Generative model for Pseudomonad genomes

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent advances in genomic sequencing have resulted in several thousands of full genomes of pseudomonads, a genera of bacteria important in many science areas ranging from biogeochemical cycling in the environment to bacterial pneumonia in humans. With these high-quality data sets, combined with tens of thousands of somewhat lower quality metagenomically assembled genomes, we create a generative model for pseudomonad genomes. We present a GAN model that generates gene family presence absence list for a genome. We also demonstrate that the discriminator of this model can be used as a binary classifier to identify incorrect genomes. In the future, our desired model can be used to generate genomes within a given set of parameters such as, "Generate a genome that is root associated, drought resistant, salt tolerant that will produce this natural product".

1 Introduction

Most of us have heard that traits are passed from parents to children in the form of genes. Each gene contains information about a certain trait and together, they encode all the traits of an organism. For example a gene might code for eye color[6]. All the DNA contained in one cell is called a genome. With the recent advances in genome sequencing[10], larger datasets and novel generative modeling techniques, the research problem we are trying to work towards is: what if we can generate a genome with desired combination of traits.

This paper is an initial step towards our desired model. Here, we present a generative model which helps in identifying incorrect genomes and generates a list of genes present absent in a genome. We train our model with genomes of pseudonomad bacteria which are an important genera in many science areas with an abundance of fully sequenced genome data.

2 Methods

Genus level Pseudomonas genome data is sourced and downloaded from Genome Taxonomy Database (GTDB)[11], licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. We use their fna files of each genome which contain nucleotide sequence (DNA) information for the model. The analysis in this study is done on 3000 genomes.

2.1 Pangenome Analysis

The first step towards building the model is to perform a pangenome analysis. A pangenome is the entire set of genes from all strains within a group. It can be understood as the union of all the genomes of a group. The pangenome can be broken down into a "core pangenome" that contains genes present in all genomes, a "shell pangenome" that contains genes present in two or more strains, and a "cloud pangenome" that contains genes only found in a single strain [2, 3, 4]. Pangenome

Learning Meaningful Representations of Life (NeurIPS 2022).

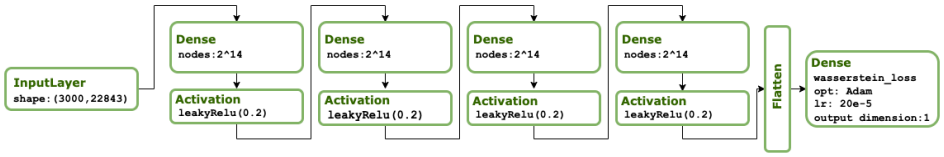
analysis helps in understanding the genetic determinants of biological activity. To perform pangenome analysis we use PPanGGOLiN [1], a Free Software suite used to create and manipulate prokaryotic pangenomes from a set of genomic DNA sequences. We run the ppanggolin workflow by providing a tsv-separated file, a list with first column being a unique genome name and the second column being its path to the associated FASTA file(compressed fna file). The PPanGGoLin analysis on 3000 genomes was run using 128 AMD EPYC 7763 CPUs, around 450GB of memory was utilized and took 13 hours to complete. PPanGGOLiN provides multiple outputs to describe a pangenome. For the current study, we use the core, shell, cloud genes lists from the output partition folder and gene_presence_absence.Rtab file.

2.2 Data Input

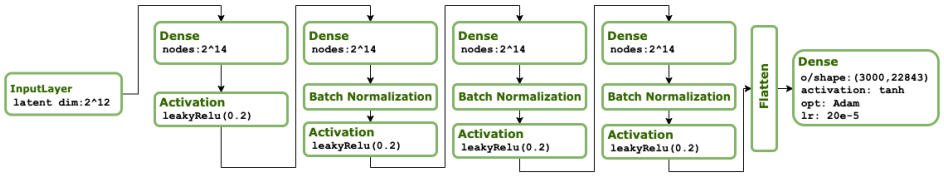
The data input to the model is a presence absence matrix extracted from gene_presence_absence.Rtab file of PPanGGOLiN ouput, where the columns are the genomes used to build the pangenome, the rows are the gene families. There is a 1 if the gene family is present in a genome, and 0 otherwise. For the current analysis we are interested in finding the genes which are present in two or more genomes, and hence we filter the presence absence matrix to contain only the core genes (genes present in all the genomes) and shell genes (genes present in 10-95% of the genomes) and remove the cloud genes (genes less than 10% occurrence). This matrix data to the model can be described as a 2D tensor with genomes and their corresponding genes present/absent.

2.3 Model

To build the generative model we use Wasserstein GAN-Gradient Penalty method. Generative adversarial networks (GANs) [7] are a powerful class of generative modeling subjectively regarded as producing better samples than other methods [8]. GANs have two neural networks playing against each other where one, the generator learns to generate reasonable data as training and other, the discriminator learns to distinguish the generator's fake data from real data. The generated instances become negative training examples for the discriminator. Instead of using a discriminator to classify or predict the probability of generated data as being real or fake, we use WGAN which changes the discriminator model with a critic that scores the realness or fakeness of a given data using Wasserstein loss [5]. The WGAN gradient penalty method [9] is used in order to ensure model convergence and stability. The network architecture is designed based on the recommendations dealing with optimization using larger matrix data [12] with a sequence of densely connected layers having 2^{14} neurons in each layer, around 70% of the input genes size (22843 core + shell genes), generator input latent vector of size 2^{12} shown in Fig.1. The input dataset is a gene presence absence matrix of 3000



a) Critic



(b) Generator

Figure 1: WGAN Model for Pseudomonad genomes

genomes with 22843 core and shell genes. We train on randomly selected 80% of the dataset with a batch size of 64. The Adam optimizer with a learning rate of 0.0002, beta_1 of 0.5 and beta_2 of 0.9 are utilized. In the WGAN model, as the critic model must be updated more than the generator

model, we update it 3 times more than the generator. The model is trained for 50 epochs on AMD EPYC 7763 CPUs.

3 Results

To evaluate the performance of the models, we first test the generator produced genomes for the percentage of core genes present in their gene presence absence lists. These gene presence absence lists of the genomes produced by the generator have a median of 73% of the required core genes which can be shown in a clustermap in Fig. 2, with number of genomes on the y-axis and core genes on the x-axis. Next, we test the critic model on unseen test genomes, which are the 20% data split before training. The critic model predicted 598 genomes as real among 600 test genomes showing 99% accuracy. When we evaluate the critic using synthetic incorrect genomes created by removing the core genes, the model was only able to rightly predict 1 as incorrect out of the 4 incorrect genomes. This result on the incorrect genomes is obvious as the critic was trained only on the fake data generated by the generator but was not pre-trained with the fake data from the input source. Hence, we plan to further train the model with a combination of real and incorrect genomes and test the critic on a larger number of incorrect genomes. From these observations we can say that the model has performed fairly well on generating gene presence absence matrices with 73% accuracy and the critic has identified 99% of the real genomes.

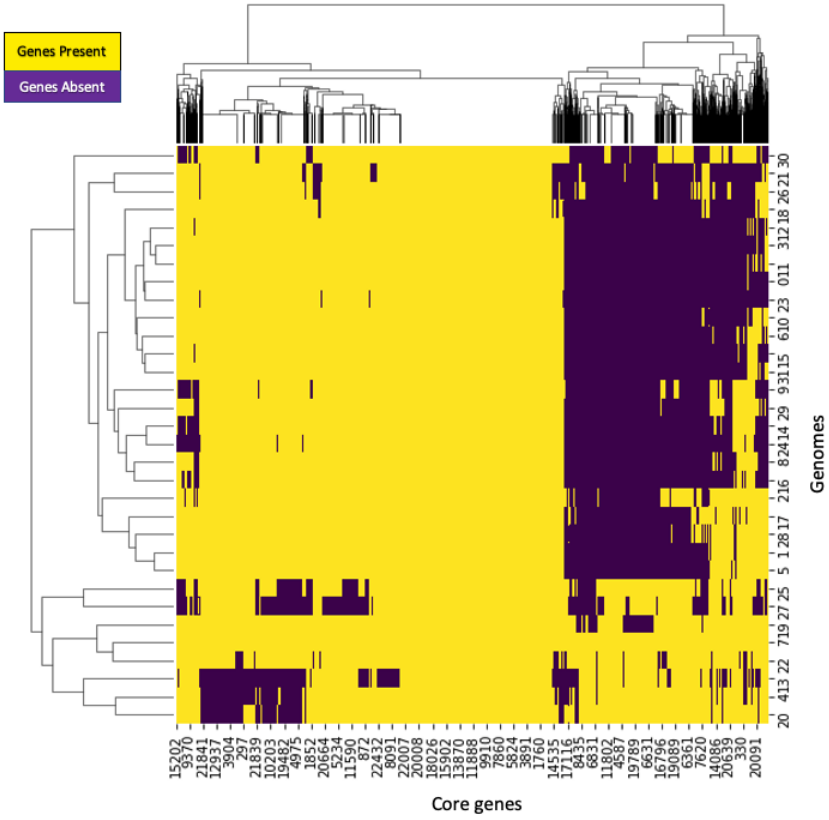


Figure 2: Generator ouptut evaluation

4 Conclusion

In this work, we present the pangenome data of Pseudomonad bacteria, a generative model to identify incorrect genomes and generate gene family presence absence list for a genome. This research contributes to finding incomplete gene sequences as well as the study of gene functions and positional information by introducing a novel method for tackling this problem. Given the strong

performance of our current model using only gene presence absence, we believe incorporation of biologically significant gene level features, such as gene functions and position; multi gene modules or operons that function together; and organism level trait annotations, will make the generative model significantly more informative.

References

[1] Gautreau G et al. "PPanGGOLiN: Depicting microbial diversity via a partitioned pangenome graph." In: *PLOS Computational Biology 16(3): e1007732*. (2021). DOI: 10.1371/journal.pcbi.1007732.

[2] Medini D et al. "The microbial pan-genome". In: *Current Opinion in Genetics Development* 15 (2005). DOI: 10.1016/j.gde.2005.09.006.

[3] Vernikos G et al. "Ten years of pan-genome analyses". In: *Current Opinion in Microbiology* 23 (2015), pp. 148–154. DOI: 10.1016/j.mib.2014.11.016.

[4] Wolf Y.I et al. "Updated clusters of orthologous genes for Archaea: a complex ancestor of the Archaea and the byways of horizontal gene transfer." In: *Biology direct* 7 46. (2012). DOI: 10.1186/1745-6150-7-46.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein Generative Adversarial Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. 2017.

[6] Cooper GM. *The Cell: A Molecular Approach. 2nd edition.* 2000. DOI: https://www.ncbi.nlm.nih.gov/books/NBK9944/.

[7] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Vol. 27. 2014. URL: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[8] Ian J. Goodfellow. "NIPS 2016 Tutorial: Generative Adversarial Networks". In: *ArXiv* abs/1701.00160 (2017). DOI: 10.48550/arXiv.1701.00160.

[9] Ishaan Gulrajani et al. "Improved Training of Wasserstein GANs". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6-Paper.pdf.

[10] James M Heather and Benjamin Chain. "The sequence of sequencers: The history of sequencing DNA." In: *Genomics* 107,1 (2016). DOI: 10.1016/j.ygeno.2015.11.003.

[11] Donovan Parks et al. "GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy". In: *Nucleic Acids Research* 50 (2021). DOI: 10.1093/nar/gkab776.

[12] Vincent Vanhoucke, Andrew Senior, and Mark Z. Mao. "Improving the speed of neural networks on CPUs". In: *Deep Learning and Unsupervised Feature Learning Workshop, NIPS 2011*.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] There are no potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] There are no theoretical results.
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The gene_presence_absence.Rtab file, input to the model for 3000 genomes has a size of around 4GB. Github does not support larger files upload. In regards to uploading through other websites, we are not certain that anonymity could be maintained. We will release the data generated and code for the camera-ready version.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] Used the existing software of PPanGGOLiN, downloaded data from GTDB and cited them.

(b) Did you mention the license of the assets? [Yes]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A] We have not included any new assets in the supplemental material or as a URL

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] PPanGGOLiN is a open source CeCiLL-licensed software. The genome data downloaded from GTDB is a Creative Commons Attribution-ShareAlike 4.0 International License.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We have not used crowdsourcing or conducted research with human subjects

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]