria of the genus Azospirillum from the soil of a Sphagnum peat bog.' In: *Mikrobiologiia* 76.1, pp. 107–115.

Dusko Ehrlich, S. and MetaHIT consortium (Sept. 2010). 'Metagenomics of the intestinal microbiota: potential applications.' In: *Gastroenterologie clinique et biologique* 34 Suppl 1, S23–S28.

Edwards, A., A. R. Debbonaire, B. Sattler et al. (Sept. 2016). 'Extreme metagenomics using nanopore DNA sequencing: a field report from Svalbard, 78 N'. In: *bioRxiv*, p. 073965.

Efron, B., T. Hastie, I. Johnstone et al. (Apr. 2004). 'Least angle regression'. In: *The Annals of Statistics* 32.2, pp. 407–499.

Ezzat, A., M. N. M. Isa, I. Sapian et al. (2014). 'Metagenomic Study of the Liver Microbiota in Liver Cancer-Metagenomic and Metatranscriptomic Analyses of the Hepatocellular Carcinoma-Associated Microbial Communities and the Potential Role of Microbial Communities in Liver Cancer'. In: *Journal of Gastrointestinal & Digestive System* 4.228, p. 2.

Faure, M. C., E. Zundel, S. Aubin et al. (2011). *Le référentiel maladies animales: de CERISA au Linked Data.* URL: `http://agroportal.lirmm.fr/ontologies/ADO`.

Federhen, S. (Jan. 2012). 'The NCBI Taxonomy database'. In: *Nucleic acids research* 40.D1, pp. D136–D143.

Federhen, S., K. Clark, T. Barrett et al. (June 2014). 'Toward richer metadata for microbial sequences: replacing strain-level NCBI taxonomy taxids with BioProject, BioSample and Assembly records.' In: *Standards in Genomic Sciences* 9.3, pp. 1275–1277.

Field, D., P. Sterk, R. Kottmann et al. (June 2014). 'Genomic Standards Consortium Projects'. In: *Standards in Genomic Sciences* 9.3, pp. 599–601.

Fierer, N., A. Barberán and D. C. Laughlin (2014). 'Seeing the forest for the genes: using metagenomics to infer the aggregated traits of microbial communities.' In: *Frontiers in microbiology* 5, p. 614.

Finn, R. D., J. Clements, W. Arndt et al. (July 2015). 'HMMER web server: 2015 update.' In: *Nucleic acids research* 43.W1, W30–W38.

Finn, R. D., P. Coggill, R. Y. Eberhardt et al. (Jan. 2016). 'The Pfam protein families database: towards a more sustainable future.' In: *Nucleic acids research* 44.D1, pp. D279–285.

Fisher, R. A., A. S. Corbet and C. B. Williams (May 1943). 'The Relation Between the Number of Species and the Number of Individuals in a Random Sample of an Animal Population'. In: *The Journal of Animal Ecology* 12.1, pp. 42–58.

Fitz-Gibbon, S., S. Tomida, B.-H. Chiu et al. (Sept. 2013). ‘Propionibacterium acnes strain populations in the human skin microbiome associated with acne.’ In: *The Journal of investigative dermatology* 133.9, pp. 2152–2160.

Fix, E. and J. L. Hodges Jr (Feb. 1951). ‘Discriminatory Analysis - Nonparametric Discrimination: Consistency Properties’. In: *Technical Report 4, USAF School of Aviation Medicine.*

Freund, Y. and R. E. Schapire (Mar. 1995). ‘A desicion-theoretic generalization of on-line learning and an application to boosting’. In: *Computational Learning Theory.* Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp. 23–37.

Friedman, J. H. (2001). ‘Greedy function approximation: a gradient boosting machine’. In: *Annals of statistics* 29.5, pp. 1189–1232.

Galperin, M. Y., K. S. Makarova, Y. I. Wolf et al. (Jan. 2015). ‘Expanded microbial genome coverage and improved protein family annotation in the COG database’. In: *Nucleic acids research* 43.D1, pp. D261–D269.

Gilbert, J. A., J. K. Jansson and R. Knight (Aug. 2014). ‘The Earth Microbiome project: successes and aspirations’. In: *BMC biology* 12.69.

Gkanogiannis, A., S. Gazut, M. Salanoubat et al. (Aug. 2016). ‘A scalable assembly-free variable selection algorithm for biomarker discovery from metagenomes’. In: *BMC bioinformatics* 17.1.

Golub, G. H. and C. Reinsch (1970). ‘Singular value decomposition and least squares solutions’. In: *Numerische Mathematik* 14.5, pp. 403–420.

Gomez, J. E. and J. D. McKinney (2004). ‘M. tuberculosis persistence, latency, and drug tolerance.’ In: *Tuberculosis* 84.1, pp. 29–44.

González-Pastor, J. E., E. C. Hobbs and R. Losick (July 2003). ‘Cannibalism by sporulating bacteria.’ In: *Science* 301.5632, pp. 510–513.

Google Inc. (2010). *AngularJS.* URL: `https://angularjs.org`.

Gray, J., D. T. Liu, M. Nieto-Santisteban et al. (Dec. 2005). ‘Scientific data management in the coming decade’. In: *ACM SIGMOD Record* 34.4, pp. 34–41.

Gray, M. W., D. Sankoff and R. J. Cedergren (July 1984). ‘On the evolutionary descent of organisms and organelles: a global phylogeny based on a highly conserved structural core in small subunit ribosomal RNA’. In: *Nucleic acids research* 12.14, pp. 5837–5852.

Green, B. and S. Seshadri (Apr. 2013). *AngularJS.* O’Reilly Media, Inc.

Haeckel, E. (1866). *Generelle Morphologie der Organismen.* Allgemeine Grundzüge der organischen Formenwissenschaft, mechanisch begründet durch die von Charles Darwin reformierte Descendenz-Theorie. Verlag Georg Reimer.

Hahnke, S., T. Langer, D. E. Koeck et al. (Mar. 2016). ‘Description of Proteiniphilum saccharofermentans sp. nov., Petrimonas mucosa sp. nov. and Fermentimonas caenicola gen. nov., sp. nov., isolated from mesophilic laboratory-scale biogas reactors, and emended description of the genus Proteiniphilum’. In: *Escherichia Coli and Salmonella Cellular and Molecular Biology.* Microbiology Society, pp. 1466–1475.

Han, J., J. Pei and M. Kamber (June 2011). *Data Mining: Concepts and Techniques.* Elsevier.

Hanemaaijer, M., W. F. M. Röling, B. G. Olivier et al. (2015). ‘Systems modeling approaches for microbial community studies: from metagenomics to inference of the community structure.’ In: *Frontiers in microbiology* 6.299.

Hankinson, T. R. and E. L. Schmidt (June 1988). ‘An acidophilic and a neutrophilic nitrobacter strain isolated from the numerically predominant nitrite-oxidizing population of an Acid forest soil.’ In: *Applied and Environmental Microbiology* 54.6, pp. 1536–1540.

Hansen, L. K. and P. Salamon (1990). ‘Neural network ensembles’. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.10, pp. 993–1001.

Harshey, R. M. (Aug. 1994). ‘Bees aren’t the only ones: swarming in gram-negative bacteria.’ In: *Molecular microbiology* 13.3, pp. 389–394.

Hastie, T., R. Tibshirani and J. Friedman (Aug. 2009). *The Elements of Statistical Learning.* Data Mining, Inference, and Prediction, Second Edition. Springer.

Hautala, T., H. Syrjälä, V. Lehtinen et al. (Apr. 2005). 'Blood culture Gram stain and clinical categorization based empirical antimicrobial therapy of bloodstream infection.' In: *International journal of antimicrobial agents* 25.4, pp. 329–333.

Hedlund, B. P., J. A. Dodsworth, S. K. Murugapiran et al. (Sept. 2014). 'Impact of single-cell genomics and metagenomics on the emerging view of extremophile "microbial dark matter".' In: *Extremophiles: life under extreme conditions* 18.5, pp. 865–875.

Hellberg, R. S. and E. Chu (Aug. 2016). 'Effects of climate change on the persistence and dispersal of foodborne bacterial pathogens in the outdoor environment: A review.' In: *Critical reviews in microbiology* 42.4, pp. 548–572.

Henrich, B., M. Rumming, A. Sczyrba et al. (2014). 'Mycoplasma salivarium as a dominant coloniser of Fanconi anaemia associated oral carcinoma.' In: *PloS one* 9.3, e92297.

Henrichsen, J. (1983). 'Twitching motility.' In: *Annual review of microbiology* 37.1, pp. 81–93.

Hess, M., A. Sczyrba, R. Egan et al. (Jan. 2011). 'Metagenomic discovery of biomass-degrading genes and genomes from cow rumen.' In: *Science* 331.6016, pp. 463–467.

Ho, T. K. (1998). 'The random subspace method for constructing decision forests'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.8, pp. 832–844.

Honeyman, A. L. and G. C. Stewart (Sept. 1989). 'The nucleotide sequence of the rodC operon of Bacillus subtilis.' In: *Molecular microbiology* 3.9, pp. 1257–1268.

Hugenholtz, P., A. Skarshewski and D. H. Parks (June 2016). 'Genome-Based Microbial Taxonomy Coming of Age.' In: *Cold Spring Harbor perspectives in biology* 8.6.

Hughes, J. B., J. J. Hellmann, T. H. Ricketts et al. (Oct. 2001). 'Counting the uncountable: statistical approaches to estimating microbial diversity.' In: *Applied and Environmental Microbiology* 67.10, pp. 4399–4406.

Human Microbiome Jumpstart Reference Strains Consortium, K. E. Nelson, G. M. Weinstock et al. (May 2010). 'A catalog of reference genomes from the human microbiome.' In: *Science* 328.5981, pp. 994–999.

Huntemann, M., N. N. Ivanova, K. Mavromatis et al. (Oct. 2015). 'The standard operating procedure of the DOE-JGI Microbial Genome Annotation Pipeline (MGAP v.4)'. In: *Standards in Genomic Sciences* 10.1.

Huo, W., H. M. Adams, M. Q. Zhang et al. (June 2015). 'Genome Modification in Enterococcus faecalis OG1RF Assessed by Bisulfite Sequencing and Single-Molecule Real-Time Sequencing'. In: *Journal of Bacteriology* 197.11, pp. 1939–1951.

Huson, D. H., A. F. Auch, J. Qi et al. (Mar. 2007). 'MEGAN analysis of metagenomic data.' In: *Genome research* 17.3, pp. 377–386.

Hyatt, D., G.-L. Chen, P. F. Locascio et al. (Mar. 2010). 'Prodigal: prokaryotic gene recognition and translation initiation site identification.' In: *BMC bioinformatics* 11.1.

Imelfort, M. and D. Edwards (Nov. 2009). 'De novo sequencing of plant genomes using second-generation technologies.' In: *Briefings in bioinformatics* 10.6, pp. 609–618.

Ishoey, T., T. Woyke, R. Stepanauskas et al. (June 2008). 'Genomic sequencing of single microbial cells from environmental samples.' In: *Current opinion in microbiology* 11.3, pp. 198–204.

James, G., D. Witten, T. Hastie et al. (2013). *An Introduction to Statistical Learning.* Ed. by G. Casella, S. Fienberg and I. Olkin. Vol. 103. Springer Texts in Statistics. Springer.

Jia, B., A. R. Raphenya, B. Alcock et al. (Jan. 2017). 'CARD 2017: expansion and model-centric curation of the comprehensive antibiotic resistance database.' In: *Nucleic acids research* 45.D1, pp. D566–D573.

Johnson, C. H., J. Ivanisevic and G. Siuzdak (July 2016). 'Metabolomics: beyond biomarkers and towards mechanisms'. In: *Nature Reviews Molecular Cell Biology* 17.7, pp. 451–459.

Jolliffe, I. (2002). *Principal Component Analysis.* 2nd ed. Vol. 1. John Wiley & Sons, Ltd.

Kambouris, M. E., C. Pavlidis, E. Skoufas et al. (Apr. 2017). 'Culturomics: A New Kid on the Block of OMICS to Enable Personalized Medicine.' In: *Omics: a journal of integrative biology* 22.2.

Kanehisa, M. (1996). 'Toward pathway engineering: a new database of genetic and molecular pathways'. In: *Science and Technology Japan* 59, pp. 34–38.

Kanehisa, M., M. Furumichi, M. Tanabe et al. (Jan. 2017). 'KEGG: new perspectives on genomes, pathways, diseases and drugs.' In: *Nucleic acids research* 45.D1, pp. D353–D361.

Kang, D. D., J. Froula, R. Egan et al. (Aug. 2015). 'MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities'. In: *PeerJ* 3.8, e1165.

Katz, P. (2007). *ZIP File Format Specification, 2007. version 6.3.* URL: `http://www.pkware.com/documents/casestudies/APPNOTE`.

Kibbe, W. A., C. Arze, V. Felix et al. (Jan. 2015). 'Disease Ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data.' In: *Nucleic acids research* 43.Database issue, pp. D1071–D1078.

Kirschner, R., T. Hsu, N. Ngoc Tuan et al. (2015). 'Characterization of Fungal and Bacterial Components in Gut/Fecal Microbiome'. In: *Current Drug Metabolism* 16.4, pp. 272–283.

Knight, R., J. Jansson, D. Field et al. (June 2012). 'Unlocking the potential of metagenomics through replicated experimental design.' In: *Nature Biotechnology* 30.6, pp. 513–520.

Koeck, D. E., I. Maus, D. Wibberg et al. (Nov. 2015). 'Draft genome sequence of Herbinix hemicellulosilytica T3/55T, a new thermophilic cellulose degrading bacterium isolated from a thermophilic biogas reactor'. In: *Journal of biotechnology* 214, pp. 59–60.

– (July 2016a). 'Complete Genome Sequence of Herbinix luporum SD1D, a New Cellulose-Degrading Bacterium Isolated from a Thermophilic Biogas Reactor.' In: *Genome Announcements* 4.4, e00687–16.

– (June 2016b). 'Draft Genome Sequence of Propionispora sp. Strain 2/2-37, a New Xylan-Degrading Bacterium Isolated from a Mesophilic Biogas Reactor.' In: *Genome Announcements* 4.3, e00609–16.

Koeck, D. E., D. Wibberg, I. Maus, A. Winkler, A. Albersmeier, V. V. Zverlov, W. Liebl et al. (Oct. 2014). 'Complete genome sequence of the cellulolytic thermophile Ruminoclostridium cellulosi wild-type strain DG5 isolated from a thermophilic biogas plant'. In: *Journal of biotechnology* 188, pp. 136–137.

Koeck, D. E., D. Wibberg, I. Maus, A. Winkler, A. Albersmeier, V. V. Zverlov, A. Pühler et al. (Dec. 2014). 'First draft genome sequence of the amylolytic Bacillus thermoamylovorans wild-type strain 1A1 isolated from a thermophilic biogas plant'. In: *Journal of biotechnology* 192, pp. 154–155.

Koeppel, A. F. and M. Wu (May 2013). 'Surprisingly extensive mixed phylogenetic and ecological signals among bacterial Operational Taxonomic Units'. In: *Nucleic acids research* 41.10, pp. 5175–5188.

Kohavi, R. (1995). 'A study of cross-validation and bootstrap for accuracy estimation and model selection'. In: *International Joint Conference on Artificial Intelligence*, pp. 1137–1145.

Kõljalg, U., R. H. Nilsson, K. Abarenkov et al. (Nov. 2013). 'Towards a unified paradigm for sequence-based identification of fungi.' In: *Molecular ecology* 22.21, pp. 5271–5277.

Kopf, A., M. Bicak, R. Kottmann et al. (June 2015). 'The ocean sampling day consortium'. In: *GigaScience* 4.1.

Kyrpides, N. C. (Sept. 1999). 'Genomes OnLine Database (GOLD 1.0): a monitor of complete and ongoing genome projects world-wide.' In: *Bioinformatics* 15.9, pp. 773–774.

Kyrpides, N. C., P. Hugenholtz, J. A. Eisen et al. (May 2014). 'Genomic Encyclopedia of Bacteria and Archaea: Sequencing a Myriad of Type Strains'. In: *PLoS biology* 12.8, e1001920.

Kyrpides, N. C., T. Woyke, J. A. Eisen et al. (June 2014). 'Genomic Encyclopedia of Type Strains, Phase I: The one thousand microbial genomes (KMG-I) project'. In: *Standards in Genomic Sciences* 9.3, pp. 1278–1284.

Lagier, J.-C., S. Edouard, I. Pagnier et al. (Jan. 2015). 'Current and past strategies for bacterial culture in clinical microbiology.' In: *Clinical microbiology reviews* 28.1, pp. 208–236.

Lagier, J.-C., P. Hugon, S. Khelaifia et al. (Jan. 2015). 'The rebirth of culture in microbiology through the example of culturomics to study human gut microbiota.' In: *Clinical microbiology reviews* 28.1, pp. 237–264.

Laguerre, G., G. Depret, V. Bourion et al. (Nov. 2007). 'Rhizobium leguminosarum bv. viciae genotypes interact with pea plants in developmental responses of nodules, roots and shoots'. In: *New Phytologist* 176.3, pp. 680–690.

Laksanalamai, P. and F. T. Robb (2004). 'Small heat shock proteins from extremophiles: a review'. In: *Extremophiles: life under extreme conditions* 8.1, pp. 1–11.

Langille, M. G. I., J. Zaneveld, J. G. Caporaso et al. (Sept. 2013). 'Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences'. In: *Nature Biotechnology* 31.9, pp. 814–821.

Lebuhn, M., A. Hanreich, M. Klocke et al. (Oct. 2014). 'Towards molecular biomarkers for biogas production from lignocellulose-rich substrates'. In: *Anaerobe* 29, pp. 10–21.

Leinonen, R., R. Akhtar, E. Birney et al. (Jan. 2011). 'The European Nucleotide Archive.' In: *Nucleic acids research* 39.Database issue, pp. D28–31.

Leuken, J. P. G. van, A. N. Swart, P. Droogers et al. (2016). 'Climate change effects on airborne pathogenic bioaerosol concentrations: a scenario analysis.' In: *Aerobiologia* 32.4, pp. 607–617.

Li, D., C.-M. Liu, R. Luo et al. (May 2015). 'MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph'. In: *Bioinformatics* 31.10, pp. 1674–1676.

Li, D., R. Luo, C.-M. Liu et al. (June 2016). 'MEGAHIT v1.0: A fast and scalable metagenome assembler driven by advanced methodologies and community practices'. In: *Methods* 102, pp. 3–11.

Litchman, E., K. F. Edwards and C. A. Klausmeier (2015). 'Microbial resource utilization traits and trade-offs: implications for community structure, functioning, and biogeochemical impacts at present and in the future.' In: *Frontiers in microbiology* 6.

Liu, B. and M. Pop (Jan. 2009). 'ARDB—Antibiotic Resistance Genes Database'. In: *Nucleic acids research* 37.suppl1, pp. D443–D447.

Lozupone, C., K. Faust, J. Raes et al. (Oct. 2012). 'Identifying genomic and metabolic features that can underlie early successional and opportunistic lifestyles of human gut symbionts.' In: *Genome research* 22.10, pp. 1974–1984.

Luo, F., C. E. Devine and E. A. Edwards (Sept. 2016). 'Cultivating microbial dark matter in benzene-degrading methanogenic consortia.' In: *Environmental microbiology* 18.9, pp. 2923–2936.

Lupan, I., M. B. Ianc, C. Ochis et al. (2013). 'The evidence of contaminant bacterial DNA in several commercial Taq polymerases'. In: *Romanian Biotechnological Letter* 18.1.

Lux, M., J. Krüger, C. Rinke et al. (Dec. 2016). 'acdc – Automated Contamination Detection and Confidence estimation for single-cell genome data'. In: *BMC bioinformatics* 17.1.

Macnab, R. M. (1996). 'Flagella and motility'. In: *Escherichia Coli and Salmonella.* Ed. by F. C. Neidhardt and R. Curtis. ASM Press, pp. 123–145.

Madigan, M. T., J. M. Martinko, K. Bender et al. (2014). *Brock Biology of Microorganisms 13th edition.* Global Edition, 14th edition. Pearson.

Madison, B. M. (July 2009). 'Application of stains in clinical microbiology'. In: *Biotechnic & Histochemistry* 76.3, pp. 119–125.

Magoč, T. and S. L. Salzberg (Nov. 2011). 'FLASH: fast length adjustment of short reads to improve genome assemblies'. In: *Bioinformatics* 27.21, pp. 2957–2963.

Al-Mailem, D. M., M. Al-Deieg, M. Eliyas et al. (May 2017). 'Biostimulation of indigenous microorganisms for bioremediation of oily hypersaline microcosms from the Arabian Gulf Kuwaiti coasts.' In: *Journal of environmental management* 193, pp. 576–583.

Manyi-Loh, C., S. Mamphweli, E. Meyer et al. (Sept. 2013). 'Microbial Anaerobic Digestion (Bio-Digesters) as an Approach to the Decontamination of Animal Wastes in Pollution Control and the Generation of Renewable Energy'. In: *International Journal of Environmental Research and Public Health* 10.9, pp. 4390–4417.

Mardia, K. V., J. T. Kent and J. M. Bibby (1980). *Multivariate analysis.* 1st ed. Academic Press.

Margesin, R. and F. Schinner (2001). 'Potential of halotolerant and halophilic microorganisms for biotechnology'. In: *Extremophiles: life under extreme conditions* 5.2, pp. 73–83.

Margulies, M., M. Egholm, W. E. Altman et al. (Sept. 2005). 'Genome sequencing in microfabricated high-density picolitre reactors.' In: *Nature* 437.7057, pp. 376–380.

Markowitz, V. M., I.-M. A. Chen, K. Chu et al. (Jan. 2014). 'IMG/M 4 version of the integrated metagenome comparative analysis system'. In: *Nucleic acids research* 42.D1, pp. D568–D573.

Matsen, F. A., R. B. Kodner and E. V. Armbrust (Oct. 2010). 'pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree'. In: *BMC bioinformatics* 11.1.

Matturro, B., C. Ubaldi, P. Grenni et al. (July 2016). 'Polychlorinated biphenyl (PCB) anaerobic degradation in marine sediments: microcosm study and role of autochthonous microbial communities.' In: *Environmental science and pollution research international* 23.13, pp. 12613–12623.

Maus, I., K. G. Cibis, D. Wibberg et al. (June 2015). 'Complete genome sequence of the strain Defluviitoga tunisiensis L3, isolated from a thermophilic, production-scale biogas plant'. In: *Journal of biotechnology* 203, pp. 17–18.

Maus, I., D. E. Koeck, K. G. Cibis et al. (2016). 'Unraveling the microbiome of a thermophilic biogas plant by metagenome and metatranscriptome analysis complemented by characterization of bacterial and archaeal isolates.' In: *Biotechnology for biofuels* 9.1, p. 171.

Maus, I., R. Stantscheff, D. Wibberg et al. (Dec. 2014). 'Complete genome sequence of the methanogenic neotype strain Methanobacterium formicicum MFT'. In: *Journal of biotechnology* 192, pp. 40–41.

Maus, I., D. Wibberg, R. Stantscheff, K. Cibis et al. (Dec. 2013). 'Complete genome sequence of the hydrogenotrophic Archaeon Methanobacterium sp. Mb1 isolated from a production-scale biogas plant'. In: *Journal of biotechnology* 168.4, pp. 734–736.

Maus, I., D. Wibberg, R. Stantscheff, F.-G. Eikmeyer et al. (Oct. 2012). 'Complete genome sequence of the hydrogenotrophic, methanogenic archaeon Methanoculleus bourgensis strain MS2(T), Isolated from a sewage sludge digester.' In: *Journal of Bacteriology* 194.19, pp. 5487–5488.

McArthur, A. G., N. Waglechner, F. Nizam et al. (July 2013). 'The comprehensive antibiotic resistance database.' In: *Antimicrobial agents and chemotherapy* 57.7, pp. 3348–3357.

McBride, M. J. (2001). 'Bacterial gliding motility: multiple mechanisms for cell movement over surfaces.' In: *Annual review of microbiology* 55.1, pp. 49–75.

McDonald, D., J. C. Clemente, J. Kuczynski et al. (July 2012). 'The Biological Observation Matrix (BIOM) format or: how I learned to stop worrying and love the ome-ome.' In: *GigaScience* 1.7.

McHardy, A. C., H. G. Martin, A. Tsirigos et al. (Dec. 2006). 'Accurate phylogenetic classification of variable-length DNA fragments'. In: *Nature Methods* 4.1, pp. 63–72.

McLachlan, G., K.-A. Do and C. Ambroise (Aug. 2004). *Analyzing Microarray Gene Expression Data.* McLachlan/Microarray Gene Expression. John Wiley & Sons.

Mendes, L. W., E. E. Kuramae, A. A. Navarrete et al. (Aug. 2014). 'Taxonomical and functional microbial community selection in soybean rhizosphere'. In: *The ISME Journal* 8.8, pp. 1577–1587.

Merriam-Webster, Incorporated (2017). *Definition of METADATA*. URL: `https://www.merriam-webster.com/dictionary/metadata`.

Molinari, G. (2010). 'Natural Products in Drug Discovery: Present Status and Perspectives'. In: *Pharmaceutical Biotechnology*. Ed. by C. A. Guzmán and G. Z. Feuerstein. Springer Science & Business Media, pp. 13–27.

Mukherjee, S., M. Huntemann, N. Ivanova et al. (Mar. 2015). 'Large-scale contamination of microbial isolate genomes by Illumina PhiX control'. In: *Standards in Genomic Sciences* 10.18.

Mukherjee, S., D. Stamatis, J. Bertsch et al. (Jan. 2017). 'Genomes OnLine Database (GOLD) v.6: data updates and feature enhancements.' In: *Nucleic acids research* 45.D1, pp. D446–D456.

Mulcahy-O'Grady, H. and M. L. Workentine (2016). 'The Challenge and Potential of Metagenomics in the Clinic'. In: *Frontiers in immunology* 7.Suppl 2.

Nandy, S. K., P. M. Bapat and K. V. Venkatesh (Jan. 2007). 'Sporulating bacteria prefers predation to cannibalism in mixed cultures.' In: *FEBS letters* 581.1, pp. 151–156.

National Research Council US Committee (2007). 'The New Science of Metagenomics: Revealing the Secrets of Our Microbial Planet'. In: *National Research Council US Committee on Metagenomics Challenges and Functional Applications.*

NCBI Resource Coordinators (Jan. 2017). 'Database Resources of the National Center for Biotechnology Information.' In: *Nucleic acids research* 45.D1, pp. D12–D17.

Niu, B., Z. Zhu, L. Fu et al. (June 2011). 'FR-HIT, a very fast program to recruit metagenomic reads to homologous reference genomes'. In: *Bioinformatics* 27.12, pp. 1704–1705.

Nunoura, T., H. Oida, M. Miyazaki et al. (Mar. 2008). 'Thermosulfidibacter takaii gen. nov., sp. nov., a thermophilic, hydrogen-oxidizing, sulfur-reducing chemolithoautotroph isolated from a deep-sea hydrothermal field in the Southern Okinawa Trough.' In: *International Journal of Systematic and Evolutionary Microbiology* 58.Pt 3, pp. 659–665.

Olson, D. L. and D. Delen (Jan. 2008). *Advanced Data Mining Techniques*. Springer.

Oren, A. (2002). 'Diversity of halophilic microorganisms: Environments, phylogeny, physiology, and applications'. In: *Journal of Industrial Microbiology and Biotechnology* 28.1, pp. 56–63.

Ortseifen, V., Y. Stolze, I. Maus et al. (Aug. 2016). 'An integrated metagenome and -proteome analysis of the microbial community residing in a biogas production plant'. In: *Journal of biotechnology* 231, pp. 268–279.

Osorio-Lozada, A., S. Surapaneni, G. L. Skiles et al. (Feb. 2008). 'Biosynthesis of Drug Metabolites Using Microbes in Hollow Fiber Cartridge Reactors: Case Study of Diclofenac Metabolism by Actinoplanes Species'. In: *Drug Metabolism and Disposition* 36.2, pp. 234–240.

Osterholz, B., P. Wiebke, A. Fust et al. (2015). 'A Bioinformatics Pipeline for the Detection of β-lactamase Genes in Metagenome Sequence Data and its Application to Production-Scale Biogas Plants'. In: *rd International Symposium on the environmental Dimension of Antibiotic Resistance.*

Palop, A., P. Manas and S. Condon (Apr. 1999). 'Sporulation temperature and heat resistance of Bacillus spores: a review '. In: *Journal of Food Safety* 19.1, pp. 57–72.

Pandey, A., P. Singh and L. Iyengar (Mar. 2007). 'Bacterial decolorization and degradation of azo dyes'. In: *International Biodeterioration & Biodegradation* 59.2, pp. 73–84.

Parisutham, V., T. H. Kim and S. K. Lee (June 2014). 'Feasibilities of consolidated bioprocessing microbes: From pretreatment to biofuel production'. In: *Bioresource technology* 161, pp. 431–440.

Parks, D. H., M. Imelfort, C. T. Skennerton et al. (July 2015). ‘CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes.’ In: *Genome research* 25.7, pp. 1043–1055.

Patil, V. U., V. Girimalla, V. Sagar et al. (June 2017). ‘Genome sequencing of four strains of Phylotype I, II and IV of Ralstonia solanacearum that cause potato bacterial wilt in India.’ In: *Brazilian journal of microbiology* 48.2, pp. 193–195.

Peason, K. (1901). ‘On lines and planes of closest fit to systems of point in space’. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11, pp. 559–572.

Pedregosa, F., G. Varoquaux, A. Gramfort et al. (2011). ‘Scikit-learn: Machine Learning in Python’. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Peng, J., K. Bai, X. Shang et al. (Jan. 2017). ‘Predicting disease-related genes using integrated biomedical networks.’ In: *BMC genomics* 17.1043.

Peruani, F., J. Starruß, V. Jakovljevic et al. (Feb. 2012). ‘Collective Motion and Nonequilibrium Cluster Formation in Colonies of Gliding Bacteria’. In: *Physical Review Letters* 108.9.

Pii, Y., L. Borruso, L. Brusetti et al. (Feb. 2016). ‘The interaction between iron nutrition, plant species and soil type shapes the rhizosphere microbiome’. In: *Plant Physiology and Biochemistry* 99, pp. 39–48.

Pomerantz, J. (Nov. 2015). *Metadata*. MIT Press.

Poole, K. (Apr. 2001). ‘Multidrug efflux pumps and antimicrobial resistance in Pseudomonas aeruginosa and related organisms.’ In: *Journal of molecular microbiology and biotechnology* 3.2, pp. 255–264.

Portillo, A., L. Pérez-Martínez, S. Santibáñez et al. (Aug. 2007). ‘Detection of Rickettsia africae in Rhipicephalus (Boophilus) decoloratus ticks from the Republic of Botswana, South Africa.’ In: *The American journal of tropical medicine and hygiene* 77.2, pp. 376–377.

Poulsen, H. V., F. W. Willink and K. Ingvorsen (Oct. 2016). ‘Aerobic and anaerobic cellulase production by Cellulomonas uda.’ In: *Archives of microbiology* 198.8, pp. 725–735.

Prasad, R. (2014). ‘New approaches and insights into bioremediation of hazardous waste.’ In: *Reviews on environmental health*, pp. 33–35.

Qin, J., R. Li, J. Raes et al. (Mar. 2010). ‘A human gut microbial gene catalogue established by metagenomic sequencing.’ In: *Nature* 464.7285, pp. 59–65.

Quast, C., E. Pruesse, P. Yilmaz et al. (Jan. 2013). ‘The SILVA ribosomal RNA gene database project: improved data processing and web-based tools.’ In: *Nucleic acids research* 41.Database issue, pp. D590–D596.

Rao, R. B., G. Fung and R. Rosales (Dec. 2013). ‘On the Dangers of Cross-Validation. An Experimental Evaluation’. In: *Proceedings of the 2008 SIAM International Conference on Data Mining*. Philadelphia, PA: Society for Industrial and Applied Mathematics, pp. 588–596.

Rinke, C., J. Lee, N. Nath et al. (May 2014). ‘Obtaining genomes from uncultivated environmental microorganisms using FACS–based single-cell genomics’. In: *Nature protocols* 9.5, pp. 1038–1048.

Rinke, C., P. Schwientek, A. Sczyrba et al. (July 2013). ‘Insights into the phylogeny and coding potential of microbial dark matter.’ In: *Nature* 499.7459, pp. 431–437.

Rondon, M. R., P. R. August, A. D. Bettermann et al. (June 2000). ‘Cloning the soil metagenome: a strategy for accessing the genetic and functional diversity of uncultured microorganisms.’ In: *Applied and Environmental Microbiology* 66.6, pp. 2541–2547.

Rossi, O., P. van Baarlen and J. M. Wells (2013). ‘Host-recognition of pathogens and commensals in the mammalian intestine.’ In: *Between Pathogenicity and Commensalism*. Ed. by U. Dobrindt, J. H. Hacker and C. Svanborg. Springer Berlin Heidelberg, pp. 291–321.

Round, J. L. and S. K. Mazmanian (May 2009). ‘The gut microbiota shapes intestinal immune responses during health and disease’. In: *Nature Reviews Immunology* 9.5, pp. 313–323.

Rusch, D. B., A. L. Halpern, G. Sutton et al. (2007). 'The Sorcerer II Global Ocean Sampling Expedition: Northwest Atlantic through Eastern Tropical Pacific'. In: *PLoS biology* 5.3, e77.

Russell, S. J. and P. Norvig (2010). *Artificial Intelligence.* A Modern Approach. Prentice Hall.

Samuel, A. L. (1959). 'Some Studies in Machine Learning Using the Game of Checkers'. In: *IBM Journal of research and development* 3.3, pp. 210–229.

Sander, J., M. Ester, H.-P. Kriegel et al. (1998). 'Density-based clustering in spatial databases: The algorithm gdbscan and its applications'. In: *Data mining and knowledge discovery* 2.2, pp. 169–194.

Sanger, F. and A. R. Coulson (May 1975). 'A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase.' In: *Journal of molecular biology* 94.3, pp. 441–448.

Sanger, F., S. Nicklen and A. R. Coulson (Dec. 1977). 'DNA sequencing with chain-terminating inhibitors.' In: *Proceedings of the National Academy of Sciences* 74.12, pp. 5463–5467.

Sarntivijai, S., D. Vasant, S. Jupp et al. (2016). 'Linking rare and common disease: mapping clinical disease-phenotypes to ontologies in therapeutic target validation.' In: *Journal of biomedical semantics* 7.8.

Schapire, R. E. (1990). 'The strength of weak learnability'. In: *Machine learning* 5.2, pp. 197–227.

Schloss, P. D., S. L. Westcott, T. Ryabin et al. (Dec. 2009). 'Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities.' In: *Applied and Environmental Microbiology* 75.23, pp. 7537–7541.

Schoch, C. L., K. A. Seifert, S. Huhndorf et al. (Apr. 2012). 'Nuclear ribosomal internal transcribed spacer (ITS) region as a universal DNA barcode marker for Fungi.' In: *Proceedings of the National Academy of Sciences of the United States of America* 109.16, pp. 6241–6246.

Schouten, S., E. C. Hopmans and J. S. Sinninghe Damsté (Jan. 2013). 'The organic geochemistry of glycerol dialkyl glycerol tetraether lipids: A review'. In: *Organic geochemistry* 54, pp. 19–61.

Schriml, L. M., C. Arze, S. Nadendla et al. (Jan. 2012). 'Disease Ontology: a backbone for disease semantic integration.' In: *Nucleic acids research* 40.D1, pp. D940–D946.

Selle, P. H. and V. Ravindran (May 2007). 'Microbial phytase in poultry nutrition'. In: *Animal Feed Science and Technology* 135.1-2, pp. 1–41.

Seni, G. and J. F. Elder (Feb. 2010). ' Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions'. In: *Synthesis Lectures on Data Mining and Knowledge Discovery* 2.1, pp. 1–126.

Shuib, F. N. S., A. Husaini, A. Zulkharnain et al. (2016). 'Optimization of Physiochemical Parameters during Bioremediation of Synthetic Dye by Marasmius cladophyllus UMAS MS8 Using Statistical Approach.' In: *TheScientificWorldJournal* 2016.8296239.

Siliakus, M. F., J. van der Oost and S. W. M. Kengen (May 2017). 'Adaptations of archaeal and bacterial membranes to variations in temperature, pH and pressure.' In: *Extremophiles: life under extreme conditions* 65, pp. 1–20.

Simpson, E. H. (1949). 'Measurement of diversity.' In: *Nature* 163, p. 688.

Singh, S. B., K. Young and L. L. Silver (June 2017). 'What is an "ideal" antibiotic? Discovery challenges and path forward.' In: *Biochemical pharmacology* 133, pp. 63–73.

Sneath, P. and R. R. Sokal (1973). *Numerical taxonomy.* The principles and practice of numerical classification. WH Freeman.

Sokal, R. R. and P. H. A. Sneath (1963). *Principles of Numerical Taxonomy.* W. H. Freeman.

Spratt, B. G. (Aug. 1975). 'Distinct penicillin binding proteins involved in the division, elongation, and shape of Escherichia coli K12.' In: *Proceedings of the National Academy of Sciences* 72.8, pp. 2999–3003.

Srinivasan, R., U. Karaoz, M. Volegova et al. (June 2015). 'Use of 16S rRNA Gene for Identification of a Broad Range of Clinically Relevant Bacterial Pathogens'. In: *PloS one* 10.2, e0117617.

Stehman, S. V. (Oct. 1997a). ‘Selecting and interpreting measures of thematic classification accuracy’. In: *Remote sensing of Environment* 62.1, pp. 77–89.

– (Oct. 1997b). ‘Selecting and interpreting measures of thematic classification accuracy’. In: *Remote sensing of Environment* 62.1, pp. 77–89.

Stempfhuber, B., T. Richter-Heitmann, K. M. Regan et al. (2015). ‘Spatial Interaction of Archaeal Ammonia-Oxidizers and Nitrite-Oxidizing Bacteria in an Unfertilized Grassland Soil’. In: *Frontiers in microbiology* 6, p. 1567.

Stephens, D. S., B. Greenwood and P. Brandtzaeg (June 2007). ‘Epidemic meningitis, meningococcaemia, and Neisseria meningitidis.’ In: *Lancet* 369.9580, pp. 2196–2210.

Stephens, Z. D., S. Y. Lee, F. Faghri et al. (July 2015). ‘Big Data: Astronomical or Genomical?’ In: *PLoS biology* 13.7, e1002195.

Stevens, H. (2013). *Life Out of Sequence: A Data-Driven History of Bioinformatics.* University of Chicago Press.

Stewart, F. J. and C. M. Cavanaugh (2006). ‘Symbiosis of thioautotrophic bacteria with Riftia pachyptila.’ In: *Progress in molecular and subcellular biology* 41, pp. 197–225.

Stock, J. B. and M. G. Surette (1996). *Chemotaxis.* Ed. by F. C. Neidhardt and R. Curtis. 2nd. Escherichia Coli and Salmonella: Cellular and Molecular Biology. ASM Press.

Stolze, Y., A. Bremges, M. Rumming et al. (2016). ‘Identification and genome reconstruction of abundant distinct taxa in microbiomes from one thermophilic and three mesophilic production-scale biogas plants.’ In: *Biotechnology for biofuels* 9.156.

Stuart, A. L. and D. A. Wilkening (Apr. 2005). ‘Degradation of biological weapons agents in the environment: implications for terrorism response.’ In: *Environmental science & technology* 39.8, pp. 2736–2743.

Tatusov, R. L., E. V. Koonin and D. J. Lipman (Oct. 1997). ‘A Genomic Perspective on Protein Families’. In: *Science* 278.5338, pp. 631–637.

Telenti, A., L. C. T. Pierce, W. H. Biggs et al. (Oct. 2016). ‘Deep sequencing of 10,000 human genomes.’ In: *Proceedings of the National Academy of Sciences of the United States of America* 113.42, pp. 11901–11906.

The PostgreSQL Global Development Group (Jan. 2016). *PostgreSQL 9.5.* URL: `https://www.postgresql.org/`.

Timmusk, S., I. A. Abd El-Daim, L. Copolovici et al. (2014). ‘Drought-tolerance of wheat improved by rhizosphere bacteria from harsh environments: enhanced biomass production and reduced emissions of stress volatiles.’ In: *PloS one* 9.5, e96086.

Tomazetto, G., S. Hahnke, D. E. Koeck et al. (Aug. 2016). ‘Complete genome analysis of Clostridium bornimense strain M2/40T: A new acidogenic Clostridium species isolated from a mesophilic two-phase laboratory-scale biogas reactor’. In: *Journal of biotechnology* 232, pp. 38–49.

Tomazetto, G., S. Hahnke, I. Maus et al. (Dec. 2014). ‘Complete genome sequence of Peptoniphilus sp. strain ING2-D1G isolated from a mesophilic lab-scale completely stirred tank reactor utilizing maize silage in co-digestion with pig and cattle manure for biomethanation’. In: *Journal of biotechnology* 192, pp. 59–61.

Toribio, A. L., B. Alako, C. Amid et al. (Jan. 2017). ‘European Nucleotide Archive in 2016.’ In: *Nucleic acids research* 45.D1, pp. D32–D36.

Ursu, O., J. Holmes, J. Knockel et al. (Jan. 2017). ‘DrugCentral: online drug compendium.’ In: *Nucleic acids research* 45.D1, pp. D932–D939.

Van de Wiele, T., J. T. Van Praet, M. Marzorati et al. (July 2016). ‘How the microbiota shapes rheumatic diseases’. In: *Nature Reviews Rheumatology* 12.7, pp. 398–411.

Vandamme, E. J., A. Cerdobbel and W. Soetaer (2005). ‘Biocatalysis on the rise: Part 1 Principles’. In: *Chimica oggi* 23.6.

VanMensel, D., S. R. Chaganti, R. Boudens et al. (Mar. 2017). 'Investigating the Microbial Degradation Potential in Oil Sands Fluid Fine Tailings Using Gamma Irradiation: A Metagenomic Perspective.' In: *Microbial ecology*, pp. 1–11.

Vapnik, V. N. and A. Y. Chervonenkis (1963). 'Pattern recognition using generalized portrait method'. In: *Automation and Remote Control* 24, pp. 774–780.

Venter, J. C., K. Remington, J. F. Heidelberg et al. (Apr. 2004). 'Environmental Genome Shotgun Sequencing of the Sargasso Sea'. In: *Science* 304.5667, pp. 66–74.

Ventosa, A., J. J. Nieto and A. Oren (June 1998). 'Biology of moderately halophilic aerobic bacteria.' In: *Microbiology and Molecular Biology Reviews* 62.2, pp. 504–544.

Vogel, T. M., P. Simonet, J. K. Jansson et al. (2009). 'TerraGenome: a consortium for the sequencing of a soil metagenome'. In: *Nature Reviews Microbiology* 7.252, p. 252.

Vogtmann, E., X. Hua, G. Zeller et al. (Dec. 2016). 'Colorectal Cancer and the Human Gut Microbiome: Reproducibility with Whole-Genome Shotgun Sequencing'. In: *PloS one* 11.5, e0155362.

Walls, R. L., B. Smith, J. Elser et al. (2012). 'A plant disease extension of the Infectious Disease Ontology.' In: *International Conference on Biomedical Ontology*, pp. 1–5.

Wang, Y., J. K. Hatt, D. Tsementzi et al. (Apr. 2017). 'Quantifying the Importance of the Rare Biosphere for Microbial Community Response to Organic Pollutants in a Freshwater Ecosystem.' In: *Applied and Environmental Microbiology* 83.8.

Wdowiak-Wróbel, S., M. Marek-Kozaczuk, M. Kalita et al. (May 2017). 'Diversity and plant growth promoting properties of rhizobia isolated from root nodules of Ononis arvensis.' In: *Antonie van Leeuwenhoek*, pp. 1–17.

Weimann, A., K. Mooren, J. Frank et al. (Dec. 2016). 'From Genomes to Phenotypes: Traitar, the Microbial Trait Analyzer'. In: *mSystems* 1.6, e00101–16.

Whittaker, R. H. (Feb. 1960). 'Vegetation of the Siskiyou Mountains, Oregon and California'. In: *Ecological monographs* 30.3, pp. 279–338.

Whittaker, R. J., K. J. Willis and R. Field (Apr. 2001). 'Scale and species richness: towards a general, hierarchical theory of species diversity'. In: *Journal of Biogeography* 28.4, pp. 453–470.

Williamson, S. J., D. B. Rusch, S. Yooseph et al. (2008). 'The Sorcerer II Global Ocean Sampling Expedition: Metagenomic Characterization of Viruses within Aquatic Microbial Samples'. In: *PloS one* 3.1, e1456.

Wood, J. M. (May 2015). 'Bacterial responses to osmotic challenges'. In: *The Journal of general physiology* 145.5, pp. 381–388.

Wooley, J. C., A. Godzik and I. Friedberg (Feb. 2010). 'A primer on metagenomics.' In: *PLoS computational biology* 6.2, e1000667.

World Wide Web Consortium (2000). *XHTML 1.0 The Extensible HyperText Markup Language.* URL: `https://www.w3.org/TR/1999/WD-xhtml1-19991124/`.

Woyke, T. and J. Jarett (Jan. 2015). 'Function-driven single-cell genomics'. In: *Microbial biotechnology* 8.1, pp. 38–39.

Wright, M. H., S. M. Farooqui, A. R. White et al. (Sept. 2016). 'Production of Manganese Oxide Nanoparticles by Shewanella Species.' In: *Applied and Environmental Microbiology* 82.17, pp. 5402–5409.

Xu, Y., Y. Nogi, C. Kato et al. (Mar. 2003). 'Moritella profunda sp. nov. and Moritella abyssi sp. nov., two psychropiezophilic organisms isolated from deep Atlantic sediments'. In: *International Journal of Systematic and Evolutionary Microbiology* 53.2, pp. 533–538.

Yang, B., Q. Wang, X. Zhao et al. (Mar. 2017). 'Increased energy production from sucrose by controlled hydrogen-producing fermentation followed by methanogenic fermentation'. In: *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* 39.6, pp. 576–582.

Yang, L. (2011). 'Classifiers selection for ensemble learning based on accuracy and diversity'. In: *Procedia Engineering* 15, pp. 4266–4270.

Yarza, P., P. Yilmaz, E. Pruesse et al. (Sept. 2014). ‘Uniting the classification of cultured and uncultured bacteria and archaea using 16S rRNA gene sequences’. In: *Nature reviews. Microbiology* 12.9, pp. 635–645.

Yergeau, E., C. Michel, J. Tremblay et al. (Feb. 2017). ‘Metagenomic survey of the taxonomic and functional microbial communities of seawater and sea ice from the Canadian Arctic.’ In: *Scientific reports* 7, p. 42242.

Yilmaz, P., R. Kottmann, D. Field et al. (May 2011). ‘Minimum information about a marker gene sequence (MIMARKS) and minimum information about any (x) sequence (MIxS) specifications’. In: *Nature Biotechnology* 29.5, pp. 415–420.

Yilmaz, P., L. W. Parfrey, P. Yarza et al. (Jan. 2014). ‘The SILVA and “All-species Living Tree Project (LTP)” taxonomic frameworks’. In: *Nucleic acids research* 42.D1, pp. D643–D648.

Yin, Y., X. Mao, J. Yang et al. (July 2012). ‘dbCAN: a web resource for automated carbohydrate-active enzyme annotation.’ In: *Nucleic acids research* 40.W1, W445–W451.

Zhang, H. (Mar. 2005). ‘Exploring conditions for the optimality of naive Bayes’. In: *International Journal of Pattern Recognition and Artificial Intelligence* 19.02, pp. 183–198.

Zhang, R., Z. Cheng, J. Guan et al. (Mar. 2015). ‘Exploiting topic modeling to boost metagenomic reads binning’. In: *BMC bioinformatics* 16.Suppl5:S2.

Zhu, J., H. Zou, S. Rosset et al. (2009). ‘Multi-class adaboost’. In: *Statistics and its Interface*, pp. 349–360.

List of Figures

List of Tables

Appendix

Summary of prediction performances

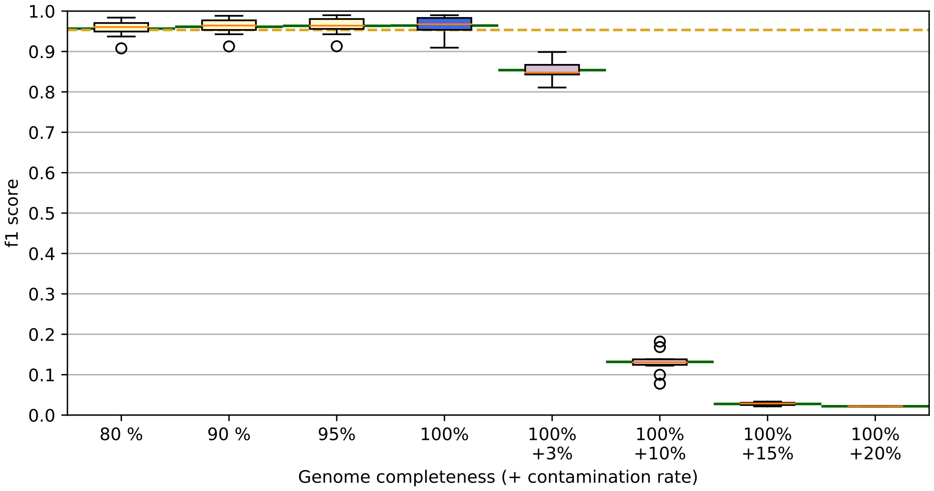


Figure .1: **Biotic Relationships** – Solid green line: median value of 10-fold cross validation measurements; Dotted yellow line: f1 score of validation data set.

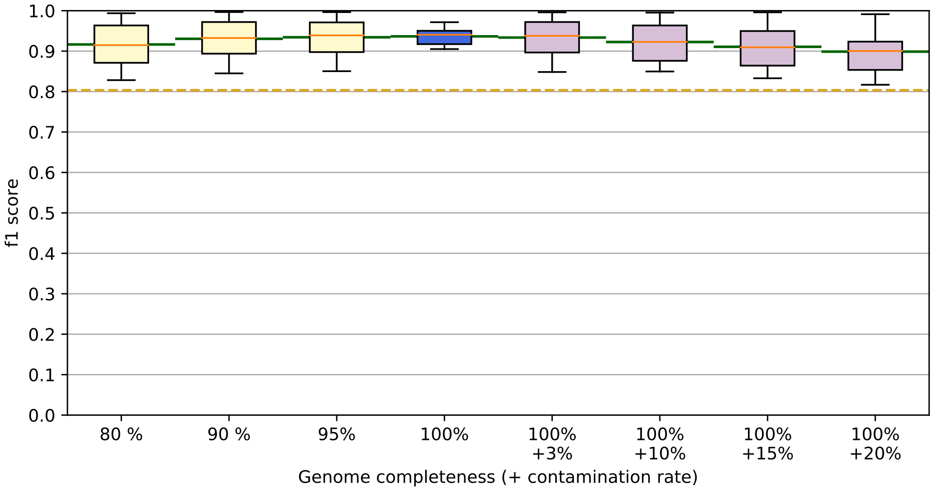


Figure .2: **Cell Arrangement** – Solid green line: median value of 10-fold cross validation measurements; Dotted yellow line: f1 score of validation data set.

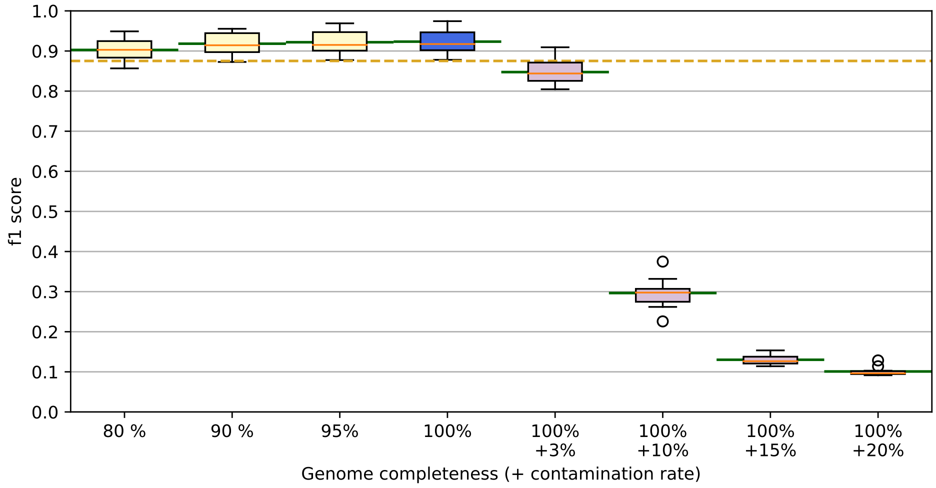


Figure .3: **Cell Shape** – Solid green line: median value of 10-fold cross validation measurements; Dotted yellow line: f1 score of validation data set.

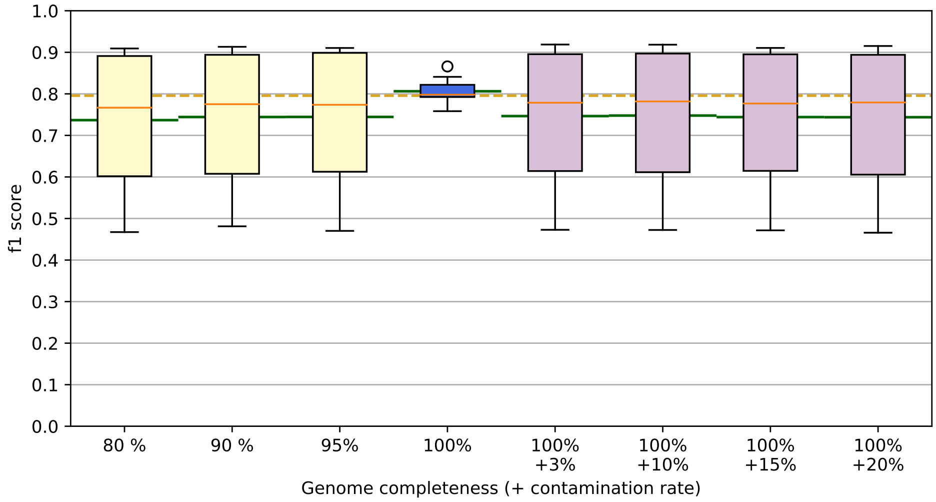


Figure .4: **Diseases** – Solid green line: median value of 10-fold cross validation measurements; Dotted yellow line: f1 score of validation data set.

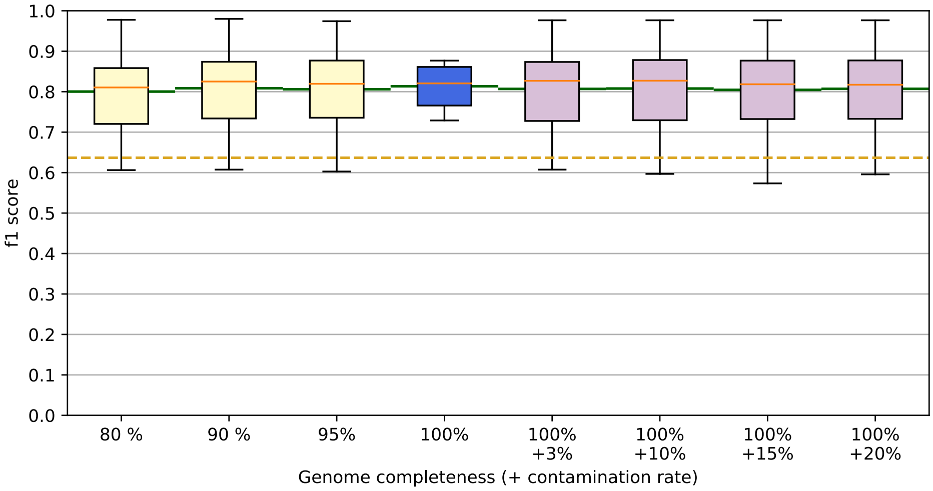


Figure .5: **Energy Source** – Solid green line: median value of 10-fold cross validation measurements; Dotted yellow line: f1 score of validation data set.

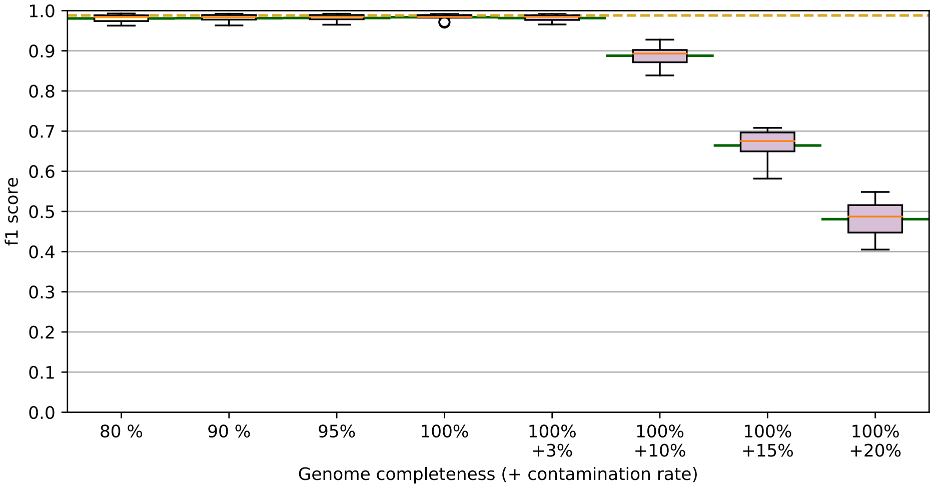


Figure .6: **Gram Staining** – Solid green line: median value of 10-fold cross validation measurements; Dotted yellow line: f1 score of validation data set.

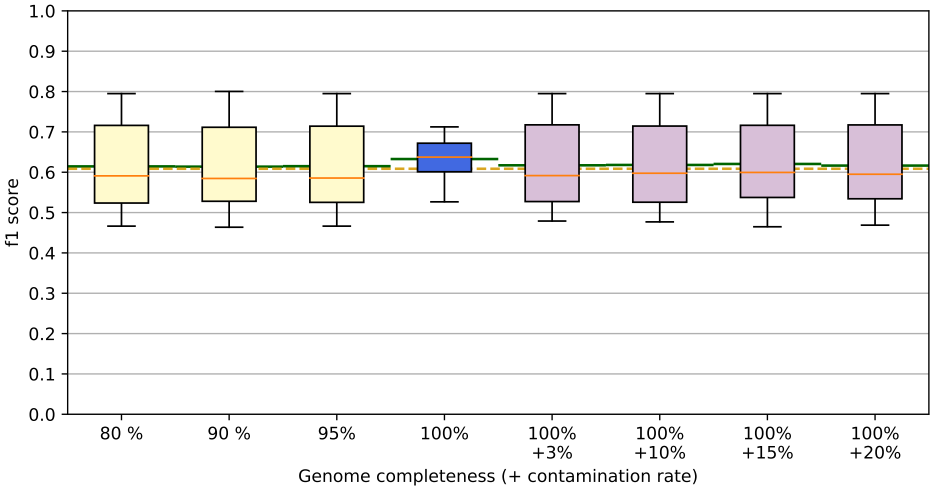


Figure .7: **Metabolism** – Solid green line: median value of 10-fold cross validation measurements; Dotted yellow line: f1 score of validation data set.

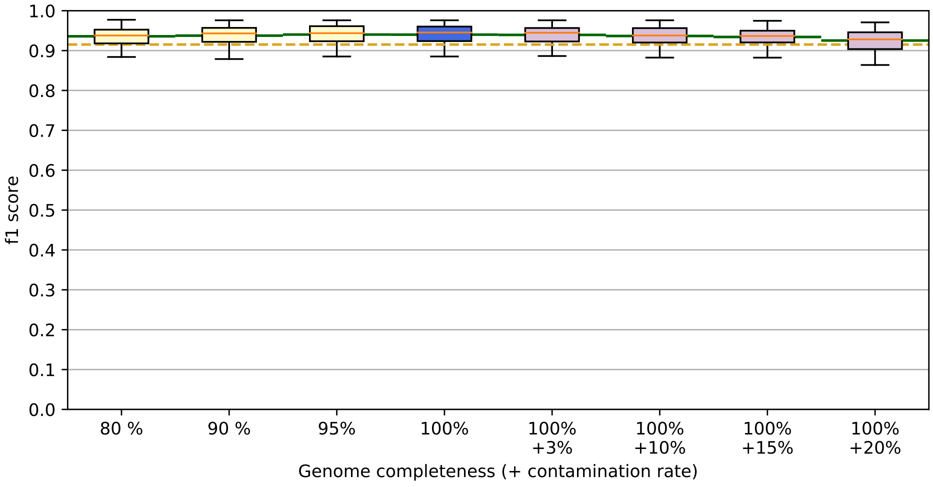


Figure .8: **Motility** – Solid green line: median value of 10-fold cross validation measurements; Dotted yellow line: f1 score of validation data set.

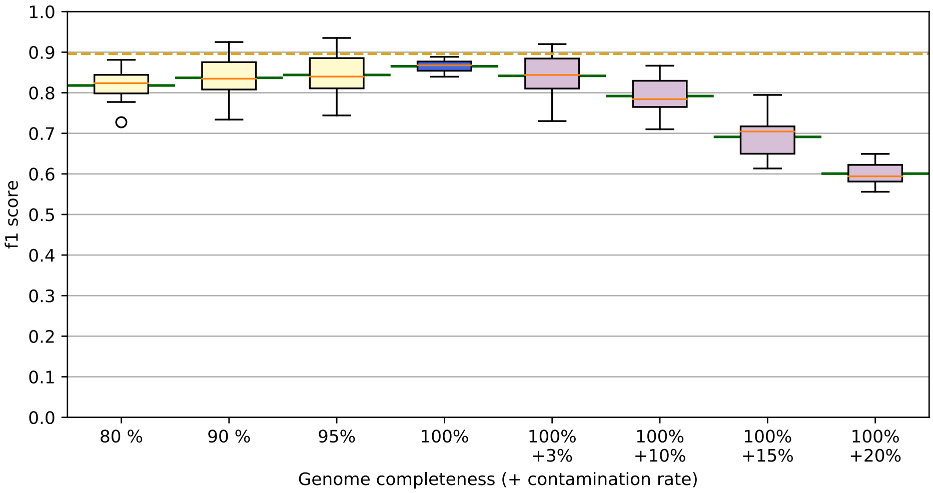


Figure .9: **Oxygen Requirement** – Solid green line: median value of 10-fold cross validation measurements; Dotted yellow line: f1 score of validation data set.

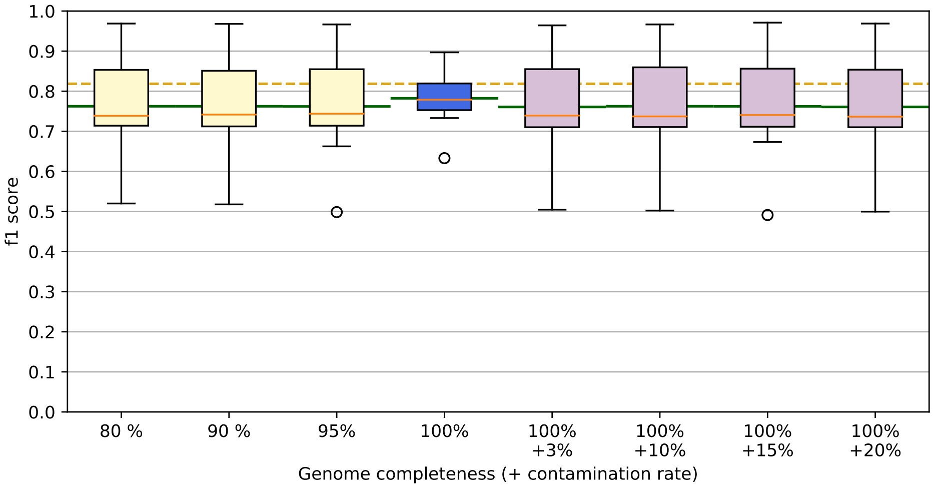


Figure .10: **Phenotype** – Solid green line: median value of 10-fold cross validation measurements; Dotted yellow line: f1 score of validation data set.

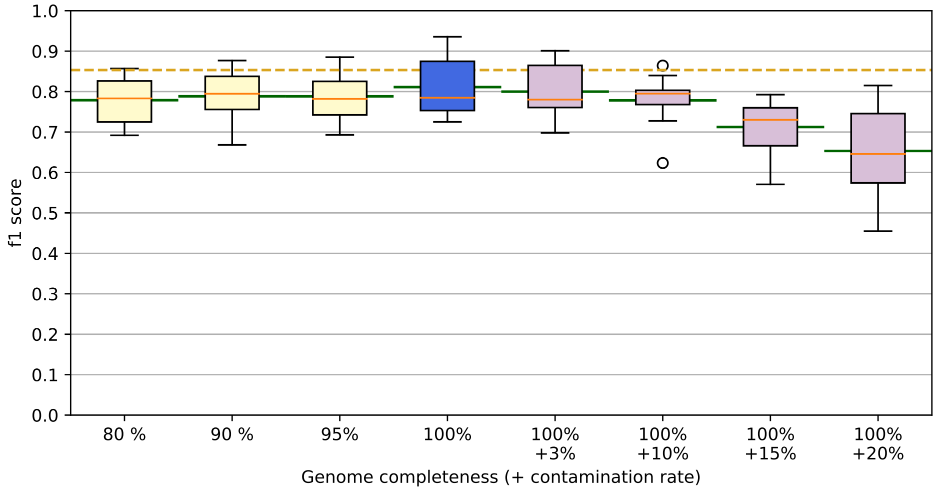


Figure .11: **Salinty** – Solid green line: median value of 10-fold cross validation measurements; Dotted yellow line: f1 score of validation data set.

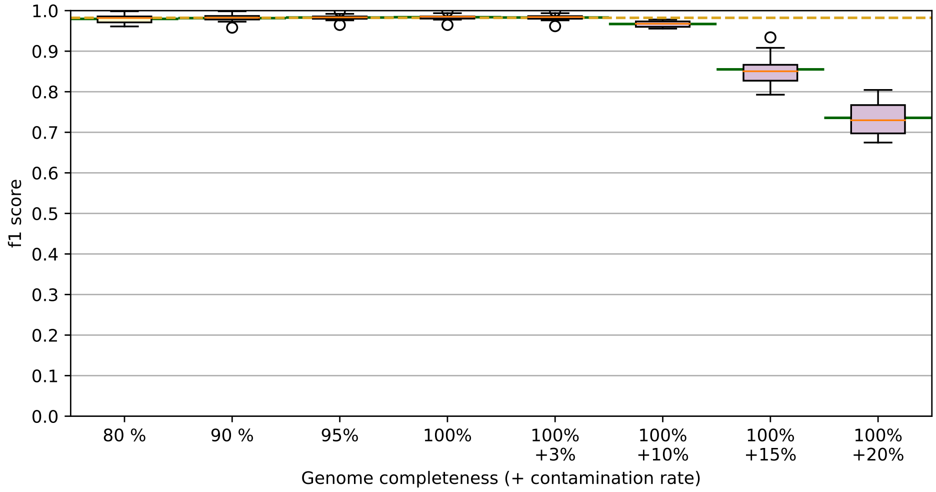


Figure .12: **Sporulation** – Solid green line: median value of 10-fold cross validation measurements; Dotted yellow line: f1 score of validation data set.

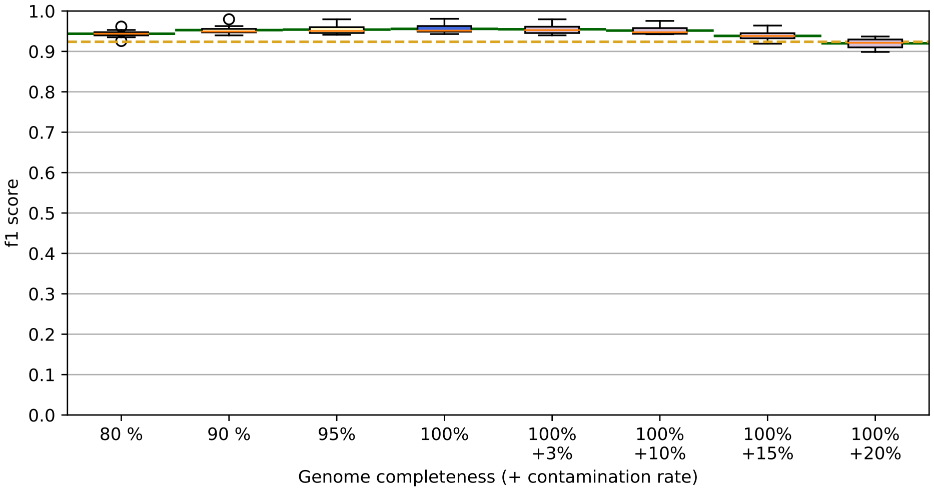


Figure .13: **Temperature Range** – Solid green line: median value of 10-fold cross validation measurements; Dotted yellow line: f1 score of validation data set.

Evaluated settings

C-Support Vector Classifier Default settings as defined at `http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html` Altered settings:

kernel linear, poly, sigmoid, rbf
C 0.5, 0.9, 0.95, 1.0, 1.05, 1.5

AdaBoost Default settings as defined at `http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html` Altered settings:

estimators 10, 50, 100, 200, 400, 600

Decision Tree Default settings as defined at `http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html` Altered settings:

splitter random, best
criterion gini, entropy
presort true, false
label description
max features none, auto, sqrt, log2

Gradient Boosting Default settings as defined at `http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html` Altered settings:

estimators 10, 50, 100, 200, 600
loss deviance, exponential
max depth 1, 3, 5, 7, 20, 50, 100
max features none, auto, sqrt, log2

k-Neighbors Default settings as defined at `http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html` Altered settings:

p 1, 2
weights uniform, distance
algorithm 1, 3, 5, 7, 20, 50, 100
neighbors 2, 5, 10, 12, 20

Naïve Bayes Default settings as defined at `http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html` and `http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html`

Random Forest Default settings as defined at `http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html` Altered settings:

estimators 1, 2, 5, 10, 20, 35, 50, 100, 200, 400, 600
criterion entropy, gini