

Supplementary Information

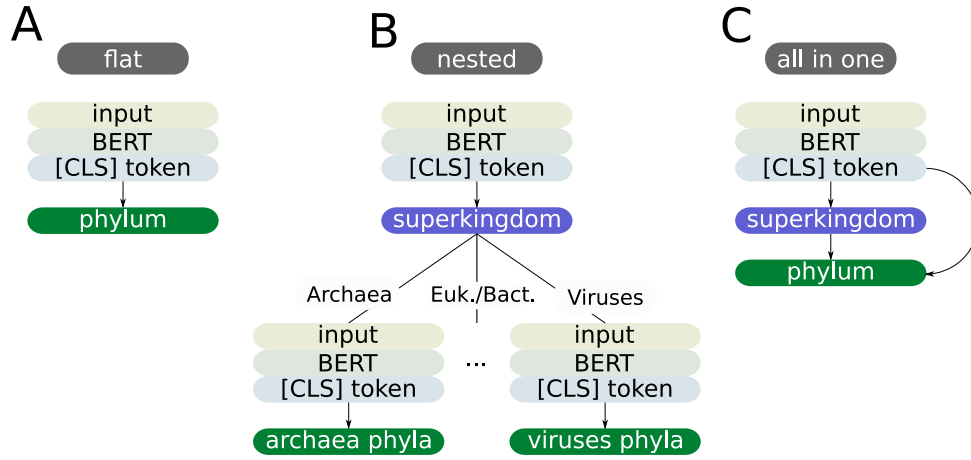


Figure S1: The evaluated architectures. Note this comparison was made on the taxonomic ranks, superkingdom, and phylum. However, the final **BERTax** model can furthermore predict the genus. **A** The **flat architecture** consists of a BERT model which is trained to directly predict the lowest taxonomic rank (the phylum), from which higher taxonomic ranks are inferred. **B** The **nested architecture** consists of multiple BERT models that are nested such that the top model takes each input and classifies the superkingdom and the sample is then further classified by the model that was trained to predict the phyla for that superkingdom. **C** The **all-in-one architecture** consists of a single BERT model which predicts all taxonomic ranks simultaneously. For this, all output layers (all taxonomic ranks) of the model have access to the BERT model itself and the output layer (the prediction) for higher taxonomic ranks.

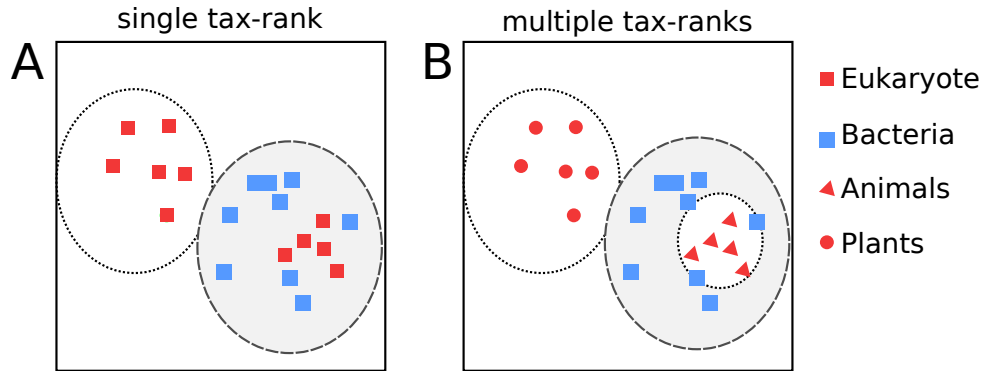


Figure S2: This example shows a possible distribution of training samples. **A** The activation of the output-layer for each sample of the of a model trained on a single rank, here Superkingdom, in a PCA like manner. All samples in the white circle are predicted as Eukaryota and in the gray circle as Bacteria. The figure shows that it can be difficult to correctly classify the red samples in the gray circle, as any adjustment to the model would initially decrease the prediction accuracy. **B** The activation of the output-layer for each sample of the of a model trained on Superkingdom and Kingdom simultaneously. Here, the training for Kingdom automatically adjusts the weights of the model to also address the Plants class, which ultimately also helps the Superkingdom prediction.

Details to PR-curve plots

General Remarks

For all tools, we only consider the top prediction for each sample. This means that less likely predictions (e.g. the second most likely class) are not included. Note that for all samples in the test set **BERTax** provides taxonomic predictions. The test set was balanced regarding the taxonomic rank phylum. Other taxonomic ranks are imbalanced. Furthermore, not all tools return predictions for all samples. Therefore, the subset of samples with predictions can cause imbalance as well. To address this, all samples not predicted are treated as a random guesses. This results in observable steps in the PR curves. The imbalance in the test set itself is addressed by two approaches: On the one hand, a weighted micro-average approach is used, where the weights are chosen so all classes contribute equally. On the other hand, we used the macro-average, where for each class a PR curve is calculated and then the average of these curves is build.

Micro-average PR curves

We weigh the samples in the micro-average PR curves according to the number of predicted samples for each taxonomic class. As with all micro-average PR curves, predictions are ranked solely by the probability of class membership (how certain the method is), regardless of actual class membership. Decreasing thresholds are set for these probabilities, for each of which recall and precision are calculated. Here, the proportion of positively and negatively predicted samples per class could differ significantly depending on the position in the ranking. Due to their specific relationship between threshold and precision-recall, micro-average PR curves represent the precision and recall expected in a real-world application.

macro-average PR curves

These curves show the average performance of all classes, where the performance per class is calculated in an all-vs-one manner. In contrast to the micro-average here, a threshold does not correspond to a specific precision and recall. Instead, the recall (x-axis) corresponds to the same proportion of correct predicted samples per class.

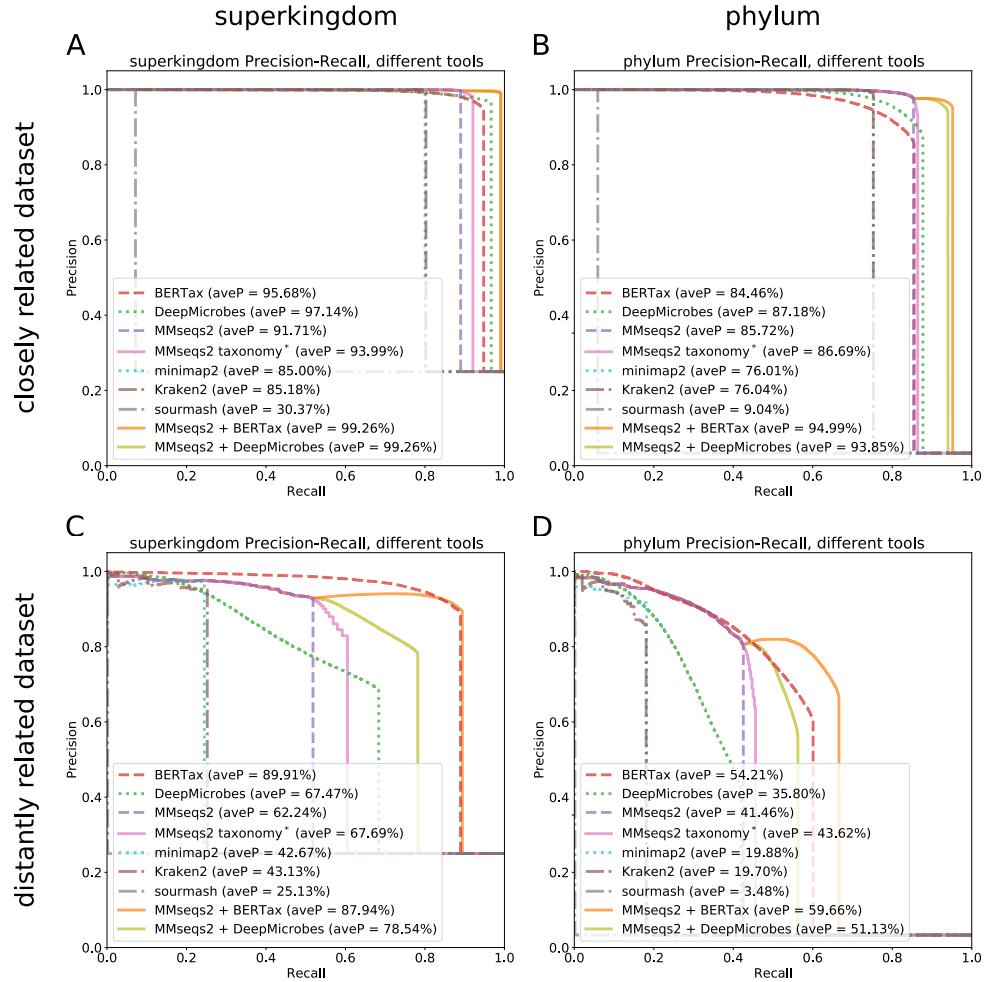


Figure S3: Class weight normalized micro-averaged precision-recall curve for the closely and distantly related datasets. The recall (x-axis) indicates the proportion of all samples for which the method determined the correct superkingdom or phylum. This recall corresponds directly to a specific threshold. However, the proportion of predicted classes can be different for each recall value. This PR-curve represents the ‘real world’ performance using these applications. The y-axis shows the precision achieved for the respective recall. Figures **A** and **B** show the results for the closely related dataset, which comprises samples with the same genus in the query database (test set) as in the target database (training set). Figures **C** and **D** show the distantly related dataset, which comprises samples that do not have closely related (identical genus) samples in the query and target database.

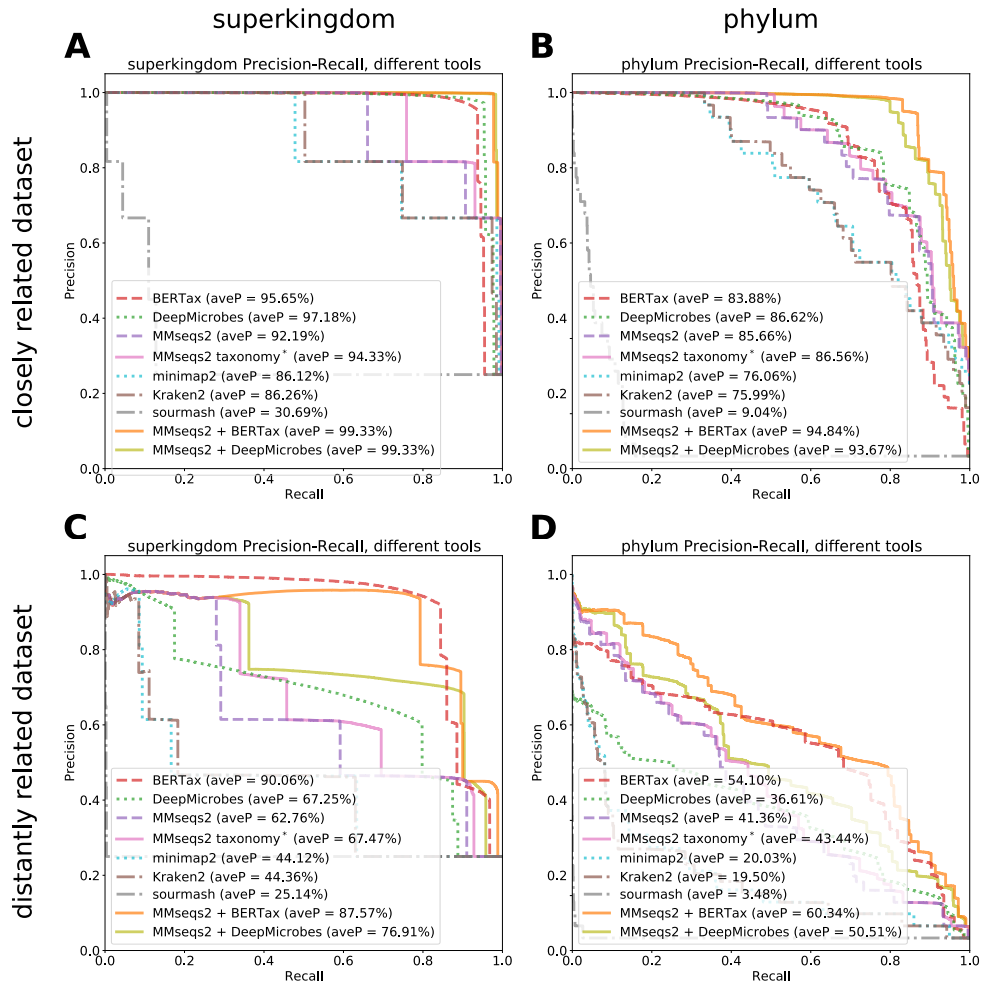


Figure S4: Macro-averaged precision-recall curve for the closely and distantly related datasets. The recall (x-axis) indicates the proportion of samples per class for which the method determined the correct superkingdom or phylum. Therefore, the recall does not correspond to a specific threshold. This PR curve represents the performance normalized to the recall per class. The y-axis shows the precision achieved for the respective recall. Figures **A** and **B** show the results for the closely related dataset, which comprises samples with the same genus in the query database (test set) as in the target database (training set). Figures **C** and **D** show the distantly related dataset, which comprises samples that do not have closely related (identical genus) samples in the query and target database.

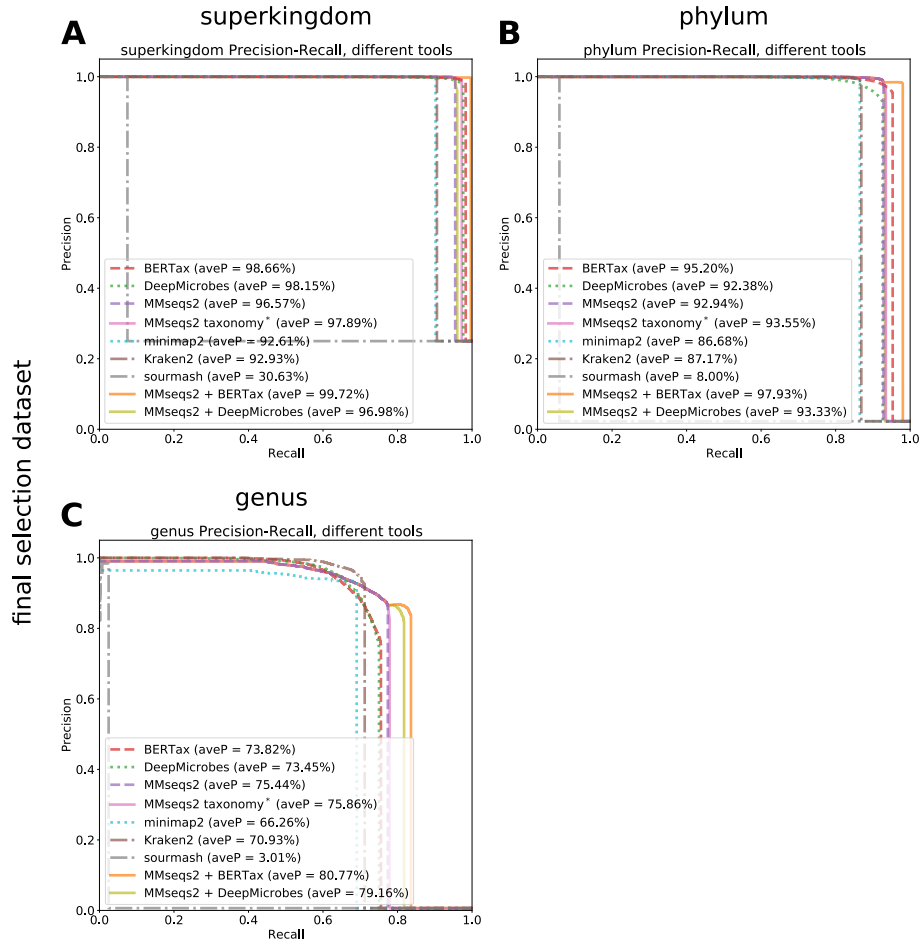


Figure S5: Class weight normalized micro-averaged precision-recall curve on the final selection dataset. The recall (x-axis) indicates the proportion of all samples for which the method determined the correct superkingdom (A), phylum (B) or genus (C). This recall corresponds directly to a specific threshold and thus represents the PR curve that occurs in the application. However, the proportion of predicted classes can be different for each recall value. The y-axis shows the precision achieved for the respective recall.

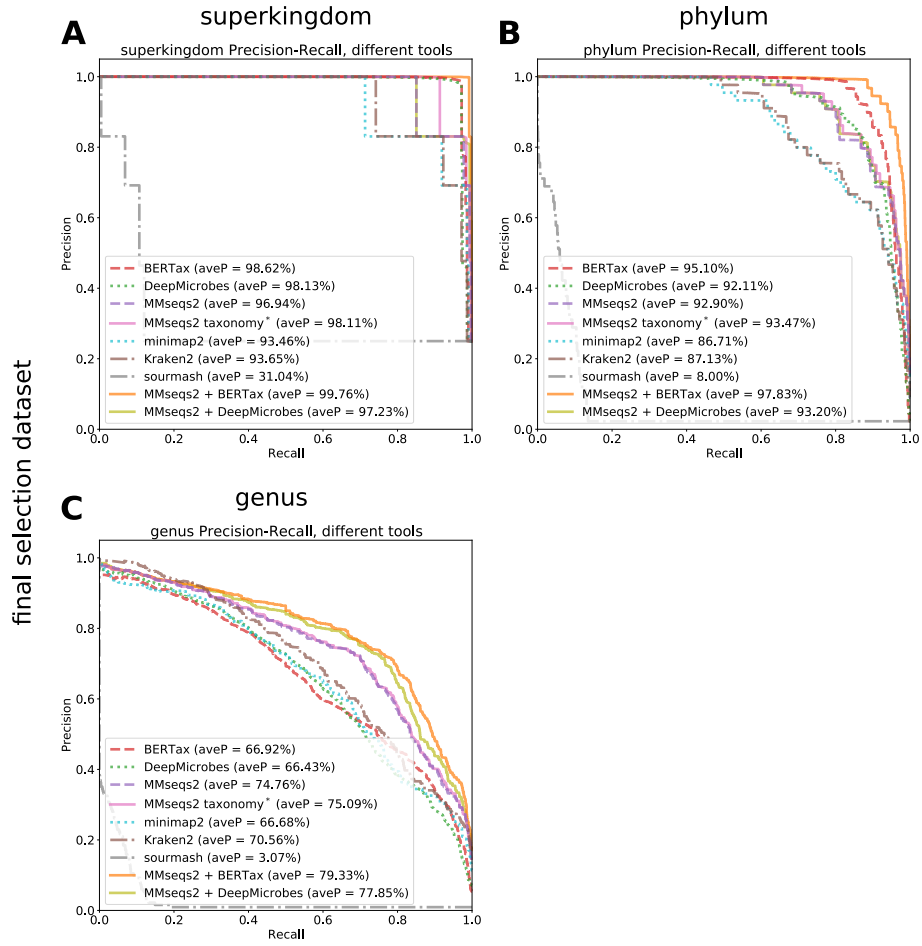


Figure S6: Macro-averaged precision-recall curve on the final selection dataset. The recall (x-axis) indicates the proportion of samples per class for which the method determined the correct superkingdom (**A**), phylum (**B**) or genus (**C**). Therefore, the recall does not correspond to a specific threshold. This PR curve represents the performance normalized to the recall per class. The y-axis shows the precision achieved for the respective recall.

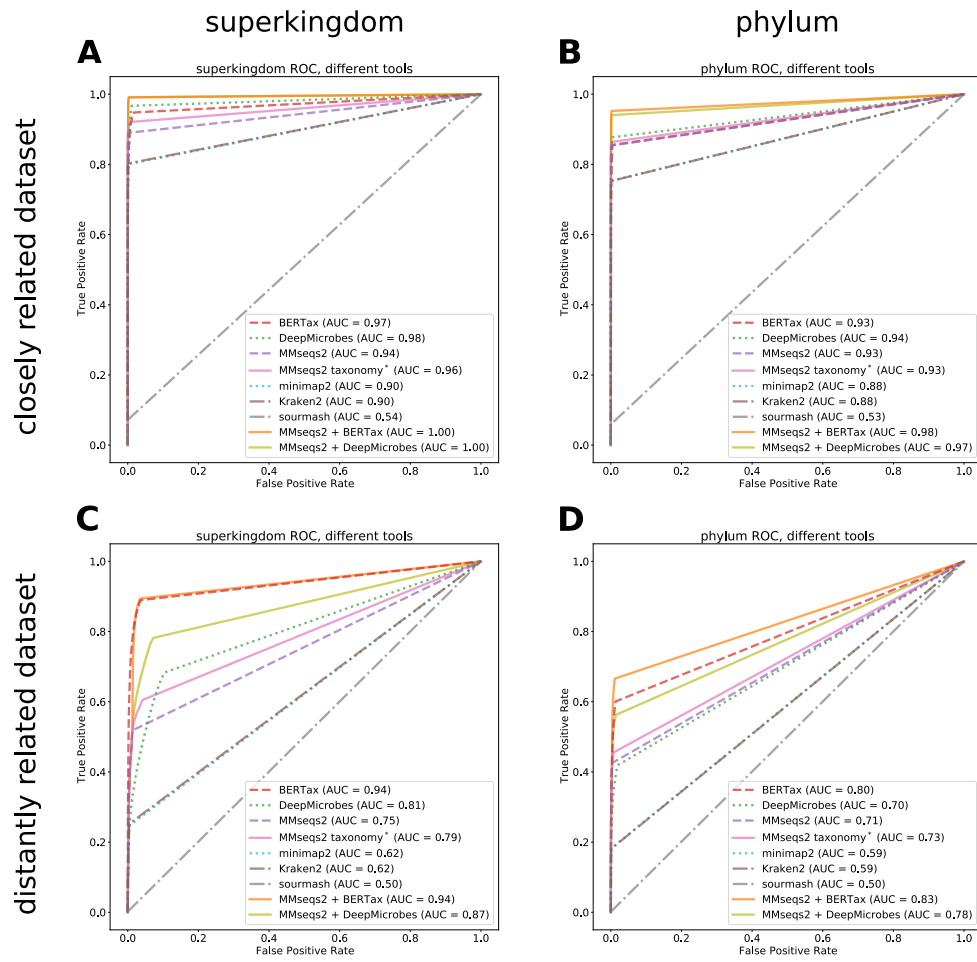


Figure S7: Micro-averaged ROC curve for the closely and distantly related datasets.

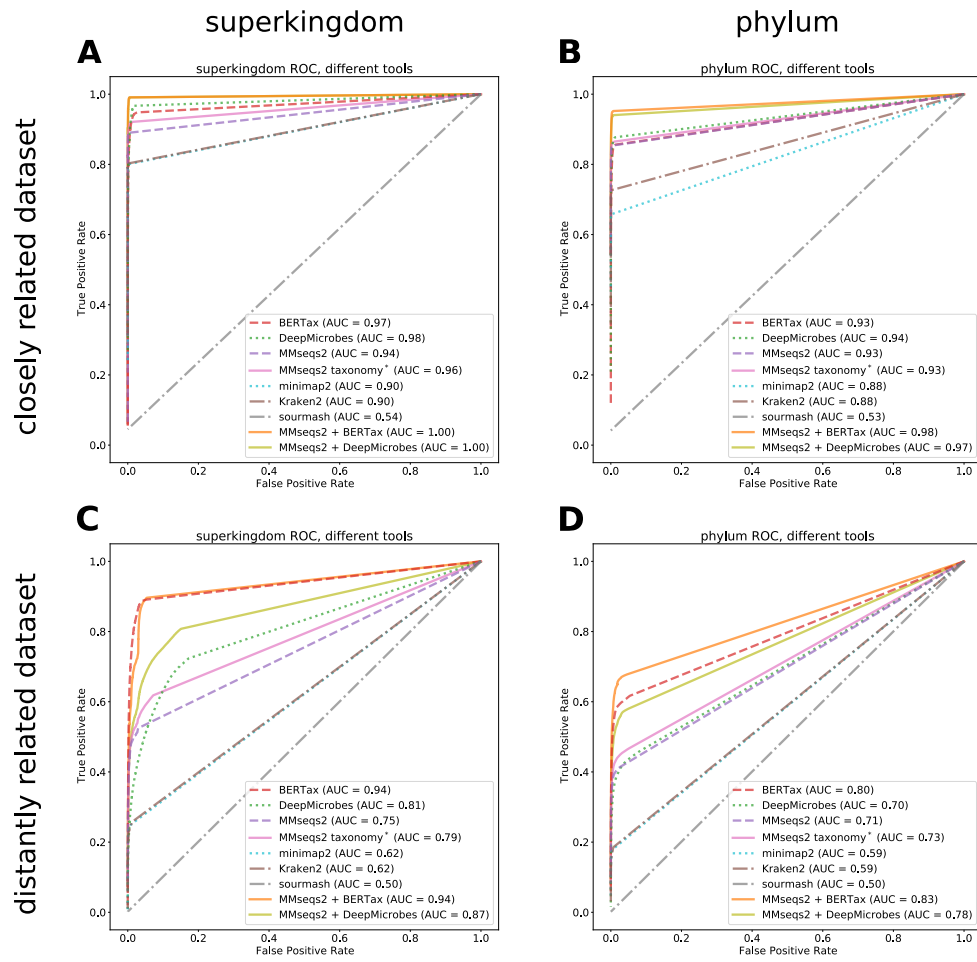


Figure S8: Macro-averaged ROC curve for the closely and distantly related datasets.

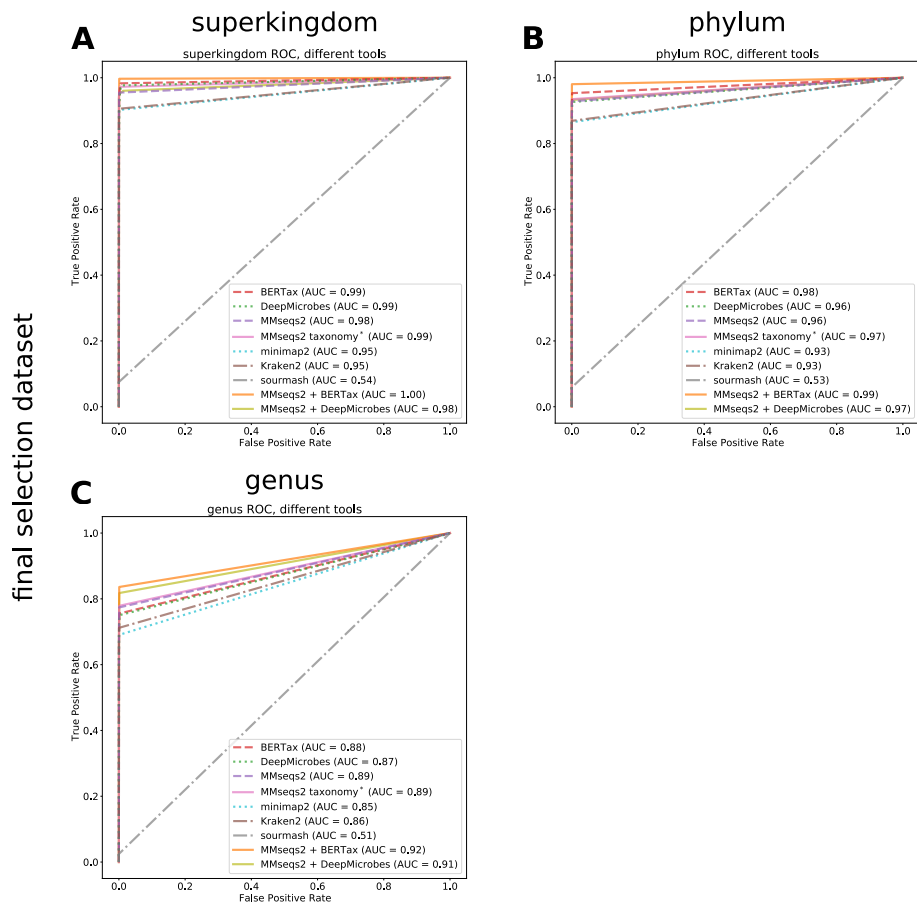


Figure S9: Micro-averaged ROC curve on the final selection dataset.

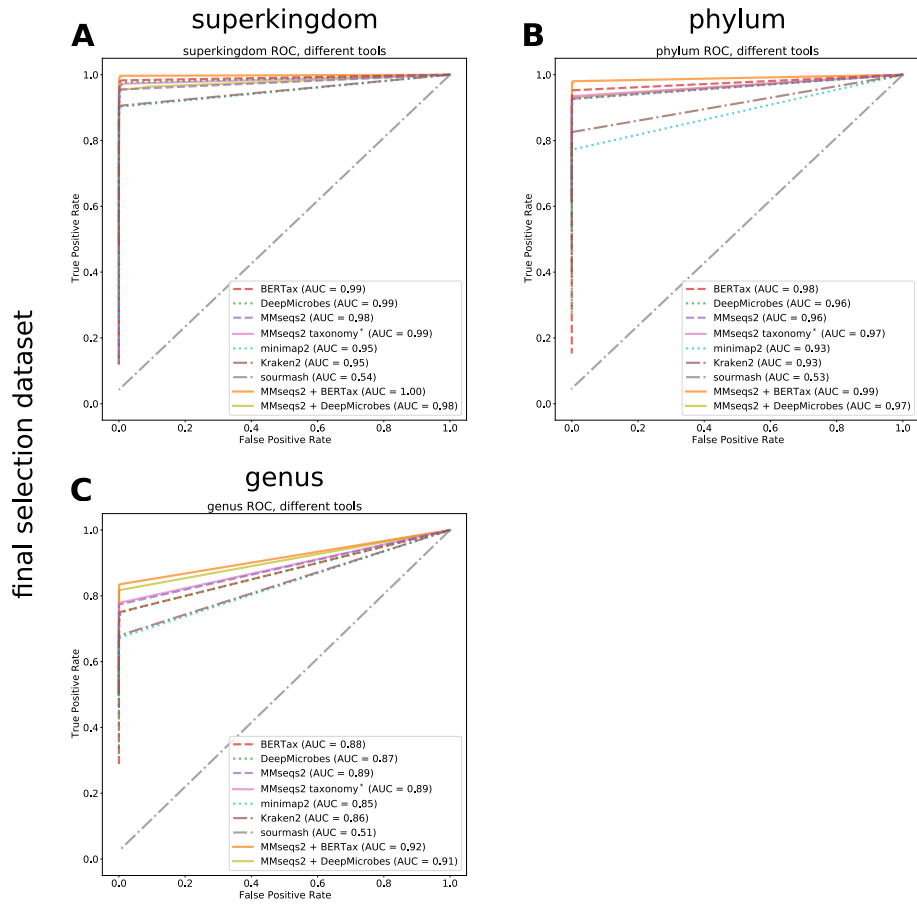


Figure S10: Macro-averaged ROC curve on the final selection dataset.

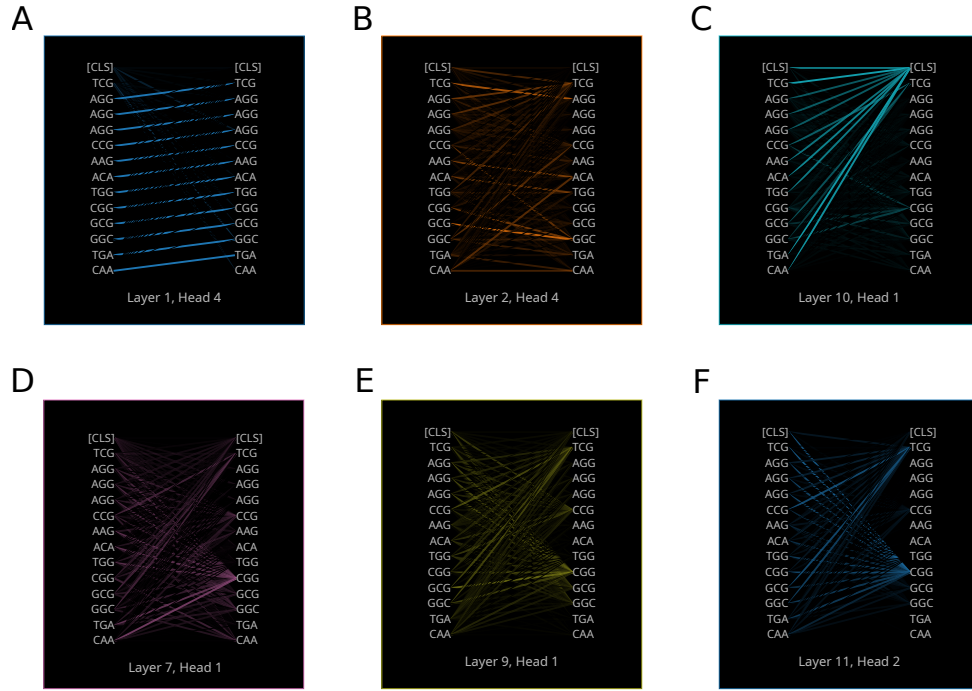


Figure S11: Visualization of the self-attention behaviour of **BERTax**. Self-attention weights between tokens of a short (39 nt) sequence are shown as graphs. The attention heads represent independent relationships between the input tokens of the same sequence. The tokens are visualized from top to bottom as nodes, edges represent weights: thicker edges depict higher weights. **A** First generic pattern in which each token attends to the previous token. **B** Second generic pattern in which each token attends to the next token, albeit less clearly. **C** Pattern which shows high attention towards the generic [CLS] token, which serves to aggregate the information of the whole sequence. **D–F** Three attention heads which show disproportionately high attention for the tokens ‘TCG’, ‘CGG’ and in two cases also for ‘CCG’. It is argued, that for lack of a generic separation token (or a period in natural languages), these tokens in this instance serve a similar purpose of marking segments.

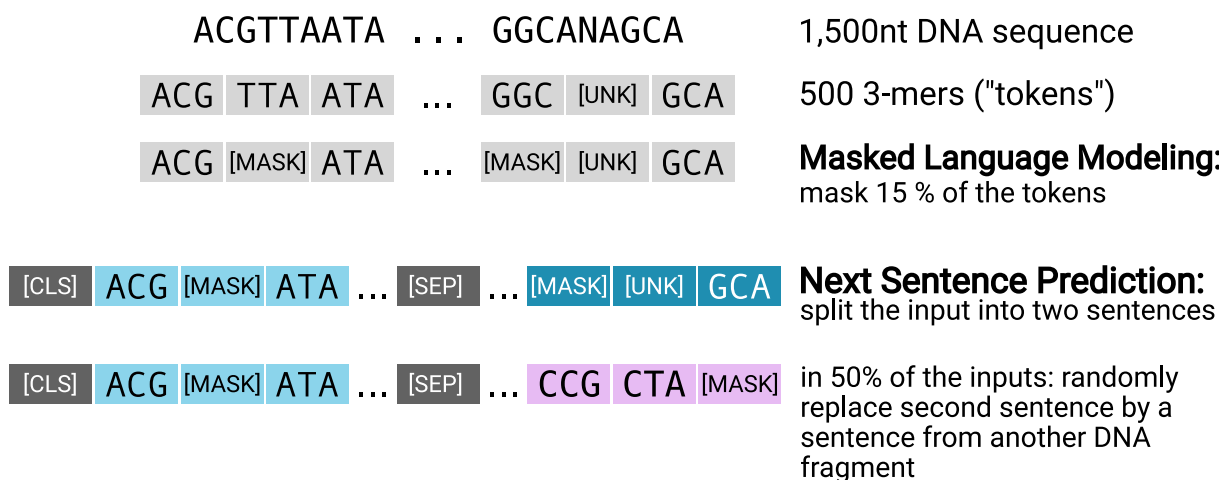


Figure S12: Pre-training of BERT on DNA. DNA sequences of length 1500 are divided into 500 non-overlapping tokens (each representing a 3-mer), of which 15 % are randomly masked so that the neural network can no longer know the original token. Further, the 500 tokens are divided into two sentences of length 250 tokens, of which the second sentence is randomly replaced 50 % of the time by a sentence from a different sequence. The task of predicting whether the sentence has been replaced or not is called Next Sentence Prediction (NSP), the task of predicting the masked tokens is called Masked Language Model (MLM). Internally, in our BERT model, each token is represented by a vector of length 250.

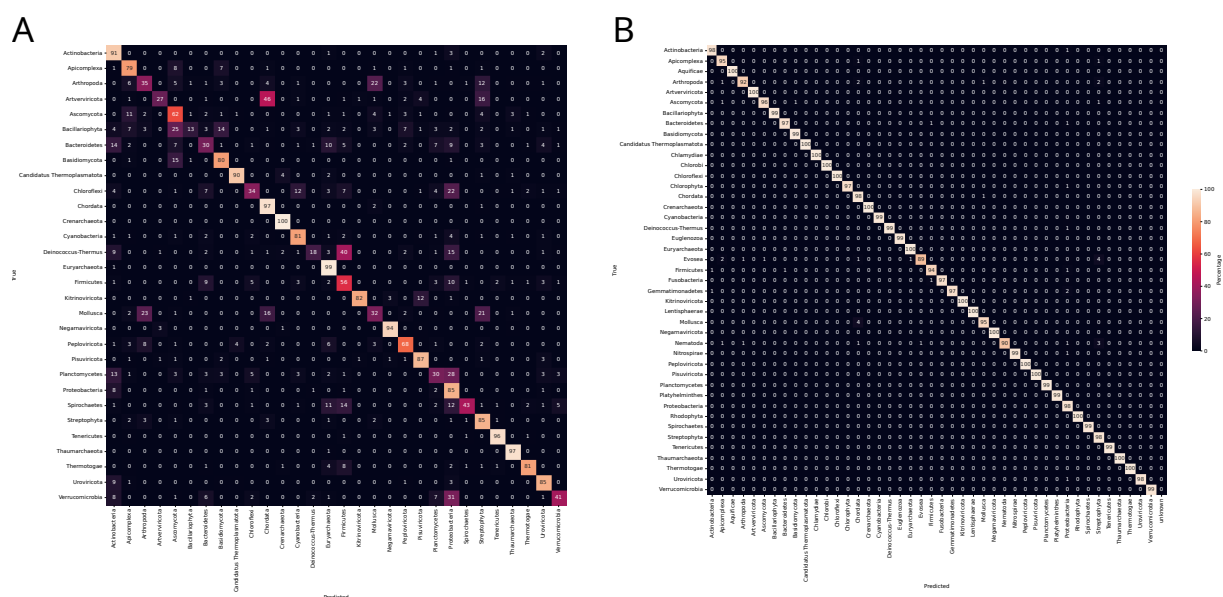


Figure S13: MMseqs2 + BERTax-predicted vs. true classification ("confusion matrices") for the rank phylum of the **A** distantly related and **B** final dataset. The values of each row are percentages of the true number of samples from the respective taxonomic class.

Attention weight statistics

Three types of DNA sequences were examined for patterns in BERTax attention weights. To obtain **gene-coding sequences**, the NCBI nucleotide database was searched for coding sequences (CDS) ranging from 1,000 to 1,500 nt in length. 100 hits per superkingdom were retrieved to obtain 400 sequences in total. Whole **genomic regions** of four species were retrieved: *Homo sapiens* (eukaryotic, accession NC_060939.1 (region 2,013,821–6,049,313)), *Escherichia coli* (bacterial, accession U00096.3), *Pyrococcus furiosus* (archaeal, accession AE009950.1) and SARS-CoV-2 (viral, accession NC_045512.2). For *Homo sapiens* the region which contains the ribosomal genes was selected. Third, 1,000 **fragments** from the test-split of the “final” dataset were drawn at random.

For all three types of sequence sources, statistics on 3-mer frequencies and BERTax attention weight were collected, non-overlapping 3-mers were considered. Here, attention weights were average per sequence and finally per 3-mer, therefore the averaged attention weights are independent of 3-mer frequencies. **Genomic regions** were processed as overlapping 1,500 nt fragments with a step size of 500 nt. Attention weights for **genomic regions** were visualized in the UCSC genome browser [5, 4] as custom tracks. Genbank annotations of genomic regions were used to retrieve the positions of annotated features alongside the feature type.

Various measures expressing the importance of individual tokens for a transformer model’s classification decision exist [1, 2, 6]. Here the method described in [3] is used, the corresponding values per token are referred to as “attention weights” here.

Solely visual interpretation of attention weights of **genomic regions**, as performed in the genome browser using custom tracks, does not show visible correlation to regions of interest such as splice sites, untranslated regions or regions with particularly high or low evolutionary conservation. Furthermore, the same genomic region shows substantially different attention patterns depending on the shift (for fragments of size 1,500 nt with step size 500 nt there are 3 “shifts”). This difference in attention weights when comparing the same region in different contexts is likely owed to the fundamentally local nature of even sophisticated measures for attention weights.

However, relating average attention weights to annotated feature types reveals certain – albeit minor – patterns (see Fig. S14). In all considered genomic regions containing ribosomal RNA (rRNA) genes (thus excluding the viral genomic regions), attention weights for rRNA sites are slightly higher on average. As is commonly utilized in metagenomics, rRNA sequences can be highly informative for taxonomic classification. Therefore a likely conclusion is that BERTax also pays special attention to these regions if they are part of the query sequence. For other feature types, no clear observations can be made; it has to be emphasized again that due to the sequence-local nature of attention-weights these statistics have to be interpreted with extreme care.

On a more basic level, investigating the distribution of average attention weights for the different datasets (see Fig. S15) shows clear distinctions: BERTax seems to be able to differentiate gene-coding and arbitrary genomic sequences and accordingly decide to which 3-mers to pay special attention to. The distribution for **fragments** seems to suggest that there is a more pronounced distinction between 3-mers deemed important and those with little influence compared to **gene-coding fragments**. A possible conclusion could be that BERTax has learned that certain individual 3-mers can be highly important to differentiate taxonomy in regulatory regions for instance – while others can be practically ignored.

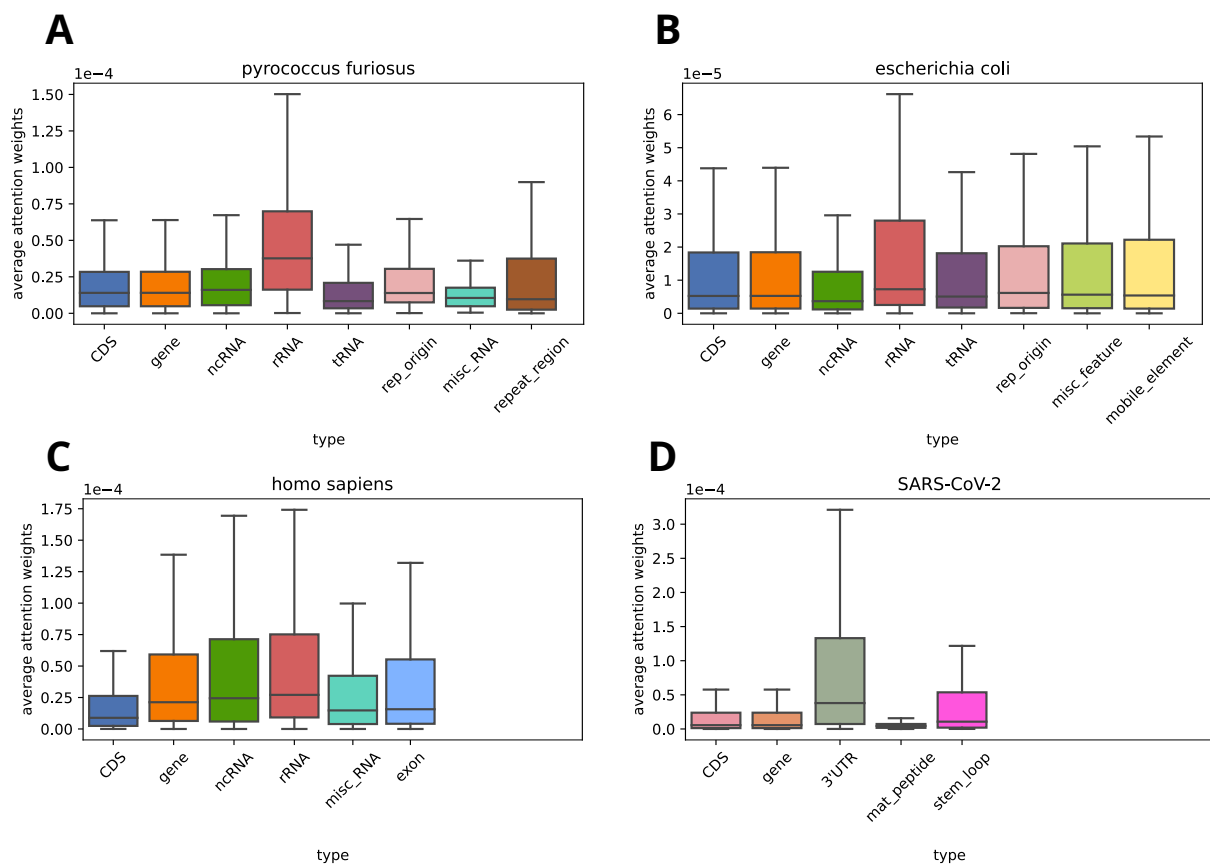


Figure S14: Attention weights grouped by annotated feature for species **A** *Pyrococcus furiosus*, **B** *Escherichia coli*, **C** *Homo sapiens* and **D** SARS-CoV-2. Outliers are not shown.

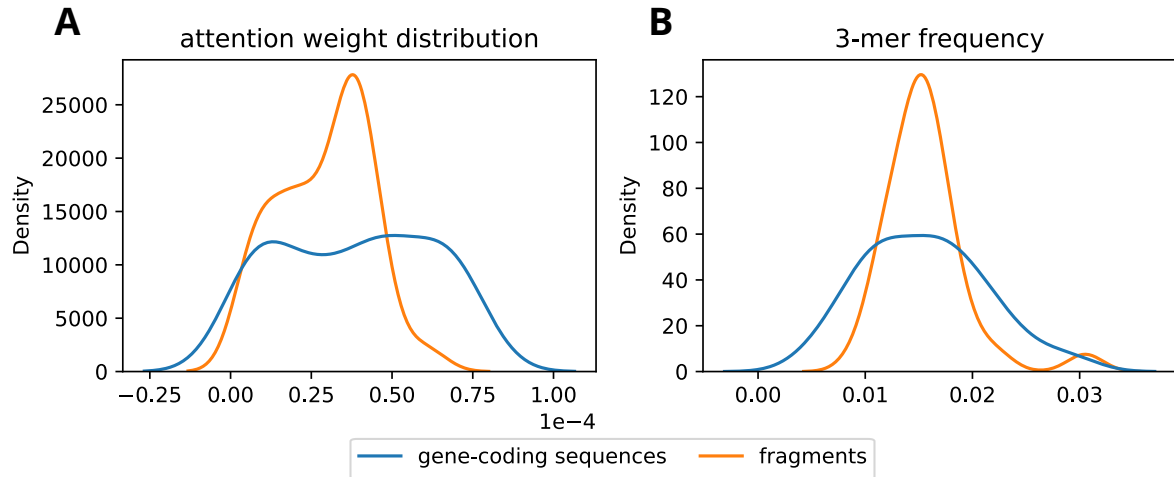


Figure S15: **A** Average attention weight distribution of 3-mers (kernel density estimation) comparing **gene-coding sequences** and **fragments** **B** 3-mer frequency distribution (kernel density estimation) for **gene-coding sequences** and **fragments**.

Table S1: Comparing different BERTax architectures (flat, nested, all-in-one) for taxonomic classification. The closely related selection test set contains samples from the same genera as the training set. The distantly related selection test set contains samples from different genera as the training set. Each sample has been classified by superkingdom and phylum. The all-in-one architecture has the highest accuracy in all cases.

architecture	closely related dataset		distantly related dataset	
	superkingdom	phylum	superkingdom	phylum
flat	94.64	83.16	87.66	55.77
nested	90.31	73.12	86.95	50.12
all-in-one	94.78	85.55	88.95	60.10
random	25.00	3.33	25.00	3.33

Table S2: Comparison of the accuracy (acc.) and the proportion (prop.) of the predicted samples per method on the closely and the distantly related dataset.

Tool	closely related dataset				distantly related dataset			
	superkingdom		phylum		superkingdom		phylum	
	acc.	prop.	acc.	prop.	acc.	prop.	acc.	prop.
MMseqs2	99.63	87.45	97.30	87.45	90.73	52.78	75.04	52.78
MMseqs2_taxonomy*	98.06	92.69	92.62	92.69	80.50	71.37	59.19	71.37
minimap2	99.95	75.59	99.55	75.59	94.42	19.88	77.12	19.88
Kraken2	99.82	76.11	98.82	75.95	90.71	21.56	68.26	21.08
sourmash	99.96	6.91	99.81	6.90	99.56	0.17	91.67	0.16
DeepMicrobes	96.68	100	87.72	100	68.39	100	41.95	100
BERTax	94.78	100	85.55	100	88.95	100	60.10	100
MMseqs2 + DeepMicrobes	99.10	100	94.03	100	78.20	100	56.28	100
MMseqs2 + BERTax	99.10	100	95.21	100	89.42	100	66.60	100
random	25.00	100	25.00	100	25.00	100	25.00	100

Table S3: Average precision (AveP) of the precision-recall curve for the final **BERTax** model compared to the model trained on the closely related dataset.

training dataset	Eukaryota	Bacteria	Archaea	Viruses
closely related dataset	96.07	96.41	95.40	94.73
final dataset	99.30	98.99	98.64	97.53

Table S4: Number of epochs trained per evaluated **BERTax** architecture before early stopping, as the nested architecture comprises of multiple trained models the numbers can vary.

architecture	flat	nested	all-in-one
epochs	16	11-13	15

Table S5: Comparison of accuracy (acc.) when removing potentially homologous sequences from the distantly related dataset. To account for how sequence similarity between the test- and train-partition of the distantly related dataset affects performance, a BLAST search has been performed with the test sequences as query and the train sequences as database. All query sequences with at least one hit have been removed (filtered); default parameters have been used for the BLAST search (**megablast**). The accuracy for the sequences removed is also included (left-out). Note that **MMseqs2 taxonomy*** is only able to make predictions for 71.37% of all sequences, **BERTax** and **DeepMicrobes** for all sequences (see S2).

Tool	superkingdom			phylum		
	acc. unfiltered	acc. filtered	acc. left-out	acc. unfiltered	acc. filtered	acc. left-out
BERTax	88.95	88.30	93.48	60.10	59.15	63.52
DeepMicrobes	68.39	63.24	81.13	41.95	39.89	47.80
MMseqs2 taxonomy*	80.50	74.51	95.83	59.19	53.31	80.36

Table S6: Comparison of accuracy (acc.) for **BERTax** for each superkingdom for potentially homologous sequences from the distantly related dataset.

BERTax	Archaea	Bacteria	Eukaryota	Viruses
acc. potentially homologous	96.41	91.89	95.07	90.55
acc. non-homologous	97.50	82.54	87.72	85.45
difference	-1.09	9.35	7.35	5.09
number of phyla	4	12	8	6

Table S7: Ablation studies for the number of attention heads and transformer blocks in the **BERTax** architecture. Performance, measured in macro average precision (AveP) is evaluated when varying the number of attention heads from “5” to “2” and “3”, as well as varying the number of transformer blocks from “12” to “6”. Additional to the final dataset, performance on the distantly related dataset is evaluated, too to investigate generalizability. Note that varying the number of attention heads also leads to a variation in the embedding dimension to keep the ratio of 50 per attention head constant.

Variation	dataset	distantly related		final		Genus
		Supk.	Phyl.	Supk.	Phyl.	
BERTax (attention heads = 5, transformer blocks = 12)		90.06	54.10	98.62	95.10	66.92
attention heads = 2 , transformer blocks = 12		83.61	48.34	96.17	86.30	48.37
attention heads = 3 , transformer blocks = 12		87.44	52.22	98.35	94.31	64.64
attention heads = 5, transformer blocks = 6		89.01	55.40	98.78	95.90	69.31

References

- [1] Samira Abnar and Willem Zuidema. “Quantifying attention flow in transformers”. In: *arXiv preprint arXiv:2005.00928* (2020).
- [2] Alexander Binder et al. “Layer-wise relevance propagation for neural networks with local renormalization layers”. In: *International Conference on Artificial Neural Networks*. Springer. 2016, pp. 63–71.
- [3] Hila Chefer, Shir Gur, and Lior Wolf. “Transformer interpretability beyond attention visualization”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 782–791.
- [4] Jason D Fernandes et al. “The UCSC SARS-CoV-2 genome browser”. In: *Nature Genetics* 52.10 (2020), pp. 991–998.
- [5] W James Kent et al. “The human genome browser at UCSC”. In: *Genome research* 12.6 (2002), pp. 996–1006.
- [6] Ramprasaath R Selvaraju et al. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.