ENHANCING DOWNSTREAM ANALYSIS IN GENOME SEQUENCING: SPECIES CLASSIFICATION WHILE BASECALLING

Riselda Kodra
Swiss Federal Institute of Technology
1008 Lausanne, Switzerland
riselda.kodra@epfl.ch

Hadjer Benmeziane, Irem Boybat, William Andrew Simon
IBM Research Zurich
8803 Rüschlikon, Switzerland
{hadjer.benmeziane, ibo, william.simon1}@ibm.com

ABSTRACT

The ability to quickly and accurately identify microbial species in a sample, known as metagenomic profiling, is critical across various fields, from healthcare to environmental science. This paper introduces a novel method to profile signals coming from sequencing devices in parallel with determining their nucleotide sequences, a process known as basecalling, via a multi-task deep neural network for simultaneous basecalling and multi-class genome classification. We introduce a new multi-objective loss strategy where basecalling and classification losses are back-propagated separately, with model weights combined for the shared layers, and a pre-configured ranking strategy allowing top-K species accuracy, giving users flexibility to choose between higher accuracy or lower latency at identifying the species. We achieve state-of-the-art basecalling accuracies, while multi-class classification accuracies meet and exceed the results of state-of-the-art binary classifiers, attaining an average of 92.5%/98.9% accuracy at identifying the top-1/3 species among a total of 17 genomes in the Wick bacterial dataset. This work has implications for future studies in metagenomic profiling by accelerating the bottleneck step of matching the DNA sequence to the correct genome.

1 INTRODUCTION

The recent falling cost of genome sequencing (AccessWire, 2022) has led to increased usage across a range of fields (Alvarez-Cubero et al., 2017; Cruz-Silva et al., 2023; LaPierre et al., 2020). In particular, metagenomics, the measurement of relative abundances (Quince et al., 2017), enables contextualization of a single genome within its community (Handelsman, 2004).

Alignment-based metagenomic profiling has been demonstrated to perform well in both precision (false-positive rate) and recall (false-negative rate) at the cost of drastically increased memory and computational requirements (LaPierre et al., 2020), as each read must be aligned against a vast database of possible genomes. Methods for reducing these runtime requirements while maintaining high profiling accuracy are thus attractive research opportunities.

In parallel to the above research, pre-basecalling analysis is being pursued by many research groups. Basecalling, preceding alignment in the processing pipeline, translates raw signal samples, or reads, into a chain of nucleotide bases. Inferring these k-mer sequences can be accomplished via Deep Neural Networks (DNNs), which provide State of the Art (SotA) speed and accuracy in comparison to previous methods (Boža et al., 2017). Despite this, the basecalling step is still commonly the bottleneck in the analysis pipeline, consuming up to 40% of runtime even running on a GPU (Simon et al., 2025). Therefore, much research has focused on eliminating unnecessary basecalling on non-target reads by identifying and rejecting them early (Cavlak et al., 2022; Dunn et al., 2021; Kovaka et al., 2021). Most of these methods rely on DNNs to identify a single target genome, i.e. human, and reject all others, making them binary classifiers.

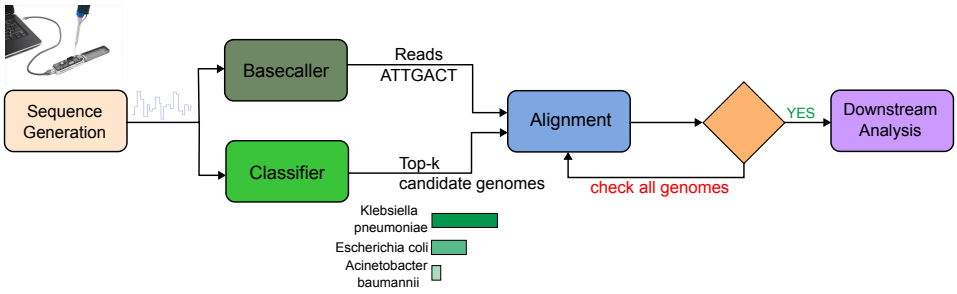

Figure 1: Proposed genome sequencing pipeline with species classification while basecalling.

An intuitive extension of these read classifiers would be to move towards multi-class classification, where a read is categorized amongst a pool of possible genomes.By narrowing down the possible genomes early in the process, computational requirements are significantly reduced compared to traditional metagenomic profiling methods.
In this work, we present the first multi-task deep neural network for multi-class classification and basecalling to reduce downstream alignment overhead. To accomplish this:

- We augment the traditional Bonito DNN basecaller (Wright, 2020) with a classification layer, enabling classification while basecalling.
- We develop a custom multi-objective loss strategy that combines basecalling and classification losses, giving more weight to classification predictions made later in the process when the model has seen more sequence data and is more confident.
- We propose a pre-configured ranking strategy, where the top-K predicted classes are passed to the next stages of the genome sequencing pipeline, allowing flexible trade-offs between accuracy and computational efficiency. A consensus-based testing metric is used to assess the final classification accuracy.
- We train our network on a set of 17 genomes, demonstrating between 92.5%/98.89% top-1/top-3 per-read classification accuracy without degrading basecalling accuracy.

2 Background

For extracting nucleotide sequences of and performing analysis on DNA/RNA samples, Oxford Nanopore Technologies (ONT) (ONT, 2024b) utilizes flow cells, consisting of nanoscopic pores which produce electrical signals as molecules pass through. The MinION device is a long read device, producing samples up to millions of bases long (ONT, 2015), enhancing downstream accuracy (ONT, 2024b) after passing through the downstream processing steps.

2.1 Basecalling Algorithms

Basecalling algorithms are responsible for converting the raw electrical signal into sequences of nucleotide bases representing the original DNA or RNA molecule. ONT relies on DNNs for basecalling, offering scalability and efficient handling of large datasets, recognition of complex patterns, and superior performance to prior approaches (Heather & Chain, 2015). The DNN commonly consists of networks such as CNNs acting as feature extractors and an inference trunk, such as LSTMs or transformers, with a final fully connected layer. The DNN output is processed by a Conditional Random Field (CRF)/Connectionist Temporal Classifier (CTC), which handles well the time variant nature of the raw electrical signals (ONT, 2024a). In our study, we utilize the Bonito DNN within the framework described in (Paga, 2023a), since this model is the standard provided by ONT and demonstrates high basecalling performance based on the metrics outlined in the framework's associated paper (Pagès-Gallego & de Ridder, 2023).

2.2 Genome Classification

Classification is the process of identifying to which genome a read belongs. Typically, this is performed post-basecalling; however, several studies have explored the possibility of pre-basecalling classification. Specifically, binary classification is studied across a variety of works, often exploring algorithms to support the "Read-Until" (i.e. early read ejection) option of the nanopore sequencer. This feature refers to the capability of the sequencing device to discard a partially sequenced molecule

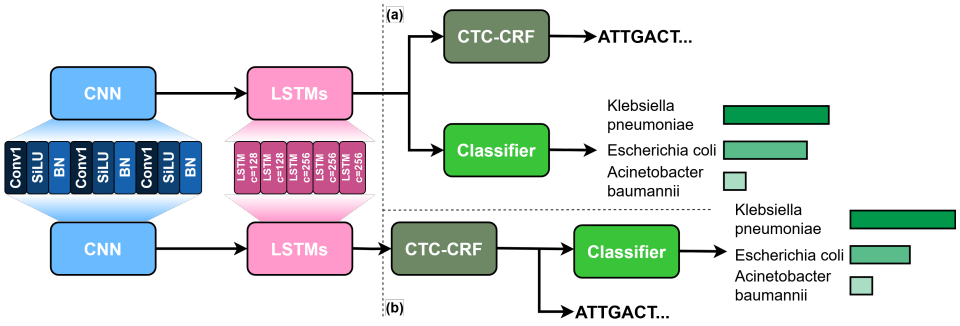

Figure 2: Proposed parallel (a) and serial (b) models for the task of classification while basecalling.

deemed off-target (ONT, 2020). SquiggleNet (Bao et al., 2021), DeepSelectNet (Senanayake et al., 2023) and TargetCall (Cavlak et al., 2022) are examples of previous works which utilize DNNs to classify whether a genome belongs to a target species directly from the electrical signal.
While the aforementioned works perform well in applications where binary classification is sufficient, tasks such as metagenomic profiling where the identification of the genome requires comparisons to more than one species would benefit from a multi-class classifier, as this work presents.

3 Pre-Classification while Basecalling

The novelty of this work is to predict during the basecalling step, i.e. during sequencing, to which species a read belongs, as illustrated in Fig. 1. By having a preliminary candidate or K candidates for classification, the database to which the read must be matched can be reduced to 1 or K species, reducing classification latency and computational overhead. Metalign previously demonstrated how pre-filtering the database for which reads to align to improves throughput while maintaining balance between precision and recall (LaPierre et al., 2020), a concept this paper extends. By reducing the processing time of this method, the entire pipeline of metagenomic profiling will be enhanced, raising the standard for this and other methods in terms of the accuracy vs. latency trade-off.

3.1 DNN-based basecaller model architecture

The proposed model architecture is based on the existing Bonito (Wright, 2020) basecaller, but can be applied to any DNN basecalling network. Bonito, illustrated in Fig. 2, consists of 3 convolutional layers followed by 5 LSTM layers and a fully connected layer F_0. The CRF-CTC decoder is fed the output of the fully connected layer to produce the sequence of nucleotides. The classification portion of the network consists of a newly added fully connected layer whose output size is equal to the number of species to be classified against.

3.2 Model architectures for classification while basecalling

While the Bonito model is employed for executing the basecalling step, the framework from (Paga, 2023a) is extended in two ways by choosing where to add the aforementioned classification fully connected layer, either in parallel or in series with F_0. In the parallel approach, Fig. 2(a), the backpropagation in the classifier is independent from the basecaller's decoder. In the second approach displayed in Fig. 2(b), the loss of the classifier includes the decoder, creating a dependent relationship between them during the backward pass.

3.3 Training for basecalling/classification

During the training and optimization of the network, separate losses are calculated for the CRF-CTC block and the classifier block. Basecalling loss is calculated from the CRF-CTC decoding via the *seqdist* library (Studer et al., 2024). For the classification loss, CrossEntropy is applied without reduction, maintaining individual loss values for each element. The obtained result contains loss values for all the time-steps, representing the prediction of the species at each base. Since the model is necessarily less confident of its prediction at earlier time-steps in comparison to later, a scaling factor between 0 and 1 is applied to each time-step, with the average of the scaled loss across all time-steps taken as the final loss. Explorations were made with various scaling factor functions, with the logarithmic scaling factor resulting in best performance.

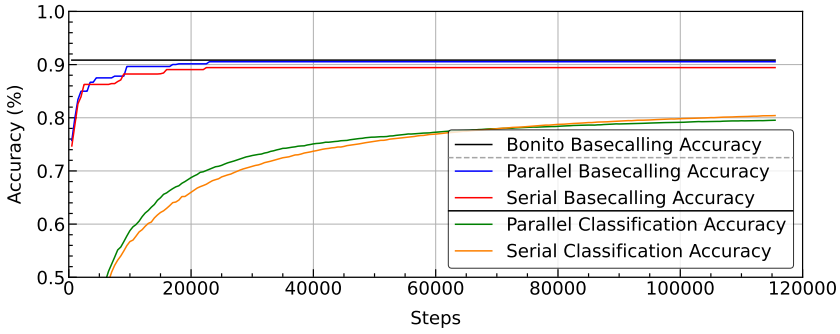

Figure 3: Validation accuracies for basecalling and top-1 classification for the parallel and serial model architectures.

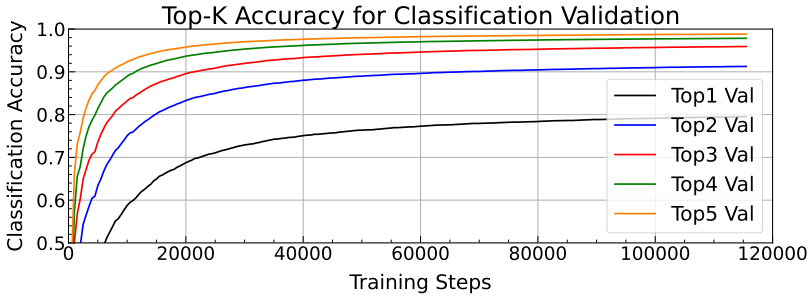

Figure 4: Top-K classification accuracy evolution during training of parallel model architecture.

Basecalling and classification loss are back propagated independently through their respective linear layers, then summed and back propagated through layers shared by both loss contributors. The classifier and the decoder thus contribute equally to the CNN and LSTM portions of the networks. Classifier loss is also back propagated through F_0 in the case of the serial implementation. Section 4.1 discusses the accuracy impact of the parallel vs. series architectures.

We evaluate the training process via a validation step every 500 batches. Basecalling accuracy is calculated by an alignment score for each basecalled read using the parasail library (Daily, 2016). For calculating classification accuracy, parametric top-K MulticlassSo Accuracy (PyTorch, 2022) from PyTorch is used. The last timestep's prediction is passed to that metric along with the correct species label. Different values of K are studied to analyze the trade-off between prediction accuracy and reduced computational complexity during downstream analysis.

4 Experiments and Results

In our experiments we basecall and classify 17 species from the Wick dataset, prepared as described in Appendix A.1.

4.1 Parallel vs. Serial model architecture classification accuracy studies

Fig. 3 shows the top-1 classification accuracies of the parallel and serial model architectures over 17 epochs. Classification accuracies using both architectures are similar, around $\sim$80%. While the parallel model architecture improves faster in the first 40,000 steps, both networks converge towards identical values.

4.2 Impact on basecalling accuracy

We compare the basecalling validation accuracies of the new models during 15 epochs with the original Bonito basecaller model trained on the same dataset in Fig. 3. Both model architectures exhibit similar trends, with the parallel model slightly outperforming the serial, approaching the baseline accuracy within $<$0.5%. The serial basecalling accuracy suffers as its fully connected layer is affected by the loss of the classifier output, while classifier accuracy does not significantly benefit from the extra fully connected layer in its pipeline.

The proposed models' post-alignment accuracy is also evaluated as detailed in Appendix A.2.1.

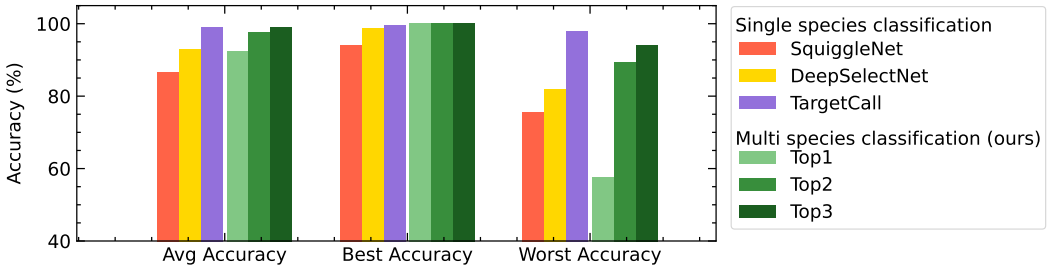

Figure 5: The proposed basecaller/classifier model acheives SotA classification accuracy even while classifying between multiple species, against the single species classification of the SotA works.

4.3 TOP-K PER-CHUNK ACCURACY

Fig. 4 illustrates the evolution of top-K classification accuracy during training for the parallel model architecture. It can be observed that top-1 classification saturates around 80%, while it approaches 99% as the top-K is increased up to 5. This configuration of top-K accuracy indicates flexibility when integrating the classifier model in downstream pipelines, as discussed in Section A.3. We note that the choice of K does not necessitate retraining of the network and can be chosen after training dependent on downstream pipeline requirements.

4.4 PER-READ CLASSIFICATION ACCURACY

While Section 4.3 reports per-chunk classification accuracy, classification of an entire read consisting of multiple chunks is determined by a consensus-based approach, where predictions are made for individual chunks of each read, and the read is classified as the species with the highest vote amongst its chunks. The top-1 overall accuracy is calculated as the ratio of correctly classified reads to the total number of reads, using 500 reads per species for evaluation. For calculating top-K accuracy, top-K species with highest number of classified reads are included in the ratio, if the correct species is amongst those top-K species. The results for K=1-3 of this approach for the 17 unique species included in both training and testing are displayed in Fig. 5, alongside SotA binary classifiers. On average our model's multi-class accuracy meets and exceeds that of single-class networks SquiggleNet and DeepSelectNet in all configurations and matches TargetCall with top-3 classification.
Our model achieves SotA classification accuracy while also extending the functionality from binary to multi-class classification, all within the original basecalling framework and without introducing a classifier-specific DNN.

5 CONCLUSION

In summary, this study demonstrates how a DNN-based basecaller like Bonito can be expanded with a classification layer in two possible architectures, parallel and serial. A tailored multi-objective loss method is developed which encapsulates the basecalling and classification loss. While for basecalling, the prediction of bases at each time-step contributes equally to the loss calculation, for classification, the predictions in the later stages carry more weight than the initial ones. The model is trained on a set of 17 genomes. During testing, for the generated sequences, an alignment score is produced using a read mapper and their reference genome, and the classification accuracy is obtained using a consensus-based approach which is implemented to produce a per-read prediction from the per-chunk predictions. Both of the tasks prove to be successful in achieving high accuracies, e.g. 90% for the basecaller and an average accuracy of 92.5% for top-1 classification and 98.89% for top-3 classification. These classification results will help speed up species identification in the metagenomic profiling pipeline by reducing the amount of required genome comparisons.

ACKNOWLEDGEMENTS

This work was supported by European Union's Horizon Europe Research and Innovation Program (BioPIM, Grant 101047160), and Swiss State Secretariat for Education, Research and Innovation (SERI) (Grant 22.00076).

REFERENCES

AccessWire. Dna sequencing costs: Data from the nhgri genome sequencing program (gsp), 2022. https://www.accesswire.com/695260/ONT-Shows-New-High-Accuracy-High-Output-Chemistry.

Maria Jesus Alvarez-Cubero, Maria Saiz, Belén Martínez-García, Sara M. Sayalero, Carmen Entrala, Jose Antonio Lorente, and Luis Javier Martinez-Gonzalez. Next generation sequencing: an application in forensic sciences? *Annals of Human Biology*, 2017.

Yuwei Bao, Jack Wadden, John R Erb-Downward, Piyush Ranjan, Weichen Zhou, Torrin L McDonald, Ryan E Mills, Alan P Boyle, Robert P Dickson, David Blaauw, and Joshua D Welch. SquiggleNet: real-time, direct classification of nanopore signals. *Genome Biology*, 2021.

Vladimír Boža, Broňa Brejová, and Tomáš Vinař. Deepnano: Deep recurrent neural networks for base calling in minion nanopore reads. *PLOS ONE*, 2017.

Meryem Banu Cavlak, Gagandeep Singh, Mohammed Alser, Can Firtina, Joël Lindegger, Mohammad Sadrosadati, Nika Mansouri Ghiasi, Can Alkan, and Onur Mutlu. Targetcall: Eliminating the wasted computation in basecalling via pre-basecalling filtering. *bioRxiv*, 2022.

Karen Clark, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and Eric W Sayers. GenBank. *Nucleic Acids Res*, 2015.

Ana Cruz-Silva, Gonçalo Laureano, Marcelo Pereira, Ricardo Dias, José Moreira da Silva, Nuno Oliveira, Catarina Gouveia, Cristina Cruz, Margarida Gama-Carvalho, Fiammetta Alagna, Bernardo Duarte, and Andreia Figueiredo. A new perspective for vineyard terroir identity: Looking for microbial indicator species by long read nanopore sequencing. *Microorganisms*, 2023.

Jeff Daily. Parasail: Simd c library for global, semi-global, and local pairwise sequence alignments. *BMC Bioinformatics*, 2016.

Tim Dunn, Harisankar Sadasivan, Jack Wadden, Kush Goliya, Kuan-Yu Chen, David Blaauw, Reetuparna Das, and Satish Narayanasamy. Squigglefilter: An accelerator for portable virus detection. In *MICRO*, 2021.

Hasindu Gamaarachchi, Hiruna Samarakoon, Sasha P. Jenner, James M. Ferguson, Timothy G. Amos, Jillian M. Hammond, Hassaan Saadat, Martin A. Smith, Sri Parameswaran, and Ira W. Deveson. Fast nanopore sequencing data analysis with slow5. *Nature Biotechnology*, 2022.

Jo Handelsman. Metagenomics: application of genomics to uncultured microorganisms. *Microbiol Mol Biol Rev*, 2004.

James M Heather and Benjamin Chain. The sequence of sequencers: The history of sequencing DNA. *Genomics*, 2015.

Sam Kovaka, Yunfan Fan, Bohan Ni, Winston Timp, and Michael C. Schatz. Targeted nanopore sequencing by real-time mapping of raw electrical signal with uncalled. *Nature Bio.*, 2021.

Nathan LaPierre, Mohammed Alser, Eleazar Eskin, David Koslicki, and Serghei Mangul. Metalign: efficient alignment-based metagenomic profiling via containment min hash. *Genome Biology*, 2020.

Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 2018.

ONT. MinION portable nanopore sequencing device, 2015. https://nanoporetech.com/products/sequence/minion.

ONT. "Read Until" adaptive sampling, 2020. https://nanoporetech.com/resource-centre/read-until-adaptive-sampling.

ONT. Oxford nanopore tech update: new duplex method for q30 nanopore single molecule reads, promethion 2, and more, 2021. https://nanoporetech.com/news/news-oxford-nanopore-tech-update-new-duplex-method-q30-nanopore-single-molecule-reads-0.

ONT. Tombo package, 2023. https://nanoporetech.github.io/tombo/.

ONT. Transforming basecalling in genomic sequencing, 2024a. https://nanoporetech.com/blog/transforming-basecalling-in-genomic-sequencing.

ONT. Nanopore sequencing devices, 2024b. https://nanoporetech.com/products/sequence.

Marc Paga. Basecalling architectures, 2023a. https://github.com/marcpaga/basecalling_architectures.

Marc Paga. Nanopore benchmark, 2023b. https://github.com/marcpaga/nanopore_benchmark.

Marc Pagès-Gallego and Jeroen de Ridder. Comprehensive benchmark and architectural analysis of deep learning models for nanopore sequencing basecalling. *Genome Biology*, 2023.

Kim D. Pruitt, Garth R. Brown, Tatiana A. Tatusova, and Donna R. Maglott. Chapter 18 : The reference sequence (refseq) database. In *The NCBI Handbook*, 2013. URL https://api.semanticscholar.org/CorpusID:16411792.

PyTorch. Multiclass accuracy metric class, 2022. https://pytorch.org/torcheval/stable/generated/torcheval.metrics.MulticlassAccuracy.html.

Christopher Quince, Alan W. Walker, Jared T. Simpson, Nicholas J. Loman, and Nicola Segata. Shotgun metagenomics, from sampling to analysis. *Nature Biotechnology*, 2017.

Anjana Senanayake, Hasindu Gamaarachchi, Damayanthi Herath, and Roshan Ragel. DeepSelectNet: deep neural network based selective sequencing for oxford nanopore sequencing. *BMC Bioinformatics*, 2023.

William Andrew Simon, Irem Boybat, Riselda Kodra, Elena Ferro, Gagandeep Singh, Mohammed Alser, Shubham Jain, Hsinyu Tsai, Geoffrey W. Burr, Onur Mutlu, and Abu Sebastian. Cimba: Accelerating genome sequencing through on-device basecalling via compute-in-memory. *IEEE Transactions on Parallel and Distributed Systems*, 2025.

Gagandeep Singh, Mohammed Alser, Kristof Denolf, Can Firtina, Alireza Khodamoradi, Meryem Banu Cavlak, Henk Corporaal, and Onur Mutlu. Rubicon: a framework for designing efficient deep learning-based genomic basecallers. *Genome Biology*, February 2024.

Matthias Studer, Gilbert Ritschard, Pierre-Alexandre Fonta, Alexis Gabadinho, and Nicolas S. Müller. Distances (dissimilarities) between sequences: seqdist, 2024. http://traminer.unige.ch/doc/seqdist.html.

Ryan Wick. Raw fast5s, 2019. https://bridges.monash.edu/articles/dataset/Raw_fast5s/7676174.

Chris Wright. Bonito basecalling with r9.4.1, 2020. http://www.forestry.ubc.ca/conservation/power/.

A APPENDIX

A.1 EXPERIMENTAL SETUP

A.1.1 DATASET SETUP

As the task is to differentiate between different species, we found that a balanced dataset between the target species greatly improved accuracy. As lengths of each read in an ONT dataset may vary significantly, it is necessary to number the samples in each read and balance the dataset according to sample count. Each is then split into chunks which are passed through the basecaller. We also shuffled the genomes in the training set so that the network trains on classes in a homogeneous manner.

A.1.2 TRAINING/TEST SET AND EXPERIMENTAL SETUP

We utilize the popular Wick dataset for experimental analysis of our basecalling/classification network (Wick, 2019). We utilize the data preparation strategy presented in (Paga, 2023b) to build the training and validation sets. As species classification requires a balanced dataset, and the Wick dataset consists of many species of varying read counts, we first select the datasets with more than 5,000 reads. This results in a total of 30 datasets as listed in the Table 1, where 17 of them are unique species. For each of them, 500 reads are set aside for testing, and the rest are available for training and validation. To maintain consistency during training in species classification, we randomly select one collection of *Klebsiella pneumoniae*, namely *Klebsiella pneumoniae-INF042*, from its 14 options listed in Table 1. The other 13 *Klebsiella pneumoniae* sets are discarded from training, but their testing reads are used.
We use the Wick dataset due to its popularity among researchers and its open-source nature. As this dataset utilizes ONT's R9 chemistry, its basecalling accuracy is not comparable to that of R10 chemistry which boosts accuracy to over 99% (ONT, 2021) but does not currently have widely available comprehensive open source datasets. Importantly, our methodology is not tied to the Wick dataset and can be extended to datasets using the latest flow cell chemistry without loss of generality.

A.1.3 DATA PREPROCESSING

As the Wick dataset does not provide ground truth nucleotide sequences for most of its data, it is necessary to generate these sequences for the training set. For each species there are *fast5* files (Gamaarachchi et al., 2022) containing raw electrical signals (reads), and reference genomes given as *fna* or *fasta* files, and for some of them there are *fastq* files which represent the ground truth of the nucleotide sequence. *fastq* files containing inferred nucleotide sequences are generated for each species using the *dorado dna_r9.4.1_e8_sup@v3.6* basecalling network (ONT, 2024b). The original reads are then annotated with the generated files and "resquiggled" using their corresponding reference files (*fna* or *fasta*). The resquiggle process refers to the correction of basecalling errors by re-assigning the nanopore reads to a reference sequence (ONT, 2023). After these steps, the reads of each species are divided into a ratio of 3:1 training/validation sets, with each read divided into "chunks" of signals of window-size 4,000. As reads contain a widely varying number of signal values, up to 3x difference in amount of total chunks per species, it is necessary to balance the dataset at a chunk granularity. Thus, the number of chunks included for each species is limited to that of the species with the least number of chunks. This species is *Pseudomonas_aeruginosa-MINF_7A*, consisting of 68k chunks, or $\sim$2.03 GB. The complete training and validation datasets are then shuffled so the network learns to classify all species in parallel. Each chunk in the final dataset consists of the original sample, the ground truth according to the resquiggling process, and a classification index between 0 and 16, corresponding to a unique species shown in Table 1.

A.1.4 TRAINING SETUP

The training is performed with an x86 architecture, 16-core CPU and a single NVIDIA Tesla V100 GPU supported by 64GB of RAM. The implementation is performed using PyTorch 2.3, with CUDA 12.1 for GPU acceleration. Python 3.7 is employed, along with other dependencies mentioned in (Paga, 2023a), from which the training framework is retrieved. The model is trained with a window size of 4,000 (Bonito default setting), window overlap of 0, and a batch size of 64 as dictated by GPU VRAM capacity. The initial learning rate is set to 0.01 with a warm-up phase of 1,000 steps (Pagès-Gallego & de Ridder, 2023) and is reduced using the *ReduceLROnPlateau* LR strategy as loss converges. Training is conducted for 17 epochs until convergence, taking around 13 hours. We note that the addition of the classifier layer has negligible ($<$3%) impact on training time in either series or parallel configuration.

Table 1: Datasets and their Read Counts for the experiments

Index	Name of species	Nr. of Reads
1	Acinetobacter_baumannii-AYP_A2	6558
2	Acinetobacter_nosocomialis-MINF_5C	6722
3	Acinetobacter_ursingii_MINF_9C	6976
4	Burkholderia_cenocepacia-MINF_4A	7096
5	Citrobacter_freundii-MSB1_1H	7093
6	Escherichia_coli-MSB2_1A	6985
7	Escherichia_marmotae-MSB1_5C	7064
8	Haemophilus_haemolyticus-M1C132_1	8669
9	Klebsiella_pneumoniae-INF032	14320
10	Klebsiella_pneumoniae-INF042	10695
11	Klebsiella_pneumoniae-INF116	6776
12	Klebsiella_pneumoniae-INF215	7142
13	Klebsiella_pneumoniae-INF322	7212
14	Klebsiella_pneumoniae-KSB1_1I	7031
15	Klebsiella_pneumoniae-KSB1_6G	7040
16	Klebsiella_pneumoniae-KSB1_7E	5832
17	Klebsiella_pneumoniae-KSB1_9A	6787
18	Klebsiella_pneumoniae-KSB2_1B	16847
19	Klebsiella_pneumoniae-NUH11	7336
20	Klebsiella_pneumoniae-NUH27	7321
21	Klebsiella_pneumoniae-NUH29	15178
22	Klebsiella_pneumoniae-SGH07	5645
23	Klebsiella_variicola-INF022	6501
24	Morganella_morganii-MSB1_1E	6307
25	Pseudomonas_aeruginosa-MINF_7A	7082
26	Salmonella_enterica-21_06152	6638
27	Serratia_marcescens-17_147_1671	11742
28	Shigella_sonnei-212_0237	23583
29	Staphylococcus_aureus-CAS38_02	11047
30	Stenotrophomonas_maltophilia-17_G_0092	16010

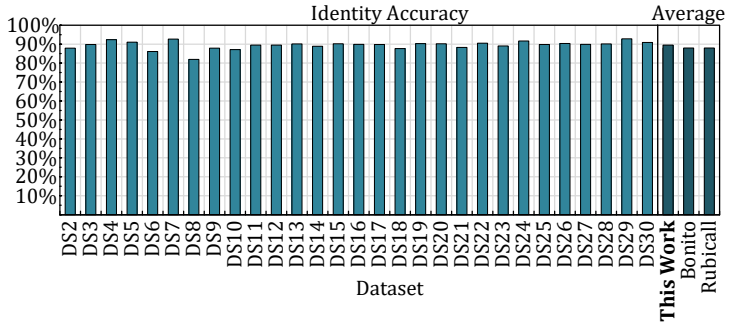

Figure 6: Post-alignment identity accuracy of this work vs. Bonito and SotA classifier RUBICALL (Singh et al., 2024).

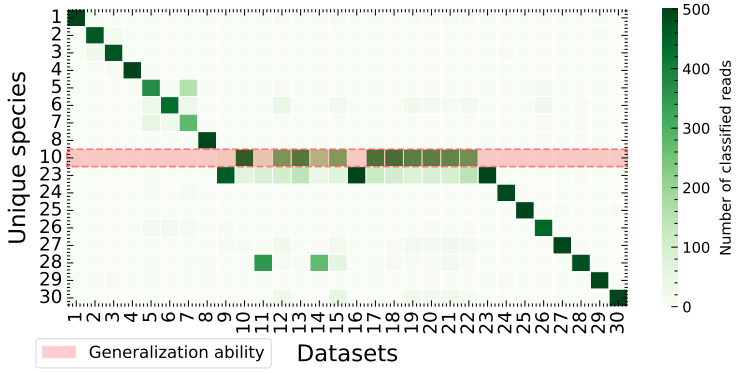

Figure 7: Read classification heatmap for datasets in Table 1. Datasets of same species classify with high accuracy to training class of the same species.

A.2 Further experimental results

A.2.1 Downstream basecalling accuracy analysis

For downstream accuracy analysis setup, the framework proposed in (Singh et al., 2024) is utilized, namely, minimap2 (Li, 2018) is used to align each read to its source genome. Fig. 6 reports the identity accuracies for each dataset, the indices of which correspond to Table 1, with dataset one left off due to failure to align. The average classification accuracy is comparable to both the standard Bonito model and the RUBICALL model, an SotA model demonstrated to outperform many previous advanced models on the same Wick dataset (Singh et al., 2024). Additionally, when comparing common datasets in both works, it was observed that the results were generally consistent, with only minor differences of 1–2%. These results indicate that the addition of the classifier layer does not impact basecalling accuracy.

A.2.2 Species-Level Generalization in Per-Read Classification

The heatmap in Fig. 7 illustrates how the 30 datasets in Table 1 are classified amongst the 17 unique training species. Clear diagonals on the left and right of the figure are noticeable, showcasing the accuracy of our model in differentiating unique species. In the center of the figure, it can be seen that the majority of the datasets belonging to *Klebsiella pneumoniae* converge towards the class of *Klebsiella pneumoniae*. This demonstrates our model's generalizability for classifying datasets unseen in the training set. Poorly classified species, namely, *Escherichiaq marmotae*, incorrectly classified 31% of the time as *Citrobacter freundi*, and *Klebsiella pneumoniae* (datasets 11 and 14), which are misclassified as *Shigella sonnei*, can be attributed to the similarity between these genomes, as for example the Jensen-Shannon divergence between the 9-mer relative counts of *Escherichia marmotae* and *Citrobacter freundi* is less than 0.1 (Pagès-Gallego & de Ridder, 2023). This suggests further research into gene family classification for highly similar genes.

A.3 INTEGRATION IN METAGENOMIC PROFILING PIPELINES

While this work focuses primarily on the accuracy of our proposed method, we provide here some insight into how it may be integrated into the wider metagenomic classification pipeline. This pipeline faces a challenge in that, while basecalling and assembly computational overhead do not scale with number of species, classification computational requirements scale as the genome database grows. Metalign demonstrates a reduction of up to 100x on number of genomes against which to match a given set of samples for a database of 199,807 microbial genome assemblies compiled from the RefSeq (Pruitt et al., 2013) and GenBank (Clark et al., 2015) database, with a comparable reduction in alignment time. Even so, a reduction of 100x still results in $\sim$2000 genomes to which each read must be aligned. While we initially study here a relatively small database of 17 species to understand the feasibility of multi-class read classification, we plan to expand the study by developing a training dataset containing larger numbers of genomes, and classifying to families of genomes.

The method proposed here reduces the number of alignments that must be made for each read to the top-K most likely candidates while maintaining alignment accuracy as, if the network misclassifies a read resulting in no or poor alignment, the read can be re-aligned against the comprehensive genome database. This motivates an interesting research avenue of exploring optimal top-K values to balance the trade-off between the number of network misclassifications against the necessity of aligning against more candidate genomes. This strategy most benefits alignment-based classifiers, who suffer more from the expensive computational alignment step, but also applies to alignment-free classifiers.