

Peer Review

Review of: "METAGENE-1: Metagenomic Foundation Model for Pandemic Monitoring"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The authors pretrain METAGENE-1, a 7-billion-parameter decoder-only transformer (Llama-2-7B architecture, Table 1), on DNA and RNA reads from deep metagenomic sequencing of human wastewater. The wastewater was collected over six months by the Nucleic Acid Observatory in cities in California and Missouri (Section 3.1). The corpus is described as "over 1.5 trillion base pairs," with most reads 100–300 base pairs long. A BPE tokenizer with 1,024 tokens is trained on a 2-billion-base-pair sample (Section 3.2), and the context is 512 tokens. Training ran on 4 nodes of 8 H100 GPUs at 40% MFU (Section 4.1). A second pretraining stage, about 9% of tokens, mixes in multi-species genomes at a 1:8 ratio (Section 4.4).

The paper claims state-of-the-art results on a new pathogen-detection benchmark (Table 2, MCC average 92.96 against 87.92 for DNABERT-2) and on a new embedding benchmark, Gene-MTEB (Table 3, global average 0.590 against 0.545). It also reports the top score on 13 of 28 GUE tasks (Table 4). Finally, model loss separates metagenomic reads from human, mouse, and random sequences (Section 5.5, Table 5). Weights and code are released, and Section 6 argues that the current model poses minimal misuse risk.

2. Whether the conclusion holds on the data

What I value most is the new data domain. Wastewater metagenomics is a setting where about 41% of reads are unclassified by Kraken 2 (Section 3.1, Figure 3), which makes it a worthwhile domain to build for. I also appreciate the open weights, the frank account of infrastructure failures (three node failures, one GPU failure, one network failure, and one disk failure, Section 4.1), and the Safety section, which states the reasoning behind the open release (Section 6).

On pathogen detection (Section 5.2), METAGENE-1 reaches an average MCC of 92.96 against 87.92 for DNABERT-2 (Table 2). Each dataset has 1,600 training and 2,000 test samples. The labels come from Kraken 2 k-mer hits plus alignment to human-infecting virus genomes in GenBank. For this reason, I would add a non-learned baseline, such as Kraken 2 itself or alignment. Without it, the reader cannot see what the model adds over the tools that made the labels.

I would also make the comparison more even. DNABERT variants are fully fine-tuned, while NT variants and METAGENE-1 use LoRA, and DNABERT-S uses a halved learning rate (Section 5.2, Table 6). It would help to align these settings or to explain why they differ. Finally, no repeated runs, seeds, or variance are reported for Tables 2-4. A gap such as 90.91 for METAGENE-1 against 86.90 for DNABERT-2 on PATHOGEN-DETECT-2 (Table 2) therefore has no error bar. A few seeds with standard deviations would let readers judge how large that gap really is.

In Section 5.3, the text says METAGENE-1 shows "strong embedding performance across the board," and I think Table 3 supports a softer sentence. The model is behind on HUMAN-VIRUS-4 (NT-2.5b-1000g is better), on HVR-P2P (NT-2.5b-1000g again), and on HMPR-P2P (DNABERT-S is better). The HMPR average rests mostly on HMPR-S2S-TINY, where METAGENE-1 is far ahead of the next model. I recomputed the other three HMPR tasks without it, and the mean is 0.350 for METAGENE-1 against 0.407 for NT-2.5b-1000g. I would add this view next to the average and change "across the board" to something more specific.

On GUE (Section 5.4, Table 4), the baseline numbers are adapted from the DNABERT-2 paper [4]. METAGENE-1 gets a grid search over learning rates and two sets of LoRA modules, selected on validation (Table 7). The wins on GUE are real, and I am glad they are reported. It would help to show that the baselines had a similar tuning budget, or to say plainly that they did not.

For anomaly detection (Section 5.5), the in-distribution group is sampled from "our metagenomics pretraining data," so the model has already seen these sequences. The threshold is read from the validation curve in Figure 5. I would explain that the low loss on seen reads in Table 5 partly measures memorization. A held-out in-distribution set would make this result much easier to trust.

3. Reproducibility

The weights (huggingface.co/metagene-ai) and the code (github.com/metagene-ai) are released, as stated in the abstract and in Section 7. The Gene-MTEB datasets are said to be on the project HuggingFace page (Section 5.3). Thank you for making these public.

The hyperparameters are given in good detail. Table 1 lists the architecture, batch size 30,720, learning rate $6e-4$, cosine decay, and 2,000 warmup steps. Tables 6 and 7 cover fine-tuning (LoRA rank 8, alpha 16, 10 epochs, A100 80GB). The Kraken 2 database (2024-06 Standard) and the GenBank release (2024-06) used for the labels are named in footnotes 5 and 6, which will help anyone reproducing the labels.

I searched the text for a few items and did not find them. I would add whether the wastewater pretraining reads and the four pathogen-detection datasets are released, the random seeds, the number of runs, and the total GPU hours or wall-clock time for pretraining. Section 4.1 gives the hardware and MFU only.

One small point on consistency. The data size is "over 1.5 trillion base pairs" in the abstract and Section 3.1, but "about 370 billion tokens (about 1.69 trillion base pairs)" in Section 3.2. It would help to add a sentence on what the 1.69 figure includes.

4. What specifically to fix

1. In Section 5.2 / Table 2, I would add a non-learned baseline that uses the same tools as the labels, either Kraken 2 classification or alignment to the GenBank viral genomes, run on the same 2,000 test reads. That way, the MCC gain of METAGENE-1 is measured against the labeling pipeline itself, and readers can see what the model adds beyond it.
2. For Tables 2-4, it would help to report the mean and spread over at least three seeds. Also, Table 6 lists full fine-tuning for DNABERT but LoRA for NT and METAGENE-1, so Section 5.2 currently compares training regimes as much as models. I would either fine-tune DNABERT with LoRA as well or fine-tune METAGENE-1 fully on one task.
3. In Section 5.3, I would replace "across the board" with a precise statement. Table 3 shows METAGENE-1 behind on HUMAN-VIRUS-4 (0.697 vs 0.755), HVR-P2P (0.466 vs 0.650), and HMPR-P2P (0.479 vs 0.580). A short discussion would also be welcome on why HMPR-S2S-TINY reaches 0.855 when no other model exceeds 0.391.
4. A small point of naming. The task list in Section 5.3 writes "MHPD-single," while Table 3 uses "HMPD-SINGLE." Please align the two.
5. For Section 5.4 / Table 4, I would state clearly that the baseline numbers come from [4] with their own tuning. Then either rerun DNABERT-2 under the same grid as Table 7 or give METAGENE-1 a single fixed configuration, so the comparison is on equal footing.
6. In Section 5.5, I would take the in-distribution sample from held-out deliveries, as Section 5.2 does, rather than from the pretraining data. Since BPE was trained on metagenomic reads, it would also

help to report whether the separation survives when token counts are matched. Table 5 gives a mean tokenized length of 24.91 for metagenomics against 27.16–27.29 for the other groups. Finally, the column label "Loss (Std. Err)" with 1.24 (1.31) looks like a standard deviation rather than a standard error, so please check the label.

7. In Section 6, the claim that the model is "tailored for detection, not design" would be stronger with a small generation test. For example, one could check whether sampled continuations of viral reads match known pathogen sequences, rather than relying on context length and read length alone.

5. Recommendation

Major revisions. The model, the data domain, and the open release are a good contribution to pathogen monitoring, and I would like to see them stand on firmer evaluation. I would add a non-learned baseline for the pathogen task, report seeds, and match the fine-tuning budgets across models. It would also help to use held-out data for the anomaly test and to word the Gene-MTEB claim more precisely. These are mostly changes to the evaluation, and none of them needs a new pretraining run, so I hope they are quite feasible for the authors.

Declarations

Potential competing interests: No potential competing interests to declare.