MT-MAG: Accurate and interpretable machine learning for complete or partial taxonomic assignments of metagenome-assembled genomes

by

Wanxin Li

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2022

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The main contribution of this thesis consists of the article "MT-MAG: Accurate and interpretable machine learning for complete or partial taxonomic assignments of metagenome-assembled genomes" [23], which was submitted to the journal *PLOS ONE*. The author contributions are listed below.

Conceptualization: W.Li, L.Kari, Y.Yu, L.A.Hug

Data curation: W.Li

Formal analysis: W.Li, L.Kari, Y.Yu, L.A.Hug

Funding acquisition: L.Kari, Y.Yu

Investigation: W.Li, L.Kari, Y.Yu, L.A.Hug

Methodology: W.Li, L.Kari, Y.Yu, L.A.Hug

Project administration: L.Kari

Resources: L.Kari, Y.Yu, L.A.Hug

Software: W.Li

Supervision: L.Kari, Y.Yu

Validation: W.Li

Visualization: W.Li

Writing - original draft: W.Li

Writing - review & editing: W.Li, L.Kari, Y.Yu, L.A.Hug

Abstract

We propose MT-MAG, a novel machine learning-based software tool for the complete or partial hierarchically-structured taxonomic classification of metagenome-assembled genomes (MAGs). MT-MAG is capable of classifying large and diverse metagenomic datasets: a total of 245.68 Gbp in the training sets, and 9.6 Gbp in the test sets analyzed in this study. MT-MAG is, to the best of our knowledge, the first machine learning method for taxonomic assignment of metagenomic data that offers a "partial classification" option, whereby a classification at a higher taxonomic level is provided for MAGs that cannot be classified to the Species level. MT-MAG outputs complete or partial classification paths, and interpretable numerical classification confidences of its classifications, at all taxonomic ranks. To assess the performance of MT-MAG, we define a "weighted classification accuracy," with a weighting scheme reflecting the fact that partial classifications at different ranks are not equally informative. For the two benchmarking datasets analyzed (genomes from human gut microbiome species, and bacterial and archaeal genomes assembled from cow rumen metagenomic sequences), MT-MAG achieves an average of 80.13% in weighted classification accuracy. At the Species level, MT-MAG outperforms DeepMicrobes, the only other comparable software tool, by an average of 35.75% in weighted classification accuracy. In addition, MT-MAG is able to completely classify an average of 67.7% of the sequences at the Species level, compared with DeepMicrobes which only classifies 47.45%. Moreover, MT-MAG provides additional information for sequences that it could not classify at the Species level, resulting in the partial or complete classification of 95.15%, of the genomes in the datasets analyzed. Lastly, unlike other taxonomic assignment tools (e.g., GDTB-Tk), MT-MAG is an alignment-free and genetic marker-free tool, able to provide additional bioinformatics analysis to confirm existing or tentative taxonomic assignments.

Acknowledgements

The completion of this thesis left me with many beautiful memories of University of Waterloo. Coming to Waterloo through pure serendipity, I could never imagine how wonderful this journey would be with the help and support from many individuals.

I would like to express my heartfelt thanks to my supervisors Dr. Lila Kari and Dr. Yaoliang Yu. I am grateful for their inspirational ideas, timely encouragement and tremendous patience over the past two years. I feel extremely lucky to have both Lila and Yaoliang as my supervisors who have always been supportive of my research and truly cared about me as a person. Also, I would like to offer my sincere thanks to Dr. Laura Hug for her insightful inputs throughout this project. As a knowledgeable microbiologist, Laura could always clarify my questions about metagenomics using language that I could understand. I learned a lot from Lila, Yaoliang, and Laura about how to conduct research, and more importantly, how to become an interesting person! In addition, I would like to thank Dr. Bin Ma for his constructive suggestions for my thesis.

I would like to thank Dr. Reza Ramezan, Dr. Anisoara Nica, Dr. Grant Weddell and Dr. Yu-Ru Liu for supervising me as a research assistant or research intern when I was an undergraduate. It is truly fortunate to be exposed to different research topics as an undergraduate. In addition, I am thankful for their encouragement for me to pursue graduate studies.

I would like to acknowledge the help from my labmates Fatemeh Alipour, Pablo Millan Arias, Zihao Wang, Dr. Gurjit Randhawa and Nikhil Anil George. Surrounded by a group of people who are all interested in bioinformatics, I benefited incredibly from their ideas and generous help.

Many thanks to my friends in Waterloo - Liyuan, Yining, Qianying, Chantelle, Amber, Evan, Shun and Jayden. You not only helped me a lot with my studies but also put up with my stress from different aspects. Special thanks to my roommates Ruiyao and Yanbing, you made my work term in Toronto unforgettable with hotpots and boardgames!

Finally, I would like to thank my family. To my parents, the constant love from you have formed my understanding of a family - a happy, supportive and responsible unit: Thank you for helping me broaden my horizon and bringing so much happiness into my life! To my grandparents, the carefree childhood with you has become an indispensable part of my life: Thank you for making me the apple of your eye!

Dedication

To my parents and my grandparents.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Metagenome assembled genomes (MAGs) are a technological innovation that has allowed detailed insights into environmental microbial communities, and has strengthened understanding of the uncultured majority of microorganisms ([46], [40]). Accurate taxonomic assignment for these environmentally-derived genomes is a necessary step for identifying populations, making connections across communities and environments, and anchoring hypotheses on metabolic function and roles in biogeochemical cycles ([33], [14]).

As methods for determining phylogeny, evolutionary relationships, and taxonomy, evolved from physical to molecular characteristics, so did many species definitions change. Recently, microbial taxonomy underwent drastic changes through the Genome Taxonomy Database (GTDB, http://gtdb.exogenomic.org/) in an effort to ensure that taxonomic classifications were standardized, normalized, and evolutionary consistent. In the first GTDB release (i.e., GTDB release 80) nearly 58% of the approximately 84,000 genomes with an attached National Center for Biotechnology Information (NCBI) taxonomy saw a difference in nomenclature above Species-level [38]. With the fourth release (i.e., GTDB release 89) of GTDB, over 30% of the nearly 114,000 genomes with an NCBI taxonomy (out of 143,000 total genomes in GTDB at the time) saw a change in the assigned Species taxon [37].

In the absence of a definitive ground truth, any existing and newly proposed Species clusters would benefit from additional bioinformatics analysis by complementary genome-based classification methods, to confirm tentative taxonomic assignments. Even though existing taxonomic assignment tools (e.g., CheckM [39], BERTax [31], GTDB-Tk [8]) have achieved good classification accuracies on benchmarked tasks, they are constrained by various limitations, as described below.

Alignment-based tools (e.g., GTDB-Tk [8]) require DNA sequences to be aligned to reference sequences to obtain sequence similarities [12]. In addition, alignment-based tools assume that homologous sequences are composed of a series of linearly arranged and more

or less conserved sequence stretches, assumptions that may not always hold due to high mutation rates, frequent genetic recombination events, etc. [54]. Lastly, the utility of alignment-based tools is limited by their often prohibitive consumption of runtime and computational memory.

Genetic marker-based tools such as IDTAXA [33] and GTDB-Tk [8] rely on taxonomic markers (e.g., 16S ribosomal RNA genes, internal transcribed spacers) to identify microorganisms. The use of genetic marker-based tools is limited by the fact that partial genomes frequently lack major markers. The absence of major markers could be caused, e.g., by the genome not being sequenced to a sufficient depth to assemble well, resulting in markers of interest possibly missing from the assembly [45]. An additional reason could be that fragments carrying the markers do not bin with the rest of the genome, which is a frequent problem with 16S ribosomal RNA genes [53].

At the other end of the spectrum, alignment-free tools based on k-mer frequencies (e.g., DeepMicrobes [25], CLARK [36]) do not rely on alignment or genetic markers, and instead use k-mer frequencies as the input feature. However, existing k-mer-based tools are also limited by, e.g., the fact that they are only capable of taxonomic assignment at specific taxonomic levels (e.g., Genus, Species), and a lack of interpretability of their predicted taxonomic assignments.

To address these limitations, we propose MT-MAG, a **m**achine learning-based **t**axonomic assignment tool for **m**etagenome-**a**ssembled **g**enomes. Unlike most other tools (e.g., GTDB-Tk) MT-MAG is an *alignment-free* and *genetic marker-free* software tool. In addition, by using a hierarchically-structured local classification approach, MT-MAG is able to provide partial classification at higher taxonomic levels for the majority of MAGs that it could not confidently classify at the Species level. Lastly, for a query genome, MT-MAG outputs not only a classification path, but also a numerical classification confidence of its prediction, at each taxonomic rank. The main contributions of this paper are:

- **Partial Classification:** A novel feature of MT-MAG is that it outputs partial classifications for the majority of sequences that it cannot confidently classify at the Species level. This results in an average of 95.15% of the genomes in the datasets analyzed being either partially or completely classified. In particular, MT-MAG completely classifies, on average, 88.84% of the test sequences to the Phylum level, 88.39% to the Class level, 86.81% to the Order level, 81.17% to the Family level, and 71.13% to the Genus level.

- **Interpretability:** MT-MAG outputs numerical classification confidences for its classifications, at all taxonomic ranks along the classification path. In addition, reliability diagrams are used to assess the quality of the training sets and determine the reliability of the MT-MAG classification confidences.

- **Weighted Classification Accuracy:** To assess the performance of MT-MAG, we introduce the "weighted classification accuracy," a performance metric defined as the weighted sum of the proportions of complete and partial classifications. To the best of our knowledge, this is the first metric that incorporates a weighting scheme which reflects the fact that partial classifications at different ranks are not equally informative.

- **Large Datasets:** MT-MAG is capable of classifying large and diverse metagenomic datasets. The two datasets analyzed in this paper are: genomes from human gut microbiome species (training set 6.15 Gbp, test set 7.42 Gbp), and bacterial and archaeal genomes assembled from cow rumen metagenomic sequences (training set 239.53 Gbp, test set 2.18 Gbp).

- **Superior Performance:** MT-MAG achieves an average of 80.13% in weighted classification accuracy, for the datasets analyzed. In particular, at the Species level (the only comparable taxonomic rank with DeepMicrobes), MT-MAG outperforms DeepMicrobes by an average of 35.75% in weighted classification accuracy. In addition, MT-MAG is able to completely classify an average of 67.7% of the sequences at the Species level, compared to DeepMicrobes, which only classifies 47.45%.

The remainder of this thesis is organized as follows: In Chapter 2, we describe existing methods for taxonomic assignment in metagenomics. In Chapter 3, we describe the datasets analyzed and corresponding, as well as introduce the MT-MAG algorithm. In Chapter 4, we analyze our results, including a discussion of the novel features of MT-MAG, and a performance comparison with DeepMicrobes (the only other coomparable tool in the literature) at the Species level. In Chapter 5, we discuss limitations of our method, and future directions of research.

Chapter 2

Related Work

In Chapter 1, we discussed taxonomic assignment tools for metagenomics by grouping them into three categories, which are not mutually exclusive. In this chapter we discuss in detail one or two representative tools in each category. In Section 2.1, we discuss alignment-based tools. In Section 2.2, we discuss genetic marker-based tools. In Section 2.3, we discuss alignment-free tools. In Section 2.4, we present the rationale behind our selection of DeepMicrobes for the benchmark comparison for MT-MAG later (in Chapter 3).

2.1 Alignment-based tools

Metagenomics taxonomic assignment tools that are based on sequence alignment include MegaBLAST [32], DIAMOND [5], GTDB-Tk [37], QIIME [7], QIIME2 [7] and CheckM [39]. In the following, we introduce GTDB-Tk [37], which was used in this research (see Section 3.1) to determine the ground-truth labels for all training and test sets.

2.1.1 GTDB-Tk

The Genome Taxonomy Database (GTDB) [37] is an attempt to establish a standardised microbial taxonomy based on genome phylogeny. The only official taxonomic assignment tool for GTDB is GTDB-Tk [8], an alignment-based software tool developed by the GTDB team. It is designed to work with recent advances that allow hundreds or thousands of MAGs to be obtained directly from environmental samples. GTDB-Tk has two phases: the placement of reference genomes and taxonomic classification. In the placement of reference genome phase, GTDB-Tk accepts genome assemblies as FASTA files, identifies genes and marker genes using Prodigal [20] and HMMER [11]. Then, reference genomes are assigned to a domain based on the highest proportion of identified marker genes, and placed into the

domain-specific reference trees using pplacer [28]. In the taxonomic classification phase, a new tree is constructed using reference genomes and the query genome. In most situations, the classification is apparent from the topology of the tree; in other cases, the "alignment fraction" check what this is, as well as the results of running RED [38] and FastANI [21] are, used to determine whether to classify the query genome into an existing taxon, or to an unknown group. Benchmarking experiments involving a set of diverse archaeal and bacterial genomes demonstrates that GTDB-Tk classifications are nearly 90% consistent with manual curation.

Note that, while generally accepted and relatively accurate, GTDB-Tk suffers from the limitations of alignment-based tools, as discussed in Chapter 1.

2.2 Genetic marker-based tools

Metagenomics taxonomic assignment tools that are based based on genetic markers include IDTAXA [33], QIIME [7], QIIME2 [7], and GTDB-Tk [37]. Since we have introduced GTDB-Tk [37] as an alignment-based tool in Section 2.1, in the following, we introduce IDTAXA [33], which is listed as the only third-party taxonomic assignment tool on the GTDB official website (`https://gtdb.ecogenomic.org/tools`).

2.2.1 IDTAXA

IDTAXA was developed for taxonomic assignment of sequences involving marker genes (e.g. 16S ribosomal RNA genes, internal transcribed spacer). The algorithm for IDTAXA is split into a learning and a classifying phase. The learning phase consists of *(i)* learning a taxonomic tree, and *(ii)* ensuring that the training sequences can be correctly re-classified. More specifically, in *(i)*, IDTAXA takes a set of training sequences and their taxonomic labels, computes the k-mer frequencies for each training sequence, and records the "decision k-mers" at each rank, which are the 10% of k-mers that best distinguish among subgroups at each rank level. In *(ii)*, training sequences are re-classified via a "tree descent" approach. Similar to how decision trees are constructed in machine learning, a training sequence only descents to a subgroup if the subgroup is selected in 80 out of 100 bootstrapping experiments using the decision k-mers. In the classifying phase, IDTAXA computes the k-mer frequencies for a test sequence. It first classifies the sequence to a taxon via the tree descent approach as in *(i)*; except that IDTAXA increases the threshold to proceed to further classifications to be 98 out of 100 bootstrapping experiments. It then uses the subset of training sequences that are re-classified to this subset in *(ii)* to determine the final classification. In addition, IDTAXA also outputs interpretable classification confidences. The confidences are a weighted summation of bootstrap hits in the classifying phase.

Compared with MAPSeq [27], QIIME2 [7], etc., IDTAXA significantly avoids misclassifying sequences belonging to novel taxonomic groups. However, IDTAXA suffers from the general limitations of genetic marker-based tools, as discussed in Chapter 1.

2.3 Alignment-free tools

Metagenomics taxonomic assignment tools that are based on k-mer frequencies include DeepMicrobes [25], CLARK [36], CDKAM 2 [6] and BERTax [31]. In the following, we discuss DeepMicrobes [25] and BERTax [31], the two most recent alignment-free tools for metagenomic taxonomic assignment.

2.3.1 DeepMicrobes

DeepMicrobes [25] is a state-of-the-art alignment-free and genetic marker-free metagenomics taxonomic assignment tool. DeepMicrobes is a deep learning-based computational framework for taxonomic assignment of short metagenomic sequencing reads, at the Genus and Species level. It has the advantage of bypassing the need of a well-curated taxonomy tree. DeepMicrobes operates as follows. Firstly, a simulator simulates short sequencing reads from human gut microbiome metagenomes. Secondly, the short sequencing reads are converted to k-mer frequencies. Thirdly, DeepMicrobes employs a flat classification approach, and a deep neural network predicts taxonomic assignments and their corresponding confidences for the short sequencing reads. Lastly, the confidences are used for determining whether to output *(a)* a Genus or Species prediction, or *(b)* an "unclassified" message, by comparing the calculated confidences against a constant so-called "stopping threshold," as follows. Using the Species classification model, if the classification confidence of a certain classification exceeds or equals the confidence threshold, DeepMicrobes outputs a Species prediction; otherwise, it outputs that the read is "unclassified." During a classification task of classifying reads from species in human gut microbiomes, DeepMicrobes reports an average of 94% in constrained accuracy, and 43% in absolute accuracy.

Limitations of DeepMicrobes include the fact that it uses the same stopping threshold for all classifications, with no consideration for designing class-specific unbiased stopping thresholds. In addition, its classification model does not provide taxonomic assignment at any rank other than Species.

2.3.2 BERTax

BERTax [31] is a state-of-the-art alignment-free and genetic marker-free metagenomics taxonomic assignment tool. BERTax is a deep learning based framework to classify

DNA sequences (e.g. reads, contigs, or scaffolds) into Superkindom, Phylum, and Genus taxa without the need for known representative genomes from a database. Similar to DeepMicrobes, BERTax uses k-mer frequencies to transform DNA sequences into numerical sequences, followed by a deep neural network to predict taxonomic assignments. To discover "unknown" taxa for each taxonomic rank, BERTax groups taxa with fewer than 10,000 training fragments into an "unknown" taxon for each taxonomic rank. Having DNA sequences classified to this "unknown" taxon indicates the discovery of new taxa. However, this approach is problematic for classifying test/unknown sequences belonging to a taxon for which fewer than 10,000 sequences exist in the training set. Such sequences will be very likely to be misclassifed to the "unknown" taxon.

BERTax outperforms other tools such as Kraken2 [26], sourmash [4], Kaiju [29], etc., in the overall recall while preserving the same precision, and it demonstrates significantly superior performance for *de novo* sequences. This being said, although BERTax promises a significant benefit for metagenomics, it has not yet been experimented with metagenomics sequences. In addition, it is limited to classifications at Superkingdom, Phylum and Genus level.

2.4 Rationale for choosing DeepMicrobes for comparison

The rationale behind the selection of DeepMicrobes for a benchmark comparison with MT-MAG is as follows. First, alignment-based tools and genetic marker-based tools both place strong restrictions on the types of datasets that they are able to classify, due to their requirement for aligned sequences, respectively the requirement to use genetic markers. MT-MAG does not have such restrictions on the datasets it can classify, and selecting only restricted datasets for a comparison would not showcase its full capabilities.

Second, among the alignment-free tools, we aimed to select one that *(i)* outperforms most of state-of-art metagenomic taxonomic assignment tools, *(ii)* relies on input features that are similar to the ones used by MT-MAG, and *(iii)* it can output classification confidences. DeepMicrobes satisfies criterion *(i)*, as being a recently developed metagenomic taxonomic assignment tool, that has outperformed Kraken2 [26], Centrifuge [22], CLARK [36], DIAMOND-MEGAN [19] and BLAST-MEGAN [18]. DeepMicrobes satisfies criterion *(ii)*, as being based on k-mer frequencies as the input feature, similarly to MT-MAG. DeepMicrobes also satisfies criterion *(iii)*, because it not only outputs classification confidences, but also compares the classification confidences with the stopping threshold to determine the final output. To the best of our knowledge, DeepMicrobes is the only taxonomic assignment tool that satisfies these three criteria. Thus, we selected DeepMicrobes as the tool for a performance comparison.

Chapter 3

Materials and Methods

3.1 Materials: Datasets and task description

Two different tasks were performed in the computational experiments of this study, called **Task 1** and **Task 2**. The dataset analyzed in Task 1 was selected for direct performance comparison purposes, as it was the dataset analyzed by DeepMicrobes [25]. More specifically, the MT-MAG training set in Task 1 was based on representative genomes from species in human gut microbiomes, and the test set comprised high-quality MAGs reconstructed from human gut microbiomes from a European Nucleotide Archive study [1]. The MT-MAG training set in Task 2 was based on representative and non-representative microbial genomes from GTDB r202, and the test set comprised 913 "draft" bacterial and archaeal genomes assembled from rumen metagenomic sequence data derived from 43 Scottish cattle [47].

The rationale behind the selection of DeepMicrobes for a benchmark comparison with MT-MAG is as follows. Like MT-MAG, DeepMicrobes is a machine learning-based alignment-free and genetic marker-free metagenomic taxonomic assignment tool that uses k-mer frequencies as input feature to predict taxonomic assignments of short reads at the Genus and Species level. DeepMicrobes has demonstrated better performance at the Species level classification, and better comparative accuracy in Species abundance estimation over other state-of-the-art tools, see [19, 22, 29, 35, 36, 51]. In addition, like MT-MAG, DeepMicrobes estimates classification confidences: The reads with classification confidences below a (constant) threshold are considered to be *unclassified reads*, while the rest are considered to be *classified reads*. Within the set of classified reads, the reads whose classified Species taxa are the same as their ground-truth Species taxa are considered to be *correctly classified reads*. Lastly, to the best of our knowledge, DeepMicrobes is the only taxonomic assignment tool that enables probabilistic classification using machine learning classifiers, similar to MT-MAG's design goals.

As MT-MAG and DeepMicrobes have different requirements on their inputs, in that MT-MAG ideally requires the training sequences to be $> 10{,}000$ bp, while DeepMicrobes operates with short reads, the datasets were prepared separately for MT-MAG and DeepMicrobes.

We conclude these general remarks on the datasets used in this study with a discussion on the ground-truth labels that were used for both the training sets and test sets. We first note that the NCBI [13] labels are outdated, due to the lack of consensus on uncultivated taxa naming conventions [34]. In contrast, in GTDB a consistent naming scheme was achieved by naming uncultivated taxa as 'Genus name' sp1, 'Genus name' sp2, and so on [38]. Second, we note that GTDB provides a complete taxonomic hierarchy with no inconsistencies in naming, based on standardized phylogenetic distances used to define taxonomic ranks [37]. Third, we observe that the numerical labels used by DeepMicrobes are not biologically meaningful, and cannot be extended to other datasets. In consequence, to obtain ground-truth labels for the training and test sets in this study, we opted for using the results of running GTDB-Tk [8], based on GTDB R06-RS202, April 27, 2021.

3.1.1 Task 1: Sparse training set

The dataset for Task 1 was specifically chosen so as to allow a direct comparison between the quantitative performance of MT-MAG and that of DeepMicrobes (see "Performance metrics"). Since the genomes that the training sets for Task 1 were based on comprise only 2.4 % of the GTDB at the Species level, in the remainder of the paper this task will be referred to as *Task 1 (sparse)*.

We first note that we were unable to replicate the classification accuracies reported by DeepMicrobes, using the datasets and software provided in [25]. Absent the possibility to reproduce the results in [25] *ab initio*, and to give DeepMicrobes the best possible scenario for comparison, we opted for the alternative of using the already trained Species classification model reported in [25].

The training set for the Species classification model provided by [25], consisted of reads extracted from 2,505 representative genomes of human gut microbial species. These genomes were identified previously by a large-scale assembling study of the species in human gut microbiomes, and are available on `ftp://ftp.ebi.ac.uk/pub/databases/metagenomics/umgs_analyses`. This genome set comprised 1,952 MAGs, and 553 microbial gut Species-level genome representatives from the human-specific reference (HR) database. This 2,505 genome set was referred to in [25] as *"HGR."* Starting from HGR, DeepMicrobes [25] first assigned each species a numerical label from 0 to 2,504 (inclusive). Secondly, using the ART Illumina read simulator [17], one hundred thousand 150 basepair (bp) paired-end reads were simulated from HiSeq 2500 error model with the mean fragment size of 200 and standard deviation of 50 bp per species. Thirdly, the simulated reads were trimmed from the $3'$ end to 75–150 bp in equal probability. Lastly, these trimmed simulated read sets with their

numerical labels from 0 to 2504 (inclusive) were used as the input to DeepMicrobes. The total size of the training set of this Species classification model trained by DeepMicrobes is 56.03 Gbp.

The test set of DeepMicrobes was prepared in [25] in a similar way to the training set, and it comprised twenty-thousand 75-150 bp trimmed paired-end 75-150 bp reads simulated using ART Illumina from 3,269 high-quality MAGs reconstructed from human gut microbiomes from a European Nucleotide Archive study titled "A new **g**enomic **b**lueprint of the **h**uman **g**ut **m**icrobiota" (GBHGM) [1]. The ground-truth taxonomic labels for the test set were derived by running GTDB-Tk. The total test set size for DeeMicrobes is 14.71 Gbp.

The training set of MT-MAG was prepared as follows. Since MT-MAG uses an enhanced version of MLDSP as a subprocess (see "Methods: MT-MAG algorithm"), which achieves optimal performance when the input sequence length exceeds 10,000 bp, all contigs in HGR that were shorter than 5,000 bp were discarded. The remaining 14,358 contigs comprised the training set of MT-MAG, totalling 6.15 Gbp. The process by which MT-MAG handles the special case of imbalanced classes, and the special case of the input dataset being too large to be loaded in memory are described in Section 3.1.3.3.

The test set of MT-MAG comprised 3,269 full MAGs in GBHGM. The total size of the test set of MT-MAG is 7.42 Gbp.

Finally, to compare the DeepMicrobes classification results with those of MT-MAG, we post-processed the numerical labels of the reads in the DeepMicrobes training set, as follows. Recall that the reads in the training set were simulated from real genomes in the HGR database. Post-processing the numerical label of a read in the training set entailed using GTDB-Tk to obtain the GTDB ground-truth label of its originating genome, and this GTDB label was then associated to the numerical label of that read.

3.1.2 Task 2: Dense training set

The training sets used in Task 2 were based on genomes comprising 7.7% of GTDB taxonomy, hence this task will thereafter be referred to as *Task 2 (dense)*.

The training set of MT-MAG was prepared using GTDB R06-RS202. Note that the sizes of the genomes in GTDB are significantly larger than those of genomes in HGR. Most GTDB MAGs contain multiple contigs per genome. All contigs belonging to a given genome were pseudo-concatenated into a single sequence, by adding the symbol "O" between contigs, so as to avoid creating artificial k-mers at the junction of contigs. Then, 4 non-overlapping fragments of length 100,000 bp were selected from each such genome, using four random starts. The 4 obtained fragments belonging to the same genome were again pseudo-concatenated to form a *representative genomic fragment* for that genome. To ensure

that we had a sufficient number of representative genomic fragments to perform cross-validation, the above sampling process was repeated 20 times for each genome, resulting in 20 separate representative genomic fragments with the same genome label. The total size of the training set of MT-MAG is 239.53 Gbp. The process by which MT-MAG handles the special case of imbalanced classes, and the special case of the input dataset being too large or too small, are described in Section 3.1.3.3.

Regarding the preparation of the training set of DeepMicrobes, we note that the training stage of DeepMicrobes entails creating and loading in random access memory of a 49,871-dimensional tensor to encode the ground-truth labels of the training reads belonging to the 49,871 different species in GTDB. This tensor would consume an extremely large amount of random access memory, and would make the convergence of the training process difficult to achieve, due to the large number of classes (species labels) [24]. To make the benchmarking comparison with MT-MAG possible, we opted to include in the DeepMicrobes training set only reads belonging to the 601 Species present in its test set. Note that this design choice gives DeepMicrobes a significant advantage, since it now has to choose its predicted answers only from a small output space of 601 correct labels, while MT-MAG has to search for the correct answers in a large output space of 49,871 Species labels. Most likely, this advantage boosts the classification accuracy for DeepMicrobes by a large amount. Note also that, as a consequence of this design decision, the total size of the training set of DeepMicrobes is now significantly smaller than that of MT-MAG.

Following this design choice, the training set of DeepMicrobes was prepared from the representative and non-representative genomes of the afore-mentioned 601 species, in a similar way to the training set of DeepMicrobes in Task 1 (sparse). Approximately thirty-thousand 75-150 bp paired-end reads were simulated per species, and each species was assigned a numerical label between 0 and 600 (see Section 3.1.3.4 for details).

The test set of MT-MAG comprised 913 full microbial genomes from metagenomic sequencing of cow rumen, which were derived from 43 Scottish cattle [47]. The total sequence length of the test set of MT-MAG is 2.18 Gbp.

The test set of DeepMicrobes (reads) was prepared from the 913 full microbial genomes [47], in a similar way to the test set of DeepMicrobes in Task 1 (sparse). In the end, 10,000 75-150bp trimmed simulated paired-end reads per MAG were generated as the input to DeepMicrobes. The total size of the test set of DeepMicrobes is 2.04 Gbp, and 18,143,340 reads were simulated.

Table 3.1 and Table 3.2 provides a summary of the total number of basepairs analyzed, number of FASTA files, and number of contigs or reads for training and test sets in Task 1 (sparse) and Task 2 (dense), for MT-MAG and DeepMicrobes.

Table 3.1: Summary of total number of basepairs analyzed, number of samples and number of contigs or reads for the training and test sets in Task 1 (sparse) for MT-MAG and DeepMicrobes.

Dataset type	Tool	Total basepairs	# of FASTA files	# of contigs/reads
Training	MT-MAG	6.15 Gbp	2,505	314,840
	DeepMicrobes	56.03 Gbp	5,010	498,086,752
Test	MT-MAG	7.42 Gbp	3,269	245,564
	DeepMicrobes	14.71 Gbp	6,538	130,760,000

Table 3.2: Summary of total number of basepairs analyzed, number of samples and number of contigs or reads for the training and test sets in Task 2 (dense) for MT-MAG and DeepMicrobes.

Dataset type	Tool	Total basepairs	# of FASTA files	# of contigs/reads
Training	MT-MAG	239.53 Gbp	635,248	2,540,992
	DeepMicrobes	4.02 Gbp	1,202	35,765,154
Test	MT-MAG	2.18 Gbp	913	158,102
	DeepMicrobes	2.04 Gbp	1,826	18,143,340

3.1.3 Dataset and task details

In this section, we provide further details regarding the datasets and corresponding tasks. Section 3.1.3.1 provides the histograms for genome size, contig count, and percent GC distributions for the datasets in Task 1 (sparse) and Task (dense). Section 3.1.3.2 specifies the seed values used in experiments. Section 3.1.3.3 discusses four special cases during the data sampling stage for Task 1 (sparse) and Task 2 (dense). Section 3.1.3.4 specifies the hyper-parameter values we used for DeepMicrobes in Task 2 (dense).

3.1.3.1 Genome size, contig count, percent GC distributions

The following histograms show the genome size, contig count and percent GC distributions for different datasets in Task 1 (sparse) and Task 2 (dense). Specifically, we have:

- Figure 3.1 – Task 1 (sparse) training set: unclassified MAGs, and genomes from human-specific reference (HGR) database.
- Figure 3.2 – Task 1 (sparse) test set: high-quality MAGs reconstructed from human gut microbiomes from a European Nucleotide Archive study titled "A new **g**enomic **b**lueprint of the **h**uman **g**ut **m**icrobiota" (GBHGM) [1].

- Figure 3.3 – Task 2 (dense) training set: genomes from Genome Taxonomy Database (GTDB) R06-RS202.
- Figure 3.4 – Task 2 (dense) test set: microbial genomes from metagenomic sequencing of cow rumen, which were derived from 43 Scottish cattle [47].

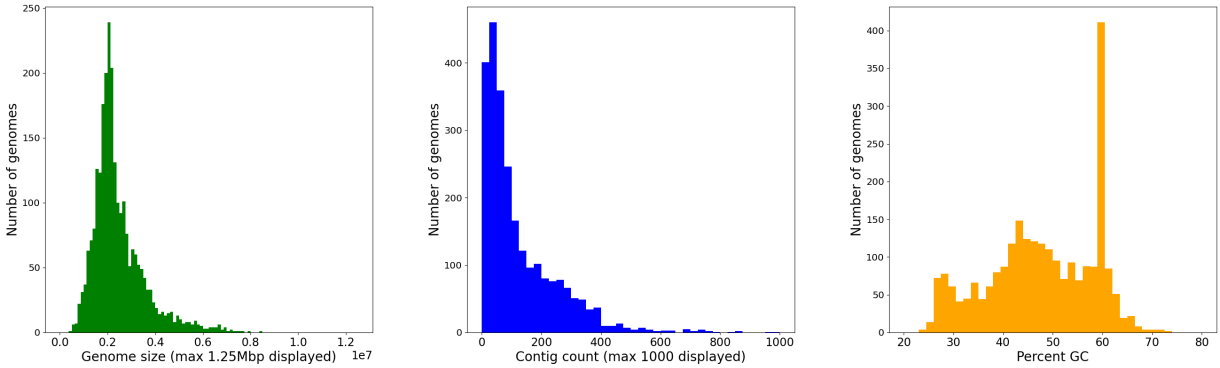


Figure 3.1: Genome size, contig count and percent GC distributions for all genomes in the HGR database. Recall that HGR comprises 1,952 MAGs and 553 microbial gut Species-level genome representatives from the human-specific reference database. From the histograms, we observe that the genome sizes are centered around 2 Mbp; the contig count is right-skewed and peaks at around 38; the percent GC peaks at around 59%.

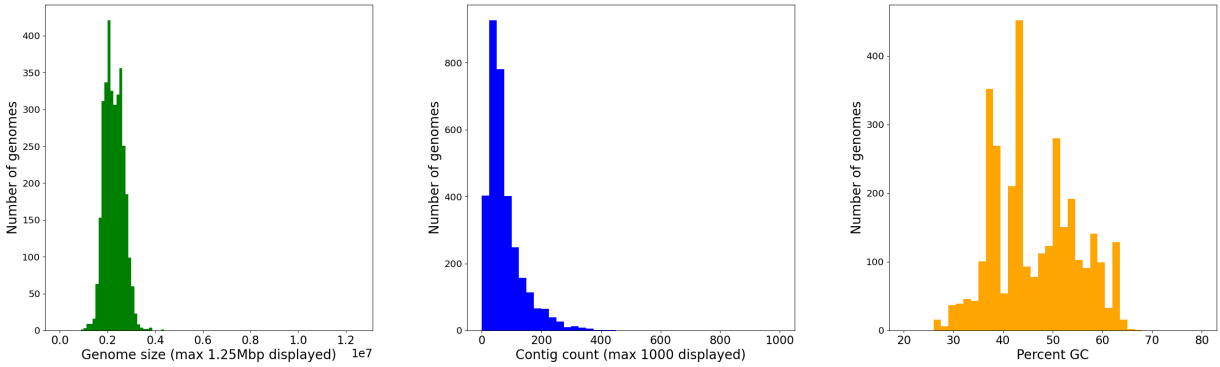


Figure 3.2: Genome size, contig count and percent GC distributions for all GBHGM MAGs. Recall that GBHGM comprises 3,269 high-quality MAGs reconstructed from human gut microbiomes from a European Nucleotide Archive study titled "A new **g**enomic **b**lueprint of the **h**uman **g**ut **m**icrobiota" (GBHGM) [1]. From the histograms we observe that the genome sizes are centered around 2 Mbp; the contig count is right-skewed and peaks at around 38; the percent GC peaks at around 44%.

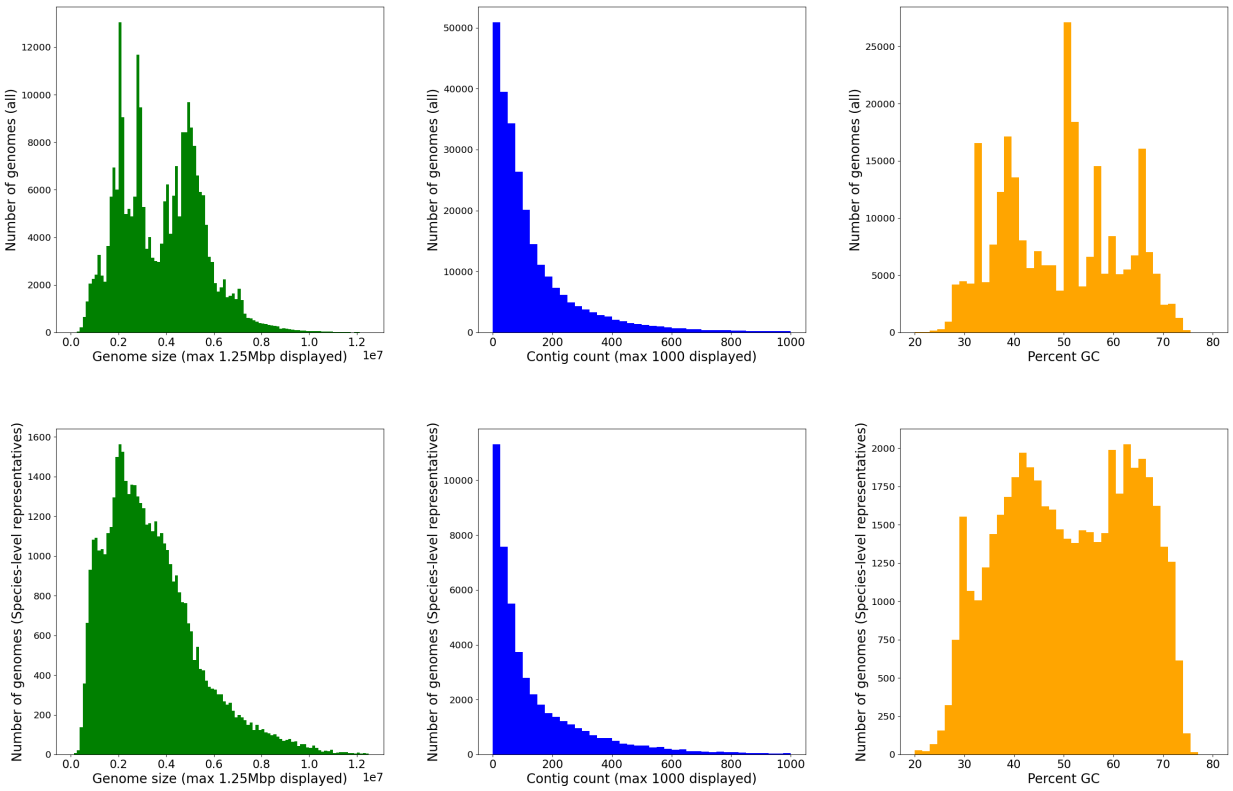


Figure 3.3: Genome size, contig count and percent GC distributions for all genomes (top panels) and Species-level representative genomes in GTDB (bottom panels). Recall that the full GTDB comprises 311,480 bacteria genomes and 6,062 archaeal genomes. From the histograms in the top panels, generated for all genomes in GTDB, we observe that the genome sizes are multimodal and peak at around 2 Mbp, 3 Mbp and 5 Mbp; the contig count is right-skewed and peaks at around 13; the percent GC peaks at around 51%. From the histograms in the bottom panels, generated for Species level representatives in GTDB, the genome sizes are right-skewed and peak at around 2 Mbp; the contig count is right-skewed and peaks at around 13; the percent GC peaks at around 41% and 64%.

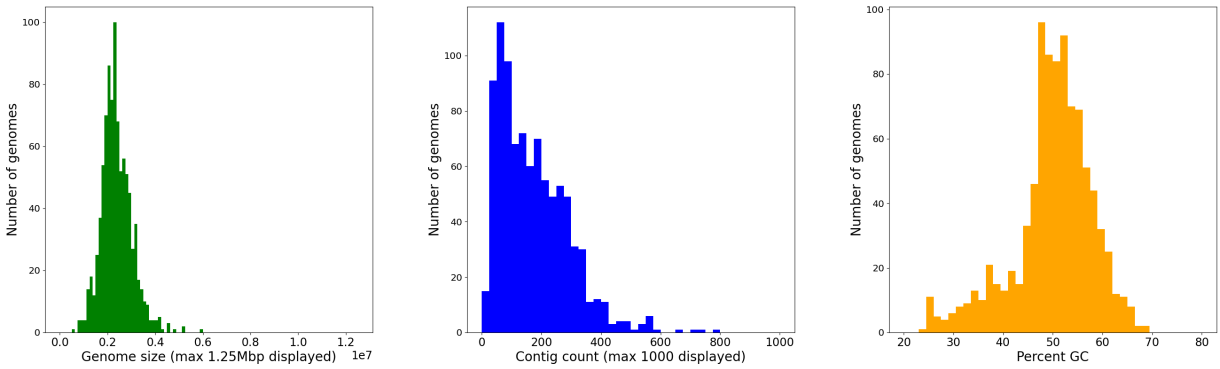


Figure 3.4: Genome size, contig count and percent GC distributions for all 913 cow rumen MAGs. Recall that there are 913 "draft" bacterial and archaeal genomes assembled from rumen metagenomic sequence data derived from 43 Scottish cattle. From the histograms, we observe that the genome sizes are centered around 2 Mbp; the contig count is right-skewed and peaks at around 63; the percent GC peaks at around 50%.

3.1.3.2 Seed fixing

To ensure our results are replicable, our presented results were produced by setting the random seed to 0 for all processes that involve randomness (unless otherwise mentioned).

3.1.3.3 Special cases

- *Dataset being too large for Task 1 (sparse).* For the Phylum level classification, as well as for the over-represented lineages from Class to Genus level classifications, we randomly selected 10% of contigs from any child taxon with more than 100 contigs. This reduced sampling bias in the datasets and reduced computational complexity without omitting any taxon within the taxonomy.

- *Dataset being too small for Task 2 (dense).* One type of special case concerns the taxa with insufficient number of representative genomic fragments. More specifically, to perform five-fold cross-validation, each child taxon needs to have at least five representative genomic fragments. To address this issue, in such cases, and for the Phylum to Genus level taxa, the child taxa with fewer than five representative genomic fragments were removed.

- *Dataset being too large for Task 2 (dense).* For Domain, Phylum, and over-represented lineages from Class to Genus, where eMLDSP consumes an extreme large amount of

random access memory, we randomly selected 10% of the representative genomes from any child taxon with more than 100 representative genomes. For less-well represented lineages from Classes to Genera, we selected all representative genomes. For Species, we selected all representative and non-representative genomes.

- *Imbalanced classification for Task 1 (sparse) and Task 2 (dense).* One scenario which could lead to problematic classifications is that of imbalanced taxon sizes. This is because significant differences in child taxon sizes may violate the assumption, necessary for most classification algorithms, that the number of the training instances for each class be roughly the same. Imbalanced class sizes may pose a challenge for predictive performance, especially for the classes with few training instances [52]. In general, there is a trade-off between balanced class sizes and the amount of variability reflected in each class. In the computational experiments in this paper, the situation of imbalanced taxon sizes was dealt with differently, depending on the taxonomic rank. For high-level classifications (i.e., Domain to Phylum, Phylum to Class, Class to Order, Order to Family), no pruning of over-sampled child taxa was performed. This is because the number of (representative) genomes in a child taxon is proportional to the amount of variability in the taxon and, for high-level classifications, the differences in the amounts of variability of different child taxa can be significant. Since the training instances are intended to capture and represent the differences in the amount of variability, the oversized child taxa must be preserved as being reflective of their respective amounts of variability, and were not pruned.

  The situation is different for low-level classifications (i.e., Family to Genus, and Genus to Species), where the differences in the amounts of variability of various child taxa is much smaller, and imbalanced taxon sizes can have a negative impact on the training model and its performance. In these latter cases, all training child taxa were pruned to relatively similar sizes as follows. After sampling, the number of contigs/representative genomic fragments in each child taxon was counted. If the number of contigs/representative genomic fragments was greater than 30, then 30 contigs/representative genomic fragments were randomly selected and used for training.

3.1.3.4 DeepMicrobes training

For Task 2 (dense), via a hyper-parameter search, we found the attention model with the following hyper-parameter values yielded the best performance: the embedding dimension being 50, the batch size being 1024, the learning rate starting with 0.001. 30,000 150 basepair (bp) paired-end reads were simulated from HiSeq 2500 error model in ART Illumina read simulator [17], with the mean fragment size of 200, standard deviation of 50 bp per species and seeds 1, 2 and 11.

3.2 Methods: MT-MAG algorithm

This section describes the hierarchically-structured local classification approach used by MT-MAG in Section 3.2.1, the eMLDSP subprocess that is at the core of MT-MAG in Section 3.2.2, and the two main phases of MT-MAG (training and classifying) in Section 3.2.3.

3.2.1 A hierarchically-structured local classification approach

Taxonomic assignment is a problem of hierarchical classification, whereby input items are grouped according to a hierarchy. A hierarchy can be formalized as a directed acyclic graph where every node can be reached by a unique path from the root node (see Figure 3.5). In machine learning, there are generally three types of approaches to hierarchical classification [3]. The simplest approach is *flat classification* where all parent nodes are ignored, and a single classifier is trained to classify each instance directly into a leaf node. The second approach is the so-called *big bang* classification where a single classifier is trained for all nodes in the hierarchy. The third approach is the *hierarchically-structured local classification*, whereby one multi-class classifier is trained for each parent-to-child relationship. This third approach is an iterative classifying process where instances classified to a child node are then further classified with the next-level classifier, where the child node is now the parent node for the next-level classifier.

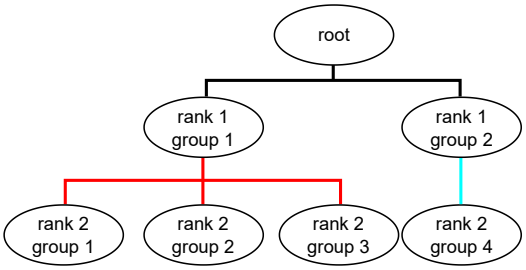


Figure 3.5: A sample hierarchy (taxonomy) with three parent-to-child relationships. A parent node with all its children nodes forms a parent-to-child relationship. A parent node without a child node is called a leaf node. The level of a node is the length of path from that node to the root node. The part highlighted in red is a multi-child classification, while the part highlighted in cyan is a single-child classification.

In contrast with DeepMicrobes which uses flat classification, MT-MAG uses hierarchically-structured local classification, for reasons detailed below. First, in the case of flat classification, an erroneous classification of a DNA sequence directly at the Species level is more likely, due to the very large number of classes at the Species level. This, in turn, results in a

higher likelihood of placing the sequence into an erroneous higher-level taxonomic rank, e.g., Order. Such a serious misplacement is less likely to happen with hierarchically-structured local classification, whereby a sequence passes through multiple classifications, from higher to lower taxonomic ranks, thus providing multiple check-points for the identification of an incorrect classification. For example, an incorrect Order classification could be prevented if any of the classifications prior to and including this level are deemed "uncertain."

In addition, in the case of flat classification, if the classification confidence of a sequence into a Species taxon does not meet the required confidence level, this sequence is simply deemed "unclassified," with no further information being provided. In contrast, the hierarchically-structured local classification provides the option of partial classification and can output partial classification paths for such sequences, even if their Species level classification is uncertain.

Finally, flat classification requires significantly more computational time and memory resources, because it involves a single big classification task wherein all the training sequences are loaded into memory simultaneously. In contrast, a hierarchically-structured local classification approach involves multiple smaller classification tasks and, for each classification task, one only needs to load into memory the sequences pertaining to the specific parent taxon being classified at this step in the hierarchy. In particular, for classifications at higher taxonomic ranks, one can use, e.g., only representative genomes as opposed to all of the genomes available for that parent taxon.

3.2.2 The enhanced MLDSP (eMLDSP) subprocess

MT-MAG uses an enhanced version of MLDSP, an alignment-free software tool that combines supervised machine learning techniques with digital signal processing for ultrafast, accurate, and scalable genome classification at all taxonomic ranks [44].

The inputs to MLDSP are pseudo-concatenated DNA sequences, together with their ground-truth taxonomic labels. After selecting a value for the parameter k, each such input DNA sequence is converted into a numerical vector containing the counts of all of its k-mers, where a k-mer is defined as a DNA subsequence of length k that does not contain the symbol "O" (used during the pseudo-concatenation process), or the symbol "N" (representing an unidentified nucleotide). Each k-mer count vector is then converted into a *k-mer frequency vector*, via dividing its k-mer counts by the total length of the sequence (excluding "O"s and "N"s). These k-mer frequency vectors are computed via order k Frequency Chaos Game Representation of a DNA sequence ($FCGR_k$) [2, 9, 50], and used as the input to MLDSP.

MLDSP consists of two main steps: (a) *Pretraining*, whereby several different classifiers' performance is evaluated by 10-fold cross validation, and (b) *Classify*, whereby MLDSP first trains the classifiers using the entire training set *(Classify-Training)*, and then classifies new DNA sequences in the test set *(Classify-Classification)*.

Figure 3.6 provides an overview of MLDSP, including the main steps to accomplish MLDSP (Pretraining), MLDSP (Classify-Training) and MLDSP (Classify-Classification).

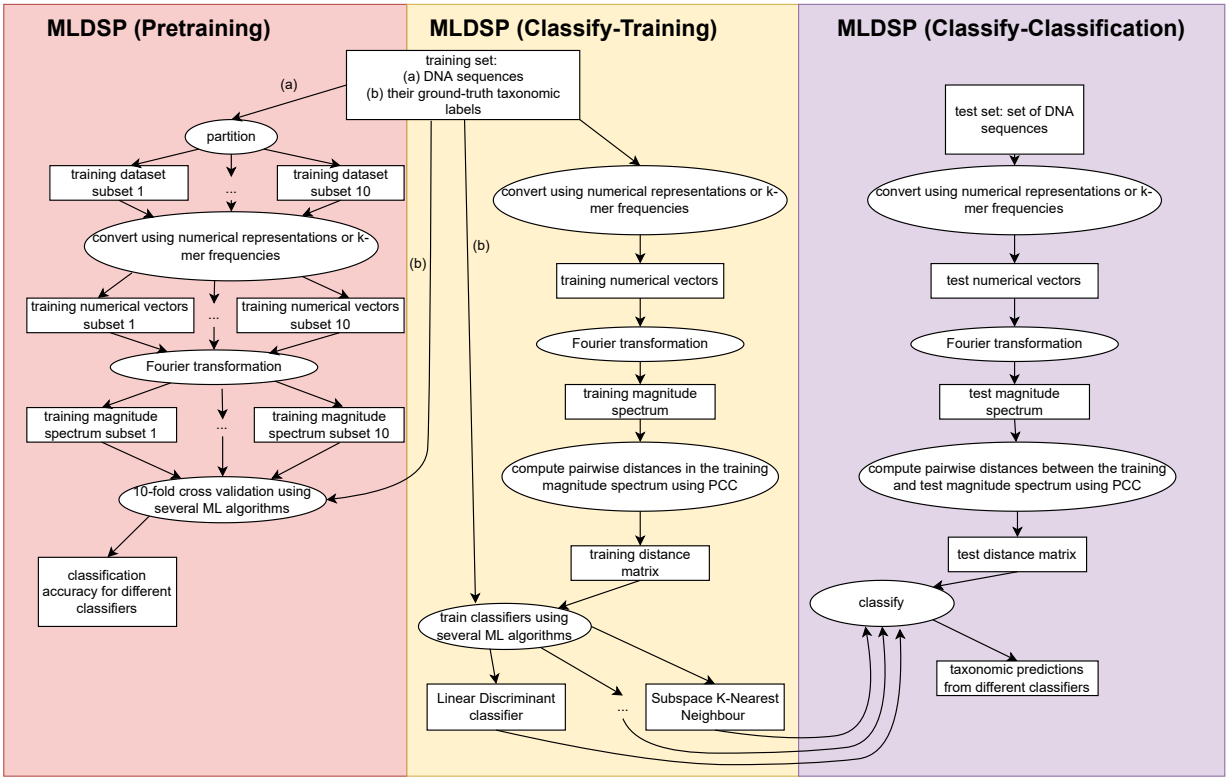


Figure 3.6: An overview of MLDSP, including the main steps to accomplish MLDSP (Pretraining), MLDSP (Classify-Training) and MLDSP (Classify-Classification). Ellipses represent computation steps. Rectangles represent inputs to and outputs from the computation steps. Note that the training set comprises both (a) DNA sequences and (b) their ground-truth taxonomic labels.

MT-MAG uses an enhanced version of MLDSP, called *eMLDSP* (enhanced MLDSP) as a subprocess. The eMLDSP subprocess augments MLDSP in several significant ways. First, it augments MLDSP by adding the capability to handle the special case where the parent taxon has only one child taxon, as well as by adding the new feature of computing classification confidences for its classifications. Second, it adds an stopping threshold picking algorithm, called "STP algorithm," which is at the core of the partial classification option feature of MT-MAG. Specifically, the STP algorithm provides an individual stopping threshold for each parent-child pair, at each taxonomic level, as opposed to the one-size-fits-all stopping threshold of DeepMicrobes at the Species level. Third, eMLDSP combines the hierarchically-structured local classification with the result of the STP algorithm to output "uncertain classification," if the classification confidence is below the stopping threshold.

Figure 3.7 provides an overview of eMLDSP, including the main steps to accomplish eMLDSP (Pretraining), eMLDSP (Classify-Training) and eMLDSP (Classify-Classification).

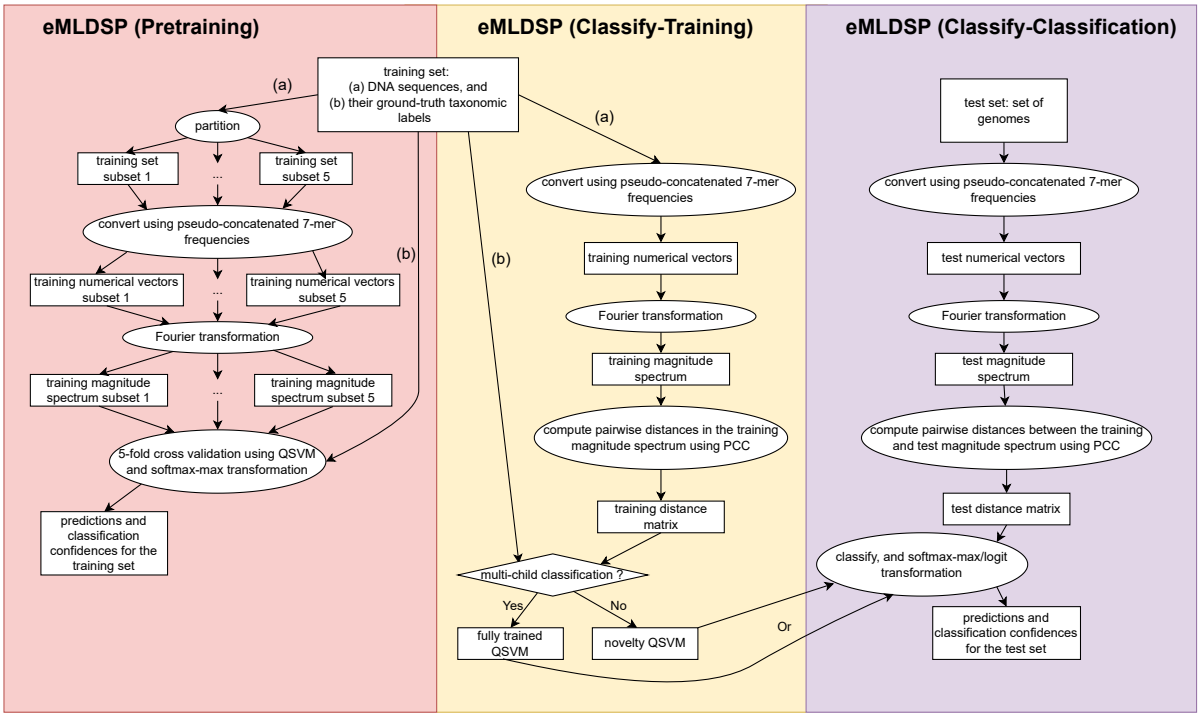


Figure 3.7: *Overview of eMLDSP*, including the main steps that comprise eMLDSP (Pretraining) (pink box), eMLDSP (Classify-Training) (yellow box), and eMLDSP (Classify-Classification) (lavender box). Ellipses represent computation steps. Rectangles represent inputs to, and outputs from, computation steps. The diamond represents a condition checking. Note that the training dataset consists of DNA sequences together with their taxonomic labels.

The MLDSP implementation of the algorithms assumes that the input DNA sequences belong to multiple child taxa (*multi-child classification*). If this is the case, in the eMLDSP (Classify-Training) step, a QSVM classifier called *fully trained QSVM* is trained, using the entire training set. In the eMLDSP (Classify-Classification) step, eMLDSP computes classifications (taxonomic assignments) for the DNA sequences in the test set by using the fully trained QSVM, and the classification confidences of these classifications using Platt scaling [42]. In contrast with its precursor, eMLDSP then applies five-fold cross-validation to obtain classifications, and uses a softmax-max transformation to compute classification confidences, for the entire training set.

Note that, when classifying a sequence belonging to a parent taxon, a single numerical classification confidence is computed for this classification, namely the confidence of

classifying the sequence into the most likely child taxon of that parent taxon. This classification confidence is computed as the maximum of the posterior likelihoods over all child taxa. These results are later used for determining the stopping thresholds for each pair of parent and child taxon.

The case where the training/test sequences belong to a single child taxon (*single-child classification*) is not addressed by MLDSP. In this case, in the eMLDSP (Classify-Training) step, a QSVM classifier called *novelty QSVM* is trained, that uses the entire training set, and sets a fraction (default 10%) of the training set as a second child-class (called outlier taxon). In the eMLDSP (Classify-Classification) step, eMLDSP computes classifications for the DNA sequences in the test set by classifying using the novelty QSVM, and computes the classification confidences of these classifications by utilizing a normalizing logit transformation. The eMLDSP (Pretraining) step is not applicable here, since there is no need for picking stopping thresholds in the case of single-child classifications.

In the following, we describe the training phase, the classifying phase, and an additional optimization step to combine the two phases formally.

3.2.3 The MT-MAG training phase and classifying phase

MT-MAG comprises two phases, training and classifying, as described below (see Section 3.2.3.1, Section 3.2.3.2 and Section 3.2.3.3 for details of the two phases, and of the optimization step that combines the two phases into a hybrid approach).

The **MT-MAG training phase** (of the training set comprising contigs in the case of Task 1 (sparse), respectively representative genomic fragments in the case of Task 2 (dense), together with their ground-truth labels) comprises multiple training processes: For each parent taxon, after preparing the training set (discarding short sequences, handling imbalances in the dataset, etc.), two situations can occur, depending on the number of child taxa:

- *Multi-child classification.* In contrast to DeepMicrobes which uses a single stopping threshold, MT-MAG has multiple stopping thresholds, one for each parent-child pair.

  Concretely, MT-MAG determines a stopping threshold for every parent-child pair, based on the confidences calculated by eMLDSP (Pretraining) with the training set as input. MT-MAG selects the stopping threshold from a list of candidate stopping thresholds, and searches for the stopping threshold T which results in the fewest number of contigs (resp. representative genomic fragments) with classification confidences lower than T, while at the same time resulting in the classification accuracies of the other contigs (resp. representative genomic fragments) being higher than the value of a user-specified accuracy parameter.

More specifically, a stopping threshold is the result of subtracting a "variability" parameter from the maximum of *(a)* the minimum of the candidate thresholds (numbers between 0 and 1) that result in a "constrained accuracy" being greater than the value of a user-specified parameter (default: 90%), and *(b)* the average of classification confidences for the contigs (resp. representative genomic fragments) with correct eMLDSP (Pretraining) classifications.

Subsequently, a QSVM classifier (the fully trained QSVM) is trained with the entire training set of this parent taxon, as part of the eMLDSP (Classify-Training) step.

- *Single-child classification.* A QSVM classifier called *novelty QSVM* is trained in the eMLDSP (Classify-Training) step. The novelty QSVM sets a fraction of the contigs (resp. representative genomic fragments) in the training set as a second child-class, called *outlier taxon*. The default fraction is set to 10%.

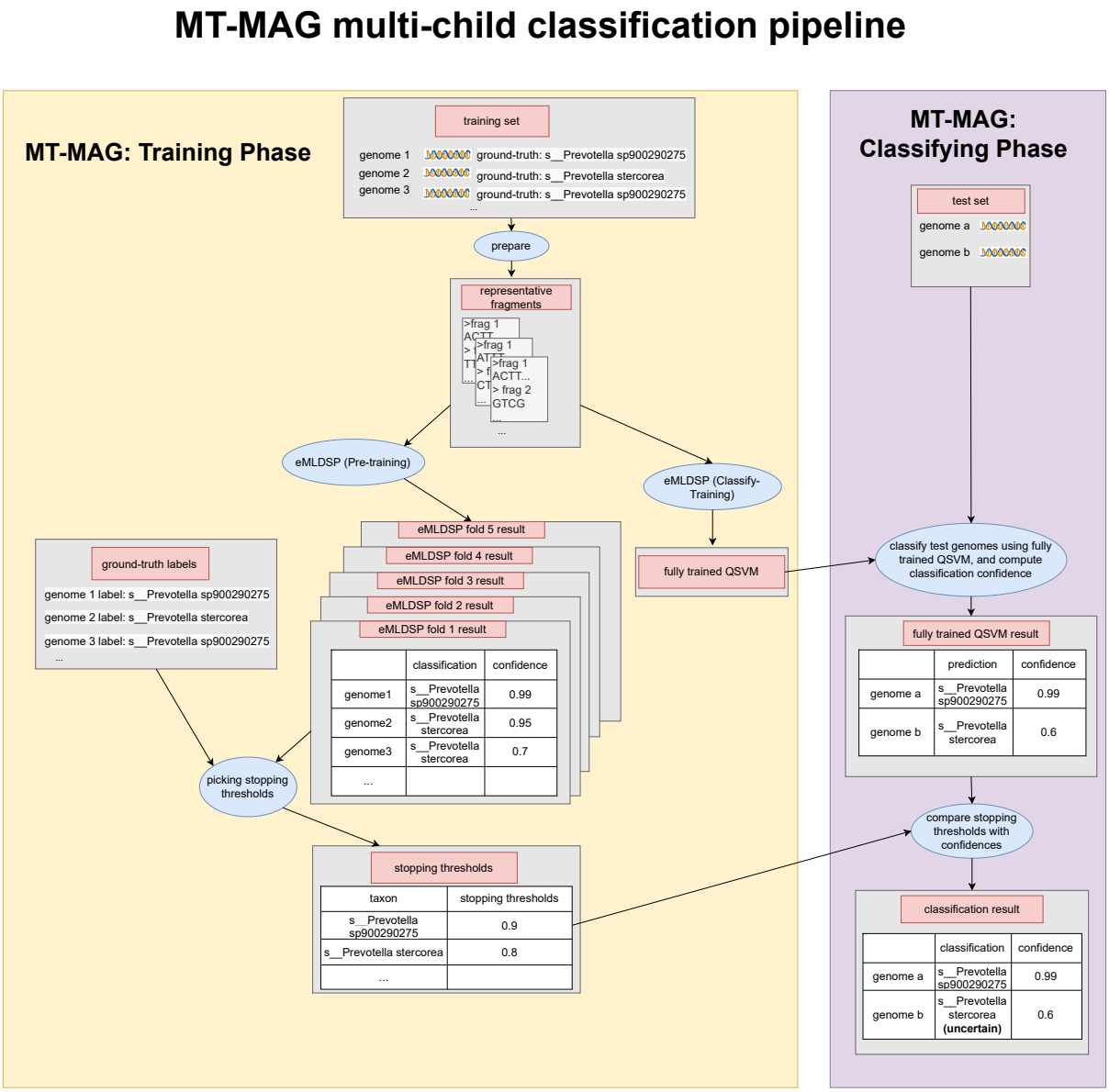
The **MT-MAG classifying phase** (of the test set comprising test genomes with known ground-truth labels, or unknown genomes) proceeds as follows. When, in the process of hierarchically-structured local classification, MT-MAG has classified a test/unknown genome into a parent taxon, and attempts to classify it further into one of its child taxa, two possibilities can occur:

- *Multi-child classification.* If the parent taxon has multiple child taxa, then the fully trained QSVM is used to classify the test/unknown genome into one of the child taxa, and this result is also used to compute a classification confidence as part of the eMLDSP (Classify-Classification) step. If this classification confidence is below the stopping threshold for this parent-child pair, then this classification is considered uncertain, and no further attempts are made to classify this test/unknown genome from the child taxon into its own child taxa.

- *Single-child classification.* If the parent taxon has a single child taxon, then the novelty QSVM is used to classify the test/unknown genome into either the child taxon or the outlier taxon as part of the eMLDSP (Classify-Classification) step, and the result is used to compute a classification confidence. If the output is the outlier taxon, then this classification is considered uncertain and no further classifications are attempted.

Given a test/unknown genome, the output of MT-MAG is either *(i)* a complete classification path down to the Species level, if all the intermediate classification confidences are greater than or equal to the stopping thresholds, or *(ii)* a partial classification path, down to the lowest taxonomic rank with a high enough classification confidence. In either case, the output of MT-MAG also includes the classification confidence for each taxon along the classification path.

Figure 3.8 illustrates the MT-MAG training phase and classifying phase for classifying two genomes belonging to a given parent taxon, into one of its two child taxa (multi-child classification). Figure 3.9 illustrates the MT-MAG training phase and classifying phase for classifying two genomes belonging to a given parent taxon, into its only child taxon (single-child classification).



(Caption on next page.)

Figure 3.8: *MT-MAG pipeline for classifying two genomes, genome a and genome b, from the parent taxon Genus* Prevotella *into its two child taxa, Species* Prevotella sp900290275*, and Species* Prevotella stercorea *(multi-child classification).* Blue ellipses represent computation steps. Gray rectangles represent inputs to, and outputs from, computation steps. In the MT-MAG training phase (yellow box), the training set is prepared and given as the input to eMLDSP (Pretraining). The classifications and classification confidences outputted in eMLDSP (Pretraining) for the training set from all folds are used for determining the stopping thresholds for every child taxon of this parent taxon. Furthermore, in eMLDSP (Classify-Training), a fully trained QSVM is trained by using the entire training data. In the MT-MAG classifying phase (violet box), the test set is given as the input to eMLDSP (Classify-Classification), together with the fully trained SVM from the training phase. eMLDSP (Classify-Classification) outputs a classification and a classification confidence for each genome in the test set. Then, the classification confidence from eMLDSP (Classify-Classification) to classify a genome from the parent taxon into a child taxon is compared with the stopping threshold of that parent taxon and child taxon pair. If the classification confidence is lower than its stopping threshold, then the output is "uncertain classification" and further classification into children of this child taxon will not be attempted.

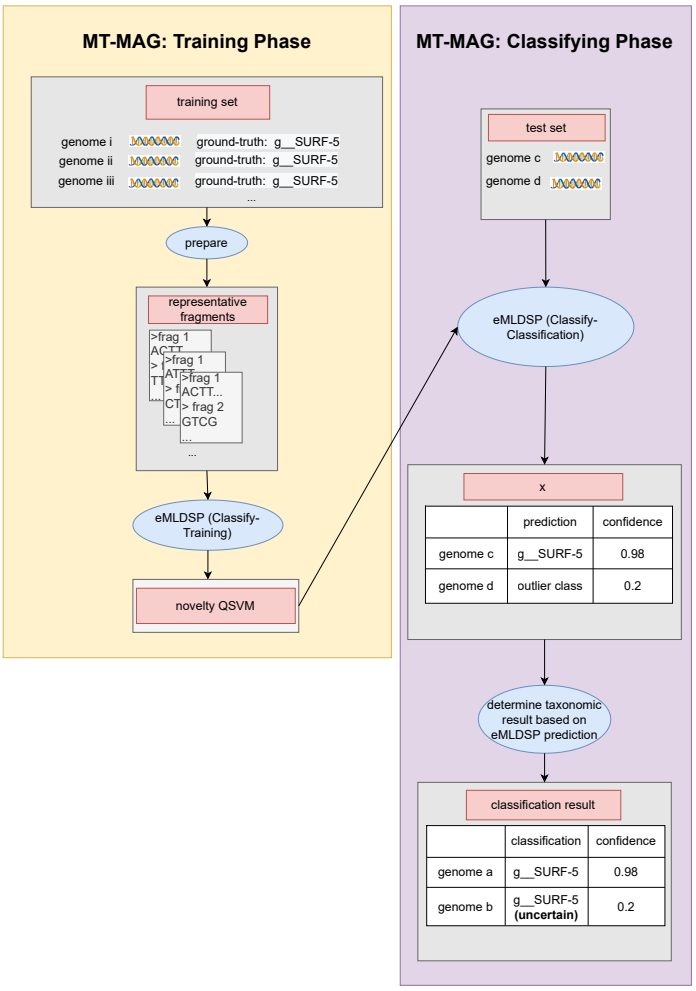


Figure 3.9: MT-MAG pipeline of classifying two genomes, genome c and genome d, from the parent taxon Phylum Abyssubacteria into the single-child taxon Class SURF-5 (single-child classification). Blue ellipses represent computation steps. Gray rectangles represent inputs to and outputs from the computation steps. In the training phase, the training set is prepared and given as the input to eMLDSP (Classify-Training), where a novelty QSVM is trained using the entire training set by considering a fraction (default 10%) of the training set to be outliers. In the classifying phase, the test set is given as the input to eMLDSP (Classify-Classification), together with the novelty QSVM from the training phase. eMLDSP (Classify-Classification) outputs a classification and a classification confidence for each genome in the test set. If a genome is classified to be the outlier taxon, then the output is an "uncertain" classification and further classification into children of this child taxon will not be attempted.

3.2.3.1 Training phase details

In this section, we will discuss MT-MAG training phase using mathematical notations. First, we give an informal description of the MT-MAG training phase. Second, we give a formal description of eMLDSP, a subprocess for MT-MAG. Third, we give a formal description of the MT-MAG training phase for multi-child classification, with the formal notations defined for eMLDSP. Lastly, we give a detailed description of the MT-MAG training phase for single-child classification.

We start by an informal description of the MT-MAG training phase.

The MT-MAG training phase for the case of multi-child classification consists of two steps: picking stopping thresholds, and training a classifier. A stopping threshold is the result of subtracting a "variability" parameter from the maximum between (a) the minimum of the candidate thresholds (numbers between 0 and 1) that result in a constrained accuracy greater than or equal to a user-specified parameter (default: 90%), and (b) the average of classification confidences of contigs/representative genomic fragments with correct eMLDSP (Pretraining) classifications.

A stopping threshold for each pair (parent-class, child-class) is stored as a pair (child-class name, child-class stopping threshold), and is utilized as follows. If a test genome is classified to a child-class with a classification confidence that is strictly smaller than the child-class's stopping threshold, then this is considered to be an "uncertain" classification and further classifications of this test genome at lower taxonomic ranks are not attempted. There are three possible cases for an "uncertain" classification. Firstly, the test genome belongs to the uncertain taxon that eMLDSP (Classify-Classification) classification classifies into, however, MT-MAG is not confident about the classification. Secondly, the test genome belongs to another existing taxon. Thirdly, the test genome belongs to a non-existing taxon that is not among the training genomes.

The stopping thresholds were used in the classifying phase to prevent further classification of test data. The classifier was used for classifying test data that have already classified into the parent taxon, into one of the parent taxon's child taxa. For example, in Figure 3.5, in the training phase of the parent-to-child relationship highlighted in red, the ground-truth labels of all the training genomes should be "rank 2 group 1", "rank 2 group 2" or "rank 2 group 3". For the entire taxonomy in Figure 3.5, MT-MAG trained three classifiers, each corresponding to one of the three parent-to-children relationships. The one from "rank 1 group 2" to "rank 2 group 4", highlighted in cyan, is for a single-child classification, and the other two (from root to "rank 1 group 1" and "rank 1 group 2", and from "rank 1 group 1" to "rank 2 group 1", "rank 2 group 2," and "rank 2 group 3") are for multi-child classifications.

The second step of the training phase in the case of multi-child classification is to train a classifier. During this step, for each parent taxon, we trained a QSVM (called fully trained

QSVM) using all the training DNA sequences of the parent taxon. This fully trained QSVM will be used, together with the aforementioned stopping thresholds, for the classifying phase.

The MT-MAG training phase for the case of single-child classification does not need a stopping threshold, since the parent taxon has a single child taxon [48]. In this case, all training DNA sequences belong to one class (the single-child taxon), and the goal is for any unknown/test genome to be either categorized as belonging to the present child taxon, or categorized as an outlier taxon (e.g., belonging to a child taxon not represented in the training set). With this goal in mind, a QSVM (called novelty QSVM) is trained, with an optional user-specified outlier fraction (default: 10%) in eMLDSP (Classify-Training). In other words, the novelty QSVM labels a fraction of the training set as outliers (i.e., a second child taxon).

To formally describe the process of picking stopping thresholds in the training phase of MT-MAG (the multi-child classifications case), we now introduce the formal definitions and notations of the concepts involved.

Given a parent-to-child relationship of the multi-child classification type, let p be the parent taxon, let D_p denote the training set of p, and let c be a child taxon of p, which is a potential classification from eMLDSP (Pretraining). Let d be a DNA sequence. In our benchmark tasks, d is a contig in the case of Task 1 (sparse), and is a representative genomic fragment in the case of Task 2 (dense).

The subprocess eMLDSP (Pretraining) for classifying the genomes belonging to a parent taxon p into one of its child taxa can be viewed as a function $M_p(d)$. This function maps each DNA sequence d in the training set (all the genomes from the parent taxon p) to a pair, i.e.,

$$M_p(d) = (pred^{M_p}(d),\ \ conf^{M_p}(d))$$

where $pred^{M_p}(d)$ is the taxonomic label of the child taxon of p that was assigned by eMLDSP to the sequence d, and the numerical classification confidence $conf^{M_p}(d)$ of this classification.

Using this notation for eMLDSP, the process of determining a stopping threshold $T_p(c)$ for each child taxon c of the parent taxon p, can be described as follows.

Denote by $l_p(d)$ the ground-truth label of the child taxon that d belongs to. Note that $l_p(d)$ is a child taxon of p.

For a child taxon c of p, define $D_p(c)$ to be the set of DNA sequences d in D_p with eMLDSP (Pretraining) classification being c, that is,

$$D_p(c) = \{d \in D_p |\ \ pred^{M_p}(d) = c\}.$$

To determine the stopping threshold for the parent taxon p and its child taxon c, we sequentially evaluate a series of candidate thresholds $\alpha \in \{0, 0.01, 0.02, ..., 1\}$, in increasing

order of their magnitude. Given a candidate threshold α, we denote by $D_p(c, \alpha)$ the set of DNA sequences d in $D_p(c)$ with $conf^{M_p}(d)$ greater than or equal to α, that is,

$$D_p(c, \alpha) = \{d \in D_p(c) |\ \ conf^{M_p}(d) \geq \alpha\}.$$

Finally, denote by $D'_p(c, \alpha)$ the set of DNA sequences d in $D_p(c, \alpha)$ whose ground-truth child labels coincide with c, that is,

$$D'_p(c, \alpha) = \{d \in D_p(c, \alpha) |\ \ l_p(d) = c\}.$$

We now define the *constrained accuracy* associated to α and c as:

$$CA_p(c, \alpha) = \begin{cases} \dfrac{card(D'_p(c, \alpha))}{card(D_p(c, \alpha))}, & \text{if } card(D_p(c, \alpha)) \neq 0 \\ 1, & \text{otherwise} \end{cases}$$

where $card(S)$ denotes the cardinality of a set S, that is, the number of its elements. In other words, $CA_p(c, \alpha)$ measures how many, out of the DNA sequences in D_p, with eMLDSP classifications equal to c and classification confidences of their classification greater than or equal to α, have been correctly classified by eMLDSP (Pretraining) (see Figure 3.10).

The *absolute accuracy* associated to α and c is defined as

$$AA_p(c, \alpha) = \begin{cases} \dfrac{card(D'_p(c, \alpha))}{card(D_p(c))}, & \text{if } card(D_p(c)) \neq 0 \\ 1, & \text{otherwise} \end{cases}.$$

In other words, $AA_p(c, \alpha)$ measures how many, out of the DNA sequences in D_p, with eMLDSP classifications equal to c, have correct classifications and classification confidences of their classification greater than or equal to α by eMLDSP (Pretraining) (see Figure 3.10).

Both $CA_p(c, \alpha)$ and $AA_p(c, \alpha)$ are between 0 and 1, and $CA_p(c, \alpha) \geq AA_p(c, \alpha)$. Indeed, note that $D_p(c, \alpha)$ is a subset of $D_p(c)$, and thus $card(D_p(c, \alpha)) \leq card(D_p(c))$. Since the numerators of $CA_p(c, \alpha)$ and $AA_p(c, \alpha)$ are the same, and the denominator of $CA_p(c, \alpha)$ is smaller than or equal to the denominator of $AA_p(c, \alpha)$, it then follows that $CA_p(c, \alpha) \geq AA_p(c, \alpha)$.

The following are the three extreme cases that are possible for $AA_p(c, \alpha)$ and $CA_p(c, \alpha)$:

- When all DNA sequences in $D_p(c)$ have classification confidences strictly less than α, we have that both $CA_p(c, \alpha)$ is 1, and $AA_p(c, \alpha)$ is 0. Indeed, in this case, since $D_p(c, \alpha)$ and $D'_p(c, \alpha)$ are empty, it follows that the numerator of $AA_p(c, \alpha)$ are 0, and by definition $CA_p(c, \alpha)$ is 1.

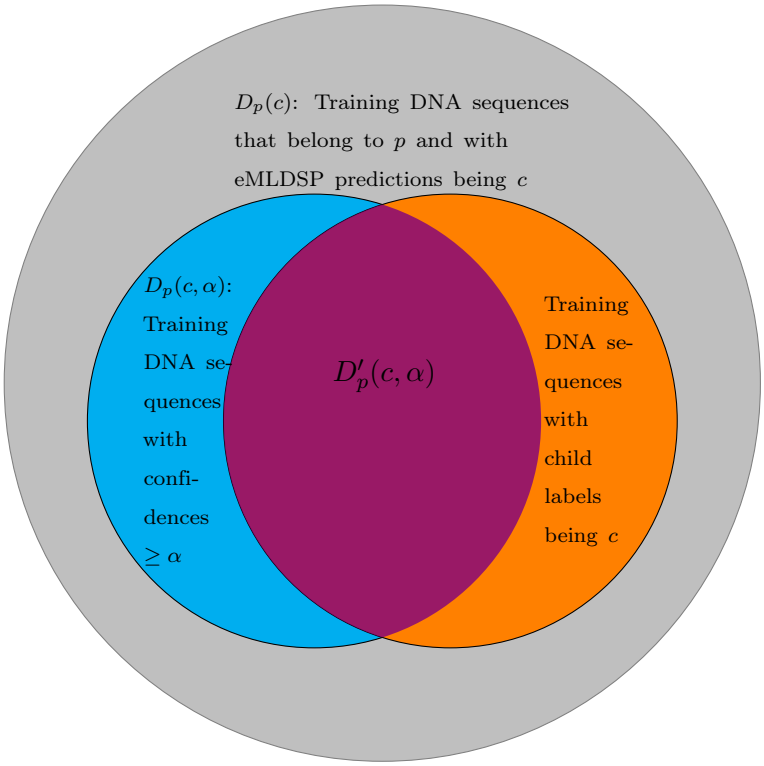


Figure 3.10: **Training set for MT-MAG (the multi-child classification case).** Relationship among the four types of DNA sequences in $D_p(c)$ (the set of DNA sequences of a parent-taxon p who are predicted by eMLDSP to have child taxon label c), for a given candidate threshold α. Within the set $D_p(c)$ (gray circle), there are two sets: the set $D_p(c, \alpha)$ (cyan circle), of DNA sequences whose classification confidence as being labelled c is greater than or equal to α, and the set of DNA sequences whose ground truth child labels is actually c (orange circle). The intersection of the two sets ($D'_p(c, \alpha)$, violet lens) is the set of DNA sequences d in $D_p(c)$ with classification confidences $\geq \alpha$ and correct eMLDSP (Pretraining) classifications. Visually, we have that $CA_p(c, \alpha)$ is the ratio of violet lens set to the cyan circle set, and $AA_p(c, \alpha)$ is the ratio of the violet lens set to the gray circle set.

- When all DNA sequences in $D_p(c)$ are correctly classified and have classification confidences greater than or equal to α, we have that $CA_p(c, \alpha)$ is 1, and $AA_p(c, \alpha)$ is 1. Indeed, in this case, since $D_p(c) = D_p(c, \alpha) = D'_p(c, \alpha)$, it follows that the numerators and denominators of $CA_p(c, \alpha)$ and $AA_p(c, \alpha)$ are the same.

- When all DNA sequences in $D_p(c)$ are incorrectly classified and have classification confidences greater than or equal to α, we have that $CA_p(c, \alpha)$ is 0, and $AA_p(c, \alpha)$ is 0. Indeed, in this case, since $D'_p(c, \alpha)$ is empty, it follows that the numerators of $CA_p(c, \alpha)$ and $AA_p(c, \alpha)$ are both 0.

To intuitively understand what a "reasonably good" α means, we now discuss the case when $CA_p(c, \alpha)$ is 1, but $AA_p(c, \alpha)$ is close to 0. In general, a high $AA_p(c, \alpha)$ indicates a good choice of α. If this is not achievable, an indicator of a"reasonably good" choice of α is that $CA_p(c, \alpha)$ is high, while $AA_p(c, \alpha)$ is not too low. Intuitively, the latter requirements mean that *(i)* a high proportion of DNA sequences classified as c with classification confidence $\geq \alpha$, are correctly classified (have ground-truth label c) (i.e., high $CA_p(c, \alpha)$), and that *(ii)* in addition, among the training DNA sequences classified to have label c, there are sufficiently many training DNA sequences whose classification confidence is at least α (i.e., not too low $AA_p(c, \alpha)$). Note that requiring only that $CA_p(c, \alpha)$ be high could result in situations as follows being considered a "reasonably good" choice for α, which would be erroneous: Suppose we have 1,000 training DNA sequences, and that only one DNA sequence has classification confidence $\geq \alpha$ and is correctly classified; then $CA_p(c, \alpha)$ attains the maximum value of 1, but $AA_p(c, \alpha)$ is close to 0.

For each pair comprising a parent taxon p and child taxon c, and given a list of candidate thresholds $\{0, 0.01, 0.02, ..., 1\}$, our goal is to determine a stopping threshold $T_p(c)$ from the candidates in this list. The algorithm for computing $T_p(c)$ uses two criteria. Firstly, the algorithm searches for the minimum candidate threshold $\alpha_1 \in \{0, 0.01, 0.02, ..., 1\}$ that results in the constrained accuracy $CA_p(c, \alpha)$ greater than or equal to an optional user-specified constrained accuracy (default: 0.9). Since $AA_p(c, \alpha)$ is a decreasing function of α, choosing the minimum threshold candidate α results in the highest possible $AA_p(c, \alpha)$, balancing the twin objectives for $CA_p(c, \alpha)$ and $AA_p(c, \alpha)$. Secondly, the algorithm computes the average α_2 of the classification confidences of the training DNA sequences in $D_p(c)$ that are correctly classified as c, and computes $max\{\alpha_1, \alpha_2\}$. Furthermore, to account for the additional variability in the test set (which results, in general, in lower classification confidences for the test set compared to the training set), the algorithm accepts an optional user-specified "variability" parameter v between 0 and 1 (default: 0.2). The stopping threshold is now computed as $T_p(c) = max\{\alpha_1, \alpha_2\} - v$.

Algorithm 1 shows the pseudocode for the Stopping Threshold Picking (STP) algorithm. Algorithm 2 shows the pseudocode for the training phase.

Algorithm 1 Stopping Threshold Picking Algorithm Pseudocode

Input p : the parent taxon with more than one child taxon
CA_u : user-specified constrained accuracy (default 0.9)
v : variability (default 0.2)
δ : the gap between candidate thresholds, default 0.01
Output T_p: pairs of child taxon of p and its stopping threshold

procedure STP($p, CA_u, v, \delta = 0.01$)
  A $\leftarrow [0, \delta, 2\delta, ..., 1]$ ▷ list of candidate thresholds
  $\alpha_{num} \leftarrow 1/\delta + 1$ ▷ number of candidate thresholds α
  $C_p \leftarrow$ child taxa of p
  $D_p \leftarrow$ training set of p
  for c `in` C_p **do**
    $D_p(c) \leftarrow \{d \in D_p : pred^{M_p}(d) = c\}$ ▷ assume non-empty
    $D'_{pc} \leftarrow \alpha_{num}$ of zeros ▷ init, list of $card(D'_p(c, \alpha))$
    $D_{pc} \leftarrow \alpha_{num}$ of zeros ▷ init, list of $card(D_p(c, \alpha))$
    ▷ init, list of confidences for sequences with correct classifications in $D_p(c)$
    $\alpha_{pc} \leftarrow []$
    for d `in` $D_p(c)$ **do**
      ▷ the maximum candidate threshold that is smaller than the confidence
      $\alpha_{start} \leftarrow max\{\alpha \leq conf^{M_p}(d) | \alpha \in \text{A}\}$
      $\alpha_{idx} = \frac{\alpha_{start}}{\delta} + 1$ ▷ the index of α_{start} in A
      ▷ if the stopping threshold α is in the first α_{idx} elements of A, d contributes 1 to $card(D_p(c, \alpha))$
      $D_{pc}[1 : d_{idx}] + +$
      if $pred^{M_p}(d) = l_p(d)$ **then** ▷ a correct classification
        $D'_{pc}[1 : \alpha_{idx}] + +$ ▷ d contributes 1 to $card(D'_p(c, \alpha))$
        ▷ store the confidence for the correct classification
        append α_{pc} with $conf^{M_p}(d)$
    for $card(D'_p(c, \alpha)), card(D_p(c, \alpha)), \alpha$ `in` D'_{pc}, D_{pc}, A **do**
      $CA_p(\alpha, c) \leftarrow 1$
      if $card(D_p(c, \alpha)) \neq 0$ **then**
        $CA_p(\alpha, c) \leftarrow \frac{card(D'_p(c,\alpha))}{card(D_p(c,\alpha))}$
      if $CA_p(\alpha, c) \geq CA_u$ **then** ▷ check the first criterion
        $\alpha_1 \leftarrow \alpha$
        break ▷ found the minimum $\alpha \in$ A that satisfies the first criterion
    if $card(\alpha_{pc}) \neq 0$ **then** ▷ no correct classifications for child taxon c
      $\alpha_2 \leftarrow mean(\alpha_{pc})$ ▷ second criterion
      $T_p(c) \leftarrow max(\alpha_1, \alpha_2) - v$
    Add $(c, T_p(c))$ to T_p

Algorithm 2 Training Phase Pseudocode

Input p : the parent taxon with more than one child taxon
CA_u : user-specified constrained accuracy (default 0.9)
v : variability (default 0.2)
Output M_p^{ft}: a fully trained QSVM, and
T_p: pairs of child taxon in p and its corresponding stopping threshold, or
M_p^{nv}: a novelty QSVM

1: **procedure** TRAINING(p, CA_u, v)
2: **if** p has multiple child taxon **then** ▷ multi-child classification
3: $M_p^{ft} \leftarrow$ a fully trained QSVM trained by the training set of p
4: $T_p \leftarrow STP(p, CA_u, v)$
5: **else** ▷ single-child classification
6: $M_p^{nv} \leftarrow$ a novelty QSVM trained by the training set of p

3.2.3.2 Classifying phase details

The classifying phase comprises both *(i)* classifying test genomes with known ground-truth labels, and *(ii)* classifying unknown genomes (without known ground-truth labels). Note that in *(i)*, the ground-truth labels are not used in the classifying phase, and are only needed for computing performance metrics.

In both cases, the classifying phase mimics the hierarchically-structured local classification to classify test/unknown genomes into a leaf taxon (Species-level). The process starts from the root (the highest level parent taxon), and it follows a classification path through increasingly lower taxonomic ranks. This is illustrated in Figure 4.1, which depicts a fictional hierarchical classification of a genome g. The idea of the process is as follows. Suppose that MT-MAG has already determined that the test/unknown genome g belongs to a taxon p.

If this is a multi-child classification, denote the fully trained QSVM associated to a parent taxon p by $M_p^{ft}(g)$, where "ft" stands for "fully trained." For a input genome g, the function $M_p^{ft}(g)$ outputs a pair

$$(pred^{M_p^{ft}}(g), \ \ conf^{M_p^{ft}}(g)),$$

where $pred^{M_p^{ft}}(g)$ is the taxonomic label of the child taxon of p that the fully trained QSVM predicts for g, and $conf^{M_p^{ft}}(g)$ is the classification confidence of this classification.

Assume that $pred^{M_p^{ft}}(g) = c_j$, where c_j is one of the child taxa of p. Two outcomes are now possible. If the classification confidence for this classification is greater than or equal to the stopping threshold for the pair p and c_j, that is, if $conf^{M_p^{ft}}(g) \geq T_p(c_j)$, then MT-MAG

outputs "the genome g as belonging to c_j, with classification confidence $conf^{M_p^{ft}}(g)$," and then proceeds to further classify g into one of the child taxa of c_j. If, on the other hand, the classification confidence is lower than the stopping threshold for this parent-child pair, that is, if $conf^{M_p^{ft}}(g) < T_p(c_j)$, then MT-MAG outputs "classification of g as c_j is uncertain, and the classification confidence is $conf^{M_p^{ft}}(g)$," and stops attempting to classify g further down the taxonomy.

If this is a single-child classification, denote the novelty QSVM associated to a parent taxon p by $M_p^{nv}(g)$, where "nv" stands for "novelty." For a input genome g, the function $M_p^{nv}(g)$ outputs a pair

$$(pred^{M_p^{nv}}(g), \ \ conf^{M_p^{nv}}(g)),$$

where $pred^{M_p^{nv}}(g)$ is the taxonomic label of the child taxon of p that the novelty QSVM predicts for g, and $conf^{M_p^{nv}}(g)$ is the classification confidence of this classification.

Two outcomes are possible. If the novelty QSVM associated to p classifies g as belonging to the single-child taxon c of p, that is, if $pred^{M_p^{nv}}(g) = c$, then MT-MAG outputs "the genome g belongs to c, with classification confidence $conf^{M_p^{nv}}(g)$," and then proceeds to further classify g into one of the child taxa of c. If, on the other hand, if M_p^{nv} classifies g to the outlier taxon, that is, if $M_p^{nv} \neq c$, then MT-MAG outputs "classification of g as c is uncertain, and the classification confidence is $conf^{M_p^{nv}}(g)$," and stops attempting to classify g further down the taxonomy.

Algorithm 3 shows the pseudocode for the classifying phase.

3.2.3.3 Optimization step details

A careful analysis of MT-MAG's time complexity reveals that a significant part of its runtime comes from its training phase. In addition, in the task of classification of a test/unknown genome, not all novelty QSVMs, fully trained QSVMs, and stopping thresholds computed during the training phase are used in the classifying phase. Indeed, for a test/unknown genome, only the novelty/fully trained QSVM's and stopping thresholds local to its classification path will be actually used for the classification.

Thus, MT-MAG can be optimized to prevent computation of unnecessary novelty/fully trained QSVMs, and unnecessary stopping thresholds, as follows. First, with the exception of the root taxon, MT-MAG will only train a novelty/fully trained QSVM of a parent taxon p if there are test/unknown genomes that have been classified to p by the novelty/fully trained QSVM of the parent of p. Second, the algorithm for determining the stopping thresholds can be optimized by computing only the stopping thresholds for pairs of parent taxon p and child taxon c, in the case where *(i)* there are test/unknown genomes that have been classified to p by the novelty/fully trained QSVM of the parent of p, and *(ii)* there are test/unknown genomes that have been classified to c by the fully trained QSVM of p.

Algorithm 3 Classifying Phase Pseudocode

Input g : the test/unknown genome to be classified
Output cp : the classification path with classification confidences for g

1: **procedure** CLASSIFYING(g)
2: $cp \leftarrow (\text{root}, 1)$ ▷ init, classification path with classification confidences
3: $t \leftarrow$ root ▷ init, current taxon
4: **while** t is not a leaf taxon **do**
5: **if** t has more than one child taxon **then** ▷ multi-child classification
6: **if** $conf^{M_p^{ft}}(g) \geq T_p(t)$ **then** ▷ confidence passes stopping threshold
7: $t \leftarrow pred^{M_p^{ft}}(g)$
8: append cp with $(t, conf^{M_p^{ft}}(g))$
9: **else** ▷ confidence does not pass stopping threshold
10: append cp with $(t, \text{“uncertain”}, conf^{M_p^{ft}}(g))$
11: **break**
12: **else** ▷ single-child classification
13: $sc \leftarrow$ the single child taxon of t
14: **if** $pred^{M_p^{nv}}(g) = sc$ **then** ▷ classified to the single child taxon
15: $t \leftarrow sc$
16: append cp with $(sc, conf^{M_p^{nv}}(g))$
17: **else** ▷ classified to the outlier taxon
18: append cp with $(sc, \text{“uncertain”}, conf^{M_p^{nv}}(g))$
19: **break**

Chapter 4

Results

In this section, we first describe the performance metrics used for measuring the performance of MT-MAG in Section 4.1. Second, in Section 4.2 we present a detailed analysis of the novel features of MT-MAG: *(i)* the capability to classify a DNA sequence at all taxonomic ranks, *(ii)* the capability to output an interpretable classification confidence for the classification at each taxonomic rank along the classification path, and *(iii)* the capability to output a "partial classification" path when the classification confidence of a classification does not meet a given threshold. Third, in Section 4.3, since DeepMicrobes is able to classify only at the Species level, we summarize the results of a comparative analysis of the performance of MT-MAG with that of DeepMicrobes at the Species level. Lastly, in Section 4.4, we discuss several details of the computational experiments.

Note that for the benchmark comparisons between MT-MAG and DeepMicrobes, two types of test genomes were excluded for Task 1 (sparse), as detailed below. First, the ground-truth labels of the test set were determined by running GTDB-Tk [8]. If GTDB-Tk classified the genomes to unnamed species, then these test genomes were excluded as not having ground-truth labels to benchmark the performance of MT-MAG. Second, the test genomes whose GTDB-Tk-predicted species did not exist in the training set were also excluded. The rationale is that the species in the training set (HGR) form a finite subset of GTDB, and the GTDB-Tk-predicted species for a test genome may not necessarily be in this finite subset. If these genomes would not have been excluded, their MT-MAG and DeepMicrobes classifications could not be correct, since their ground-truth labels had never been seen during training. For Task 2 (dense), we only excluded the test genomes for which GTDB-Tk-predicted unnamed species.

The tasks for MT-MAG were run using python3 and MATLAB R2019b on a x86_64 Ubuntu machine. The tasks for DeepMicrobes were run using python3 on Vector's Vaughan cluster.

4.1 Performance metrics

In this section, we define the terminology and the performance metrics used to discuss and assess the performance of MT-MAG's classification of test genomes.

In this section, we define the terminology and the performance metrics used to discuss and assess the performance of MT-MAG's classification of test genomes.

A classification of a genome x from taxonomic rank tr_1 to taxonomic rank tr_2, is called *a classification at* tr_2. Given a taxonomic rank tr, we call the classification of x a *complete classification at* tr, if the classification confidences of classifying x at all taxonomic levels higher than and including tr, are greater than or equal to the respective stopping thresholds. The classification of x is called an *uncertain classification at* tr if the confidence of the classification at tr is strictly less than the stopping threshold of this parent-child pair, and the confidences of the classifications at ranks higher than tr are greater than or equal to their corresponding stopping thresholds. If the classification is uncertain a rank higher than tr, we call it an *unattempted classification at* tr. At the end of the MT-MAG classifying phase for an input genome x, if the classification of x is uncertain at any taxonomic rank lower than the first non-root rank, then we say that x is *partially classified*. On the other hand, if the output of the classifying phase is that x is completely classified at the lowest taxonomic rank (herein Species), then we say that x is *completely classified.*

We note that for a given test genome, the output of its classification at rank tr can be only one of the following: complete classification at tr (if classifications all the way down to tr exceed their thresholds), uncertain classification at tr (if classifications at all higher ranks exceed their thresholds, but the classification at tr is below the threshold), or unattempted classification at tr (if the classification at any rank higher than the one right above tr is below the threshold).

Finally, we say that a classification of x is a *correct classification down to* tr if it is a complete classification at tr, and a correct classification at all taxonomic ranks higher than, and including, tr. By definition, all genomes have correct classifications down to the root.

As an example, in Figure 4.1, the classification of genome x is a complete classification at rank 1, an uncertain classification at rank 2, and an unattempted classification at rank 3. In addition, if the ground-truth taxon of genome x at rank 1 is "rank 1 group 1", then the classification of genome x is a correct classification at rank 1, as well as a correct classification down to rank 1. Note that, for each classification of a parent taxon, the number of stopping thresholds equals the number of that parent's child taxa. In contrast, each such classification has associated with it a single classification confidence, that of classifying the genome into a single, "best-guess," child taxon.

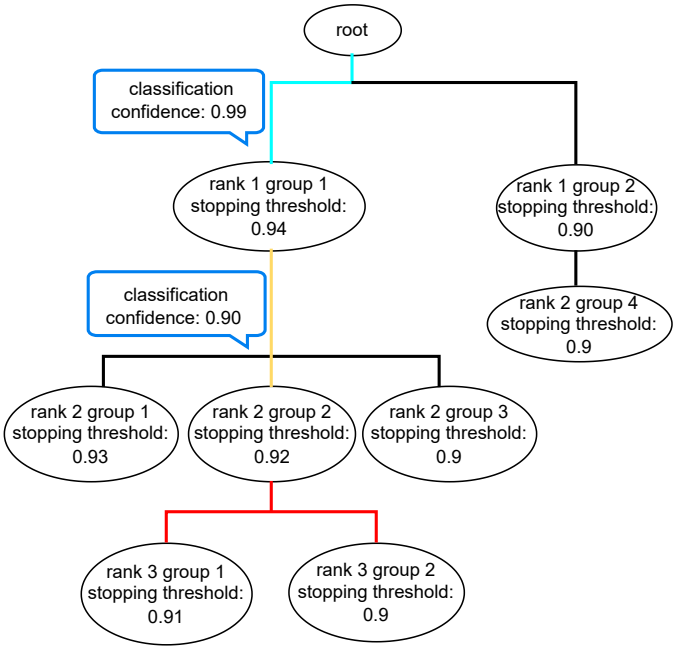


Figure 4.1: Example of the classification path for a genome x. The pre-calculated stopping thresholds are listed under the corresponding taxon labels. The classification confidences are listed inside blue-bordered rectangles. MT-MAG classifies x from root into "rank 1 group 1" with confidence 0.99, which is greater than the stopping threshold for "rank 1 group 1" (0.94), so MT-MAG continues its classification for x. In the next iteration MT-MAG classifies x from "rank 1 group 1" into "rank 2 group 2" with confidence 0.90, but since this is below the stopping threshold of the parent into its child "rank 2 group 2" (0.92), this classification is deemed "uncertain" and MT-MAG does not attempt further classifications. The path in cyan indicates complete classification(s), the path in yellow indicates uncertain classification(s), and the part in red indicates unattempted classifications.

With this terminology, for a given taxonomic rank tr, we define the following performance metrics (the subscript g indicates that these metrics refer to genomes):

- $CA_g(tr)$ (constrained accuracy): the proportion of the test genomes with correct classifications down to tr, to the test genomes with complete classifications at tr.
- $AA_g(tr)$ (absolute accuracy): the proportion of the test genomes with correct classifications down to tr, to all test genomes.
- $WA_g(tr)$ (weighted classification accuracy): the weighted sum of the proportions of the test genomes with correct classifications down to tr to all test genomes, the test genomes with uncertain classifications at tr to all test genomes, and the test genomes

with unattempted classifications at tr to all test genomes. (Hereafter, *weighted classification accuracy* will sometimes be called just *weighted accuracy*.)

The weights are assigned as follows. Consider a list of increasingly lower taxonomic ranks $tr_0, tr_1, ..., tr_i$, where tr_0 is the root, and tr_i is tr. To calculate $WA_g(tr)$, the weight for a test genome with a correct classification down to $tr_i = tr$ (complete classification at tr_i, *and* a correct classification at tr_i) is 1. Otherwise, the weight of the test genome is j/i, where $0 \leq j < i$, if it has a correct classification down to tr_j, but does not have a correct classification down to tr_{j+1} (the latter condition avoids double counting). The underlying assumption is that the test genomes are always assumed to belong to the root, and note that genomes that do not have correct classifications down to any taxonomic rank below the root are given weight 0.

This weighting scheme reflects the fact that partial classifications at different ranks are not equally informative. For example, a correct classification of a test genome down to the Phylum level is less informative than a correct classification of a test genome down to the Genus level.

- $CR_g(tr)$ (complete classification rate): the proportion of the test genomes with complete classifications at tr, to all test genomes.

The three accuracies $CA_g(tr)$, $AA_g(tr)$ and $WA_g(tr)$ are numbers between 0 and 1, with $CA_g(tr) \geq AA_g(tr)$, and where higher values indicate better performance. The complete classification rate $CR_g(tr)$ is a number between 0 and 1, and a higher value indicates a higher proportion of genomes that are completely classified at tr. See Section 4.4.1 for the formal definitions of these performance metrics.

4.2 MT-MAG novel features

We now present a detailed analysis of the novel features of MT-MAG. In Section 4.2.1, we analyze the capability to classify a DNA sequence at all taxonomic ranks. In Section 4.2.2, we analyze *(i)* the capability to output an interpretable classification confidence for the classification at each taxonomic rank along the classification path, and *(ii)* the capability to output a "partial classification" path when the classification confidence of a classification does not meet a given threshold. In Section 4.2.3, we assess the reliability of the classification confidences using reliability diagrams.

4.2.1 Classifications at all taxonomic ranks

In contrast with DeepMicrobes which only classifies reads at the Species level, a significant feature of MT-MAG is its capability to classify genomes at all taxonomic ranks. Table 4.1

and Table 4.2) provides a summary of MT-MAG's performance metrics, at all taxonomic ranks, for both Task 1 (sparse) and Task 2 (dense). Table 4.3 provides a summary of the percentages of the test sequences completely classified by MT-MAG vs. classified by DeepMicrobes, at all taxonomic ranks.

Table 4.1: Summary of MT-MAG performance metrics at all taxonomic ranks, for Task 1 (sparse): constrained accuracy $CA_g(tr)$, absolute accuracy $AA_g(tr)$, weighted accuracy $WA_g(tr)$, and complete classification rate $CR_g(tr)$ (higher is better).

Taxonomic Rank	$CA_g(tr)(\%)$	$AA_g(tr)(\%)$	$WA_g(tr)(\%)$	$CR_g(tr)(\%)$
Phylum	100.00	93.80	93.80	93.80
Class	100.00	93.80	93.80	93.80
Order	99.56	91.99	92.59	92.40
Family	100.00	82.67	87.93	82.67
Genus	99.39	71.53	84.22	71.96
Species	81.53	57.60	81.90	70.65
Average	96.75	81.90	89.04	84.21

Table 4.2: Summary of MT-MAG performance metrics at all taxonomic ranks, for Task 2 (dense): constrained accuracy $CA_g(tr)$, absolute accuracy $AA_g(tr)$, weighted accuracy $WA_g(tr)$, and complete classification rate $CR_g(tr)$ (higher is better).

Taxonomic Rank	$CA_g(tr)(\%)$	$AA_g(tr)(\%)$	$WA_g(tr)(\%)$	$CR_g(tr)(\%)$
Domain	99.66	96.13	96.13	96.46
Phylum	97.50	83.82	89.83	83.87
Class	97.74	81.10	83.35	82.98
Order	97.96	79.56	82.46	81.22
Family	98.06	78.12	81.85	79.67
Genus	96.70	67.96	78.94	70.28
Species	98.63	63.87	78.36	64.75
Average	98.03	78.36	83.56	79.89

In Task 1 (sparse) MT-MAG achieves an excellent performance at all taxonomic ranks from Phylum to Genus, with a slight drop in performance at the Species level (see Table 4.1). Specifically, the MT-MAG constrained accuracies $CA_g(tr)$ are above 99% at all taxonomic ranks, except at the Species level $CA_g(Species)$, where they drop to 81.53%. The increase in the number of incorrect classifications at the Species level explains, in part, the 13.26% drop in weighted accuracy $WA_g(tr)$ from the Genus to the Species level. In addition, due to its partial classification capability, MT-MAG is able to completely classify 93.80% of the test

Table 4.3: Summary of percentages of test sequences completely classified by MT-MAG (quantified as $CR_g(tr)$) vs. classified by DeepMicrobes (quantified as CR_r), at all taxonomic ranks. A higher $CR_g(tr)$ (respectively CR_r) is better, as it signifies that a higher proportion of genomes (resp. reads) have been completely classified (resp. classified). Dash denotes not applicable.

Task ID	**Taxonomic Rank**	MT-MAG $CR_g(tr)(\%)$	DeepMicrobes $CR_r(\%)$
Task 1 (sparse)	Phylum	93.80	—
	Class	93.80	—
	Order	92.40	—
	Family	82.67	—
	Genus	71.96	—
	Species	70.65	45.02
Task 2 (dense)	Domain	96.46	—
	Phylum	83.87	—
	Class	82.98	—
	Order	81.22	—
	Family	79.67	—
	Genus	70.28	—
	Species	64.75	49.88

genomes to the Phylum and Class levels, 92.40% to the Order level, 82.67% to the Family level, and 71.96% to the Genus level, with 70.65% of the test genomes being completely classified to the Species level (see Table 4.3). In contrast, in Task 1, DeepMicrobes classifies only 45.02% of the test reads at the Species level, and it does not assess other taxonomic levels.

In Task 2 (dense) MT-MAG has an excellent performance all around, with constrained accuracies $CA_g(tr)$ above 96% at all taxonomic ranks (see Table 4.2). In addition, due to its partial classification capability, MT-MAG completely classifies 83.87% of the test genomes to the Phylum level, 82.98% to the Class level, 81.22% to the Order level, 79.67% to the Family level, and 70.29% to the Genus level, with 64.75% of the test genomes being completely classified to the Species level (see Table 4.3). In contrast, in Task 2, DeepMicrobes only classifies 49.88% of the test reads to the Species level, and does not assess other taxonomic levels.

Overall, for the two benchmarking datasets, MT-MAG completely classifies an average of 67.7% of the test sequences (to the Species level). In addition, MT-MAG provides partial classifications for the majority of the remaining sequences. This results in 93.80% of genomes analyzed in Task 1 (sparse) and 96.46% of genomes analyzed in Task 2 (dense) being partially classified or completely classified. In particular, due to its partial classification

capability, MT-MAG completely classifies on average 88.84% of the test sequences to the Phylum level, 88.39% to the Class level, 86.81% to the Order level, 81.17% to the Family level, and 71.13% to the Genus level.

4.2.2 Numerical classification confidences

In addition to the final classification path, MT-MAG also outputs numerical classification confidences along the classification path, indicating how confident MT-MAG is in the classification, at each taxonomic rank. For example, the final classification path for genome x, illustrated in Figure 4.1 is interpreted as MT-MAG being 99% confident in classifying x from "root" to "rank 1 group 1," and 90% confident in classifying x from "rank 1 group 1" to "rank 2 group 2." However, since the confidence of the latter classification is strictly less than the pre-calculated stopping threshold of 92%, this classification is deemed "uncertain" and no further classifications are attempted for genome x.

As an example of a complete classification down to the Species level, in Task 2 (dense) the final classification path for genome hRUG888 is "Domain Bacteria (confidence 97%) → Phylum Bacteroidota (confidence 97%) → Class Bacteroidia (confidence 100%) → Order Bacteroidales (confidence 100%) → Family Muribaculaceae (confidence 99%) → Genus *Sodaliphilus* (confidence 99%) → Species *Sodaliphilus* sp900314215 (confidence 99%)." As an example of a partial classification path, the final classification path for genome RUG412 is "Domain Bacteria (confidence 93%) → Phylum Bacteroidota (confidence 100%) → Class Bacteroidia (confidence 100%) → Order Bacteroidales (confidence 100%) → Family Muribaculaceae (confidence 98%) → Genus *Sodaliphilus* (confidence 99%) → Species *Sodaliphilus* sp900318645 (uncertain)." The last output means that MT-MAG is uncertain regarding its classification of RUG412 from Genus *Sodaliphilus* into Species *Sodaliphilus* sp900318645.

4.2.3 Determining the reliability of the MT-MAG classification confidences

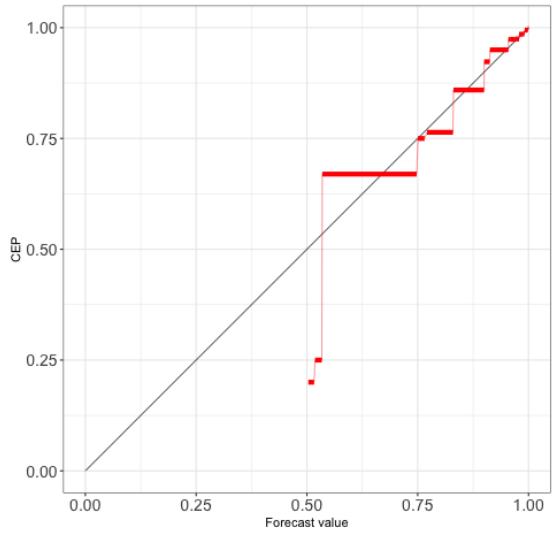
The selection of training sets is important for MT-MAG, since the composition of the training sets affects both *(i)* the parameters of the classifier that is trained on the training set and then used to classify unknown genomes, and *(ii)* the computation of stopping thresholds used to stop classifications once they become uncertain. A tool that can be used to determine whether the training sets are well selected and whether the MT-MAG classification confidences are reliable estimates is the so-called *reliability diagram* [16].

Reliability diagrams plot the observed frequency of an event against the predicted probability of that event (see Figure 4.2 for an example). In the case of taxonomic

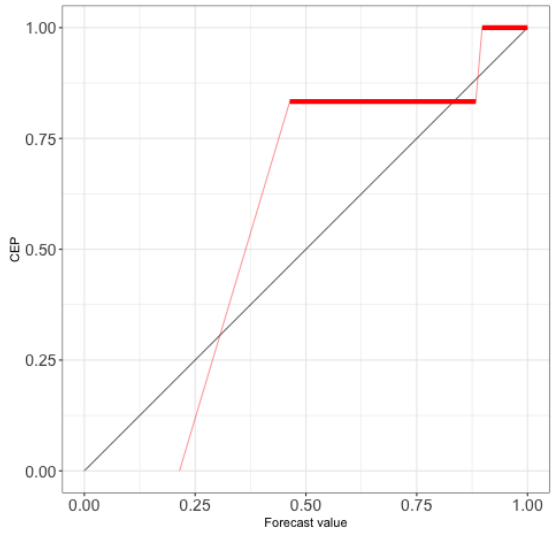
classification, the event is a match between the ground-truth and the class assignment predicted by the classifier under consideration. In a reliability diagram, the step-wise lines (red in Figure 4.2) represent the reliability curves, that is, the observed frequency of the event against the predicted probability of that event. The diagonal indicates a perfect match between the observed frequency and predicted probability. A curve that is closer to the diagonal (more specifically, where the total area formed by the curves and the diagonal is smaller) indicates a better selected training set and a more reliable classifier. Usually, a reliability diagram of a classification is accompanied by a *reliability score*, which is a number greater or equal to zero (the smaller the number, the better the reliability, with 0 being the best reliability score).

In the case of MT-MAG, the predicted probability is the classification confidence, and a reliability diagram is an indicator of the suitability of the training set and of the reliability of the classification confidences computed by MT-MAG. Consider for example the scenario whereby MT-MAG classifies a training DNA fragment to be Domain Bacteria with classification confidence 0.90. If the classification confidence matches the observed frequency, then out of all the training DNA fragments classified by MT-MAG as Domain Bacteria with classification confidence 0.90, we would expect roughly 90% to have their ground-truth taxa to be Domain Bacteria. In the reliability diagram, this match will be indicated by a point on the reliability curve that falls on or near the diagonal. If the classification confidences do not match the observed frequencies, this will also be observed on the reliability diagram, and could be an indicator that the training set is imbalanced at some parent-to-child classification (some of the child taxa have too few or too many training DNA fragments compared to the others) [49]. Note that for analyzing MT-MAG, we actually used an extended version of reliability diagrams, called *stable reliability diagrams* [10] (see Section 4.4.2 for details).

Figure 4.2 displays the reliability diagrams and reliability scores of two parent-to-child classifications for Task 2 (dense): (a) the parent taxon GTDB root into its two child taxa, Domain Bacteria and Domain Archaea, and (b) the parent taxon Family Campylobacteraceae into its five child taxa, Genus *Campylobacter*, Genus *Campylobacter_A*, Genus *Campylobacter_B*, Genus *Campylobacter_D*, and Genus *Campylobacter_E*. If we compare the two reliability diagrams, we first observe that the reliability curve for the classification of Family Campylobacteraceae deviates more from the diagonal than that for the classification of the GTDB root. Secondly, we observe that the 0.001 reliability score of the classification of the GTDB root is smaller than the 0.005 reliability score of the classification of Family Campylobacteraceae. Together, these indicators suggest that the training set of the GTDB root into its Domains is better selected, and its classification confidences are higher, than those of Family Campylobacteraceae into its five genera. Thus, for test/unknown genomes, we can expect more accurate classifications of the GTDB root to its child taxa than from Family Campylobacteraceae to its child taxa.



(a) Reliability diagrams for the classification of the GTDB root into Domains Bacteria and Archaea. Reliability score is 0.001.



(b) Reliability diagram for the classification the Family Campylobacteraceae into genera. Reliability score is 0.005.

Figure 4.2: Reliability diagrams and reliability scores (smaller is better) for the classification of *(a)* the GTDB root to its child taxa Domain Archaea and Domain Bacteria, and *(b)* Family Campylobacteraceae to its five child taxa (i.e., Genus *Campylobacter*, Genus *Campylobacter_A*, Genus *Campylobacter_B*, Genus *Campylobacter_D*, Genus *Campylobacter_E*). The larger deviation of the reliability curve (red) from the diagonal in *(b)*, and the larger reliability score of *(b)*, both indicate a lower reliability of the classification of Family Campylobacteraceae *(b)* than that of the GTDB root *(a)*.

4.3 Species level comparison of MT-MAG with DeepMicrobes

In this section, we compare the performance of MT-MAG against the performance of DeepMicrobes at the Species level, the only taxonomic rank at which DeepMicrobes classifies. The performance metrics we define here are used to assess the quality of DeepMicrobes's classification, and are defined analogously to the performance metrics for MT-MAG. The subscript r indicates that these metrics refer to reads, and the exact definitions of the terms used can be found in , "Materials: Datasets and task description". These performance metrics are:

- CA_r (constrained accuracy): the proportion of correctly classified test reads, to classified test reads.

- AA_r (absolute accuracy): the proportion of correctly classified reads, to all test reads.
- WA_r (weighted classification accuracy): the proportion of correctly classified reads. Note that in this case WA_r coincides with AA_r, since DeepMicrobes does not provide any classification at ranks other than Species. (hereafter, *weighted classification accuracy* will sometimes simply be called *weighted accuracy.*)
- CR_r (classified rate): the proportion of classified test reads to all test reads. Note the difference in the definition between $CR_g(tr)$ for MT-MAG (complete classification rate for genomes, at rank tr), and CR_r (classified rate for reads, at the Species level) for DeepMicrobes.

The three accuracies CA_r, AA_r and WA_r are numbers between 0 and 1, with $CA_r \geq AA_r$, and where higher values indicate better performance. The classified rate CR_r is a number between 0 and 1, and a higher value indicates a higher proportion of classified reads, at the Species level (for exact definitions, see Supplemental Information Section 3.3).

Since DeepMicrobes only makes classifications at the Species level, to compare its performance with that of MT-MAG, we set the parameter tr (taxonomic rank) to Species in MT-MAG, and proceeded to compare CA_r with CA_g(Species), AA_r with AA_g(Species), WA_r with WA_g(Species), and CR_r with CR_g(Species).

Of all the metrics we defined, we posit that the most informative metric for comparing MT-MAG with DeepMicrobes is the *weighted (classification) accuracy* at the Species level. Indeed, in the case of MT-MAG, WA_g(Species) combines, into a single numerical indicator, the information on the proportion of genomes that MT-MAG correctly classifies together with that of genomes that it partially classifies. In the case of DeepMicrobes, WA_r combines the information on the proportion of reads that it correctly classifies together with that of reads that it is unable to classify. In addition to this main comparison performance metric, and for a more nuanced discussion, in the following we also compare the other performance metrics, namely CA_r with CA_g(Species), AA_r with AA_g(Species), and CR_r with CR_g(Species).

Table 4.4 summarizes the MT-MAG and DeepMicrobes constrained accuracies, absolute accuracies, and *weighted accuracies*, as well as the complete classification rates of MT-MAG, respectively the classified rates of DeepMicrobes.

For Task 1 (sparse), as seen in Table 4.4, MT-MAG demonstrates significantly better overall performance than DeepMicrobes, with the weighted accuracy of MT-MAG being 39.96% higher than that of DeepMicrobes. Regarding other performance metrics, the constrained accuracy of DeepMicrobes 11.61% higher than that of MT-MAG, the absolute accuracy for MT-MAG is 15.66% higher than that of DeepMicrobes, and the complete classification rate of MT-MAG is 25.63% higher than the classified rate of DeepMicrobes. The latter indicates that MT-MAG completely classifies significantly more sequences

The metric that best captures the performance of the methods is the weighted accuracy (in blue), since this metric combines information about sequences that have been completely classified with information about the sequences that have not been completely classified to the Species level.

Table 4.4: Summary of MT-MAG and DeepMicrobes accuracy statistics, as well as the complete classification rates of MT-MAG and the classified rates of DeepMicrobes. The inputs are genomes in the case of MT-MAG, and reads in the case of DeepMicrobes. A higher value indicates better performance (in boldface).

Task ID	**Metric**	**MT-MAG(%)**	**DeepMicrobes(%)**
Task 1 (sparse)	CA_g(Species)/CA_r	81.53	**93.14**
	AA_g(Species)/AA_r	**57.60**	41.94
	WA_g(Species)/WA_r	**81.90**	41.94
	CR_g(Species)/CR_r	**70.65**	45.02
Task 2 (dense)	CA_g(Species)/CA_r	**98.63**	93.87
	AA_g(Species)/AA_r	**63.87**	46.82
	WA_g(Species)/WA_r	**78.36**	46.82
	CR_g(Species)/CR_r	**64.75**	49.88

than DeepMicrobes, though DeepMicrobes demonstrates a slightly higher constrained classification accuracy for the classified sequences.

For Task 2 (dense), as seen in Table 4.4, MT-MAG demonstrates significantly better overall performance than DeepMicrobes, with the weighted accuracy of MT-MAG being 31.54% higher than that of DeepMicrobes. Comparing the other performance metrics, the constrained accuracy of MT-MAG is 4.76% higher than that of DeepMicrobes, the absolute accuracy for MT-MAG is 17.05% higher than that of DeepMicrobes, and the complete classification rate of MT-MAG is 14.87% higher than the classified rate of DeepMicrobes. This indicates that MT-MAG not only completely classifies significantly more sequences than DeepMicrobes, but also demonstrates a slightly higher MT-MAG classification accuracy for the completely classified sequences.

Overall, for Task 1 (sparse) and Task 2 (dense), MT-MAG outperforms DeepMicrobes by an average of 35.75% in weighted accuracy. In addition, MT-MAG is able to completely classify an average of 67.7% of the sequences at the Species level, the only comparable taxonomic rank of DeepMicrobes, which only classifies 47.45%.

4.4 Result details

In this section, we discuss several details of the computational experiments. In Section 4.4.1, we introduce the formal definitions of the performance metrics used for benchmarking comparisons. Lastly, in Section 4.4.2 we discuss a significant limitation of conventional reliability diagrams, and our approach to addressing it.

4.4.1 Rigorously defined performance metrics

In the following, we will define the performance metrics for MT-MAG.

Let G denote the test set. Given a test genome $g \in G$ and a taxonomic rank tr, let $l(g, tr)$ denote the ground-truth label of g at taxonomic rank tr and let $out^{MT\text{-}MAG}(g, tr)$ denote the label computed by MT-MAG for g at taxonomic rank tr. We use "*uc*" to denote uncertain & unattempted classifications at tr, that is, if g has an uncertain or unattempted classification at tr, then $out^{MT\text{-}MAG}(g, tr)$ is "*uc*."

We denote by $G_c(tr)$ the set of genomes with complete classifications at tr (where c indicates "complete classifications") that is,

$$G_c(tr) = \{g \in G : out^{MT\text{-}MAG}(g, tr) \neq uc\}.$$

We denote by $G'_c(tr)$ the set of genomes with correct classifications down to tr, that is,

$$G'_c(tr) = \{g \in G_c(tr) : out^{MT\text{-}MAG}(g, tr) = l(g, tr)\}.$$

We define the *constrained accuracy (of classifying genomes) for taxonomic rank tr* as:

$$CA_g(tr) = \begin{cases} \dfrac{card(G'_c(tr))}{card(G_c(tr))}, & \text{if } card(G_c(tr)) \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

In other words, $CA_g(tr)$ measures how many, out of the test genomes in G with complete classifications at taxonomic rank tr, have correct classifications down to tr (See Figure 4.3).

We define the *absolute accuracy (of classifying genomes) for taxonomic rank tr* as:

$$AA_g(tr) = \frac{card(G'_c(tr))}{card(G)}.$$

In other words, $AA_g(tr)$ measures how many, out of the test genomes in G, have correct classifications down to tr (See Figure 4.3).

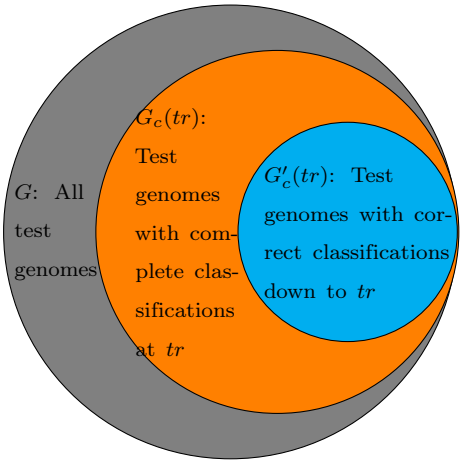


Figure 4.3: **Composition of the test set for MT-MAG.** A Venn diagram to show the relationship of three types of genomes in G (the set of all test genomes, gray circle) for taxonomic rank tr. The set $G_c(tr)$ (orange circle), of the test genomes which have complete classifications at taxonomic rank tr, is a subset of G, and the set $G'_c(tr)$ (cyan circle), of the test genomes which have correct classifications down to taxonomic rank tr, is a subset of $G_c(tr)$. Visually, we have that $CA_g(tr)$ is the ratio of the cyan circle set to the orange circle set, $AA_g(tr)$ is the ratio of the cyan circle set to the gray circle set, and $CR_g(tr)$ is the ratio of the orange circle set to the gray circle set.

For a given classification task, denote by $tr_0, tr_1, ..., tr_i$ the list of the i increasingly lower taxonomic ranks, where tr_0 is the root, and $tr_i = tr$. We define the *weighted accuracy (for the classifications of genomes) for a given taxonomic rank* tr as:

$$WA_g(tr_i) = \frac{card(G'_c(tr_i)) + \sum_{j=0}^{i-1} \frac{j}{i}(card(G'_c(tr_j) \setminus G'_c(tr_{j+1})))}{card(G)},$$

where "\" denotes set difference. The following theorem demonstrates that the definition of $WA_g(tr_i)$ is sound, by proving that for all $k \in \{0, ..., i-1\}$, the two sets $G'_c(tr_i)$ and $G'_c(tr_k) \setminus G'_c(tr_{k+1})$ form a partition of the test set (that is, the two sets are disjoint, and their union equals the test set). We begin by stating an auxiliary result, and the underlying assumption is that all genomes belong to the root, tr_0.

Lemma 4.4.1. *For all* $0 \leq k_1 < k_2 \leq i$*, we have that* $G'_c(tr_{k_2}) \subseteq G'_c(tr_{k_1})$*. Equivalently, if a genome* g *in the test set* G *has a correct classification down to* tr_k *for some* $k \in \{1, ..., i\}$*, then* g *also has correct classifications down to* tr_0*, ...,* tr_{k-1}*.*

Proof. Follows from the definition of $G'_c(tr)$. □

Theorem 4.4.2. *For every* g *in the test set* G*, one and only one of the following statements holds:*

1. *The genome g has a correct classification down to tr_i, i.e., $g \in G'_c(tr_i)$,*

2. *There uniquely exists $k \in \{0, ..., i-1\}$ such that $g \in G'_c(tr_k) \setminus G'_c(tr_{k+1})$*

   (that is, g has a correct classification down to tr_k, but does not have a correct classification down to tr_{k+1}).

Proof. Suppose case 1. holds, that is, $g \in G'_c(tr_i)$. By Lemma 4.4.1, we have that $g \in G'_c(tr_{k+1})$ for all $k \in \{0, ..., i-1\}$. Consequently, for all $k \in \{0, ..., i-1\}$, we have that $g \notin G'_c(tr_k) \setminus G'_c(tr_{k+1})$. In other words, case 2. cannot hold.
Suppose now that there exists $k \in \{0, ..., i-1\}$ such that $g \in G'_c(tr_k) \setminus G'_c(tr_{k+1})$.

We first prove that $g \notin G'_c(tr_i)$, that is, case 1. cannot hold. Since $g \notin G'_c(tr_{k+1})$, it follows that, for all $k_2 \in \{k+1, ..., i\}, g \notin G'_c(tr_{k_2})$. Thus, $g \notin G'_c(tr_i)$.

Second, we prove that $g \notin G'_c(tr_s) \setminus G'_c(tr_{s+1})$ where s is different from k.

Since $g \in G'_c(tr_k) \setminus G'_c(tr_{k+1})$, we have that $g \in G'_c(tr_k)$ and $g \notin G'_c(tr_{k+1})$. Since $g \in G'_c(tr_k)$, by Lemma 4.4.1, for all $s \in \{0, ..., k-1\}, g \in G'_c(tr_{s+1})$. Hence, for all $s \in \{0, ..., k-1\}$, we have that $g \notin G'_c(tr_s) \setminus G'_c(tr_{s+1})$. Also, for all $s \in \{k+1, ..., i\}$, we have that $g \notin G'_c(tr_s) \setminus G'_c(tr_{s+1})$. This proves the second claim.

The two claims above together prove that the sets $G'_c(tr_i)$, $G'_c(tr_0) \setminus G'_c(tr_1)$, $G'_c(tr_1) \setminus G'_c(tr_2)$, ..., $G'_c(tr_{i-1}) \setminus G'_c(tr_i)$ are all mutually disjoint.

Note now that the union of the aforementioned sets is

$$\begin{aligned}
&[G'_c(tr_0) \setminus G'_c(tr_1)] \cup [G'_c(tr_1) \setminus G'_c(tr_2)] \cup ... \cup [G'_c(tr_{i-1} \setminus G'_c(tr_i)] \cup G'_c(tr_i) \\
&= G'_c(tr_0) \\
&= G
\end{aligned}$$

This, together with the fact that the sets are mutually disjoint completes the proof of the theorem. □

Corollary 4.4.2.1. *The $(i+1)$ sets $G'_c(tr_0) \setminus G'_c(tr_1)$, $G'_c(tr_1) \setminus G'_c(tr_2)$,..., $G'_c(tr_{i-1}) \setminus G'_c(tr_i)$, and $G'_c(tr_i)$ form a partition of the test set G.*

Recall now that $WA_g(tr_i)$ is defined by a weighted summation of $card(G'_c(tr_i))$ with $card(G'_c(tr_j) \setminus G'_c(tr_{j+1}))$, for all $j \in \{0, 1, ..., i-1\}$. By Theorem 4.4.2, in this sum g contributes to $WA_g(tr_i)$ through either the cardinality of $G'_c(tr_i)$, or through the cardinality of a single one of $G'_c(tr_j) \setminus G'_c(tr_{j+1})$.

Specifically, if $g \in G'_c(tr_i)$, then g contributes 1 to $card(G'_c(tr_i))$ and does not contribute to $card(G'_c(tr_j) \setminus G'_c(tr_{j+1}))$, so the weight for g with a correct classification down to tr_i is 1.

If, on the other hand, there exists a $k \in \{0, 1, ..., i-1\}$, such that $g \in G'_c(tr_k) \setminus G'_c(tr_{k+1})$, then g does not contribute to $card(G'_c(tr_i))$, it does not contribute to $card(G'_c(tr_j) \setminus G'_c(tr_{j+1})$ for any $j \in \{0, 1, ..., k-1, k+1, ..., i-1\}$, and it contributes 1 to $card(G'_c(tr_k) \setminus G'_c(tr_{k+1}))$. Consequently, the weight of a genome g with a correct classification down to tr_k, but without a correct classification down to tr_{k+1} is $(1 \times k)/i$.

One special case to note is when g does not have a correct classification down to any taxonomic rank below the root. In this case, $g \in G'_c(tr_0) \setminus G'_c(tr_1)$, and g contributes to $card(G'_c(tr_0) \setminus G'_c(tr_1))$ by 1. Since $(1 \times 0)/i = 0$, the weight for such a genome g is 0.

Finally, we note that the formula defining $WA_g(tr)$ can be simplified to:

$$WA_g(tr) = \frac{\frac{1}{i}\sum_{j=1}^{i} card(G'_c(tr_j))}{card(G)}.$$

We also define the *complete classification rate (for genome classifications) for taxonomic rank* tr as:

$$CR_g(tr) = \frac{card(G_c(tr))}{card(G)}.$$

In other words, $CR_g(tr)$ measures how many, out of the test genomes in G, have complete classifications at tr (See Figure 4.3).

For DeepMicrobes, we define the following performance metrics, that correspond to the MT-MAG metrics $CA_g(tr), AA_g(tr), WA_g(tr)$, and $CR_g(tr)$. To this end, we first define set notations for different categories of test reads for DeepMicrobes. Using these set notations we then formally define the DeepMicrobes performance metrics.

Let R denote the test set of reads. Recall that, given a test read, the output from DeepMicrobes is either a classification of that read at the Species level, or "unclassified." Given a test read $r \in R$, let $l(r)$ denote the ground-truth Species label of r, and let $out^{DM}(r)$ denote the label computed by DeepMicrobes for this read if the read was classified, or "uc" (unclassified) if the read was not classified.

Denote by R_c the set of classified test reads, that is,

$$R_c = \{r \in R : out^{DM}(r) \neq uc\},$$

where c stands for "classified" (at the Species level).

Denote by R'_c the set of correctly classified reads, that is

$$R'_c = \{r \in R_c : out^{DM}(r) = l(r)\}.$$

We now define the *constrained accuracy (for the classification of reads)*, herein at the Species level, as:

$$CA_r = \begin{cases} \dfrac{card(R'_c)}{card(R_c)}, & \text{if } card(R_c) \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

In other words, CA_r measures how many, out of the test reads in R with a Species classification, have been correctly classified by DeepMicrobes (See Figure 4.4).

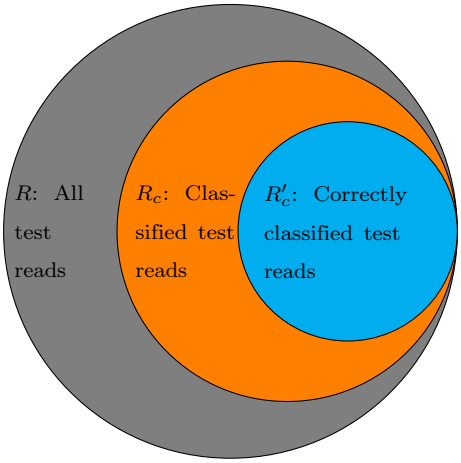


Figure 4.4: **Composition of the test set for DeepMicrobes.** A Venn diagram to show the relationship of three types of test reads in R (the set of all test reads, gray circle). The set R_c (orange circle), of classified reads, is a subset of R, and the set R'_c (cyan circle), of correctly classified test reads, is a subset of R_c. Visually, we have that CA_r is the ratio of the cyan circle set to the orange circle set, AA_r is the ratio of the cyan set to the gray set, and CR_r is the ratio of the orange circle set to the gray circle set.

We define the *absolute accuracy (for the classification of reads)*, herein at the Species level, as:

$$AA_r = \frac{card(R'_c)}{card(R)}.$$

In other words, AA_r measures how many, out of the test reads in R, have been correctly classified by DeepMicrobes (See Figure 4.4). Note that CA_r and AA_r were originally called as $\text{Precision}_{\text{read}}$ and $\text{Recall}_{\text{read}}$ in DeepMicrobes [25].

We define the *weighted accuracy (for the classification of reads)*, herein at the Species level, as being equal to the absolute accuracy AA_r. This is because we introduced weighted accuracy as a metric meant to combine the classification results of the software for completely classified sequences, with its classification results for partially classified sequences. The latter category does not exist for DeepMicrobes, as it does not provide any classification

output about ranks other than Species, hence

$$WA_r = AA_r = \frac{card(R'_c)}{card(R)}.$$

We also define the *classified rate (for the classification of reads)*, herein at the Species level, as:

$$CR_r = \frac{card(R_c)}{card(R)}.$$

In other words, CR_r measures how many, out of the test reads in R, could be classified by DeepMicrobes (See Figure 4.4).

4.4.2 Reliability diagrams

Reliability diagram were introduced as graphical diagnostic of model reliability. A reliability diagram is used to visually assess whether the probability predicted by the model for an event matches with the observed frequency of the event. In this section, we discuss a significant limitation of conventional reliability diagrams (caused by the choice of bins, as detailed below), and describe *stable reliability diagrams* proposed by [10] to overcome this limitation.

One limitation of reliability diagrams is that they are not stable, with their shape being affected by the choice of bins. Specifically, the choice of bins affects the computation of observed frequencies. Stable reliability diagrams were introduced in [10] to address this limitation, through the pool-adjacent-violators algorithm (PAVA) where PAVA is an efficient and iterative algorithm to solve monotonic regression problems. Given the predicted probabilities, ground-truth labels and classified labels, the algorithm proposed in [10] is able to generate statistically consistent, optimally binned, and reproducible reliability diagrams. In addition to the traditional reliability diagram, this method also outputs classification *confidence bands*, and a *reliability score*. The confidence band measures, if one repeats the experiment numerous times, the fraction of confidence intervals that contain the true conditional event probabilities. The reliability score measures how much the conditional event frequencies deviate from the forecast probabilities in terms of a Brier decomposition. A reliability score is a number greater than or equal to zero, with a smaller magnitude indicating higher level of reliability.

Chapter 5

Concluding Discussion and Future Work

We proposed MT-MAG, a novel *alignment-free* and *genetic marker-free* software tool that uses machine learning to obtain taxonomic assignments of metagenome-assembled genomes. This is, to the best of our knowledge, the first machine learning method for taxonomic assignment of metagenomic data that has a partial classification option, whereby MT-MAG outputs a partial classification at a higher taxonomic rank for the majority of MAGs that it could not confidently classify to the lowest taxonomic rank. In addition, MT-MAG outputs interpretable numerical classification confidences of its classifications, at each taxonomic rank. Reliability diagrams confirmed the quality of the training sets and the overall reliability of the MT-MAG classification confidences.

To assess the performance of MT-MAG, we defined a "weighted accuracy," with a weighting scheme reflecting the fact that partial classifications at different ranks are not equally informative. Compared with DeepMicrobes (the only other machine learning tool for taxonomic assignment of metagenomic data, with confidence scores), for the two datasets analyzed (genomes from human gut microbiome species, respectively bacterial and archaeal genomes assembled from cow rumen metagenomic sequences), MT-MAG outperforms DeepMicrobes by an average of 35.75% in weighted accuracy. In addition, MT-MAG is able to completely classify an average of 67.7% of the sequences at the Species level, the only comparable taxonomic rank of DeepMicrobes, which only classifies 47.45%. Moreover, a novel feature of MT-MAG is that it provides additional information for the sequences that are not completely classified at the Species level. This results in 95.15% of the genomes analyzed being either partially classified or completely classified, averaged over the two datasets analyzed. In particular, due to its partial classification capability, MT-MAG completely classifies, on average, 88.84% of the test genomes to the Phylum level, 88.39% to the Class level, 86.81% to the Order level, 81.17% to the Family level, and 71.13% to the Genus level.

Limitations of MT-MAG include the fact that, being a supervised machine learning

classification algorithm, its performance relies on the availability of ground-truth taxonomic labels for the DNA sequences in the training set. In addition, any incorrect or unstable ground-truth labels in the training set may cause erroneous future classifications. This limitation could be addressed, e.g., by extending the supervised machine learning approach to semi-supervised machine learning (where some, but not all, information about the training set is available), or even to unsupervised machine learning (where the training process does not require any ground-truth taxonomic labels, see, e.g., [30]).

Second, even though MT-MAG significantly outperforms DeepMicrobes in Task 1 (sparse training set) and Task 2 (dense training set) in weighted accuracy, there is still room for improvement in accuracies and complete classification rates. An analysis of the Task 1 (sparse) training set suggests two possible reasons contributing to incorrect classifications. One reason is the fact that the training set was the HGR database, which constitutes a very small subset of the GTDB taxonomy, in terms of both the number of representative genomes and of coverage of the GTDB taxonomy. This could be addressed by requiring a specific level of coverage for known taxa, to ensure that feature characteristics are reasonably well-represented. Another reason is the fact that, due to computational requirements of MLDSP, the training set had to exclude any contigs shorter than 5,000 bp, and this selection process resulted in the removal of 93% of the available basepairs. This could be addressed by finding ways to relax the selection criteria for the training set, to allow more sequences to participate in the training process without compromising the classification performance.

Third, the interpretability of classification could be further enhanced by exploring the last layer of the classifier. For example, the process of computing classification confidences could be used to identify pairs of child taxa that are difficult to distinguish from each other, which could potentially be biologically relevant. In addition, while single-child cases are few in the case of real DNA datasets, we note that their classification confidences are computed via a transformation of the distances between a test sequence and decision boundaries in the feature space into a valid probability distribution. To enhance the interpretability of these single-child class confidences, one could consider applying more interpretable training process and transformations such as those proposed in [15, 41].

Fourth, the classification accuracy and computational efficiency of MT-MAG could be further improved by taking advantage of user-provided information, so that the computation does not always start from the root of the taxonomy. For example, if the user already knows that an input genome belongs to Class Bacteroidia, then MT-MAG could bypass the higher taxonomic ranks and start its training and classifying phases at the Class-to-Order level directly.

Fifth, when defining the weighted accuracy for a classification at given taxonomic rank tr (i.e., $WA_g(tr_i)$), the weights used in this computation can be further refined, to reflect the dataset analyzed. Recall that the intent of defining a weighted classification accuracy was to account for the fact that partial classifications of a genome at different

ranks are not equally informative. For example, a partial classification of a genome down to the Genus level is intuitively more informative than a partial classification to, say, the Phylum level, and this is quantified as follows in the definition of weighted accuracy. The root is assigned weight 0, the last taxonomic rank with a correct classification is assigned weight 1, and intermediate taxonomic ranks are assigned weights that increase in *equal* fractional increments, from the root to the last correctly classified rank. However, this assumption of equal increments at each intermediate rank could be inadequate if, e.g., some of the intermediate taxonomic ranks are missing from the path. In such cases, the individual weights of taxonomic ranks could be defined as being different, with each weight corresponding to the amount of information that a classification at that rank contributes.

Lastly, even though MT-MAG achieves superior performance on the datasets analyzed in this paper, it would be desirable to obtain mathematical proofs of the optimality of the classifier, such as the Bayes optimality proofs in [43].

References

[1] Alexandre Almeida, Alex L Mitchell, Miguel Boland, Samuel C Forster, Gregory B Gloor, Aleksandra Tarkowska, Trevor D Lawley, and Robert D Finn. A new genomic blueprint of the human gut microbiota. *Nature*, 568(7753):499–504, 2019.

[2] Jonas S Almeida, Joao A Carrico, Antonio Maretzek, Peter A Noble, and Madilyn Fletcher. Analysis of genomic sequences by chaos game representation. *Bioinformatics*, 17(5):429–437, 2001.

[3] Rohit Babbar, Ioannis Partalas, Eric Gaussier, and Massih R Amini. On flat versus hierarchical classification in large-scale taxonomies. *Advances in neural information processing systems*, 26, 2013.

[4] C Titus Brown and Luiz Irber. Sourmash: A library for MinHash sketching of DNA. *Journal of Open Source Software*, 1(5):27, 2016.

[5] Benjamin Buchfink, Chao Xie, and Daniel H Huson. Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, 12(1):59–60, 2015.

[6] Van-Kien Bui and Chaochun Wei. CDKAM: A taxonomic classification tool using discriminative k-mers and approximate matching strategies. *BMC Bioinformatics*, 21(1):1–13, 2020.

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

[8] Pierre-Alain Chaumeil, Aaron J Mussig, Philip Hugenholtz, and Donovan H Parks. GTDB-Tk: A toolkit to classify genomes with the genome taxonomy database. *Bioinformatics*, 36(6):1925–1927, 2020.

[9] Patrick J Deschavanne, Alain Giron, Joseph Vilain, Guillaume Fagot, and Bernard Fertil. Genomic signature: Characterization and classification of species assessed by

chaos game representation of sequences. *Molecular Biology and Evolution*, 16(10):1391–1399, 1999.

[10] Timo Dimitriadis, Tilmann Gneiting, and Alexander I Jordan. Stable reliability diagrams for probabilistic classifiers. *Proceedings of the National Academy of Sciences*, 118(8), 2021.

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

[12] Raphael Eisenhofer and Laura Susan Weyrich. Assessing alignment-based taxonomic classification of ancient microbial DNA. *PeerJ*, 7:e6594, 2019.

[13] Scott Federhen. The NCBI taxonomy database. *Nucleic Acids Research*, 40(D1):D136–D143, 2012.

[14] Clémence Frioux, Dipali Singh, Tamas Korcsmaros, and Falk Hildebrand. From bag-of-genes to bag-of-genomes: Metabolic modelling of communities in the era of metagenome-assembled genomes. *Computational and Structural Biotechnology Journal*, 18:1722–1734, 2020.

[15] Jing Gao and Pang-Ning Tan. Converting output scores from outlier detection algorithms into probability estimates. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 212–221. IEEE, 2006.

[16] Holly C Hartmann, Thomas C Pagano, Soroosh Sorooshian, and Roger Bales. Confidence builders: Evaluating seasonal climate forecasts from user perspectives. *Bulletin of the American Meteorological Society*, 83(5):683–698, 2002.

[17] Weichun Huang, Leping Li, Jason R Myers, and Gabor T Marth. Art: A next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012.

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

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

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

[21] Chirag Jain, Luis M Rodriguez-R, Adam M Phillippy, Konstantinos T Konstantinidis, and Srinivas Aluru. High throughput ANI analysis of 90k prokaryotic genomes reveals clear species boundaries. *Nature Communications*, 9(1):1–8, 2018.

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

[23] Wanxin Li, Lila Kari, Yaoliang Yu, and Laura A Hug. MT-MAG: Accurate and interpretable machine learning for complete or partial taxonomic assignments of metagenome-assembled genomes. *PLoS Computational Biology*, submitted.

[24] Qiaoxing Liang. personal communication.

[25] Qiaoxing Liang, Paul W Bible, Yu Liu, Bin Zou, and Lai Wei. DeepMicrobes: Taxonomic classification for metagenomics with deep learning. *NAR Genomics and Bioinformatics*, 2(1):lqaa009, 2020.

[26] Jennifer Lu and Steven L Salzberg. Ultrafast and accurate 16s rRNA microbial community analysis using kraken 2. *Microbiome*, 8(1):1–11, 2020.

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

[28] Frederick A Matsen, Robin B Kodner, and E Virginia Armbrust. Pplacer: Linear time maximum-likelihood and bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC Bioinformatics*, 11(1):1–16, 2010.

[29] Peter Menzel, Kim Lee Ng, and Anders Krogh. Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nature Communications*, 7(1):1–9, 2016.

[30] Pablo Millán Arias, Fatemeh Alipour, Kathleen A Hill, and Lila Kari. DeLUCS: Deep learning for unsupervised clustering of DNA sequences. *Plos One*, 17(1):e0261531, 2022.

[31] Florian Mock, Fleming Kretschmer, Anton Kriese, Sebastian Böcker, and Manja Marz. BERTax: Taxonomic classification of DNA sequences with deep neural networks. *BioRxiv*, 2021.

[32] Aleksandr Morgulis, George Coulouris, Yan Raytselis, Thomas L Madden, Richa Agarwala, and Alejandro A Schäffer. Database indexing for production MegaBLAST searches. *Bioinformatics*, 24(16):1757–1764, 2008.

[33] Adithya Murali, Aniruddha Bhargava, and Erik S Wright. IDTAXA: A novel approach for accurate taxonomic classification of microbiome sequences. *Microbiome*, 6(1):1–14, 2018.

[34] Alison E Murray, John Freudenstein, Simonetta Gribaldo, Roland Hatzenpichler, Philip Hugenholtz, Peter Kämpfer, Konstantinos T Konstantinidis, Christopher E Lane, R Thane Papke, Donovan H Parks, et al. Roadmap for naming uncultivated Archaea and Bacteria. *Nature Microbiology*, 5(8):987–994, 2020.

[35] Rachid Ounit and Stefano Lonardi. Higher classification sensitivity of short metagenomic reads with CLARK-S. *Bioinformatics*, 32(24):3823–3825, 2016.

[36] Rachid Ounit, Steve Wanamaker, Timothy J Close, and Stefano Lonardi. CLARK: Fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics*, 16(1):1–13, 2015.

[37] Donovan H Parks, Maria Chuvochina, Pierre-Alain Chaumeil, Christian Rinke, Aaron J Mussig, and Philip Hugenholtz. A complete domain-to-species taxonomy for Bacteria and Archaea. *Nature Biotechnology*, 38(9):1079–1086, 2020.

[38] Donovan H Parks, Maria Chuvochina, David W Waite, Christian Rinke, Adam Skarshewski, Pierre-Alain Chaumeil, and Philip Hugenholtz. A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. *Nature Biotechnology*, 36(10):996–1004, 2018.

[39] Donovan H Parks, Michael Imelfort, Connor T Skennerton, Philip Hugenholtz, and Gene W Tyson. CheckM: Assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Research*, 25(7):1043–1055, 2015.

[40] Donovan H Parks, Christian Rinke, Maria Chuvochina, Pierre-Alain Chaumeil, Ben J Woodcroft, Paul N Evans, Philip Hugenholtz, and Gene W Tyson. Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. *Nature Microbiology*, 2(11):1533–1542, 2017.

[41] Lorenzo Perini, Vincent Vercruyssen, and Jesse Davis. Quantifying the confidence of anomaly detectors in their example-wise predictions. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 227–243. Springer, 2020.

[42] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999.

[43] Harish Ramaswamy, Ambuj Tewari, and Shivani Agarwal. Convex calibrated surrogates for hierarchical classification. In *International Conference on Machine Learning*, pages 1852–1860. PMLR, 2015.

[44] Gurjit S Randhawa, Kathleen A Hill, and Lila Kari. ML-DSP: Machine Learning with Digital Signal Processing for ultrafast, accurate, and scalable genome classification at all taxonomic levels. *BMC Genomics*, 20(1):1–21, 2019.

[45] Nicola Segata, Levi Waldron, Annalisa Ballarini, Vagheesh Narasimhan, Olivier Jousson, and Curtis Huttenhower. Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature Methods*, 9(8):811–814, 2012.

[46] Itai Sharon and Jillian F Banfield. Genomes from metagenomics. *Science*, 342(6162):1057–1058, 2013.

[47] Robert D Stewart, Marc D Auffret, Amanda Warr, Andrew H Wiser, Maximilian O Press, Kyle W Langford, Ivan Liachko, Timothy J Snelling, Richard J Dewhurst, Alan W Walker, et al. Assembly of 913 microbial genomes from metagenomic sequencing of the cow rumen. *Nature Communications*, 9(1):1–11, 2018.

[48] David Martinus Johannes Tax. *One-class classification: Concept learning in the absence of counter-examples.* PhD thesis, Technische Universiteit Delft (The Netherlands), 2002.

[49] Byron C Wallace and Issa J Dahabreh. Class probability estimates are unreliable for imbalanced data (and how to fix them). In *2012 IEEE 12th International Conference on Data Mining*, pages 695–704. IEEE, 2012.

[50] Yingwei Wang, Kathleen Hill, Shiva Singh, and Lila Kari. The spectrum of genomic signatures: From dinucleotides to chaos game representation. *Gene*, 346:173–185, 2005.

[51] Derrick E Wood and Steven L Salzberg. Kraken: Ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, 15(3):1–12, 2014.

[52] Zhuoyuan Zheng, Yunpeng Cai, and Ye Li. Oversampling method for imbalanced classification. *Computing and Informatics*, 34(5):1017–1037, 2015.

[53] Fengfeng Zhou, Victor Olman, and Ying Xu. Barcodes for genomes and applications. *BMC Bioinformatics*, 9(1):1–11, 2008.

[54] Andrzej Zielezinski, Susana Vinga, Jonas Almeida, and Wojciech M Karlowski. Alignment-free sequence comparison: Benefits, applications, and tools. *Genome Biology*, 18(1):1–17, 2017.