Microbiology Spectrum

| Open Peer Review | Environmental Microbiology | Research Article

Investigation of machine learning algorithms for taxonomic classification of marine metagenomes

Helen Park,[1,2] Shen Jean Lim,[3,4,5] Jonathan Cosme,[6] Kyle O'Connell,[7,8] Jilla Sandeep,[9] Felimon Gayanilo,[9] George R. Cutter Jr.,[10] Enrique Montes,[3,4] Chotinan Nitikitpaiboon,[11] Sam Fisher,[7] Hassan Moustahfid,[12] Luke R. Thompson[4,13]

AUTHOR AFFILIATIONS See affiliation list on p. 14.

ABSTRACT Microbial communities play key roles in ocean ecosystems through regulation of biogeochemical processes such as carbon and nutrient cycling, food web dynamics, and gut microbiomes of invertebrates, fish, reptiles, and mammals. Assessments of marine microbial diversity are therefore critical to understanding spatiotemporal variations in microbial community structure and function in ocean ecosystems. With recent advances in DNA shotgun sequencing for metagenome samples and computational analysis, it is now possible to access the taxonomic and genomic content of ocean microbial communities to study their structural patterns, diversity, and functional potential. However, existing taxonomic classification tools depend upon manually curated phylogenetic trees, which can create inaccuracies in metagenomes from less well-characterized communities, such as from ocean water. Herein, we explore the utility of deep learning tools—DeepMicrobes and a novel Residual Network architecture—that leverage natural language processing and convolutional neural network architectures to map input sequence data (k-mers) to output labels (taxonomic groups) without reliance on a curated taxonomic tree. We trained both models using metagenomic reads simulated from marine microbial genomes in the MarRef database. The performance of both models (accuracy, precision, and percent microbe predicted) was compared with the standard taxonomic classification tool Kraken2 using 10 complex metagenomic data sets simulated from MarRef. Our results demonstrate that time, compute power, and microbial genomic diversity still pose challenges for machine learning (ML). Moreover, our results suggest that high genome coverage and rectification of class imbalance are prerequisites for a well-trained model, and therefore should be a major consideration in future ML work.

IMPORTANCE Taxonomic profiling of microbial communities is essential to model microbial interactions and inform habitat conservation. This work develops approaches in constructing training/testing data sets from publicly available marine metagenomes and evaluates the performance of machine learning (ML) approaches in read-based taxonomic classification of marine metagenomes. Predictions from two models are used to test accuracy in metagenomic classification and to guide improvements in ML approaches. Our study provides insights on the methods, results, and challenges of deep learning on marine microbial metagenomic data sets. Future machine learning approaches can be improved by rectifying genome coverage and class imbalance in the training data sets, developing alternative models, and increasing the accessibility of computational resources for model training and refinement.

KEYWORDS metagenomics, machine learning, marine microbiology

Microbial community profiling in marine ecosystems is essential to our understanding of how microbes interact and respond to changes in their environments,

Editor Paul A. Jensen, University of Michigan-Ann Arbor, Ann Arbor, Michigan, USA

Address correspondence to Luke R. Thompson, luke.thompson@noaa.gov.

The authors declare no conflict of interest.

See the funding table on p. 15.

Received 21 December 2022
Accepted 30 June 2023
Published 11 September 2023

for end-to-end marine ecosystem modeling (1) and for informing habitat management and conservation (2). Community DNA shotgun sequencing (metagenomics) facilitates the characterization of the diversity, abundance, and functional potential of microbial communities from diverse habitats. The assignment of taxonomic labels to short metagenomic reads is useful for the profiling and comparison of microbial diversity and composition across metagenomic libraries. Various tools, including Centrifuge (3), Kaiju (4), Sourmash (5), and Kraken2, (6) have been developed to accurately and quickly assign metagenomic reads to microbial taxa. These tools identify unique nucleotide sequences (k-mers) from each metagenomic read and map them to reference genomes with nodes on the taxonomic trees curated by the National Center for Biotechnology Information (NCBI). Reliance on a curated taxonomic tree optimizes read classification for microbial taxa with well-defined lineages, such as those in the human gut (7, 8); however, these algorithms may not be sufficient for classification of complex microbial communities in ecosystems where many microbial taxa remain to be characterized, such as those in marine habitats (9).

Machine learning (ML) is a promising alternative approach for read-based taxonomic classification that circumvents the requirement for a taxonomic tree, due to its ability to handle complex data-heavy prediction problems without *a priori* knowledge. Deep learning (DL) is a branch of ML that uses a many-layered (i.e., deep) structure. This allows complex tasks to be learned as a stack of modules that recognize increasingly abstract concepts (10). DL models can be complicated to build and time-consuming to train but have successfully solved otherwise intractable problems (11). For example, certain DL algorithms have been shown to improve upon classic taxonomic classification tools because they are able to classify new genomes that are not yet characterized in NCBI, thereby improving the percentage of classified sequences in a metagenome and reducing the number of sequences incorrectly predicted (7). Certain types of neural networks in DL, such as artificial neural networks and architectures with a foundation in image analysis, have been borrowed to explore DNA sequence-based applications (12). For example, convolutional neural network (CNN) architectures have been used in host phenotype prediction (13) and gene identification in sequenced genomes (14). CNNs are the most utilized of the DL architectures with the added benefit of training faster natural language processing (NLP) models that are more adept at learning the patterns of language and words (15). NLP-based methods appear to be better suited for metagenomics since nucleotide sequences can be processed as words and are less amenable to representation as images (16). For example, NLP-based methods that use filters to slide off DNA "sentence" matrices have been applied for genomic binning (17), viral sequence identification, and phenotypic classification for cancer reads (18). DL applications for taxonomic classification have been explored to some degree for NLP (19) and CNN architectures (20) but have not yet been widely applied (7).

Here, we investigated whether deep learning models could outperform tree-based prediction approaches in standard tools like Kraken2 (6) for read-based taxonomic classification from metagenomic libraries sequenced from marine habitats, which typically have high microbial diversity (21). Specifically, we first re-trained DeepMicrobes, a NLP-analogous DL model that leverages k-mer embedding to create a custom genetic code "dictionary" in order to learn the relationship between the input sequence reads (k-mers) and output labels (taxonomic classifications) without an input phylogenetic tree (22). For this study, we used parameters of the best-performing DeepMicrobes algorithm, which was found to be a bidirectional long short-term memory (bi-LSTM) model with self-attention mechanism and k-mer embedding, implemented in the TensorFlow library (22). With this architecture, DeepMicrobes was reported to outperform standard tools when trained exclusively on the human-specific reference (HGR) database of gut metagenomes. When trained on the HGR data set at a model confidence threshold of 0.5, DeepMicrobes could obtain a precision and recall of 0.969/0.866. Here, DeepMicrobes was re-trained and optimized in order to develop taxonomic classification models

specific for marine metagenomes, which have markedly higher microbial diversity than human gut metagenomes (8, 21).

Recognizing that DL models such as DeepMicrobes are hindered by slow training and prediction times, we also built and tested a CNN architecture as an alternative approach for taxonomic classification. Specifically, we selected the Residual Network (ResNet) model, since it is a well-known and effectively scaled CNN architecture (23). Results from both models were used to explore how metrics such as percent genome coverage, GC (guanine-cytosine) content, count of species samples in training, and the class distribution in training set might impact ML classification performance on marine metagenomics data.

MATERIALS AND METHODS

Training, testing, and validation data sets

Taxonomic and sequence data from version 1.6 of the MarRef database (24, 25) was used both to simulate metagenomic reads for taxonomic classification and as a source of artificial marine metagenomes for model training, testing, and validation. We selected the best-performing model architecture and embedding (k-mer embedding) from the prior DeepMicrobes research, in order to limit time and computational power. MarRef is a manually curated marine microbial reference database containing fully sequenced genomes (24, 25). A total of 1,271 genomes of marine prokaryotic species were retrieved from MarRef to develop the training, testing, and validation data sets. For each genome, we extracted the Genome Taxonomy Database (GTDB) (26) genus and species classification from the MarRef database's metadata file. Records without GTDB classifications were manually queried against GTDB release 07-RS207 to retrieve their taxonomic classifications. Of the 1,271 MarRef genomes, four records had undefined GTDB taxonomy due to failed quality checks and were dropped from the data set. The remaining 1,267 genomes (27) were used for model training and testing.

Training and testing data sets were created using methods described by Liang et al. (22) (Fig. 1). For each genome, 10,000 forward and 10,000 reverse reads were simulated with the ART Illumina read simulator (7) version ART-MountRainier-2016-06-05 using the following parameters: 150 bp read length (-l 150), 400 bp mean insert size (-m 400) with 50 bp standard deviation (-s 50), and HiSeq 2500 error model (-ss HS25). Different random seeds were used for the training set (rs 747) and the testing set (rs 808). Simulated reads were randomly trimmed from the 3′ end to generate variable read lengths between 75 bp and 150 bp in equal probability, using the random_trim.py script available in the DeepMicrobes repository (https://github.com/MicrobeLab/DeepMicrobes). For training and prediction with DeepMicrobes, read sequences and their accompanying genus or species labels were converted to the binary TensorFlow format using DeepMicrobe's tfrec_train_kmer.sh script for the training data, while read sequences without label data were converted to TensorFlow format using DeepMicrobe's tfrec_predict_kmer.sh script. Jellyfish v1.1.11 (4) was used to construct a comprehensive 12 bp k-mer vocabulary for DeepMicrobes from 47,894 genomes from GTDB release 202 (26), since the 12 bp k-mer model was the best performing in the original manuscript. For performance comparison, the default parameters of CAMISIM v1.3 (28) were used to perform 10 separate read simulations from the 1,267 MarRef genomes, with random seeds different from those used in the training and testing sets. Simulated reads were then combined at random proportions by CAMISIM to create 10 blind metagenomic data sets (Fig. 1).

Deepmicrobes implementation

For this study, we used the best-performing parameters from the original DeepMicrobes algorithm (22), which was trained on microbial genomes from the human gut, to re-train our deep learning models for species and genus classification from marine metagenomic

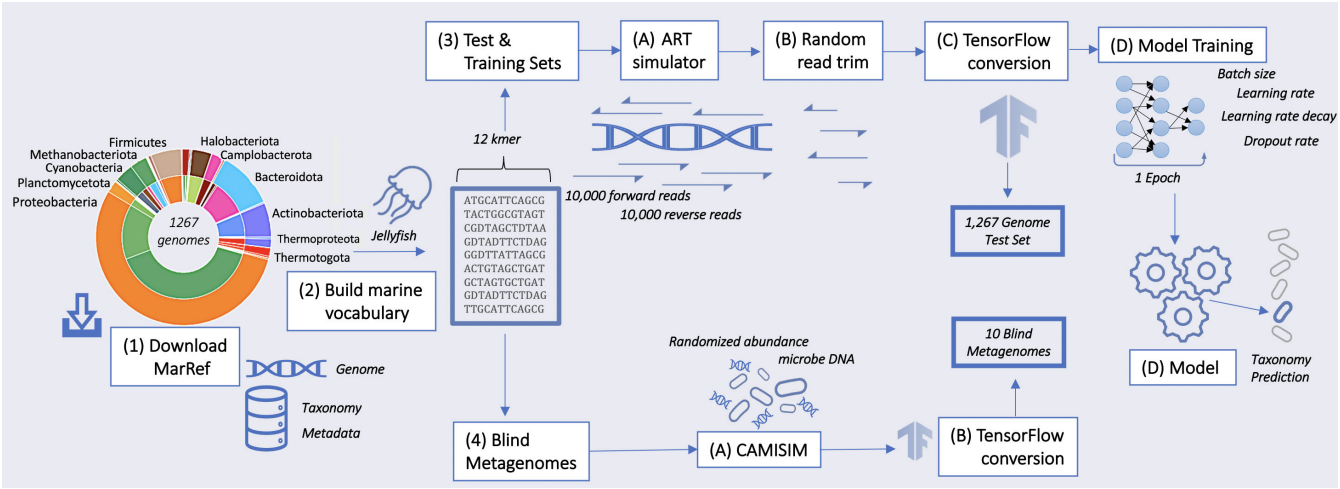


FIG 1 Overview of data preparation and machine learning pipeline. (1) Full-length genome records and their metadata were downloaded from MarRef v1.6. (2) A k-mer DNA vocabulary was built using the genomes in the GTDB database. We used a k-mer vocabulary of 12 base pairs. (3) Testing and training sets were created using the ART simulator, generating 10,000 forward reads and 10,000 reverse reads from each microbial genome, then randomly trimming the reads, and then converting into TensorFlow record format. (4) Blind data sets were created using CAMISIM, which randomly generates metagenomic data sets from the MarRef genomes. Each model's performance was evaluated using the 10 blind metagenomic data sets.

reads. Due to computational resource limitations, other models and parameters explored by the original DeepMicrobes algorithm, including those based on one-hot encoding, were not re-evaluated in this study.

The models were constructed using Deep Neural Networks implemented in the TensorFlow Python library. Each 12-mer was mapped to a k-mer embedding vector, and all vectors were concatenated into a 2D matrix that is processed by bi-LSTM. A self-attention mechanism was applied between the bi-LSTM and three fully connected (dense) classifiers with rectified linear unit (ReLu) activations in between. The first two layers contain 3,000 units each, and the final layer uses a softmax function that outputs class probabilities for each species ($n = 914$) or genus ($n = 515$). The model was optimized with the Adam Optimizer, which minimizes the distance (cross-entropy loss) between the output probabilities and the ground truth.

Training was performed using the DeepMicrobes.py script and accelerated using a NVIDIA v100 graphics processing unit (GPU). Each model was trained one epoch at a time due to the wall time limitation imposed by the server (4 days). During hyperparameter tuning, different batch sizes (1,024, 2,048, 4,096, 10,000), learning rates (0.0005, 0.005, 0.01, 0.001), dropout rates (0.5, 0.9, 1.0), and learning rates of decay (0.001, 0.05, 0.01, 0.1, 0.5) were varied. Each parameter-adjusted model was trained for one epoch, and performance of each model was compared using the accuracy and cross entropy generated during training. A full grid search varying all parameters together was deemed infeasible due to lack of compute power, and regardless, results indicate that parameter tuning did not have a large impact on model accuracy. The hyperparameter screen was used to compare DeepMicrobes model results in order to select the best model, and later sections only used the hyperparameter-optimized models (batch size: 4,096; learning rate: 0.05; dropout rate: none; learning rate of decay: 0.001) for further performance comparison with standard tools.

Model performance metrics

We used well-documented classification metrics (29) to evaluate the performance of our optimized models in predicting microbial genus or species. Metrics used included precision, recall, true positive rate, false positive rate, and Matthews correlation coefficient (MCC) (30):

$$\text{Precision} = \frac{TP}{TP+FP} \quad \text{Recall} = \frac{TP}{TP+FN} \quad \text{TPR} = \frac{TP}{TP+FN} \quad \text{FPR} = \frac{FP}{FP+TN}$$

$$\text{MCC} = \frac{TN \times TP - FN \times FP}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. Precision and recall are standard ML metrics used to measure model performance. MCC was used as a cutoff to identify microbes that were consistently predicted incorrectly in the test set, as MCC is only high if the classifier is doing well on both negative and positive elements.

To assess the performance of our optimized genus/species models on metagenomics data, we additionally calculated the percentage of reads predicted correctly (% accuracy) and the percentage of reads that can be assigned to a genus/species (% characterized) at different model CTs:

$$\%\ \text{Accuracy}_{\text{CT}} = \frac{\#\ \text{reads predicted correctly}_{\text{CT}}}{\text{total}\ \#\ \text{reads}_{\text{CT}}} \quad \%\ \text{Characterized}_{\text{CT}} = \frac{\text{total}\ \#\ \text{reads}_{\text{CT}}}{\text{total}\ \#\ \text{reads}}$$

Kraken2 read classification

The performance of our optimized genus/species classification models was compared with the performance of Kraken2 using the blind metagenomic data sets. Kraken2 is a widely used k-mer-based taxonomic classification tool that maps each k-mer to a lowest common ancestor using information from a curated taxonomy tree (6) with high accuracy and speed. Kraken2 was one of the tools that performed best in a comparison of machine learning and k-mer-based taxonomic classification methods for metagenomics (6, 22). Kraken2 v2.1.2 was used to assign taxonomy to metagenomic reads in the blind data sets using MarRef v1.6 as the reference database. Whole-genome FASTA files (n = 1,267) were downloaded (24, 25) and concatenated, and FASTA headers were updated to contain NCBI taxids, according to the Kraken2 manual. The database was built using the commands kraken2-build --download-taxonomy, kraken2-build --add-to-library, and kraken2-build --build. Sequences were classified using the command Kraken2 with options --paired, -- use-names, and --threads 24. A blind data set of 333,120 sequences (99.94 Mbp) was processed in 2.469 s. Species-level reports were generated using Bracken version 2.7 (31). The performance of Kraken2 was evaluated by percent accuracy and percent characterized, as described above.

Calculation of percent coverage of reference genome

Bowtie2 v2.4.5's bowtie-build command was used to generate a reference database containing all 1,267 MarRef genomes from the training set. Trimmed simulated reads used for training were mapped to this reference database using the following parameters: -p 12 (number processors), -f (FASTA files format), and –no-unal (repress unaligned reads). Samtools v1.9's view and sort commands were used to generate a sorted binary alignment map file, and the mpileup command was used to retrieve the total number of bases with at least 1× coverage, according to the online guide (32). Percent coverage was calculated by taking the total number of bases with a mapped fragment in the training set divided by the total length of the reference genome. Our shell script can be found at https://github.com/helloftroy/MarRef_DeepMicrobes.

CNN implementation

We also explored the utility of a ResNet CNN architecture for the taxonomic classification of marine microbial reads. CNN has the potential to train faster than NLP models like DeepMicrobes, and ResNet is a well-utilized and effectively scaled CNN architecture (23). We created a subset of the training data set by randomly selecting 10% (~530,000) of all FASTA sequences, and created a training split (80%), a validation split (10%), and

a test split (10%) from the subset. All data manipulation was performed on a NVIDIA Rapids version 22.10 GPU, after converting the FASTA files to a Rapids DataFrame. We utilized a parquet format for reading and writing and observed substantial accelerations in data processing. k-mer sampling of 1-mers, 3-mers, and 12-mers were tested in different ResNet models. Notably, the original DeepMicrobes found a 12-mer sampling performed best; however, we expected the CNN's convolution mechanism should be able to more aptly capture complex relationships between neighboring values, and we initially deemed a 1-mer may even be sufficient for our model. The 3-mers and 12-mers were created by applying a rolling window of 3 or 12 to the original one-dimensional (1D) input, leading to input shapes of 1-mer (1, 150, 1), 3-mer (148, 3, 1), and 12-mer (139, 12, 1).

For the CNN, we began with the ResNet-50 architecture because of its popularity and known effectiveness for classification tasks. Each of our ResNet models was modified to take a 1D vector. Our baseline model contained ~24.8 million trainable parameters, and takes a 1×150 vector as input (Fig. 2a). The input vector represented a nucleotide sequence, where each possible base (A, C, G, T) is encoded and represented by an integer. Sequences with fewer than 150 bases were zero-padded (right side) to a length of 150. We then adjusted the ResNet baseline by varying the number of trainable parameters. Specifically, we trained smaller versions of the ResNet with ~6.7 million (Fig. 2b), ~36,000 (Fig. 2c), and ~32,000 (Fig. 2d) parameters. Notably, these models changed the number of layers from 50 (baseline) to only 6 (ResNet-smaller-3).

Next, we altered the input shape of the data. We adjusted the input 1×150 vector by applying a rolling window of 12 to the vector and transformed it into a 139×12 matrix. We tested two versions of this approach, one where the convolution kernel size was 1×3 (the same as our baseline) (Fig. 2e), and one where the convolution kernel size was 3×3 (Fig. 2f). We also tested a variation of the ResNet-smaller-3 architecture where the first convolution kernel size was 1×3 (rather than 1×7) with a stride of 1 (rather than 2) and omitted the maxpool layer (Fig. 2g). Finally, we tested a version of the ResNet-smaller-3 architecture that applied a sliding window of 3 to the input vector to make a 148×3 matrix, the first convolution kernel size was 3×3 (rather than 7×7) with a stride of 1, the maxpool layer was omitted, and the convolution kernel size was 1×3 (Fig. 2h).

For all cases, the hyperparameters were kept constant (batch size: 1,024; epochs: 3; learning rate: 0.0003; inner-layers activation: ReLU; batch-norm momentum: 0.6; batch-norm scale: false; batch-norm center: true; batch normalization epsilon: 1e-8; last-layer activation: softmax; number of classes (species): 912; optimization algorithm: Adam; loss function: categorical cross entropy; metric: categorical accuracy).

RESULTS

DeepMicrobes hyperparameter tuning indicates high confidence threshold and low epoch number is needed for best precision and recall

To train an optimal DeepMicrobes model, we varied the learning rate, rate of decay, dropout rate, and batch size for one epoch each (2.5- to 4-day training time) (Fig. S1). We found that the cross-entropy loss and accuracy were optimal with a learning rate of 0.005 but were unaffected by changes in dropout probability and learning rate decay (Fig. S1). We also found that lower batch size improved performance but can increase training time, often prohibitively if the training set is large (Fig. 3d and e). For all further training, we selected a learning rate of 0.001 (default), decay rate 0.05, and no dropout rate (default), with a batch size of 4,096 that did not impede training time.

We next tested how the read-level confidence threshold (0–100) and number of epochs (3, 7, 10) impact the model's precision/recall (Fig. 3c; Fig. S2). Thresholds of 0 and 60 are shown to illustrate how the model metrics change as the model's confidence is increased (Fig. 3c) and the threshold of 60 in particular is shown because it leads to a peak in precision/recall (0.92/0.901) (Fig. 4d). As confidence threshold increased for each model, the precision/recall transitioned from a linear spread between 0 and 1 to a tight distribution approaching 1, as reads predicted with low confidence were dropped. We

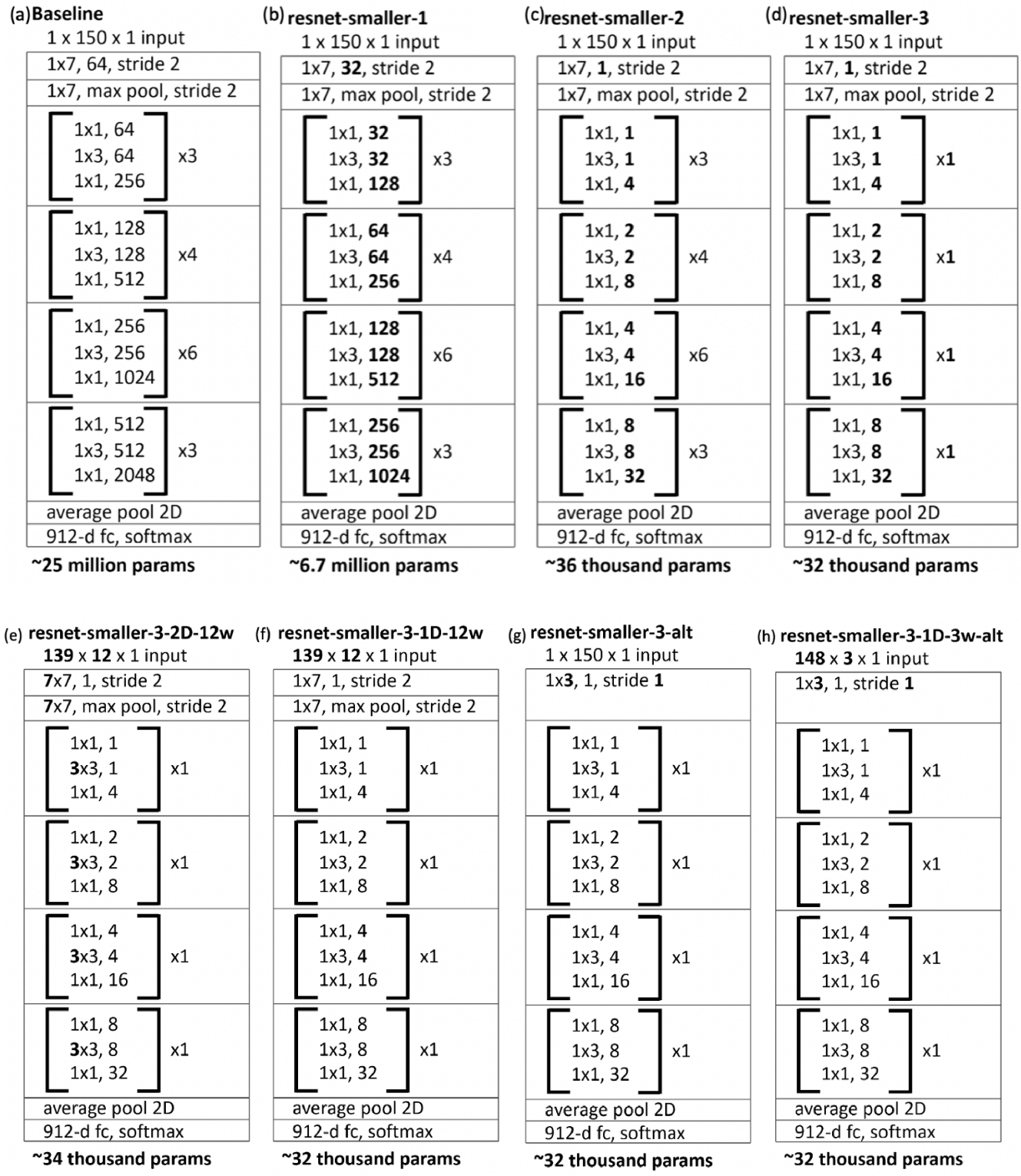
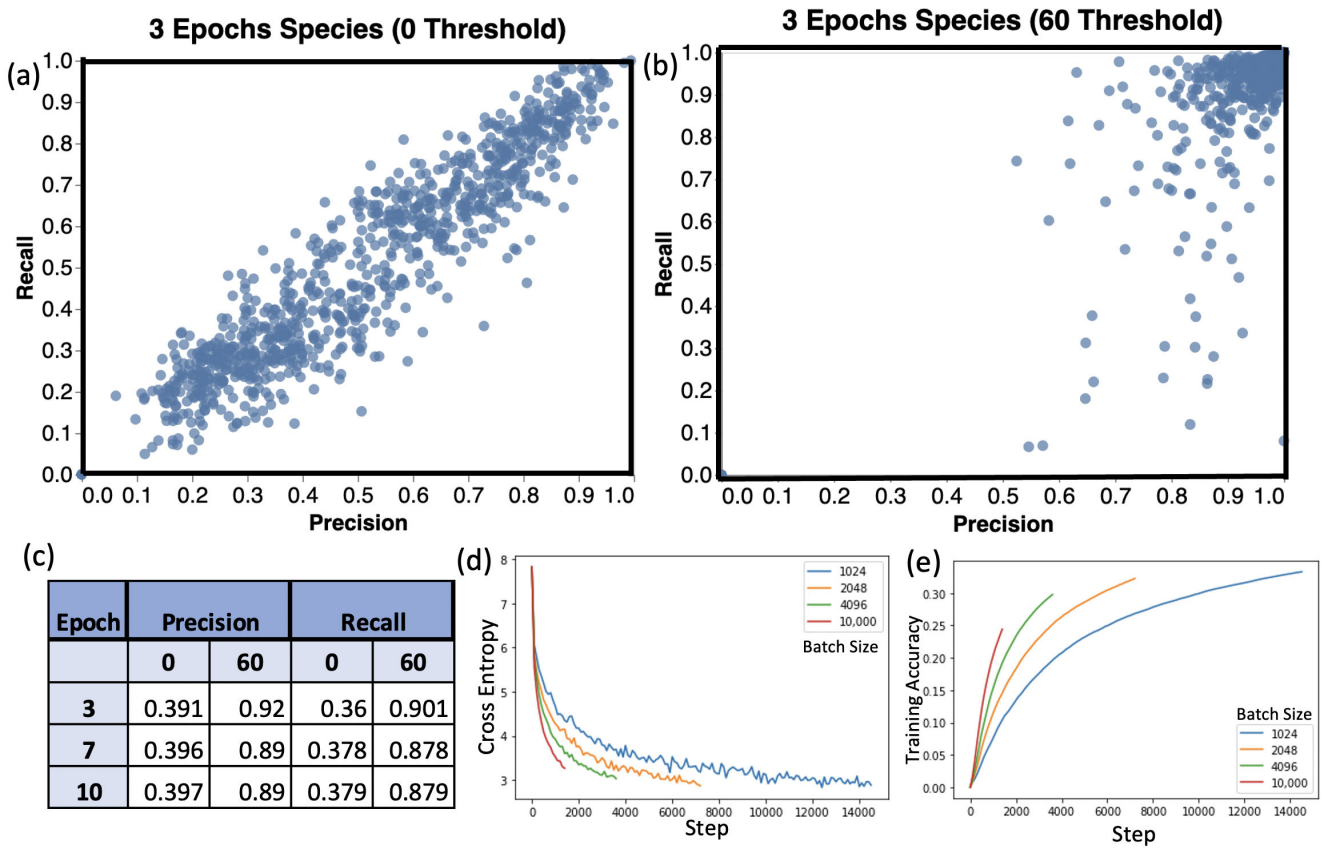


FIG 2 Convolutional neural network model with ResNet architecture. Numbers in bold indicate which variables were changed between architectures. (a) Baseline ResNet-50 architecture, with 24.8 million trainable parameters. Modified ResNet with (b) 6.7 million trainable parameters, (c) 36,000 trainable parameters, and (d) 32,000 trainable parameters. (e–h) Alternative tested ResNet architectures with adjusted input data shape after application of size 12 rolling window, with kernel size (e) 1 × 3 and (f) 3 × 3. (g,h) Variation on ResNet-smaller-3 with a stride of 1, the second maxpool layer omitted, and with a convolutional kernel size 1 × 3 (g) or with a transformed 148 × 3 input matrix (h).



Epoch	Precision		Recall	
	0	60	0	60
3	0.391	0.92	0.36	0.901
7	0.396	0.89	0.378	0.878
10	0.397	0.89	0.379	0.879

FIG 3 Hyperparameter tuning, epoch, and confidence threshold impacts on DeepMicrobes read-level prediction. (a and b) DeepMicrobes species model trained for three epochs at (a) 0 and (b) 60 confidence threshold. Each point represents a microbial species. As confidence increases, so do both precision/recall until nearly all microbes are predicted correctly. (c) Table illustrating how the number of epochs influences precision and recall. (d and e) Cross entropy and training accuracy, respectively, for hyperparameter tuning of four different batch sizes (1,024, 2,048, 4,096, 10,000). Batch size refers to the number of reads the model uses before updating its weights; A smaller batch size effectively allows the model to update its weights more frequently. Additional hyperparameter tuning can be found in (Fig. S1).

plotted the precision and recall for each microbial species at two different thresholds (0, 60), where each point represents a microbial species. For the best-performing model trained with three epochs (*S-3E-60*; species-3 epoch-60 threshold), we observed an increase in precision/recall between confidence thresholds 0 (0.39/0.36) and 60 (0.92/0.90) (Fig. 3a and b). Our results indicate that a high confidence threshold is needed to consistently obtain good performance. Interestingly, the highest precision/recall was found by the model trained with the least number of cycles (*S-3E-60*) at threshold of 60, but decreased at longer training time. Longer training appears to deteriorate performance, possibly due to overfitting. Meanwhile, at 0 confidence, the precision/recall increased only 0.1%–0.5% between epochs 3 and 10.

Kraken2 outperforms DeepMicrobes when benchmarked with blind data sets

We benchmarked the performance of our best-performing species and genus classification models (*S-2E, S-3E, G-2E, G-3E*; species 2 and 3 epoch, genus 2 and 3 epoch) against Kraken2 using 10 simulated blind metagenomes that contain random proportions of reads from the MarRef genomes. The models were compared using read-level accuracy and % reads characterized (Fig. 4). At confidence threshold 0, each model's accuracy was between 0.42 and 0.48; however, as the confidence threshold increased, accuracy increased to near one but percent of reads characterized decreased substantially. Indeed,

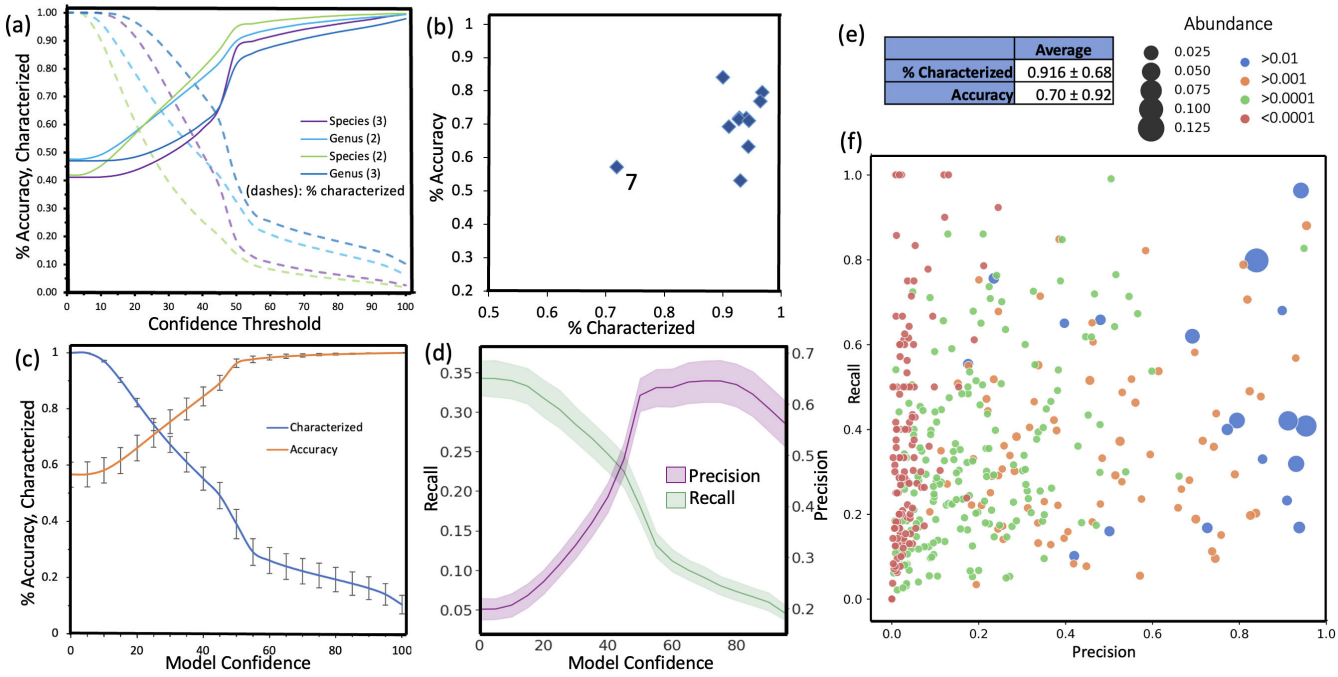


FIG 4 The performance of DeepMicrobes compared to Kraken2. (a) DeepMicrobes top four performing models across all confidence thresholds. A taxonomy prediction for accuracy and % characterized for 10 blind metagenomes using (b) Kraken2 and (c) DeepMicrobes top performing model (*G-2E-0*) across all confidence thresholds. In (b), each diamond is a blind set, and in (c), the bars indicate the standard deviation between blind sets. Notably, blind set 7 had markedly lower accuracy and percent characterized for Kraken2; upon further investigation, we found 42% of missed or mispredicted microbes were the *Actinoalloteichus* genus. (d) Precision (purple) and recall (green) for optimal model (*G-2E-0*). The precision increases and recall decreases as the confidence threshold increases, as expected. However, noticeably, there is a peak and decrease in precision at confidence threshold between 0.45 and 0.80, and precision never exceeds 0.65. (e) Average % characterized and average accuracy for 10 Kraken blind sets. Averages for DeepMicrobes are highly dependent on the confidence threshold. (f) DeepMicrobes precision and recall (*G-2E-0*) as a function of genus abundance, using the first blind data set as a representative example. Precision appears especially impacted by the genus abundance in the data set.

we found the highest count of reads predicted correctly was at 0 confidence, using the *G-2E-0* model (0.48 accuracy, 100% characterized reads) (Fig. 4a and b).

Next, using the *G-2E-0* model, we tested all 10 blind sets, and found that the accuracy and percent characterized had little variation (maximum standard deviation ±5%) (Fig. 4c). We found *G-2E-0* precision never exceeded 0.65 and recall 0.35 (Fig. 4d). The abundance was calculated by dividing the number of reads from a genus by the total number of reads in the data set. It appears that as genus abundance increases, the precision also increases. Interestingly, no pattern was discerned for recall, which indicates the model could detect microbial presence no matter the abundance (Fig. 4f). Notably, each of our models showed lower average percent characterized and accuracy, no matter the confidence threshold (Fig. 4a), than Kraken2 (0.92, 0.70).

DeepMicrobes performance on various microbial taxa

The original DeepMicrobes model was trained exclusively on human gut bacterial genomes, referred to as the HGR data set (22). Our results suggest that the DeepMicrobes marine (MarRef)-trained model showed lower accuracy than the prior human gut model (0.65 vs 0.96). Understanding possible explanations for this discrepancy is important to improve further DL models and to improve the data set preparation before model training. We calculated the MCC for each microbe in our test data set, with microbes below 0.5 MCC taken to be consistently mispredicted. All 10 genera mispredicted at confidence threshold 60 were correctly predicted at the family level (Fig. 5). Notably, in the *G-2E-60* model, 30% of misclassified microbes were assigned as the actinomycete

(a)

Species (3 epochs)	Count
< 0.5 MCC	16/1272
Micromonospora	4
Streptomyces	4

(b)

Species (2 epochs)	Count
< 0.5 MCC	42/1272
Micromonospora	4
Streptomyces	6

(c)

Genus (2 epochs)	Count
< 0.5 MCC	10/1272
Agarilytica	1
Amycolatopsis	1
Cognaticolwellia	1
Granulosicoccus	1
Marinobacter_A	1
Nocardiopsis	1
Rhodococcus_B	1
Altererythrobacter_D	1
Spirillospora	1
Zunongwangia	1

(d)

Ground truth	Prediction	Model	GC content	Length (bp)	# in MarRef
Agarilytica	Vibrio	Genus	40.9	6.9 M	1
Amycolatopsis	Streptomyces	Genus	70.7	9.1 M	1
Granulosicoccus	Vibrio	Genus	52.8	7.7 M	1
Nocardiopsis	Streptomyces	Genus	72.7	6.5 M	1
Saccharospirillum	Streptomyces	Genus	57.4	3.6 M	1
Zunongwangia	Gramella	Genus	36.22	5.1 M	1
Streptomyces	Species-level confusion	Species	72.2	7.7 M	16
Micromonospora	Species-level confusion	Species	72.9	6.8 M	6

(e)

Database	Mean # samples / genus	Mean GC content	Mean genome length (bp)
HGR	20.7	47.8	2.5 M
MarRef	3.1	48.5	4.1 M

FIG 5 DeepMicrobes results suggest that certain taxa are consistently poorly predicted. Number of species (a, b) and genera (c) with MCC <0.5; (a, b) show total number of species at the genus level. MCC is a balanced statistic we use to discern which microbes are consistently mispredicted. (d) Incorrectly predicted microbes assigned labels and other genomic data for *G-2E-60*. (e) Genomic data comparison between HGR and MarRef genomes. Notably, all genomes mispredicted for the genus models have only one representative genome in MarRef for the model to train on; many also have long genome length and high GC content. The species mispredictions have long genome length and high GC content.

genus *Streptomyces*. Similarly, for the species model, ~25% of misclassified microbes were predicted to be *S. albidoflavus*, and ~33% were predicted to be *Streptomyces* species. Reads that were mispredicted as *Streptomyces* did not have a discernable pattern in taxonomy but included *Amycolatopsis* and *Nocardiopsis* among others (Fig. 5).

DeepMicrobes performance was positively correlated with genome coverage in the training set

To evaluate whether performance differences between the DeepMicrobes gut and marine models was affected by the input genomes, we compared the HGR and MarRef data sets used for training. We noticed average genome size in the MarRef data set was larger than the average genome size in the HGR data set (4.1 vs 2.5 Mbp). However, the average number of genomes represented in each genus was lower than the MarRef data set compared to the HGR data set (3.1 vs 20.7) (Fig. 5). Moreover, we noticed microbes below 0.5 MCC in our *G-2E-60* model often had a genome size >6 Mbp, and all 10 mispredicted genera had only one genome/genus for training (Fig. 5). Lastly,

we observed that though the average GC content was similar between both database genomes (0.48), many MarRef genus with <0.5 MCC had high GC content (0.52–0.72).

We used the blind metagenomic data sets to test the effects of GC content, genome length, and number of genomes per genus on model accuracy. Notably, we saw no relationship in GC content. However, genome size and number of genomes per genus were negatively and positively correlated, respectively, with model accuracy, though the *R*-value was not high (0.199, 0.27) (Fig. 6a through c). We hypothesized that the genome subsampling in the training set (10,000 forward reads and 10,000 reverse reads) might not be adequate in capturing genetic variations in large genomes. To test this, we calculated the % genome coverage of the training set by mapping the trimmed simulated reads back to their reference genome. We identified a positive correlation between % coverage and model accuracy (*R*-value 0.40) (Fig. 6d). The larger, light blue outliers in Fig. 6d have low coverage but high accuracy and also tend to have more genome samples to train upon per genus. This suggests that increasing the number of genomes per genus in training can partially rectify any detrimental effects originating from a low genome coverage in training. The same analysis with Kraken2 results for the blind data sets (Fig. S3) found no clear impact from GC content, length of genome, number of genomes per genus, or genome coverage. Instead, Kraken2 results cluster most genera with a perfect classification of 1.0 accuracy, with a few genera having complete misprediction at 0.0 accuracy.

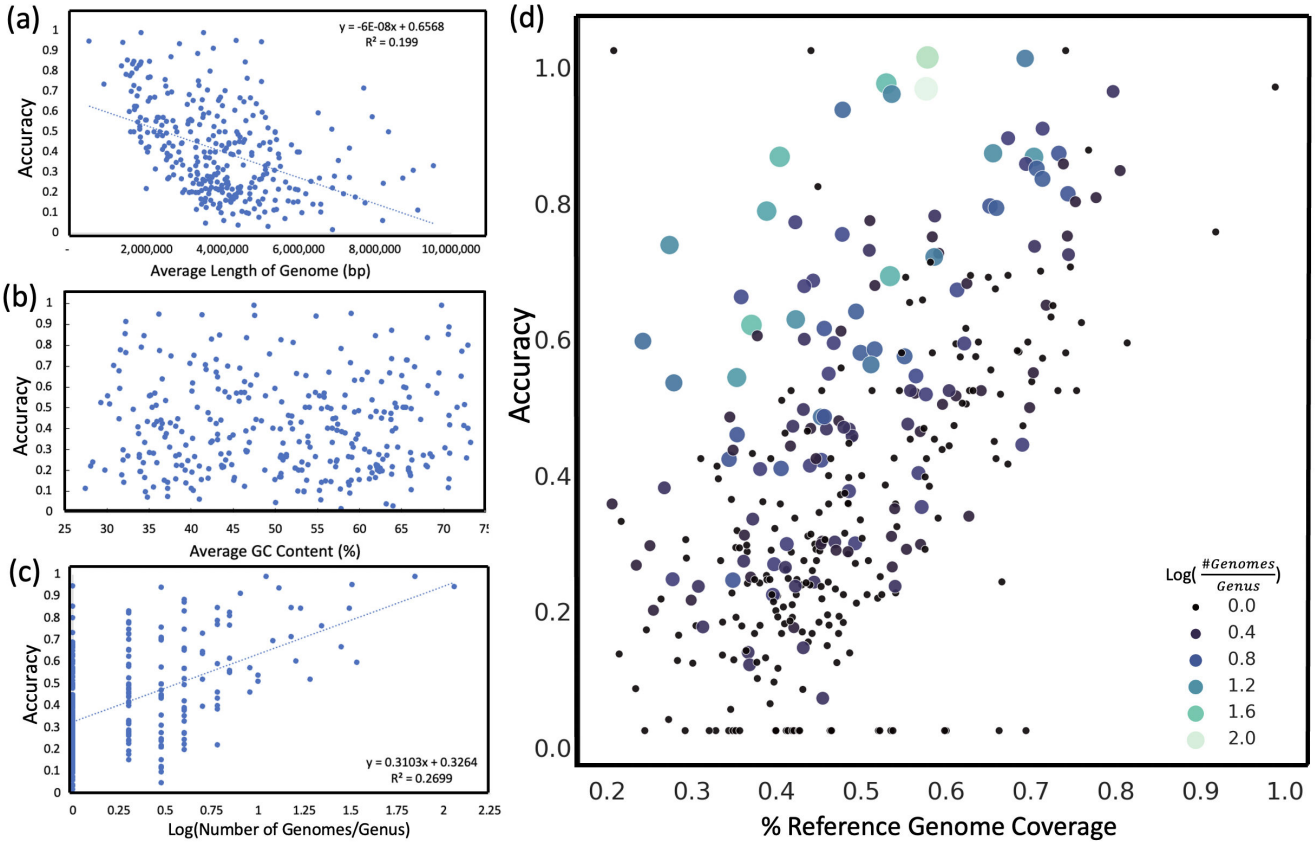


FIG 6 Accuracy of DeepMicrobes is positively associated with genome coverage and number of genomes per genus in training but not GC content. The color and size represent the average number of genomes per genus, and each point corresponds to a genus. Correlations between model accuracy and (a) genome length, (b) GC content, (c) number of genomes per genus, and (d) % genome coverage to reference genome. In (d), accuracy differences are described best by % genome coverage (*R*-value 0.4) and appear also to be impacted by the number of genomes per genus.

Initial benchmark testing using convolutional neural network ResNet models

A new ResNet CNN architecture was tested to measure if performance could be improved and to expand upon the functionality of ML in taxonomic classification. Like for DeepMicrobes, we used all 915 species classes for the CNN training, and randomly subsampled the data set to be 10% of reads per species because the full data set was too computationally burdensome to train a model in a reasonable amount of time. Our baseline ResNet-50 model achieved accuracy of only 0.3/0.01 for training/validation at three epochs, but the architecture with reduced size (ResNet-smaller-3) saw an elevated validation accuracy of 0.132 at two epochs (Fig. 7). Interestingly, after three epochs, ResNet-smaller-3 accuracy increased for training (0.21) but dropped for validation data sets (0.018). We also found that ResNet-smaller-3–2D-12w validation accuracy dropped from 0.132 to 0.116 between epochs two and three, while ResNet-smaller-3-1D-3w-alt maintained 0.132 accuracy for all three epochs. Notably, no model architecture could achieve accuracy higher than 0.132. Full results for all CNN models that we tested are given in Table S1. The decrease in accuracy seen in multiple models, similar to the DeepMicrobes results, suggests the models are being overfit, perhaps memorizing the training examples rather than learning how to correctly categorize reads for species identification.

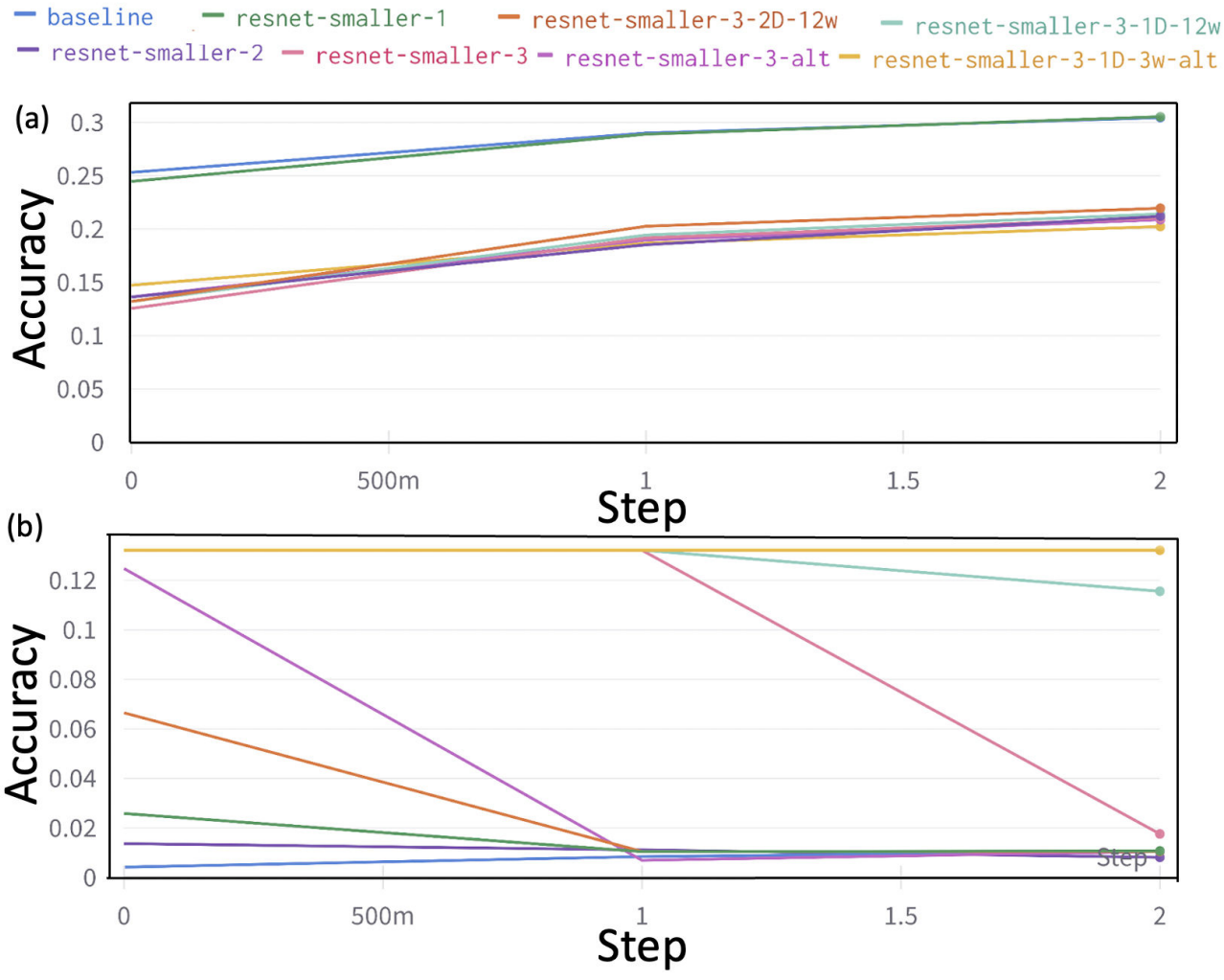


FIG 7 ResNet architecture results for all models using sample of MarRef genomes. (a) Training accuracy and (b) validation accuracy for different ResNet architectures.

DISCUSSION

This research explored whether DL can handle the complexity of marine metagenomes for read-based taxonomic classification. We found that Kraken2 outperforms each ML model developed in this study. This might be because Kraken2 computes k-mer composition for the entire genome, while DeepMicrobes' *in silico* subsampling only captures a fraction of the genome. Notably, our results suggest the microbe abundance impacts precision. Our model reaches high precision/recall (0.92/0.9) at confidence threshold 60 for the testing data, which contained only reads from individual microbial genomes, but does not surpass 0.65/0.35 precision/recall on the blind metagenomics data containing microbial reads from various genomes in varying abundances.

Our results indicate DeepMicrobes disproportionately misclassifies reads to be *Streptomyces*. *Streptomyces* genomes are long, with twice as many genes as *Escherichia coli* or *Bacillus subtilis* (33, 34). Moreover, *Streptomyces* acquire diverse gene clusters from their environment, and gene transfer between species is prolific (33, 34). Indeed, *Streptomyces albidoflavus* has a 16S rRNA gene sequence that is 100% identical to that of 10 other species of *Streptomyces*, and there are proposals to reclassify this species into multiple new species (35). Taken together, this suggests that DeepMicrobes cannot discern between reads from *Streptomyces* and bacteria whose genomes they have accumulated, indicating that bacteria with high levels of horizontal gene transfer might be difficult for a ML model to train on.

We also explored how differences in GC content, genome length, number of genomes per genus, and percent genome coverage affects model accuracy and found positive correlation between percent genome coverage and accuracy. Further analysis on the CNN training set indicates that one of the 914 species we tested represents 13.2% of the training data. Notably, 13.2% is also the highest accuracy our CNNs achieved, suggesting that our model is impacted by class imbalance, and is memorizing the one species with disproportionately high reads in the training data. Class imbalance in input data for machine learning can severely affect model accuracy because the model may "anchor" itself to overrepresented classes during training (36). The model then has high recognition of overrepresented classes but mispredicts underrepresented classes during testing (37).

In order to explore and visualize the class imbalance in our data set, we created a confusion matrix for each species' TP, FP, TN, and FN for the DeepMicrobes model at a 60 confidence interval (Table S2). Because the matrix is quite large, we also calculated a summation of the TP, FP, TN, and FP by phylum (34 total phyla). From these two tables, it is clear that certain microbes far overrepresent the data set, with counts in the tens of thousands, while other microbes have fewer than 10 instances. We observed that our CNN models were affected by class imbalance, which compounded the apparent issues in low genome coverage for certain species. Therefore, in future tests, a balanced training set with higher numbers of representative genomes per species/genus may improve the model's performance, albeit at the cost of computational resources.

Unexpected time and compute power hurdles

Time and compute power are important factors in the implementation of read classification tools and were unanticipated impediments to this research. For example, a FASTA to tfrecord conversion took 3–4 h (one genome, 10,000 reads/genomes) and training ranged 2.5–4 days (1,272 genomes, 10,000 reads/genomes) for one epoch, and prediction scaled with metagenome contig size, preventing prediction on big, complex metagenomes. Moreover, the DeepMicrobes software often used 15–38 CPU cores despite multiple attempts to constrain its usage. In contrast, Kraken2 was usually finished within 30 min for prediction and required no training.

We used our new CNN models to test alternative, faster data processing techniques. For the CNN model, we used the NVIDIA Rapids.ai library for GPU access and easy export of data in universal file formats (38). Specifically, the Rapids.ai suite uses a combination of Rapids Dask and Rapids cuDF to leverage a GPU and speed the preprocessing

time by 85% over standard CPUs used by most libraries. Notably, significant lifts in data preprocessing speeds were seen because the input data set is quite large and the Rapids.ai library speed improvements correlate with data size. In addition, using Rapids.ai avoided the time-consuming conversion step from FASTA to tfrecord required by DeepMicrobes TensorFlow libraries. Instead, we found the fastest process was to convert the FASTA files into parquet format during preprocessing, and then exported the data into universal csv format for model training or testing.

Future promise and potential of deep learning: are Transformers the way?

Overall, compared to DeepMicrobes, ResNet CNN models saw promise in read-based taxonomic classification, especially in terms of data processing speed-ups. However, CNN architectures are typically used for image recognition and therefore are perhaps less suitable for genomic classification than architectures that address NLP problems such as Transformer models. Transformers are able to identify meaningful patterns of sequence data and leverage parallelization to avoid recursion problems to allow for faster scaling to large data sets in training. Transformer models have shown success in computational genetics, including genome annotation (39), sequence recognition (40), and in isolation of genomic features from metadata (41). Additionally, the Transformer-XL model can learn dependencies beyond a fixed length, which could increase upon the fixed 150 bp length used here. One challenge with the application of Transformers to large sequence data sets is the number of "tokens," the word embeddings that make up the vocabulary for the model to train with. Typically, it is recommended that a Transformer model not exceed ~50,000 tokens, which would restrict our token (k-mer) length to 8-mers, preventing use of smaller and finer-grain token sizes (i.e., 4^8 = 65,536 tokens). Additionally, in a typical NLP problem, the "words" and the letters in a word are pre-determined in the English language. However, in the context of the genome, which DNA sequence or length creates meaningful "sentences"s unknown, which makes result validation more challenging. Nevertheless, we suggest that in order to obtain the full benefits of ML and improve upon standard tools, further research should focus on Transformers in classification.

Conclusion

A goal for this research was to develop a read-based taxonomic classification deep learning model for marine metagenomics data that might be competitive with existing tools that rely on a curated taxonomic tree. We found that DeepMicrobes can successfully identify microbial species at high accuracy when characterizing reads from individual genomes, but it cannot reach the same accuracy as Kraken2 for complex metagenomics data. Despite this, the model shows promise by reaching 60% accuracy, even though the input genome reads often only cover a fraction of the reference genomes. Future efforts can be made to develop a well-balanced training data set with high genome coverage, refine the model, and develop alternative models like the Transformer model. More generally, ML approaches are still just beginning to be applied to marine omics and environmental DNA data sets, with room for development through sustained interdisciplinary efforts.

AUTHOR AFFILIATIONS

[1]Center for Synthetic and Systems Biology, School of Life Sciences, Tsinghua-Peking Center for Life Sciences, Tsinghua University, Beijing, China
[2]EPSRC/BBSRC Future Biomanufacturing Research Hub, EPSRC Synthetic Biology Research Centre SYNBIOCHEM Manchester Institute of Biotechnology and School of Chemistry, The University of Manchester, Manchester, United Kingdom
[3]Cooperative Institute for Marine and Atmospheric Studies, Rosenstiel School of Marine, Atmospheric, and Earth Science, University of Miami, Miami, Florida, USA

[4]Ocean Chemistry and Ecosystems Division, Atlantic Oceanographic and Meteorological Laboratory, National Oceanic and Atmospheric Administration, Miami, Florida, USA
[5]College of Marine Science, University of South Florida, St Petersburg, Florida, USA
[6]Run:AI, Office of the CTO, Tel Aviv, Israel
[7]Deloitte Consulting LLP, Biomedical Data Science Team, Arlington, Virginia, USA
[8]Department of Vertebrate Zoology, National Museum of Natural History, Smithsonian Institution, Northwest, Washington, DC, USA
[9]Harte Research Institute, Texas A&M University-Corpus Christi, Corpus Christi, Texas, USA
[10]Southwest Fisheries Science Center, Antarctic Ecosystem Research Division, National Oceanic and Atmospheric Administration, La Jolla, California, USA
[11]Department of Computational Biology and Medical Sciences, Graduate School of Frontier Sciences, The University of Tokyo, Tokyo, Japan
[12]NOAA/US Integrated Ocean Observing System (IOOS), Silver Spring, Maryland, USA
[13]Northern Gulf Institute, Mississippi State University, Mississippi, USA

AUTHOR ORCIDs

Luke R. Thompson http://orcid.org/0000-0002-3911-1280

FUNDING

Funder	Grant(s)	Author(s)
National Oceanic and Atmospheric Administration	NA21OAR4320190	Luke R. Thompson
DOC | National Oceanic and Atmospheric Administration (NOAA)	NA20OAR4320472	Shen Jean Lim

AUTHOR CONTRIBUTIONS

Helen Park, Data curation, Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review and editing | Shen Jean Lim, Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Writing – review and editing | Jonathan Cosme, Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review and editing | Kyle O'Connell, Conceptualization, Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review and editing | Jilla Sandeep, Conceptualization, Data curation, Formal analysis, Investigation, Methodology | Felimon Gayanilo, Conceptualization, Formal analysis, Investigation, Methodology, Writing – review and editing | George R. Cutter Jr., Formal analysis, Investigation, Methodology | Enrique Montes, Writing – review and editing | Chotinan Nitikitpaiboon, Data curation, Formal analysis, Investigation, Methodology | Sam Fisher, Formal analysis, Investigation | Hassan Moustahfid, Conceptualization, Funding acquisition, Project administration, Resources, Supervision | Luke R. Thompson, Conceptualization, Methodology, Project administration, Supervision, Writing – review and editing

DATA AVAILABILITY

All scripts and commands used for testing, training, and blind datasetdata set creation are available at DeepMetagenomics. Jupyter Notebooks used to create figures and analyze datasetsdata sets, along with shell scripts to run DeepMicrobes, Bowtie2, CAMISIM, and other program code can be found at MarRef_DeepMicrobes. The CNN ResNet notebooks can be found at GitHub. Instructions for training and testing the model can be found in the DeepMicrobes GitHub repository. All training, testing, and blind datasetsdata sets and labels used for machine learning algorithms for taxonomic classification can be found at 7429932.

ADDITIONAL FILES

The following material is available online.

Supplemental Material

Supplemental material FOR publication (Spectrum05237-22-s0001.pdf). Supplemental material FOR publication

Open Peer Review

PEER REVIEW HISTORY (review-history.pdf). An accounting of the reviewer comments and feedback.

REFERENCES

1. Audzijonyte A, Pethybridge H, Porobic J, Gorton R, Kaplan I, Fulton EA, Poisot T. 2019. Atlantis: a spatially explicit end-to-end marine ecosystem model with dynamically integrated physics, ecology and socio-economic modules. Methods Ecol Evol 10:1814–1819. https://doi.org/10.1111/2041-210X.13272
2. Goodwin M, Halvorsen KT, Jiao L, Knausgård KM, Martin AH, Moyano M, Oomen RA, Rasmussen JH, Sørdalen TK, Thorbjørnsen SH, Demer D. 2022. Unlocking the potential of deep learning for marine ecology: overview, applications, and outlook. ICES J Mar Sci 79:319–336. https://doi.org/10.1093/icesjms/fsab255
3. Kim D, Song L, Breitwieser FP, Salzberg SL. 2016. Centrifuge: rapid and sensitive classification of metagenomic sequences. Genome Res 26:1721–1729. https://doi.org/10.1101/gr.210641.116
4. Menzel P, Ng KL, Krogh A. 2016. Fast and sensitive taxonomic classification for metagenomics with kaiju. Nat Commun 7:11257. https://doi.org/10.1038/ncomms11257
5. Pierce NT, Irber L, Reiter T, Brooks P, Brown CT. 2019. Large-scale sequence comparisons with sourmash. F1000Res 8:1006. https://doi.org/10.12688/f1000research.19675.1
6. Wood DE, Lu J, Langmead B. 2019. Improved metagenomic analysis with kraken 2. Genome Biol 20:257. https://doi.org/10.1186/s13059-019-1891-0
7. Mathieu A, Leclercq M, Sanabria M, Perin O, Droit A. 2022. Machine learning and deep learning applications in metagenomic taxonomy and functional annotation. Front Microbiol 13:811495. https://doi.org/10.3389/fmicb.2022.811495
8. Yang J, Pu J, Lu S, Bai X, Wu Y, Jin D, Cheng Y, Zhang G, Zhu W, Luo X, Rosselló-Móra R, Xu J. 2020. Species-level analysis of human gut microbiota with metataxonomics. Front Microbiol 11:2029. https://doi.org/10.3389/fmicb.2020.02029
9. Thompson LR, Sanders JG, McDonald D, Amir A, Ladau J, Locey KJ, Prill RJ, Tripathi A, Gibbons SM, Ackermann G, Navas-Molina JA, Janssen S, Kopylova E, Vázquez-Baeza Y, González A, Morton JT, Mirarab S, Zech Xu Z, Jiang L, Haroon MF, Kanbar J, Zhu Q, Jin Song S, Kosciolek T, Bokulich NA, Lefler J, Brislawn CJ, Humphrey G, Owens SM, Hampton-Marcell J, Berg-Lyons D, McKenzie V, Fierer N, Fuhrman JA, Clauset A, Stevens RL, Shade A, Pollard KS, Goodwin KD, Jansson JK, Gilbert JA, Knight R, Earth Microbiome Project Consortium. 2017. A communal catalogue reveals earth's multiscale microbial diversity. Nature 551:457–463. https://doi.org/10.1038/nature24621
10. Goodfellow I, Bengio Y, Courville A. 2016. Genetic programming and Evolvable machines. In Deep learning. The MIT press. https://doi.org/10.1007/s10710-017-9314-z
11. Sarker IH. 2021. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Comput Sci 2:420. https://doi.org/10.1007/s42979-021-00815-1
12. Poplin R, Chang P-C, Alexander D, Schwartz S, Colthurst T, Ku A, Newburger D, Dijamco J, Nguyen N, Afshar PT, Gross SS, Dorfman L, McLean CY, DePristo MA. 2018. A universal SNP and small-Indel variant caller using deep neural networks. Nat Biotechnol 36:983–987. https://doi.org/10.1038/nbt.4235
13. Reiman D, Metwally AA, Sun J, Dai Y. 2020. Popphy-CNN: a phylogenetic tree embedded architecture for convolutional neural networks to predict host phenotype from metagenomic data. IEEE J Biomed Health Inform 24:2993–3001. https://doi.org/10.1109/JBHI.2020.2993761
14. Al-Ajlan A, El Allali A. 2019. CNN-MGP: convolutional neural networks for metagenomics gene prediction. Interdiscip Sci 11:628–635. https://doi.org/10.1007/s12539-018-0313-4
15. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L. 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8:53. https://doi.org/10.1186/s40537-021-00444-8
16. Daoud HK, Hren M, Curk T. 2022. NLP-Basedclassificationofsoftware-toolsfor Metagenomics sequencing data analysis into EDAM semantic annotation. arXiv. arXiv,. https://doi.org/10.48550/arXiv.2210.00831
17. Matougui B, Boukelia A, Belhadef H, Galiez C, Batouche M. 2021. NLP-metaxa: a natural language processing approach for metagenomic taxonomic binning based on deep learning. CBIO 16:992–1003. https://doi.org/10.2174/1574893616666210621101150
18. Kishk A, Elzizy A, Galal D, Razek EA, Fawzy E, Ahmed G, Gawish M, Hamad S, El-Hadidi M. 2018. "A hybrid machine learning approach for the Phenotypic classification of Metagenomic colon cancer reads based on Kmer frequency and biomarker profiling" 2018 9th Cairo International Biomedical Engineering Conference (CIBEC); Cairo, Egypt. https://doi.org/10.1109/CIBEC.2018.8641805
19. Fiannaca A, La Paglia L, La Rosa M, Lo Bosco G, Renda G, Rizzo R, Gaglio S, Urso A. 2018. Deep learning models for bacteria taxonomic classification of metagenomic data. BMC Bioinformatics 19:198. https://doi.org/10.1186/s12859-018-2182-6
20. Fioravanti D, Giarratano Y, Maggio V, Agostinelli C, Chierici M, Jurman G, Furlanello C. 2018. Phylogenetic convolutional neural networks in metagenomics. BMC Bioinformatics 19:49. https://doi.org/10.1186/s12859-018-2033-5
21. Crist DT, O'Dor R. 2013. Census of marine life, p 1–22. In Levin SA (ed), Encyclopedia of Biodiversity, Second Edition. Academic Press, Waltham.
22. Liang Q, Bible PW, Liu Y, Zou B, Wei L. 2020. Deepmicrobes: taxonomic classification for metagenomics with deep learning. NAR Genom Bioinform 2:lqaa009. https://doi.org/10.1093/nargab/lqaa009
23. He K, Zhang X, Ren S, Sun J. 2016. Deep residual learning for image recognition.and pattern recognition. https://doi.org/10.1109/CVPR.2016.90
24. Marine Metagenomics portal. 2021. MarRef. Available from: https://mmp2.sfb.uit.no/marref/
25. Klemetsen T, Raknes IA, Fu J, Agafonov A, Balasundaram SV, Tartari G, Robertsen E, Willassen NP. 2018. The MAR databases: development and implementation of databases specific for marine metagenomics. Nucleic Acids Res 46:D692–D699. https://doi.org/10.1093/nar/gkx1036
26. Parks DH, Chuvochina M, Rinke C, Mussig AJ, Chaumeil P-A, Hugenholtz P. 2022. GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy. Nucleic Acids Res 50:D785–D794. https://doi.org/10.1093/nar/gkab776
27. GTDB-genome Taxonomy database. 2023. Available from: https://gtdb.ecogenomic.org
28. Fritz A, Hofmann P, Majda S, Dahms E, Dröge J, Fiedler J, Lesker TR, Belmann P, DeMaere MZ, Darling AE, Sczyrba A, Bremges A, McHardy AC. 2019. CAMISIM: Simulating Metagenomes and microbial communities. Microbiome 7:17. https://doi.org/10.1186/s40168-019-0633-6
29. Ye SH, Siddle KJ, Park DJ, Sabeti PC. 2019. Benchmarking metagenomics tools for taxonomic classification. Cell 178:779–794. https://doi.org/10.1016/j.cell.2019.07.010

30. Chicco D, Jurman G. 2020. The advantages of the matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21:6. https://doi.org/10.1186/s12864-019-6413-7
31. Lu J, Breitwieser FP, Thielen P, Salzberg SL. 2017. Bracken: estimating species abundance in metagenomics data. PeerJ Comput Sci 3:e104. https://doi.org/10.7717/peerj-cs.104
32. SAMtools:Get breadth of coverage. 2022. Available from: https://www.metagenomics.wiki/tools/samtools/breadth- of-coverage
33. Hopwood DA. 2022. Highlights of Streptomyces Genetics. Heredity 123:23–32. https://doi.org/10.1038/s41437-019-0196-0
34. Mcdonald BR, Currie CR. 2017. Lateral gene transfer Dynamics in the ancient bacterial genus *Streptomyces*. MBio 8. https://doi.org/10.1128/mBio.00644-17
35. Rong X, Guo Y, Huang Y. 2009. Proposal to reclassify the *Streptomyces albidoflavus* clade on the basis of multilocus sequence analysis and DNA–DNA hybridization, and taxonomic elucidation of *Streptomyces griseus* subsp. *solvifaciens*. Syst Appl Microbiol 32:314–322. https://doi.org/10.1016/j.syapm.2009.05.003
36. Johnson JM, Khoshgoftaar TM. 2019. Survey on deep learning with class imbalance. J Big Data 6. https://doi.org/10.1186/s40537-019-0192-5
37. Buda M, Maki A, Mazurowski MA. 2018. A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw 106:249–259. https://doi.org/10.1016/j.neunet.2018.07.011
38. Hricik T, Bader D, Green O. "Using RAPIDS AI to accelerate graph data science workflows" 2020 IEEE High Performance Extreme Computing Conference (HPEC); Waltham, MA, USA, p 1–4. https://doi.org/10.1109/HPEC43674.2020.9286224
39. Clauwaert J, Waegeman W. 2022. Novel transformer networks for improved sequence labeling in genomics. IEEE/ACM Trans Comput Biol Bioinform 19:97–106. https://doi.org/10.1109/TCBB.2020.3035021
40. Clauwaert J, Menschaert G, Waegeman W. 2021. Explainability in transformer models for functional genomics. Brief Bioinform 22:bbab060. https://doi.org/10.1093/bib/bbab060
41. Serna Garcia G, Leone M, Bernasconi A, Carman MJ. 2022. Gemi: interactive interface for transformer-based genomic metadata integration. Database 2022:baac036. https://doi.org/10.1093/database/baac036