RICE UNIVERSITY

GEM
Incorporating Context into Genomic Distance Estimation

by

CJ Barberan

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
Master's of Engineering

APPROVED, THESIS COMMITTEE:

Richard G. Baraniuk, Chair
Victor E. Cameron Professor of Electrical and Computer Engineering and Computer Science

Anshumali Shrivastava
Assistant Professor of Computer Science

Reinhard Heckel
Assistant Professor of Electrical and Computer Engineering

Houston, Texas
May, 2019
ABSTRACT

GEM
Incorporating Context into Genomic Distance Estimation

by

CJ Barberan

A pivotal question in computational biology is how similar two organisms are based on their genomic sequences. Unfortunately, classical sequence-alignment-based methods for estimating genomic distances do not scale well to the massive number of organisms that have been sequenced to date. Recently, composition-based methods have gained interest due to their computational efficiencies for massive distance estimation problems. However, these methods reduce the computation time at the cost of distorting the genomic distances. The main problem with composition-based methods is their reliance on the occurrence of length-k subsequences of the genome, known as k-mers, which ignores their ordering, i.e., their context in the genome. In this paper, we take inspiration from computational linguistics to develop a new genomic distance estimation approach that exploits not only the frequency of the k-mers but also their context. In our Genomic distance EstiMation (GEM) algorithm, we first learn a context-aware, low-dimensional embedding for k-mers by training on a large corpus of genomic data comprising over one million bases of whole genome sequence data from microbial organisms in the National Center of Biotechnology Information (NCBI) repository. We then define the distance between two organisms using a generalization of the Jaccard similarity that incorporates the context-aware
embedding of the constituent $k$-mers. A range of experiments demonstrate that GEM estimates the distance between unseen organisms while outperforming the baseline approach.
Acknowledgments

I would like to thank Dr. Amirali Aghazadeh since it was with this collaboration that led to the creation of the GEM model, and for providing input and advice throughout the years. In addition, I would like to thank Dr. Todd J. Treangen for his guidance and enthusiasm into assessing GEM’s impact. I would like to thank the other committee members: Dr. Anshumali Shrivastava and Dr. Reinhard Heckel for providing input. Lastly, I would like to thank my advisor, Dr. Richard G. Baraniuk, for all of his guidance and mentorship.
Contents

Abstract ii
Acknowledgments iv
List of Illustrations vii
List of Tables ix

1 Introduction 1
   1.1 Thesis Overview ............................................. 3

2 Genomic Distance Estimation 4
   2.1 Genomic Distance Estimation Problem ........................... 4

3 GEM: Genomic distance EstiMation Algorithm 6
   3.1 Overview of GEM ............................................. 6
   3.2 GEM Embedding Model ...................................... 7
       3.2.1 GEM Contextual Similarity Measure ................. 9

4 Experiments and Results 15
   4.1 Experiment Details and Parameters .......................... 15
   4.2 Distance Estimation on Bacteria ............................. 17
       4.2.1 Distance Estimation on Training Set .............. 17
       4.2.2 Distance Estimation on Test Set ................. 17
   4.3 Distance Estimation on Different Domains ............... 19
   4.4 Larger-Scale Distance Estimations ....................... 21
   4.5 Phylogenetic Tree Construction ............................ 21
4.6 How Different Parameters Affect GEM ........................................... 23
4.7 Visualization of $K$-mers ................................................................. 24

5 Conclusion ..................................................................................... 35

A Supplementary Material ................................................................ 37

Bibliography ..................................................................................... 58
Illustrations

3.1 Illustration of the preprocessing section of GEM. A We start by segmenting the sequence of a large dictionary of different organisms’ genomes into short non-overlapping subsequences of length $k$ called $k$-mers. We produce a $k$-mer representation of the $N'$ extracted varying-length $k$-mers from DNA sequences. B Each $k$-mer in the context window has a sliding window applied to acquire smaller-length $k$-mers to send as additional input to the network. C With all the $k$-mers extracted in A and B, they are sent to be formatted in the one hot encoding format to be used for the GEM Network.

3.2 Illustration of the GEM Network and Embedding. A We then train a two layer neural network which predicts the occurrence of a $k$-mer given its surrounding $k$-mers as a one-hot encoded input. The weight matrix $W$ in the trained neural network acts as an embedding operator into a lower ($M \ll N'$) dimensional space. B We use the weight matrix $W$ to embed the $k$-mers into a low-dimensional space in $\mathbb{R}^M$. The new so-called GEM embedding of $k$-mers incorporates the spatial ordering of the original DNA sequences. The $k$-mers in black denote the full length $k$-mers while the $k$-mers in gray denote the smaller length $k$-mers that will be used to approximate the unseen full length $k$-mers not in the vocabulary.
3.3 Illustrative example of an unseen $k$-mer not found in the training set.
The $k$-mer in the example is approximated by using a sliding window
to create the collection of smaller length $k$-mers that will
approximate the unseen $k$-mer. ........................................ 14

4.1 Comparison of the GC content across the different sets. ............... 27
4.2 Scatterplot showing between the ANI and distance for GEM and the
baseline approach for a subset of organisms in D1. RMSE for GEM is
0.0691 ± 0.0051. RMSE for the baseline is 0.0786 ± 0.0102. ............ 28
4.3 Scatterplot showing between the ANI and distance for GEM and the
baseline approach for a subset of organisms in D2. RMSE for GEM is
6.491e$^{-4}$ ± 4.7662e$^{-7}$. RMSE for the baseline is 6.497e$^{-4}$ ± 4.747e$^{-7}$. 29
4.4 Comparison of different phylogenetic trees. ............................ 30
4.5 Comparison of different phylogenetic trees.I ............................. 31
4.6 Comparison of different phylogenetic trees. ............................. 32
4.7 UMAP dimensionality reduction visualization of a subset of $k$-mers
from $Staph. aureus$ (red) and $Staph. epidermidis$ (blue). The ANI for
these two organisms is 0.767. ........................................ 33
4.8 UMAP dimensionality reduction visualization of a subset of $k$-mers
from $Cupriavidus metallidurans$ (blue) and $Bartonella henselae$ (red).
The ANI for these two organisms is 0.727. ............................ 34
### Tables

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>RMSE for the baseline and the best GEM $\alpha$ parameter for the bacteria organisms that were used in the training set. GEM model trained on bacteria sequences with an $\alpha = 8$ outperforms the baseline.</td>
</tr>
<tr>
<td>4.2</td>
<td>RMSE for the baseline and the best GEM $\alpha$ parameter for the bacteria organisms in the test set. The best performance is the GEM model with an $\alpha = 8$ trained on bacteria.</td>
</tr>
<tr>
<td>4.3</td>
<td>RMSE for the baseline and the best GEM $\alpha$ parameter for viral organisms. These RMSE errors are only for the viral pairs that had the ground truth greater than 0. All the GEM models trained on different domains outperform the baseline. The best performance was from the GEM model with an $\alpha = 15$ trained on fungi.</td>
</tr>
<tr>
<td>4.4</td>
<td>RMSE for the baseline and the best GEM $\alpha$ parameter for fungi organisms. The best performer is the baseline on the fungi test set.</td>
</tr>
<tr>
<td>4.5</td>
<td>RMSE for the baseline approach and GEM and other variant GEM models to assess how the different parameters affect performance. The best GEM model outperforms the baseline approach. This demonstrates that more training data or approximating with more $k$-mers on unseen $k$-mers doesn’t benefit the model. This experiment was conducted on a subset of the D1 dataset.</td>
</tr>
</tbody>
</table>
4.6 RMSE for the baseline approach and GEM and other variant GEM models to assess how the different parameters affect performance. The best GEM model outperforms the baseline approach. This demonstrates that more training data or approximating with more $k$-mers on unseen $k$-mers doesn’t benefit the model. This test experiment was conducted on a separate subset of D1 organisms different from the previous table.
Chapter 1

Introduction

Measuring the biological distances between organisms is a fundamental problem in evolutionary genetics. A prime application is in defining the biological units of life—a new species is typically defined based on its genomic distance from previously defined organisms [1]. Besides its importance in defining biological organisms, rapid genomic distance estimation enables the organization of DNA sequences into data structures (evolutionary trees and clusters) that enable efficient search and query of massive genomic datasets.

About 60 years ago, DNA-DNA hybridization (DDH) became the gold standard technique to measure the genomic distance between organisms in vitro [2]. Unfortunately, DDH experiments are time-consuming, labor-intensive, and scale-wise limiting in building cumulative databases [3].

The rapid growth of DNA sequencing technologies has provided alternatives to DDH for estimating genomic distances using DNA sequences in silico. Average Nucleotide Identity (ANI) [4], the most widely used similarity measure today, finds the average alignment level between the DNA sequences of the target organisms. ANI breaks the DNA sequence of one genome into 1020-nucleotide fragments and then counts the portion of fragments that align with the DNA sequence of the other genome. An ANI score greater than 95% corresponds to 70% DDH similarity, which is popularly used as a threshold to define a new species [4, 5]. While accurate in estimating DDH, ANI does not scale adequately to handle the growing number of mi-
crobial organisms sequenced today, because it requires the multi-alignment of millions of large DNA fragments from thousands of organisms.

Recently, new scalable methods have been proposed to efficiently approximate the alignment step in ANI. These alignment-free methods [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18] use the frequency (or occurrence) of short DNA subsequences of length $k$, called $k$-mers, for their genomic task. These implementations, in particular, [6, 11, 12, 13, 14, 15] use $k$-mers to approximate the distance of organisms and avoid the time-consuming ANI alignment step by operating on a vector representation of DNA sequences that enables fast algebraic operations. Some, like [16], produce a speedup of at least $50\times$ compared to ANI. Unfortunately, decomposing a DNA sequence into its $k$-mers discards information on their ordering, i.e., the context of the $k$-mers, which degrades the performance of these genomic distance estimation algorithms.

In this thesis, we develop the Genomic distance EstiMation (GEM) algorithm to address the above shortcomings in ANI-based distance estimation of massive genomes. GEM uses the context information of co-occurring $k$-mers in hundreds of samples (comprising over millions of nucleotide base pairs) of microbial DNA sequences from the National Center of Biotechnology Information (NCBI) repository to learn an embedding of $k$-mers represented in a low-dimensional, biologically meaningful subspace. Then, GEM incorporates the contextual $k$-mer embeddings into a new Generalized Jaccard (GJ) similarity measure that captures the pairwise interactions between the $k$-mers to estimate the genomic distance between the organisms.

An important hallmark of GEM is that its ANI-distance estimate preserves the structure of $k$-mers in new, unseen target organisms. Since an unseen organism might have $k$-mers that did not appear in the training data, GEM leverages a recently developed machine learning technique from the natural language processing (NLP)
literature called FastText [19] to approximate an unseen $k$-mer using its constituent shorter subsequences that are shared among the larger length $k$-mers.

We demonstrate that even a small randomly subsampled set of $k$-mers is sufficient to estimate ANI using GEM, since the distance between $k$-mers encapsulates the structure of the entire genome sequence. We further demonstrate that GEM significantly outperforms the baseline approach and GEM is inspired from state-of-the-art algorithms such as Mash [6] in estimating the ANI score with the same number of sub-sampled $k$-mers. We validate GEM on a test set of microbial organisms and show that it generalizes well on organisms with divergent genomes.

We demonstrate GEM’s generalization capabilities by showing that it can accurately estimate ANI distances between organisms with vast divergent genomes. We show that with GEM’s genomic distance estimations can affect how phylogenetic tree is formed compared to the baseline.

1.1 Thesis Overview

In this thesis, background information about genomic distance estimation will be in Chapter 2. The formulation of the GEM algorithm will be in Chapter 3. Experiments and results will be in Chapter 4. The conclusion and discussion will be in Chapter 5. Supplementary material such as the organisms used in the experiments will be in the Appendix.
Chapter 2

Genomic Distance Estimation

2.1 Genomic Distance Estimation Problem

Consider two DNA sequences $S_1$ and $S_2$ of lengths $L_1$ and $L_2$, respectively. We are interested in finding an estimate of the genomic distance $\hat{D}(S_1, S_2)$ between the sequences that best approximates the ANI distance. Alignment-free approaches estimate the genomic distance using a function $f(S_1, S_2) \approx \hat{D}(S_1, S_2)$ of the two sets of $k$-mers $S_1 = \{k_1, k_2, \ldots, k_{N_1}\}$ and $S_2 = \{k'_1, k'_2, \ldots, k'_{N_2}\}$ where $N_1$ and $N_2$ are the number of $k$-mers in each set. Here, $k_i$ and $k'_i$ denote the $i^{th}$ $k$-mers in the sets $S_1$ and $S_2$, respectively. Recall that the $k$-mers of a DNA sequence are all subsequences of length $k$.

Consider several examples of composition-based ANI estimation algorithms. Mash [6] estimates the distance based on the Jaccard similarity [20] $J(S_1, S_2)$ between the $k$-mer sets. The Jaccard similarity [20],

$$\text{J}(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}, \quad (2.1)$$

finds the ratio of the number of shared elements to the total number of distinct $k$-mers in the sets $S_1$ and $S_2$. Maximum unique matches (MUMs) [14] estimates the distance by summing the length of nonoverlapping $k$-mers shared between the two sequences $S_1$ and $S_2$ and dividing by the minimum or average length of the genome-sequence pair. kWIP [15] estimates the distance using a weighted inner product between a pair of sketches, which is a structure that holds the frequency of the $k$-mers, of $S_1$ and $S_2$. 
The problem with the current alignment-free algorithms for genomic distance estimation is that the $k$-mer sets $S_1$ and $S_2$ have lost information regarding the ordering of the $k$-mers in the sequences of $S_1$ and $S_2$. Consider a common scenario in genomics where two $k$-mers $k_i$ and $k_j$ frequently co-occur in the vicinity of each other in an organism’s genome sequence. For example, assume that they are part of a shared gene or a regulatory sequence (i.e., they have a shared genomic context). Then it is highly likely that in two very close species’ sequences, one of the two $k$-mers, say $k_i$, acquires a mutation due to a sequencing error or some other non-evolutionary reason. As a result of such a mutation, the exact $k$-mer does not explicitly show up in the second organism’s $k$-mer set, $S_2$, while being present in $S_1$. Current alignment-free algorithms are not able to leverage such a pairwise interaction between the co-occurring $k$-mers, resulting in them not accurately estimating the genomic distance. A standing challenge in large-scale genomic distance estimation is to define a similarity measure between organisms that leverages such pairwise interactions between their constituent $k$-mers.
Chapter 3

GEM: Genomic distance EstiMation Algorithm

3.1 Overview of GEM

The Genomic distance EstiMation (GEM) algorithm comprises the following three steps:

Training: Using a corpus of DNA sequences, we take inspiration from natural language processing (NLP) and train a neural network to predict a $k$-mer given its surrounding $k$-mers in a local context window. Training is performed offline and one-time only.

Embedding: The first stage of the neural network produces a linear projection of the one-hot vectors into a low-dimensional embedding space that we call the GEM Embedding. In this embedding space, the distance between two $k$-mers is given by their contexts, i.e, how similar are their surrounding $k$-mers in the training sequence data.

Distance Estimation: We approximate the ANI distance between two genomes by converting the DNA sequences of each to $k$-mers, projecting the $k$-mers into the GEM embedding space, and then computing the Generalized Jaccard similarity between the two sets of points. Once the Generalized Jaccard similarity is computed it will be used into the genomic distance formulation to acquire the ANI estimate.
Like other alignment-free ANI estimation methods, the embedding step significantly reduces both the computational and memory footprint of GEM. In contrast to other alignment-free methods, however, GEM takes into account the local context of each $k$-mer, which as we will see leads to an improved genomic distance estimate. This section provides the details of each of the above steps.

### 3.2 GEM Embedding Model

We learn a low-dimensional $k$-mer embedding by first collecting a set of DNA training sequences. It is important to note that no information other than raw DNA sequences is required to train the GEM embedding. We segment the DNA training sequences into shorter non-overlapping $k$-mer subsequences of length $k$; we refer to the set of $k$-mers as the *vocabulary* of the DNA training set. In addition, a sliding window is applied to every full-length $k$-mer to create smaller-length $k$-mers. In Fig. 3.1, the figure demonstrates the process of creating the full-length and smaller-length $k$-mers.

Let $N$ denote the number of unique full-length $k$-mers in the vocabulary. Let $n'$ denote the number of unique smaller length $k$-mers, where $N' = N + n'$. We encode the $i^{th}$ $k$-mer in the vocabulary as a length-$N'$ vector of all zeros except a 1 in the $i^{th}$ position; this is called a “one-hot” encoding in machine learning parlance.

We loop over all the $k$-mers extracted from the database and for each $k$-mer we collect $C$ one-hot encoded vectors as inputs of the network. The dimensionality of the input is $N'$, a hidden layer of dimensionality is $M \ll N'$, $W$ is the hidden layer’s weight matrix, and the output dimensionality is $N$ (see Fig. 3.2). Notice that the network will not predict any of the smaller length $k$-mers just the full length $k$-mers. The input has to be of dimension $N'$ in order to learn the embedded smaller length $k$-mers vectors.
The objective of the neural network is to predict the correct $k$-mer given its surrounding $k$-mers in a predefined window, known as the context window, $C$, using the single hidden layer. The loss function of the neural network is the negative log likelihood of the particular $k$-mer given its context of other $k$-mers surrounding it. Hence, $k$-mers with similar context windows will have similar weights in $W$. Drawing an analogy to language (text) processing, the prediction task is analogous to completing a sentence with a missing word. The network is trained to fill in, as an example, the following sentence “We see . . . we want.” with no prior knowledge of English grammar with the word “what”. This type of neural network is also known as a “continuous bag of words” model.

Take the bold $k = 4$-mer “ATTA” in Fig. 3.1A, as an example. This $k$-mer is surrounded by $C = 1$ other $k$-mers. These vectors correspond to $k$-mers occurring before and the other vectors correspond to $k$-mers occurring after the $k$-mer. Then a sliding window will be applied to the $k$-mers in the context window to create smaller length $k$-mers. In this scenario, the smaller length $k$-mers are $k = 2$ and 3. Looking at Fig. 3.1B, those are the smaller length $k$-mers that will be sent additionally as the sum of one-hot vectors to the network. The output of the neural network is the one-hot encoded vector corresponding to the full-length $k$-mer that the network is predicting is the correct $k$-mer (Fig. 3.2).

Once the network is trained for the task of predicting a $k$-mer given its surrounding $k$-mers and smaller length $k$-mers, we use the hidden layer’s weights $W$ to embed each $k$-mer from a subspace in $\mathbb{R}^{N'}$ to a much lower-dimensional subspace in $\mathbb{R}^M$, where $M \ll N'$ (see Fig. 3.2 b). In contrast to the canonical representation of $k$-mers, in which the distance between two $k$-mers is given by the Hamming distance between their DNA sequences. In the GEM representation, the distance between two $k$-mers
is given by the context of the $k$-mers, i.e, how similar are the surrounding $k$-mers in the training sequence data.

Notice that the input has the dimensionality of $N'$ while the output’s dimensionality is $N$. The rationale is that with $k$ getting larger, the number of unique $k$-mers is growing exponentially. Thus, it will be infeasible to store all of the $M$-dimensional $k$-mer vectors. In addition, with the data provided, it can be impractical to have all the data to learn all the unique $k$-mers from $N$. Hence, there will be smaller length $k$-mers that have an $M$-dimensional vector where they will be used to approximate the unseen $k$-mer. Let $k_u$ be the unseen $k$-mer and let $S_u$ be the set of smaller length $k$-mers of $k_u$. The $M$-dimensional vector representation of $k_u$ can be expressed as $\sum_{s \in S_u} k_s$ where $k$ is the $M$-dimensional vector representation of the smaller length $k$-mer. Therefore, the unseen $k$-mer is being approximated by the summation of the smaller length $k$-mers that are composed of it. An illustrative example is provided Fig. 3.3 where a 4-mer was not in the training set of the vocabulary and had to be approximated by the summation of it’s 2- and 3-mers.

### 3.2.1 GEM Contextual Similarity Measure

Now that we are equipped with a contextual embedding of $k$-mers we are ready to define a new similarity measure that incorporates $k$-mers’s pairwise interactions. Our genomic distance has the potential to be applied to other composition-based techniques described earlier to account for pairwise $k$-mer interactions. In this scenario, we choose to build the genomic distance based on the Jaccard similarity [20] in Mash for its superior performance over other composition-based methods.

In GEM we extend a measure of similarity that incorporates the context of the $k$-mer embeddings. In order to define the new measure we first use an alternative
formulation of the Jaccard similarity, also known as Ruzicka similarity [21],

\[ J(S_1, S_2) = \frac{\sum_{i=1}^{N} \min(S_{1i}, S_{2i})}{\sum_{i=1}^{N} \max(S_{1i}, S_{2i})}, \]  

(3.1)

where \( S_{1i} \) and \( S_{2i} \) are binary variables that indicate the presence of the \( i^{th} \) k-mer in sets \( S_1 \) and \( S_2 \), respectively and \( N \) counts the total number of k-mers in both sets. It is easy to inspect that equation (2.1) is the equivalent of equation (3.1). We modify equation (3.1) to sum over every pair of k-mers weighted by the distance between the pair of k-mers and arrive at our Generalized Jaccard similarity,

\[ GJ(S_1, S_2) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{i,j}^{\alpha} \min(S_{1i}, S_{2j})}{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{i,j}^{\alpha} \max(S_{1i}, S_{2j})}, \]  

(3.2)

where \( w_{i,j} = \langle GEM(k_i), GEM(k_j) \rangle \) is the inner product between the embedding vectors of \( k_i \) and \( k_j \) k-mers and if the embeddings were orthogonal (as in the one-hot encodings), then GJ similarity is equivalent to Jaccard similarity. Notice that \( \alpha \) is a parameter to adjust the weight of the pairwise k-mer interactions in the overall genomic distance estimation. In addition, \( \alpha \) can help offset the bias from the training data. Hence, the \( \alpha \) parameter can be set such that if the k-mers are from a noisy dataset, then \( \alpha \) can be set to a larger value to put less emphasis on the genomic distance estimation. If the k-mers have a reliable vector representation then \( \alpha \) can be a smaller value. For large values of \( \alpha \), equation (3.2) converges to the other original Jaccard similarity equation (2.1 as expected).

The Generalized Jaccard similarity is summing over both k-mer sets using the inner product of the pairwise k-mer vectors calculated from GEM to provide the contextual information encoded by learning the k-mers’ interactions. Fine-tuning with the \( \alpha \) parameter will show how much of the cosine similarity weight is needed to provide the right amount of contextual information to determine the best distance
estimation due to the training data bias.

To estimate the genomic distance using the Jaccard similarity we use the same derivation from [11] and [6]. Suppose that the probability of a single nucleotide substitution in a $k$-mer is denoted as $d$. The probability that a substitution will happen within that $k$-mer is given as $e^{-kd}$ provided that the expected number of mutations to happen within a $k$-mer is $kd$. Assuming that the model is Poisson, the probability that a mutation does not happen is given as $e^{-kd}$. Consider that letting $t$ be the number of all the $k$-mers and $w$ be the conserved $k$-mer count, then by solving $e^{-kd} = \frac{w}{t}$ for $d$, then we can express $d$ as $-\frac{1}{k} \ln(\frac{w}{t})$. Integrating this to the Jaccard similarity, it can be expressed in terms of $w$ and a new term called the average size of the genomes, $n$, such that $j$ is approximately $\frac{w}{2n-w}$. By replacing $n$ for $t$ from the earlier statement, we can approximate $\frac{w}{n}$ as $\frac{2j}{1+j}$. From that approximation, $\frac{w}{n}$ can be substituted for $\frac{w}{t}$. With the substitution of $t$ for $n$, $\hat{D} = -\frac{1}{k} \ln(\frac{2j}{1+j})$. With $\hat{D}$ in terms of $j$, then we can substitute $j$ in lieu of GJ to have our genomic distance estimate,

$$\hat{D}(S_1, S_2) = -\frac{1}{k} \ln\left(\frac{2GJ(S_1, S_2)}{1 + GJ(S_1, S_2)}\right).$$  \hspace{1cm} (3.3)
Figure 3.1: Illustration of the preprocessing section of GEM. A We start by segmenting the sequence of a large dictionary of different organisms’ genomes into short non-overlapping subsequences of length $k$ called $k$-mers. We produce a $k$-mer representation of the $N'$ extracted varying-length $k$-mers from DNA sequences. B Each $k$-mer in the context window has a sliding window applied to acquire smaller-length $k$-mers to send as additional input to the network. C With all the $k$-mers extracted in A and B, they are sent to be formatted in the one hot encoding format to be used for the GEM Network.
We then train a two layer neural network which predicts the occurrence of a $k$-mer given its surrounding $k$-mers as a one-hot encoded input. The weight matrix $W$ in the trained neural network acts as an embedding operator into a lower ($M \ll N'$) dimensional space.

We use the weight matrix $W$ to embed the $k$-mers into a low-dimensional space in $\mathbb{R}^M$. The new so-called GEM embedding of $k$-mers incorporates the spatial ordering of the original DNA sequences. The $k$-mers in black denote the full length $k$-mers while the $k$-mers in gray denote the smaller length $k$-mers that will be used to approximate the unseen full length $k$-mers not in the vocabulary.
Figure 3.3 : Illustrative example of an unseen $k$-mer not found in the training set. The $k$-mer in the example is approximated by using a sliding window to create the collection of smaller length $k$-mers that will approximate the unseen $k$-mer.
Chapter 4

Experiments and Results

We showcase the performance of the GEM algorithm in estimating the genomic distances by testing our approach against a baseline approach that is inspired from Mash [6] across a multitude of experiments.

The task is to estimate the average nucleotide identity (ANI) from these papers [5, 3] that were available from the PyANI Github*. This provided the ground truth to evaluate the performance of using GEM compared to the baseline approach.

4.1 Experiment Details and Parameters

A list of organisms is provided in the Supplementary Material. The choice of $k$ for the $k$-mers is 15 and we subsample 5,000 $k$-mers from the union of the two organisms in estimating the ANI with Jaccard. The 5,000 $k$-mers of the union of the two organisms would be compared to the set of $k$-mers from each organism. The rationale is that if we sampled from the original $4^k$ and for organisms that share very few $k$-mers, the Jaccard estimate would be much lower or it would be 0 since there were no shared $k$-mers that were sampled from the union. When acquiring the distance estimation between organisms, we use GenomeTester [22] to provide unions and intersections of their $k$-mers for each organism. For the weights of GEM, we utilize FastText to learn the $k$-mer vectors in order to assess the similarities between $k$-mers. For

*https://github.com/widdowquinn/pyani
the $\alpha$ parameter in GEM, we set $\alpha = \{1,2,3,4,5,6,7,8,9,10,12,15,20,25,50\}$ for most situations. In the hyperparameter section, the $\alpha$ parameter would be set due to the changes in the GEM model. With the ground truth provided, we next calculate the root mean square error (RMSE) $\sqrt{\frac{\sum_{n=1}^{N}(\hat{y}_n-y_n)^2}{N}}$. Having different $\alpha$ parameters shows how the performance fluctuates and when $\alpha$ goes to $\infty$ the RMSE error will converge to the baseline RMSE error, which is to be expected.

To train the network we extract non-overlapping $k$-mers of the genome sequences into one large corpus. The GEM model is trained on 44 bacteria organisms. The dimensionality of the embedded space is set to $M = 100$ unless it is specified differently in other sections. For most scenarios, in learning the $k$-mer embedding, the context window uses 1000 $k$-mers before and after the $k$-mer ($C = 1000$) that is to be predicted from the network. For negative sampling, which is a technique that provides negative instances for the model to learn more effectively what is the data and what is noise, the parameter is set to 5. The sampling parameter is set to $10^{-4}$ in order to counter the imbalance between rare and frequent $k$-mers. This model is trained using FastText [19] where the choice of model is a continuous bag-of-words to embed the $k$-mers. For most of the GEM models, the minimum number of n-grams is set to 4 and the maximum amount of n-grams is set to 6. Other parameters were default from the FastText implementation. Code for the implementation will be available on Gitlab$^\dagger$.

For comparing against the GEM model, the baseline approach was inspired from Mash. With Mash, it applies a technique called MinHash [23] where it applies hashing to the $k$-mers. By using hashing, it results in not having the ability to recover what those $k$-mers were. Hence, by applying random sampling, GEM and the baseline

$^\dagger$https://gitlab.com/cjbarb7/GEM-genome
approach can have the same set of $k$-mers to be evaluated as a fair comparison. Thus, GEM and the baseline approach can have the same information in the form of $k$-mers and to assess would implementation would outperform. There could be future work such that the $k$-mers are sampled using a hashing function to sample a small amount of $k$-mers. Yet, to repeat, this approach was given the same set of $k$-mers would GEM outperform the other approach. Plus, it would be a good assessment to evaluate how Generalized Jaccard similarity compares to regular Jaccard similarity.

### 4.2 Distance Estimation on Bacteria

#### 4.2.1 Distance Estimation on Training Set

The first experiment is to assess the performance of GEM on bacteria compared to a method that doesn’t take contextual information. When using GEM to learn the $k$-mer vectors we had a training set of bacteria. For this experiment, we used the same training data of bacteria to get their distance estimations. The number of organisms are 44 and each organism is being compared against the other organisms.

In Table 4.1, the best $\alpha$ parameter with the lowest RMSE is 8. In Table 4.1, GEM outperforms the baseline approach. Thus, this is the first experiment to show that context can provide more information in genomic distance estimation. In addition, the alpha parameter with the best performance in this experiment will be used for the other experiments to assess the overall performance in generalization.

#### 4.2.2 Distance Estimation on Test Set

This experiment is to assess on a test set of bacteria that were not included in the training set of bacteria. In addition, the test set of bacteria could have $k$-mers that
Table 4.1: RMSE for the baseline and the best GEM $\alpha$ parameter for the bacteria organisms that were used in the training set. GEM model trained on bacteria sequences with an $\alpha = 8$ outperforms the baseline.

were not in the training set for GEM to learn. For this reason, this experiment is to assess how GEM can estimate the ANIs from bacteria that were not in the training set.

In Table 4.2, GEM outperforms the baseline with $\alpha = 8$. This demonstrates that GEM was able to learn a good $k$-mer model to approximate the unseen $k$-mers from the test set. To add, this experiment illustrates how the GEM embedding can generalize to unseen bacteria and approximate $k$-mers that were not in the training set well.

Table 4.2: RMSE for the baseline and the best GEM $\alpha$ parameter for the bacteria organisms in the test set. The best performance is the GEM model with an $\alpha = 8$ trained on bacteria.
4.3 Distance Estimation on Different Domains

The experiments conducted before demonstrate GEM’s capability on bacteria. Now for the next experiments, we train GEM on other training sets of genomic sequences such as viruses and fungi. The purpose is to assess GEM’s ability in estimating genomic distances outside of bacteria. The first GEM model is trained on 44 bacteria organisms. The second GEM model is trained on 102 viruses. The third GEM model is trained on 10 fungi. The list of particular organisms is in the Appendix.

This new experiment is similar from the previous experiment, but this experiment included additional GEM models trained on viruses and fungi. In Table 4.2, the GEM model was only trained on bacteria and now with training using different types of organism, the question would be how GEM would perform. In addition, what different types of bias would be created by training on different organisms that have a different genomic composition from bacteria.

In Table 4.3, the experiment is to predict the genomic distances from viruses and the best performers are the GEM models trained with bacteria and fungi. We suspect that the virus-trained GEM model was not the best because in Fig. 4.1 the distributions of the GC content of the organisms demonstrates the bias with the embedding. The virus test set has a similar histogram distribution to the fungi training set compared to the histogram distribution of the virus training set. This offers a more biological explanation to why the bias of the embedding mimics the performance. In the NLP community, there has been evidence [24] about bias in the training corpus that mimics to how the word vectors are represented.

In Table 4.4, the experiment is to predict the genomic distances from fungi. None of the GEM models outperform the baseline. Noticing that the baseline’s genomic distance estimations were underestimating the true ANI value is an indicator that
GEM will have poor performance compared to the baseline. With the other experiments, the baseline approach tends to overestimate the genomic distance estimations which leads to GEM outperforming in the previous experiments.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$\alpha$</th>
<th>RMSE</th>
<th>Trained on</th>
<th>Tested on</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEM</td>
<td>8</td>
<td>0.00892 ± 0.0048</td>
<td>Bacteria</td>
<td>Virus</td>
</tr>
<tr>
<td>GEM</td>
<td>15</td>
<td><strong>0.0219 ± 0.000433</strong></td>
<td>Fungi</td>
<td>Virus</td>
</tr>
<tr>
<td>GEM</td>
<td>50</td>
<td>0.0578 ± 0.0062</td>
<td>Virus</td>
<td>Virus</td>
</tr>
<tr>
<td>Baseline</td>
<td>n/a</td>
<td>0.0732 ± 0.0086</td>
<td>n/a</td>
<td>Virus</td>
</tr>
</tbody>
</table>

Table 4.3: RMSE for the baseline and the best GEM $\alpha$ parameter for viral organisms.
These RMSE errors are only for the viral pairs that had the ground truth greater than 0. All the GEM models trained on different domains outperform the baseline. The best performance was from the GEM model with an $\alpha = 15$ trained on fungi.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$\alpha$</th>
<th>RMSE</th>
<th>Trained on</th>
<th>Tested on</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEM</td>
<td>8</td>
<td>0.1125 ± 0.00716</td>
<td>Bacteria</td>
<td>Fungi</td>
</tr>
<tr>
<td>GEM</td>
<td>15</td>
<td>0.10928 ± 0.00736</td>
<td>Fungi</td>
<td>Fungi</td>
</tr>
<tr>
<td>Baseline</td>
<td>n/a</td>
<td><strong>0.101 ± 0.00649</strong></td>
<td>n/a</td>
<td>Fungi</td>
</tr>
</tbody>
</table>

Table 4.4: RMSE for the baseline and the best GEM $\alpha$ parameter for fungi organisms.
The best performer is the baseline on the fungi test set.
4.4 Larger-Scale Distance Estimations

From [25], there were five datasets comprised of bacteria genomes. We are using one of the five datasets called D1 since numerous genomes are more divergent to each other in order to assess the performance of GEM and the baseline approach. In the previous sections with the test set there were a multitude of varying divergent genomes but this was another instance of a separate dataset with different organisms. In addition, another dataset from [25] called D2 was used. Multiple organisms from D2 had ANI values that were greater than 90 percent. For both datasets, a random selection of organisms were used in D1 and D2. In D1, about 482 pairwise genomic distance estimations were calculated while D2 had about 45 pairwise genomic distance estimations.

In Fig. 4.2, it demonstrates how much better GEM is in providing the genomic distance estimation compared to the baseline approach. With comparing more divergent organisms, there will be less intersection of shared \( k \)-mers. Hence, having the reformulation of the Jaccard similarity take into account the pairwise \( k \)-mer interactions aids. There are not many differences between GEM and the baseline approach in Fig. 4.3. This informs that with comparing two organisms that are very similar there is little benefit since the two organisms share a multitude of shared \( k \)-mers.

4.5 Phylogenetic Tree Construction

With the GEM model outperforming the baseline approach, the next step was to assess how a group of organisms can be clustered into a phylogenetic tree based on their genomic distance estimations. In the previous experiments, there were genomic distance estimations for a group of organisms that would enable the creation of the
phylogenetic tree. In the previous experiments, not all organisms had a genomic distance estimation to every other organism in the dataset. The visualizations of the phylogenetic trees from the different approaches were constructed by [26]. In order to calculate the differences of the trees from GEM and the baseline to ANI, DendroPy [27] is used to calculate the weighted and unweighted Robinson-Foulds (RF) distance [28].

In Fig. 4.4, there are the different phylogenetic trees of ANI, baseline, and GEM. Using the weighted/unweighted RF distance to the ANI tree, the weighted and unweighted distances for GEM are 0.01917 and 2. For the baseline approach, the respective values are 0.01678 and 2. The organisms involved in this tree were of *Bacillus anthracis* and *Bacillus cereus*. This is a good example to assess the phylogenetic tree is constructed because other approaches can confuse anthracis and cereus based on the genomic composition. However, the baseline and GEM were very similar in how their trees were constructed.

In Fig. 4.5, there are the different phylogenetic trees of ANI, baseline, and GEM. Using the weighted/unweighted RF distance to the ANI tree, the weighted and unweighted distances for GEM are 1.36827 and 12. For the Baseline approach, the respective values are 1.43618 and 12. Another example of both the Baseline and GEM trees having the same unweighted RF distance while GEM has a lower weighted RF distance. The organisms used in this tree were from another subset of the D1 dataset. In Fig. 4.5, there are some differences in how the baseline and GEM construct their trees and their trees are different from how ANI constructed the correct tree.

In Fig. 4.6, there are the different phylogenetic trees of ANI, Baseline, and GEM. Using the weighted/unweighted RF distance to the ANI tree, the weighted and unweighted RF distances for GEM is 0.55851 and 12. For the baseline approach, the
respective values are 0.63158 and 10. The organisms used in this tree were from a subset in the bacteria testing set. It is interesting that the weighted RF distance is lower for GEM, yet the unweighted RF distance is lower for the Baseline. Looking at Fig. 4.6, there are differences in how GEM and the baseline constructed their own phylogenetic tree. In Fig. 4.6, the baseline tree has a similar tree composition with the branches compared to the GEM tree. This makes sense why the unweighted RF distance is lower.

Overall, the GEM approach showed that for most of these scenarios is that the weighted RF distance was lower compared to the ground truth. For the unweighted RF distance, it would have the same or worse unweighted RF distance.

4.6 How Different Parameters Affect GEM

In the prior sections, they outline how the GEM model can add benefit to genomic distance estimations in lieu of the baseline approach. Hence, more experiments were conducted to assess a study across adjusting different parameters of GEM.

One experiment was to apply a cross validation study in varying the parameters of the GEM model. The parameters to modify were the amount of training data provided, the range of smaller length \(k\)-mers to approximate the unseen \(k\)-mers, the dimensionality size of the embedding, and the number of \(k\)-mers in the context window. One GEM model was created with a training size of 104 bacteria genomes. This model will be referred to as GEM More Data. Second GEM model was created with having the same training data as the original GEM model but to approximate the unseen \(k\)-mers with \(k\) values of 4-9 instead of 4-6 as the original GEM model. This model is to be referred to as GEM More \(k\)-mers. Third different GEM model is similar to the GEM model with a training size of 104 bacteria genomes but the
dimensionality was increased to 200 instead of 100. This model is to be referred to as GEM Higher Dimension. Fourth GEM model is similar to the original GEM model except the context window size for the \(k\)-mers is reduced from 1000 to 10. This model is to be referred as GEM Smaller Context.

In Table 4.5, different \(\alpha\) values outside of the range that was specified before were used to find the best one for each GEM model. With the different parameter being adjusted, it demonstrates how \(\alpha\) varies for performance. With Table 4.5, it shows that the original GEM model outperforms all the different GEM models with their different parameter changes.

Using the same \(\alpha\) parameter from Table 4.5, a different subset of the D1 data is used to assess the generalization performance of GEM with the different hyperparameters adjusted. In Table 4.6, it details that having a smaller context window from the original GEM model outperforms it and the baseline approach. From assessing the different parameter adjustments, it demonstrates some of the limitations or bias that the GEM model has. For example, more data does not always benefit, especially the different types of organism could bias the embedded vectors.

### 4.7 Visualization of \(K\)-mers

In Figures 4.7 and 4.8, it is using UMAP [29] to project the \(M\)-dimensional \(k\)-mer vectors into a two-dimensional space. UMAP is a dimensionality reduction technique that is commonly used in word vector applications to project the word vectors into the 2D space for visualization purposes. UMAP is a useful tool to provide an idea of how the \(k\)-mers are being clustered since the idea is that \(k\)-mers from similar organisms should have similar context windows. Hence the similar \(k\)-mers should be grouped closer together while other dissimilar \(k\)-mers are separated further. In Fig. 4.7, there
Table 4.5: RMSE for the baseline approach and GEM and other variant GEM models to assess how the different parameters affect performance. The best GEM model outperforms the baseline approach. This demonstrates that more training data or approximating with more k-mers on unseen k-mers doesn’t benefit the model. This experiment was conducted on a subset of the D1 dataset.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>(\alpha)</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEM</td>
<td>8</td>
<td><strong>0.0567 ± 0.0039</strong></td>
</tr>
<tr>
<td>Baseline</td>
<td>n/a</td>
<td>0.0799 ± 0.0091</td>
</tr>
<tr>
<td>GEM More Data</td>
<td>50</td>
<td>0.06494 ± 0.0058</td>
</tr>
<tr>
<td>GEM More k-mers</td>
<td>135</td>
<td>0.06329 ± 0.00563</td>
</tr>
<tr>
<td>GEM Higher Dimension</td>
<td>50</td>
<td>0.06904 ± 0.0063</td>
</tr>
<tr>
<td>GEM Smaller Context</td>
<td>9</td>
<td>0.0567 ± 0.00419</td>
</tr>
</tbody>
</table>

are 500 k-mers each from *Staph. aureus* and *Staph. epidermidis*. Notice that the k-mers from each organism are in their own respective sets and not from the set of the intersection of the two organisms. In Fig. 4.7, the k-mers from each organism are grouped together. In Fig. 4.8, each organism’s k-mers seem to have a separate cluster while there are some k-mers of each organism grouped together. More analysis would have to be assessed with the context windows that those k-mers are being grouped in the UMAP dimensionality reduction.
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$\alpha$</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEM</td>
<td>8</td>
<td>0.05041 ± 0.0017</td>
</tr>
<tr>
<td>Baseline</td>
<td>n/a</td>
<td>0.06198 ± 0.0074</td>
</tr>
<tr>
<td>GEM More Data</td>
<td>50</td>
<td>0.0642 ± 0.00316</td>
</tr>
<tr>
<td>GEM More $k$-mers</td>
<td>135</td>
<td>0.06306 ± 0.00277</td>
</tr>
<tr>
<td>GEM Higher Dimension</td>
<td>50</td>
<td>0.05777 ± 0.00438</td>
</tr>
<tr>
<td>GEM Smaller Context</td>
<td>9</td>
<td><strong>0.0497 ± 0.00161</strong></td>
</tr>
</tbody>
</table>

Table 4.6: RMSE for the baseline approach and GEM and other variant GEM models to assess how the different parameters affect performance. The best GEM model outperforms the baseline approach. This demonstrates that more training data or approximating with more $k$-mers on unseen $k$-mers doesn’t benefit the model. This test experiment was conducted on a separate subset of D1 organisms different from the previous table.
Figure 4.1: Comparison of the GC content across the different sets.

(a) Histogram of the GC content of the training set of viruses.

(b) Histogram of the GC content of the test set of viruses.

(c) Histogram of the GC content of the training set of fungi.
Figure 4.2: Scatterplot showing the relationship between the ANI and distance for GEM and the baseline approach for a subset of organisms in D1. RMSE for GEM is $0.0691 \pm 0.0051$. RMSE for the baseline is $0.0786 \pm 0.0102$. 
Figure 4.3: Scatterplot showing the relationship between ANI and distance for GEM and the baseline approach for a subset of organisms in D2. RMSE for GEM is $6.491 \times 10^{-4} \pm 4.766 \times 10^{-7}$. RMSE for the baseline is $6.497 \times 10^{-4} \pm 4.747 \times 10^{-7}$. 
Figure 4.4: Comparison of different phylogenetic trees.
Figure 4.5: Comparison of different phylogenetic trees.

(a) Tree generated from ANI

(b) Tree generated from baseline

(c) Tree generated from GEM
Figure 4.6: Comparison of different phylogenetic trees.
Figure 4.7: UMAP dimensionality reduction visualization of a subset of k-mers from *Staph. aureus* (red) and *Staph. epidermidis* (blue). The ANI for these two organisms is 0.767.
Figure 4.8: UMAP dimensionality reduction visualization of a subset of $k$-mers from *Cupriavidus metallidurans* (blue) and *Bartonella henselae* (red). The ANI for these two organisms is 0.727.
Chapter 5

Conclusion

We have developed and validated GEM, a new genomic distance estimation approach that accounts for the contextual information of where $k$-mers are located within the genomic sequence. GEM is able to outperform through most of the experiments especially in divergent genomes. By taking into account the $k$-mer interactions as weights, the estimations are closer to the ground truth (ANI) than the baseline. This is very beneficial in the situation where the two organisms to be compared diverge compared to organisms that are related by the species level. In addition, with the amount of $k$-mers that we are sampling, the baseline approach had an issue with divergent organisms by the Jaccard similarity. Hence, incorporating the $k$-mer pairwise interaction is a method to help solve that issue. The GEM model is able to learn to assign $k$-mers with similar context windows to have similar embedded vectors while $k$-mers with different context windows would be dissimilar. The new Generalized Jaccard similarity formulation provides the additional information in the form of the $k$-mer interaction weighting that aided in estimating the genomic distance between organisms.

The GEM model’s learned representation of $k$-mers can be applied in other downstream tasks. For example, in the task of species/genus classification the input could be the embedded $k$-mer vectors as opposed to one hot encoding of the $k$-mers. The constant dimensionality of $k$-mers of arbitrary length affords flexibility over feature extraction methods for downstream genomic applications. GEM’s other feature which
is approximating unseen $k$-mers shows promise to extending this to higher $k$ values. For evaluating mammalian genomes, a higher $k$ would be preferred, where learning all $4^k$ is intractable in memory and unrealistic where a dataset can provide all $4^k$ $k$-mers. Therefore, having the ability to approximate with smaller length $k$-mers helps deal with these factors.

There are more avenues to where GEM could be applied in the field of genomics besides genomic distance estimation. Plus, there are areas to investigate more about the strengths and limitations of GEM. For instance, more data did not benefit for genomic distance estimations. There could be multiple follow-up experiments to probe more about GEM. For instance, using Simulome [30] to artificially create genomes and control the variables of insertions and deletions to provide detailed analysis of GEM’s performance. Another example is to hash the GEM model in order to compare against Mash for a fair comparison. In conclusion, based on the initial experiments with GEM, it holds promise to use for the genomic field and can showcase how machine learning can benefit this field.
Appendix A

Supplementary Material

The bacteria, viral, and fungi genome files were downloaded from the National Center of Biotechnology Information website. The genome files that were in the training corpus of FastText were in their FASTA format. For all of the genome sequences, they were downloaded in genbank files in the Refseq section. The files were complete genomes, not just shotgun sequences. Then using BioPython [31] to convert them into the FASTA format so GenomeTester could be used to count and collect the $k$-mers from each set and the union from the organisms’ sets.

The bacteria in the training set are

- *Cupriavidus metallidurans* CH34 chromosome
- *Bacteroides fragillis* 638R
- *Bacillus cereus* ATCC 14579
- *Bartonella henselae* str. Houston-1
- *Bordetella pertussis* Tohama I
- *Brucella abortus* S19 chromosome I
- *Campylobacter jejuni* subsp. doylei 269.97 chromosome
- *Chlamydia trachomatis* B/Jalio20/OT chromosome
- Clostridium botulinum B1 str. Okra
- Corynebacterium jeikeium K411
- Coxiella burnetii RSA 331 chromosome
- Escherichia coli str. K-12 substr. MG1655 chromosome
- Enterobacter aerogenes EA1509E
- Enterococcus faecalis OG1RF chromosome
- Haemophilus influenzae F3047
- Helicobacter pylori B38
- Klebsziella pneumoniae 342
- Lactobacillus fermentum F-6
- Legionella pneumophila str. Corby chromosome
- Leptospira interrogans serovar Copenhageni str. Fiocruz L1-130 chromosome I
- Listeria monocytogenes 08-5578
- Francisella tularensis subsp. holarctica LVS chromosome
- Micrococcus luteus NCTC 2665 chromosome
- Staphylococcus aureus subsp. aureus USA300 FPR3757
- Mycobacterium tuberculosis CAS/NITR204
- Mycoplasma pneumoniae M129
• *Neisseria meningitidis* MC58

• *Prevotella melaninogenica* ATCC 25845 chromosome chromosome I

• *Propionibacterium acnes* KPA171202

• *Proteus mirabilis* HI4320 chromosome

• *Pseudomonas aeruginosa* DK2 chromosome

• *Rickettsia rickettsii* str. Iowa chromosome

• *Salmonella enterica* subsp. enterica serovar Paratyphi A str. ATCC 9150 chromosome

• *Serratia proteamaculans* 568 chromosome

• *Shigella sonnei* Ss046, Staphylococcus aureus strain MS4

• *Staphylococcus epidermidis* ATCC 12228 chromosome

• *Staphylococcus equorum* strain KS1039

• *Staphylococcus haemolyticus* JCSC1435 DNA

• *Staphylococcus lugdunensis* HKU09-01

• *Staphylococcus saprophyticus* subsp. saprophyticus ATCC 15305 DNA

• *Staphylococcus xylosus* strain HKUOP18

• *Vibrio alginolyticus* NBRC 15630 ATCC 17749 chromosome 1

• *Yersinia pestis* CO92 chromosome
The bacteria in the test set are

- *Acinetobacter baumanii* strain HJ13
- *Borrelia miyamotoi* LB-2001
- *Burkholderia mallei* A188
- *Ensifer sojae* CCBAU 05684 chromosome
- *Enterococcus faecium* strain VREN1834
- *Escherichia coli* strain SN137 CVAPIL01c1
- *Escherichia coli* HVH 24 (4-5985145) aXtE-supercont1.1
- *Geobacillus kaustophilus* strain Et7/4 LG52.Contig21
- *Helicobacter pylori* SA157A
- *Lactobacillus plantarum* strain RI-011 20150709.B-R11 R1 contig 1
- *Mycobacterium tuberculosis* strain 2926STDY5723521
- *Pseudomonas aeruginosa* BWHPSA009 adggc-supercont1.1
- *Serratia marcescens* strain 2880STDY5682914
- *Shigella sonnei* strain 95234
- *Staphylococcus aureus* strain 3688STDY6125011
- *Staphylococcus aureus* M0084 adACT-supercont1.1
- *Staphylococcus aureus* W45750 adLVH-supercont1.1
• *Streptococcus mutans* DSM 20523 B093DRAFT scaffold 0.1 C

• *Streptococcus pneumoniae* strain SMRU1338

• *Streptomyces pathocidini* strain NRRL B-24287 B24287 contig 1

• *Thermus scotoductus* DSM 8553 F604DRAFT scaffold00001.1

• *Vibrio cholerae* HC-55B2 vcoHC55B2.contig.0

The viruses in the training set are

• Bacillus phage B103

• Ectromelia virus

• Sulfolobus virus Kamchatka 1

• Enterobacteria phage Sf6

• Mycobacterium phage L5

• Enterobacteria phage HK97

• Mushroom bacilliform virus

• Spinach latent virus

• Canine calicivirus

• Raspberry bushy dwarf virus RNA 1

• Grapevine virus A

• Staphylococcus phage 187
• Vaccinia virus

• Semliki forest virus

• Potato virus V

• Squash mosaic virus RNA 1

• Influenza A virus (A/Puerto Rico/8/1934(H1N1))

• Heterocapsa circularisquama RNA virus

• Potato virus X

• Enterococcus phage phiEF24C

• Cyclovirus NGchicken15/NGA/2009

• Pineapple bacilliform comosus virus

• Cyclovirus bat/USA/2009

• Cyclovirus PKgoat11/PAK/2009

• Avian leukemia virus isolate SCDY1

• Ralstonia phage PE226

• Vernonia yellow vein Fujian virus betasatellite

• French bean leaf curl virus isolate FbLCV-Kanpur segment DNA-A

• Avian paramyxovirus 4 strain APMV-4/duck/Delaware/549227/2010

• Florida woods cockroach-associated cyclovirus isolate GS140
- Equus ferus caballus papillomavirus type 4
- Nse virus strain F24/CI/2004
- Human rotavirus B strain Bang373 RNA dependent RNA polymerase (VP1) mRNA
- Meno virus strain E9/CI/2004
- Human cyclovirus VS5700009
- Sebokele virus 1 ORF1 gene for polyprotein
- Pseudomonas phage phiIBB-PAA2
- Tomato blistering mosaic virus isolate SC50
- Pseudomonas phage TL
- Escherichia phage vB EcoS FFH1
- Cyclovirus VN isolate hcf1
- Switchgrass mosaic-associated virus 1 isolate Cloud Nine movement protein
- Tortoise picornavirus strain 14-04
- Pseudomonas phage vB PaeP C2-10 Ab22
- Escherichia phage HY01
- Paenibacillus phage Vegas
- Paenibacillus phage Fern
- Pseudomonas phage PhiCHU
- Tilapia lake virus isolate Til-4-2011
- Avian paramyxovirus goose/Shimane/67/2000 viral cRNA
- Gordonia phage Splinter
- Staphylococcus phage Stau2
- Sanxia Water Strider Virus 4 strain
- Tacheng Tick Virus 6 strain
- Bastrovirus/VietNam/Rat/16715 10 non-structural polyprotein and structural polyprotein genes
- Ferret parechovirus isolate MpPeV1 polyprotein gene
- Bat cyclovirus GF-4c putative Rep and putative Cap genes
- Giant panda circovirus 1 strain gpci001
- Giant panda polyomavirus isolate GPPyV1
- Human fecal virus Jorvi4
- Pansavirus 1 strain gppn001
- Human fecal virus Jorvi2
- Fur seal picorna-like virus isolate KGI-Bel-22
- NY 014 poxvirus strain 2013
- Sus scrofa papillomavirus 2 isolate DE1018-16
- Wild melon vein banding virus isolate Su03-07
- Morelia viridis nidovirus strain S14-1323 MVNV
- Abisko virus isolate Abisko-6
- Wild vitis virus 1 isolate WVV1-NY1468
- Citrus concave gum-associated virus isolate CGW2 segment RNA1
- Myrmica scabrinodis virus 1 isolate Cambridge-Msc
- Porcine stool-associated circular virus/BEL/15V010 isolate 15V010
- Murmansk poxvirus strain LEIV-11411
- Actinidia virus 1 isolate K75
- Tomato leaf curl purple vein virus isolate BR:793:15 clone P793
- Lasius niger virus 1 isolate Cambridge-Lni
- Sudan watermelon mosaic virus isolate Su94-54
- Bastrovirus/VietNam/Bat/16715 78 non-structural polyprotein gene
- Felis catus papillomavirus type 5
- Metallosphaera turreted icosahedral virus strain MTIV1
- Solenopsis invicta virus 4 isolate Gainesville-Sin
- Porcine feces-associated gemycircularvirus isolate BEL/15V010
- Bastrovirus 7
- Tomato associated geminivirus 1 isolate Tomato BR
- Australian Anopheles totivirus isolate AATV 150840
- Squirrelpox virus Berlin 2015
- Tioga picorna-like virus 1
- Lasius neglectus virus 1 isolate Cambridge-Lne
- Bat mastadenovirus WIV18
- Macroptilium golden yellow mosaic virus isolate DR:M45:16
- Cyclovirus PK5006
- Stenotrophomonas phage PSH1
- Chequa iflavirus isolate A14-49.4
- Allium cepa amalgavirus 2 strain AcAV2-DH5225
- Flamingopox virus FGPVKD09
- Armigeres iflavirus gene for polyprotein
- Babaco mosaic virus isolate Tandapi
- Passerivirus sp. strain waxbill/DB01/HUN/2014
- Maize-associated totivirus 3 isolate ANETF3S2 gag-pol-like fusion protein
- Dianke virus strain SEN235030
• Rosellinia necatrix hypovirus 1 genomic RNA

• Rosellinia necatrix partitivirus 8 genomic RNA

The viruses in the test set are

• Canarypox virus

• Cowpox virus

• Duck hepatitis A virus 1 strain R85952

• Fowlpox virus

• Hepatitis C virus genotype 4

• Hepatitis E virus

• Monkeypox virus Zaire-96-I-16

• Parrot hepatitis B virus

• Seal Parapox virus isolate AFK76s1

• Swinepox virus

The fungi in the training set are

• Verticillium alfalfa VaMs.102 supercont1.26

• Ogataea parapolymorpha DL-1 chromosome I

• Wallemia mellicola CBS 633.66

• Marssonina brunnea f. sp. ‘multigermtubi’ MB m1
• *Wickerhamomyces ciferrii* WGS project CAIF00000000 data, strain NRRL Y-1031 F-60-10

• *Pseudozyma hubeiensis* SY62 DNA

• *Scedosporium apiospermum* strain IHEM 14462 chromosome

• *Fonsecaea erecta* strain CBS 125763

• *Aspergillus neoniger* CBS 115656

The fungi in the test set are

• *Kluyveromyces lactis* strain NRRL Y-1140 chromosome A

• *Cryptococcus gattii* WM276 chromosome A

• *Coniosporium apollinis* CBS 1000218 chromosome unknown

• *Anthracocystis flocculosa* PF-1

• *Aspergillus bombycis* strain NRRL26010 chromosome unknown

The bacteria in the test set of D1 are

• *Flammeovirgaeae bacterium* 311

• *Flavobacteriaceae bacterium* 3519-10

• *Flavobacterium branchiophilum* FL-15

• *Flavobacterium columnare* ATCC 49512

• *Flavobacterium indicum* GPTSA100-9
- *Flavobacterium johnsoniae* UW101
- *Flavobacterium psychrophilum* strain Z2
- *Flavobacterium psychrophilum* JIP02/86
- *Flexibacter litoralis* DSM 6794
- *Flexistipes sinusarabici* DSM 4947
- *Dactylococcopsis salina* PCC 8305
- *Dechloromonas aromatica* RCB
- *Dechlorosoma suillum* PS
- *Deferribacter desulfuricans SSM1 DNA*
- *Dehalobacter restrictus* DSM 9455
- *Dehalococcoides ethenogenes* 195
- *Dehalococcoides mccartyi* CG4
- *Dehalococcoides mccartyi* CG5
- *Dehalogenimonas lykanthroporepellens* BL-DC-9
- *Deinococcus deserti* VCD115
- *Deinococcus geothermalis* DSM 11300
- *Deinococcus gobiensis* I-0
- *Deinococcus maricopensis* DSM 21211
- *Deinococcus peraridilitoris* DSM 19664
- *Deinococcus proteolyticus* MRP
- *Thermoanaerobacter tengcongensis* MB4
- *Caldicellulosiruptor bescii* DSM 6725
- *Caldicellulosiruptor saccharolyticus* DSM 8903
- *Caldilinea aerophila* DSM 14535
- *Caldisericum exile* AZM16c01
- *Caldisphaera lagunensis* DSM 15908
- *Calditerrivibrio nitroreducens* DSM 19672
- *Caldithrix abyssi* DSM 13497
- *Caldivirga maquilingensis* IC-167
- *Campylobacter coli* RM4661
- *Marivirga tractuosa* DSM 4126
- *Martelella endophytica* strain YC6887
- *Megasphaera elsdenii* strain DSM 20460
- *Meiothermus ruber* DSM 1279
- *Meiothermus silvanus* DSM 9946
- *Melioribacter roseus* P3M
• *Melissococcus plutonius* ATCC 35311

• *Mesoplasma florum* L1 chromosome

• *Mesorhizobium ciceri* biovar biserrulæ WSM1271

• *Mesorhizobium ciceri* ca181 chromosome

• *Mesorhizobium loti* MAFF30399

• *Mesotoga prima* MesG1.Ag.4.2

• *Metallosphaera cuprina* Ar-4

• *Metallosphaera sedula* DSM 5348

• *Methanobacterium formicicum* strain BRM9

• *Acaryochloris marina* MBIC11017

• *Acetobacter ghanensis*

• *Acetobacterium woodii* DSM 1030

• *Acetobacter pasteurianus* IFO 3283-01

• *Acetobacter senegalensis*

• *Acetohalobium arabaticum* DSM 5501

• *Acholeplasma brassicae*

• *Acholeplasma laidlawii* PG-8A

• *Acholeplasma oculi* strain 19L
• *Acholeplasma palmae*

• *Achromobacter xylosoxidans* A8

• *Acidaminococcus fermentans* DSM 20731

• *Acidaminococcus intestini* RyC-MR95

• *Acidianus hospitalis* W1

• *Acidilobus saccharovorans* 345-15

• *Chlamydophila abortus* strain S26

• *Chlamydia avium* 10DC88

• *Chlamydophila felis* Fe

• *Chlamydia gallinacea* 08-1274

• *Chlamydia muridarum* Nigg

• *Chlamydophila psittaci* 6BC

• *Chlamydia trachomatis* 434

• *Chlamydia trachomatis* D

• *Chlamydophila abortus* strain S26

• *Chlamydia caviae* GPIC

• *Chlamydophila percourm* E58

• *Chlamydia pneumoniae* CWL029
• *Chloracidobacterium thermophilum* B

• *Chlorobaculum parvum* NCIB 8327

The bacteria in the test set of D2 are

• *Bacillus anthracis* 52-G

• *Bacillus anthracis* 8903-G

• *Bacillus anthracis* 9080-G

• *Bacillus anthracis* strain Cvac02

• *Bacillus anthracis* strain Han

• *Bacillus anthracis* strain Sterne 34F2

• *Bacillus anthracis* CZC5

• *Bacillus anthracis* strain ANSES-08-8-20 scaffold 25

• *Bacillus anthracis* strain ANSES-99-100 scaffold 24

• *Bacillus anthracis* strain ANSES-00-82 scaffold 22

• *Bacillus anthracis* strain STI-1 plasmid pX01 A0485-11

• *Bacillus anthracis* strain Zimbabwe 89

• *Bacillus anthracis* strain A.Br.003

• *Bacillus anthracis* strain 2000031006

• *Bacillus anthracis* strain 2000031023
• *Bacillus anthracis* strain 2000031027

• *Bacillus anthracis* strain 2000031038

• *Bacillus anthracis* strain 2000031031

• *Bacillus anthracis* strain 2000031039

• *Bacillus anthracis* strain 2000031052

The Bacillus organisms used in the small study are

• *Bacillus cereus* ATCC 14579

• *Bacillus anthracis* str. Ames chromosome

• *Bacillus cereus* E33L

• *Bacillus anthracis* str. CDC 684

• *Bacillus cereus* biovar anthracis str. CI

• *Bacillus anthracis* CZC5

The additional bacteria organisms used for creating a larger training data corpus are

• *Agrobacterium radiobacter* K84 chromosome 2

• *Alcanivorax dieselolei* B5

• *Alkalilimnicola ehrlichii* MLHE-1

• *Amycolatopsis orientalis* HCCB10007

• *Arthrospira platensis* NIES-39
• Azospirillum brasilense Sp245
• Candidatus Portiera aleyrodidarum BT-B
• Endosymbiont of Llaveia axin axin
• Arthrobacter arilaitensis RE117
• Helicobacter hepaticus ATCC 51449
• Lactobacillus mucosae LM1
• Marinomonas posidonica IVIA-Po-181
• Megasphaera elsdenii strain DSM 20460
• Methanosarcina barkeri str. Fusaro
• Methylobacterium nodulans ORS 2060
• Microcystis aeruginosa NIES-843
• Nakamurella multipartita DSN 44233
• Nocardia curiaceigorgica GUH-2 chromosome
• Octadecabacter temperatus strain SB1
• Oligotropha carboxidovorans OM5
• Pantoea rwandensis strain ND04
• Burkholderia xenovorans LB400 chromosome 3
• Paracoccus denitrificans PD1222 chromosome 1
• *Pediococcus pentosaceus* ATCC 25475
• *Planktothrix agardhii* NIVA-CYA 126
• *Porphyromonas gingivalis* W83
• *Pragia fontium* strain 24613
• *Prochlorococcus marinus* MED4
• *Providencia rettgeri* Dmel1 chromosome
• *Arthrobacter sulfonivorans* strain Ar51
• *Pseudoxanthomonas suwonensis* strain J1
• *Rhodobacter sphaeroides* 2.4.1 chromosome 1
• *Rhodobacter sphaeroides* 2.4.1 chromosome 1
• *Rhodomicrobium vannielii* ATCC 17100
• *Rhodospirillum centenum* SW
• *Rickettsia helvetica* C9P9 chromosome
• *Planctomyces brasiliensis* DSM 5305
• *Rufibacter tibetensis* strain 1351
• *Saccharomonospora viridis* DSM 43017
• *Saccharothrix espanaensis* DSM 44229
• *Selenomomas ruminantium* subsp. lactilytica TAM6421 DNA
• *Shewanella loihica* PV-4
• *Sideroxydans lithotrophicus* ES-1

• *Solitalea canadensis* DSM 3403

• *Spiroplasma cantharicola* strain CC-1

• *Stanieria cyanosphaera* PCC 7437

• *Stigmatella aurantiaca* DW4

• *Sulfolobus tokodaii* str. 7 DNA

• *Tannerella forsythia* 92A2

• *Thermobaculum terrenum* ATCC BAA-798

• *Thermosipho africanus* TCF52B

• *Thioalkalibibrio versutus* strain D301

• *Tistrella mobilis* KA081020-065

• *Treponema caldaria* DSM 7334

• *Veillonella parvula* DSM 2008

• *Wigglesworthia glossinidia* endosymbiont of Glossina brevipalpis

• Wolbachia endosymbiont of Drosophila melangoaster

• *Xenorhabdus bovienii* SS-2004 chromosome

• *Yersinia enterocolitica* subsp. enterocolitica 8081 chromosome

• *Zobellia galactanivorans* strain DsiJT chromosome

• *Zunongwangia profunda* SM-A87
Bibliography


actions on Computational Biology and Bioinformatics (TCBB), vol. 6, no. 1, pp. 46–61, 2009.

