nature portfolio

# Peer Review File

# megaDNA: a long-context language model for deciphering and generating bacteriophage genomes

Corresponding Author: Dr BIN SHAO

**This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.**

Version 0:

Reviewer comments:

Reviewer #3

(Remarks to the Author)
The manuscript by Bin Shao presents a very interesting proof-of-concept for a generative model trained on and designed for bacteriophage genomes. The idea of leveraging the latest developments in machine learning to help sequence analysis and, eventually, offer tools capable of de novo sequence generation is certainly timely, and the model and analysis presented in this manuscript provide new potential avenues towards these goals. The author seemed to make the choice to present the results of many analyses, most relatively preliminary and/or without in-depth characterization. While this is efficient to showcase all the potential applications of the model, I found that this made the overall conclusion of the manuscript somewhat difficult to grasp, as in the end it is not clear to me what would be the real strengths and limitations of this model when faced with “real” biological questions or problems.
My concerns about the analyses presented are two-fold. First, I have several concerns regarding a potential over-fitting (e.g. the taxonomic classification analysis), and on the interpretation of the geNomad analysis (for instance it seems like the author ignored the geNomad score for all sequences not predicted as viruses, even though these scores are provided by geNomad). Second, I have questions about the applicability of this tool, i.e. in some cases there seems to be a signal (for instance the analysis of mutation impact presented), but it’s unclear if the signal is strong enough to be valuable for biological analysis.

In my opinion, to be fully valuable to the biological research community, this manuscript needs (i) some additional and/or corrected analyses (detailed below), and (ii) more discussion about the state of these approaches/models, i.e. what can it be useful for now (which type of phage, which type of analysis), and what is needed to make it more broadly applicable

Major comments:

l. 53-54: “NCBI genebank, the Metagenomic Gut Virus (MGV) catalogue, and the Gut Phage Database (GPD)”: This training set will be strongly biased towards a relatively small part of the global phage diversity, which will have implications for how generalizable the model will be. I believe this should be acknowledged and discussed in the text. For instance l. 58: “captures the structural patterns of bacteriophage genomes” is an over-statement in my opinion, as “bacteriophage genomes” is a much larger and much more diverse sequence space than the one used for training here

l. 60: “predict essential genes in the lambda phage genome”: I believe that this results should be accompanied by an estimation of how many genomes of phages identical, near-identical, or similar to Lambda are present in the training set. This would help a reader understand the constraint of this method, i.e. which type of training is needed for the model to achieve this performance (a single reference ? A few references ? Hundreds of similar references ?).

l. 72: “Our model's prediction performance closely matched the state-of-the-art model DeepSequence […] our model successfully predicted the impact of SNPs across the T7 bacteriophage genome”: I agree with the author that MegaDNA performances seem on par with the performances of DeepSequence, however I struggle somewhat to understand how a Spearman correlation coefficient of 0.5 can lead to a useful predictor for biologists. Said otherwise, I agree that there is some signal here, but I don’t see in the current figures a way to understand whether this signal would be sufficient to truly predict the impact of individual SNPs on gene fitness. I would encourage the author to provide, as Supplementary Figure, the x-y plots of the observed vs predicted fitness for each correlation bar chart (panels f, g, and i in Fig. 1), and discuss these x-y

plots in the text.

l. 82: “The embeddings from […] demonstrate the broad applicability of our model.”: As currently described, I am worried about potential over-fitting in this linear regressor. The author mentions “5-fold Stratified K-Fold cross-validation test”, however there is no mention of controlling for the redundancies between training and testing set in this cross-validation. The methods l. 303 mention “maintaining the same proportion of samples for each class as in the complete dataset”, but to my understanding this would be the same proportion of viruses, bacteria, and archaea, without checking the similarities between viral, bacterial, and archaeal sequences across training and testing sets. Without this important control, it is very common when training on genome databases that near-identical genomes are present in both training and testing folds. For taxonomic classification, this should be evaluated by leaving out entire taxa from the training set, and testing on these classes, or by weighing the fold split based on pairwise genome similarity.

l. 92: “Among all these sequences, 607 have a virus score larger than zero.”: This does not seem to correspond to a geNomad output (since by default the minimum virus score considered is 0.7). It also seems inconsistent with Fig. 2c, which suggests that all generated sequences had a virus score greater than 0.7. Please double-check and correct as needed. In particular, please make sure to report the virus score of all sequences, and not just the ones geNomad predicted as viral (see geNomad output “aggregated_classification.tsv” in the output folder “aggregated_classification”).

l. 96: “The median virus score of these generated sequences is 0.84 and the maximum score is 0.97, comparable to the virus scores of natural bacteriophage genomes which range from 0.70 to 0.98 (Fig. 1c).”: I am not sure I understand how the median virus score can be 0.84 when only 607 out of 1,024 sequences had a score larger than zero ? From Fig. 1c, it also seems as if the median score for nature phage genome is around 0.99, so I feel like calling the score distribution of generated sequences “comparable to the virus score of natural bacteriophage genomes” is quite a stretch. In my opinion, the distribution of scores for these generated sequences is clearly lower than the one of natural phage genomes (especially when considering that the distribution of score between 0.7 and 1.0 is a parameter of geNomad, not a feature of the sequences themselves).

L. 100-102: I have concerns about this analysis, which I believe are more concerns with the results from DeepHost than with MegaDNA. For instance, the fourth more common predicted host is Cellulophaga baltica, a marine bacteria, which is suspicious for sequences generated based on human gut phages. Given the uncertainty around these predictions, the lack of demonstrated reliability of DeepHost for novel sequences, and the lack of benchmarking of this tool for its ability to discriminate bacteriophages from viruses of archaea, viruses of eukaryotes, and non-viral sequences, I would suggest removing this analysis or add these caveats to the text. (Said otherwise, I believe that bacterial sequences used as input to DeepHost would yield similar results, so that in the end it is not really a good test for the generation of phage sequences per se).

l. 116: “functional regulatory sequences”: I don’t believe this analysis demonstrate that these are “functional”, and would suggest rephrasing as “potential regulatory sequences”

l. 123: “suggesting that these generated proteins are more likely to adopt a stable conformation.”: I disagree with this claim, higher pLDDT score in my opinion could very well reflect that these sequences are similar to the ones used to train ESM-Fold (which would make sense considering the overlap in training set between megaDNA and ESM-Fold).

l. 125: To my knowledge, deepFRI is not a state-of-the-art tool for phage genome annotation. Instead, I would recommend the author use a standard pipeline such as pharokka, and a phage-specific database such as PhrogDB.

l. 128: One analysis I think would be valuable here would be an estimation of how diverse the set of generated sequences is. This could be achieved e.g. based on a clustering of these generated genomes with vContact2, but even if using another tool, the idea would be to illustrate for the reader how often the model generates closely related sequences (or not).

l. 135: “with further scaling up and fine-tuning, we envision that generative genomic models have the potential to enable de novo design of the whole genome”: The first part (“with further scaling up and fine-tuning”) is quite an under-statement in my opinion. I believe the author nicely demonstrated that this kind of model may have the potential to learn some of the constraints underlying phage genomes, however from the limited results presented here, it seems like these models are still very far away from the de novo design of whole functional genomes, and rather represent a first step towards this goal.

l. 231: This section should ideally include a paragraph describing how the model was built. For instance, the manuscript mentions l. 38: “a long-context language model”, however it was not clear to me what “long-context” really meant, as the manuscript did not illustrate the size of the context window. A paragraph describing the structure of the model and its main characteristics would be helpful in my opinion.

Minor comments:

l. 53: “genebank,” should be “genbank,”

l. 81-82: “We collected unannotated sequences from bacteriophage, bacteria, and archaea.”: Please briefly clarify the source of these sequences here.

l. 99: “Caudoviricetes”, and all viral taxa names, should be italicized (also “Caudovirales.” l. 234).

l. 230 and throughout: All mentions of software and database should also include version number.

l. 232: The different sources of training sequences should come with a citation and, ideally, a specific list of accession numbers of the exact sequences used for training (so that others could try to reproduce/reconstruct the training set).

l. 237: “whose predicted host is not a unicellular organism”: It is not entirely clear to me how this was derived from the taxonomy. Please provide a list of taxa that were excluded and/or included in the final dataset.

l. 252: “gens” should be “genes”

l. 279: “that was calculated” should be “was calculated”

Fig. 2d: “generated sequences” should be “generated sequences predicted as viral”

Reviewer #4

(Remarks to the Author)
The manuscript by Bin Shao presents a very interesting proof-of-concept for a generative model trained on and designed for bacteriophage genomes. The idea of leveraging the latest developments in machine learning to help sequence analysis and, eventually, offer tools capable of de novo sequence generation is certainly timely, and the model and analysis presented in this manuscript provide new potential avenues towards these goals. The author seemed to make the choice to present the results of many analyses, most relatively preliminary and/or without in-depth characterization. While this is efficient to showcase all the potential applications of the model, I found that this made the overall conclusion of the manuscript somewhat difficult to grasp, as in the end it is not clear to me what would be the real strengths and limitations of this model when faced with “real” biological questions or problems.

My concerns about the analyses presented are two-fold. First, I have several concerns regarding a potential over-fitting (e.g. the taxonomic classification analysis), and on the interpretation of the geNomad analysis (for instance it seems like the author ignored the geNomad score for all sequences not predicted as viruses, even though these scores are provided by geNomad). Second, I have questions about the applicability of this tool, i.e. in some cases there seems to be a signal (for instance the analysis of mutation impact presented), but it’s unclear if the signal is strong enough to be valuable for biological analysis.

In my opinion, to be fully valuable to the biological research community, this manuscript needs (i) some additional and/or corrected analyses (detailed below), and (ii) more discussion about the state of these approaches/models, i.e. what can it be useful for now (which type of phage, which type of analysis), and what is needed to make it more broadly applicable

Major comments:

l. 53-54: “NCBI genebank, the Metagenomic Gut Virus (MGV) catalogue, and the Gut Phage Database (GPD)”: This training set will be strongly biased towards a relatively small part of the global phage diversity, which will have implications for how generalizable the model will be. I believe this should be acknowledged and discussed in the text. For instance l. 58: “captures the structural patterns of bacteriophage genomes” is an over-statement in my opinion, as “bacteriophage genomes” is a much larger and much more diverse sequence space than the one used for training here

l. 60: “predict essential genes in the lambda phage genome”: I believe that this results should be accompanied by an estimation of how many genomes of phages identical, near-identical, or similar to Lambda are present in the training set. This would help a reader understand the constraint of this method, i.e. which type of training is needed for the model to achieve this performance (a single reference ? A few references ? Hundreds of similar references ?).

l. 72: “Our model's prediction performance closely matched the state-of-the-art model DeepSequence […] our model successfully predicted the impact of SNPs across the T7 bacteriophage genome”: I agree with the author that MegaDNA performances seem on par with the performances of DeepSequence, however I struggle somewhat to understand how a Spearman correlation coefficient of 0.5 can lead to a useful predictor for biologists. Said otherwise, I agree that there is some signal here, but I don’t see in the current figures a way to understand whether this signal would be sufficient to truly predict the impact of individual SNPs on gene fitness. I would encourage the author to provide, as Supplementary Figure, the x-y plots of the observed vs predicted fitness for each correlation bar chart (panels f, g, and i in Fig. 1), and discuss these x-y plots in the text.

l. 82: “The embeddings from […] demonstrate the broad applicability of our model.”: As currently described, I am worried about potential over-fitting in this linear regressor. The author mentions “5-fold Stratified K-Fold cross-validation test”, however there is no mention of controlling for the redundancies between training and testing set in this cross-validation. The methods l. 303 mention “maintaining the same proportion of samples for each class as in the complete dataset”, but to my understanding this would be the same proportion of viruses, bacteria, and archaea, without checking the similarities

between viral, bacterial, and archaeal sequences across training and testing sets. Without this important control, it is very common when training on genome databases that near-identical genomes are present in both training and testing folds. For taxonomic classification, this should be evaluated by leaving out entire taxa from the training set, and testing on these classes, or by weighing the fold split based on pairwise genome similarity.

l. 92: “Among all these sequences, 607 have a virus score larger than zero.”: This does not seem to correspond to a geNomad output (since by default the minimum virus score considered is 0.7). It also seems inconsistent with Fig. 2c, which suggests that all generated sequences had a virus score greater than 0.7. Please double-check and correct as needed. In particular, please make sure to report the virus score of all sequences, and not just the ones geNomad predicted as viral (see geNomad output “aggregated_classification.tsv” in the output folder “aggregated_classification”).

l. 96: “The median virus score of these generated sequences is 0.84 and the maximum score is 0.97, comparable to the virus scores of natural bacteriophage genomes which range from 0.70 to 0.98 (Fig. 1c).”: I am not sure I understand how the median virus score can be 0.84 when only 607 out of 1,024 sequences had a score larger than zero ? From Fig. 1c, it also seems as if the median score for nature phage genome is around 0.99, so I feel like calling the score distribution of generated sequences “comparable to the virus score of natural bacteriophage genomes” is quite a stretch. In my opinion, the distribution of scores for these generated sequences is clearly lower than the one of natural phage genomes (especially when considering that the distribution of score between 0.7 and 1.0 is a parameter of geNomad, not a feature of the sequences themselves).

L. 100-102: I have concerns about this analysis, which I believe are more concerns with the results from DeepHost than with MegaDNA. For instance, the fourth more common predicted host is Cellulophaga baltica, a marine bacteria, which is suspicious for sequences generated based on human gut phages. Given the uncertainty around these predictions, the lack of demonstrated reliability of DeepHost for novel sequences, and the lack of benchmarking of this tool for its ability to discriminate bacteriophages from viruses of archaea, viruses of eukaryotes, and non-viral sequences, I would suggest removing this analysis or add these caveats to the text. (Said otherwise, I believe that bacterial sequences used as input to DeepHost would yield similar results, so that in the end it is not really a good test for the generation of phage sequences per se).

l. 116: “functional regulatory sequences”: I don’t believe this analysis demonstrate that these are “functional”, and would suggest rephrasing as “potential regulatory sequences”

l. 123: “suggesting that these generated proteins are more likely to adopt a stable conformation.”: I disagree with this claim, higher pLDDT score in my opinion could very well reflect that these sequences are similar to the ones used to train ESM-Fold (which would make sense considering the overlap in training set between megaDNA and ESM-Fold).

l. 125: To my knowledge, deepFRI is not a state-of-the-art tool for phage genome annotation. Instead, I would recommend the author use a standard pipeline such as pharokka, and a phage-specific database such as PhrogDB.

l. 128: One analysis I think would be valuable here would be an estimation of how diverse the set of generated sequences is. This could be achieved e.g. based on a clustering of these generated genomes with vContact2, but even if using another tool, the idea would be to illustrate for the reader how often the model generates closely related sequences (or not).

l. 135: “with further scaling up and fine-tuning, we envision that generative genomic models have the potential to enable de novo design of the whole genome”: The first part (“with further scaling up and fine-tuning”) is quite an under-statement in my opinion. I believe the author nicely demonstrated that this kind of model may have the potential to learn some of the constraints underlying phage genomes, however from the limited results presented here, it seems like these models are still very far away from the de novo design of whole functional genomes, and rather represent a first step towards this goal.

l. 231: This section should ideally include a paragraph describing how the model was built. For instance, the manuscript mentions l. 38: “a long-context language model”, however it was not clear to me what “long-context” really meant, as the manuscript did not illustrate the size of the context window. A paragraph describing the structure of the model and its main characteristics would be helpful in my opinion.

Minor comments:

l. 53: “genebank,” should be “genbank,”

l. 81-82: “We collected unannotated sequences from bacteriophage, bacteria, and archaea.”: Please briefly clarify the source of these sequences here.

l. 99: “Caudoviricetes”, and all viral taxa names, should be italicized (also “Caudovirales.” l. 234).

l. 230 and throughout: All mentions of software and database should also include version number.

l. 232: The different sources of training sequences should come with a citation and, ideally, a specific list of accession numbers of the exact sequences used for training (so that others could try to reproduce/reconstruct the training set).

l. 237: “whose predicted host is not a unicellular organism”: It is not entirely clear to me how this was derived from the taxonomy. Please provide a list of taxa that were excluded and/or included in the final dataset.

l. 252: “gens” should be “genes”

l. 279: “that was calculated” should be “was calculated”

Fig. 2d: “generated sequences” should be “generated sequences predicted as viral”

Reviewer #5

(Remarks to the Author)
In this paper, the author discusses megaDNA, a language model trained on bacteriophage genome to yield predictions (gene essentiality, mutations, etc.) as well as generating new phage genomes. The model was trained on bacteriophage genome data with byte-level tokenisation. The manuscript is of high quality, as it is well written with little to no errors and beautiful figures. Even the supplementary figures are of high quality, which is a rare sight.

I will start out by saying I am by no means an expert when it comes to large language models, so my review should not be read as such. That said, I have a basic understanding on the matter. I have therefore tried to mostly review the interpretation of the results, which I found somewhat difficult. I believe the latter was not due to my lack of knowledge in the field of LMs, but due to the brevity of the manuscript (which I understand may be a requirement), but left me unsatisfied with what the results of the tool actually mean.

The zero-shot predictions of gene essentiality intrigued me, although I am left wondering whether model loss indeed is a good predictor of gene essentiality. That said, figure 1c-d seems to indeed indicate that this assumption is fair.

The predictions on gene functionality is the first bit I am scratching my head about. This is not necessarily because of my scepticism, but after reading the current manuscript it is not clear what exactly the related figures (e.g. 1f, 1g) show. I have read the method a couple of times to confirm that I wasn't missing something, but in the end I have to simply conclude it is unclear. Perhaps this part is very straightforward for people working in the LLM field, but with the broad audience of NatCom that is simply not good enough. To some extent, I have the same concerns about the translational and taxonomic predictions, although I think I understand those a bit better. Please elaborate better on what these data show.

Next, the de novo generation of sequences is discussed. While this is very exciting on the one hand, the ability to generate novel variations is a known property of LLMs. I am not saying this is not exciting, but it would be better to highlight specific examples rather than showing how it can generally produce phage-like genomes. With AI becoming an increasingly important tool in science, illustrating that "it works" is only moderately exciting. Note that I am fully aware of the efforts that must have gone into megaDNA, and that I say this with the best intentions of seeing the tool and its utilities grow.

I am more than happy to read a revised manuscript in the future.

Some minor issues:

> L63: "Used *as* a zero-shot predictor"?
> A lot of the subfigures from figure 2 are never referred to in the text. This makes it very hard to contextualise why these results are there in the first place.

Reviewer #6

(Remarks to the Author)
In their article, Dr. Shao presents their transformer-based language model for generation and evaluation of bacteriophage sequences. This article represents one of the first known applications of transformer models to nucleotide sequences. Through a variety of analyses, Dr. Shao demonstrates the impressive performance of megaDNA for a wide range of classification and generation tasks. The methods and results are mostly well-described and the vast majority of my comments are requests for additional detail or clarification.

Major Comments

- Data and Code availability: A list of the NCBI accession numbers and identifiers for the other sequences used to train the model is important for future replication and evaluation. For example, Ratcliff 2024 noted that the compositional features of megaDNA generated sequences are more similar to specific bacteriophage families than others - documentation of the included sequences could provide evidence as to whether this observed effect is due to family-level differences in the number of sequences present in the training set. These could be provided as a .txt or .tsv file within the megaDNA GitHub repository.

Minor Comments
- Lines 53, 232, 296, and 329: Correct “genebank" to “GenBank”.
- Line 77: The assessment of regulatory element detection in bacteria is a clever method of evaluating the bacteriophage DNA model. Please adjust “in bacteria” to “in bacterial genomes” to make it more clear these predictions are being done on

the bacteriophage hosts rather than the bacteriophages themselves.
- Line 83 and 301: Please confirm if the taxonomic classification used linear regression (as stated in 83) or logistic regression (as stated in 301).
- Lines 81-87: In the paragraph, please make it clear that the taxonomic classification is only to the domain level.
- Line 92: This sentence states that 607 sequences had virus scores greater than 0. However, figure 2D has taxonomic classifications for 610 sequences. Please clarify the discrepancy. Is it possible that some of the taxonomic classifications are for viruses with geNomad scores below 0.7?
- Line 252: Correct “gens” to “genes”
- Line 285: Please use “dimensional” instead of “dim”
- Line 305-306: ROC and AUROC are previously defined on lines 259-260.
- Line 316: Please clarify that the geNomad results were run with default parameters
- Lines 329-331: Please list the extent of representation for each of the three datasets in this section (it’s included in the supplementary figure 1)
- Figure 1C: This is a great analysis. In the figure caption, can you clarify the step size for the moving average?
- Figure 2: I have several comments for this figure:
- Within the figure captions, the use of external tools to derive values (e.g., geNomad for panel C and Promoter Calculator for panel E) should be clarified in all instances.
- Panels B, F, and H should have statistical tests run to test if the distributions are significantly different. For example, the Kolmogorov-Smirnov test can be used.
- Panels F: Other than for analytical ease, what is the justification for limiting the analysis in panel F to just sequence #87? Panel G is essentially replicated for all sequences in supplementary figure 10. I would expect that these trends would be consistent across generated sequences. An interesting analysis would be whether the difference in medians for random vs generated promoters correlates with the predicted virus scores.
- Supplementary Figure 5A: Please sort X values as ([-80,0],[0,80],[-160,0],[0,160],[-160,160]) so there is relative consistency in the input window length going from low to high.
- Supplementary Figures 6 and 7 can be combined into a single figure as they are the same analyses just using different layers. In lines 85-86, can the author please comment on their hypothesis for the variation in accuracy across model layers - particularly local vs middle?
- Supplementary Figure 8: The conclusions from this figure are hard to evaluate given the sequence length differences between generated sequences and natural sequences. Can you please add a second panel that depicts the ratio of predicted genes to sequence length for natural vs generated sequences? I am envisioning two box-plots or something similar.

Version 1:

Reviewer comments:

Reviewer #3

(Remarks to the Author)
I thank the authors for their thorough response and manuscript revision. All my previous comments were addressed, and I have no further concern at this time.

Reviewer #4

(Remarks to the Author)
I believe the authors have successfully addressed the major comments on the paper, resulting in a much improved manuscript that better demonstrates the robustness and potential of their method. I have a few minor suggestions:

- While a whole paragraph of the manuscript is dedicated to describing the individual genes encoded by the generated genomes, no analysis was performed to evaluate whether the entire set of genes within a given genome is coherent. This would demonstrate that the model has learned the components that make up a phage. A simple check would be to evaluate the fraction of generated genomes containing proteins annotated as capsid, terminase, or portal, since such proteins are expected in all phage genomes.

- Lines 182, 183: The alignments are so small that it is unclear whether they correspond to the same “underlying gene family.” Could you verify if the generated genes and their corresponding matches have the same annotations?

- Lines 184-189: Although little homology was found between generated and “natural” proteins at the amino acid level, it would be interesting to see if the generated proteins have reasonable structures. This could be easily evaluated by aligning the predicted structures of generated proteins to experimental structures in the PDB using Reseek/Foldseek. Note that phold doesn’t use structures directly, only a structure-informed representation of the protein. This would serve as additional confirmation that the model has learned biologically relevant features.

- Lines 417-419: There is a repetition of “genomes of bacteriophage, bacteria, and archaea.”

Reviewer #5

(Remarks to the Author)

Dear authors,

You have done an excellent job clarifying some of my initial confusions. While I unfortunately still feel like I understand only 70-80% of the manuscript, I now feel that is mostly my lack of knowledge in this particular field, and not reflective of the quality of the manuscript. Although this makes it hard for me to wholeheartedly recommend the manuscript, I am certainly in favour of publication as I assume that other (perhaps more suited reviewers) can cover for my lack of expertise.

I wish you all the best!

Reviewer #6

(Remarks to the Author)
I commend the author for the substantial effort in revising this manuscript. The changes made assuage all concerns that I had noted within my initial review.

We sincerely thank the three reviewers for the positive comments and highly constructive feedback on our work. Based on their insights, we have conducted more comprehensive analyses and thoroughly revised our manuscript. Please find below a point-to-point response to all the remarks raised by the reviewers.

**Reviewer #3** (Remarks to the Author):

The manuscript by Bin Shao presents a very interesting proof-of-concept for a generative model trained on and designed for bacteriophage genomes. The idea of leveraging the latest developments in machine learning to help sequence analysis and, eventually, offer tools capable of de novo sequence generation is certainly timely, and the model and analysis presented in this manuscript provide new potential avenues towards these goals. The author seemed to make the choice to present the results of many analyses, most relatively preliminary and/or without in-depth characterization. While this is efficient to showcase all the potential applications of the model, I found that this made the overall conclusion of the manuscript somewhat difficult to grasp, as in the end it is not clear to me what would be the real strengths and limitations of this model when faced with "real" biological questions or problems.
My concerns about the analyses presented are two-fold. First, I have several concerns regarding a potential over-fitting (e.g. the taxonomic classification analysis), and on the interpretation of the geNomad analysis (for instance it seems like the author ignored the geNomad score for all sequences not predicted as viruses, even though these scores are provided by geNomad). Second, I have questions about the applicability of this tool, i.e. in some cases there seems to be a signal (for instance the analysis of mutation impact presented), but it's unclear if the signal is strong enough to be valuable for biological analysis.

In my opinion, to be fully valuable to the biological research community, this manuscript needs (i) some additional and/or corrected analyses (detailed below), and (ii) more discussion about the state of these approaches/models, i.e. what can it be useful for now (which type of phage, which type of analysis), and what is needed to make it more broadly applicable.

We would like to thank the reviewer for the insightful suggestions.

Major comments:

l. 53-54: "NCBI genebank, the Metagenomic Gut Virus (MGV) catalogue, and the Gut Phage Database (GPD)": This training set will be strongly biased towards a relatively small part of the global phage diversity, which will have implications for how generalizable the model will be. I believe this should be acknowledged and discussed in the text. For instance l. 58: "captures the structural patterns of bacteriophage genomes" is an over-statement in my opinion, as "bacteriophage genomes" is a much larger and much more diverse sequence space than the one used for training here

We thank the reviewer for this comment. We have revised our text to discuss the limitation of our training dataset.

**Revised text** (Section "Discussion"):

> In addition, the training dataset only covers a limited subset of the global phage diversity, which may impact the model's ability to generalize to phage taxa not included in the training data.

**Revised text** (Section "megaDNA allows zero-shot prediction of gene essentiality"):

> We hypothesize that our pretrained language model captures the structural patterns of bacteriophage genomes in our training dataset.

l. 60: "predict essential genes in the lambda phage genome": I believe that this results should be accompanied by an estimation of how many genomes of phages identical, near-identical, or similar to Lambda are present in the training set. This would help a reader understand the constraint of this method, i.e. which type of training is needed for the model to achieve this performance (a single reference? A few references? Hundreds of similar references?).

We thank the reviewer for this valuable suggestion. To address this issue, we have added a new supplementary figure (**Fig. R1**) that illustrates the sequence similarity between the lambda phage genome and the training dataset. Through BLAST analysis, we found 847 significant hits for the lambda phage genome in the training sequences, including 50 sequences with a sequence identity above 0.4, and 8 sequences with an identity greater than 0.9. This result indicates that the model is exposed to a broad spectrum of related sequences, primarily dominated by low similarity sequences and complemented by a small number of sequences with high similarity.

Next, we estimated the proportion of phage genomes that have more representations in the training dataset than the lambda phage. For each genome sequence, we define the similarity score as the sum of the identity of all matched training sequences. Our analysis revealed that 34% of the randomly sampled phage genomes have a similarity score higher than the lambda phage. This result suggests that our model could be potentially utilized to study essential genes within these genomes.

We have updated the manuscript to include these results.

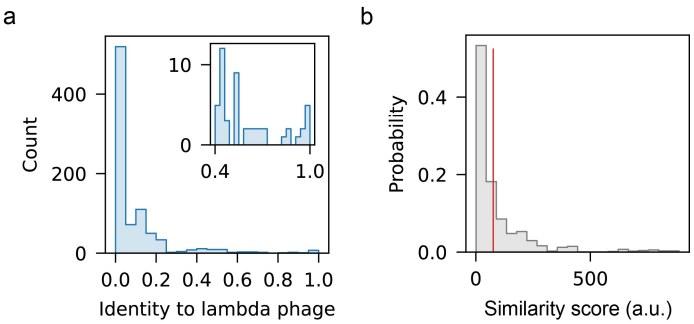

**Figure R1**: **Sequence similarity of training dataset and the lambda phage genome**. **a**) Histogram showing the distribution of identity of training sequences that aligned with the lambda phage genome through BLAST analysis (n = 847). The identity for a specific sequence is calculated by summing the lengths of all aligned segments with the lambda phage genome and then dividing this total by the full sequence length. Inset, the counts of sequences with an identity above 0.4. **b**) Histogram showing the distribution of the similarity score of randomly sampled phage genomes from the training dataset (n = 2,000). For a specific genome, the similarity score is defined as the sum of identity for all the training sequences with significant hits. The similarity score for lambda phage is denoted in red. This figure is labeled as **Supplementary Figure 3** in the revised manuscript.

**Revised text** (Section "megaDNA allows zero-shot prediction of gene essentiality"):

> We further analyzed the similarity of sequences in the training dataset to the lambda phage genome (Supplementary Fig. 3). We found that 847 training sequences aligned with the lambda phage genome through BLAST analysis. Among these sequences, 50 show a sequence identity above 0.4, and 8 have an identity exceeding 0.9. This finding indicates that our model's performance benefits from a broad spectral of related references in the training dataset, predominantly consisting of low similarity sequences and supplemented by a small proportion of highly similar ones. In addition, about 34% of the phage genomes have more representations in the training sequences than the lambda phage (Supplementary Fig.3), suggesting that the megaDNA model could be potentially utilized to study essential genes within these genomes.

l. 72: "Our model's prediction performance closely matched the state-of-the-art model DeepSequence […] our model successfully predicted the impact of SNPs across the T7 bacteriophage genome": I agree with the author that MegaDNA performances seem on par with the performances of DeepSequence, however I struggle somewhat to understand how a Spearman correlation coefficient of 0.5 can lead to a useful predictor for biologists. Said otherwise, I agree that there is some signal here, but I don't see in the current figures a way to understand whether this signal would be sufficient to truly predict the impact of individual SNPs on gene fitness. I would encourage the author to provide, as Supplementary Figure, the x-y plots of the observed vs predicted fitness for each correlation bar chart (panels f, g, and i in Fig. 1), and discuss these x-y plots in the text.

We thank the reviewer for raising this important point. In the revised manuscript, we have included the observed vs predicted fitness plots for Fig. 1f, 1g and 1i (**Fig. R2**).

For the prediction of protein fitness in Fig.1f, the standard deviation of the prediction error is 0.16, much lower than the dynamic range of the fitness measurement which ranges from -0.4 to 1.2. We observed similar results for the prediction of the impact of SNPs on gene fitness in T7 bacteriophage. For example, standard deviation of the prediction error for gene 1.3 is 8.0, while the dynamic range of the measured fitness is 87. It is also worth noting that the model's performance depends on the availability of training dataset (**Fig. 1f**). The T7 bacteriophage dataset covers only about 15% of all possible SNPs

(Supplementary Fig. 6), in contrast to the DMS dataset that includes all single codon mutations. Despite this constraint, our model achieved a Spearman correlation coefficient of 0.71 for gene 1.3 and 0.54 for T7 RNAP. For genes with lower prediction accuracy, such as gene 17, the model still identifies mutations with significant fitness effects.

We have also revised our manuscript to discuss these results, in line with the reviewer's suggestion.

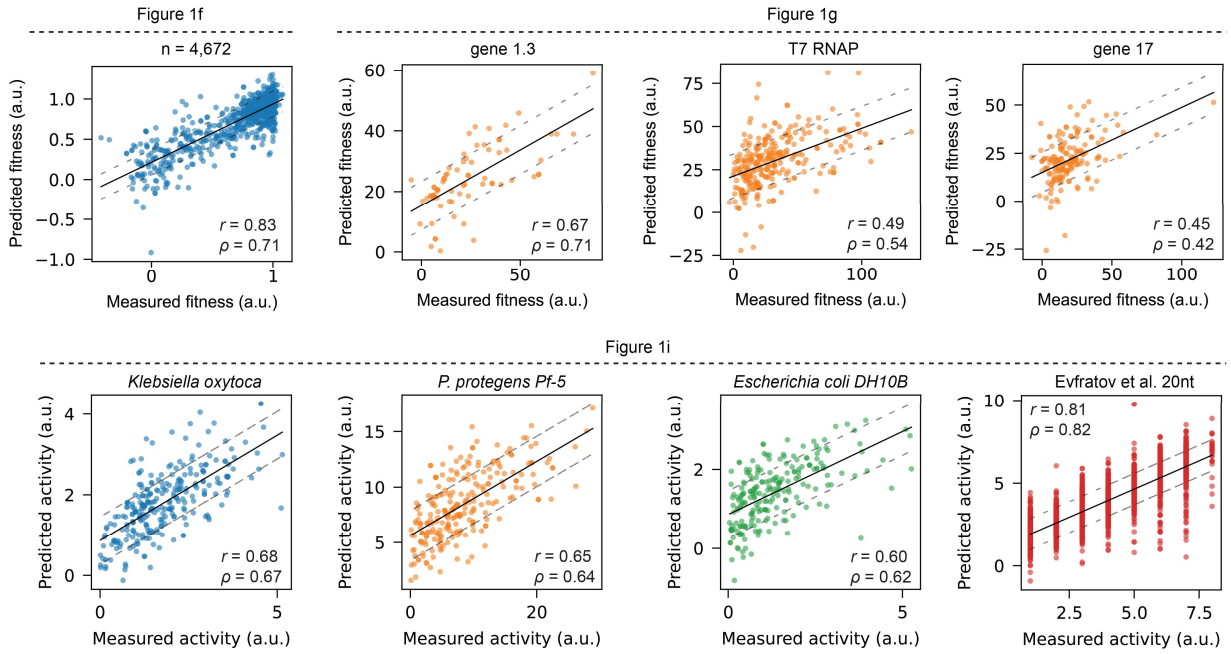


**Figure R2**. **Correlations between model predictions and experimental measurements of protein fitness and regulatory element activities**. Each dot represents an observed vs predicted value of either measured protein fitness (Figure 1f and 1g) or translational activity of 5'UTR sequences (Figure 1i). *r* and *p* are Pearson and Spearman correlation coefficients between the true and predicted values, respectively. The black line represents the linear regression fit, and the dashed lines show the fitted values plus or minus standard deviation of the prediction residuals. This figure is labeled as **Supplementary Figure 4** in the revised manuscript.

**Revised text** (Section "megaDNA learns functional properties of proteins and regulatory elements"):

> The standard deviation of the prediction error is 0.16, which is much smaller than the full dynamic range of the protein fitness measurement (Supplementary Fig. 4).
>
> It is worth noting that this dataset covers only about 15% of all possible SNPs (Supplementary Fig. 6), which is substantially smaller than the DMS dataset. Despite this constraint, the Spearman correlation coefficient between predicted and measured impact is 0.71 for gene 1.3 and 0.54 for T7 RNAP (Supplementary Fig. 4). For genes with lower prediction accuracy, such as gene 17, our model still identifies specific mutations with significant fitness effects.

The Spearman correlation coefficients range from 0.62 to 0.82 for these datasets, illustrating our model's capacity to capture translation-related sequence features.

l. 82: "The embeddings from […] demonstrate the broad applicability of our model.": As currently described, I am worried about potential over-fitting in this linear regressor. The author mentions "5-fold Stratified K-Fold cross-validation test", however there is no mention of controlling for the redundancies between training and testing set in this cross-validation. The methods l. 303 mention "maintaining the same proportion of samples for each class as in the complete dataset", but to my understanding this would be the same proportion of viruses, bacteria, and archaea, without checking the similarities between viral, bacterial, and archaeal sequences across training and testing sets. Without this important control, it is very common when training on genome databases that near-identical genomes are present in both training and testing folds. For taxonomic classification, this should be evaluated by leaving out entire taxa from the training set, and testing on these classes, or by weighing the fold split based on pairwise genome similarity.

We thank the reviewer for this insightful comment. Following the reviewer's suggestion, we have adjusted the cross-validation tests to incorporate sequence-to-sequence similarity. In brief, for each training-testing dataset split, we used BLAST analysis to identify overlaps between training and testing sequences. Each test sequence with a significant match in the training dataset was assigned a weight ranging from 0 to 1. A weight of 0 removes these test sequences from the AUROC calculation and a weight of 1 corresponds to the original result where all the test sequences are included.

We found that introducing the weighing scheme only have a minor impact on our model's performance (**Fig. R3**). Even when excluding all similar sequences in the testing set, the mean reduction in AUROC is 0.03, 0.01, 0.02 for the global, middle and local layers. These results demonstrate the robust performance of our model for distinguishing sequences from different domains. Our modifications to the manuscript are reproduced in the box below.

**Revised text** (Section "megaDNA learns functional properties of proteins and regulatory elements"):

We further weighted the model predictions on the test datasets based on their sequence similarity to the training datasets. A weight of 0 excludes test sequences that have at least one matched training sequence and a weight of 1 includes all test sequences in the AUROC calculation (methods). Our results indicate that completely ruling out similar test sequences only results in a reduction of AUROC by 0.03, 0.01, and 0.02 for different model layers (Supplementary Fig. 9).

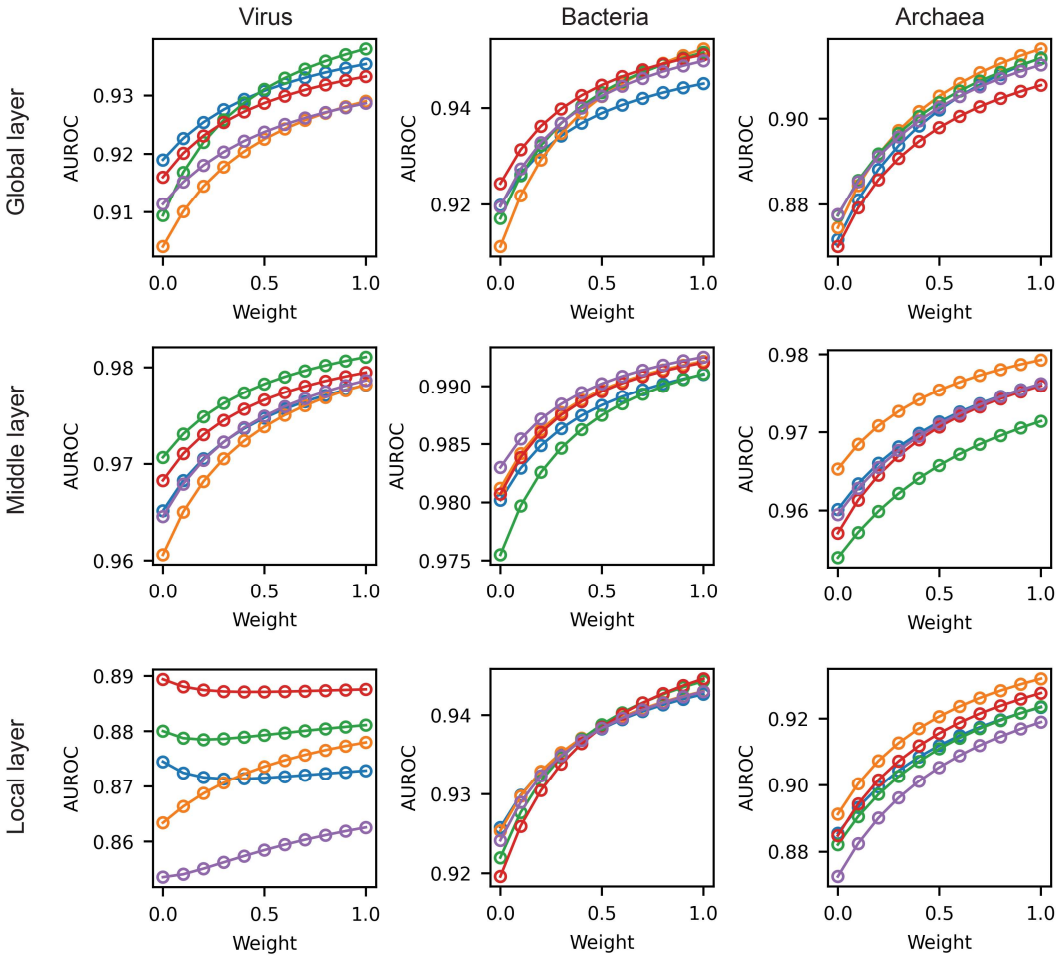


**Figure R3. Effect of sequence similarity on the accuracy of taxonomy prediction.** Model predictions on the test dataset are weighted by their pairwise sequence similarity to the training dataset: a weight of 0 excludes test sequences that have at least one similar sequence in the training dataset, and a weight of 1 include all the sequences (methods). Each color represents AUROC from one of the five folds in the cross-validation tests. This figure is labeled as **Supplementary Figure 9** in the revised manuscript.

l. 92: “Among all these sequences, 607 have a virus score larger than zero.”: This does not seem to correspond to a geNomad output (since by default the minimum virus score considered is 0.7). It also seems inconsistent with Fig. 2c, which suggests that all generated sequences had a virus score greater than 0.7. Please double-check and correct as needed. In particular, please make sure to report the virus score of all sequences, and not just the ones geNomad predicted as viral (see geNomad output “aggregated_classification.tsv” in the output folder “aggregated_classification”).

We would like to thank the reviewer’s constructive feedback concerning the virus scores. We apologize for the confusion caused by our initial figure and text.

To address this issue, we have rerun geNomad on all the generated sequences with the "--relax" flag to disable the post-classification filters that implies the virus score cutoff of 0.7. In addition, we have reported the virus scores from the file "aggregated_classification.tsv" in **Fig. R4**, as the reviewer suggested. The virus scores now range from 0.08 to 0.97 and the median value is 0.75. We have updated the figure to ensure that they accurately reflect virus scores of all the sequences.

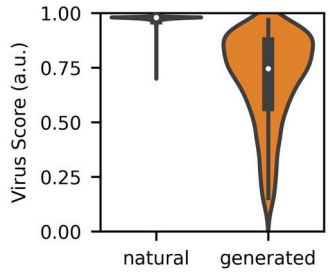


**Figure R4. Comparison of the predicted virus scores for all generated sequences and the training dataset.** This figure is labeled as **Figure 2c** in the revised manuscript.

**Revised text** (Section "megaDNA generates de novo genomic sequences"):

> The median virus score of all generated sequences is 0.75 and the maximum score is 0.97.

l. 96: "The median virus score of these generated sequences is 0.84 and the maximum score is 0.97, comparable to the virus scores of natural bacteriophage genomes which range from 0.70 to 0.98 (Fig. 1c).": I am not sure I understand how the median virus score can be 0.84 when only 607 out of 1,024 sequences had a score larger than zero? From Fig. 1c, it also seems as if the median score for nature phage genome is around 0.99, so I feel like calling the score distribution of generated sequences "comparable to the virus score of natural bacteriophage genomes" is quite a stretch. In my opinion, the distribution of scores for these generated sequences is clearly lower than the one of natural phage genomes (especially when considering that the distribution of score between 0.7 and 1.0 is a parameter of geNomad, not a feature of the sequences themselves).

We agree with the reviewer that the distribution of virus scores for the generated sequences is lower than that of natural bacteriophage genomes (median value 0.75 vs. 0.98, **Fig. R4**). We have revised our manuscript to clarify this point.

**Revised text** (Section "megaDNA generates de novo genomic sequences"):

> These scores are lower than those of natural bacteriophage genomes, which have a median value of 0.98 (Fig. 2c).

L. 100-102: I have concerns about this analysis, which I believe are more concerns with the results from DeepHost than with MegaDNA. For instance, the fourth more common predicted host is Cellulophaga baltica, a marine bacteria, which is suspicious for sequences generated based on human gut phages. Given the uncertainty around these predictions, the lack of demonstrated reliability of DeepHost for novel sequences, and the lack of benchmarking of this tool for its ability to discriminate bacteriophages from viruses of archaea, viruses of eukaryotes, and non-viral sequences, I would suggest removing this analysis or add these caveats to the text. (Said otherwise, I believe that bacterial sequences used as input to DeepHost would yield similar results, so that in the end it is not really a good test for the generation of phage sequences per se).

Following the reviewer's suggestion, we have deleted this analysis from the revised manuscript.

l. 116: "functional regulatory sequences": I don't believe this analysis demonstrate that these are "functional", and would suggest rephrasing as "potential regulatory sequences"

We have changed our phrase to "potential regulatory sequences". Our modifications are reproduced in the boxes below.

**Revised text** (Section "Abstract"):

> Furthermore, it generates de novo sequences up to 96K base pairs, which contain potential regulatory elements…

**Revised text** (Section "megaDNA generates de novo genomic sequences"):

> We then examined the 5'UTR of the annotated genes in the generated sequences to determine if they contain potential regulatory elements…

**Revised text** (Section "megaDNA generates de novo genomic sequences"):

> In short, our generated sequences harbor potential regulatory sequences that…

l. 123: "suggesting that these generated proteins are more likely to adopt a stable conformation.": I disagree with this claim, higher pLDDT score in my opinion could very well reflect that these sequences are similar to the ones used to train ESM-Fold (which would make sense considering the overlap in training set between megaDNA and ESM-Fold).

We appreciate the reviewer's comment on the potential overlap between the training datasets of megaDNA and ESM-Fold, and we acknowledge that there might be inherent similarities between the generated sequences and sequences used in training ESM-Fold. We have deleted the original statement regarding protein stability and revised the manuscript to clarify this point.

**Revised text** (Section "megaDNA generates de novo genomic sequences"):

> We further randomly sampled 10K annotated genes from all the generated sequences and found high pLDDT scores for them (median value of 36, Supplementary Fig. 14). This result may suggest inherent similarities between the generated sequences and sequences used in training ESM-Fold.

l. 125: To my knowledge, deepFRI is not a state-of-the-art tool for phage genome annotation. Instead, I would recommend the author use a standard pipeline such as pharokka, and a phage-specific database such as PhrogDB.

We would like to thank the reviewer for this valuable suggestion. Following this advice, we have used two protein function annotation tools including pharokka (Bouras et al., 2023) and phold (https://github.com/gbouras13/phold), both of which utilize the phage specific database PhrogDB. In contrast to Pharokka's sequence-based search method, phold employs protein structural homology to predict protein functions. We found phold identifies more functional proteins for the generated sequences, in accord with the author's claim "phold strongly outperforms Pharokka, particularly for less characterised phages such as those from metagenomic datasets" (https://github.com/gbouras13/pharokka?tab=readme-ov-file#phold). Consequently, we have chosen to report protein functions using phold.

Our analysis identifies several large protein families with functions related to phages, including tail length measurement, DNA metabolism, and transcription regulation (**Fig. R5**). These findings demonstrate our model's ability to generate potentially functional proteins.

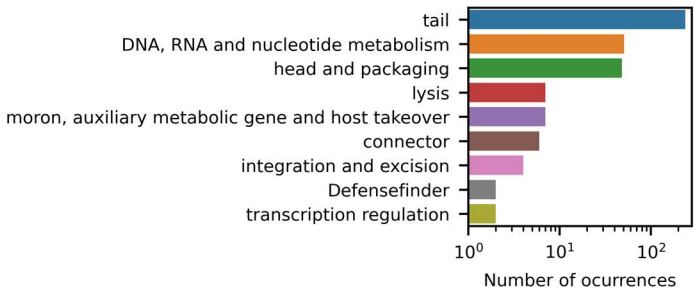


**Figure R5. Top 10 predicted functions of proteins derived from the generated sequences.** Phold was used for protein function annotations. This figure is labeled as **Figure 2i** in the revised manuscript.

**Revised text** (Section "megaDNA generates de novo genomic sequences"):

> We further annotated gene functions using phold[24]. In brief, phold leverages a protein language model[25] to derive structural information from protein sequences. This information is compared against a structural database via Foldseek[26] to obtain PHROG annotations[27]. Our analysis reveals several large

protein families associated with phage-related functions, such as head and packaging, and nucleotide metabolism (Fig. 2i).

**References**

Bouras G, Nepal R, Houtak G, Psaltis AJ, Wormald PJ, Vreugde S, Pharokka: a fast scalable bacteriophage annotation tool, Bioinformatics, Volume 39, Issue 1, January 2023, btac776

l. 128: One analysis I think would be valuable here would be an estimation of how diverse the set of generated sequences is. This could be achieved e.g. based on a clustering of these generated genomes with vContact2, but even if using another tool, the idea would be to illustrate for the reader how often the model generates closely related sequences (or not).

We thank the reviewer for this constructive suggestion. We have used vContact2 to cluster the generated sequences with default parameters (Bin Jang et al., 2019). We found all generated sequences form singletons, suggesting a high degree of diversity. Furthermore, we have run BLAST analysis of the generated sequences against each other and found no significant hit. We have updated the manuscript to include these results.

**Revised text** (Section "megaDNA generates de novo genomic sequences"):

We further analyzed the generated sequences using vContact2[20] and found that all generated sequences form singletons, indicating a high diversity among them.

**References**

Bin Jang, H., Bolduc, B., Zablocki, O. et al. Taxonomic assignment of uncultivated prokaryotic virus genomes is enabled by gene-sharing networks. Nat Biotechnol 37, 632–639 (2019)

l. 135: "with further scaling up and fine-tuning, we envision that generative genomic models have the potential to enable de novo design of the whole genome": The first part ("with further scaling up and fine-tuning") is quite an under-statement in my opinion. I believe the author nicely demonstrated that this kind of model may have the potential to learn some of the constraints underlying phage genomes, however from the limited results presented here, it seems like these models are still very far away from the de novo design of whole functional genomes, and rather represent a first step towards this goal.

We appreciate the reviewer's perspective and agree that our model represents the first step towards the more ambitious goal of the de novo design of functional genomes. We have revised our manuscript to clarify this point.

**Revised text** (Section "Discussion"):

> Despite these limitations, we envision that our generative genomic model represents the first step towards the *de novo* design of the whole functional genome…

l. 231: This section should ideally include a paragraph describing how the model was built. For instance, the manuscript mentions l. 38: "a long-context language model", however it was not clear to me what "long-context" really meant, as the manuscript did not illustrate the size of the context window. A paragraph describing the structure of the model and its main characteristics would be helpful in my opinion.

We have added a new paragraph to describe the main characteristics of our model, as the reviewer suggested. The modifications are reproduced in the box below.

**Revised text** (Section "megaDNA allows zero-shot prediction of gene essentiality"):

> Traditionally, transformer-based language models only process a few thousand tokens of context because the computational cost of the self-attention mechanism scales quadratically with sequence length. This context window is not sufficient to model nucleotide-level tokenized phage genomes. To overcome this problem, we employed a multi-scale transformer structure, adapted from Yu et al.[8], to model the long-range context information. This architecture consists of three decoder-only transformer layers with multi-head attention and each layer captures sequence information at different resolutions: the local layer processes embeddings of tokenized sequences within a 16 bp window. Its output serves as the input to the middle layer, which has a context window of 1024 bp. Finally, the global layer utilizes information from the middle layer to model sequence dependencies across the whole input context (96K bp).

Minor comments:

l. 53: "genebank," should be "genbank,"

Thanks. We have fixed this error.

l. 81-82: "We collected unannotated sequences from bacteriophage, bacteria, and archaea.": Please briefly clarify the source of these sequences here.

We have included the source of these sequences in the methods section.

**Revised text** (Section "Methods - Classification taxonomy of unannotated sequences"):

> We analyzed 10K bp sequences randomly sampled from bacteriophage, bacteria, and archaea genomes downloaded from NCBI GenBank (n = 5,000 each). More specifically, we collected complete genome sequences of bacteria, archaea and bacteriophage from NCBI (ftp://ftp.ncbi.nlm.nih.gov/genomes) and randomly sampled 10K bp sequence fragments from genomes longer than 10K bp.

l. 99: "Caudoviricetes", and all viral taxa names, should be italicized (also "Caudovirales." l. 234). Caudoviricetes

We have made all viral taxa names italicized.

l. 230 and throughout: All mentions of software and database should also include version number.

We have included the version number of the software and database in the methods section.

l. 232: The different sources of training sequences should come with a citation and, ideally, a specific list of accession numbers of the exact sequences used for training (so that others could try to reproduce/reconstruct the training set).

We thank the reviewer for this constructive feedback. We have provided the accession numbers of the training sequences as a tsv file in the GitHub Repository (https://github.com/lingxusb/megaDNA/blob/main/training/training_seqs.tsv).

l. 237: "whose predicted host is not a unicellular organism": It is not entirely clear to me how this was derived from the taxonomy. Please provide a list of taxa that were excluded and/or included in the final dataset.

We have provided the predicted taxonomy of the training dataset as a tsv file in the GitHub repository (https://github.com/lingxusb/megaDNA/blob/main/training/training_taxanomy.tsv).

l. 252: "gens" should be "genes"

We have corrected this typo in the revised manuscript.

l. 279: "that was calculated" should be "was calculated"

We have fixed this error.

Fig. 2d: "generated sequences" should be "generated sequences predicted as viral"

We have corrected this sentence, as the reviewer suggested.

Reviewer #3 (Remarks on code availability):

The code is well documented, comes with multiple examples (in the Readme and in notebooks), and I was able to install and run the tool with (very) minimal issues.

We appreciate the reviewer's positive comments.

**Reviewer #5** (Remarks to the Author):

In this paper, the author discusses megaDNA, a language model trained on bacteriophage genome to yield predictions (gene essentiality, mutations, etc.) as well as generating new phage genomes. The model was trained on bacteriophage genome data with byte-level tokenisation. The manuscript is of high quality, as it is well written with little to no errors and beautiful figures. Even the supplementary figures are of high quality, which is a rare sight.

I will start out by saying I am by no means an expert when it comes to large language models, so my review should not be read as such. That said, I have a basic understanding on the matter. I have therefore tried to mostly review the interpretation of the results, which I found somewhat difficult. I believe the latter was not due to my lack of knowledge in the field of LMs, but due to the brevity of the manuscript (which I understand may be a requirement), but left me unsatisfied with what the results of the tool actually mean.

The zero-shot predictions of gene essentiality intrigued me, although I am left wondering whether model loss indeed is a good predictor of gene essentiality. That said, figure 1c-d seems to indeed indicate that this assumption is fair.

We would like to thank the reviewer for the positive feedback on our manuscript.

The predictions on gene functionality is the first bit I am scratching my head about. This is not necessarily because of my scepticism, but after reading the current manuscript it is not clear what exactly the related figures (e.g. 1f, 1g) show. I have read the method a couple of times to confirm that I wasn't missing something, but in the end I have to simply conclude it is unclear. Perhaps this part is very straightforward for people working in the LLM field, but with the broad audience of NatCom that is simply not good enough. To some extent, I have the same concerns about the translational and taxonomic predictions, although I think I understand those a bit better. Please elaborate better on what these data show.

We thank the reviewer for this helpful suggestion, and we apologize for the lack of clarity in the original version of the manuscript. To fully address this issue, we have systematically revised our manuscript to better explain our methods and results. More specifically, we have discussed the general workflow for using the language model to predict functional properties of the inputs. For each predictive task, we have provided detailed information about the dataset used and explained how the megaDNA model was used to make the final predictions.

(1) Utilizing sequence embeddings for predictive tasks.
Sequence embeddings are high-dimensional representations of the model input produced by the language model. The assumption is that these embeddings capture complex patterns and relationships within the data and can be used to make quantitative predictions about the property of the input. This assumption has been supported by previous studies in the field of protein language models (Villegas-Morcillo et al., 2022, Marquet et al., 2022). Since our megaDNA model takes DNA sequences as the

model input, we could harness the model's learned representations for a wide range of predictive tasks including genetic variant effect prediction and translation activity prediction for 5'UTR. Technically speaking, model embeddings are vectors corresponding to the activities of a series of neurons within the language model. For each input sequence, we calculate the model embedding using the megaDNA model, and this embedding can be used as the input for another machine learning model, such as the linear regression model, to predict experimental metrics associated with the original sequence input.

(2) Prediction of genetic variant effect.

We focused on a deep mutational scanning (DMS) dataset for genetic variant effect prediction. This dataset includes all possible single codon mutations of the essential gene *infA* in *E. coli* and the mutational effects were measured as fitness values through a growth competitive assay. The model inputs were the mutated gene coding sequences. We generated the corresponding sequence embeddings which were then used to predict fitness scores via linear regression. To make our analysis more reproducible, we have used 5-fold cross validation tests and reported the correlations between the measured and predicted fitness for the test datasets (**Fig. 1f).** Our model performs better with increasing number of training samples (n), and the scatter plot for the measured vs predicted protein fitness is shown in **Fig. R6**.

(3) Prediction of the impact of SNPs for T7 bacteriophage.

In this dataset, the fitness of SNPs was measured using a multiplexing sequencing approach. The mutated gene sequences were used as model inputs to derive embeddings. These embeddings and the corresponding SNP fitnesses were then used to train linear regression models, which was evaluated using 5-fold cross validation tests. It is worth noting that this dataset only covers about 15% of all possible SNPs. However, we still observed a Spearman correlation larger than 0.5 for genes including T7 RNAP and gene 1.3, demonstrating the robust performance of our model to extract useful protein function information from the DNA sequence (**Fig. 1g and Fig. R6**).

(4) Prediction of translation efficiency of 5'UTRs in non-model bacteria.

We leveraged the paired RNA-seq and ribosome profiling datasets to quantify the translation efficiency of 5'UTRs across three bacteria genomes. The translation efficiency was defined as the ratio of the normalized ribosome density and the RNA level for the genes. The 5'UTR sequences were used as model input to derive embeddings, which were then used to predict their translation efficiencies through linear regression. In addition to the endogenous regulatory elements, we also benchmarked the predictive performance of our model based on a high-throughput Flow-seq experiment which measures the translation activity of a random 5'UTR library in *E. coli*. We observed a high correlation between measured and predicted translational activities, indicating our model's capacity to capture translation-related sequence features (**Fig. 1i and Fig. R6)**.

(5) Taxonomy classification of DNA sequences based on model embeddings.

We randomly sampled 10K bp DNA fragments from bacteria, archaea and bacteriophage genomes (methods). These DNA sequences serve as model input and the corresponding embeddings were mapped into a low-dimensional space, where we observed clear separations among different domains

(**Fig. 1k**). We then used the sequence embeddings and the domain labels to train logistic regression models for taxonomy classification. These models achieved a mean area under the ROC curve (AUROC) scores of 0.98, meaning that the model has a 98% chance of correctly predicting a true positive over a false positive across all decision thresholds.

The revisions to the manuscript are reproduced in the boxes as below.

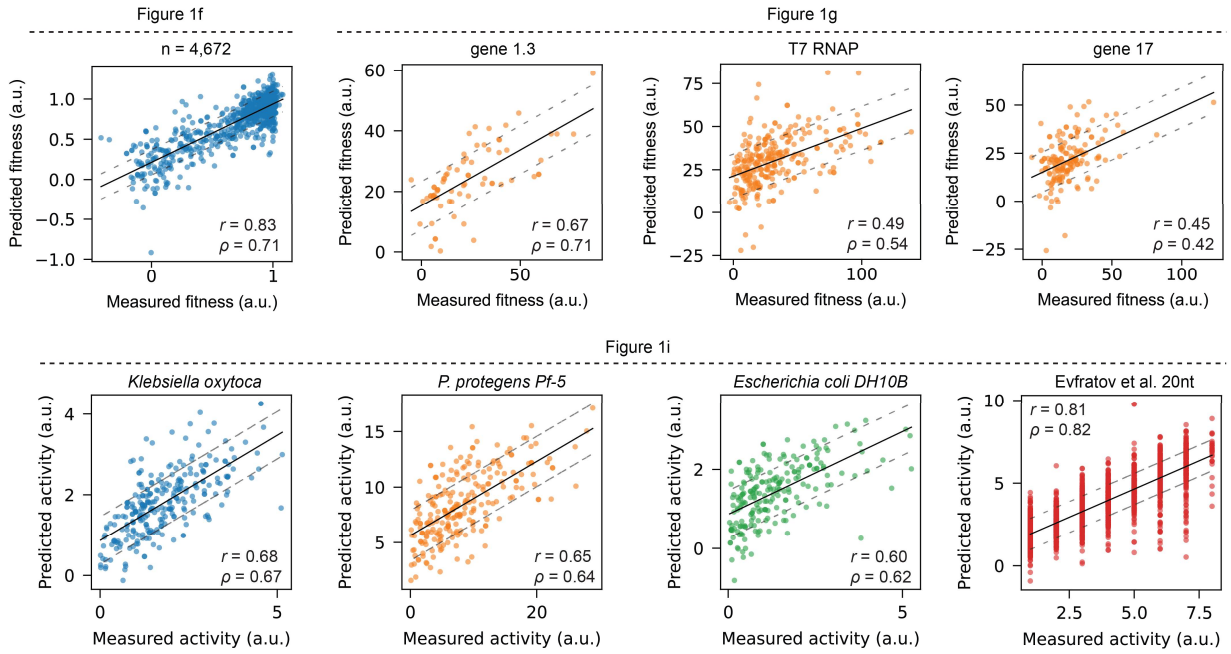


**Figure R6. Correlation between model predictions and experimental measurements of protein fitness and regulatory element activities.** Each dot represents an observed vs predicted value of either measured protein fitness (Figure 1f and 1g) or translational activity of 5'UTR sequences (Figure 1i). *r* and *p* are Pearson and Spearman correlation coefficients between the true and predicted values, respectively. The black line represents the linear regression fit, and the dashed lines show the fitted values plus or minus standard deviation of the prediction residuals. This figure is labeled as **Supplementary Figure 4** in the revised manuscript.

**Revised text** (Section "megaDNA learns functional properties of proteins and regulatory elements"):

Sequence embeddings are high-dimensional representations of the model input that capture rich contextual information. These embeddings can be used to make predictions about the quantitative property related to the original input. Since our megaDNA model takes DNA sequences as the model input, we could harness the model's learned representations for a wide range of predictive tasks. We first evaluated our model's ability to predict effects of sequence mutations on protein functions using a deep mutational scanning (DMS) dataset for the *E. coli* essential gene *infA* (Fig. 1e). This dataset includes all possible single codon mutations for *infA* and the corresponding mutational effects measured as

fitness values through a growth competitive assay. To model protein fitness, mutated gene coding sequences were used as inputs. Then a linear regression model was trained on sequence embeddings derived from the internal activities of neurons within the megaDNA model to predict the mutational effects. The standard deviation of the prediction error is 0.16, which is much smaller than the full dynamic range of the protein fitness measurement (Supplementary Fig. 4).

In this case, the mutated gene sequences were used as model input and the sequence embeddings were utilized to train regression models that predict SNP impacts, quantified as rates of mutability[15]. It is worth noting that this dataset covers only about 15% of all possible SNPs (Supplementary Fig. 6), which is substantially smaller than the DMS dataset. Despite this constraint, the Spearman correlation coefficient between predicted and measured impact is 0.71 for gene 1.3 and 0.54 for T7 RNAP (Supplementary Fig. 4). For genes with lower prediction accuracy, such as gene 17, our model still identifies specific mutations with significant fitness effects.

The 5'UTR sequences were used as model input to derive embeddings, which were then used to predict their translation efficiencies via linear regression. We leveraged the paired RNA-seq and ribosome profiling datasets to quantify the translation efficiency of 5'UTR across three bacteria genomes. The translation efficiency was defined as the ratio of the normalized ribosome density and the RNA level for the genes. Our model effectively predicted the translation efficiencies of 5'UTR in both model and non-model organisms, including *K. Oxytoca*, *P. Protegens*, and *E. coli* (Fig. 1i). In addition to the endogenous regulatory elements, we also benchmarked the predictive performance of our approach on the high-throughput measurement of the translational activity of a random 5'UTR library in *E. coli* (Fig. 1i). The Spearman correlation coefficients range from 0.62 to 0.82 for these datasets, illustrating our model's capacity to capture translation-related sequence features.

Lastly, we extended our model to identify taxonomy of unannotated sequences at the domain level (Fig. 1j). We collected unannotated sequences from bacteriophage, bacteria, and archaea genomes, which were used as model input to calculate their embeddings. Then these embeddings were mapped into a low-dimensional space, where we observed clear separations among different domains (Fig. 1k). By training logistic regression models based on sequence embeddings and the domain labels, we achieved a high classification accuracy in the cross-validation tests (average AUROC of 0.98, Supplementary Fig. 8).

**Revised text** (Section "Methods -- Prediction of mutational effects on protein function"):

Sequence embeddings are high-dimensional representations of the model input produced by the language models. The assumption is that these embeddings capture complex patterns and relationships within the data, which has been supported by previous studies in the field of protein language models[30, 31]. Technically speaking, model embeddings are vectors corresponding to the activities of a series of neurons within our model.

**References**

Villegas-Morcillo A., Gomez A.M., Sanchez V., An analysis of protein language model embeddings for fold prediction, Briefings in Bioinformatics, 23:3, bbac142 (2022).

Marquet, C., Heinzinger, M., Olenyi, T. et al. Embeddings from protein language models predict conservation and variant effects. Hum Genet 141, 1629–1647 (2022).

Next, the de novo generation of sequences is discussed. While this is very exciting on the one hand, the ability to generate novel variations is a known property of LLMs. I am not saying this is not exciting, but it would be better to highlight specific examples rather than showing how it can generally produce phage-like genomes. With AI becoming an increasingly important tool in science, illustrating that "it works" is only moderately exciting. Note that I am fully aware of the efforts that must have gone into megaDNA, and that I say this with the best intentions of seeing the tool and its utilities grow.

We would like to thank the reviewer for this constructive comment, and we agree that showing specific examples of de novo generated sequences would strengthen the manuscript. In the revised manuscript, we have included detailed examples to better illustrate the capabilities of our model. More specifically, we present the generated sequence #87 in **Fig. 2** and a new example in **Fig. R7**. In both examples, we have highlighted the annotated genes and potential regulatory elements for transcriptional and translation initiation.

(1) Example of generated sequence #87

The generated sequence #87 was not referenced correctly in the original manuscript. We apologize for this error and have corrected the citations in the revised manuscript. We have highlighted the promoter and ribosome binding site (RBS) regions for the predicted phage stabilization gene (**Fig. 2e**). In addition, we observed that the promoters of predicted genes have higher transcriptional activity than random sequences with the same length (**Fig. 2f**). Notably, there is a high ratio of adenine (A) and guanine (G) nucleotides before the start codons, which suggests potentially functional ribosome binding sites to initiate translation (**Fig. 2g**).

(2) Example of generated sequence #212

We have provided another example of the generated sequence in **Fig. R7**. In this example, we identified a promoter with high transcriptional activity in the upstream of the predicted capsid protein N-terminus. Additionally, we observed high promoter activities for the 5'UTRs in this generated sequence, similar to the previous example. The ratio of A and G nucleotides of the 5'UTRs show peaks around the -10 bp region relative to the start codon, which is also favorable for efficient translation initiation.

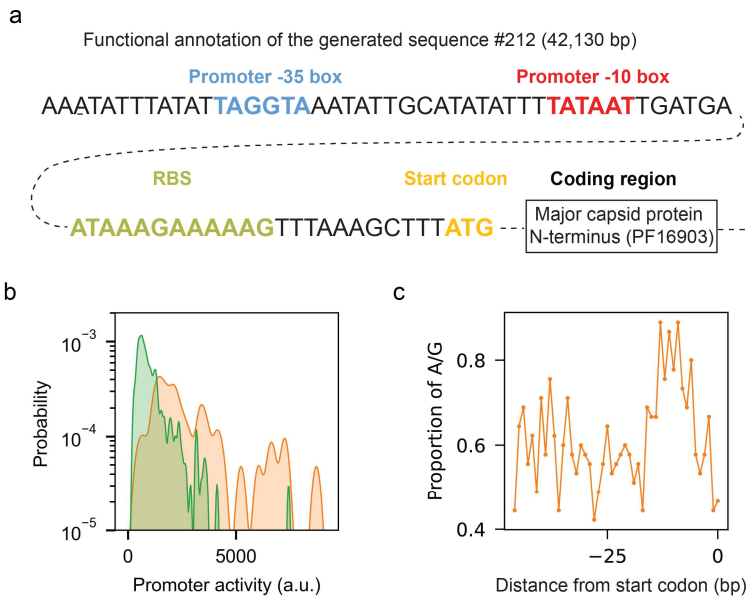


**Figure R7. Example of a generated sequence. a**) Functional annotation of a selected sequence fragment (generated sequence #212). **b**) Predicted promoter activity for all the 5'UTRs in the generated sequence (n = 45), along with the promoter activity of the random sequences with the same length. Promoter activities were calculated using the Promoter Calculator. Kolmogorov-Smirnov test: p value = $6.8 \times 10^{-15}$. **c**) Proportions of adenine (A) and guanine (G) nucleotides preceding the start codon of the predicted genes in the generated sequence #212. This figure is labeled as **Supplementary Figure 11** in the revised manuscript.

In summary, we have provided two examples of the generated sequences in the updated manuscript. We also acknowledge that the megaDNA model is not without limitations and further improvement of the model would be a valuable avenue for future research. The modifications to our manuscript are reproduced in the boxes below.

**Revised text** (Section "megaDNA generates de novo genomic sequences"):

> In another sequence example (#212), we observed similar promoter activities and RBS characteristics, and we found a consensus -10 sequence TATAAT that located upstream of the N-terminus of the major capsid protein (Supplementary Fig. 11). This trend of A/G enrichment and high promoter activity within the 5'UTRs is also consistent across all the generated sequences (Supplementary Fig. 12 and 13).

**Revised text** (Section "Discussion"):

> Improvements of natural language models have been driven by the scaling-up of training dataset and model size, along with techniques for model fine-tuning and alignment with human input. Such approaches are likely to further improve the performance of the megaDNA model.

I am more than happy to read a revised manuscript in the future.

Some minor issues:

> L63: "Used *as* a zero-shot predictor"?

We have added “as” in the text to correct for this error.

> A lot of the subfigures from figure 2 are never referred to in the text. This makes it very hard to contextualise why these results are there in the first place.

We apologize for this mistake. We have refereed to all the panels of Fig.2 in the revised manuscript.

Reviewer #5 (Remarks on code availability):

I have not run the software, but the documentation is well done.

We thank the reviewer for the positive feedback on our software.

**Reviewer #6** (Remarks to the Author):

In their article, Dr. Shao presents their transformer-based language model for generation and evaluation of bacteriophage sequences. This article represents one of the first known applications of transformer models to nucleotide sequences. Through a variety of analyses, Dr. Shao demonstrates the impressive performance of megaDNA for a wide range of classification and generation tasks. The methods and results are mostly well-described and the vast majority of my comments are requests for additional detail or clarification.

We appreciate the reviewer's positive comments on our work.

Major Comments

- Data and Code availability: A list of the NCBI accession numbers and identifiers for the other sequences used to train the model is important for future replication and evaluation. For example, Ratcliff 2024 noted that the compositional features of megaDNA generated sequences are more similar to specific bacteriophage families than others - documentation of the included sequences could provide evidence as to whether this observed effect is due to family-level differences in the number of sequences present in the training set. These could be provided as a .txt or .tsv file within the megaDNA GitHub repository.

We thank the reviewer for this valuable feedback. We have included the accession numbers of the training sequences as a tsv file in the GitHub repository (https://github.com/lingxusb/megaDNA/blob/main/training/training_seqs.tsv).

We have also updated the manuscript to include this citation.

**Revised text** (Section "Discussion"):

| |
|---|
| A recent study by Ratcliff et al. shows that the generated sequences are compositionally more similar to certain bacteriophage families than the others[29]. |

Minor Comments
- Lines 53, 232, 296, and 329: Correct "genebank" to "GenBank".

We thank the reviewer for careful reading of our work. We have fixed this error.

- Line 77: The assessment of regulatory element detection in bacteria is a clever method of evaluating the bacteriophage DNA model. Please adjust "in bacteria" to "in bacterial genomes" to make it more clear these predictions are being done on the bacteriophage hosts rather than the bacteriophages themselves.

We have revised our text following the reviewer's comment.

**Revised text** (Section "megaDNA learns functional properties of proteins and regulatory elements"):

> We investigated the potential of the model embeddings to predict regulatory element activity in bacteria genomes (Fig. 1h).

- Line 83 and 301: Please confirm if the taxonomic classification used linear regression (as stated in 83) or logistic regression (as stated in 301).

We apologize for the confusion. We have used logistic regression to predict the taxonomy for the input sequences and we have revised the text to improve its clarity.

**Revised text** (Section "megaDNA learns functional properties of proteins and regulatory elements"):

> By training logistic regression models based on sequence embeddings and the domain labels, we achieved a high classification accuracy in the cross-validation tests (average AUROC of 0.98, Supplementary Fig. 8).

- Lines 81-87: In the paragraph, please make it clear that the taxonomic classification is only to the domain level.

We have clarified that the sequences were classified at the domain level.

**Revised text** (Section "megaDNA learns functional properties of proteins and regulatory elements"):

> Lastly, we extended our model to identify taxonomy of unannotated sequences at the domain level (Fig. 1j)

- Line 92: This sentence states that 607 sequences had virus scores greater than 0. However, figure 2D has taxonomic classifications for 610 sequences. Please clarify the discrepancy. Is it possible that some of the taxonomic classifications are for viruses with geNomad scores below 0.7?

We thank the reviewer for careful reading of our work and noticing this. In the original Fig. 2D, the 223 *Caudoviricetes* sequences includes both the *Crassvirales* and *Kyanoviridae*, so these taxas were counted twice in the original figure. We apologize for this issue, and we have revised this figure to report the unique taxon for each sequence. In addition, the original geNomad analysis was done with the default setting which filter out sequences with virus score below 0.7, and we have rerun it with "--relaxed" flag to include all results.

In the revised manuscript, we have reported the virus scores for all the 1024 generated sequences (**Fig. R8**, from aggregated_classification.tsv file in the geNomad output) and the predicted host for 873 sequences (**Fig. R9**, from the virus_summary.tsv file in the geNomad output). We noticed that the virus scores for all the generated sequences range from 0.08 to 0.97, while geNomad provides the predicted taxonomy for sequences with virus scores higher than 0.41. This explains the difference in the number of sequences presented in the two figures.

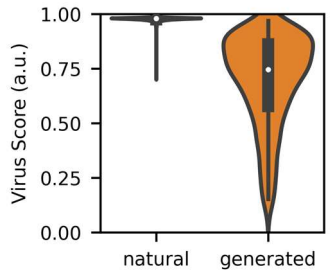


**Figure R8. Comparison of the predicted virus scores for all generated sequences and the training dataset.** This figure is labeled as **Figure 2c** in the revised manuscript.

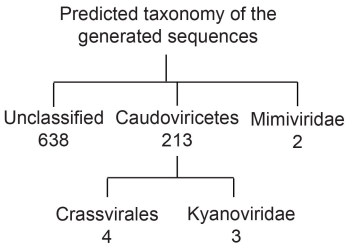


**Figure R9. Predicted taxonomy for the generated sequences predicted as viral.** Only taxonomies with > 1 sequence are shown. This figure is labeled as **Figure 2d** in the revised manuscript.

- Line 252: Correct “gens” to “genes”

Thanks. We have fixed this error.

- Line 285: Please use “dimensional” instead of “dim”

We have revised our text to replace 'dim' with 'dimensional' as suggested.

- Line 305-306: ROC and AUROC are previously defined on lines 259-260.

We have deleted the redundant definition of AUROC.

- Line 316: Please clarify that the geNomad results were run with default parameters

We thank the reviewer for pointing this out. We have rerun geNomad with the “--relax” flag to report results for sequences with virus score lower than 0.7. We have clarified this point in the revised manuscript.

**Revised text** (Section "Methods- Analysis of the generated sequence"):

geNomad[15] was used for sequence annotation of all generated sequences with default parameters and the "—relaxed" flag (version 1.6.1).

- Lines 329-331: Please list the extent of representation for each of the three datasets in this section (it's included in the supplementary figure 1)

Following the reviewer's suggestion, we have listed the extent of representation for three datasets.

**Revised text** (Section "Data availability"):

The bacteriophage genomes were downloaded from public databases including NCBI GenBank (ftp://ftp.ncbi.nlm.nih.gov/genomes/genbank/, n = 16,609), MGV (https://portal.nersc.gov/MGV, n = 53,032), and GPD (https://www.sanger.ac.uk/data/gut-phage-database/, n = 30,032).

- Figure 1C: This is a great analysis. In the figure caption, can you clarify the step size for the moving average?

We thank the reviewer for the positive comment. We have included the step size in the revised figure legend, which is 50 bp.

- Figure 2: I have several comments for this figure:
- Within the figure captions, the use of external tools to derive values (e.g., geNomad for panel C and Promoter Calculator for panel E) should be clarified in all instances.

We thank the reviewer for this constructive feedback. We have included the names of external tools in the revised figure legend.

- Panels B, F, and H should have statistical tests run to test if the distributions are significantly different. For example, the Kolmogorov-Smirnov test can be used.

We have included the two-sided Kolmogorov-Smirnov test results in the figure legends of Fig. 2B, F and H, In line with the reviewer's suggestion.

- Panels F: Other than for analytical ease, what is the justification for limiting the analysis in panel F to just sequence #87? Panel G is essentially replicated for all sequences in supplementary figure 10. I would expect that these trends would be consistent across generated sequences. An interesting analysis would be whether the difference in medians for random vs generated promoters correlates with the predicted virus scores.

We thank the reviewer for this insightful comment. We have reported the promoter activities for all generated sequences in **Fig. R10**. Notably, we observed a similar trend that the predicted promoters in

the generated sequences have higher transcriptional activity than random sequences of the same length. In addition, we have calculated the difference in medians for the generated promoters and random sequences and compared them with the virus scores (**Fig. R10**). We found a low correlation between the two, which may indicate that the model learns these two sequence features independently.

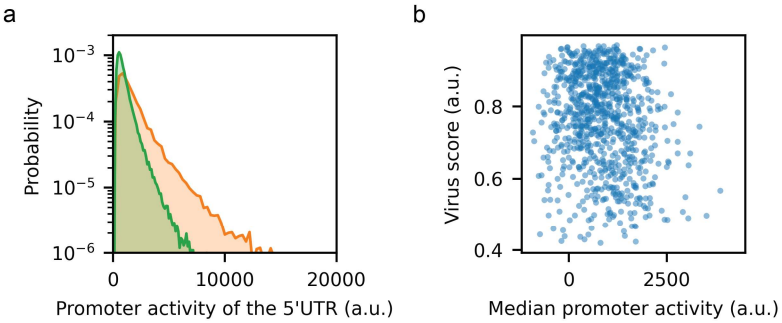


**Figure R10. Predicted promoter activity for 5'UTRs in generated sequence and its relationship to the virus score. a**) Histograms showing the distribution of predicted promoter activities for 5'UTRs in all generated sequences (n = 49,931, orange) and for an equal number of random sequences with the same length (green). Promoter activities were calculated using the Promoter Calculator[22]. **b**) Correlation of the promoter activity and the virus score for the generated sequences. The promoter activity is reported as the difference in medians for random sequences and generated promoters. Each dot represents one generated sequence, and the Spearman correlation coefficient is -0.15. This figure is labeled as **Supplementary Figure 12** in the revised manuscript.

**Revised text** (Section "megaDNA generates de novo genomic sequences"):

> This trend of A/G enrichment and high promoter activity within the 5'UTRs is also consistent across all the generated sequences (Supplementary Fig. 12 and 13).

> We found a low correlation between the promoter activities and virus scores, which may indicate that the model learns the two features independently (Supplementary Fig. 12).

- Supplementary Figure 5A: Please sort X values as ([-80,0],[0,80],[-160,0],[0,160],[-160,160]) so there is relative consistency in the input window length going from low to high.

We have revised Supplementary Fig. 5A following the reviewer's advice (**Fig. R11**).

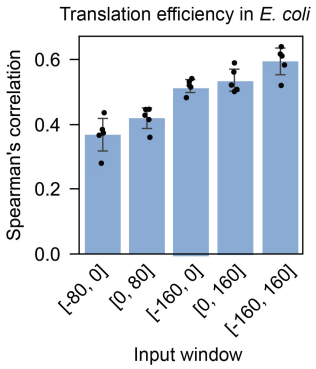


**Figure R11. Impact of the input window position for the prediction of translation efficiency of endogenous genes in *E. coli*.** Positions are reported relative to the start codons. This figure is labeled as **Supplementary Figure 7a** in the revised manuscript.

- Supplementary Figures 6 and 7 can be combined into a single figure as they are the same analyses just using different layers. In lines 85-86, can the author please comment on their hypothesis for the variation in accuracy across model layers - particularly local vs middle?

We would like to thank the reviewer for this helpful suggestion. We have combined the original supplementary Figure 6 and 7 into a single supplementary figure (**Fig. R12**). We assume that the different performance of the middle vs local layer is due to the local layer's shorter context window (16 bp). Furthermore, the global layer focuses on long-range information and may not provide sufficient resolution to identify specific local sequence features. In contrast, the middle layer's context window offers an optimal balance of length and resolution, which may explain its superior performance. We have also updated the manuscript to clarify this point.

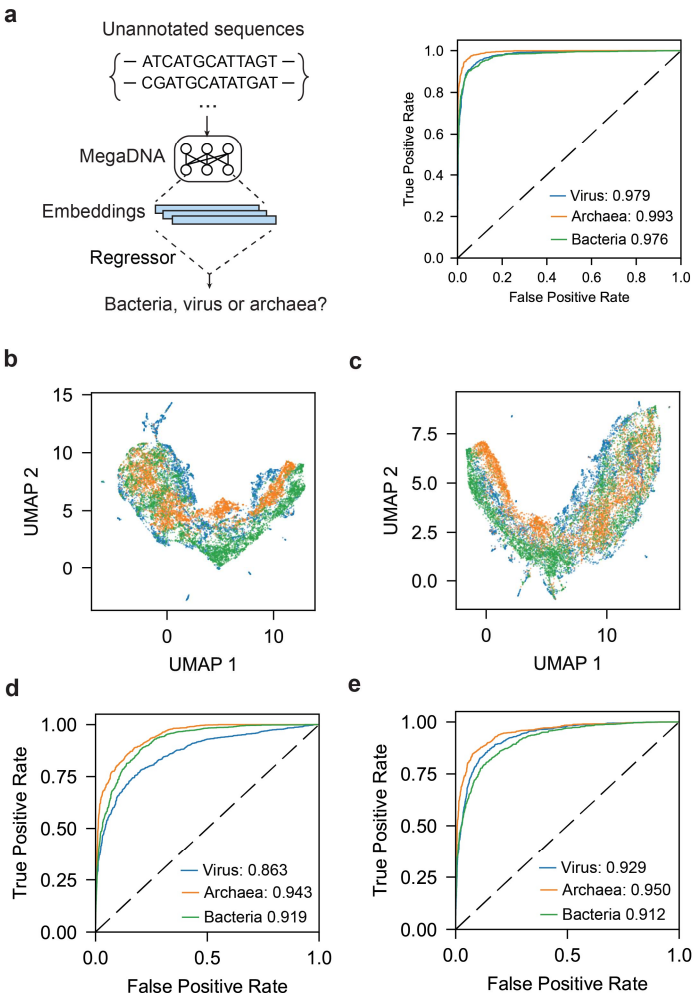


**Figure R12. Taxonomy prediction of unannotated sequences. a**) Embeddings from the middle layer was used to classify sequences into virus, bacteria, and archaea. Visualization of sequence embeddings from the local layer (**b**) and global layer (**c**), and their taxonomy prediction performances (**d**) and (**e**) are shown. For a), d) and e), the model's performance was assessed using 5-fold Stratified K-Fold cross-validation tests. The receiver operating characteristic (ROC) curves are shown (n = 5,000 for each category). The mean ROC curve (AUROC) scores from 5-fold cross-validation tests are reported. This figure is labeled as **Supplementary Figure 8** in the revised manuscript.

**Revised text** (Section "megaDNA learns functional properties of proteins and regulatory elements"):

> The prediction performance of the embeddings from the local and global layer was slightly lower compared to the middle layer. This difference may result from the local layer's short context window and the global layer's limited resolution for local sequence features. In contrast, the middle layer's context window provides an optimal balance of length and resolution, enabling effective distinction of sequences from different domains.

- Supplementary Figure 8: The conclusions from this figure are hard to evaluate given the sequence length differences between generated sequences and natural sequences. Can you please add a second panel that depicts the ratio of predicted genes to sequence length for natural vs generated sequences? I am envisioning two box-plots or something similar.

We have added a second panel that shows the density of the predicted genes in generated and natural sequences, which is defined as the ratio of the number of predicted genes to sequence length (**Fig. R13**). We found that the natural sequences and generated sequences show similar gene density (median value $1.57 \times 10^{-3}$ for generated sequences and $1.44 \times 10^{-3}$ for natural sequences).

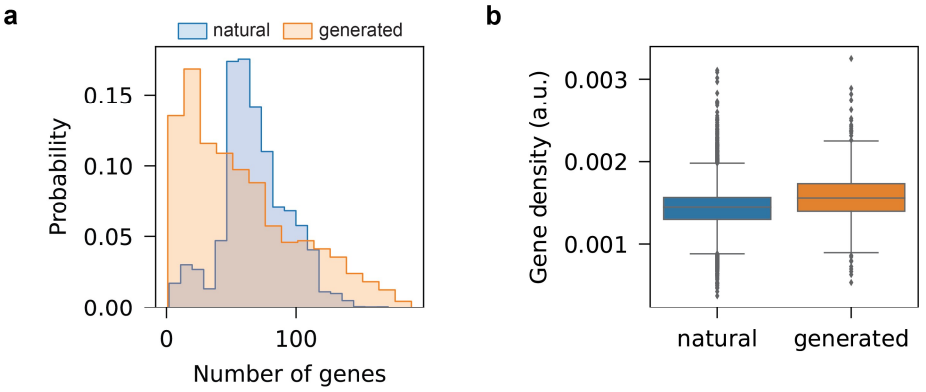


**Figure R13. Predicted gene numbers and densities for the generated sequences and the training dataset. a**) Comparison of the number of predicted genes in generated sequences (n = 607) versus those in the training dataset (n = 99,673). **b**) Gene density distributions for the generated and training sequences. Gene density is defined as the ratio of gene numbers and the sequence length. The central line inside the box represents the median value. The top and bottom borders of the box represent the third (upper) and first (lower) quartiles, respectively. This figure is labeled as **Supplementary Figure 10** in the revised manuscript.

**Revised text** (Section "megaDNA generates de novo genomic sequences"):

> In addition, the gene densities of the generated and training sequences are close with each other (Supplementary Fig. 10).

Reviewer #6 (Remarks on code availability):

I have successfully executed the code on https://github.com/lingxusb/megaDNA to generate synthetic bacteriophage sequences on both Google Colab and a high-performance computing cluster.

We thank the reviewer for the thorough test of our codes.

**Reviewer #3** (Remarks to the Author):

I thank the authors for their thorough response and manuscript revision. All my previous comments were addressed, and I have no further concern at this time.

We thank the reviewer for the positive comments on our revised manuscript.

**Reviewer #4** (Remarks to the Author):

I believe the authors have successfully addressed the major comments on the paper, resulting in a much improved manuscript that better demonstrates the robustness and potential of their method. I have a few minor suggestions:

We appreciate the reviewer's positive comments and insightful suggestions.

- While a whole paragraph of the manuscript is dedicated to describing the individual genes encoded by the generated genomes, no analysis was performed to evaluate whether the entire set of genes within a given genome is coherent. This would demonstrate that the model has learned the components that make up a phage. A simple check would be to evaluate the fraction of generated genomes containing proteins annotated as capsid, terminase, or portal, since such proteins are expected in all phage genomes.

We thank the reviewer for this helpful comment. Unfortunately, we didn't find a generated genome containing the entire set of genes including capsid, terminase and portal protein, within a limited number of generated sequences. This may be due to the fact that the generated proteins have little homolog with the natural bacteriophage proteins (Supplementary Table 1). As the reviewer pointed out in the previous revision, we are still far away from the design of whole functional genomes, and we acknowledge that further improvements to our model would be required to generate coherent and functional gene sets.

- Lines 182, 183: The alignments are so small that it is unclear whether they correspond to the same "underlying gene family." Could you verify if the generated genes and their corresponding matches have the same annotations?

To address this question, we have checked the annotations for both the query genes and their matches. The generated sequence #202, gene #97 was annotated as a phage-related minor tail protein (COG5280) by geNomad, while its corresponding match in bacteriophage genome (MGV-GENOME-0232742) was annotated as a phage-related protein (COG5412). The remaining genes were not functionally annotated by either geNomad or phold.

- Lines 184-189: Although little homology was found between generated and “natural” proteins at the amino acid level, it would be interesting to see if the generated proteins have reasonable structures. This could be easily evaluated by aligning the predicted structures of generated proteins to experimental structures in the PDB using Reseek/Foldseek. Note that phold doesn’t use structures directly, only a structure-informed representation of the protein. This would serve as additional confirmation that the model has learned biologically relevant features.

We would like to thank the reviewer for this insightful suggestion. We have provided a new figure illustrating an example of the alignment of the predicted structure of a generated protein with experimental structures in the PDB database (**Fig. R1**). This result provides further evidence of the model's ability to capture biologically relevant features.

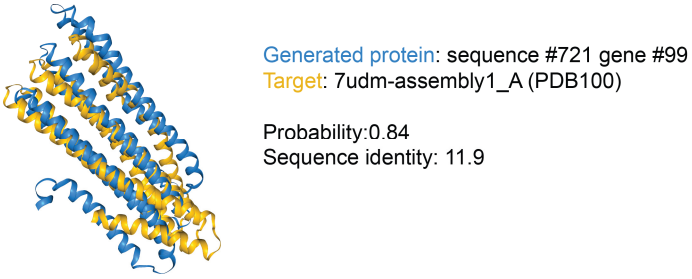


**Figure R1. Alignment of a generated protein to the PDB database using Foldseek.** The structure of sequence #721 gene #99 was predicted using ESMfold with default parameters. The predicted structure was then searched against the PDB100 database via the Foldseek online server in 3Di/AA mode (https://search.foldseek.com/search). This figure is labeled as **Supplementary Figure 15b** in the revised manuscript.

**Revised text** (Section “megaDNA generates de novo genomic sequences”):

> Moreover, the predicted structure of a generated protein aligns with the experimental structure in the PDB database, further demonstrating the model's ability to capture biologically relevant features (Supplementary Fig. 15).

- Lines 417-419: There is a repetition of “genomes of bacteriophage, bacteria, and archaea.”

We thank the reviewer for careful reading of our manuscript. We have fixed this error.

**Reviewer #5** (Remarks to the Author):

Dear authors,

You have done an excellent job clarifying some of my initial confusions. While I unfortunately still feel like I understand only 70-80% of the manuscript, I now feel that is mostly my lack of knowledge in this particular field, and not reflective of the quality of the manuscript. Although this makes it hard for me to

wholeheartedly recommend the manuscript, I am certainly in favour of publication as I assume that other (perhaps more suited reviewers) can cover for my lack of expertise.

I wish you all the best!

We appreciate the reviewer's thoughtful feedback on our revised manuscript. We are committed to further developing our model and making it accessible to a broader audience.

**Reviewer #6** (Remarks to the Author):

I commend the author for the substantial effort in revising this manuscript. The changes made assuage all concerns that I had noted within my initial review.

Reviewer #6 (Remarks on code availability):

I have reviewed the code and successfully executed in in Google Colab, locally, and in an HPC.

We would like to express our gratitude to the reviewer for the positive comments on our revised manuscript and software.