A universal DNA signature for the Tree of Life

Bruno A. S. de Medeiros[1,2,3], Liming Cai[4, 5], Peter J. Flynn[4], Yujing Yan[4], Xiaoshan Duan[4,6], Lucas C. Marinho[4, 7], Christiane Anderson[8], and Charles C. Davis[4]

[1]Field Museum of Natural History, Chicago, Illinois, 60605, USA

[2]Department of Organismic and Evolutionary Biology, Museum of Comparative Zoology, Harvard University, Cambridge, Massachusetts, 02138 USA

[3]Smithsonian Tropical Research Institute, Panama City, Panama

[4]Department of Organismic and Evolutionary Biology, Harvard University Herbaria, Harvard University, Cambridge, Massachusetts, 02138 USA

[5]Department of Integrative Biology, The University of Texas at Austin, Austin, Texas, 78712 USA

[6]College of Forestry, Northwest Agriculture & Forestry University, Yangling 712100, Shaanxi, China

[7]Departamento de Biologia, Universidade Federal do Maranhão, Av. dos Portugueses 1966, Bacanga 65080-805, São Luís, Maranhão, Brazil

[8]University of Michigan Herbarium, 3600 Varsity Drive, Ann Arbor, Michigan 48108, USA

Corresponding authors:

Bruno A. S. de Medeiros, Field Museum of Natural History, Chicago, IL, 60605; E-mail: bdemedeiros@fieldmuseum.org

Charles C. Davis, Department of Organismic and Evolutionary Biology, Harvard University Herbaria, Cambridge, MA 02138, USA; E-mail: cdavis@oeb.harvard.edu

Abstract

Species identification using DNA barcodes has revolutionized biodiversity sciences and society at large. However, conventional barcoding methods may lack power and universal applicability across the Tree of Life. Alternative methods based on whole genome sequencing are hard to scale due to large data requirements. Here, we develop a novel DNA-based identification method, varKoding, using exceptionally low-coverage genome skim data to create two-dimensional images representing the genomic signature of a species. Using these representations, we train neural networks for taxonomic identification. Applying a taxonomically verified novel genomic dataset of Malpighiales plant accessions, we optimize training hyperparameters and find the highest performance by combining a transformer architecture with a new modified chaos game representation. Remarkably, >91% precision is achieved despite minimal input data, exceeding alternative methods tested. We illustrate the broad utility of varKoding across several focal clades of eukaryotes and prokaryotes. We also train a model capable of identifying all species in NCBI SRA using less than 10 Mbp sequencing data with 96% precision and 95% recall and robust to sequencing platforms. Enhanced computational efficiency and scalability, minimal data inputs robust to sequencing details, and modularity for further development make varKoding an ideal approach for biodiversity science.

Keywords: biodiversity science, computer vision, DNA barcoding, genomic signature, Malpighiaceae, natural history collections, neural networks, species identification, taxonomy

Introduction

For two decades, conventional DNA barcoding, which relies on standardized short sequences (400–800 bp) for species identification[1–5], has enabled novel and massively scalable science spanning evolution[4,6–9]; ecology[10–14] and paleontology[15–19]. Practical applications of barcoding have also made major contributions to environmental health, including the ability to authenticate medicinal plants[20], detect agricultural pests[21], and monitor poaching and the trade of endangered species[22–27]. Despite these remarkable achievements, conventional DNA barcoding suffers from at least four limitations. First, barcodes are customized specifically for a taxon (e.g., plants, animals, and fungi), and therefore are not universal. For example, commonly used plant barcodes from chloroplast genes such as *mat*K and *rbc*L cannot be applied as barcodes for all plants[28,29], or for animals and fungi. Second, conventional barcode loci may fail to distinguish closely related taxa, a pervasive shortcoming in plants[2,30]. Third, reliance on a single locus may lead to spurious results in the case of complex evolutionary scenarios such as hybridization in deep or shallow time[31–34]. And fourth, the necessary comparison of homologous genes may fail when PCR primers are not universal[35], the source DNA is fragmented[27], or paralogy and the presence of pseudogenes confounds accurate orthology assessments[36,37].

Newer alternatives to conventional barcoding have begun to address these challenges by leveraging high-throughput sequencing and machine-learning powered by deep neural networks. High-throughput sequencing facilitates more comprehensive assessments of total genomic space[38,39]. For example, presence and absence patterns among short DNA sequences (k-mers) from low-coverage reads (i.e., genome skims) can estimate overall sequence distances, bypassing genome alignments entirely as implemented in *Skmer*[40]. Machine learning enables more complex sequence comparisons than conventional methods that rely on homology and simple metrics[41]. Machine-learning models can cluster DNA sequences without supervision[42,43] or classify sequences based on reference datasets[44–49]. In particular, neural networks are exceptionally powerful for sophisticated computer-vision tasks, such as image classification[50]. Thus, the combination of low-coverage genome skimming data and neural networks holds enormous promise for accurate and scalable DNA barcoding, but its potential has yet to be fully realized[39].

Genomes differ substantially in many features beyond the simple nucleotide divergence commonly used in conventional barcoding, but these genomic features have been overlooked in species identification[31,51–55]. We propose that (1) relevant genomic features can be captured by nucleotide composition with short k-mer counts and very small sequence coverage; and (2) these counts can be used to distinguish species and higher taxa efficiently and accurately using machine learning. Prior work on k-mer-based representations of genome composition (i.e., genomic signatures) has shown high accuracy can be achieved with high-coverage data or a large number of replicates per taxon, particularly for identification at higher taxonomic ranks[42–47,56–63]. However, given the millions of existing species and the sparse genetic data available, a practical scalable method would require: (1) consistently high accuracy despite limited evolutionary divergence; (2) fast computations; and (3) high accuracy with small training datasets (both in number of samples and DNA data per sample). Here we developed a novel genomic signature method, which we call **varKoding**, that integrates very low-coverage genome skim data with optimized training of machine-learning models using two-dimensional images representing genome composition (**Figure 1A**). We focus on images as forms of genomic representation since they can be easily stored and accessed across computing platforms, annotated with metadata and readily employed as input data in popular machine learning frameworks such as pytorch[64]. Specifically, our method relies on raw unassembled genomic reads sampling a very small fraction of a genome, since sequence assembly is costly both in terms of DNA sequencing and computation[40,58] and sparse sampling of genomic regions may be sufficient to summarize its features[39]. To develop and optimize varKoding for accurate species identification, we generated a *de novo* genome skim dataset including hundreds of samples derived primarily from historical herbarium specimens for the diverse plant genus *Stigmaphyllon* (Malpighiaceae), which has received extensive phylogenetic and taxonomic treatment[65–69]. Next, we explored the utility of varKoding and compared it to alternatives at different phylogenetic depths from families to species within the flowering plant order Malpighiales (Malpighiaceae, Chrysobalanaceae, and Elatinaceae). Finally, we demonstrate the scalability of varKoding and its potential application in forensics and related fields by testing it on (1) species-level datasets from fungi, plants, animals, and bacteria; (2) massive datasets retrieved from the NCBI sequence read archive (SRA); and (3) a previously published environmental DNA (eDNA) dataset.

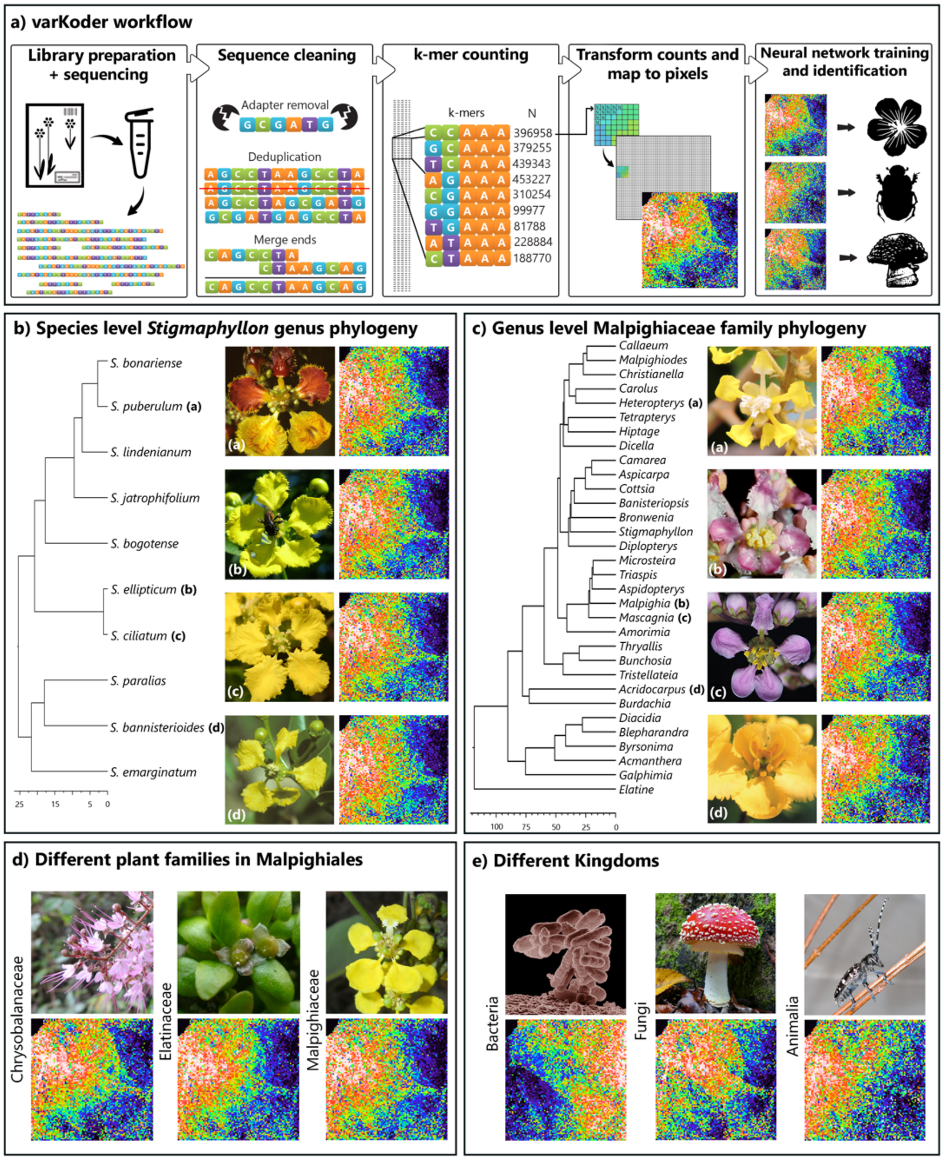


Figure 1. Overview of varKoding. (**A**) Image generation workflow, depicting varKodes. Images are natively grayscale, but here they are mapped to a rainbow color scale for increased contrast. (**B**) Phylogeny and example varKodes of *Stigmaphyllon* species. (**C**)

Phylogeny and example varKodes of Malpighiaceae genera including their closest outgroup (*Elatine*, Elatinaceae). Time trees in 1B and 1C were derived from an ongoing family-wide phylogenomic investigation of the family Malpighiaceae (C. C. Davis personal communication) using methods and fossil constraints described in Cai et al.[66]. (**D**) Examples of varKodes from across plant families of Malpighiales, and (**E**) across kingdoms. Chronograms depicted for each representative set with timelines in millions of years (Myr) at the bottom of **B** and **C**.

Results and Discussion

Genomic signature images can be classified with generalized neural networks

We first generated a novel kind of image representation of a genomic signature based on raw reads, which we termed a **varKode**. varKodes map k-mers onto pixels of a 2-D image based on their similarity and represent ranked k-mer frequencies as pixel brightness. Variation in varKodes can be small but remain visually perceptible among species (**Figure 1B**) and genera (**Figure 1C**). Variation is more striking among higher levels of phylogenetic divergence, such as between families in the order Malpighiales (**Figure 1D**) or different kingdoms of eukaryotes and prokaryotes (**Figure 1E**). We expected, therefore, that neural network architectures developed for image classification, (e.g., deep residual networks, resnets[70] or vision transformers, ViT[71,72]) would be able to differentiate varKodes.

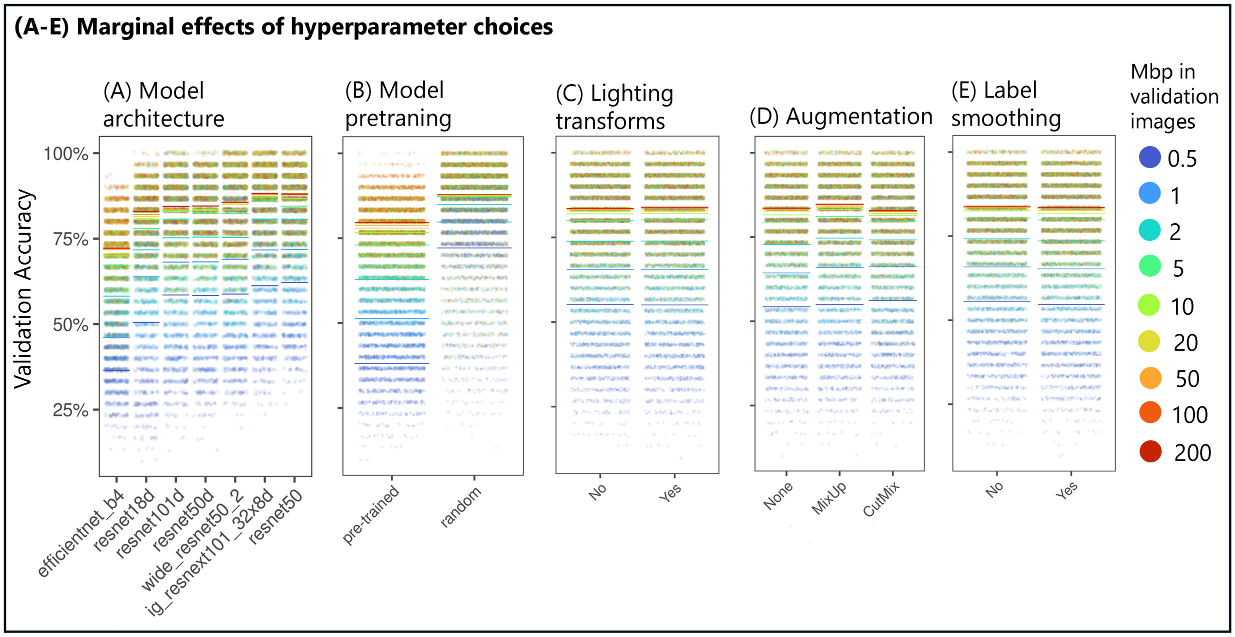


Figure 2. Marginal effects of neural network model and training options. Dots represent individual replicates, and bars depict averages. All parameters were identified to be significant in a linear model: more complex model architectures, lighting transformations, and augmentation methods *MixUp* and *CutMix* improved accuracy. However, pretraining with large image datasets and label smoothing decreased accuracy.

We first optimized hyperparameters and training conditions to maximize accuracy for species-level identification of *Stigmaphyllon*. We identified that varKodes depicting k-mer length = 7 struck a good balance between accuracy and the amount of input sequence data (**Figure 3A**). Furthermore, models trained with augmented data from several subsampled sequences drawn from each individual exhibited substantially better performance (**Figure 3A**). A linear model demonstrated that neural network architectures and training methods designed for image classification of photographs[70,73–76] are extremely useful for varKode-based identification. Specifically, we observed increased accuracy with more parameter-rich neural network architectures (*ResNeXt101*[77], among those tested), augmentation with lighting transformations, *CutMix*[76] and *MixUp*[75]. Label smoothing[78] and pretraining models on generalized photographs decreased accuracy (**Figure 2**). Contrary to the widely held idea that deep neural networks require very large training datasets[60,79], the aforementioned approaches enabled training with very modest data amounts: four biological replicates per taxon was sufficient for 100% median accuracy (**Figure 3B**). Errors in species-level identification were concentrated among sequences derived from herbarium samples that demonstrated evidence of DNA damage, as is sometimes reported for ancient DNA[80] (**Figure** 3**B**). However, including low-quality training samples slightly decreased mean validation accuracy—from 73% to 71%—for low-quality validation samples, but had no effect on high-quality validation samples (89–90% mean accuracy, **Figure 4**).

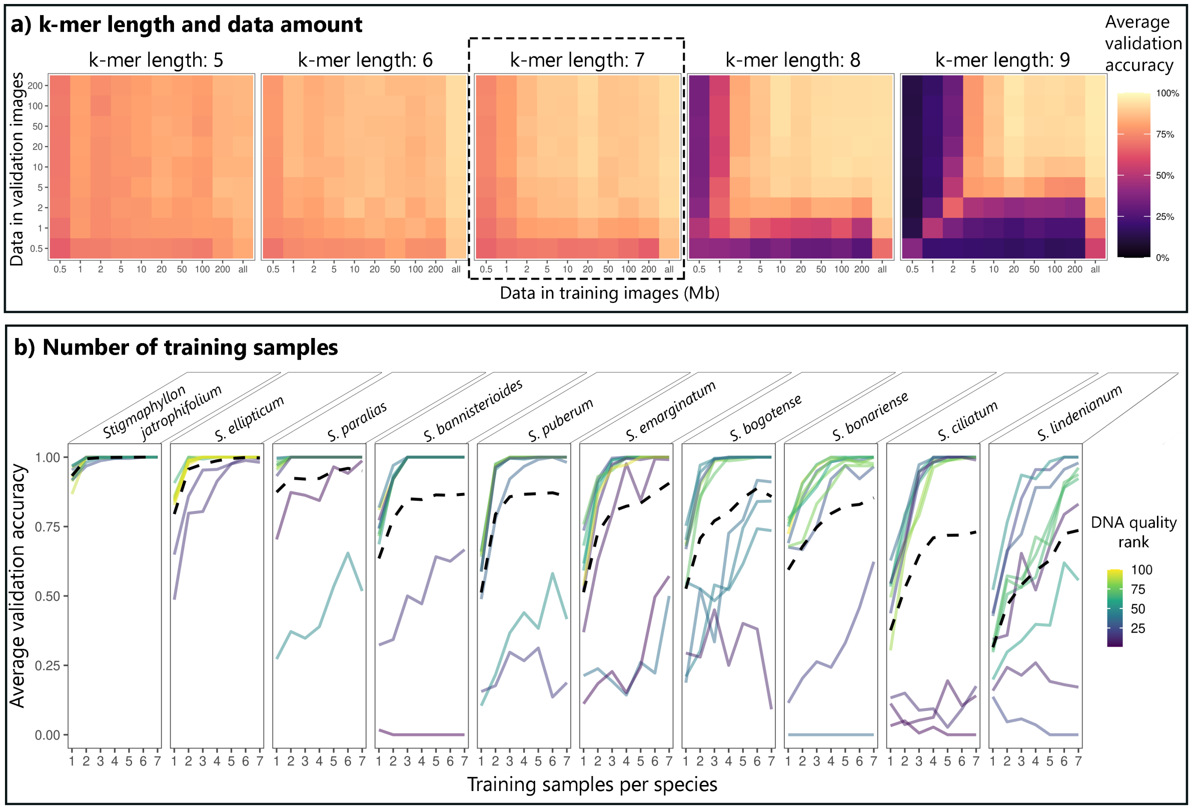


Figure 3. Neural network training of varKodes for species identification. (**A**) Effect of k-mer length and input data amount used to produce varKodes on validation accuracy. Longer k-mers increase accuracy when more data are used. Mixing varKodes subsampled from different amounts of data improves accuracy. Box with dashed line (k-mer length = 7) strikes a good balance between model accuracy and amount of required data. (**B**) Validation accuracy improves with increased number of training samples per species, but even 3–4 samples are sufficient in most cases for achieving high accuracy. Each solid line

represents one sample, colored by DNA quality (i.e., variation in base pair frequencies). Higher rank indicates better quality. Dashed lines represent averages across all samples.

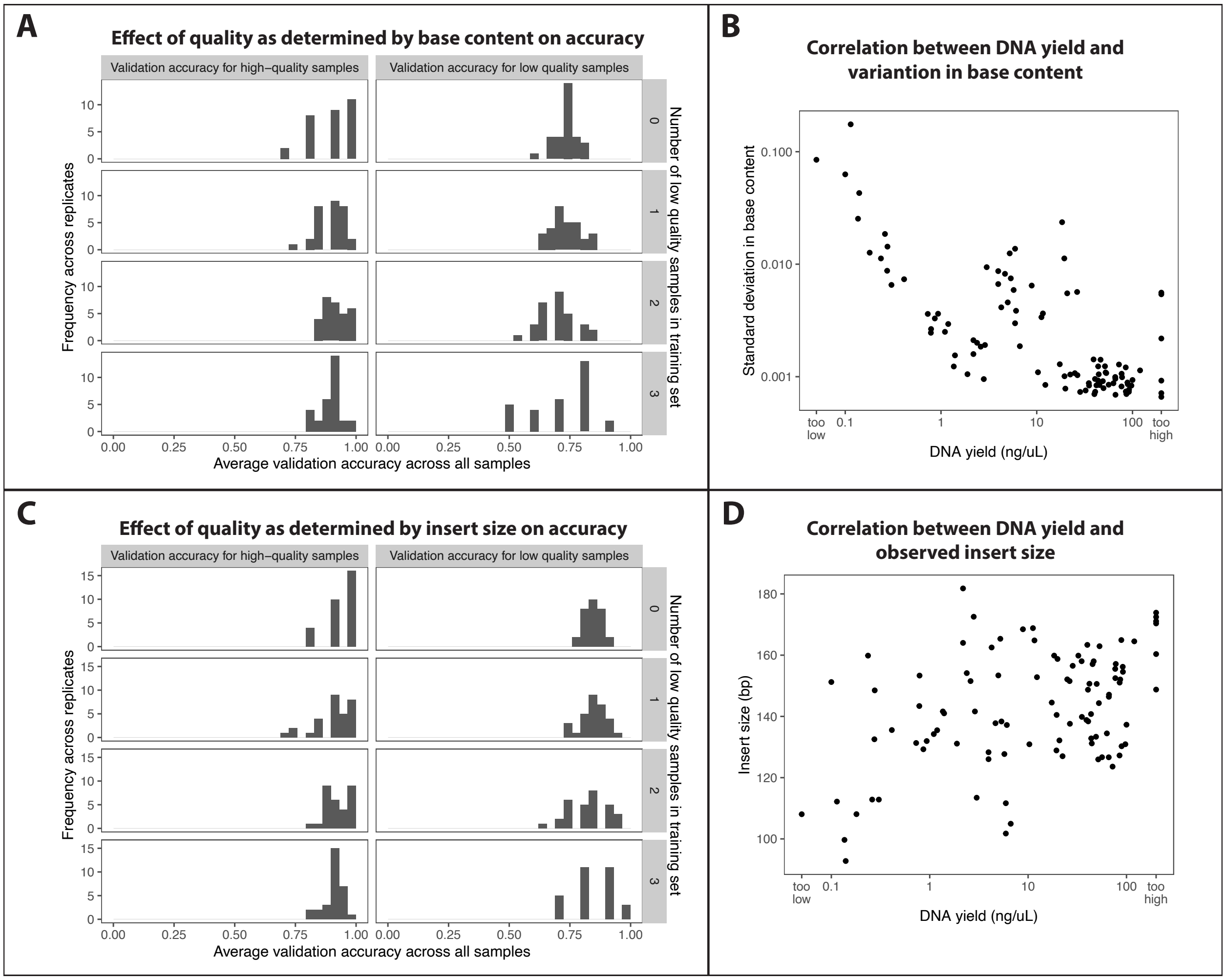


Figure 4. Effect of the inclusion of low-quality training samples, inferred from variation in base pair content (A, B) or insert size (C, D). Increasing the fraction of samples in the training set that were low-quality did not strongly affect the average validation accuracy, but it increased dispersion. Low-quality samples are the four samples with highest variation in base-pair content or shortest insert size in raw reads for each species. Panels **B** and **D** show the correlation of each quality metric with DNA extraction yield.

We hypothesized that lower-quality samples shared similar sequences resulting from common patterns of DNA damage and greater levels of microbial or human contaminants, resulting in spurious similarities in varKodes (**Figure 5**). Contaminants also are thought to increase errors in other genome skim methods[81]. To mitigate this problem, we applied multi-label classification[82] to our neural network models. Although single-label classification models always return a single prediction (that is, an inferred label), multi-label models can return zero or more predictions, avoiding spurious results when there is uncertainty. For a set of samples with known labels used for validation, a prediction is a true positive if the predicted label matches the actual label, and a false positive if not. Failure to predict an actual label is deemed a false negative. For each validation sample, we summarized predictions as (1) correct (true positives only); (2) incorrect (false positives only); (3) ambiguous (multiple predictions, including true and false positives); or (4) inconclusive (i. e. no prediction above the confidence threshold). For each test, we summarized results across all validation samples using two metrics: precision (the sum of all true positives divided by the sum of all true and false positives) and recall (the sum of all true positives divided by the sum of all true positives and negatives).

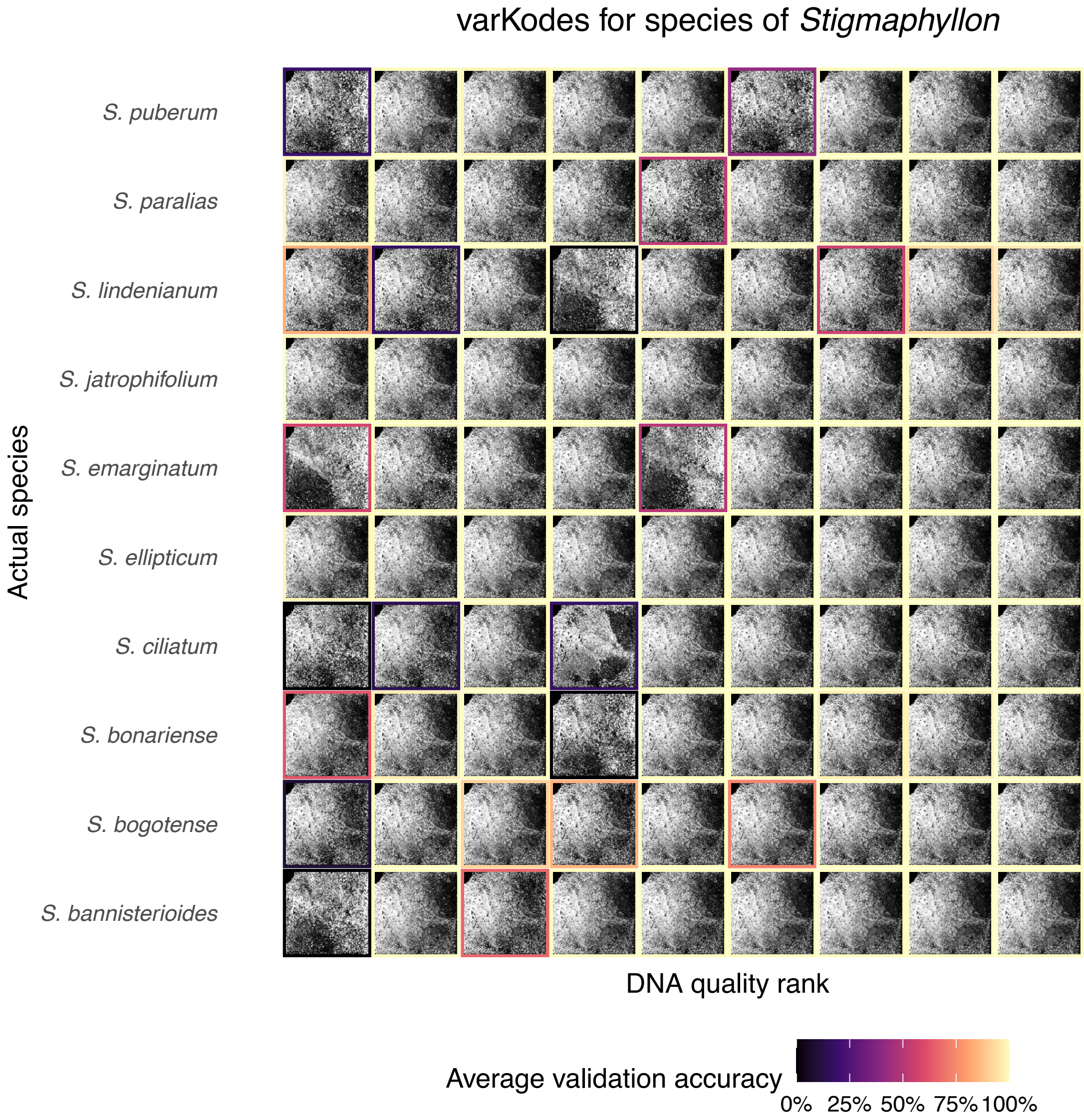


Figure 5. Low-quality DNA may lead to spurious patterns of similarity in varKodes. Samples with lower quality show varKode patterns divergent from their species more often than high-quality ones. These divergent patterns may be similar between low-quality samples across species. These samples also show reduced validation accuracy in a single-label model. For each sample, we show the varKodes produced from all DNA data available. Within each species, samples are organized from lowest (left) to highest (right) DNA quality. Bounding boxes around each sample indicate the average validation accuracy across 30 random replicates with 7 training samples per species.

After optimizing these training conditions, we directly compared varKodes to an existing method of genomic signature representation: the frequency chaos game representation (*fCGR*)[56,59]. In *fCGR*s, k-mers are mapped to pixels based on their oriented sequence and pixel brightness represents the rescaled k-mer frequency. To isolate the effects of pixel mapping and brightness, we created a new representation combining *fCGR* mapping with *varKode* ranked frequency transformation (*rfCGR*). Because raw sequence reads often contain artifactual k-mers at very high frequencies, especially when low-quality DNA is used to construct libraries, we hypothesized that *rfCGRs* would perform better than *fCGR*s, where pixel brightness is linearly scaled to k-mer counts. By directly comparing these 3 kinds of representation combined with four neural network architectures, including (1) two previously employed with *fCGR*s[42,44,60], (2) the optimal architecture in our initial tests (ResNeXt101[77]), and (3) a Vision Transformer (ViT[71,72]), we found that ViT combined with *rfCGR* representation maximizes performance (**Figure 6**). While fCGRs have been initially proposed as tools to study single sequences[56], here we focus all of our comparisons on the task of supervised classification-based genomic composition from very low coverage sequencing. A multilayer perceptron, as employed in previous work[42,60], could not identify any species correctly here (**Figure 6**). Similarly, a previously employed shallow 1D convolutional neural network[44] underperformed more complex architectures (**Figure 6**). *fCGR* showed much higher error rates than either *rfCGR* or *varKodes*, which yielded similar results but with slightly higher accuracy for *rfCGR* (**Figure 6**). These results indicate that deep complex neural networks, while not explicitly developed for genomic signature, are necessary to extract features from very low-coverage data and distinguish closely related species. Moreover, the method of k-mer frequency data transformation seems more consequential than the mapping of k-mers to pixels for the performance of different image representations. Due to its higher performance, we adopt the combination of *ViT* and *rfCGR*s for subsequent tests.

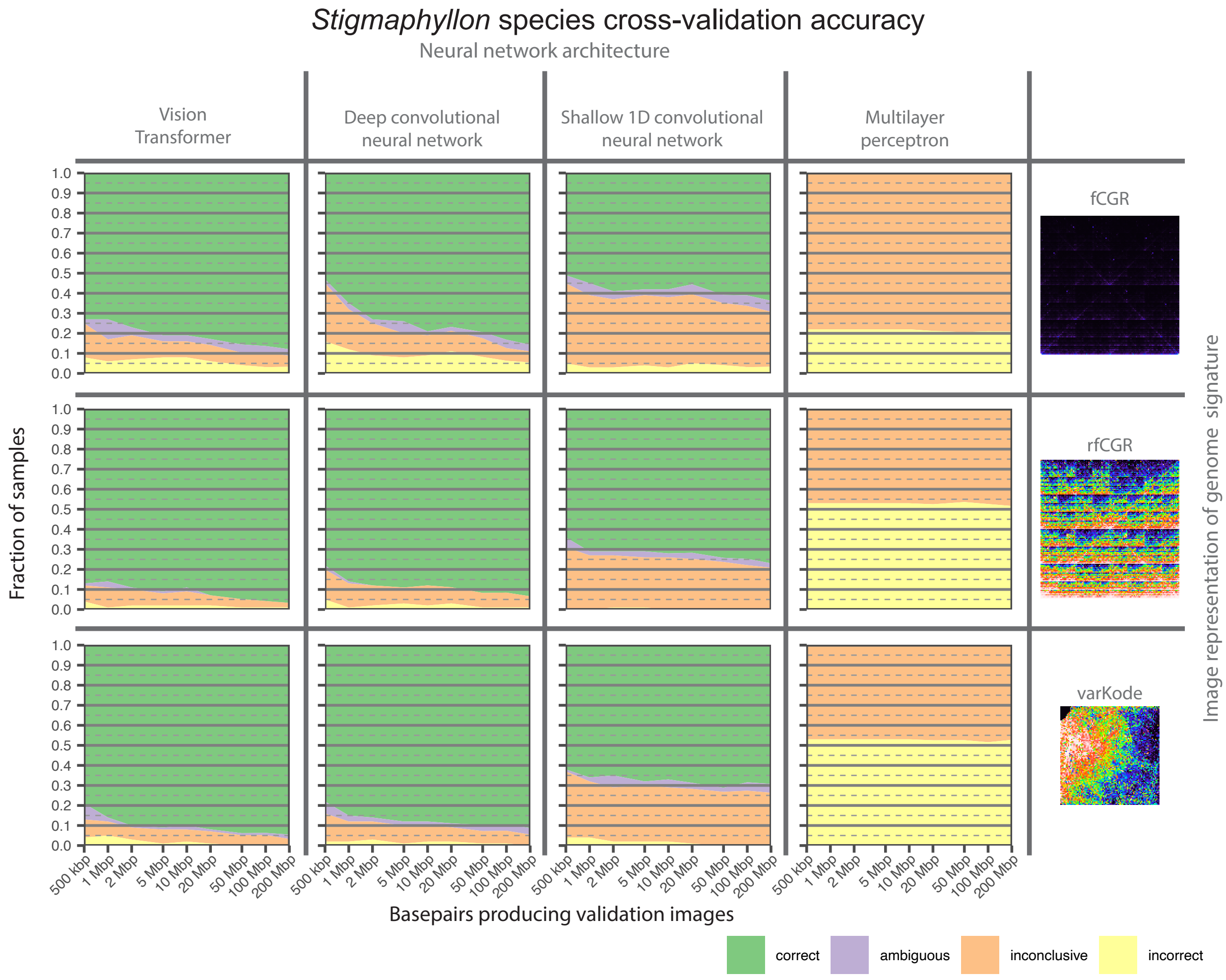


Figure 6. Effect of image representation and neural network architecture on cross-validation accuracy of species identification in *Stigmaphyllon*. One example for each image representation is shown, drawn from the same DNA data (SRA accession XXXX) and mapped to a rainbow color scale for increased contrast. See text for details on architectures.

In summary, we developed and tested a robust and scalable method of DNA barcoding capable of training with small amounts of data, and implemented it in the ***varKoder*** software, which can process sequence data, train an image-classification neural network using varKodes or rfCGRs, query new data with a trained neural network, and convert between the alternative k-mer mappings. These tasks are accomplished with widely used tools for sequence processing[83–87] and for neural network training[64,88–90].

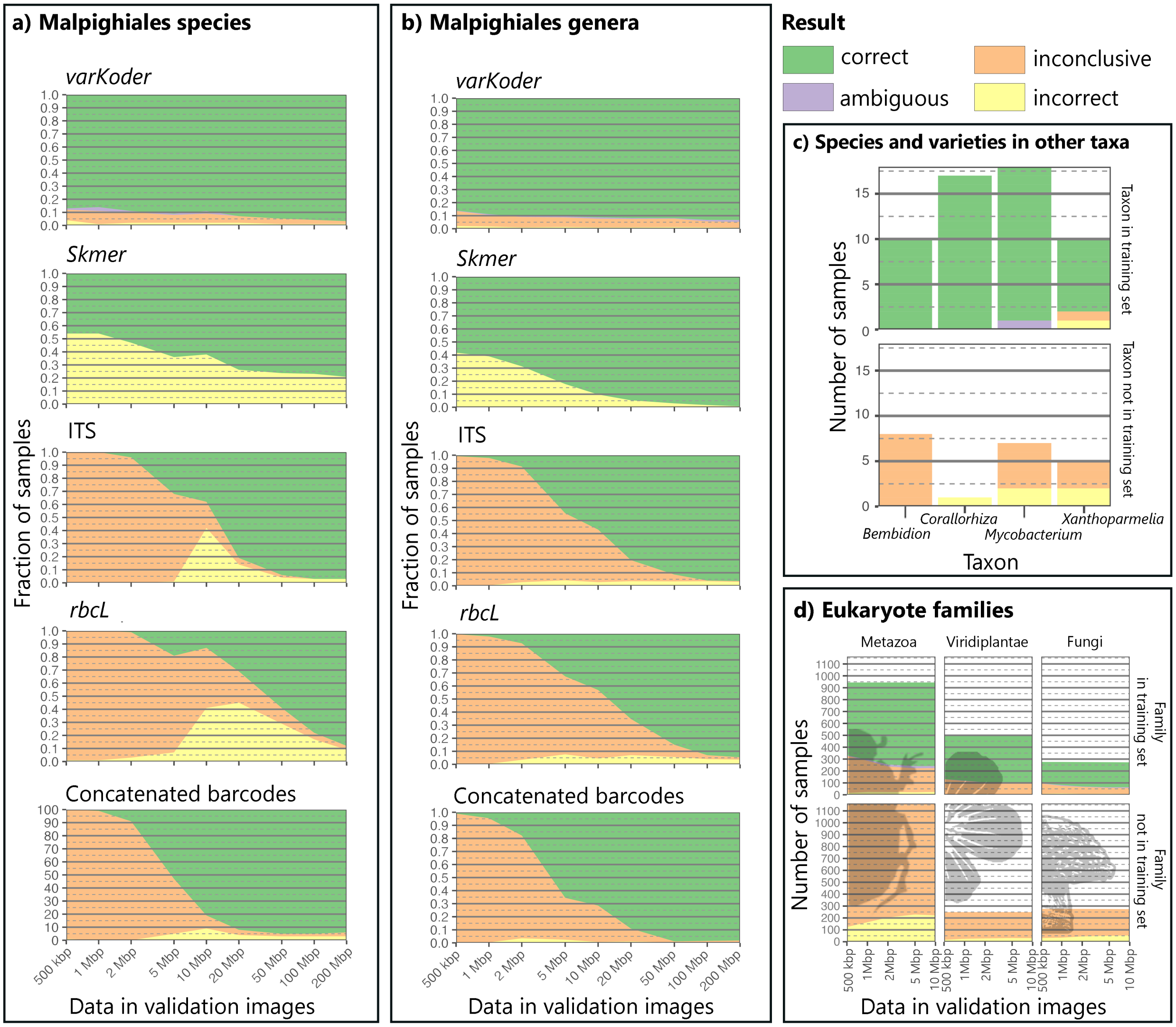


Figure 7. Performance of *varKoder* and alternative barcoding methodologies across different data sets. (**A**) Leave-one-out cross-validation to identify species of Malpighiales using different approaches and amounts of data to assemble query samples. (**B**) Same as (**A**), but for genera. (**C**) Performance for species-level identification across different publicly-available datasets: *Bembidion* beetles, *Corallorhiza* orchids, *Mycobacterium tuberculosis* bacteria, and *Xanthoparmelia* fungi. All query samples used as much data as were available. (**D**) Performance for Eukaryote family-level identification for different amounts of input data.

varKodes are highly accurate for identification of species, genera, and families

To test *varKoder* under a real-world scenario with heterogeneous data (e.g., large numbers of taxa, multiple replicates per taxon, varying sequence depth and sample quality), our *de novo* genomic data set included 287 accessions: 100 samples of *Stigmaphyllon* from our initial development outlined above, plus additional genera in the families Malpighiaceae (31 genera; 151 samples), Chrysobalanaceae (8 genera; 30 samples), and Elatinaceae (1 genus; 6 samples) in the order Malpighiales. We found high cross-validation accuracies for species identity of *Stigmaphyllon* (87.0–96.7% correct, 94.6%–98.9% precision, 88.0%–96.7% recall depending on data input amount; **Figure 7A**). Most errors were inconclusive predictions (2.2–10%), instead of ambiguous (0–3%) or incorrect (1–4%) predictions. *varKoder* is robust to the amount of input sequence data necessary for model training, performing well even at the lower range of input data (**Figure 7A**). Assuming an average genome size of about 2 Gbp for the average species of Malpighiaceae[91], the 500Kbp–200Mbp of data used here represented exceptionally low coverages of about ~0.0002× – 0.107×. Such low coverages imply that we are likely not comparing homologous regions across taxa, but rather more general genomic properties that can be inferred from extremely sparse sampling. Moreover, when compared to cross-validation accuracies of alternative barcoding methods, *varKoder* accuracy is higher than *Skmer*, which showed 46% correct predictions (57.5% precision, 46% recall) with minimal data amounts and peaked at 79.1% for the larger data amounts (80% precision, 79.1% recall, **Figure 7A**). On the other hand, conventional barcodes including individual plastid genes and nuclear ribosomal ITS regions performed well for both BLAST-based (25–97% correct, 66.6–97.3% precision, 25–97% recall depending on the gene) and phylogenetic-based (94–95% correct, >99% precision, 97.2–98.4% recall for concatenated matrices) approaches when at least 50 Mbp of data was provided (**Figure 7A, Figure 8**). However, these results were much worse when <50 Mbp of data were available (down to zero correct for BLAST). In this case, unsuccessful locus assembly leading to inconclusive predictions as the primary reason for the failure (**Figure 7A, Figure 8**), so we expect that alternative methods to BLAST(e.g. [48,92]) would not perform substantially better. Finally, an unsupervised clustering method based on neural networks applied to *fCGR*s (*iDeLUCS*[93]) reached 24–60% clustering accuracy depending on input data amount when prompted to cluster *Stigmaphyllon* sequences into 10 groups. In summary, *varKoder* reaches much higher accuracy for species determination than existing methods for unprecedentedly small amounts of data and demonstrates similar accuracies when greater amounts of sequence data are available.

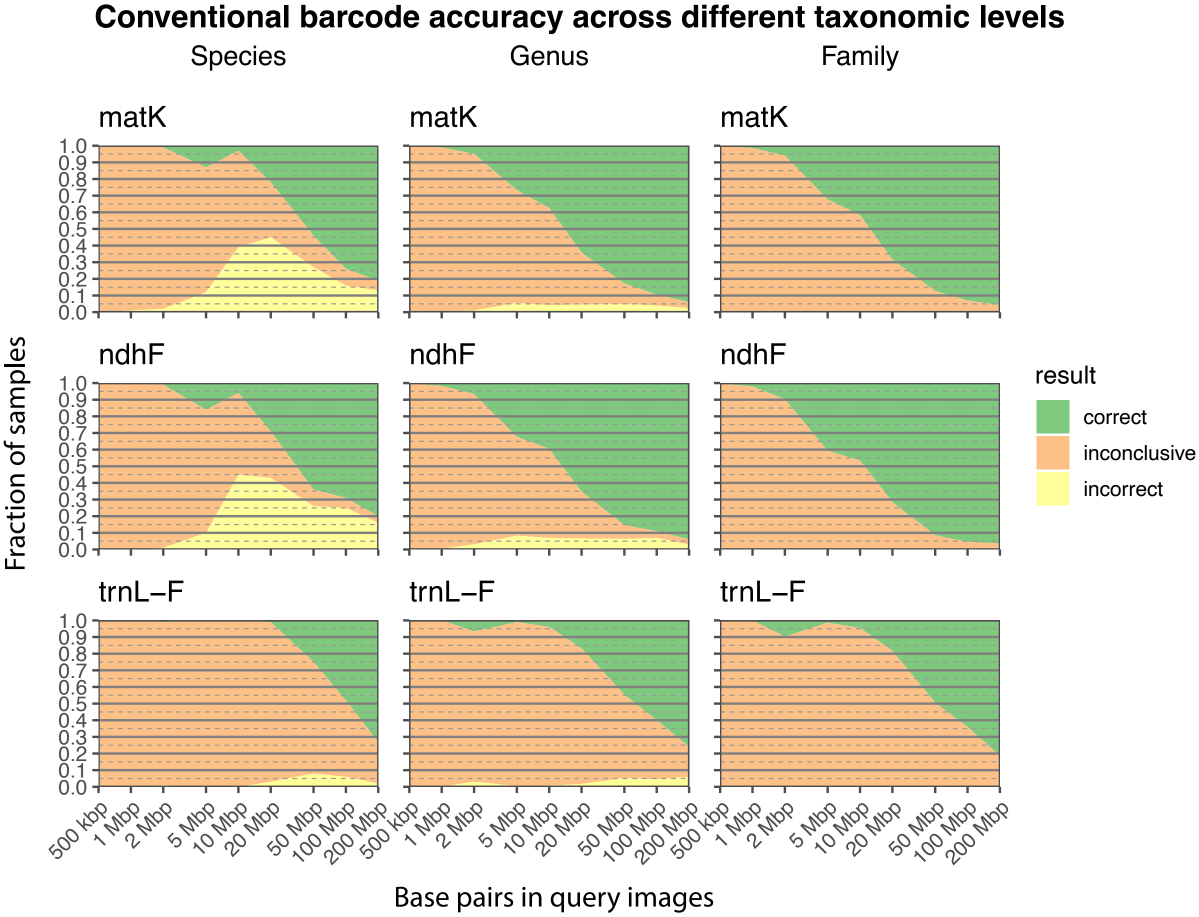
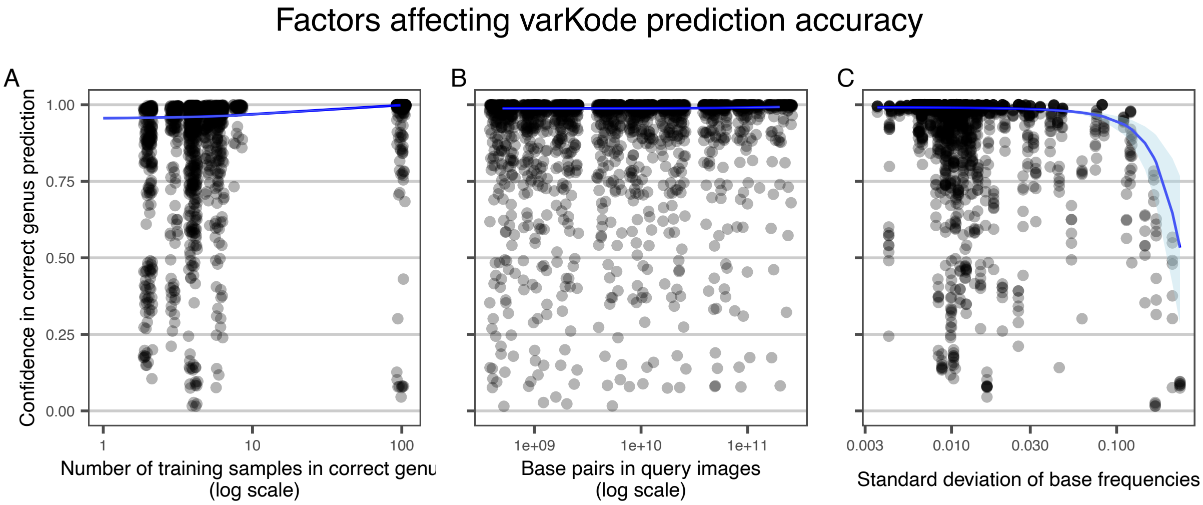


Figure 8. Accuracy of conventional barcode loci for species, genera and families within the Malpighiales.

Genus-level identification yielded similar high accuracies with *varKoder* (86.1–93.3% correct, 97.2%–97.7% precision, 86.4%–94.7% recall depending on input amount, **Figure 7B**), but with a higher rate of inconclusive predictions (4.5–11.5%). A linear model demonstrated that this higher uncertainty can be attributed to two factors: (1) samples exhibiting higher levels of DNA damage in genera other than *Stigmaphyllon*; and (2) genera trained with fewer replicates (e.g., down to 3 samples for some genera; **Figures 9-10**). Despite this trend, the vast majority of genera with fewer replicates and lower DNA quality can still be correctly predicted, resulting in the >97% prediction and >86% recall across the whole dataset. Additionally, samples within genera share fewer genetic similarities than samples within species, which likely poses a more challenging classification problem. However, the incorrect rate was very small in all cases (0.7–2.1%), with most errors being inconclusive or ambiguous predictions. In contrast, *Skmer* exhibited better performance when larger amounts of data were used (99.2% correct, 99.2% precision, 99.2% recall for 200 Mbp), but performed poorly for lower amounts of data like those commonly generated from genome skim experiments (58.2% correct, 58.2% precision, 58.2% recall for 500 Kbp) (**Figure 7B**). Genus-level identifications using conventional barcodes in a concatenated phylogeny were up to 98.1% correct (99.2% precision, 97.2%% recall) when a large amount of data (200 Mbp) was available (**Figure 7B**). But like its application at species-level identification, most predictions were inconclusive when less than 20 Mbp reads were used (**Figure 7B**). Although genome skimming can be used to sequence conventional barcodes, they are more often obtained with amplicon sequencing, which has failure rates ranging from 15–75% even with highly optimized protocols[94], leading to an even higher number of inconclusive predictions. At the family level, *Skmer* and *varKoder* had near-perfect accuracy across all data amounts (>97% correct), while conventional barcodes performed well when there were sufficiently large amounts of data (**Figures 8, 11**).



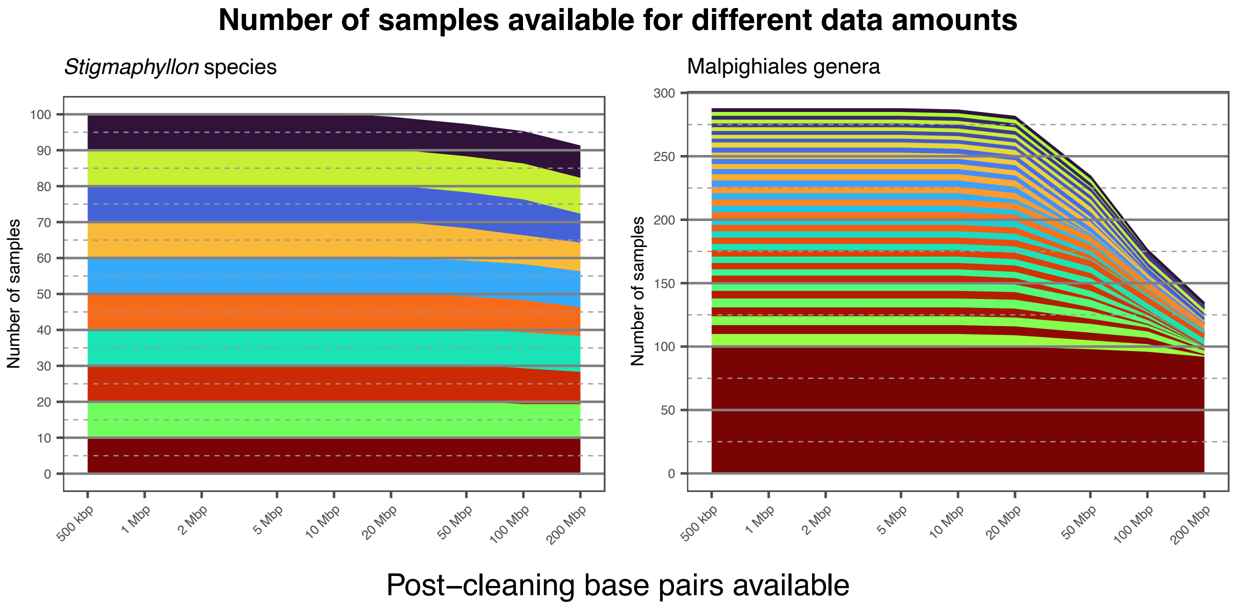
335

336 **Figure 9.** Predictors of confidence in correct genus. A) Confidence increases with more

337 training samples per genus. B) Amount of data per validation image has little effect. C)

338 Validation samples with low quality have lower confidence.

339



340

341 **Figure 10.** Number of samples available for different data amounts in the Malpighiales and

342 Eukaryote families datasets. Arbitrary colors are assigned to individual taxa.

343

344

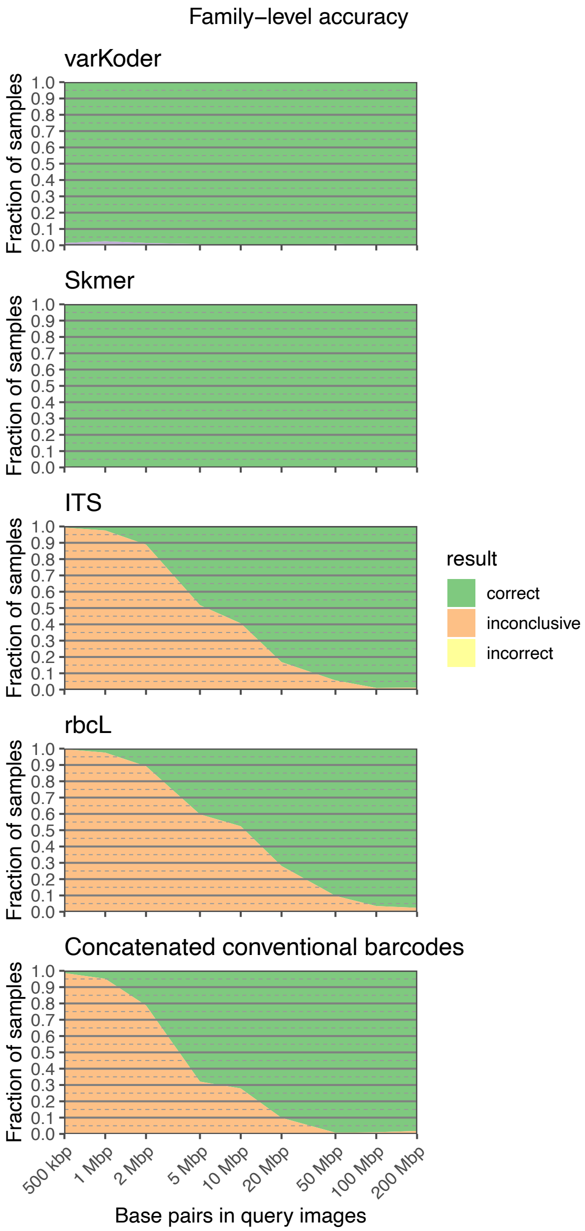


Figure 11. Comparison of *varKoder*, *Skmer*, and conventional barcode accuracy for identifying families of Malpighiales.

varKodes are universal and scalable across the Tree of Life

To further test the universality of varKodes, we expanded to sequencing data from diverse clades of plants, fungi, animals, and bacteria (**Figure 7C**). These tests included species-level identification in insects (*Bembidion* beetles[54,95]) and lichen-forming fungi (*Xanthoparmelia*[96]), species and infra–specific taxon identification in coralroot orchids

(*Corallorhiza*[97]), and clinical isolate identification of strains of human pathogenic bacteria (*Mycobacterium tuberculosis*[98]). In all cases, we tested the performance of *varKoder* on taxa included in the training set and on taxa not included in the training set. We identified perfect species identification (100% correct, 100% precision, 100% recall) for beetles and coralroot orchids included in the training set. For bacteria, 5.6% of the validation set returned ambiguous predictions; the remaining samples were correctly identified (94.7% precision, 100% recall). In lichen-forming fungi, which include DNA from both the fungal and algal partners, and thus are more challenging, 10% of the test samples returned incorrect predictions and another 10% were inclusive; the remainder were correct (89% precision, 80% recall). For all cases, species or varieties not included in the training set generally resulted in inconclusive results, with a minority yielding incorrect predictions **(Figure 7C)**. Precision and recall using varKodes instead of *rfCGRs* were very similar for all four datasets.

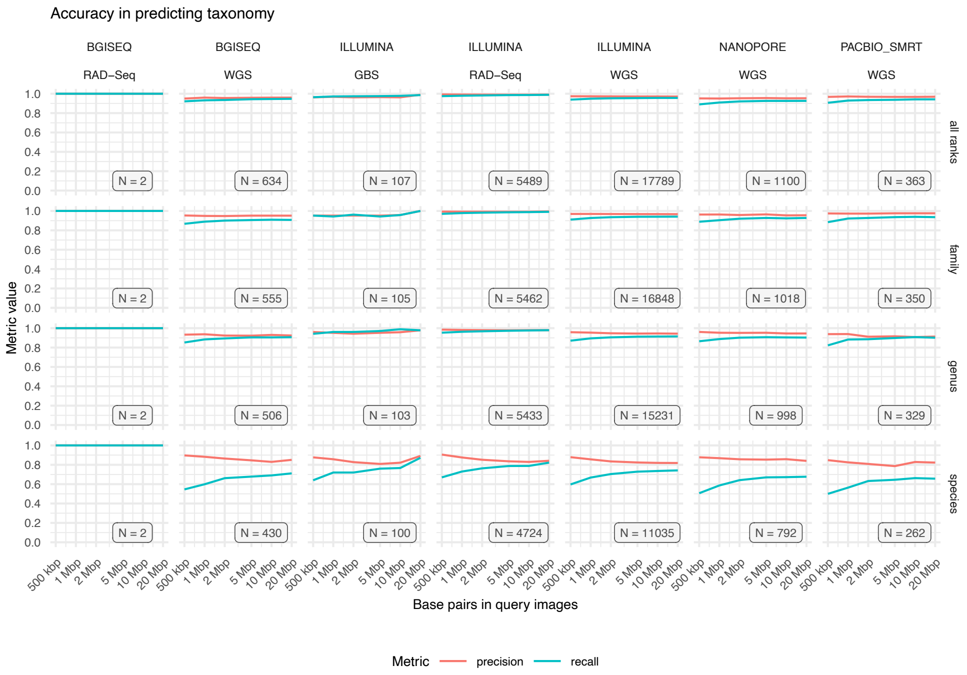


Figure 12. varKoder performance in predicting taxonomy for all data on SRA. Sample sizes refer to the number of validation accessions available for each combination of platform, sequencing strategy and taxonomic rank.

Finally, we tested the scalability of varKodes in three large-scale datasets: (1) all 861 eukaryotic families with Illumina data on NCBI SRA, (2) all taxa with multiple accessions on NCBI SRA, including different sequencing platforms and library strategies (254,819 accessions and 14,151 taxa across all taxonomic ranks), and (3) a previously published dataset of 2916 soil eDNA samples from all seven continents[99]. Owing to NCBI download speed bottlenecks, we restricted varKode construction to a very limited maximum of 10 Mbp of DNA data in the former 2 cases. The family-level eukaryote data achieved a rate of correct predictions of 65.2–81.3% across all kingdoms when families were included in the training set (**Figure** 7**D**), with most errors being inconclusive predictions (17.5–33.1%). Precision varied from 95.3% to 97.3% and recall from 67.9% to 78.3%. Similarly to the species- and variety-level exercise, families not included in the training set often yielded inconclusive predictions **(Figure 7D)**, suggesting a potential for varKoding to be used as a discovery tool when reasonably well-sampled training data sets are available. The expanded data with all taxa from NCBI SRA revealed that varKoding is robust to sequencing platform and library preparation method (**Figure 12**). Predictions at the family level or pooled for all the taxonomic hierarchy are accurate regardless of sequencing details (>94% precision, >86% recall). The much higher accuracy when compared to the dataset based on Eukaryotic families alone may be an effect of a completely random validation set instead of stratified by family, resulting in higher representation of commonly sampled families. At the genus and species level, results are more dependent on the sequencing method (**Figure 12**). For genera, precision/recall using 10Mbp of data varies from 90.8%/90.8% with whole genome shotgun libraries in PacBio to 97.9%/97.6% with genotype-by-sequencing in Illumina. Finally, the eDNA data shows promise in using varKoding to identify the geographical origin of an environmental sample (**Figure 13**): in the validation set, at 10Mbp of DNA data, 94.0% of the samples had continent correctly identified, with 2.6% being incorrect, 1.9% being ambiguous, and 1.5% being inconclusive (84.7% prediction, 84.5% recall). Precision and recall using varKodes instead of *rfCGRs* were very similar for both datasets.

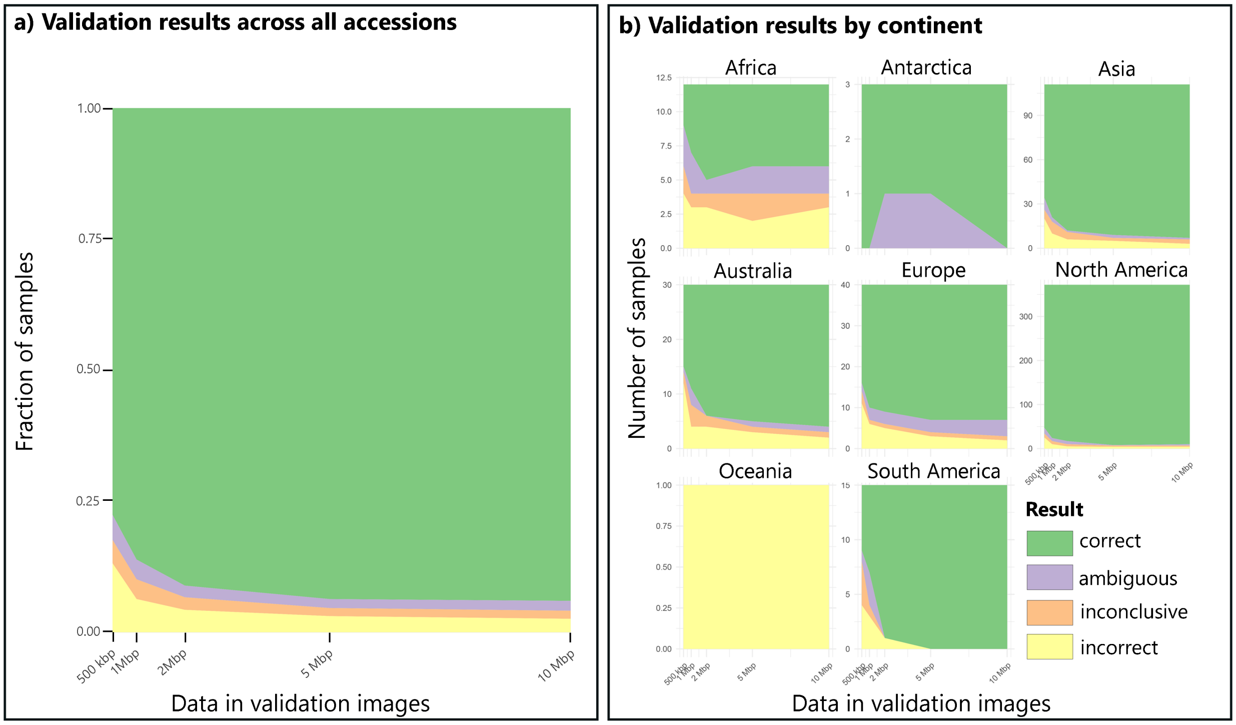


Figure 13. Varkoder performance in identifying the geographical origin of a soil metabarcoding sample. A) Performance across the whole dataset. B) Performance for each continent.

A single model classifying all of life is not possible with conventional barcodes. *Skmer*, the state-of-the-art genome skimming alternative, cannot be scaled to a dataset of this size: our attempt to apply it to Eukaryote families could not be finished after more than 40 days using 32 high-performance computing cores. In general, conventional barcodes, when derived from genome skimming data, require memory- and processor-intensive sequence assembly, and *Skmer* relies on pairwise all-by-all sample comparisons; its computing time and required storage both increase quadratically with the number of samples. Neural network models, on the other hand, have a fixed size, independent of the number of samples used in training, and training time scales linearly with the number of input samples. Our most complex model, trained on all taxa available from the NCBI SRA, has about 1.3GB of disk size. varKode images also are tiny replacements (8.2 KB on average for k-mer length of 7) for much larger genomic data sets (on average, 144 MB per sample here). Downloading up to 20Mb of sequence data for over 250,000 accessions from the

NCBI SRA was the bottleneck, taking over 70 days. By parallelizing processing over 40 cores, processing this data into varKodes was about 10 times faster, resulting in approximately 18GB of data for all of these accessions. Training a model on more than 1.3 million images took about 45 hours using only 2 GPUs. Therefore, a model with the millions of species on Earth could be trained in just a few days in a dedicated server, provided that sequence data to generate varKodes can be transferred at high speeds. Although training on large datasets requires powerful GPUs and large memory, training on small datasets and querying is possible on personal computers in a few seconds to minutes. To reduce the computational resources required for training new datasets, we provide a pre-trained model from both varKodes and rfCGRs from all taxa on SRA using the huggingface hub (https://huggingface.co/brunoasm/vit_large_patch32_224.NCBI_SRA). See Asprino et al.[100] for details on the data used for this model. Whenever the data become available, a model potentially trained on millions of species can easily be ported to devices without continuous internet access. Moreover, the minimal data amounts needed for identification could be generated in seconds in a portable Nanopore device. Finally, the library preparation method based on shotgun sequencing is very simple and can be automated with portable consumer devices, such as the Nanopore Voltrax. Together, these properties allow for more widely distributed applications of varKoding, such as field-laboratory environments[101] or proposed distributed genetic databases[102].

Conclusions

varKoding is universal, accurate, efficient, and holds tremendous promise for documenting and discovering Earth's biodiversity. It achieves accurate identification with minimal data compared to existing next-generation sequencing methods, while maintaining universal applicability across the Tree of Life. Its modular framework relying on widely used image formats and machine learning frameworks can evolve alongside advances in sequencing technologies, bioinformatics, and machine learning, as exemplified here by the update in image representation (*varKodes* to *rfCGR*s) and neural network architecture (resnext to ViT) after initial testing. For these reasons, we expect it will contribute for the wider adoption of genomic signatures on biodiversity assessments and ecological research, overcoming current challenges[39]. Reference data for varKoding will be increasingly available from ambitious efforts in genome sequencing[103–107]. However, we note that reference data for varKoding is much easier and cost-effective to obtain from low-coverage genome skims than high-quality contiguous genomes: the robustness to minimal levels of coverage a central advantage of our method. For example, our cost for a 3× skim of herbarium samples is about $34 per sample, versus a high-quality genome which may cost tens-of-thousands of dollars each. Thus, varKoding shows tremendous promise for further automating species identification from natural history collections[108–110].

We expect that varKoding will be invaluable to the biodiversity science community in numerous ways, with many avenues remaining to be explored. One of them is the identification of samples with poor-quality and degraded DNA, such as unidentified fragmentary fossil and subfossil remains in natural history collections[108,111]. For example, Malpighiales samples with signs of DNA damage could be correctly identified using *varKoder* to species or genus in many cases and to family in almost every case. Future research could explore the lower limits of sample quality and sequence coverage to achieve accurate identification at different divergence levels. Moreover, a promising avenue of research is to identify the genomic features driving the success of identification based on such low sequence coverage. It is possible that the changes in repeat patterns are more important drivers of genomic evolution than currently appreciated[31,51–55]. Finally, we expect that new neural network architectures and forms of DNA representation will continue to be explored. One limitation of varKoding, as applied here, is the challenging identification of samples within mixed components such as lichens or environmental DNA. However, with long-read sequencing, *varKodes* and *rfCGR*s from single reads could potentially include sufficient data for that end. Moreover, mixed samples could be useful for other ends: varKodes could be used to classify a set of sequences based on any kind of metadata, beyond taxonomy as demonstrated by our test on the geographical origin of a soil sample.

Author contributions

BASM conceived varKodes and wrote the program *varKoder*. BASM and CCD designed the research. CCD, CA and XD designed sampling and lab methodology for the new sequence data. CCD, XD, YY, LCM, and CA collected the new sequence data. BASM and PJF collated datasets from published data. BASM, CCD, LC, YY and PJF analyzed and interpreted the data. BASM, CCD, LCM and PF prepared the figures. BASM and CCD wrote the manuscript with key contributions from LC, YY, CA and PJF. All authors approved the manuscript.

Acknowledgments

BdM was supported by the Harvard University Museum of Comparative Zoology, the Smithsonian Tropical Research Institute and the Walder Foundation. LC was supported by Harvard University and by a Stengl Wyer scholarship from the University of Texas at Austin. PF was supported by LVMH Research, and Dior Science. YY was supported by a postdoctoral fellowship from Harvard University Herbaria. CCD was supported by Harvard University, LVMH Research, Dior Science, and National Science Foundation grants DEB-1355064 and DEB-0544039. Computations were performed at the Harvard Cannon Cluster and the Field Museum Grainger Bioinformatics Center. We thank the Bauer Core Facility, and especially Claire Reardon, at Harvard University for providing technical support during the laboratory process. We thank Renata Asprino and Kylee Peterson for their assistance in obtaining the newly sequenced data under Harvard's Binding Participation Agreement. The team at Sound Solutions for Sustainable Science carefully edited early versions of our manuscript.

Online Methods

Sequence data

Taxon sampling, DNA sequencing, assembly, and annotation for newly acquired genetic data—The newly generated plant data used here and the methods to obtain these data are described in detail in a data descriptor article[100]. Briefly, they included members of the large and diverse order Malpighiales[34]: Malpighiaceae (251 accessions representing 31 genera), Elatinaceae (6 accession for 1 genus), and Chrysobalanaceae (30 accessions for 8 genera). Malpighiaceae includes *Stigmaphyllon* with the most comprehensive species sampling: 10 species and 10 accessions sampled per species. All 100 *Stigmaphyllon* samples were sequenced specifically to build, validate, and test our identification models at shallower phylogenetic depths, since their taxonomy has been extensively revised by coauthor C. Anderson[68,69]. Each of these samples was labeled with species, genus, and family names. The focus for the remainder of the Malpighiaceae, Chrysobalanaceae, and Elatinaceae sampling was to identify a given sample to genus. In this case, among the non-*Stigmaphyllon* samples we included 3–9 species per genus. Each accession in this case was labeled with its corresponding genus and family identification. Unlike *Stigmaphyllon,* where we included multiple accessions per species, there were no additional replicates per species for our genus-level sampling. For this dataset, we used leave-one-out cross validation in all assessments, and therefore there are no train and validation sets. For additional information see Asprino et al.[100] .

Public genomic data compilation—To further understand the versatility of varKodes more broadly across the Tree of Life, we tested species identification using genome skim data sets from four genera of plants, animals, fungi, and a bacterial species. For each of the four organismal clades, we trained a multi-label model that included five species with at least three samples per species. This involved a plant data set from coralroot orchids (genus *Corallorhiza*)[97], with five species (or varieties) with at least five samples per species, except for *C. striata* var*. vreelandii* and *C. striata* var*. striata,* for which we included six and seven samples each, respectively. The animal data consisted of a beetle data set in the genus *Bembidion*[54,95], which included five species with five samples per species. The fungal dataset focused on a lichen-forming fungus in the genus *Xanthoparmelia*[96]. Since the *Xanthoparmelia* species were paraphyletic, we subsampled only monophyletic groups for model training. In this case, four species included three samples per species (*X. camtschadalis*, *X. mexicana*, *X. neocumberlandia*, and *X. coloradoensis*) and one species included five samples per species (*X. chlorochroa*). One potential confounding factor for the

Xanthoparmelia model is that *Xanthoparmelia* is a lichen-forming fungus and thus genome skim data represents a chimera of fungal and algal genomes representing both partners in this unique symbiosis. Species of the algal symbiont *Trebouxia* are flexible generalists across fungal species *Xanthoparmelia*. Since these genome skims are a mix of both algal photobiont and fungus, we hypothesize that the accuracy of our model decreased because of the more generalist nature of *Trebouxia*[112]. Finally, the bacterial data set included clinical isolates from *Mycobacterium tuberculosis*, the species of pathogenic bacteria that causes tuberculosis[98]. We included representatives of five monophyletic *M. tuberculosis* lineages (L1, L2, L3, L4.1.i1.2.1, and L4.3.i2) with seven clinical isolates per lineage. In all these cases, we labeled samples with the lowest-level taxonomic identification available (species, subspecies or isolates). For taxa with two or more samples available, 20% (with a minimum of 1) were randomly selected for the validation set, which also included all taxa represented by a single sample (therefore, absent from the training set). The remaining accessions were used in the training set. See Asprino et al.[100] for further information.

We also compiled two broad datasets from the NCBI SRA. The first consists of all 861 eukaryotic families with sequenced under the Illumina platform from whole genome shotgun (WGS) libraries and up to 10 Mbp of data (download date March 7, 2023). This comprised 8,222 accessions, including families of animals (5,642 accessions, 1,426 families), plants (2,705 accessions, 401 families) and fungi (1,572 accessions, 363 families). We labeled samples with family name only and included taxa with at least two associated accessions in the training set. Our validation set consisted of 20% randomly selected accessions from each family (with a minimum of one), plus all accessions in families with a single accession available (therefore not part of the training set). Only eight of the 8,222 samples included yielded less than 10Mbp after sequence cleanup for varKode preparation, and all at least 100 Kbp. The second broad-scale dataset includes all taxa on NCBI SRA that could be represented by at least 3 independent accessions. In this case, we included data amounts of up to 20 Mbp, different sequencing platforms (Illumina, PacBio, Nanopore, BGIseq) and library preparation methods (whole genome shotgun, RADseq, GBS) downloaded on January 9, 2024. For taxa with too many sequences available (such as humans, crops, disease agents, etc.), we randomly chose up to 20 accessions for each combination of sequencing platform and library preparation method. The resulting dataset includes 253,820 accessions associated 28,636 taxonomic labels. In the training set, 97.52% of the accessions included 10Mbp of cleaned data, with the remainder having at least 500Kbp. Accessions were labeled with all NCBI taxonomy ranks available (from infra-specific taxa to domain), the library preparation method, and the sequencing platform. The validation set, in this case, consisted of a random selection of 10% of all samples, not stratified by taxon. For additional information, see Asprino et al.[100] .

Our final dataset was assembled with the aim to extend varKoder beyond taxonomic identification. We compiled a global soil metagenome eDNA dataset labeled with continent of origin from Ma et al.[99] We filtered out any metagenomic sample which lacked information on continent in the Ma et al. This yielded 2916 soil metagenome samples across all seven continents. We downloaded 10Mbp DNA data for each sample directly from NCBI. All code used to download and analyze these data can be found in the GitHub repository for our study (https://www.github.com/brunoasm/varkoder_development).

varKode design and testing

Sequence data preprocessing—Prior to the construction of images, raw reads were lightly cleaned using the following steps: identical reads were de-duplicated using *clumpify.sh* as implemented in *BBtools*[84,113], adapters were removed, low-quality tails trimmed, and overlapping read pairs merged using *fastp*[86] with options "--detect_adapter_for_pe", "--dedup", "--dup_calc_accuracy 1", "--disable_quality_filtering", "--disable_length_filtering", "--trim_poly_g", “--merge", "--include_unmerged", . Next, we randomly selected subsets of cleaned reads with predefined data amounts, ranging from 500 kbp to 200 Mbp, with *BBtools*. These data subsets were used to generate a variety of input varKodes for a single sample and all such images were used for training (see main text and Figure 2A). Finally, we applied *dsk*[85] to count k-mers of a given length based on clean raw reads (i. e. k-mers are counted for each read and their frequencies are pooled across reads). *dsk* exhibits good performance with low memory requirements, which is ideal for potential applications using varKodes on low-memory devices. We note that analyses for species-level public datasets have low computational requirements and were performed on an Apple MacBook with ARM processor architecture.

varKode and rfCGR construction— We designed novel images—**varKodes**—that portray relative frequencies of k-mers from low-coverage raw Illumina reads. These are similar to a frequency chaos game representation (*fCGR*) *sensu* Jeffrey[53], but optimized for raw reads in which sequence orientation is unknown, and therefore canonical k-mers and their reverse complement are indistinguishable. This averaging of canonical k-mer frequencies and their reverse complements is widely used in the context of raw reads[40,61,62,114,115]. We call these images varKodes because they en*CODE* the *VAR*iation in k-mer frequencies in a sample. We name our method **varKoding** after varKodes, but notice that it is modular and can use other kinds of DNA image representation. They are meant to represent a genomic signature by mapping k-mer identity to pixel position in an image, such that k-mers with more similar composition are closer together. Additionally, the brightness of these pixels represents the abundance of the associated k-mer, but we use ranks instead of raw frequencies to decrease the effect of overabundant and artifactual k-mers. In summary, varKodes are produced by mapping k-mer counts onto a pre-computed map of k-mers to pixels, and transforming frequency data to pixel brightness. varKode design employed t-SNE[116] and the python libraries *numpy*[88] and *pillow*[117]. In addition to varKodes, here we also developed a new image representation that uses the same pixel mapping as *fCGRs* but represents k-mer abundance as ranks instead of raw frequencies. We named these ranked frequency chaos game representation (*rfCGR*). Both varKodes and *fCGRs* are saved as 8-bit PNG images including labels as exif metadata.

Testing k-mer length and data amount—We employed *fastai*[89] for, a high-level implementation of neural networks based on *pytorch*[64] for training and prediction. All the model architectures we applied are image classification models available from the *timm* library[90], which have been widely tested using a variety of image types. To identify the optimal training hyperparameters for our neural network, we conducted a series of tests using the species-level data set for the genus *Stigmaphyllon*. We generated varKodes for each of the *Stigmaphyllon* samples. We first tested the joint effect of k-mer length and input data amount for neural network classification accuracy by selecting three samples per species as a validation set; the remaining samples were used to train neural networks using different amounts of input data across 10 randomly generated training sets. As input data for both the validation and training sets, we randomly subsampled the original sequences into fastq files containing from 500 Kb to 200 Mb (equivalent to about 1,700 to 670,000 2x150bp Illumina reads). In this test, we only included samples that yielded at least 200 million base pairs after cleaning. We also tested the effect of including images for all data amounts during training. For each replicate, we applied the widely used image classification neural network *resnet50* architecture[118] to classify varKodes and trained models for 30 epochs. We visualized the distribution of validation accuracy for each combination of input data amount and k-mer lengths to find a good balance between both. Visualizations and code applied for training and evaluation is available in our GitHub repository (https://www.github.com/brunoasm/varkoder_development).

Neural network optimization—After identifying an appropriate k-mer length and input data used to produce varKodes (**Figure** 3), we next tested a series of neural network training conditions. We varied the neural network model complexity, choosing from seven commonly used architectures: *resnet50*[118], *resnet-D*[70] with different depths (18, 50, 101), a wide *resnet50*[70], *efficientnet-B4*[119], and ResNeXt101[77]. We also tested the effect of the following: random initial weights vs. pretrained weights from the *timm* library[90], presence or absence of lighting transforms, presence or absence of label smoothing, and presence or absence of augmentation strategies (i.e., *CutMix*[76] or *MixUp*[75]). Because these parameters may have complex interactions, we tested all combinations of architecture, pretraining, transforms, label smoothing, and augmentation, with 20 replicates for each combination of conditions. In each replicate, we randomly chose 20% of the samples for each species of *Stigmaphyllon* as validation and trained the model using the remainder for 30 epochs. Training was performed using all varKodes available for each sample (from 500kbp to 200Mbp). For validation, we separately evaluated whether each varKode with a different amount of data was correctly identified. For each replicate and amount of data used to validate varKodes, we recorded the average validation accuracy across the validation set.

We then applied a linear model to predict the effect of all training parameters and amount of data in varKodes in the validation set on validation accuracy. Validation accuracy in this case was arc-sin transformed for linear modling due to its bounded range of 0–1. We started from the full model containing all parameters and their interactions and reduced the model step-wise based on AIC scores (i. e. Akaike Information Criteria), as implemented in the R function step. Visualizations and code applied for training and evaluation is available in our GitHub repository (https://www.github.com/brunoasm/varkoder_development).

Testing sample number requirements—A legitimate concern with complex neural networks is that they may require vast amounts of training data and that typical skimming data sets might be insufficient for them to be useful. We tested the robustness of our models to the effect of the number of samples per species included in training by using from one to seven samples per species as training set and the remaining as validation, with 50 replicates per number of training samples. The batch size used in training was adjusted for the cases with very few samples included, so that each training epoch included about 10 batches. We included varKodes from 1Mbp to 200Mbp in both training and validation sets. In this case, we applied the training parameters informed by our previous analyses: a *resnext101* architecture, random initial weights, *CutMix* augmentation, and label smoothing for 30 epochs. We visualized the effect of the number of samples by plotting the average validation accuracy of each sample against the number of training samples used in each case. Visualizations and code applied for training and evaluation is available in our GitHub repository (https://www.github.com/brunoasm/varkoder_development).

Testing the effect of data quality—Most of the cases with low accuracy corresponded to samples with low DNA yield (**Figure 3B**). We identified that DNA extraction yield was significantly correlated with two metrics of DNA quality: average insert size and variation in nucleotide composition along reads[80] (**Figure 4**). *varKodes* produced from these samples may be visually distinct from other samples of the same species (**Figure 5**). For this reason, we further tested whether sample quality in training or validation impacted accuracy. Using both quality metrics, we identified the five lowest quality samples for each species. We next produced training sets using six randomly chosen samples per species, varying the number of low-quality samples included in training from zero to four. We included varKodes from 1Mbp to 200Mbp in both training and validation sets. We repeated this for 30 replicates for each number of low-quality samples. Like our tests with varying sample numbers, we applied the following training parameters: a *resnext101* architecture, random initial weights, *CutMix* augmentation, label smoothing for 30 epochs. For the validation set, we separately recorded the accuracy for high- and low-quality samples. We then visualized the effect of inclusion of low-quality samples in the training set by observing the distribution of validation accuracies for high-quality and low-quality samples across the range of number of low-quality samples included in the training set. Visualizations and code applied for training and evaluation is available in our GitHub repository (https://www.github.com/brunoasm/varkoder_development).

Implementation of varKoder—Following all the tests described above, we implemented the optimal neural network training strategies in a python program named ***varKoder***. *varKoder* can process, train and query varKodes and is freely available on our GitHub: https://github.com/brunoasm/varKoder. Because it employs standard neural network frameworks (namely, *pytorch*[64], *fastai*[89], and *timm*[90]), any of the image classification models and training hyperparamenters available now or in the future via these libraries can be easily adapted and applied to varKode classification. Moreover, we have implemented a multi-label model as the default to increase robustness to low-quality varKodes with little diagnostic information in the training set. This was done by using an asymmetric multi-label loss function[82] instead of the standard cross-entropy loss function used in single-label classification. Analyses used development versions of *varKoder* starting with v.0.8.0. Improvements suggested during the peer-review process are now implemented in *varKoder* v.1.1.0.

varKoder evaluation and comparison to alternatives

varKoder—To test *varKoder* performance on a complex dataset spanning multiple taxonomic levels and varying phylogenetic depths, we used the Malpighiales dataset including genera in Elatinaceae, Chrysobalanaceae and Malpighiaceae. Species of *Stigmaphyllon* (Malpighiaceae) were labeled with species, genus, and family names; all other samples were labeled with genus and family names. We tested the performance of *varKoder* in each sample with leave-one-out cross-validation. For each sample, we retained it as validation and trained a neural network using all the other samples. In preliminary assessments, we found that a ViT[72] architecture combined with a multi-label model sometimes led to instability in training for some datasets. For that reason, we used a two-step approach. Models first were pre-trained for 20 epochs as single-label, using the least inclusive taxonomic assignment available for each sample and a base learning rate of 0.05. Next, we trained for an additional 10 epochs using the pre-trained weights but with a much smaller learning rate (0.005) and a multi-label output. Training samples included varKodes from 500 Kbp to 200 Mbp, and we recorded validation accuracy separately for varKodes produced from each amount of data. We used an arbitrary confidence threshold of 0.7 to make predictions in the multilabel models. For validation samples, we deemed a prediction correct if only the correct taxon was predicted for each taxonomic rank (i.e., species, genus, family). We deemed a prediction incorrect if one or more predictions passed the threshold for a taxonomic rank, but none match the actual label. When predicted labels included both the correct and incorrect taxa, we deemed it ambiguous. If the output prediction included no taxon with confidence above the threshold, we considered it as inconclusive. As metrics across all samples, we used prediction and recall, averaged across all predictions. We visualized the fraction of correct, incorrect, ambiguous, and inconclusive samples for each taxonomic rank and each amount of data used to produce varKodes. The code to reproduce training conditions and evaluation tests is available on GitHub (https://www.github.com/brunoasm/varkoder_development).

To test the joint effect of neural network architecture and image representation method, we applied this cross-validation approach to all combinations of three image representations and four neural network architectures. The architectures tested included: (1) *ResNeXt101*[77], the optimal convolutional neural network architecture in our initial tests, (2) *ViT*[72], a transformer-based architecture that became available after our initial testing, (3) a neural network with two convolutional layers processing vectorized k-mer counts, following Fiannaca et al[44] and (4) a multi-layer perceptron formed by a series of fully connected layers as specified in Millán Arias et al[42]. The two latter have been previously employed for *fCGR* data. The three representations tested include *varKodes* and *rfCGRs* as developed here, and *fCGRs* as estimated by iDeLUCS[93]. In the latter case, we used iDeLUCS functions to produce *fCGRs* as 2D python arrays of k-mer counts. Next, we rescaled these counts to the range of 0–255 and rounded them to the nearest integer. These arrays were then saved as 8-bit png images. In all tests, we employed the same data augmentation methods and loss function as for *varKodes* and *rfCGRs*. All code used in *varKoder* analyses is available on GitHub (https://www.github.com/brunoasm/varkoder_development)**.**

Skmer—To compare *varKoder* with alternative methods, we used fastq files cleaned and subsampled by *varKoder* as input files to *Skmer*. In this case, we also used leave-one-out cross-validation to evaluate performance. For each amount of input data (500Kbp to 200Mbp), we cycled through all samples, constructing a *Skmer* database with the "*skmer reference*" command and including all samples but one and default settings. We then used the "*skmer query*" command with default settings on the sample left out and deemed the identification as correct if the sample in the reference database with closest estimated genetic distance had the correct taxon label. Because *Skmer* could always query a sample and there is no objective criterion to consider matches beyond the best match, the output predictions can only be correct or incorrect, but not inconclusive or ambiguous. We visualized the results similarly as we did with *varKoder*. The code to reproduce *Skmer* analyses is available on GitHub (https://www.github.com/brunoasm/varkoder_development)**.**

Conventional plant barcodes —For conventional barcodes, we applied standard BLAST- and phylogeny-based methods, which do not involve machine learning. To infer phylogenies from our genome skim data (Figure 1), we applied the *PhyloHerb* bioinformatic pipeline[120], which has been applied recently to a taxa ranging from algae to flowering plants[121–123]. Briefly, this pipeline works as follows: for plastid loci, *PhyloHerb* maps raw short reads to a database of land plant plastid genomes. Mapped reads are then assembled into scaffolds using *SPAdes*[124] and plastid loci are identified using nucleotide BLAST searches with a default e-value threshold of 1e-40. *PhyloHerb* then outputs orthologous plastid genes into individual FASTA files, which are fed directly into MAFFT v7.407[125] for alignment. Alignments are then concatenated into a super matrix using the 'conc' function within the *PhyloHerb* package. Phylogenies for both individual locus and the concatenated alignment were inferred with IQTREE v2.0.6 using the GTR+GAMMA model with 1000 ultrafast bootstrap replicates[126].

To recover the conventional plant barcodes, *rbc*L, *mat*K, *trn*L-F, *ndh*F, and ITS, from our Malpighiales genome skim data, we applied GetOrganelle v1.7.7.0[127] and *PhyloHerb* v1.1.1[120] to automatically assemble and extract these DNA markers, respectively. Briefly, the complete or subsampled genome skim data were first assembled into plastid genomes or nuclear ribosomal regions using *GetOrganelle* with its default settings. Next, *PhyloHerb* was applied to extract the relevant barcode genes using its built-in BLAST database. To test whether these traditional barcodes provided accurate identification to species, genus, and family, we ran an all-by-all BLASTn analysis for each individual gene across the same data subsampling schemes as *Skmer* and *varKoder*. BLAST targets were always drawn from assemblies using all the data available for each specimen, whereas queries included assemblies from input data amounts varying from 500 Kbp to 200 Mbp. Within each BLAST analysis for each one of the Malpighiales accessions, we deemed an identification to be correct if the best non-self BLAST hit came from the same taxon, and incorrect otherwise. We deemed it inconclusive if the locus could not be assembled for that amount of data. For concatenated barcodes, we produced a phylogenetic tree for each amount of data and deemed an identification to be correct if the sample with lowest patristic distance came from the same taxon. We deemed it to be inconclusive when none of the genes in the concatenated dataset could be assembled for a sample. We visualized results similarly to *varKoder*, separately for each conventional barcoding gene and for the concatenated dataset. The code to reproduce conventional barcode analyses is available on GitHub (https://www.github.com/brunoasm/varkoder_development).

iDeLUCS—To evaluate the performance of *varKoder* with another deep learning based sequence classifier, we applied the sequences assembled from the *PhyloHerb* pipeline to *iDeLUCS*[93]. We first used concatenated sequences of five traditional plant barcodes (*rbc*L, *mat*K, *trn*L-F, *ndh*F, and ITS) assembled from input reads varying from 500 Kbp to 200 Mbp. *iDeLUCS* was run with k-mer length of 6, 100 training epochs, 100 data augmentations per sequence, and the SGD algorithm for neural network optimization. Unlike varKoder, iDeLUCS does unsupervised clustering and therefore does not use labels during training. Instead, all input accessions were set to be clustered into 10 groups (equal to the total number of species) and the accuracy was evaluated with the *cluster_acc* function implemented in *iDeLUCS.* We also applied the entire plastid genome and the nuclear ribosomal sequence assemblies (ETS+18S+ITS1+5.8S+ITS2+28S) in *iDeLUCS* with the same parameters to evaluate the impact of input data quality.

Application in diverse taxa

Species-level identification in plants, animals, fungi, and bacteria— For all four test cases (*Corallorhiza*, *Bembidion*, *Xanthoparmelia*, and *Mycobacterium tubercolosum*), we applied default *varKoder* v.0.8.0 parameters for generating *rfCGR* images, training each model, and testing the accuracy of the trained model using the 'query' function. In all cases, we included all the available data for each training or validation sample. To test if trained models accurately predicted species identity, we queried them using extra genome skim samples not used for training but from the same species included in the model. We also tested genome skim test samples of species within the same genus *not* used in model training. As in the case of Malpighiales, we set the threshold to make a prediction equal to 0.7 and used the same criteria to consider a prediction correct, incorrect, inconclusive, or ambiguous. We separately evaluated results for taxa with representatives included in the training set and taxa used only as queries, without conspecific samples in the training set. The code to reproduce these analyses is available on GitHub (https://www.github.com/brunoasm/varkoder_development).

All eukaryotic families data set from SRA—Each accession was labeled with its family identification obtained from NCBI. Because of the larger size of this dataset, a leave-one-out cross-validation approach would have been intractable. Therefore, we randomly selected 80% of the samples in each family as the training set and used the remainder for validation. Similarly to Malpighiales, we used a two-step training method by pre-training as a single-label model and finalizing with a multi-label model. Pre-training was done with a learning rate of 0.1 and a batch size of 300 for 30 epochs. Final training was done with the same batch size but a smaller base learning rate of 0.01 in 5 epochs with frozen body weights and three epochs with unfrozen weights. The code to reproduce these analyses is available on GitHub (https://www.github.com/brunoasm/varkoder_development).

All taxa from SRA— For each accession, we created *rfCGRs* from 500Kbp to 10Mbp of data. Each accession was labeled with all the taxa in its taxonomic tree (that is, from infra-specific taxa to domains of life), as well as library strategy (RAD, GBS or WGS) and sequencing platform (Illumina, PACBIO, Nanopore or BGISEQ). We randomly selected 10% of the samples as validation set, and eliminated from validation samples all labels absent from the training set. We used a two-step training method. First, we pre-trained using a single-label strategy, using as labels the concatenation of library strategy, sequencing platform, kingdom, family and genus. For pretraining, we used a learning rate of 0.1, a batch size of 500 and 30 epochs. We then used the weights of this pre-trained model as starting weights for a multi-label model including all labels. We trained the model for additional 50 epochs with unfrozen body weights and 10 epochs with frozen weights, learning rate of 0.05 and batch size of 600. The code to reproduce these analyses is available on GitHub (https://www.github.com/brunoasm/varkoder_development).

Environmental metagenome global identification—The downloaded soil metagenomes from Ma et al.[99] were labeled by source continent. Similarly to the eukaryotic family data set from SRA, we randomly selected 80% of the samples as the training set and used the remaining 20% as the validation set. We used a two-step training method by pre-training as a single-label model and finalizing with a multi-label model. Pre-training was done with a learning rate of 0.1 and a batch size of 64 for 30 epochs. Final training was done with the same batch size but a smaller base learning rate of 0.01 in 5 epochs with frozen body weights and three epochs with unfrozen weights. The code to reproduce all these analyses is available on GitHub (https://www.github.com/brunoasm/varkoder_development).

References

1. Hebert, P. D. N., Ratnasingham, S. & de Waard, J. R. Barcoding animal life: cytochrome c oxidase subunit 1 divergences among closely related species. Proc. R. Soc. Lond. B Biol. Sci. 270, S96–S99 (2003).

2. Kress, W. J. Plant DNA barcodes: Applications today and in the future. J. Syst. Evol. 55, 291–307 (2017).

3. Ratnasingham, S. & Hebert, P. D. N. BOLD: The Barcode of Life Data System (www.barcodinglife.org). Mol. Ecol. Notes 7, 355–364 (2007).

4. Taberlet, P., Coissac, E., Pompanon, F., Brochmann, C. & Willerslev, E. Towards next-generation biodiversity assessment using DNA metabarcoding. Mol. Ecol. 21, 2045–2050 (2012).

5. Seifert, K. A. Progress towards DNA barcoding of fungi. Mol. Ecol. Resour. 9 Suppl s1, 83–89 (2009).

6. Sharkey, M. J. et al. Minimalist revision and description of 403 new species in 11 subfamilies of Costa Rican braconid parasitoid wasps, including host records for 219 species. ZooKeys 1013, 1–665 (2021).

7. Lahaye, R. et al. DNA barcoding the floras of biodiversity hotspots. Proc. Natl. Acad. Sci. U. S. A. 0709936105 (2008) doi:10.1073/pnas.0709936105.

8. Kuzmina, M. L. et al. Using herbarium-derived DNAs to assemble a large-scale DNA barcode library for the vascular plants of Canada. Appl. Plant Sci. 5, apps.1700079 (2017).

9. Muñoz-Rodríguez, P. et al. A taxonomic monograph of Ipomoea integrated across phylogenetic scales. Nat. Plants 5, 1136–1144 (2019).

10. Hebert, P. D., Penton, E. H., Burns, J. M., Janzen, D. H. & Hallwachs, W. Ten species in one: DNA barcoding reveals cryptic species in the neotropical skipper butterfly Astraptes fulgerator. Proc. Natl. Acad. Sci. U. S. A. 101, 14812–14817 (2004).

11. Zeale, M. R., Butlin, R. K., Barker, G. L., Lees, D. C. & Jones, G. Taxon-specific PCR for DNA barcoding arthropod prey in bat faeces. Mol. Ecol. Resour. 11, 236–244 (2011).

12. Nitta, J. H., Meyer, J., Taputuarai, R. & Davis, C. C. Life cycle matters: DNA barcoding reveals contrasting community structure between fern sporophytes and gametophytes. Ecol. Monogr. 87, 278–296 (2016).

13. Kress, W. J. et al. Plant DNA barcodes and a community phylogeny of a tropical forest dynamics plot in Panama. Proc. Natl. Acad. Sci. U. S. A. 106, 18621–18626 (2009).

14. Willis, C. G., Franzone, B. F., Xi, Z. & Davis, C. C. The establishment of Central American migratory corridors and the biogeographic origins of seasonally dry tropical forests in Mexico. Front. Genet. 5, 433 (2014).

15. Willerslev, E. et al. Ancient biomolecules from deep ice cores reveal a forested Southern Greenland. Science 317, 111–114 (2007).

16. Crump, S. E. et al. Ancient plant DNA reveals High Arctic greening during the Last Interglacial. Proc. Natl. Acad. Sci. U. S. A. 118, e2019069118 (2021).

17. Kjær, K. H. et al. A 2-million-year-old ecosystem in Greenland uncovered by environmental DNA. Nature 612, 283–291 (2022).

18. Fierer, N. et al. Forensic identification using skin bacterial communities. Proc. Natl. Acad. Sci. 107, 6477–81 (2010).

19. Rollo, F., Ubaldi, M., Ermini, L. & Marota, I. Ötzi's last meals: DNA analysis of the intestinal content of the Neolithic glacier mummy from the Alps. Proc. Natl. Acad. Sci. U. S. A. 99, 12594–12599 (2002).

20. Yu, J. et al. Progress in the use of DNA barcodes in the identification and classification of medicinal plants. Ecotoxicol. Environ. Saf. 208, 111691 (2021).

21. Ashfaq, M. & Hebert, P. D. N. DNA barcodes for bio-surveillance: regulated and economically important arthropod plant pests. Genome 59, 933–945 (2016).

22. Eaton, M. J. et al. Barcoding bushmeat: molecular identification of Central African and South American harvested vertebrates. Conserv. Genet. 11, 1389–1404 (2010).

23. Liu, J. et al. Integrating a comprehensive DNA barcode reference library with a global map of yews (Taxus L.) for forensic identification. Mol. Ecol. Resour. 18, 1115–1131 (2018).

24. Ogden, R., Dawnay, N. & McEwing, R. Wildlife DNA forensics—bridging the gap between conservation genetics and law enforcement. Endanger. Species Res. 9, 179–195 (2009).

25. Williamson, J. et al. Exposing the illegal trade in cycad species (Cycadophyta: Encephalartos) at two traditional medicine markets in South Africa using DNA barcoding. Genome 59, 771–781 (2016).

26. Costa, F. O. & Carvalho, G. R. The Barcode of Life Initiative: synopsis and prospective societal impacts of DNA barcoding of Fish. Genomics Soc. Policy 3, 29 (2007).

27. Gao, Z., Liu, Y., Wang, X., Wei, X. & Han, J. DNA mini-barcoding: a derived barcoding method for herbal molecular identification. Front. Plant Sci. 10, (2019).

28. Molina, J. et al. Possible loss of the chloroplast genome in the parasitic flowering plant Rafflesia lagascae (Rafflesiaceae). Mol. Biol. Evol. 31, 793–803 (2014).

29. Cai, L. et al. Deeply altered genome architecture in the endoparasitic flowering plant Sapria himalayana Griff. (Rafflesiaceae). Curr. Biol. 31, 1002-1011.e9 (2021).

30. Richardson, J. E., Pennington, R. T., Pennington, T. D. & Hollingsworth, P. M. Rapid diversification of a species-rich genus of neotropical rain forest trees. Science 293, 2242–2245 (2001).

31. Wang, J., Luo, J., Ma, Y.-Z., Mao, X.-X. & Liu, J.-Q. Nuclear simple sequence repeat markers are superior to DNA barcodes for identification of closely related Rhododendron species on the same mountain. J. Syst. Evol. 57, 278–286 (2019).

32. Su, X., Wu, G., Li, L. & Liu, J. Species delimitation in plants using the Qinghai–Tibet Plateau endemic Orinus (Poaceae: Tridentinae) as an example. Ann. Bot. 116, 35–48 (2015).

33. Lu, Z. et al. Species delimitation and hybridization history of a hazel species complex. Ann. Bot. 127, 875–886 (2021).

34. Cai, L. et al. The perfect storm: gene tree estimation error, incomplete lineage sorting, and ancient gene flow explain the most recalcitrant ancient angiosperm clade, Malpighiales. Syst. Biol. 70, 491–507 (2021).

35. Clarke, L. J., Soubrier, J., Weyrich, L. S. & Cooper, A. Environmental metabarcodes for insects: in silico PCR reveals potential for taxonomic bias. Mol. Ecol. Resour. 14, 1160–1170 (2014).

36. Song, H., Buhay, J. E., Whiting, M. F. & Crandall, K. A. Many species in one: DNA barcoding overestimates the number of species when nuclear mitochondrial pseudogenes are coamplified. Proc. Natl. Acad. Sci. U. S. A. 105, 13486–13491 (2008).

37. Xiong, H. et al. Species tree estimation and the impact of gene loss following whole-genome duplication. Syst. Biol. 71, 1348–1361 (2022).

38. Straub, S. C. K. et al. Navigating the tip of the genomic iceberg: Next-generation sequencing for plant systematics. Am. J. Bot. 99, 349–364 (2012).

39. Bohmann, K., Mirarab, S., Bafna, V. & Gilbert, M. T. P. Beyond DNA barcoding: The unrealised potential of genome skim data in sample identification. Mol. Ecol. 1–14 (2020) doi:10.1111/mec.15507.

40. Sarmashghi, S., Bohmann, K., P. Gilbert, M. T., Bafna, V. & Mirarab, S. Skmer: assembly-free and alignment-free sample identification using genome skims. Genome Biol. 20, 34 (2019).

41. Borowiec, M. L. et al. Deep learning as a tool for ecology and evolution. Methods Ecol. Evol. 13, 1640–1660 (2022).

42. Arias, P. M., Alipour, F., Hill, K. A. & Kari, L. DeLUCS: Deep learning for unsupervised clustering of DNA sequences. PLOS ONE 17, e0261531 (2022).

43. Kari, L. et al. Mapping the space of genomic signatures. PLOS ONE 10, e0119815 (2015).

44. Fiannaca, A. et al. Deep learning models for bacteria taxonomic classification of metagenomic data. BMC Bioinformatics 19, (2018).

45. Linard, B., Swenson, K. & Pardi, F. Rapid alignment-free phylogenetic identification of metagenomic sequences. Bioinformatics (2019) doi:10.1093/bioinformatics/btz068.

46. Desai, H. P., Parameshwaran, A. P., Sunderraman, R. & Weeks, M. Comparative Study Using Neural Networks for 16S Ribosomal Gene Classification. J. Comput. Biol. 27, 248–258 (2020).

47. Shang, J. & Sun, Y. CHEER: HierarCHical taxonomic classification for viral mEtagEnomic data via deep leaRning. Methods 189, 95–103 (2021).

48. Arias, P. M. et al. BarcodeBERT: Transformers for Biodiversity Analysis. Preprint at http://arxiv.org/abs/2311.02401 (2023).

49. Badirli, S., Akata, Z., Mohler, G., Picard, C. & Dundar, M. Fine-Grained Zero-Shot Learning with DNA as Side Information. Preprint at http://arxiv.org/abs/2109.14133 (2021).

50. Lecun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015).

51. Cong, Y., Ye, X., Mei, Y., He, K. & Li, F. Transposons and non-coding regions drive the intrafamily differences of genome size in insects. iScience 25, 104873 (2022).

52. Heckenhauer, J. et al. Genome size evolution in the diverse insect order Trichoptera. GigaScience 11, 1–19 (2022).

53. Schley, R. J. et al. The ecology of palm genomes: repeat-associated genome size expansion is constrained by aridity. New Phytol. 433–446 (2022) doi:10.1111/nph.18323.

54. Sproul, J. S., Barton, L. M. & Maddison, D. R. Repetitive DNA profiles Reveal Evidence of Rapid Genome Evolution and Reflect Species Boundaries in Ground Beetles. Syst. Biol. 0, 1–12 (2020).

55. de Medeiros, B. A. S. & Farrell, B. D. Whole-genome amplification in double-digest RADseq results in adequate libraries but fewer sequenced loci. PeerJ 6, e5089 (2018).

56. Jeffrey, H. J. Chaos game representation of gene structure. Nucleic Acids Res. 18, 2163–2170 (1990).

57. Deschavanne, P. J., Giron, A., Vilain, J., Fagot, G. & Fertil, B. Genomic signature: characterization and classification of species assessed by chaos game representation of sequences. Mol. Biol. Evol. 16, 1391–1399 (1999).

58. de la Fuente, R., Díaz-Villanueva, W., Arnau, V. & Moya, A. Genomic Signature in Evolutionary Biology: A Review. Biology 12, 322 (2023).

59. Avila Cartes, J., Anand, S., Ciccolella, S., Bonizzoni, P. & Della Vedova, G. Accurate and fast clade assignment via deep learning and frequency chaos game representation. GigaScience 12, giac119 (2023).

60. Solis-Reyes, S., Avino, M., Poon, A. & Kari, L. An open-source k-mer based machine learning tool for fast and accurate subtyping of HIV-1 genomes. PLOS ONE 13, e0206409 (2018).

61. Kislyuk, A., Bhatnagar, S., Dushoff, J. & Weitz, J. S. Unsupervised statistical clustering of environmental shotgun sequences. BMC Bioinformatics 10, 316 (2009).

62. Arias, P. M. et al. Environment and taxonomy shape the genomic signature of prokaryotic extremophiles. Sci. Rep. 13, 16105 (2023).

63. Murad, T., Ali, S., Khan, I. & Patterson, M. Spike2CGR: an efficient method for spike sequence classification using chaos game representation. Mach. Learn. 112, 3633–3658 (2023).

64. Paszke, A. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. in Advances in Neural Information Processing Systems 32 8024–8035 (Curran Associates, Inc., 2019).

65. Davis, C. C. & Anderson, W. R. A complete generic phylogeny of Malpighiaceae inferred from nucleotide sequence data and morphology. Am. J. Bot. 97, 2031–2048 (2010).

66. Cai, L. et al. Phylogeny of Elatinaceae and the tropical Gondwanan origin of the Centroplacaceae (Malpighiaceae, Elatinaceae) clade. PLOS ONE 11, e0161881 (2016).

67. Davis, C. C., Anderson, W. R. & Donoghue, M. J. Phylogeny of Malpighiaceae: evidence from chloroplast ndhF and trnL-F nucleotide sequences. Am. J. Bot. 88, 1830–1846 (2001).

68. Anderson, C. Revision of Ryssopterys and transfer to Stigmaphyllon (Malpighiaceae). Blumea 56, 73–104 (2011).

69. Anderson, C. Monograph of Stigmaphyllon (Malpighiaceae). Syst. Bot. Monogr. 51, 1–313 (1997).

70. He, T. et al. Bag of Tricks for Image Classification with Convolutional Neural Networks. Preprint at http://arxiv.org/abs/1812.01187 (2018).

71. Vaswani, A. et al. Attention Is All You Need. Preprint at https://doi.org/10.48550/arXiv.1706.03762 (2017).

72. Dosovitskiy, A. et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. (2021).

73. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the Inception Architecture for Computer Vision. in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2818–2826 (2016). doi:10.1109/CVPR.2016.308.

74. Smith, L. N. A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay. Preprint at http://arxiv.org/abs/1803.09820 (2018).

75. Zhang, H., Cisse, M., Dauphin, Y. N. & Lopez-Paz, D. mixup: Beyond Empirical Risk Minimization. Preprint at http://arxiv.org/abs/1710.09412 (2018).

76. Yun, S. et al. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. Preprint at https://doi.org/10.48550/arXiv.1905.04899 (2019).

77. Xie, S., Girshick, R., Dollár, P., Tu, Z. & He, K. Aggregated Residual Transformations for Deep Neural Networks. Preprint at https://doi.org/10.48550/arXiv.1611.05431 (2017).

78. Goodfellow, I., Bengio, Y. & Courville, A. Deep Learning. (MIT Press, 2016).

79. Christin, S., Hervet, É. & Lecomte, N. Applications for deep learning in ecology. Methods Ecol. Evol. 10, 1632–1644 (2019).

80. Weiß, C. L. et al. Temporal patterns of damage and decay kinetics of DNA retrieved from plant herbarium specimens. R. Soc. Open Sci. 3, 160239 (2016).

81. Rachtman, E., Balaban, M., Bafna, V. & Mirarab, S. The impact of contaminants on the accuracy of genome skimming and the effectiveness of exclusion read filters. Mol. Ecol. Resour. 20, 649–661 (2020).

82. Ben-Baruch, E. et al. Asymmetric Loss For Multi-Label Classification. Preprint at http://arxiv.org/abs/2009.14119 (2021).

83. Bushnell, B. BBMap. (2022).

84. Bushnell, B., Rood, J. & Singer, E. BBMerge – Accurate paired shotgun read merging via overlap. PLOS ONE 12, e0185056 (2017).

85. Rizk, G., Lavenier, D. & Chikhi, R. DSK: k-mer counting with very low memory usage. Bioinformatics 29, 652–653 (2013).

86. Chen, S., Zhou, Y., Chen, Y. & Gu, J. fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics 34, i884–i890 (2018).

87. Tange, O. GNU Parallel 2018. (Ole Tange, 2018). doi:10.5281/zenodo.1146014.

88. Harris, C. R. et al. Array programming with NumPy. Nature 585, 357–362 (2020).

89. Howard, J. & Gugger, S. Fastai: A Layered API for Deep Learning. Information 11, 108 (2020).

90. Wightman, R. PyTorch Image Models. GitHub repository (2019) doi:10.5281/zenodo.4414861.

91. Pellicer, J. & Leitch, I. J. The Plant DNA C-values database (release 7.1): an updated online repository of plant genome size data for comparative studies. New Phytol. 226, 301–305 (2020).

92. Fiannaca, A., La Rosa, M., Rizzo, R. & Urso, A. Analysis of DNA Barcode Sequences Using Neural Gas and Spectral Representation. in Engineering Applications of Neural Networks (eds. Iliadis, L., Pappadopoulos, H. & Jayne, C.) 212–221 (Springer, Heidelberg, 2013).

93. Millan Arias, P., Hill, K. A. & Kari, L. i DeLUCS: a deep learning interactive tool for alignment-free clustering of DNA sequences. Bioinformatics 39, btad508 (2023).

94. D'Ercole, J., Prosser, S. W. J. & Hebert, P. D. N. A SMRT approach for targeted amplicon sequencing of museum specimens (Lepidoptera)—patterns of nucleotide misincorporation. PeerJ 9, e10420 (2021).

95. Sproul, J. S. & Maddison, D. R. Cryptic species in the mountaintops: species delimitation and taxonomy of the Bembidion breve species group (Coleoptera: Carabidae) aided by

genomic architecture of a century-old type specimen. Zool. J. Linn. Soc. 183, 556–583 (2018).

96. Keuler, R. et al. Interpreting phylogenetic conflict: hybridization in the most speciose genus of lichen-forming fungi. Mol. Phylogenet. Evol. 174, 107543 (2022).

97. Barrett, C. F., Wicke, S. & Sass, C. Dense infraspecific sampling reveals rapid and independent trajectories of plastome degradation in a heterotrophic orchid complex. New Phytol. 218, 1192–1204 (2018).

98. Freschi, L. et al. Population structure, biogeography and transmissibility of Mycobacterium tuberculosis. Nat. Commun. 12, 6099 (2021).

99. Ma, B. et al. A genomic catalogue of soil microbiomes boosts mining of biodiversity and genetic resources. Nat. Commun. 14, 7318 (2023).

100. Asprino, R. et al. A dataset for benchmarking molecular identification tools based on genome skimming. Preprint at https://doi.org/10.32942/X2DW6K (2024).

101. Pomerantz, A. et al. Rapid in situ identification of biological specimens via DNA amplicon sequencing using miniaturized laboratory equipment. Nat. Protoc. 17, 1415–1443 (2022).

102. Kimura, L. T. et al. Amazon Biobank: a collaborative genetic database for bioeconomy development. Funct. Integr. Genomics 23, 101 (2023).

103. Lewin, H. A. et al. The Earth BioGenome Project 2020: starting the clock. Proc. Natl. Acad. Sci. U. S. A. 119, e2115635118 (2022).

104. Ebenezer, T. E. et al. Africa: sequence 100,000 species to safeguard biodiversity. Nature 603, 388–392 (2022).

105. Cheng, S. et al. 10KP: A phylodiverse genome sequencing plan. GigaScience 7, giy013 (2018).

106. Staff, E. A reference standard for genome biology. Nat. Biotechnol. 36, 1121–1121 (2018).

107. i5K Consortium. The i5K Initiative: Advancing Arthropod Genomics for Knowledge, Human Health, Agriculture, and the Environment. J. Hered. 104, 595–600 (2013).

108. Davis, C. C. The herbarium of the future. Trends Ecol. Evol. (2022) doi:10.1016/j.tree.2022.11.015.

109. Davis, C. C. Collections are truly priceless. Science 383, 1035–1035 (2024).

110. Davis, C. C., Sessa, E. B., Paton, A., Antonelli, A. & Teisher, J. The destructive sampling conundrum and guidelines for effective and ethical sampling of herbaria. EcoEvoRxiv (2024) doi:10.32942/X2C603.

111. Card, D. C., Shapiro, B., Giribet, G., Moritz, C. & Edwards, S. V. Museum genomics. Annu. Rev. Genet. 55, 633–659 (2021).

112. Leavitt, S. D. et al. Fungal specificity and selectivity for algae play a major role in determining lichen partnerships across diverse ecogeographic regions in the lichen-forming family Parmeliaceae (Ascomycota). Mol. Ecol. 24, 3779–3797 (2015).

113. Bushnell, B. BBtools v.37.61. (2017).

114. Nissen, J. N. et al. Improved metagenome binning and assembly using deep variational autoencoders. Nat. Biotechnol. 39, 555–560 (2021).

115. Vurture, G. W. et al. GenomeScope: fast reference-free genome profiling from short reads. Bioinformatics 33, 2202–2204 (2017).

116. Maaten, L. van der & Hinton, G. Visualizing Data using t-SNE. J. Mach. Learn. Res. 9, 2579–2605 (2008).

117. Clark, A. Pillow, Version 9.4.0. Software. https://pypi.org/project/Pillow/. (2023).

118. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. arXiv 1512.03385 (2015) doi:10.1109/CVPR.2016.90.

119. Tan, M. & Le, Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ArXiv abs/1905.11946, (2019).

120. Cai, L., Zhang, H. & Davis, C. C. PhyloHerb: A high-throughput phylogenomic pipeline for processing genome skimming data. Appl. Plant Sci. 10, e11475 (2022).

121. Marinho, L. C. et al. Plastomes resolve generic limits within tribe Clusieae (Clusiaceae) and reveal the new genus Arawakia. Mol. Phylogenet. Evol. 134, 142–151 (2019).

122. Lyra, G. de M. et al. Phylogenomics, divergence time estimation and trait evolution provide a new look into the Gracilariales (Rhodophyta). Mol. Phylogenet. Evol. 165, 107294 (2021).

123. Marinho, L. C. et al. Phylogenetic Relationships of Tovomita (Clusiaceae): Carpel Number and Geographic Distribution Speak Louder than Venation Pattern. Syst. Bot. 46, 102–108 (2021).

124. Bankevich, A. et al. SPAdes: a new genome assembly algorithm and Its applications to single-cell sequencing. J. Comput. Biol. 19, 455–477 (2012).

125. Katoh, K. & Standley, D. M. MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. Mol. Biol. Evol. 30, 772–780 (2013).

126. Minh, B. Q. et al. IQ-TREE 2: New Models and Efficient Methods for Phylogenetic Inference in the Genomic Era. Mol. Biol. Evol. 37, 1530–1534 (2020).

127. Jin, J.-J. et al. GetOrganelle: a fast and versatile toolkit for accurate de novo assembly of organelle genomes. Genome Biol. 21, 241 (2020).

Data Availability

New data generated *de novo* genomic for this study is available on NCBI SRA under Bioproject PRJNA1052627. All datasets and metadata are thoroughly described in a companion Data Descriptor article[100] and deposited at Harvard dataverse (https://doi.org/10.7910/DVN/IMOX0S), which will be made public upon manuscript acceptance. A pretrained model on rfCGRs and varKodes for the all-SRA-taxa dataset is available at Huggingface hub (https://huggingface.co/brunoasm/vit_large_patch32_224.NCBI_SRA)

Code Availability

Code used in the initial development and test of varKoder is available on Github (https://www.github.com/brunoasm/varkoder_development). All code used to produce images is available in the development GitHub repository. The current version of varKoder is available at https://github.com/brunoasm/varKoder. Both repositories have been archived upon manuscript submission at the Figshare repository 10.6084/m9.figshare.8304017, and will be made public upon acceptance.