On the synergies between ribosomal assembly and machine learning tools for microbial identification

Stephanie Chau[1◕], Carlos Rojas[1◕], Jorjeta G. Jetcheva[1◕], Sudha Vijayakumar[1], Sophia Yuan[1], Vincent Stowbunenko[2], Amanda N. Shelton[3], William B. Andreopoulos[2‡*]

1 Department of Computer Engineering, San José State University, San José, CA, USA
2 Department of Computer Science, San José State University, San José, CA, USA
3 Department of Plant Biology, Carnegie Institution for Science, Stanford, CA, USA

◕These authors contributed equally to this work.
‡These authors also contributed equally to this work.
* william.andreopoulos@sjsu.edu

Abstract

We compare several bioinformatics approaches for the assembly and identification of 16S rRNA sequences from metagenomes: PhyloFlash, MEGAHIT and MetaSPAdes followed by Barrnap and Blast or cmsearch, as well as Mothur, UniCycler, PathRacer. We also evaluated two machine learning approaches: DNABERT and DeLUCS. We used 2 synthetic mock community datasets for our evaluation and evaluate the tools' effectiveness on genomes of varying properties: repetitiveness, GC content, genome size, and coverage. The assembly-based tools and the machine learning approaches showed complementary performance on identifying organisms varying on these four properties. PhyloFlash gave the most True Positives identifying 22/26 organisms, and missing 4 with 1 spurious hit. PhyloFlash made mistakes on 16S reconstruction for species of higher repetitiveness, low or high genome coverage or GC content, as well as communities with similar species. On the other hand, 16S reconstruction by the whole-genome assembly followed by cmsearch, Barrnap and Blast identified some of the organisms phyloFlash missed but had more spurious hits. Cmsearch following MetaSPAdes gave the best results in whole-genome assembly for 16S identification as it identified most of the species that phyloFlash missed. The ML tools, such as DeLUCS, identified most of the species in the mock community datasets that PhyloFlash missed, but missed other species of low or high GC content, extreme repetitiveness, and small genomes. A rRNA-focused assembler like PhyloFlash gave results faster and closer to the truth, as opposed to assembling the metagenome and finding the rRNAs in the assembly (with cmsearch or barrnap) that gave many spurious hits. DeLUCS and phyloFlash showed complementary performance on our data, individually identifying most organisms with few spurious hits and together identifying almost all organisms. Our insights show the tools have various strengths and weaknesses specific to the characteristics of the genomes involved.

1 Introduction

Metagenomics studies allow understanding the abundance of different species and their change between samples. In metagenomics, researchers might analyze sequencing reads to find out how the human microbiome differs between individuals with and without a

disease; other studies capture the bacterial diversity in samples of soil extracted at different time points from the same location. Analyses typically reveal many unknown microbial species. However, identifying precisely who is in the sample requires more than family- or genus-level identification. At the species- or strain-level organisms could exhibit different behavior.

Microbial genome assembly is an important tool in metagenomics research. Identification of species in metagenomics often involves studying the ribosomal 16S rRNA sequence and other marker genes. The ribosomal 16S gene is conserved and is used as a "fingerprint" that can identify the precise species or strain in a metagenomic sample. Widespread rRNA gene microbial surveys have shed light on the composition and characteristics of many ecosystems inhabited by bacteria [1–3]. Assembly algorithms are used to convert raw sequencing data into meaningful 16S sequences that can identify the organisms in a sample correctly, ideally at the strain level. While different bioinformatics tools are available, users are often unaware of limitations and biases associated with individual approaches, especially since genomes have different properties. There have been several methods for assembly reconstruction for resolving strains within a particular species in metagenomic samples. While existing metagenomic analysis methods have shown a reasonable performance at the genus and higher taxonomic levels, identifying closely related species or even strains presents a larger challenge, due to the diversity of databases, genetic relatedness, and strengths or weaknesses of different tools [4]. Selection of which metagenomic tool to employ for a specific application is often performed on a case-by-case basis. Each tool has strengths and weaknesses that affect their performance on specific tasks.

There are several broad approaches for identifying the species and strains in short-read metagenomic data: *i.*) amplicon PCR sequencing followed by a read-level clustering or statistical analysis, *ii.*) whole genome shotgun sequencing followed by assembly. In the former case, we can categorize an Operational Taxonomic Units (OTUs) when 97% similarity thresholds is met [5, 6]. In the latter, the whole genome is assembled (including 16S rRNA), which gives us the ability to identify organisms at the species- or strain-level. Both approaches are followed by taxonomic finding. Our work focuses on whole-genome sequencing by reconstructing the 16S from whole genome assemblies. Shotgun metagenomic sequencing can identify microorganisms at the species or even strain level by profiling the variants, while 16S amplicon sequencing is generally limited to identifying microbes at the genus level due to insufficient discriminatory power of the read pairs from the V3/V4 region or sequencing errors [7–9]. With shotgun sequencing reads generated in metagenomes in large numbers it is important to determine microbial composition accurately.

In this work, we focus on identification from whole-genome metagenome studies, which assemble the raw reads, rather than identifying microbes from a portion of the 16S sequence without assembling the reads. We focus on short reads, which are more affordable to produce than long reads and have a lower error rate. Long-read sequencing methods for 16S gene sequencing have higher error rates and, though the errors can be corrected with a consensus sequence analysis, the extra sequencing needed can be costly [10]. In the future, short read data is likely to remain widespread in a complementary fashion with long reads, and identifying species correctly will always pose a major challenge.

Our intention is to help scientists find the tool most suitable for the properties of the microorganisms under study. We compared several approaches for the assembly and analysis of 16S rRNA genes from metagenome sequence data, along with adapting two machine learning approaches to the task, and evaluate their effectiveness on genomes with different properties. We evaluated the approaches on: *i.*) the mock community MBARC-26 consisting of 26 genomes [11], and *ii.*) a synthetic Hot Springs Mat sample

dataset with 31 genomes of similar strains that we generated using the Camisim simulator [12] based on samples from Mushroom Spring and Octopus Spring, Yellowstone National Park [1–3, 13–16]. These datasets enable us to explore the effectiveness of different tools in the context of a range of identification challenges.

Our results show that phyloFlash performed better than the whole genome assembly tools, correctly identifying 22 species, missing 4, and labeling 1 erroneously. On the other hand, some whole-genome assembly (WGA) tools successfully reconstructed the species phyloFlash missed, but WGA produced many spurious labels. Tools such as cmsearch, MEGAHIT, and MetaSPAdes identified species that phyloFlash missed. Machine learning tools such as DeLUCS missed many species, but it succeeded in identifying organisms that the assembly-based tools missed. Our insights can serve to inform the relative strengths and weaknesses of various tools, and guide the choice of assembly tool to use with different samples.

This review is organized as follows. We describe related approaches for species identification. Then we describe the methodology with datasets used and factors considered in the analysis. Finally we discuss the results and conclusions.

2 Related Work

Below we review short-read approaches for microorganism resolution. Our discussion starts from taxonomic identification and reference-based assembly. Then we discuss machine-learning based, amplicon read clustering and statistical approaches.

Taxonomic profiling of rRNA sequences has been a popular approach for inferring the composition of complex microbial ecosystems. Taxonomy-based tools have been proposed, many of which are machine-learning based, though they generally go down to the genus level. Quantitative Insights Into Microbial Ecology (QIIME) and mothur have been widely used tools for taxonomic analysis, with MAPseq and QIIME2 being two recent alternatives. Almeida et al. compared the default classifiers of MAPseq, mothur, QIIME, and QIIME2 using synthetic simulated datasets comprised of the most abundant genera found in the human gut, ocean, and soil environments [17, 18]. QIIME2 provided the best recall and F-scores at genus and family levels. However, MAPseq showed the highest precision, with miscall rates consistently $< 2\%$. Using the SILVA database generally yielded a higher recall than Greengenes.

Prodan et al. compared six bioinformatics pipelines for 16s rRNA amplicon analysis, and explore recall vs. specificity trade-offs. Generally these pipelines make predictions at the order or phylum level [19]. ADA2 offered the best recall, at the expense of decreased specificity. USEARCH-UNOISE3 showed a good balance between recall and specificity. QIIME-UCLUST produced large number of spurious OTUs, as well as inflated diversity estimation measures.

Wang et al. introduce the RDP classifier, a naïve Bayesian classifier, which provides taxonomic assignments from domain to genus, with confidence estimates for each assignment. The majority of classifications (98%) were of high accuracy [20].

Related to our work, Fiannaca et al. explore the effectiveness of using Deep Learning models in taxonomic classification for both amplicon and shotgun short-reads based on simulated bacterial datasets [21]. The work explores the use of deep learning methods in the context of several bacterial datasets, and uses a number of factors to characterize the properties of the organisms and how that impacts the performance of traditional and machine-learning-based metagenomic assembly approaches, especially in the context of strains of the same organism. Deep Learning methods performed better than the RDP reference classifier for bacterial identification at taxonomic levels from phylum to genus with both architectures. For instance, at the genus level, deep learning reached 91.3% of accuracy, while RDP classifier obtained 83.8% with the same data.

Liang et al. [22] present DeepMicrobes, a deep learning-based computational framework for taxonomic classification that is less hindered by the lack of a taxonomic tree for newly discovered species, which is required by current metagenomics tools. They trained DeepMicrobes on genomes reconstructed from gut microbiomes and discovered potential novel signatures in inflammatory bowel diseases.

Lonear-Turukalo [23] evaluates the possibilities to identify clusters in a human microbiome based on taxonomic profiles. The results show that careful selection of the algorithm parameters and ensemble design are needed to ensure a stable clustering.

There has been considerable work on reference-based assembly with an aim to species and strain resolution, which often improves upon denovo assembly methods. These methods are generally limited by whether the exact organism's reference exists in the database. Anyansi et al. [4] present an overview of recent methods based on assembly reconstruction and methods operating with or without a reference database for resolving strains within a species in metagenomic samples. They confirm that identifying close strains in metagenomes is more challenging, due to the diversity of databases and genetic relatedness. They find that tools have different strengths and weaknesses relative to a sample that will affect performance, and propose a workflow for selecting which tool to employ considering the organism properties, such as genome complexity.

Zeng et al. [24] report reference-based ribosome assembly (RAMBL), a computational pipeline that combines a 16S rRNA reference gene database with metagenomic shotgun sequences. RAMBL reconstructs full-length 16S gene sequences from metagenomic sequencing data accurately and could identify low and high abundance microbes in mock, human and soil communities [24].

Cepeda et al. [25] present another method for reference-guided assembly of metagenomes, which reduces the computational cost of mapping reads to reference genomes through a sample-specific indexing, resolves ambiguous read mappings and improves human sample assemblies.

Besides reference-guided assembly, there has been work on improving the assembly of metagenomes using machine learning, often toward an aim to improve species and strain resolution. An early work was iMetAMOS for whole-genome assembly, an automated ensemble assembly pipeline that selects a single assembly from multiple assemblies [26]. They demonstrated the utility of the ensemble assembly on 225 previously unassembled Mycobacterium tuberculosis genomes, as well as a Rhodobacter sphaeroides benchmark dataset, and for identifying potential contamination.

Another early work was PERGA, which resolves branches in the assembly graph that are due to tandem repeats using a machine learning-based greedy-like prediction strategy. PERGA detects different copies of the repeats to resolve the branches and make the contig extensions longer, thus improving genome assembly [27].

Padovanu de Souza et al. review a variety of machine-learning methods for pre-processing of genomic datasets and post-assembly error detection [28]. They discuss application of ML during assembly, such as for identifying read overlaps prior to the construction of contigs, deciding whether to branch in de-Bruijn graphs, chimeric contigs, and to detect homologous chromosomes in diploid genomes. Greener et al. review general machine learning principles for biologists and techniques to improve genome assemblies, including deep neural networks for processing the assembly graph [29].

Quince et al. introduce STrain Resolution ON assembly Graphs (STRONG), which identifies strains de novo, from multiple metagenome samples [7]. STRONG performs coassembly and binning into metagenome assembled genomes (MAGs), and stores the coassembly graph. A Bayesian algorithm, BayesPaths, determines the number of strains present, their haplotypes and abundances. STRONG was validated using synthetic communities, showing they can generate haplotypes that match known genome

sequences.

Harris et al. compared two different machine learning approaches to distinguish the taxonomic profiling among different metagenomic samples; one is a read-based taxonomy profiling of each sample and the other is a reduced representation assembly-based method [30]. Among various machine learning techniques tested, the random forest technique showed promising results as a suitable classifier for both approaches. Random forest models developed from read-based taxonomic profiling could achieve an accuracy of 91%.

Different sequence representation methods have also been proposed, which combined with the above approaches may improve species resolution. Woloszynek et al. explore using word and sentence embedding approaches for nucleotide sequences, which may be a suitable numerical representation for downstream machine learning applications (especially deep learning). The results show that embedding sequences results in meaningful representations that can be used for exploratory analyses or for downstream applications, such as binning [31–33]. Choi et al. present all-to-all comparison of metagenomes using k-mer content and Hadoop for precise clustering [34].

Ji et al. present DNABert, a model to generate genomic DNA embeddings inspired by the state-of-the-art Natural Language Processing (NLP) approach BERT [35, 36]. Natural languages have been shown to exhibit similar properties to DNA, such as polysemy (the coexistence of many possible meanings for a word or phrase). Similarly to the way BERT is used in NLP, DNABert is a pre-trained bidirectional encoder representation based on a transformer model, which can be further tuned to a specific task using an approach called transfer learning. This enables the application of deep learning to tasks for which lower amounts of data may be available.

Unlike whole-genome sequencing assembly, 16S amplicon studies focus on sequencing of marker genes, especially 16S rRNA. Often the goal of these studies is to assess differences between conditions, as well as linking back to whole-genome assemblies for profiling purposes. These studies assess the diversity of microbial species in a metagenomic sample by focusing on V3/V4 hypervariable regions of 16S rRNA genes using primers, instead of full length gene sequencing due to the limitations of short read sequencing platforms. Due to the cost-effectiveness and widespread use of 16S amplicon sequencing there has been much research activity on analysis tools. Such amplicon studies group the individual reads into OTUs to allow species identification, which requires a preprocessing involving error correction and clustering [37]. A common pipeline for 16S amplicon analysis starts by clustering sequences within a percent sequence similarity threshold (typically 97%) into OTUs. In each OTU one sequence is selected as a representative. This representative sequence is annotated and it is applied to all other sequences within that OTU. Such studies often find it challenging to determine the number of OTUs and their relative abundance in a sample. Challenges involve identifying bacterial population boundaries and differentiating real sample diversity from sequencing errors [38]. Since their focus is a portion of the 16S gene around the V3/V4 hypervariable region, they often have genus-level granularity and cannot identify the precise species or strains.

An alternative approach for amplicon analysis is the ASV approach, which is a statistical-based species identification method that goes the opposite way of OTU clustering methods. ASVs attempt to discriminate errors from true variants based on statistical methods, then mapping the resulting variants to databases of species to find the closest species. While with OTUs there is a risk of clustering different species into the same OTU if the threshold is too lenient, with ASVs there is a risk of artificially splitting a single genome into separate clusters [39–41].

Issues with amplicon studies involve the algorithms and parameters selected and the artifacts/errors of the sequencing. In an early study, White et al. studied clustering

methods for 16S rRNA sequences, showing that algorithm selection is critical to the quality of the results [42]. They provided a critical assessment of analysis methodologies that cluster sequences into OTUs showing that small changes in algorithm parameters lead to significantly varying results.

Nguyen et al. [43] focus on the problems of using sequence similarity for defining OTUs and discuss the shortcomings of this approach. They showed that the approach of using pairwise sequence alignments to compute sequence similarity results in poorly clustered OTUs. With OTUs annotated based upon a single representative sequence, poorly clustered OTUs can have significant impact on downstream analyses.

Mahe etl al. [44] determine that amplicon clustering methods have two fundamental flaws: arbitrary global clustering thresholds, and input-order dependency induced by centroid selection. They developed Swarm to address these issues by first clustering nearly identical amplicons iteratively using a local threshold, and then by using clusters' internal structure and amplicon abundances to refine its results.

Schloss and Westcott [45] also investigate the parameters that affect the implementation of OTU-based methods. They show that it is impossible to create an accurate distance-based threshold for defining taxonomic levels and instead advocate for a consensus-based method of classifying OTUs. They propose a new heuristic that has a minimal effect on the robustness of OTUs.

Preheim et al. [38] determine two major challenges in calling OTUs: identifying bacterial population boundaries and differentiating true diversity from sequencing errors. They show that using the distribution of sequences across samples can help identify population boundaries even in erroneous data. They present a "distribution-based clustering" OTU-calling algorithm that uses the distribution of sequences across samples and demonstrate that its accuracy is competitive with other methods at grouping reads into OTUs in a mock community.

Huse et al. [46] examine deep sequencing of 16S amplicon libraries to facilitate the detection of low-abundance microbes in microbial communities. They found that a single-linkage pre-clustering methodology followed by an average-linkage clustering based on pairwise alignments more accurately predicts expected OTUs.

Edgar [47] tackles the high level of sequencing and amplification artifacts/errors found in amplified 16S sequences used to study microbial community structure. The UPARSE pipeline reports OTU sequences with $< 1\%$ incorrect bases in artificial microbial community tests, compared with $> 3\%$ incorrect bases commonly reported by other methods. The improved accuracy results in fewer OTUs, closer to the expected number of species in a community.

Hao et al. [48] proposed Bayesian clustering to tackle sequencing errors and profile the diversity of organisms, named "Clustering 16S rRNA gene sequences for OTU Prediction" (CROP). This method is useful to determine the number of OTUs and their relative abundance in a sample. CROP was robust against sequencing errors, which may provide a benefit for short and long-read sequencing methods.

Rasheed et al. [49] focused on the scalability of species diversity estimation by using locality sensitive hashing (LSH). The algorithm achieves efficiency by approximating the pairwise sequence comparison operations using hashing. They use LSH-based similarity function to cluster similar sequences and build groups of OTUs.

Besides OTU read clustering methods, there exist other statistical methods for identifying different microorganisms in 16S amplicon sequencing. Chudhary et al. [50] developed 16S Classifier using a machine learning method, Random Forest, for taxonomic classification of the short hypervariable regions of 16S rRNA sequence. On real metagenomic datasets, it showed up to 99.7% accuracy at the phylum level and up to 99.0% accuracy at the genus level. Tikhonov et al. [51] hypothesize that the standard approach to analyzing 16S sequence data, which relies on clustering reads by sequence

similarity into OTUs, under-exploits the accuracy of modern sequencing technology. They present a clustering-free approach to multi-sample Illumina datasets that can identify bacterial sub-populations, regardless of the similarity of their 16S sequences. Another approach puts sequences into bins based on their similarity to reference sequences (phylotyping), besides similarity to other sequences in the community (OTU clustering) [28, 45, 52].

There also exist statistical methods involving whole-genome shotgun sequencing of microbial communities. StrainFinder is a method that assigns strain genotypes and assesses strain frequencies for subpopulations of microbial communities with maximum likelihood estimates based on allele frequencies [53]. Ventolero et al. investigated 20 tools that attempt to infer bacterial strain genomes from shotgun read sequences [54]. They systematically evaluated six novel-strain-targeting tools on the same datasets and found that BHap, mixtureS and StrainFinder performed better than other tools. The performance of the best tools was still suboptimal though. Nayfach et al. [55] present the Metagenomic Intra-species Diversity Analysis System (MIDAS) pipeline for quantifying bacterial species abundance and strain-level genomic variation, including gene content and single-nucleotide polymorphisms (SNPs), from shotgun metagenomes. The method leverages a database of more than 30,000 bacterial reference genomes clustered into species groups. Subedi et al. propose a finite mixture of Dirichlet–multinomial regression models that accounts for the underlying group structure across gut microbiome samples and allows for a probabilistic investigation of the relationship between bacterial abundance within each inferred group [56, 57].

The results in the above work suggest that there is a need to move beyond simple clustering and assembly techniques for 16S identification and analysis. Our work expands on these by including additional tools in our analysis, both traditional bioinformatics tools and state of the art machine-learning-based tools. We included additional factors in our analysis of the genomes (e.g., coverage, repeats, GC) in order to get a deeper understanding of when different tools perform well and when they face challenges, especially when discriminating for species and strains.

3 Methodology

In this section, we describe the datasets used for our evaluation, the factors to characterize the data, and both the traditional bioinformatics tools and machine learning tools we used for organism identification. Our goal is to understand the differences in performance between the different types of tools when applied to organisms of different characteristics. We also want to identify the combination of tools that produced an improved identification, relative to using tools individually. We describe the properties of each dataset included in our study next.

3.1 Datasets

In order to explore and evaluate the performance of different metagenomic approaches for differentiating between species and strains, we utilized two datasets:

1. The mock community MBARC-26 benchmark dataset [11] with 26 microbial species.

2. A simulated dataset we created to represent the widely studied Hot Springs Mat microbial environment from Mushroom Spring in Yellowstone National Park [1, 2, 13, 14] with 31 strains.

3.1.1 *Microbial Mock Community MBARC-26 Dataset*

The mock community MBARC-26 [11], consists of 26 microbial organisms with finished reference genomes. The species contained in the dataset represent 10 phyla and 14 classes, and provide a range of GC content levels, genome sizes, and repeat content. We chose to include this dataset in our study because it represents a diverse set of genomes, is well-documented and widely used as a benchmark.

The organisms within the MBARC-26 dataset serve to provide a range of identification challenges. For example, this is visible in the Fig 1 and Fig 2 heatmaps, which shows the Blast identities for an all-to-all comparison of the 16S rRNAs of the organisms. There are organisms that are far apart in terms of BLAST ID percentage (<90%) and thus generally easier to differentiation from each other, there are also organisms (>95%) of the same genus that are difficult to disambiguate, and finally, there are organisms that are from different genera but have $90 - 95\%$ similarity in terms of 16s composition, posing a significant identification challenge for standard metagenomic pipelines. We conduct the same analysis using a Hamming Distance percentage heatmap and observe a comparable distribution of 16s similarities across the organisms (Fig 1 and Fig 2). Finally, we use t-SNE to project the 16s reference genomes onto a 2-dimensional plane (Fig 3) and observe several distinct clusters of genomes, with some genomes being very close to each other and others far apart.

3.1.2 *Synthetic Hot Springs Mat Dataset*

We created a synthetic Hot Springs Mat dataset by simulating reads from 31 microbial organisms with finished reference genomes detected in Octopus Spring and Mushroom Spring, Yellowstone National Park [1, 2, 13, 14]. This is a widely studied environment with rich biodiversity and many unknown organisms, and is believed to hold the key to understanding important functional interactions between organisms over a range of environmental conditions and time frames. Reference genomes were chosen to represent closest available relatives to genera previously found in the mat and in an unpublished metagenome (Yu et al, unpublished [14]). To simulate diversity, a number of reference genomes was chosen for some genera, including those sequenced from other hot springs (Supp.Table 1 - Hot Spring Mat Synthetic Metagenome Accessions).

The simulation data was generated with the CAMISIM simulator, which is widely used for generating synthetic genomic and metagenomic data [12]. We gave the NCBI IDs for each of the reference genomes to use as input to CAMISIM, which uses NCBI taxonomy for strain simulation and maps a reference genome to its NCBI entry using the NCBI ID. Relative abundance of the strains for the CAMISIM input (Supplemental Table 1) were made to reflect the relative abundances of genera reported in Liu et al 2011 [58] and from new datasets (SRA accession: SRP213403) (Yu et al, unpublished [14]). We simulated Illumina-generated short reads of length 150B with a mean fragment size of 1000 and a standard deviation of 100 bases. The resulting coverage is lower than for the mock community dataset due to computational limitations related to CAMISIM. The resulting simulated metagenome contained 39.8M contigs with 6GB of data.

Based on the BLAST ID percentage similarity heatmap (Fig 1, Fig 2), we observe that the synthetic dataset contains more strains (>95%) than MBARC-26, and also more organisms that have 90% similarity and are thus challenging to disambiguate. These dataset characteristics are reflected in the Hamming Distance heatmap as well (Fig 1, Fig 2). Thus, we expect that the Hot Springs Mat dataset presents a more significant challenge to organism identification methods than MBARC-26.

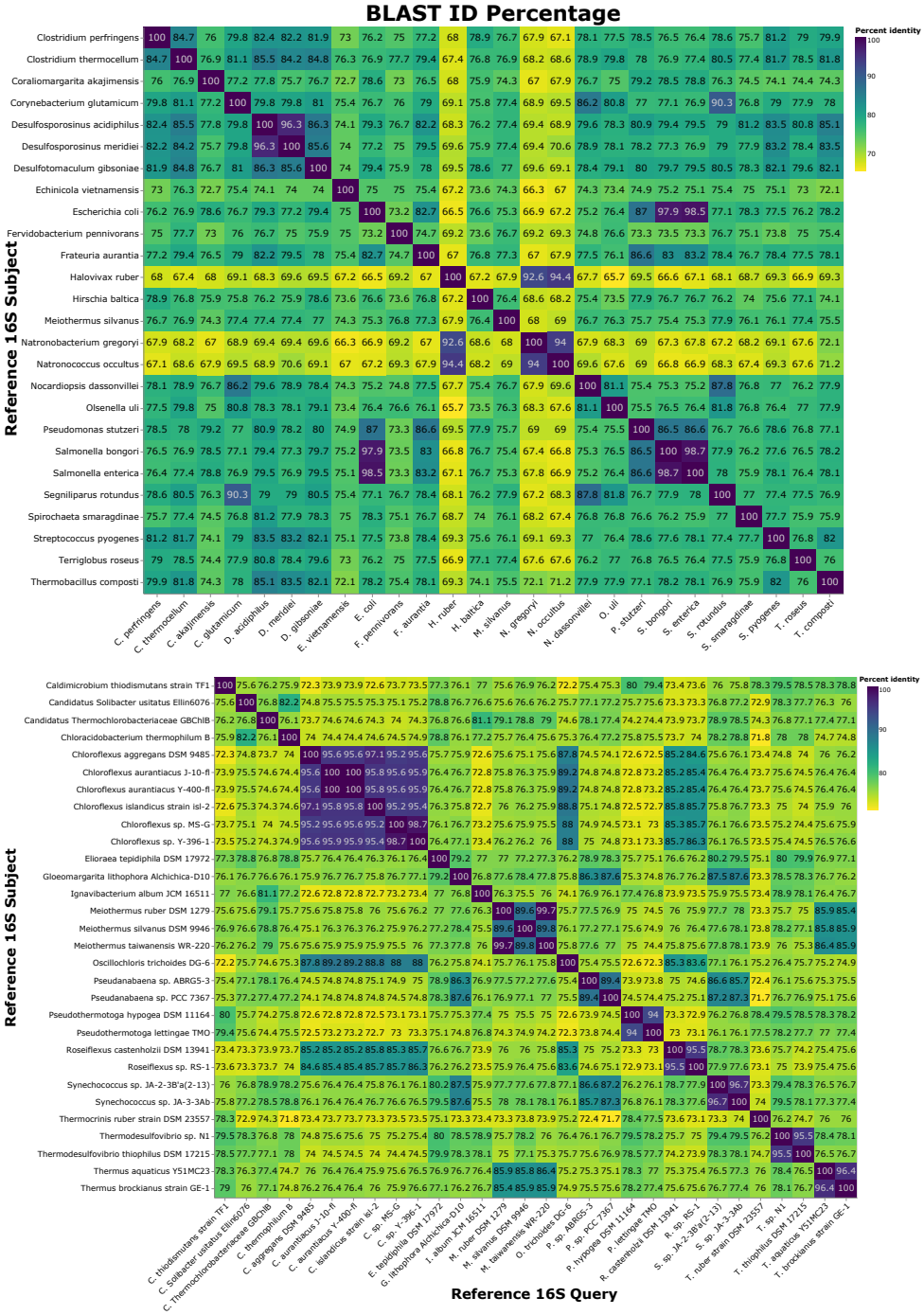

Fig 1. Blast identities between all vs. all of the 16S sequences in the MBARC-26 mock community dataset (top) and the synthetic Hot Springs Mat dataset (bottom). The matrices were generated with all-to-all Blast of the 16S sequences from the reference genomes. The similarity matrices between all of the species in the MBARC-26 mock community dataset show the diversity of the MBARC-26 dataset, but also clusters of similar species.

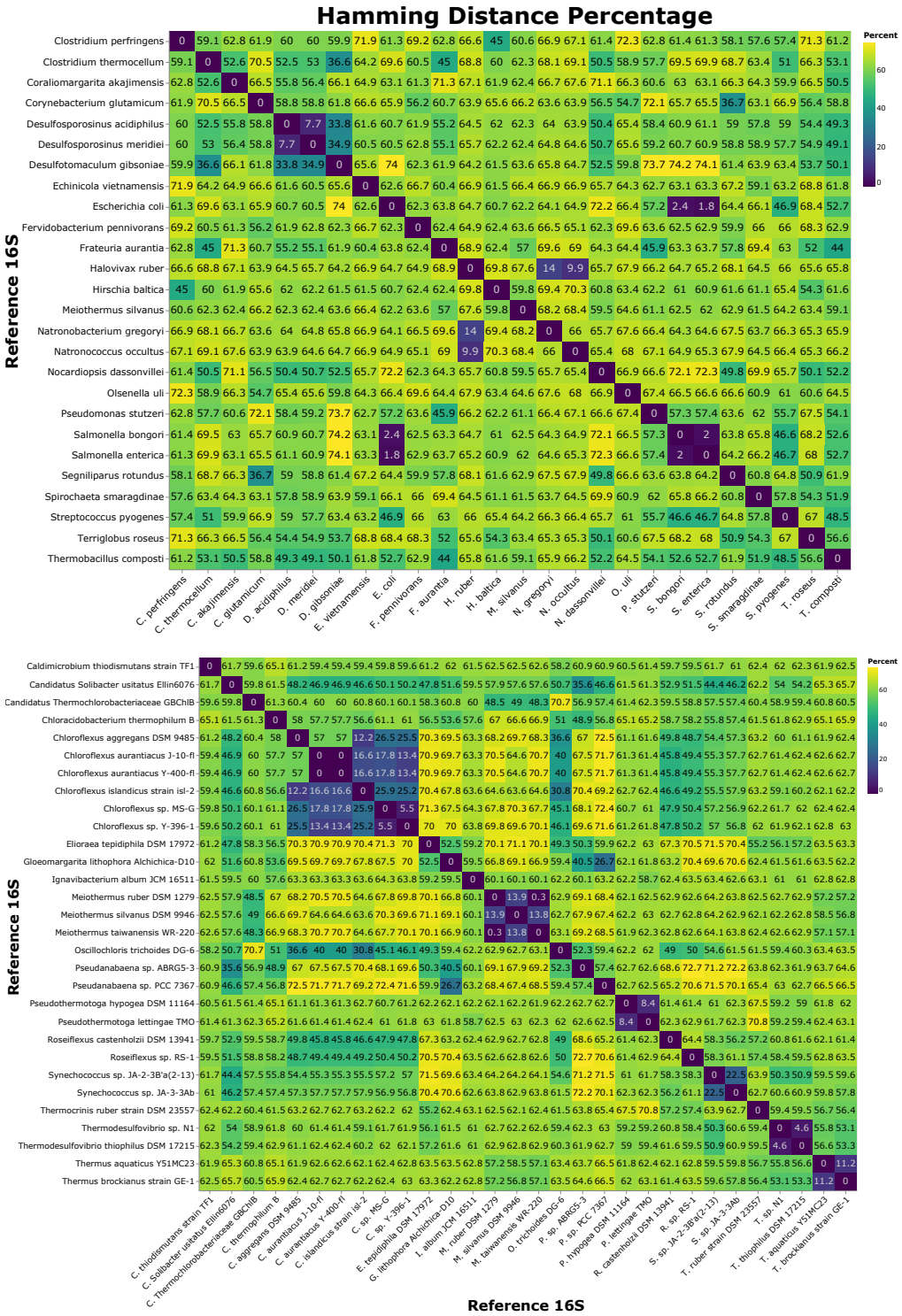

Fig 2. Hamming Distances between all vs. all of the 16S sequences in the MBARC-26 mock community dataset (top) and the synthetic Hot Springs Mat dataset (bottom). The matrices were generated with all-to-all Hamming Distances of the 16S sequences from the reference genomes. The similarity matrices between all of the species in the MBARC-26 mock community dataset show the diversity of the MBARC-26 dataset, but also clusters of similar species.

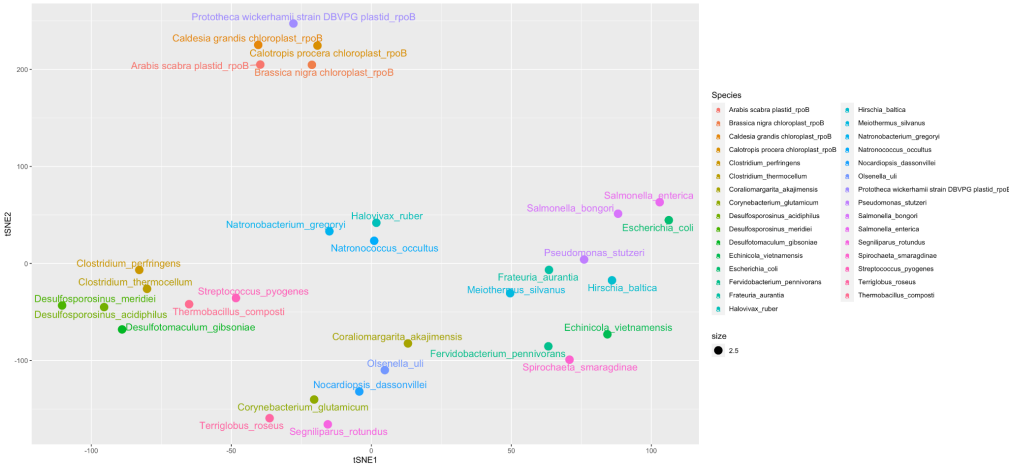

Fig 3. 2-dimensional projection of 16S MBARC sequences using t-SNE. We show several clusters of similar genomes.

3.2 Organism Characterisation

We used four factors to help characterize the organisms in our datasets: genome size, coverage, repetitiveness, and GC content. Each factor is computed for both the whole genome and the 16s sequence of each organism. We use the results to assess what types of organisms the various identification methods are able to differentiate well and which ones pose a challenge.

Genome size: The genome size is computed by counting the number of bases in the reference genome. Genome size may pose a challenge for machine learning tools that employ long segments of the reference genome for species identification in addition to or in place of reads.

Repetitiveness: Genomes are naturally very repetitive, and living organisms tend to have very similar sequences overall. Low complexity reads with high repetitiveness are considered particularly challenging as they tend to get mapped to the wrong species, e.g., during read alignment used in reference guided assembly approaches. We use the RepeatMasker tool to measure repetitiveness of the genomes in our datasets [59].

Coverage: Read coverage may also inhibit the assembly of certain species where the coverage is either too high (>150X) or too low (<30X). When the number of reads for a species is too low relative to other species in the sample, it may be difficult to assemble the sequence because of the shortage of reads needed to sufficiently map onto the reference genome. On the other hand, overly high coverage may cause de novo assemblers to throw out large groups of reads. We used the BBMap tool to measure coverage and genome size [60].

GC content: Segments of DNA with high or low GC content pose sequencing challenges to the lab instruments; consequently the data from segments of DNA with high or low GC tends to have more errors. Therefore, assembling genomic areas of abnormally high or low GC content is likely to contain mistakes.

3.3 Traditional Bioinformatics Tools

We evaluated the identification of metagenomic sequences in the context of our datasets with several state-of-the-art traditional bioinformatics assembly and identification tools, including phyloFlash [61], MetaSPAdes [62] and MEGAHIT [63]. We briefly overview each tool next.

PhyloFlash focuses on assembling the ribosomal 16S sequence, as opposed to the whole genome. It extracts all reads that match a ribosomal SSU database, assembles those reads using SPAdes or MetaSPAdes, and then performs a taxonomic assignment (using the Silva reference database). MetaSPAdes and MEGAHIT are whole-genome assembly algorithms that reconstruct longer segments of DNA (contigs) and involve a post-analysis to identify the ribosomes with cmsearch or barrnap and Blast. MetaSPAdes tends to provide more accurate results and takes into consideration the varying coverage levels within the datasets, and includes error correction, whereas MEGAHIT tends to be more efficient in terms of memory and time complexity. We used both tools to reconstruct the entire genome including the ribosomal sequence of each organism and experimented with several options for post-processing. The first option was the ribosomal database to select for Blast alignment. We experimented with both IMG [64] and Silva [65]. Another post-processing option we explored was to extract the ribosomal content with a profiler prior to Blasting. We experimented with both cmsearch and Barrnap (-outseq option), which are hidden-markov model (HMM) based profilers that detect sections of a genome sequence that match a ribosome followed by alignment to identify the correct species. We also evaluated Mothur, which compares sequences against ribosomal sequence databases Silva and RDP [65, 66] to identify the organisms [18]; as well as UniCycler [67] and PathRacer [68], which complement assembly tools to improve the result. UniCycler and PathRacer attempt to optimize the assembly graph using heuristics.

3.4 Machine Learning Tools

In this Section we describe the machine learning tools we used for our experiments, along with key parameter choices we arrived for configuring them.

We chose to explore the use of the unsupervised algorithm DeLUCS [69], which has been shown to work well on genomic classification tasks, and DNABert [35], an adaptation of state-of-the-art in Natural Language Processing deep learning and transfer learning approaches to the classification of genomic data. These algorithms are representative of a range of promising state-of-the art machine learning approaches and can help us gain insight into machine learning performance on different types of organisms, relative to traditional bioinformatics approaches.

We detail each algorithm next, along with the data pre-processing steps we used to enable the execution of our experiments.

3.4.1 *DeLUCS Algorithm*

DeLUCS is an unsupervised approach and as such does not require the input sequences to be labeled for training the model, though it requires the number of clusters specified in training. DeLUCS starts out by generating a Chaos Game Representation (CGR) graphical representation of each DNA sequence of organisms it needs to learning to classify [70]. The CGR is a two-dimensional unit square image where the intensity of each pixel represents the frequency of a particular k-mer in the sequence. The k-mers from all organisms under consideration are used to generate each CGR image in order to ensure that images are comparable. The typical k-mer size used by DeLUCS is of length 6 (six-mers). Once the CGRs for a particular dataset of reference genomes is generated, the next key step in the algorithm is to generate mutant (mimic) sequences for each original sequence in order to imitate base mutations (transversions and transitions) that may occur in real life. Pairs of CGRs, one being the CGR of the original sequence and the other being one of the mimics, are grouped in batches and used to train a set of artificial neural networks (ANNs). The difference between the original versus mutant prediction in the output is used to represent loss, which the ANN

tries to minimize. In particular, each ANN gets trained independently and has a softmax output layer, which outputs the probability that a sequence belongs to a particular class. The majority vote across the ANNs for each sequence is output as the final classification for the sequence. The taxonomic labeling and correspondence between clusters is a post-processing but essential component of DeLUCS. We used the known organism labels for our model evaluation.

We used initial parameter settings as recommended in the original DeLUCS paper, and then performed additional experiments to arrive at the best combination of parameters for our data. In particular, we used the following key parameter values, and used DeLUCS defaults for the rest:

- k parameter for k-mer construction: 6
- Modification parameter: mutation
- Number of mimics: 5-50 mimics

3.4.2 *DNABert Algorithm*

DNABert is a deep learning-based pre-trained bidirectional encoder representation that can be used for downstream genomic tasks [35]. This approach was inspired by the BERT approach where a pre-trained language representation is created by training a deep learning model on a large general text corpus (e.g., news), and can then be trained further on smaller domain specific datasets, and applied to a variety of downstream tasks. This type of approach is referred to as transfer learning because a mathematical representation built based on one dataset/process is used effectively in the context of another dataset/process. This kind of transfer learning approach has been shown to perform well and is very common in deep learning and NLP due to the high compute requirements of training models on billions of documents from scratch. We use the DNABert pre-trained model and fine tune it by further training it with our data. We started out with the parameters recommended in [35], validated the key parameter choices with our data, and selected different parameter values than those recommended as needed. We outline our parameter choices below.

- We explored k-mer sizes with k = [3,6] and found that 6-mers performed the best. We utilized 6-mers for all of the results reported in this paper.
- 12 Transformer layers, each with 768 hidden units and 12 attention heads
- AdamW optimizer
- Weight decay and output layer dropout were employed to reduce overfitting
- Each input sequence used during training is labeled with the organism to which it belongs

3.4.3 *Data Pre-Processing*

While the traditional bioinformatics tools were applied directly to raw reads, the amount of data contained in the raw reads dataset cannot be feasibly handled by a machine learning algorithm at this time. We conducted several experiments running the machine learning algorithms with samples of the raw reads (500-1000 reads per organism) but were not able to achieve reasonable performance and the tools were not able to handle a larger sample of reads.

DeLUCS has its own data pre-processing pipeline and generates k-mers directly from reference genomes. For DNABert, we generated 100- and 500- base pair shreds from the

	MEGAHIT (Hits)	MEGAHIT (TP)	MetaSPAdes (Hits)	MetaSPAdes (TP)	phyloFlash (Hits)	phyloFlash (TP)
phyloFlash Default config.	N/A	N/A	N/A	N/A	23	22 (4 missing)
cmsearch with Rfam 13.0	186	20	71	19	26	18
Blast to IMG	28	17	25	10	25	21
Barrnap –outseq, Blast with IMG	28	17	27	15	25	21

	MEGAHIT (Hits)	MetaSPAdes (Hits)	phyloFlash (Hits)
Barrnap –outseq	17 Complete & 21 Partial	18 Complete & 8 Partial	45 Complete & 1 Partial
Blast to SILVA	1540	451	N/A
Barrnap –outseq, Blast with SILVA	1540	192	N/A

Table 1. A comparison of both whole-genome and 16S assembly tools on the mock community MBARC-26. PhyloFlash achieved the most true positives (TPs) and few spurious hits that were not TPs.

reference genome of each organism in our datasets, and used those to represent reads when using DNABert for classification [35]. We used 70% of the shreds for training and 30% for testing.

4 Results

In this section, we overview the results of the factor analysis of the organisms in our datasets and evaluate the performance of the traditional microorganism identification tools, along with the classification accuracy of our machine learning tools.

4.1 Characterization of Genomes under Consideration

Both the MBARC-26 and Hot Springs Mat mock communities provide a diverse range of features as surfaced by the four factors we used in our study (coverage, repetitiveness, GC content and genome size) and thus provide a range of challenges for the assembly and identification approaches we explore. Fig 6 to Fig-9 show the values for each factor for both datasets, both for the whole genomes and 16s rRNA.

One notable difference between the two datasets is the higher repetitiveness in the Hot Springs Mat dataset: up to 170 times higher between the organism with highest repetitiveness in MBARC-26 and the Hot Springs Mat dataset. Also, the coverage is lower for the Hot Springs Mat dataset, because of simulator limitations. We have a range of similarities amongst the genomes, as shown in the Fig 4 and Fig 5 heatmaps. The similarities and differences between genomes in the datasets provide different challenges for each of the tools we tested.

4.2 Metagenome Assembly Tool Performance

We additionally tested PathRacer, Mothur, and Unicycler, which missed most of the organisms at the species level. As a result, we decided to focus on phyloFlash, MEGAHIT, and MetaSPAdes, since they performed the best amongst the traditional assembly tools we evaluated. Next we examine the results for each dataset and how their differences impacted the results of our tool evaluation.

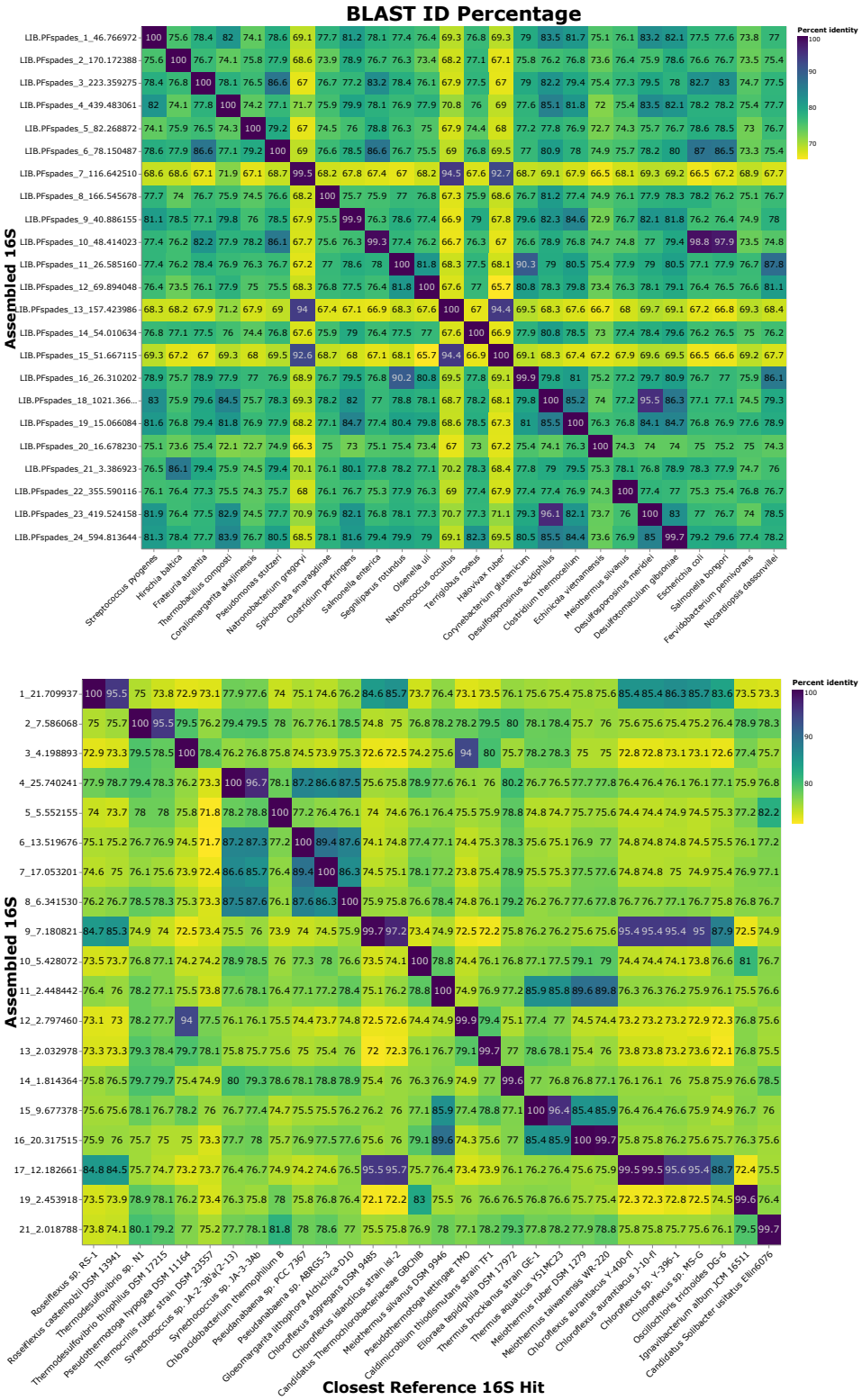

Fig 4. Blast identities of assembled vs. reference 16Ss for the MBARC-26 mock community (top) and the Hot Springs Mat Synthetic dataset (bottom). The matrices were generated with Blast of all assembled to all 16S reference sequences.

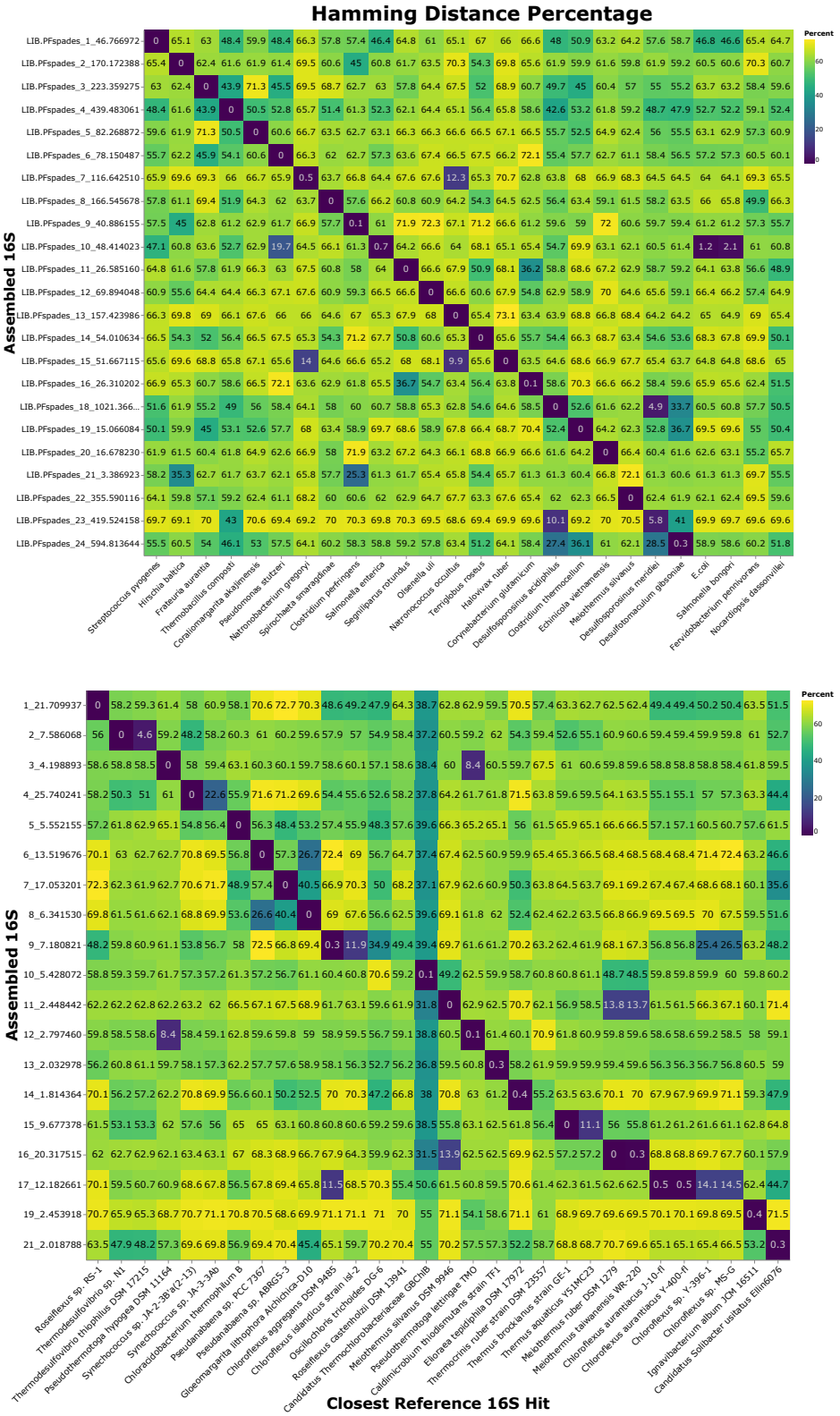

Fig 5. Hamming Distances of assembled vs. reference 16Ss for the MBARC-26 mock community (top) and the Hot Springs Mat Synthetic dataset (bottom). The matrices were generated with Hamming Distances of all assembled to all 16S reference sequences.

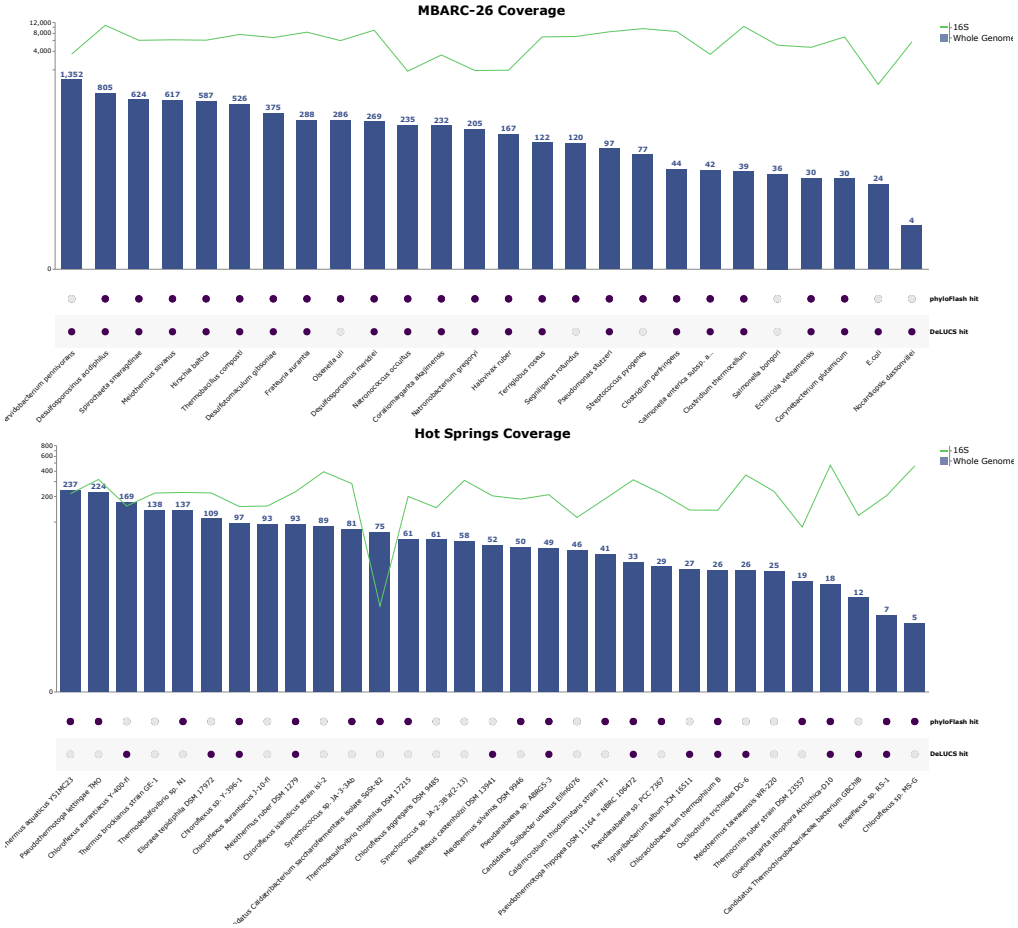

Fig 6. The coverage of each organism in the MBARC-26 mock community (top) vs. the synthetic Hot Springs (bottom). The dotplot indicates which organisms phyloFlash vs. DeLUCS (on whole genome) missed or identified. The assembly-based algorithms perform worse when coverage is too low or high. F.pennivorans was missed by phyloFlash due to high coverage, while N.dassonvillei, E.coli and S.bongori due to low coverage (N.dassonvillei also had a large coverage difference between the genome and 16S). Three of these except S.bongori were identified by DeLUCS, though it missed 3 other genomes with normal coverage but extreme GC (S.rotundus, O.uli, S.pyogenes). On the Hot Springs dataset phyloFlash identified 18 and DeLucs 13 organisms, but together they identified 24 out of 31 organisms.

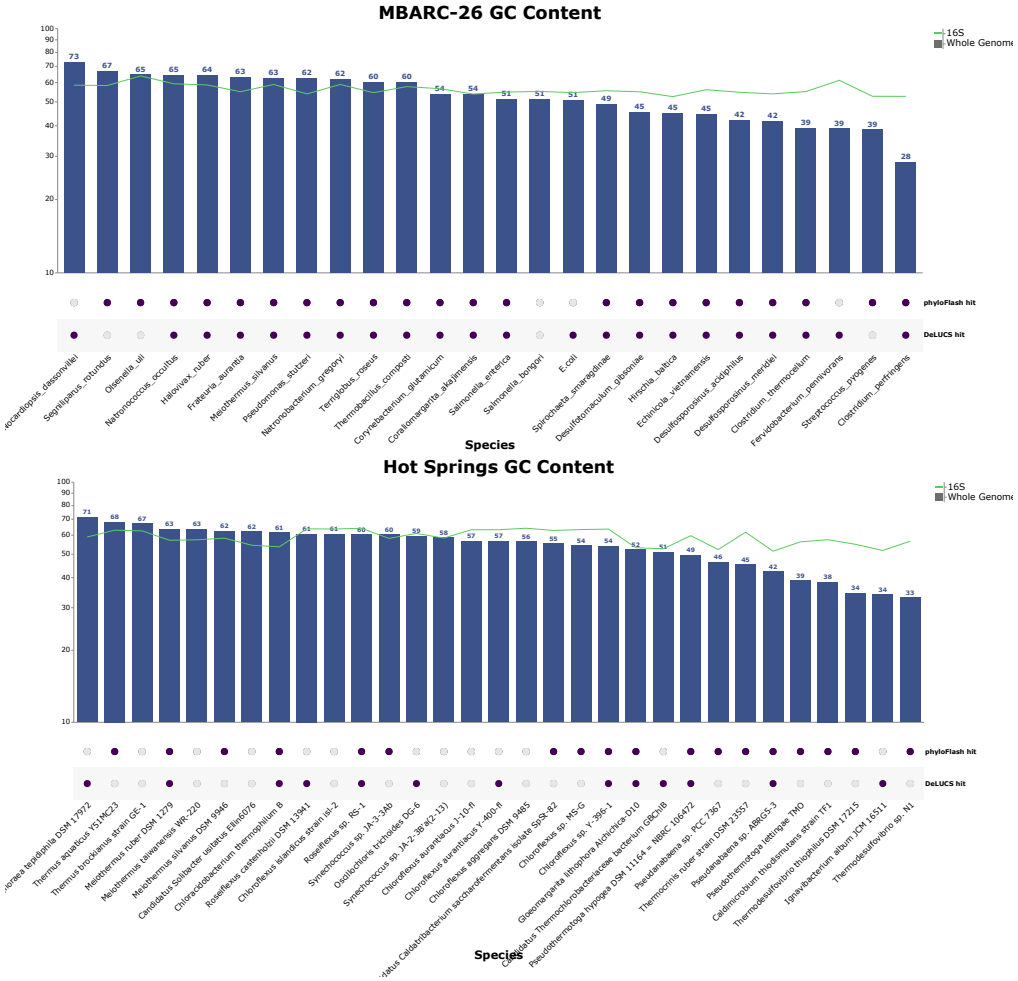

Fig 7. The GC content of each organism in the MBARC-26 mock community (top) vs. the synthetic Hot Springs (bottom). The dotplot indicates which organisms phyloFlash vs. DeLUCS (on whole genome) missed or identified. Genomes of high or low GC content tend to be missed by either tool. N.dassonvillei and F.pennivorans that have high or low GC content were missed by phyloFlash, but identified by DeLUCS. DeLUCS on the other hand missed three other genomes of high or low GC content (S.rotundus, O.uli, S.pyogenes).

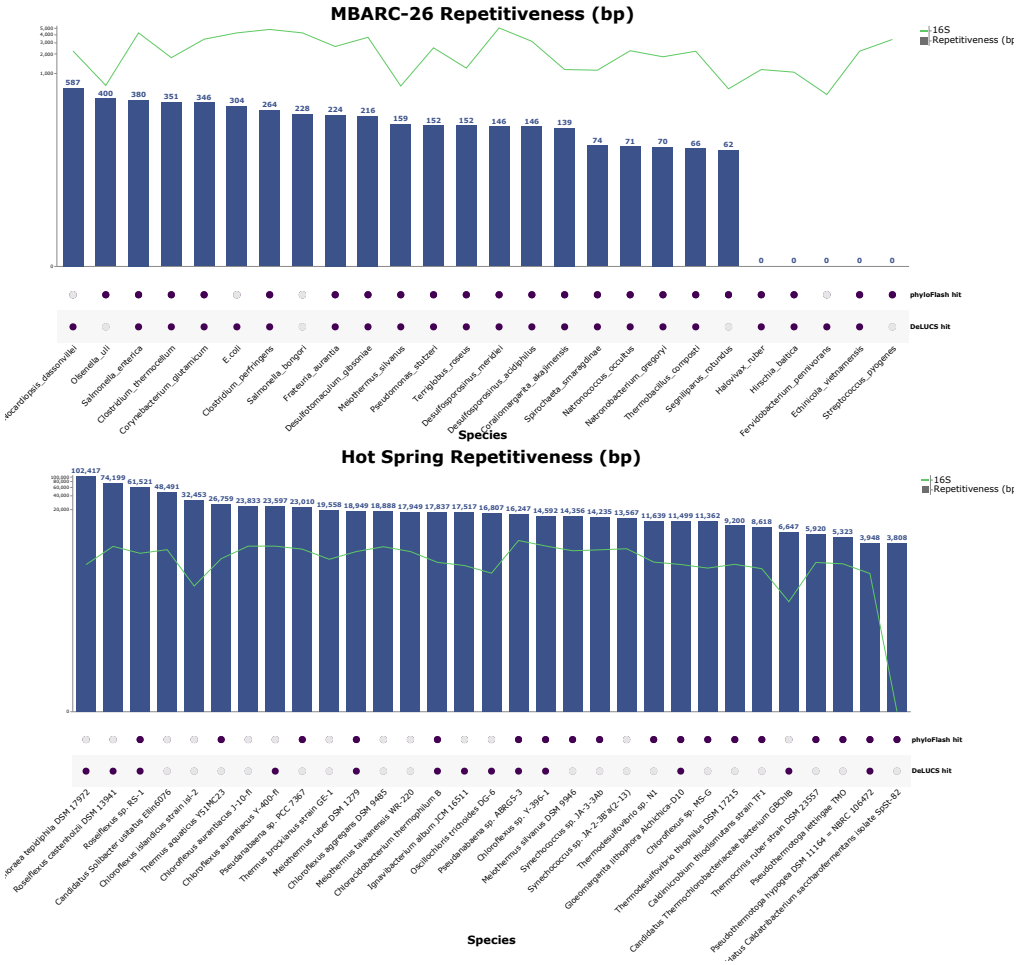

Fig 8. The repetitiveness of each organism in the MBARC-26 mock community (top) vs. the synthetic Hot Springs (bottom). The dotplot indicates which organisms phyloFlash vs. DeLUCS (on whole genome) missed or identified. Both tools have a better chance of species identification on genomes that are not extremely repetitive. On MBARC-26 N.dassonvillei was missed by phyloFlash due to high repeat content, but identified by DeLUCS, which instead missed the repetitive O.uli.

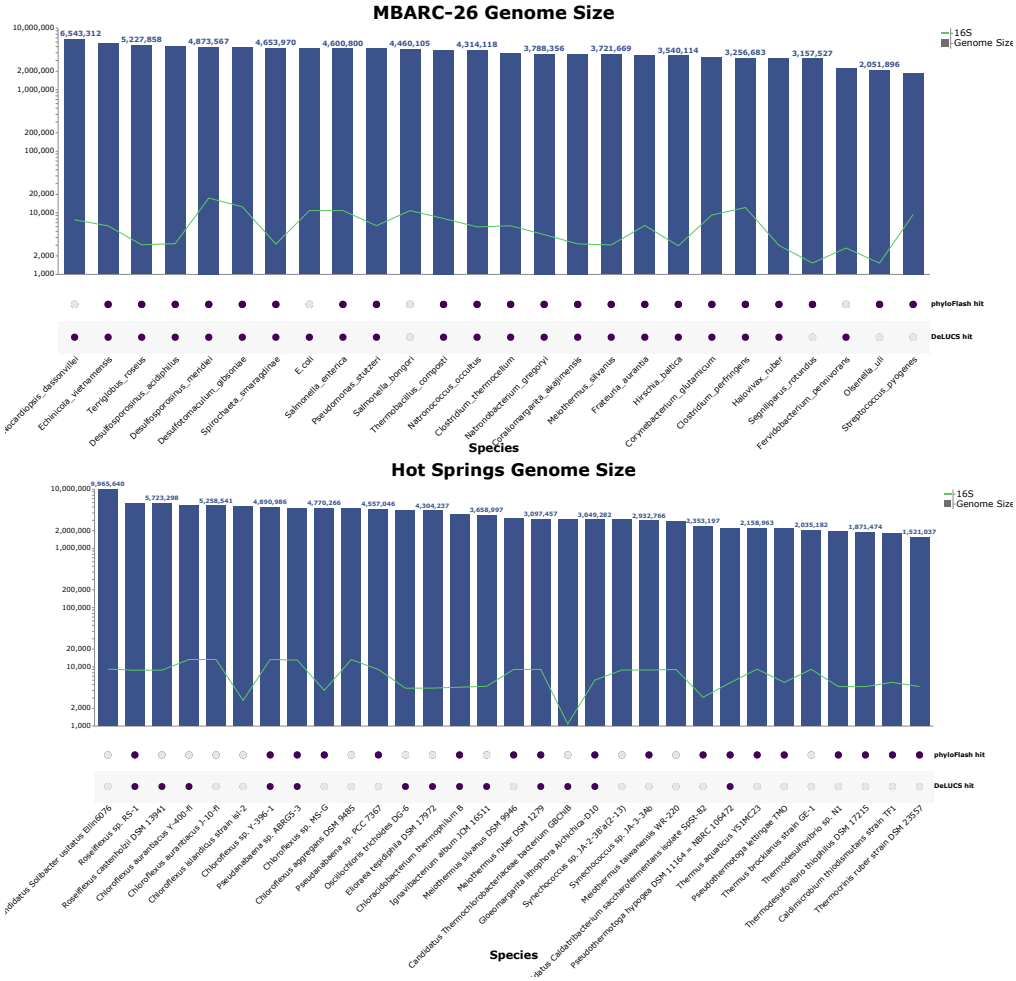

Fig 9. The genome size of each organism in the MBARC-26 mock community (top) vs. the synthetic Hot Springs (bottom). The dotplot indicates which organisms phyloFlash vs. DeLUCS (on whole genome) missed or identified. DeLUCS performs worse on small genome sizes.

4.2.1 *Evaluation on MBARC-26 Mock Community*

The results of applying phyloFlash, MEGAHIT and MetaSPAdes on MBARC-26 are summarized in Table 1 in terms of true positive (TP) and spurious hits for each of the three tools. Many of the metagenome assembly tools we tested produced many spurious hits. On the mock community dataset, both MEGAHIT and MetaSPAdes followed by Barrnap with Blast to Silva resulted in many spurious hits, while using Blast to IMG resulted in fewer spurious hits but also missed many organisms. Overall, cmsearch performed best after MetaSPAdes and MEGAHIT as it resulted in fewer spurious hits, and slightly more correct hits (TPs) than the other choices.

Our results show that the rRNA-focused assembler phyloFlash performed better overall, giving more accurate results within a shorter amount of time, relative to other approaches where finding the rRNA is performed after the metagenome assembly. Without any specific fine tuning relative to its default settings, phyloFlash was able to correctly identify 22 out of the 26 species in the mock community dataset. PhyloFlash followed by either Barrnap or cmsearch and Blast to IMG produced 4 spurious hits out of 25 hits, and identified 21 species, showing no need for a post-analysis to improve species identification.

On the MBARC-26 mock community phyloFlash missed 4 species that did not get assembled: Fervidobacterium pennivorans, Nocardiopsis dassonvillei, Escherichia coli, and Salmonella bongori. These species have the following characteristics, according to the factor plots. N.dassonvillei is characterized by high GC content, high repeat content, low genome coverage but high 16S coverage. F.pennivorans is characterized by high coverage (though a normalization could result in its proper identification). E.coli and S.bongori both have low genome coverage but high 16S coverage and are genomically similar to S.Enterica (Fig 4 and Fig 5) making their 16S reconstruction challenging.

Explanations for phyloFlash's failure to reconstruct 4 species are either that it did not assemble 16Ss at all, or it assembled but 16S was not identified at a later stage. To determine the answer we BLASTed the results of all phyloFlash steps and found that these 4 species were not assembled and not detected at any stage. Next, in order to determine if phyloFlash failed during the mapping stage with BBmap or during the assembly stage with SPAdes, phyloFlash was rerun using reads from the 4 missing species. The reads were obtained by using BBsplit labeling each of the reads from the mock community dataset with the references for the 26 species. None of the 4 species were correctly assembled by phyloFlash, though the tool was able to execute the entire pipeline without crashing and without errors. Although there were no assemblies, there were reads mapped to phyloFlash's reference database, showing that the issues stemmed from the SPAdes assembler used by phyloFlash.

When looking at the factor plots (Fig 6), we note that of the 26 species in the dataset, N. dassonvillei had the lowest coverage according to analysis using BBmap. The coverage for E. coli and S. bongori was also lower than average. On the other hand, all of their 16S sequences had high coverage as expected. The F. pennivorans genome had the highest coverage of all 26 species.

In the next experiment we ran phyloFlash on the whole dataset using MetaSPAdes instead of SPAdes with the idea that MetaSPAdes would be better suited to assemble metagenomic data. While phyloFlash with SPAdes was unable to assemble F. pennivorans due to the species's outlying high coverage, phyloFlash with MetaSPAdes was able to recover this species in the final assembly. In this experiment, F. pennivorans was assembled successfully but the other 3 species remained missing. Among the 26 species in the dataset, N.dassonvillei had significantly lower coverage and PhyloFlash was unable to map the few reads to the reference database, regardless of whether SPAdes or MetaSPAdes was used.

The phyloFlash script with MetaSPAdes was then run just on the reads belonging to

E. coli, S. bongori, and N. dassonvillei. Unlike in a previous experiment with SPAdes, phyloFlash was able to assemble E.coli. After removing the reads for E. coli and rerunning the experiment, S. bongori was assembled. However, after removing the reads for both E. coli and S. bongori, phyloFlash was still unable to assemble N. dassonvillei. The N. dassonvillei reconstruction was still unsuccessful due to the low coverage. According to the reads phyloFlash extracted, only 3 out of the 399 reads for N. dassonvillei were mapped to the reference database. This low coverage caused phyloFlash to abort before the assembly stage, suggesting that N. dassonvillei's low coverage is the problematic factor. Without sufficient alignment, the assembly phase of the pipeline did not have enough reads to reconstruct the whole sequence. Compared to SPAdes, MetaSPAdes may perform better when there is a wide variance in the coverage of organisms as found in metagenomes, though a low read coverage can still cause issues.

PhyloFlash could not assemble S. bongori and E. coli when both are present either. The phyloFlash report suggested the presence of low complexity repetitive sequences. According to the phyloFlash report, low complexity (repetitive) reads tend to map to the same reference even if the reads did not match the species. Therefore, if there are low complexity sequences in the data, some sequences may be mis-assembled. The tool Repeat Masker was used to check for unusually repetitive sequences. In the results, neither the whole genome nor the 16S sequences for any of the 4 missing species proved to be abnormally repetitive. Given these results, repetitiveness was unlikely to be the sole factor causing phyloFlash to fail for E. coli and S. bongori, but their low coverage was more likely an issue.

Assembling species that are very similar to one another is known to be especially challenging. Two species are very similar if they belong to the same genus and have 16s rRNA sequences of 97% similarity or higher. High similarity between DNA sequences within a genus may challenge an assembler's ability to accurately differentiate between distinct species or strains. Of the 4 species missed by phyloFlash, S. bongori and N. gregoryi are around 97% similar to S. enterica and H. ruber respectively. This level of similarity is at the threshold at which sequences may be considered as being from the same species. When MetaSPAdes replaced SPAdes, N. gregoryi was still missing. When combined with the reads for Halovivax ruber and Natronococus occultus, the 2 species most similar to N. gregoryi out of the 26 species, N. gregoryi was not assembled while the other 2 were.

Variations in genome size did not seem to negatively affect phyloFlash's performance either. The 4 missing species had varying sequence lengths, and the species that were successfully assembled also included a variety of lengths.

Even though phyloFlash performed better than the other tools, the whole-genome assembly tools successfully reconstructed the species that phyloFlash could not reconstruct. This result can be seen in MEGAHIT and MetaSPAdes followed by cmsearch (and BLAST to IMG) that identified the species S.bongori, E.coli and F.pennivorans that phyloFlash missed. Table 2 marks the missing species identified by cmsearch, for each output (MEGAHIT, metaSPAdes, and phyloFlash). This suggests that ensembling phyloFlash with some of the other tools may result in an improved overall reconstruction and identification. However, creating this kind of assembly may not be straightforward as cmsearch produced a large number of spurious hits in addition to having a high number of true positives (see Table 1).

4.2.2 *Evaluation on Hot Springs Mat Dataset*

Results on the synthetic Hot Springs Mat dataset show that phyloFlash performed worse than on the mock community dataset (Table 3). In particular, phyloFlash had 18 unique hits that matched one of the 31 organisms (true positives) and 17 spurious unique hits that did not match any of the 31 organisms. In this dataset, phyloFlash

Species	MEGAHIT	metaSPAdes	phyloFlash
S.bongori	✓	-	-
E.coli	✓	✓	-
F.pennivorans	✓	✓	-
N.dassonvillei	-	-	-

Table 2. Missing species identified by cmsearch

	BLAST to IMG		BLAST to Silva	
	TP	hits	TP	hits
metaSPAdes	22	592	0	4669
MEGAHIT	14	19	0	260
phyloFlash	-	-	18	35

Table 3. Results on the synthetic Hot Springs Mat dataset.

misassembled organisms that have a higher repetitiveness and higher GC content.

MEGAHIT and MetaSPAdes produced similar spurious hits on the Hot Springs Mat dataset as on the MBARC-26 mock community dataset. In particular, MetaSPAdes with Blast to IMG had 570 spurious hits and MEGAHIT 5 spurious hits, though with Blast to Silva there were significantly more.

The Hamming Distances and Blast identities indicated that the reconstructed 16Ss by phyloFlash were accurate at identifying the ground truth organisms – the Hamming Distances and Blast identities show a close similarity of the assembled 16Ss to the true organisms' ribosomes (Fig 1, Fig 2) for both the MBARC-26 mock community and synthetic Hot Springs Mat.

Overall, we found that an rRNA-focused assembler like phyloFlash gave results both faster and closer to the truth, as opposed to using either Barrnap or cmsearch after the metagenomic assembly to find the rRNA.

4.3 Machine Learning Performance

In this section, we present the results of applying DeLUCS [69] and DNABert [19] to our microorganism identification challenge in the context of the MBARC-26 mock community and our synthetic Hot Springs datasets. We described the data pre-processing steps executed by DeLUCS and DNABert in the Methodology Section. In the case of DeLUCS, we first ran its default pipeline, which takes as input the whole reference genomes, and then conducted an experiment using just the 16S rRNAs extracted with Barrnap in order to evaluate its ability to classify based on ribosomal sequences.

4.3.1 *DeLUCS Evaluation on MBARC Dataset*

Our initial experiments focus on using DeLUCS with its default input format – whole reference genomes which it converts into k-mers, which are then used to generate CGR representations. In addition, one of the key parameters configured for DeLUCS is the number of mimics to generate during the training process – more mimics create more sequence variations to expose the algorithm to but lead to higher computational overhead. To determine a reasonable setting for this parameter, we ran DeLUCS on the MBARC dataset for 3-26 mimics (Fig 10) and found that using 5 mimics provides good performance (89% accuracy) at a reasonable computation load. The DeLUCS experiments in this section are based on using 5 mimics.

DeLUCS achieved an accuracy of 89% on the MBARC dataset, with 4 incorrectly predicted species: O.uli, S.rotundus, S.pyogenes, and S.bongori. O.uli and S.rotundus

were falsely predicted to be N.dassonvillei, S.bongori was falsely predicted to be E.coli, and S.pyogenes was falsely predicted to be D.gibsoniae.

In terms of organism identification, we note that DeLUCS and phyloFlash have complementary performance in that DeLUCS is able to identify organisms that phyloFlash missed and vice versa. On the MBARC-26 mock community they each identified 22 out of 26 species and together they identified all but 1 organism (S.bongori). To better understand when one tool works better relative to the other, we look at their performance in the context of the factors we introduced in the Methodology section.

We find that phyloFlash has challenges identifying organisms of high or low coverage in the sample and with repetitive genomes (Figs 6 to 9). Coverage does not play a role in the DeLUCS results because we applied it on the post-assembled contigs or references, rather than reads, and the default pipeline generates k-mers from the input sequence uniformly. DeLUCS mainly faces challenges in identifying organisms with small genome sizes and either high or low GC content. There is one organism that presents a challenge to both tools, and that is Salmonella bongori. The likely reason is that the tools are unable to differentiate it from the other Salmonella enterica strain in the dataset.

4.3.2 *DeLUCS Evaluation on Hot Springs Mat Dataset*

Overall, DELUCS had a lower performance on the Hot Springs Mat dataset than on the MBARC dataset. DeLUCS and phyloFlash have complementary performance on the Hot Springs dataset, as well, where phyloFlash identified 18 and DeLucs 13 organisms, but together they identified 24 out of 31 organisms. The average accuracy when incrementing over setting the number of mimics between 3 and 26 was 76% (Fig 10). The scores were mostly between $71 - 81\%$ with the exception of using 3 mimics per sequence, which resulted in 64% accuracy. The upper half of the mimics (using 14-26 mimics) performed better than the lower half (all experiments resulted in $\geq 75\%$ accuracy). Each additional mimic added around 2-4 minutes of time to the experiment on our infrastructure. We used 5 mimics for the results shown in the factor plots (Fig 6 to 9).

While phyloFlash has trouble identifying organisms with low or high coverage and high repetitiveness, DeLUCS faced challenges on low or high GC content genomes, and small genomes, on both synthetic Hot Springs Mat and MBARC-26 data. It is important to note that repetitiveness in the Hot Springs Mat dataset is much higher (Fig 7) than in the MBARC dataset. DELUCS is able to identify organisms with high repetitiveness in the Hot Springs dataset, showing a complementary behavior to Phyloflash that missed those organisms. There are a number of organisms that both tools are unable to identify, and they tend to be on the higher GC content and higher repetitiveness side.

4.3.3 *DeLUCS Performance on rRNA from Assembled Contigs*

To evaluate DeLUCS ability to identify species using the 16Ss rRNA and compare to phyloFlash, we ran DeLUCS on the Barrnap ribosomal output for the MetaSPAdes assembled contigs. We were not able to run it on the entire raw reads dataset due to computational constraints (it could only handle 500 reads per organism which resulted in only 15% classification accuracy). For rRNAs extracted with Barrnap on the MBARC mock community, the accuracy was about 44% for 10 mimics. For the Hot Springs Mat dataset, the accuracy was around 56% for 10 mimics. For MetaSPAdes contigs before ribosome extraction the classification accuracy was significantly lower at 25% for 3 mimics. The results support that DeLUCS has a potential to be applied as a complement to phyloFlash.

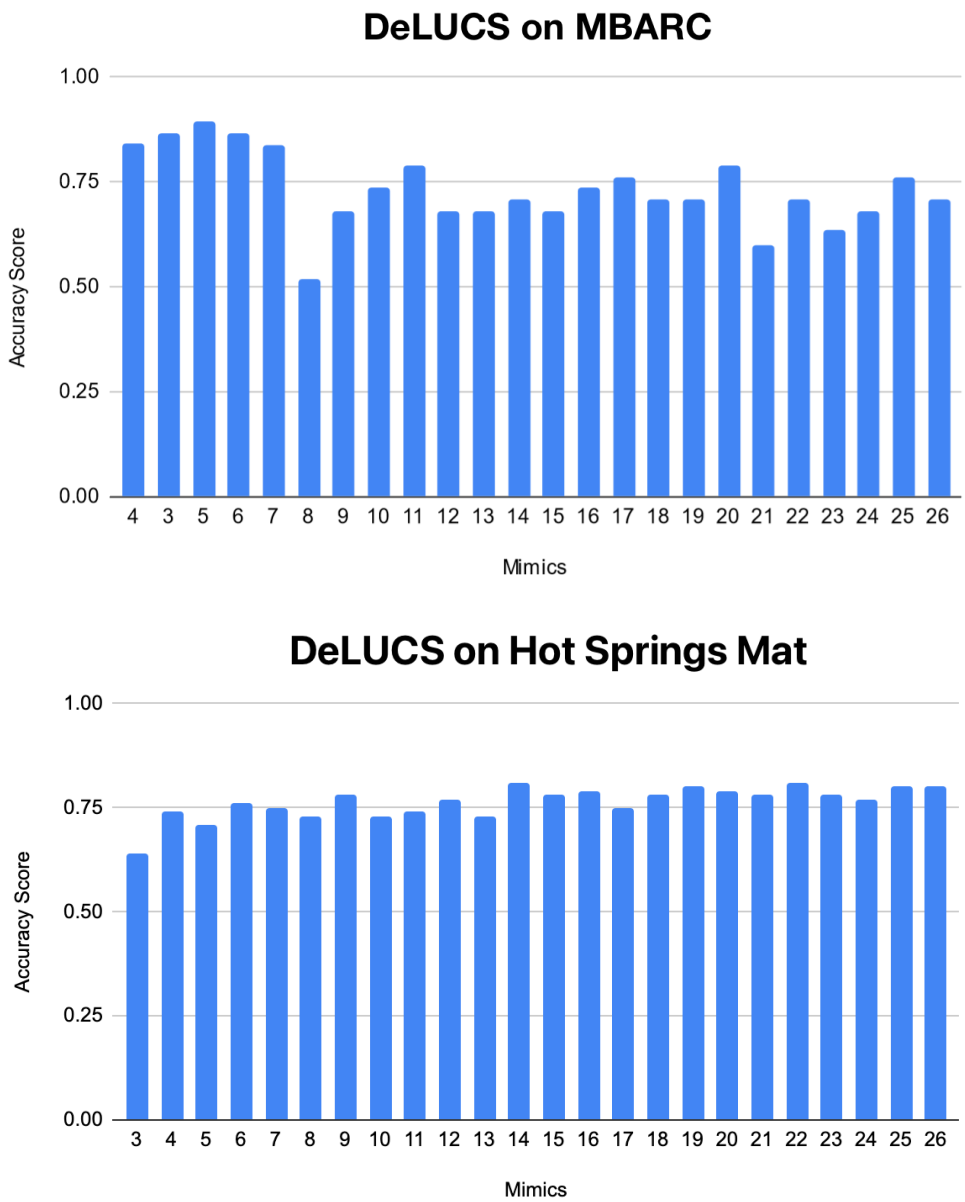

Fig 10. Accuracy achieved by different numbers of mimics in DeLUCS for MBARC-26 mock community (top) and synthetic Hot Springs Mat dataset (bottom).

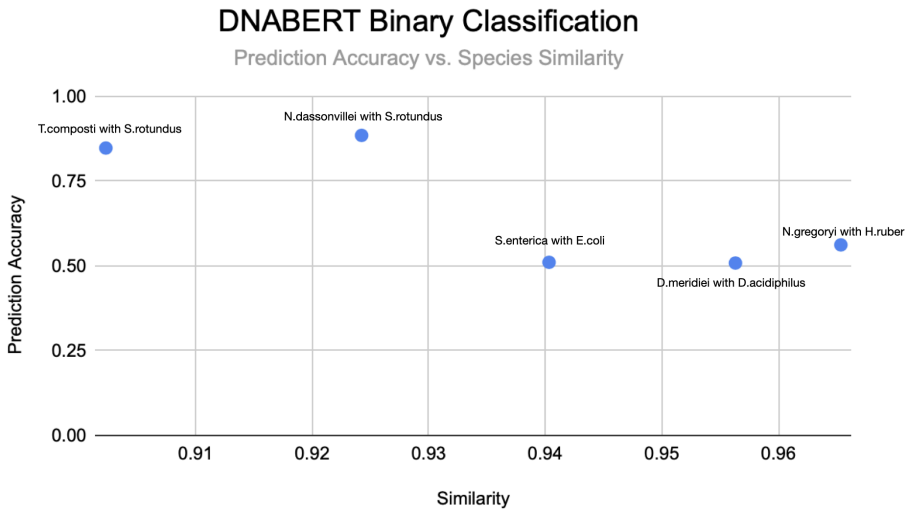

Fig 11. DNABERT classification accuracy for two species. As similarity between species increases the accuracy tends to decrease.

4.3.4 *DNABert Evaluation*

We performed DNABert experiments utilizing GPU-powered nodes. Even so, we found that classification of both our datasets took 7-9 days to execute even for shreds of size 500, and resulted in poor performance (accuracy of 44% for MBARC and 34% for the Hot Springs data).

In order to determine if DNABert is able to handle at least binary experiments well, we conducted a number of experiments to determine what level of similarity of genomes it is able to handle. Fig 11 shows our results when using a structural similarity algorithm (SSIM) comparison between the CGR representations of the organisms' 6-mer tokenized DNA sequences.

We find that while DNABERT is able to distinguish between two species when they are sufficiently dissimilar, it begins to produce poor predictions when species are at least 94% similar. For species pairs where similarity is 92.4% or less, DNABERT is much more accurate (88.5% compared to 51% prediction accuracy). Overall we found that DNABERT is both computationally cumbersome to use and not able to perform well on our data.

5 Discussion

The different tools demonstrated varying 16S reconstruction and identification capacity on genomes of varying characteristics for each of the factors we considered (repetitiveness, GC content, coverage and genome size). Different organisms were identifiable by the whole-genome assembly (as shown by cmsearch and Blast results) and others by ribosomal reconstruction tools (phyloFlash) and machine learning tools. Our results show that phyloFlash performed better than the whole genome assembly tools overall. Nevertheless, phyloFlash had trouble assembling specific organisms, while some whole-genome assembly tools successfully reconstructed the species that phyloFlash could not reconstruct. This can be seen in cmsearch or Blast over MEGAHIT and MetaSPAdes identifying the species that phyloFlash missed. In MBARC-26 the whole-genome assembly tools were able to identify 3 of the 4 genomes that phyloFlash missed. These results support the hypothesis that the different tools

have complementary behavior that we can exploit to achieve an overall better performance across different datasets.

The traditional bioinformatics tools and the machine learning approaches also showed complementary performance for organisms with different properties as surfaced by the four factors we considered. PhyloFlash made mistakes on genomes of high repetitiveness, low or high GC content or coverage extremes, while DeLUCS achieved better accuracy on those across both the MBARC-26 mock community and synthetic Hot Springs Mat datasets. DeLUCS also got confused on low or high GC content genomes, extreme repetitiveness, and small genomes, on both datasets.

6 Conclusion

As shown in the heatmaps there exist a variety of similar strains and species that can be deciphered to varying extents by the various tools. As the factor plots for the MBARC mock community show, phyloFlash gave the most True Positives, and the missed (non-reconstructed) genomes with phyloFlash fall in 3 categories: a. low or high coverage, b. high repetitiveness, c. low or high GC content. For the synthetic Hot Springs Mat dataset phyloFlash had a low 16S reconstruction capacity due to the low coverage and the similarity of different species. We found that different assembly methods and machine learning tools have complementary strengths for genomes of different properties, and identified initial characteristics of organisms where each tool performs better. This suggests that ensembling traditional 16s and whole genome assembly tools with machine learning approaches could result in an improved reconstruction and species identification accuracy. Based on the promise DeLUCS offered in species identification, we could use machine learning to identify species from the reconstructed 16Ss. We plan to explore an ensemble approach to differentiate between close strains as future work.

References

1. Thiel V, Hügler M, Ward DM, Bryant DA. The Dark Side of the Mushroom Spring Microbial Mat: Life in the Shadow of Chlorophototrophs. II. Metabolic Functions of Abundant Community Members Predicted from Metagenomic Analyses. Frontiers in Microbiology. 2017;8:943. doi:10.3389/fmicb.2017.00943.

2. Thiel V, Wood JM, Olsen MT, Tank M, Klatt CG, Ward DM, et al. The Dark Side of the Mushroom Spring Microbial Mat: Life in the Shadow of Chlorophototrophs. I. Microbial Diversity Based on 16S rRNA Gene Amplicons and Metagenomic Sequencing. Frontiers in Microbiology. 2016;7. doi:10.3389/fmicb.2016.00919.

3. Tyson GW, Chapman J, Hugenholtz P, Allen EE, Ram RJ, Richardson PM, et al. Community Structure and Metabolism through Reconstruction of Microbial Genomes from the Environment. Nature. 2004;428(6978):37–43. doi:10.1038/nature02340.

4. Anyansi C, Straub TJ, Manson AL, Earl AM, Abeel T. Computational Methods for Strain-Level Microbial Detection in Colony and Metagenome Sequencing Data. Frontiers in Microbiology. 2020;11:1925. doi:10.3389/fmicb.2020.01925.

5. Edgar RC. Updating the 97% Identity Threshold for 16S Ribosomal RNA OTUs. Bioinformatics. 2018;34(14):2371–2375. doi:10.1093/bioinformatics/bty113.

6. Stackebrandt E, Goebel BM. Taxonomic Note: A Place for DNA-DNA Reassociation and 16S rRNA Sequence Analysis in the Present Species Definition in Bacteriology. International Journal of Systematic and Evolutionary Microbiology. 1994;44(4):846–849. doi:10.1099/00207713-44-4-846.

7. Quince C, Nurk S, Raguideau S, James R, Soyer OS, Summers JK, et al. STRONG: Metagenomics Strain Resolution on Assembly Graphs. Genome Biology. 2021;22:214. doi:10.1186/s13059-021-02419-7.

8. Anyansi C, Straub TJ, Manson AL, Earl AM, Abeel T. Computational Methods for Strain-Level Microbial Detection in Colony and Metagenome Sequencing Data. Frontiers in Microbiology. 2020;11:1925. doi:10.3389/fmicb.2020.01925.

9. Peterson D, Bonham KS, Rowland S, Pattanayak CW, Klepac-Ceraj V. Comparative Analysis of 16S rRNA Gene and Metagenome Sequencing in Pediatric Gut Microbiomes. Frontiers in Microbiology. 2021;12:670336. doi:10.3389/fmicb.2021.670336.

10. Curry KD, Wang Q, Nute MG, Tyshaieva A, Reeves E, Soriano S, et al. Emu: Species-Level Microbial Community Profiling of Full-Length 16S rRNA Oxford Nanopore Sequencing Data. Nature Methods. 2022;19(7):845–853. doi:10.1038/s41592-022-01520-4.

11. Singer E, Andreopoulos B, Bowers RM, Lee J, Deshpande S, Chiniquy J, et al. Next Generation Sequencing Data of a Defined Microbial Mock Community. Scientific Data. 2016;3(1):160081. doi:10.1038/sdata.2016.81.

12. Fritz A, Hofmann P, Majda S, Dahms E, Dröge J, Fiedler J, et al. CAMISIM: Simulating Metagenomes and Microbial Communities. Microbiome. 2019;7(1):17. doi:10.1186/s40168-019-0633-6.

13. Klatt CG, Wood JM, Rusch DB, Bateson MM, Hamamura N, Heidelberg JF, et al. Community Ecology of Hot Spring Cyanobacterial Mats: Predominant Populations and Their Functional Potential. The ISME Journal. 2011;5(8):1262–1278. doi:10.1038/ismej.2011.73.

14. Yu FB, Meisel JS, Pop M, Bhaya D, Kim R. Deep Metagenomic Sequencing of Hot Spring Phototrophic Mats Generates a Comprehensive Panoramic View of Microbial Diversity; 2022.

15. Bhaya D, Grossman AR, Steunou AS, Khuri N, Cohan FM, Hamamura N, et al. Population Level Functional Diversity in a Microbial Community Revealed by Comparative Genomic and Metagenomic Analyses. The ISME Journal. 2007;1(8):703–713. doi:10.1038/ismej.2007.46.

16. Becraft ED, Cohan FM, Kühl M, Jensen SI, Ward DM. Fine-Scale Distribution Patterns of Synechococcus Ecological Diversity in Microbial Mats of Mushroom Spring, Yellowstone National Park. Applied and Environmental Microbiology. 2011;77(21):7689–7697. doi:10.1128/AEM.05927-11.

17. Almeida A, Mitchell AL, Tarkowska A, Finn RD. Benchmarking Taxonomic Assignments Based on 16S rRNA Gene Profiling of the Microbiota from Commonly Sampled Environments. GigaScience. 2018;7(5):giy054. doi:10.1093/gigascience/giy054.

18. Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB, et al. Introducing Mothur: Open-source, Platform-Independent, Community-Supported Software for Describing and Comparing Microbial Communities. Applied and Environmental Microbiology. 2009;75(23):7537–7541. doi:10.1128/AEM.01541-09.

19. Prodan A, Tremaroli V, Brolin H, Zwinderman AH, Nieuwdorp M, Levin E. Comparing Bioinformatic Pipelines for Microbial 16S rRNA Amplicon Sequencing. PLOS ONE. 2020;15(1):e0227434. doi:10.1371/journal.pone.0227434.

20. Wang Q, Garrity GM, Tiedje JM, Cole JR. Naive Bayesian Classifier for Rapid Assignment of rRNA Sequences into the New Bacterial Taxonomy. Applied and Environmental Microbiology. 2007;73(16):5261–5267. doi:10.1128/AEM.00062-07.

21. Fiannaca A, La Paglia L, La Rosa M, Lo Bosco G, Renda G, Rizzo R, et al. Deep Learning Models for Bacteria Taxonomic Classification of Metagenomic Data. BMC Bioinformatics. 2018;19(7):198. doi:10.1186/s12859-018-2182-6.

22. Liang Q, Bible PW, Liu Y, Zou B, Wei L. DeepMicrobes: Taxonomic Classification for Metagenomics with Deep Learning. NAR Genomics and Bioinformatics. 2020;2(lqaa009). doi:10.1093/nargab/lqaa009.

23. Lonèar-Turukalo T, Lazić I, Maljković N, Brdar S. Clustering of Microbiome Data: Evaluation of Ensemble Design Approaches. In: IEEE EUROCON 2019 -18th International Conference on Smart Technologies; 2019. p. 1–6.

24. Zeng F, Wang Z, Wang Y, Zhou J, Chen T. Large-Scale 16S Gene Assembly Using Metagenomics Shotgun Sequences. Bioinformatics. 2017;33(10):1447–1456. doi:10.1093/bioinformatics/btx018.

25. Cepeda V, Liu B, Almeida M, Hill CM, Koren S, Treangen TJ, et al. MetaCompass: Reference-guided Assembly of Metagenomes. bioRxiv. 2017; p. 212506. doi:10.1101/212506.

26. Koren S, Treangen TJ, Hill CM, Pop M, Phillippy AM. Automated Ensemble Assembly and Validation of Microbial Genomes. BMC Bioinformatics. 2014;15(1):126. doi:10.1186/1471-2105-15-126.

27. Zhu X, Leung HC, Chin FY, Yiu SM, Quan G, Liu B, et al. PERGA: A Paired-End Read Guided de Novo Assembler for Extending Contigs Using SVM and Look Ahead Approach. PloS one. 2014;9(12):e114253.

28. Padovani de Souza K, Setubal JC, Ponce de Leon F de Carvalho AC, Oliveira G, Chateau A, Alves R. Machine Learning Meets Genome Assembly. Briefings in Bioinformatics. 2019;20(6):2116–2129. doi:10.1093/bib/bby072.

29. Greener JG, Kandathil SM, Moffat L, Jones DT. A Guide to Machine Learning for Biologists. Nature Reviews Molecular Cell Biology. 2022;23(1):40–55. doi:10.1038/s41580-021-00407-0.

30. Harris ZN, Dhungel E, Mosior M, Ahn TH. Massive Metagenomic Data Analysis Using Abundance-Based Machine Learning. Biology Direct. 2019;14(1):12. doi:10.1186/s13062-019-0242-0.

31. Woloszynek S, Zhao Z, Chen J, Rosen GL. 16S rRNA Sequence Embeddings: Meaningful Numeric Feature Representations of Nucleotide Sequences That Are Convenient for Downstream Analyses. PLoS Computational Biology. 2019;15(2):e1006721. doi:10.1371/journal.pcbi.1006721.

32. Lamurias A, Sereika M, Albertsen M, Hose K, Nielsen TD. Metagenomic Binning with Assembly Graph Embeddings; 2022.

33. Iuchi H, Matsutani T, Yamada K, Iwano N, Sumi S, Hosoda S, et al. Representation Learning Applications in Biological Sequence Analysis. Computational and Structural Biotechnology Journal. 2021;19:3198–3208. doi:10.1016/j.csbj.2021.05.039.

34. Choi I, Ponsero AJ, Bomhoff M, Youens-Clark K, Hartman JH, Hurwitz BL. Libra: Scalable k-Mer–Based Tool for Massive All-vs-All Metagenome Comparisons. GigaScience. 2019;8(2):giy165. doi:10.1093/gigascience/giy165.

35. Ji Y, Zhou Z, Liu H, Davuluri RV. DNABERT: Pre-Trained Bidirectional Encoder Representations from Transformers Model for DNA-language in Genome. Bioinformatics. 2021;37(15):2112–2120. doi:10.1093/bioinformatics/btab083.

36. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding; 2019.

37. Marx V. Microbiology: The Road to Strain-Level Identification. Nature Methods. 2016;13(5):401–404. doi:10.1038/nmeth.3837.

38. Preheim SP, Perrotta AR, Martin-Platero AM, Gupta A, Alm EJ. Distribution-Based Clustering: Using Ecology To Refine the Operational Taxonomic Unit. Applied and Environmental Microbiology. 2013;79(21):6593–6603. doi:10.1128/AEM.00342-13.

39. Schloss PD. Amplicon Sequence Variants Artificially Split Bacterial Genomes into Separate Clusters. mSphere. 2021;6(4):e00191–21. doi:10.1128/mSphere.00191-21.

40. Callahan BJ, McMurdie PJ, Holmes SP. Exact Sequence Variants Should Replace Operational Taxonomic Units in Marker-Gene Data Analysis. The ISME Journal. 2017;11(12):2639–2643. doi:10.1038/ismej.2017.119.

41. Callahan BJ, McMurdie PJ, Rosen MJ, Han AW, Johnson AJA, Holmes SP. DADA2: High-resolution Sample Inference from Illumina Amplicon Data. Nature Methods. 2016;13(7):581–583. doi:10.1038/nmeth.3869.

42. White JR, Navlakha S, Nagarajan N, Ghodsi MR, Kingsford C, Pop M. Alignment and Clustering of Phylogenetic Markers - Implications for Microbial Diversity Studies. BMC Bioinformatics. 2010;11(1):152. doi:10.1186/1471-2105-11-152.

43. Nguyen NP, Warnow T, Pop M, White B. A Perspective on 16S rRNA Operational Taxonomic Unit Clustering Using Sequence Similarity. npj Biofilms and Microbiomes. 2016;2(1):1–8. doi:10.1038/npjbiofilms.2016.4.

44. Mahé F, Rognes T, Quince C, de Vargas C, Dunthorn M. Swarm: Robust and Fast Clustering Method for Amplicon-Based Studies. PeerJ. 2014;2:e593. doi:10.7717/peerj.593.

45. Schloss PD, Westcott SL. Assessing and Improving Methods Used in Operational Taxonomic Unit-Based Approaches for 16S rRNA Gene Sequence Analysis. Applied and Environmental Microbiology. 2011;77(10):3219–3226. doi:10.1128/AEM.02810-10.

46. Huse SM, Welch DM, Morrison HG, Sogin ML. Ironing out the Wrinkles in the Rare Biosphere through Improved OTU Clustering. Environmental Microbiology. 2010;12(7):1889–1898. doi:10.1111/j.1462-2920.2010.02193.x.

47. Edgar RC. UPARSE: Highly Accurate OTU Sequences from Microbial Amplicon Reads. Nature Methods. 2013;10(10):996–998. doi:10.1038/nmeth.2604.

48. Hao X, Jiang R, Chen T. Clustering 16S rRNA for OTU Prediction: A Method of Unsupervised Bayesian Clustering. Bioinformatics (Oxford, England). 2011;27(5):611–618. doi:10.1093/bioinformatics/btq725.

49. Rasheed Z, Rangwala H, Barbará D. 16S rRNA Metagenome Clustering and Diversity Estimation Using Locality Sensitive Hashing. BMC Systems Biology. 2013;7(4):S11. doi:10.1186/1752-0509-7-S4-S11.

50. Chaudhary N, Sharma AK, Agarwal P, Gupta A, Sharma VK. 16S Classifier: A Tool for Fast and Accurate Taxonomic Classification of 16S rRNA Hypervariable Regions in Metagenomic Datasets. PLoS ONE. 2015;10(2):e0116106. doi:10.1371/journal.pone.0116106.

51. Tikhonov M, Leach RW, Wingreen NS. Interpreting 16S Metagenomic Data without Clustering to Achieve Sub-OTU Resolution. The ISME Journal. 2015;9(1):68–80. doi:10.1038/ismej.2014.117.

52. Segata N. On the Road to Strain-Resolved Comparative Metagenomics. mSystems. 2018;3(2):e00190–17. doi:10.1128/mSystems.00190-17.

53. Smillie CS, Sauk J, Gevers D, Friedman J, Sung J, Youngster I, et al. Strain Tracking Reveals the Determinants of Bacterial Engraftment in the Human Gut Following Fecal Microbiota Transplantation. Cell Host & Microbe. 2018;23(2):229–240.e5. doi:10.1016/j.chom.2018.01.003.

54. Ventolero MF, Wang S, Hu H, Li X. Computational Analyses of Bacterial Strains from Shotgun Reads. Briefings in Bioinformatics. 2022;23(2):bbac013. doi:10.1093/bib/bbac013.

55. Nayfach S, Rodriguez-Mueller B, Garud N, Pollard KS. An Integrated Metagenomics Pipeline for Strain Profiling Reveals Novel Patterns of Bacterial Transmission and Biogeography. Genome Research. 2016;26(11):1612–1625. doi:10.1101/gr.201863.115.

56. Subedi S, Neish D, Bak S, Feng Z. Cluster Analysis of Microbiome Data by Using Mixtures of Dirichlet–Multinomial Regression Models. Journal of the Royal Statistical Society: Series C (Applied Statistics). 2020;69(5):1163–1187. doi:10.1111/rssc.12432.

57. Fang Y, Subedi S. Clustering Microbiome Data Using Mixtures of Logistic Normal Multinomial Models; 2020.

58. Liu Z, Klatt CG, Wood JM, Rusch DB, Ludwig M, Wittekindt N, et al. Metatranscriptomic Analyses of Chlorophototrophs of a Hot-Spring Microbial Mat. The ISME journal. 2011;5(8):1279–1290. doi:10.1038/ismej.2011.37.

59. Smit A, Hubley R, Green P. RepeatMasker. RepeatMasker Open-30. 2010; p. 1996.

60. Bushnell B. BBMap: A Fast, Accurate, Splice-Aware Aligner. Lawrence Berkeley National Lab. (LBNL), Berkeley, CA (United States); 2014. LBNL-7065E.

61. Gruber-Vodicka HR, Seah BKB, Pruesse E. phyloFlash – Rapid SSU rRNA Profiling and Targeted Assembly from Metagenomes. Bioinformatics; 2019.

62. Nurk S, Meleshko D, Korobeynikov A, Pevzner PA. metaSPAdes: A New Versatile Metagenomic Assembler. Genome Research. 2017;27(5):824–834. doi:10.1101/gr.213959.116.

63. Li D, Luo R, Liu CM, Leung CM, Ting HF, Sadakane K, et al. MEGAHIT v1.0: A Fast and Scalable Metagenome Assembler Driven by Advanced Methodologies and Community Practices. Methods. 2016;102:3–11. doi:10.1016/j.ymeth.2016.02.020.

64. Chen IMA, Chu K, Palaniappan K, Ratner A, Huang J, Huntemann M, et al. The IMG/M Data Management and Analysis System v.6.0: New Tools and Advanced Capabilities. Nucleic Acids Research. 2021;49(D1):D751–D763. doi:10.1093/nar/gkaa939.

65. Quast C, Pruesse E, Yilmaz P, Gerken J, Schweer T, Yarza P, et al. The SILVA Ribosomal RNA Gene Database Project: Improved Data Processing and Web-Based Tools. Nucleic Acids Research. 2012;41(D1):D590–D596. doi:10.1093/nar/gks1219.

66. De Bruijn FJ. Handbook of Molecular Microbial Ecology I: Metagenomics and Complementary Approaches. vol. 1. John Wiley & Sons; 2011.

67. Wick RR, Judd LM, Gorrie CL, Holt KE. Unicycler: Resolving Bacterial Genome Assemblies from Short and Long Sequencing Reads. PLOS Computational Biology. 2017;13(6):e1005595. doi:10.1371/journal.pcbi.1005595.

68. Shlemov A, Korobeynikov A. PathRacer: Racing Profile HMM Paths on Assembly Graph. In: Holmes I, Martín-Vide C, Vega-Rodríguez MA, editors. Algorithms for Computational Biology. Lecture Notes in Computer Science. Cham: Springer International Publishing; 2019. p. 80–94.

69. Arias PM, Alipour F, Hill KA, Kari L. DeLUCS: Deep Learning for Unsupervised Clustering of DNA Sequences. Bioinformatics; 2021.

70. Almeida JS, Carrico JA, Maretzek A, Noble PA, Fletcher M. Analysis of Genomic Sequences by Chaos Game Representation. Bioinformatics. 2001;17(5):429–437. doi:10.1093/bioinformatics/17.5.429.