CS230

Learning DNA Encodings For Metagenome Binning (Healthcare)

Peter C. McCaffrey
pemccaff@stanford.edu

Abstract

Metagenome binning is a critical component of metagenomics where sequencing data consisting of short fragments sampled from entire ecosystems are grouped according to their shared source genome. In this project, I perform such binning by encoding such sequencing data in a lower-dimensional space and demonstrate improvement of this process using disentanglement across a variety of experiments representing homogeneous and heterogeneous experimental metagenome samples.

1 Introduction

Metagenomics is the study of multi-organism ecosystems through genomic analysis. A very common technique in metagenomics is known as "metagenome shotgun sequencing" wherein a complex sample such as soil or human stool undergoes whole DNA extraction and sequencing. Unfortunately,this process requires that DNA be physically sheared into very small pieces in order to be compatible with sequencing machinery, resulting in a data set consisting of millions of 150-250 base pair sequences sampled from hundreds or even thousands of separate original genomes. Thus, shotgun sequencing faces many serious challenges in terms of reconstructing "short-read" data generated by sequencing machinery back into a collection of original source genomes. Key to this process is a step known as "binning" wherein these very short sequences, after being grouped into longer genomic segments of tens to hundreds of thousands of base pairs, are clustered according to a shared source genome. Without binning, it is not possible to precisely identify the organisms captured by a metagenomics sample and it is not possible to observe meaningful features that take place overlarge genomic regions such as rearrangements or gene transfer. In this project, I focus on using and improving upon the use of variational autoencoders to perform metagenome binning. This has a few advantages in that binning can be more accurate, in certain settings, and more computationally efficient as binning occurs in a lower dimensional encoding space.

2 Data Inputs and Outputs

The input to this model consists of raw sequencing data files which undergo pre-processing steps to include assembly into larger genomic regions called "contigs". These contigs are featurized through quantitation of tetranucleotide frequency to generate 103-dimensional vectors wherein each dimension represents the frequency of a given 4-nucleotide window.

The final outputs of this model are a collection of genomic bins which consist of groups of contigs whose encoding vectors were near to each other in encoding space. This bin-level grouping allows for all source DNA contigs that the model assigns as belonging to the same source genome to be gathered and compared against known or input source genomes to determine precision and recall.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

3 Related Work

The problem of metagenome binning has been–and remains–a long-standing and open challenge of the field. Many tools have been developed to address this challenge. Prominent example such as MaxBin and CONCOCT rely upon assessing the shared frequency of nucleotide substrings between genome contigs, the abundance of sequencing coverage between contigs and the presence of marker genes while others such as DASTool intake bins produce by another upstream tool such as MaxBin and re-shuffle contigs between bins. A more comprehensive comparison of such binning algorithms is provided in Appendix 8.2 as published recently in Yue et al. [7]. Most recently, VAMB [2] has emerged as a tool for binning that utilizes an autoencoder to compress genome contigs and then performs clustering in the latent space. This has the advantage of being computationally much more efficient than other tools as the latent space is much smaller than the input space. However, performance of this approach is likely to vary with the complexity and diversity of the input genome sample. This has not been addressed in the setting of VAMB which is currently the only tool for binning that utilizes such an encoding strategy. Moreover, the recently published work behind VAMB does not explore strategies for hyperparameter optimization in the setting if sample complexity nor does it explore the use of disentanglement as a potential strategy to cope more homogeneous samples.

4 Model Workflow and Strategy

4.1 Data Processing Pipeline

As with many genomic applications, there are several pre-processing steps required for model deployment starting from raw genome sequencing data and resulting in binning assignments of those sequencing data to a presumed common source genome. The workflow and its constituent steps are outlined in the Figure below:

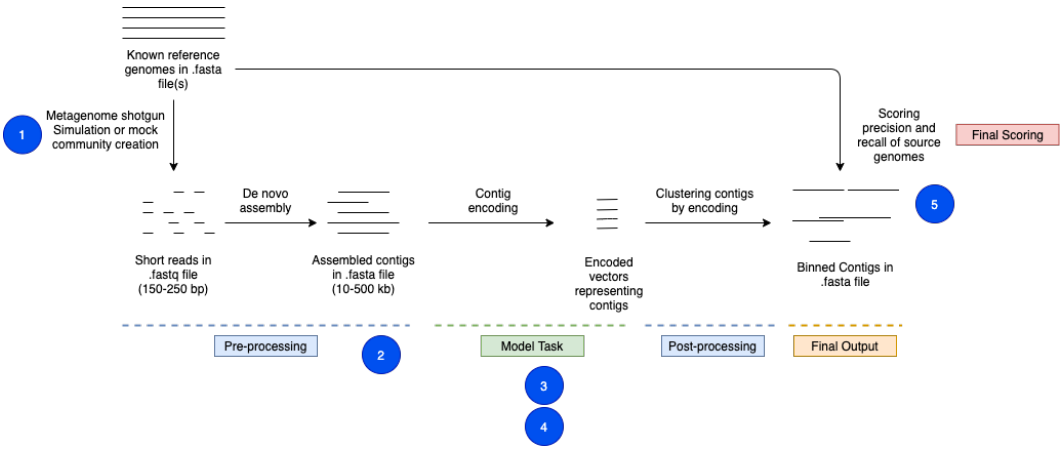


Figure 1: Schematic depicting data pre-processing, model encoding, post-processing, and scoring. Numbers map to work items enumerated below

1. Acquisition of input genomic data representing metaenome samples with known input genomes.
2. Implementation of genome featurization steps using read mapping, contig creation, and tetranucleotide frequency calculation.
3. Implementation of a baseline encoder without disentanglement. This uses the source code from VAMB and Nissen et al. [2] which has been imported into this project
4. Implementation of a modified encoder model containing a beta parameter.
5. Implementation of final output scoring of genome bin assignments.

As an important note, the role of the autoencoder is to convert features in genome space into encoded feature vectors in latent space. Once these vectors have been produced, there is still one encoded

vector for each input contig and clustering of these features into genomic bins is performed using an iterative medoid clustering algorithm as described by Nissel et al. and implemented here using the VAMB tool set from the same authors.

4.2 Model Scoring

The overall workflow consists of two general optimization goals. The first such goal, referred to below as the "Autoencoder Scoring Method" is for the autoencoder itself and seeks to optimize for reconstruction loss between vectors of tetranucleotide frequencies when encoded through a narrow bottleneck. For this work, hyperparameter tuning considered both 32-dimensional and 64-dimensional latent spaces, the results of such tuning are described in Section 4: Hyperparameter Tuning.

Importantly, each resulting encoding vector still represents an input contig and so, when binning is performed via clustering encoding vectors in encoding space, those vectors which are assigned the same source genome are then mapped back to their input contigs. These contigs and their bin assignments are evaluated using lineage and taxonomy-specific marker genes via the CheckM tool which calculates the completeness and contamination (akin to recall and precision) of these assignments. This is referred to below as the "Binning Scoring Method" as this does represent the ultimate functional goal of using an encoder to perform binning in practice.

With regard to modeling in particular, the focus of my work is at the point of the autoencoder model wherein I seek to improve upon a base implementation of a VAE for contig encoding through the use of disentanglement. Implementing a VAE with a beta parameter and modifying that as a hyperparameter is the core focus of this work with the assumption that disentanglement will improve binning by creating more orthogonal encoding dimensions and, hence, improving separability between contigs especially in situations where the originating sample consists of similar source genomes.

4.2.1 Autoencoder Scoring Method

The autoencoder model calculates several basic scoring parameters before combining them into a final output loss. Initially, the model calculates the sum of the squared error reconstruction losses between both tetranucleotide frequency values (tnfs) and read abundance (depths), as shown in equations (1) and (2) below:

$$(1)\ Loss_{tnfs} = \frac{(D_{in} - D_{out})^2}{N} \qquad (2)\ Loss_{depths} = \frac{\sum_{n=1}^{t}(T_{in}^{t} - T_{out}^{t})^2}{N}$$

Where N is the number of input contigs, t is the number of dimensions in tetranucleotide space (here t=103), T is a vector representing the abundances of a specific tetranucleotide across all contigs, and D is a vector representing the abundance of each contig in the sample as calculated by Reads per Kilobase Mapped (RPKM).

Following this, the KL Divergence, D_{KL}, is calculated in the method described and implemented by Higgins et al. [4]:

$$D_{KL}(Latent|Prior) = -0.5 * \sum(1 + ln(\sigma) - \mu^2 - \sigma)$$

Where Dim_{latent} is the dimensionality of the latent space, tuned to 32 for the results shown in Section 5.

The final loss is then the weighted sum of losses for depths, tnfs, and KL divergence with β acting as a coefficient for D_{KL}:

$$Loss_{final} = (\frac{\alpha}{t} * Loss_{depths}) + ((1 - \alpha) * Loss_{tnfs}) + (\beta * D_{KL})$$

Importantly, as the β parameter is increased, this increases the impact of D_{KL} on $Loss_{final}$

4.2.2 Binning Scoring Method

As a note on how the final scores of completeness and contamination are calculated in this case, we are using the approach described in the CheckM manuscript [6]. Contamination is estimated from the number of multi copy marker genes identified in each marker set

$$\frac{\sum_{s \in M} \frac{|\sum_{g \in s} C_g|}{|s|}}{|M|}$$

where s is a set of collocated marker genes; M is the set of all collocated marker sets s; C_g is $N - 1$ for a gene g identified $N >= 1$ times, and 0 for a missing gene.

5 Hyperparameter Tuning

Hyperparameters for this model were evaluated using Bayesian optimization across a range of values for learning rate, dropout, alpha, beta (as a cost parameter for the encoder network), and latent space dimensionality with results presented in **Figure 2** . In summary, this optimization was carried out using a development data set representing airway metagenomes from the CAMISIM project and demonstrated a reduction in loss with increasing beta with the best-performing models having a beta of 800. Dropout, latent space dimensionality, and learning rate were similar in importance and comparatively less important than beta but optimal settings for these were 0.2, 32, and 0.001 respectively based upon the best performing models. Since training requires iteration over tens of thousands of contigs and requires up to 3 hours of GPU time, these executions were performed in a long-running job with hyperparameter values and model performance being captured using the weights and biases wandb package (wandb.com) for model telemetry.

Importantly, loss for this model is calculated using the Autoencoder Scoring Method described above which captures the encoding and reconstruction performance of the model as it processes contigs. Hyperparameter tuning was not performed with the Binning Scoring Method because this would not reflect how this model would be used in practice. When performing metagenome binning, a given sample's contigs would be generated and encoded but the source genomes would not be known and, hence, could not be used to calculate contamination. For this work, hyperparameter tuning was used to identify optimal hyperparameters and to establish the relevance of the beta parameter in particular. In the Final Results section, these optimal parameters are used along with a range of beta values to bin a variety of metagenome samples of varying complexity.

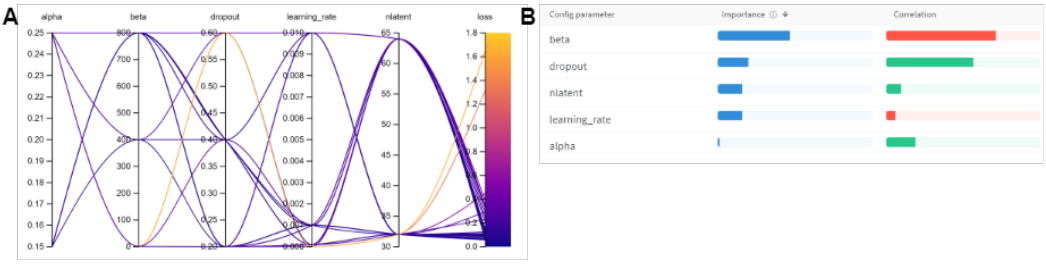


Figure 2: **A** Summary of 56 training runs assessing encoder reconstruction loss for varying values of alpha, beta, dropout, learning rate, and latent dimensions. Paths connect hyperparameters chosen for a single run. Path colors reflect the average loss for a given combination of hyperparameters. **B** Summary of purity-based hyperparameter importance calculated using a random forest classifier trained to predict loss.

6 Results

With hyperparameters tuned, the beta variational autoencoder was tested using β values of 1 (corresponding to the model as published by Nissel et al.), 5, 50, 200, 400, and 800 each using using 150 training epochs in the encoder model. Moreover, these experiments were carried out across

a collection of simulated metagenomes representing varying levels of phylogenetic similarity and complexity. These included combinations of 25 and 250 input genomes representing 1, or 10 separate genera. This was done to test the impact of the β hyperparameter in resolving genome bins when samples consist of more similar versus more divergent source genomes.

These experiments reveal that increasing beta results in fewer contig bins and a lower contamination rate (i.e. a higher precision). Full results for each experiment are recorded in Appendix item 8.1 and losses during training for these experiments are provided in Appendix 8.3). The association between beta and contamination is depicted in **Figure 3**.

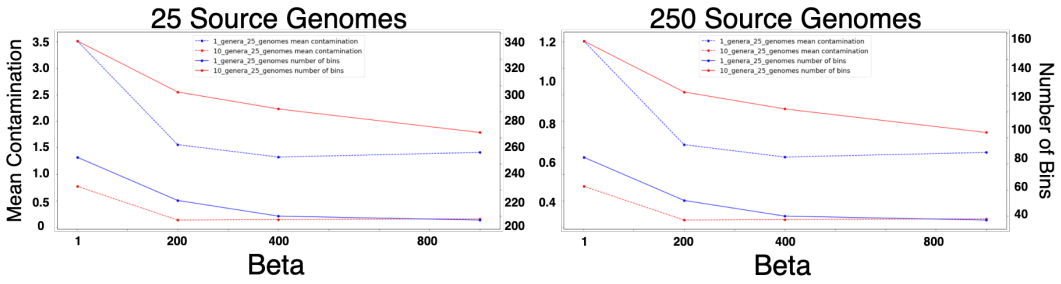


Figure 3: Relationship between β and the contamination of metagenome bins for simulated metagenomes derived from 25 and 250 source genomes samples from 1 or 10 source genera. Increasing values of beta reduce contamination

7 Discussion

As mentioned previously, metagenome binning is essentially a problem of re-grouping fragmentary samples of input data (in this case, source genomes) according to a shared sample source (in this case a single originating genome). Due to the nature of sequencing, this problem must be solved by identifying some signature shared between fragments sampled from the same source but not shared between fragments sampled from different sources. Moreover, previous attempts to solve this have invoked computationally intensive algorithms based upon pairwise comparison or marker gene detection. Embedding offers a speed up to accomplish this in a simpler, lower-dimensional space but at the cost of information content. Here, I extend initial work in using variational autoencoders for creating a lower-dimensional encoding space in which to perform binning by adding disentanglement as described in the β-VAE.

This modification enforces that the narrow bottleneck's encoding dimensions are less inter-related. Under the hypothesis that this will be useful in allowing the encoder to capture more granular distinctions between encoded contigs especially when source genomes are more related to each other, this work tested the impact of increasing the β parameter across a variety of metagenome samples ranging from fewer, less related source genomes to more abundant, more related source genomes. We would expect that making such granular distinctions between contigs would be especially important when binning a metagenome representing many related organisms. In the Results above, we note that increasing values of β proportionally reduce the contamination of recovered contigs while preserving the number of recovered contigs especially with high values such as 400 and 800 when binning a sample derived from 250 source genomes all samples from the same genus. Similarly, increasing β is also useful in metagenome samples with fewer source genomes (e.g. 25 genomes samples from the same genus) but this contamination effect plateaus at a value of 200 rather than 400 or 800 which would be expected since the metagenome sample–while still derived from related genomes–represents fewer source genomes and, therefore, should be easier to separate in encoding space than a more crowded sample with 250 genomes. Finally, this effect is far less pronounce with metagenomes sampled from more diverse genomes (e.g. 10 source genera rather than one source genus) which we would naturally expect to be easier to resolve in encoding space with less reliance on enforcing disentanglement.

8 Appendix

8.1 Simulated Genome Experiments and β Values

Beta	Num. Genera	Num. Source Genomes	Num. Bins	Avg Contamination
1	1	25	229	2.45
1	1	250	154	1.15
1	10	25	317	0.31
1	10	250	67	0.33
200	1	25	212	1.55
200	1	250	154	1.15
200	10	25	317	0.13
200	10	250	51	0.31
400	1	25	200	1.32
400	1	250	139	0.94
400	10	25	282	0.14
400	10	250	45	0.32
800	1	25	197	1.41
800	1	250	137	0.71
800	10	25	264	0.16
800	10	250	37	0.34

Table 1: Training Results using a latent space of 32 dimensions, learning rate = 0.001, epochs = 150, dropout = 0.2 with different values for $beta$

8.2 Comparison Table of Metagenome Binning Tools

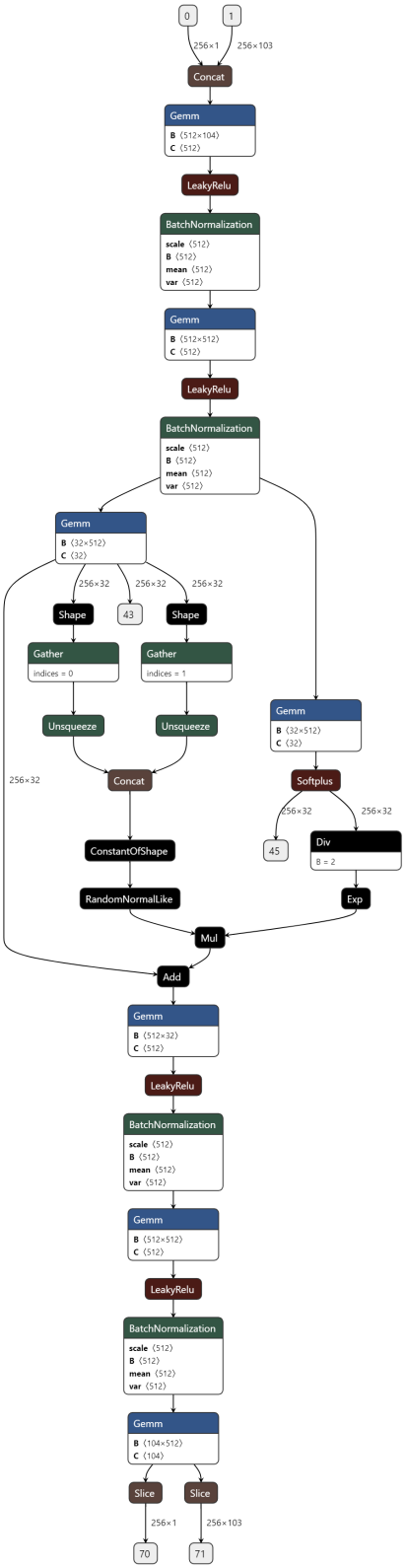
Genome binner	Parameters	Model	Version to validate	Publication	Last update	Resources
MaxBin	k-mer frequencies, coverage, single-copy genes	Expectation-maximization, bin number estimated from single-copy marker gene analysis	2.2.6	2014	2019	https://sourceforge.net/projects/maxbin
MetaBat	4-mer frequencies, coverage	Modified K-medoids algorithm	1&2.13	2015	2020	https://bitbucket.org/berkeleylab/metabat/src/master
Groopm	coverage, contig's length, tetranucleotide frequency	Two way clustering, Hough partitioning, self-organizing map	2	2014	2017	https://github.com/timbalam/GroopM
CONCOCT	k-mer frequencies, coverage	Gaussian mixture models, bin number determined by variable Bayesian	1.0.0	2014	2019	https://github.com/BinPro/CONCOCT
MyCC	k-mer frequencies, coverage (optional), universal single-copy genes	Affinity propagation	1	2016	2017	https://sourceforge.net/projects/sb2nhri
MetaWatt	tetranucleotide frequency, coverage	Firstly clustering by empirical relationship of the average standard deviation at tetranucleotide frequency mean, then employing interpolated Markov models	3.5.3	2012	2016	https://sourceforge.net/projects/metawatt
BMC3C	frequency variation of oligonucleotides, coverage, codon usage	Ensemble k-means, construct a weigh graph and partition it by Normalized cuts [49, 50]	\	2018	2018	http://mlda.swu.edu.cn/codes.php?name = BMC3C
Binsanity	coverage, tetranucleotide frequency, percent GC content	Affinity propagation	0.2.8	2017	2020	https://github.com/edgraham/BinSanity
Autometa	sequence homology, single-copy genes, 5-mer frequency, coverage, single-copy genes	Lowest common ancestor analysis, DBSCAN algorithm, supervised decision tree classifier recruite unclustered contigs	\	2019	2020	https://bitbucket.org/jason_c_kwan/autometa/src/master
COCACOLA	k-mer frequency, coverage, co-alignment, paired-end read linkage	K-means based on L1 distance, non-negative matrix factorization with sparse regularization, hierarchical clustering	\	2017	2017	https://github.com/younglululu/COCACOLA
SolidBin-naive	single-copy mark genes, tetranucleotide frequencies, coverage, pairwise constraints	Semi-supervised spectral Normalized cut	1.1	2019	2020	https://github.com/sufforest/SolidBin
Vamb	tetranucleotide frequencies, coverage	Variational autoencoders, iterative medoid clustering algorithm	2.0.1	2018	2020	https://github.com/RasmussenLab/vamb
DAS Tool	original binner output bin sets	Refine bins according shared contigs between two original binner results	1.1.1	2018	2019	https://github.com/cmks/DAS_Tool
MetaWrap	original binner output bin sets	Separating every pair of contigs in different bins, selecting the best bin sets according completion and contamination	1.2.2	2018	2019	https://github.com/bxlab/metaWRAP
Binning_refiner	original binner output bin sets, single-copy genes	Scoring bins based on single-copy genes and picking up high-score bins iteratively	1.4.0	2017	2019	https://github.com/songweizhi/Binning_refiner

Figure 4: Comparison of metagenome binning tools excerpted from Yue et al. [7]

8.3 Model Training Loss for β experiments

Figure 5: Model loss per training epoch for models trained in the assessment of β and its impact on various simulated metagenomes.

8.4 Model Graph



References

[1] Fritz, A., Hofmann, P., Majda, S. et al. CAMISIM: simulating metagenomes and microbial communities. *Microbiome* **7**, 17 (2019). https://doi.org/10.1186/s40168-019-0633-6

[2] Nissen, J.N., Johansen, J., Allesøe, R.L. et al. Improved metagenome binning and assembly using deep variational autoencoders. *Nat Biotechnol* (2021). https://doi.org/10.1038/s41587-020-00777-4

[3] Kislyuk, A., Bhatnagar, S., Dushoff, J. et al. Unsupervised statistical clustering of environmental shotgun sequences. *BMC Bioinformatics* **10**, 316 (2009). https://doi.org/10.1186/1471-2105-10-316

[4] Higgins, I., Matthey L., Pal A., Burgess C., Glorot X. et al. B-VAE: Learning Basic Visual Concepts With A Constrained Variational Framework. *ICLR* (2017). https://openreview.net/pdf?id=Sy2fzU9gl

[5] Burgess, P., Higgins, I., Pal, A. et al. Understanding Disentangling in B-VAE. *arXiv* (2018). arXiv:1804.03599.

[6] Parks, D., Imelfort, M., Skennerton, C.T., et al. CheckM: Assessing the Quality of Microbial Genomes Recovered from Isolates, Single Cells, and Metagenomes. *Genome Res* (2015). doi: 10.1101/gr.186072.114

[7] Yue, Y., Huang, H., Dou, H.M. et al. Evaluating metagenomics tools for genome binning with real metagenomic datasets and CAMI datasets.