# Reference-free Biomarker Mining in Metagenomic Data using Language Embedding

*By:*
Isha Agrawal- 5457777

in partial fulfilment of the requirements for the degree of

Master of Science

in Computer Science

at the Delft University of Technology

to be defended publicly on Friday April 21st, 2023 at 9:30 a.m.

| | | |
|---|---|---|
| Supervisor: | Dr. T. E. P. M. F. Abeel | TU Delft, Supervisor, Committee chair |
| Thesis committee: | Dr. R. Hai | TU Delft |
| | C. Peng | TU Delft, Daily supervisor |

An electronic version of this thesis is available at https://repository.tudelft.nl/

# PREFACE

This thesis report presents a culmination of my work towards obtaining the degree of Master of Science in Computer Science, in the Artificial Intelligence track and Bioinformatics specialization at the Delft University of Technology, The Netherlands. This project began with a discussion with my supervisor, Dr. Thomas Abeel, regarding the importance of understanding machine learning predictions, and its potential application in finding novel biomarkers in metagenomic data.

This report introduces the reader to the problem domain, existing research gaps, and the motivation behind exploring language embedding techniques. It presents an encoder-decoder approach to represent metagenomic samples perform prediction and find novel biomarkers. It is organized into four chapters, introducing the research direction, the steps required to reproduce the work, the results obtained, and the conclusions drawn from the observations.

I am deeply thankful to Dr. Thomas Abeel, for his excellent guidance and encouragement throughout the research process. I extend my sincere appreciation to my daily supervisor, Chengyao Peng, for her availability, insights, and amiable nature. Additionally, the AbeelLab team and the Pattern Recognition and Bioinformatics research group provided an engaging and stimulating work environment. Finally, I am grateful to my family and friends for their unwavering support and belief in my and my work.

I hope you enjoy reading this report as much as I cherished working towards it.

*Isha Agrawal*
*Delft, April 2023*

# ABSTRACT

Metagenomic Next-Generation Sequencing (mNGS) presents a promising avenue to generate massive volume of sequence reads in a short period of time. This has opened opportunities for disease diagnosis based on individual variations and mutations by considering the microbiome profile of each patient. However, the effective use of this data requires the design of appropriate algorithms which can closely represent the metagenomic data in an accurate and condensed manner.

In this work, we acknowledged the efficiency of current approaches such as reference-based methods and frequency encoding. However, we also recognized the limitations of current methods, such as limiting findings to pre-existing knowledge and inadequate representation of reads and metagenomic samples. Accordingly, we explored a natural language embedding technique, called Doc2vec, as a potential embedding approach for metagenomic study and phenotype prediction.

We introduced some modifications in the original Doc2Vec architecture to remove a bottleneck in analysing long reads. This was done by replacing kmer-level encoding with nucleotide-level representation. We used the embeddings obtained from this method as input to logistic classifier and ridge regression models. We compared the results with Kraken2 on colorectal cancer and type-2 diabetes classification, and for regression tasks on type-2 diabetes-related measures.

The results suggest a comparable performance between the proposed method and reference-based method for colorectal cancer classification. For type-2 diabetes dataset, reference-based method performs significantly better. In regression tasks to predict various metrics associated with type-2 diabetes, the proposed representation was comparable to reference-based method for some phenotypes, but lacked flexibility in others, indicating that the applicability of proposed approach strongly depends on the objective, dataset, and target phenotype.

The codes for reproducing the results and figures in this work have been made available at https://github.com/AbeelLab.

# Contents

# List of Tables

# LIST OF FIGURES

# 1

# INTRODUCTION

The early 2000s witnessed a massive success in Deoxyribose nucleid acid (DNA) and ribose nucleic acid (RNA) sequencing techniques, allowing the generation of the entire genome of microbial communities found in the environment [1]. This was achieved using metagenomic Next-Generation Sequencing (mNGS), a parallel sequencing technique which can rapidly and efficiently generate millions of sequence reads in a short period of time [2]. This has been used for diverse applications, such as variant discovery, disease diagnosis, and novel pathogen identification [3]–[5]. This presents a promising avenue to converge vast volume of data to generate new knowledge and advance clinical care and personalized treatment.

However, the effective use of this data requires the design and implementation of algorithms and high performance systems [6]. Consequently, computational techniques have been widely explored to answer biological questions. Accordingly, in this work, we explore the analysis of mNGS data from a computational perspective, specifically to diagnose diseases and make recommendations for precision medicine.

## 1.1. METAGENOMICS IS CRUCIAL FOR PRECISION MEDICINE

The affordability and flexibility of mNGS has led to an increased accessibility to microbiome profile information of each patient. Consequently, a paradigm shift has been observed from one-size-fits-all treatments to approaches tailored to individual variation and features, called precision medicine [7]–[9]. This is achieved by analyzing the host's microbial communities' DNA, and finding mutations or alternations, called biomarkers, to be targeted for treatment.

Recent years have presented several research works on the association of metagenomic data with health phenotypes [3], [10], [11]. These studies suggest that analysing metagenomic data, such as human gut microbiome [12], can indicate the presence of diseases such as type-2 diabetes [13], [14] and colorectal cancer [15], [16]. This motivates the focus on metagenomic data analysis and biomarker mining for this work.

## 1.2. REFERENCE-BASED METHODS INTRODUCE BIASES

One of the most common approaches for analyzing metagenomic data is metagenomic profiling, which predicts the relative abundance of various microbes in the sample [17]. This is usually achieved by aligning the sequences in the sample to a reference database [18] of virus, bacteria or archaea using tools like Kraken2 [19], and MetaPhlAn2 [20]. This yields a set of microbes and their frequency relative abundance in the dataset. This frequency distribution can then be used as input for prediction tasks such as linear regression and expectation maximization [21], and further explored through feature importance to find biomarkers. An overview of this process is shown in Figure 1.1.

While these methods have presented accurate results, using reference databases restricts the information and biomarkers found to the sequences already known in the domain [22]–[24]. This is because the genomic sequences identified are not limited to the variants and diversity [25] in the reference database. In addition, reference genomes can often be incomplete [26] or erroneous [27] due to low quality, leading to further challenges in identifying associated genes or microbes [24]. Another reference-based methods includes the high computational overhead of assembling reads to genomes [28], [29].

Alternatively, several works employ reference-free methods to generate new knowledge without restricting the approach to a pre-existing database. This includes modeling the frequency distribution using k-mers [30]–[32] and counting GC bias [32] for similarity analysis.

However, GC biases often under-represent GC-poor organisms [33]. In practice, the most commonly found reference-free encoding approach is frequency [34], [35]. This usually involves computing and clustering the relative abundance of the reads or kmers. The ease and efficiency of frequency encoding makes it a popular approach, and can be made scalable using hashing [36]. Yet, while frequency is an efficient method, it reduces the sequence to a single scalar value. This raises the question of whether an alternate embedding approach can be proposed which is a better representation of the sequence.

## 1.3. DOC2VEC AS A PROPOSED REPRESENTATION FOR METAGENOME EMBEDDING

Unlike data modalities such as gene expression and relative abundance, raw reads are not in a numerical format. Hence, an encoding strategy is needed to represent characters prior to any analysis.

Accordingly, natural language based approaches were of interest in this work. They are scalable, as they have been developed for large corpus of data, and can often be run in parallel, such as attention models [37]. Several research works have widely used natural language based approaches to embed sequences into vectors which represent the data based on the context they are observed in [38]–[42]. However, most of these methods analyze transcribed [38], translated [39]–[41], or targeted data [32], [42], [43], and have not been actively explored for unaligned raw reads in metagenomic samples.

In this work, we considered natural language to be analogous to biological datasets by representing sequence reads as words and metagenomic samples as documents. This analogy is closely represented by a word2vec [44] based method, called Doc2Vec [45].

Word2vec, which is a natural language method, and is widely applied to generate

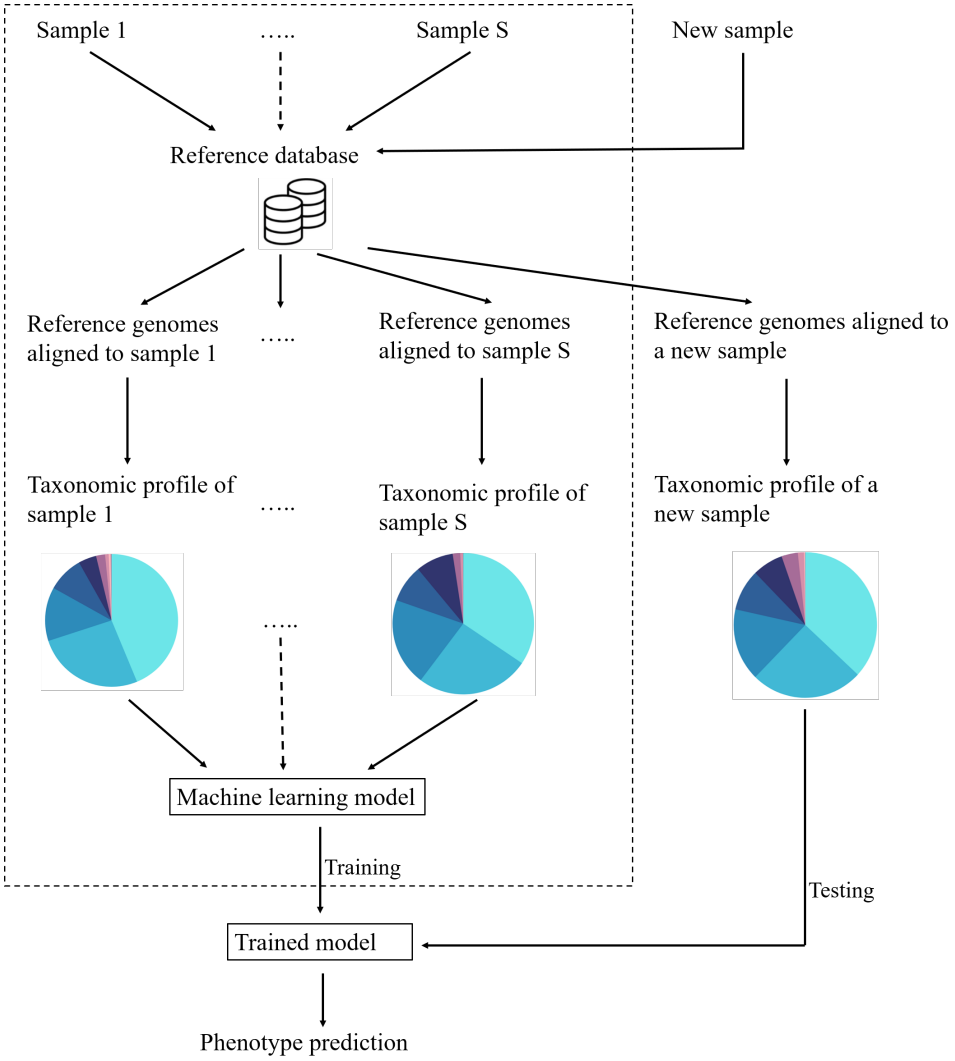



Figure 1.1: An overview of reference-based approaches for metagenome analysis.

word embeddings. In the recent years, Word2vec has been frequently used to embed omic sequences as well [46], [47]. Its proven efficiency in embedding omic data motivated the hypothesis to evaluate Doc2Vec for metagenome data to generate context-based representations of reads and samples.

Thus, in this research work, we recognized that reference-free methods offer a vast opportunity to explain uncultured data in the biosphere [48], but have been largely underdeveloped. Currently, the most commonly used reference-free embedding in practice is frequency encoding, which condenses a k-mer to its relative abundance. This presents an avenue to develop more meaningful representations, such as context-based embeddings. This directed our research focus towards natural language based methods, specifically Doc2Vec, due to its intuitive analogy and the proven efficiency of Word2vec. Further, we overcame some bottlenecks of the original Doc2vec and Word2vec models, as further discussed in Chapter 3.

## 1.4. RESEARCH QUESTIONS

With the goal to predict phenotypes and mine novel biomarkers based on metagenomic data using Doc2vec, we proposed the following two research directions.

1. Is the predictive power of proposed representation comparable to reference-based methods?
   In this thesis, we proposed Doc2vec as a representation to metagenomic data. This is based on the need for a better embedding than relative abundance, and the observed efficiency of Word2vec. However, while Doc2vec presents a vast opportunity to generate context-based embeddings and find biomarkers, the large number of variables and features can make it challenging to optimize in practice. This is in contrast to reference-based methods which analyze a small set of already discovered data, and align it to a finite reference database. Accordingly, with this research question, we explored the viability of the proposed representation for phenotype prediction.

2. Can the sequences obtained by feature importance be interpreted as biomarkers?
   As the goal of finding biomarkers is to propose them for targeted treatment, it is crucial to first establish their reliability. We did this by comparing the proposed markers with existing domain knowledge, where a significant overlap between the two sets of biomarkers would bolster the credibility of the novel biomarkers found in this work.

# 2

# MATERIALS AND METHODS

This chapter delineates the steps and metrics we used to conduct the experiments. This includes selection and analysis strategy for datasets, evaluation metrics, and reference methods. This is followed by system details used to produce this work.

## 2.1. DATASET SELECTION METHODOLOGY

To assess the proposed representation, we searched the following archives to find public shotgun metagenomic sequencing datasets.

1. National Center for Biotechnology Information (NCBI)[1] - We used the following query to find appropriate datasets in the "BioProjects" database.

   *"raw sequence reads"[Project Data Type] AND "metagenome"[Project Data Type]*

2. Human Microbiome Project (HMP)[2] - The query used in HMP is as follows.

   *Format IS FASTQ AND Node Type IS wgs_raw_seq_set*

3. European Nucleotide Archive (ENA)[3] - We used ENA to download samples and metadata of datasets found in NCBI and HMP. This information is available in the “Generated FASTQ files: FTP” and “Sample Title” columns of TSV reports of the bioprojects.

4. Apart from archives, we looked into research articles [49], [50] based on metagenomic raw data to find the datasets used by them.

[1] https://www.ncbi.nlm.nih.gov/: Accessed on 28 March 2023
[2] https://portal.hmpdacc.org/search: Accessed on 28 March 2023
[3] https://www.ebi.ac.uk/ena/browser/home: Accessed on 28 March 2023

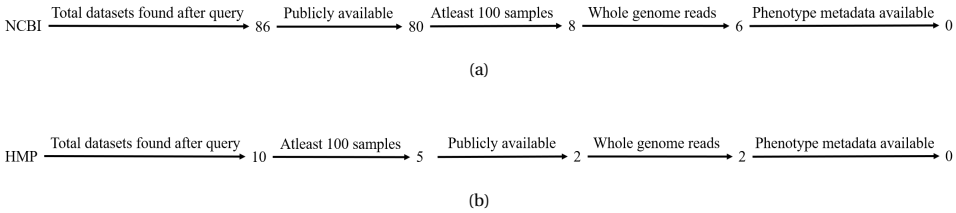



Figure 2.1: Number of datasets found in (a) NCBI and (b) HMP archives after applying each constraint.

Table 2.1: Details of colorectal cancer dataset (Bioproject accession ID - PRJEB7774) used to analyse the performance of proposed representation

| | |
|---|---|
| **Study title** | Gut microbiome development along the colorectal adenoma-carcinoma sequence |
| **Source** | Gut microbiota in stool |
| **Diagnosis** | Colorectal cancer - healthy controls, advanced adenoma, carcinoma |
| **Number of samples** | Total samples = 156 |
| **Year** | 2015 - 2016 |

Table 2.2: Details of type-2 diabetes dataset (Bioproject accession ID - PRJNA422434) used to analyse the performance of proposed representation

| | |
|---|---|
| **Study title** | A metagenome-wide association study of gut microbiota in type 2 diabetes |
| **Source** | Gut microbiota in stool |
| **Diagnosis** | Type-2 diabetes - healthy controls, diabetes<br>Related measures- Body mass index, fasting blood glucose, fasting serum insulin, systolic and diastolic blood pressure, triglyceride level, low and high density lipoprotein levels, total cholesterol, and glycosylated hemoglobin level |
| **Number of samples** | Total samples = 95 |
| **Year** | 2017 - 2022 |

We selected datasets with at least 100 samples to ensure that the machine learning model had sufficient samples for training, validation, and testing. This number was chosen arbitrarily. Datasets without phenotype information such as diagnosis, disease stage, and health measurements in the metadata were discarded. As shown in Figure 2.1, no datasets found in these archives fulfilled all the constraints.

Finally, we selected the following two datasets from research articles [50], [51].

1. Colorectal cancer (PRJEB7774) [4]- As shown in Table 2.1, we used this dataset to evaluate the performance of proposed representation in classifying patients to carcinoma, adenoma, and control labels.

2. Type-2 diabetes (PRJNA422434) [5]- As shown in Table 2.2, we used type-2 diabetes dataset for two tasks. The first was classification of samples into type-2 diabetes and control labels. The second task was a set of regression problems to predict several type-2 diabetes related measure, such as body mass index and triglyceride levels.

[4] https://www.ebi.ac.uk/ena/browser/view/PRJEB7774: Accessed on 28 March 2023
[5] https://www.ebi.ac.uk/ena/browser/view/PRJNA422434: Accessed on 28 March 2023

## 2.2. DATA PRE-PROCESSING

We analyzed the selected datasets using the below mentioned steps.

1. Quality control- We assessed each sample using FastQC v0.11.7[6]. Samples which failed in any category were discarded.

2. Assessing potential batch effect- Using Kraken2 v2.1.2[7], we generated vectors of relative abundance at species rank for each samples. These vectors were then visualized using the first two components of TSNE from scikit-learn v1.2.2 in Python v3.9.16. Following this, we applied K-means clustering using scikit-learn for three clusters with ten random initializations. The trueness of the clusters obtained was evaluated using *Adjusted rand index* and *Adjusted mutual information index*. If a value close to 0 was obtained, we concluded that the dataset has no batch effects.

3. Rarefaction- To avoid sequencing depth bias, we truncated the number of reads in each sample to the number of reads in the smallest sample in that dataset. Prior to truncation, we shuffled the reads using Python's inbuilt *.shuffle()* function in the *random* library, with random seed set to zero.

4. Removing reads based on length- For a hyperparameter setting where k-mer length is k, and window distance at each end is w, we discarded reads shorter than $(2 * w) + k$ from the sample.

5. Handling reads with missing bases- We discarded reads with "N" denoting unknown nucleotide from the sample.

6. Choosing single-end reads- Only single-end reads were chosen for this work. This was an arbitrary decision based on ease of use.

7. Balanced data- For classification problems, we balanced the datasets by discarding randomly selected extra samples from each label.

## 2.3. EVALUATING THE GENERATED EMBEDDINGS

To evaluate and compare the performance of the embeddings obtained from the proposed representation and a typical reference-based approach, we applied for several classification and regression tasks.We analysed the results using the metrics discussed below.

1. Evaluation metrics for classification- Classification models classify each object $x$ to a class $i$, where $i \in \{0, 1, ..., C-1\}$, and $C$ is the total number of classes or labels. Based on the number of correctly and incorrectly classified samples, true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are computed, as shown below.

[6] https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
[7] https://ccb.jhu.edu/software/kraken2/

$$\begin{array}{c} \\ Actual\ Positive \\ Actual\ Negative \end{array} \begin{array}{c} \begin{array}{cc} Predicted\ Positive & Predicted\ Negative \end{array} \\ \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \end{array}$$

These measures are combined to represent different aspects of the model performance [52]. As shown in Table 2.3, we used accuracy, precision, recall, f1 score, area under the curve, and confusion matrix to obtain a complete understanding of the model performance.

Table 2.3: Evaluation metrics for classification

| Metric | Formula | Evaluation focus |
|---|---|---|
| Accuracy | $\frac{\sum_{i=0}^{C-1} \frac{TP_i+TN_i}{TP_i+TN_i+FP_i+FN_i}}{C}$ | Overall efficiency of the classifier by considering the number of correctly classified samples |
| Precision | $\frac{TP}{TP+FP}$ | Accuracy of positive class prediction |
| Recall | $\frac{TP}{TP+FN}$ | Completeness of positive classes prediction |
| F1 Score | $\frac{2*Precision*Recall}{Precision+Recall}$ | Combines precision and recall into a single metric by taking their harmonic mean |
| Area under the curve | $\frac{1}{2}\left[\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right]$ | Aggregate measure of model's performance over various classification thresholds, and represents the model's ability to avoid false classification |
| Confusion matrix | $\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix}$ | Gives information about how the model is confusing one class for another |

2. Evaluation metrics for regression- For each input variable $x_i$, where $i \in \{0,...,N-1\}$, and $N$ is the total number of samples, regression models predict a target value $\hat{y}_i$. The difference between this predicted value $\hat{y}_i$ and the ground truth $y_i$ reflects the efficiency of the model. We measure this difference with mean squared error and Pearson correlation coefficient between true and predicted data, as shown in Table 2.4.

Table 2.4: Evaluation metrics for regression

| Metric | Formula | Evaluation focus |
|---|---|---|
| Pearson correlation coefficient | $\frac{\sum_{i=0}^{N-1}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=0}^{N-1}(x_i-\overline{x})^2\sum_{i=0}^{N-1}(y_i-\overline{y})^2}}$ Here, $\overline{x}$ and $\overline{y}$ represent the mean of the input data and the ground truth respectively. | Measures the strength of association between predicted data and true data. |
| Mean squared error | $\frac{1}{N}\sum_{i=0}^{N-1}(y_i-\hat{y}_i)$ | Computes the average squared distance between the predicted and true data, and penalizes large errors and outliers |



## 2.4. COSINE SIMILARITY TO FIND BIOMARKER K-MERS

Cosine similarity is a commonly used metric to measure distance in high-dimensional space. We obtained biomarkers of each sample by finding the k-mers whose embeddings have the highest cosine similarity with their respective metagenomic sample embeddings. After training, we generated k-mer embeddings using the trained weights, and computer their cosine similarity against the corresponding sample embedding, as shown in the equation below for two embedding vectors $A$ and $B$.

$$\text{Cosine similarity} \quad S_c(A,B) = \frac{A.B}{||A||.||B||} = \frac{\sum_{i=0}^{N-1} A_i B_i}{\sqrt{\sum_{i=0}^{N-1} A_i^2}\sqrt{\sum_{i=0}^{N-1} B_i^2}}$$

## 2.5. REFERENCE-BASED METHOD FOR COMPARISON

We explored several reference-based methods to find existing approaches and packages for comparison with the proposed representation. Based on the limited resources available for this work, we imposed the following constraints.

1. Research articles- The source code should be available, executable without major changes, and should finish running within 36 hours.

2. Tools and software- The setup should not take more than 36 hours and a maximum of 500 GB memory.

Of the several methods found, as shown in Table 2.5, only Kraken2 [19] could be successfully set up. Following this, we set up a typical reference-based approach, such as in Figure 1.1, for comparison, as explained below.

Table 2.5: Some reference-based methods explored for comparison with the proposed representation.

| Method | Description | Constraints in set up |
|---|---|---|
| Kraken2 [19] | Improves the memory constraint of Kraken's tree-based k-mer alignment approach [53] by using 32-bit hash cells to store key-value pairs | - |
| MetaPhlan2 [20] | Uses clade-specific marker genes for metagenome classification | Required more than 500 GB |
| Centrifuge [21] | Applies Burrows-Wheeler transform [54] and the Ferragina-Manzini index [55] to classify metagenome into microbial species | Dependency issues with Linux system while setting up Bowtie [56] |
| CLARK [57] | Generates hash tables to create sets of discriminative k-mers for fast classification of metagenomic reads | Took more than 36 hours to set up the refence databases |
| MetaML [58] | Uses metagenomic profiles from MetaPhlan2 [20] as input of machine learning classifiers for metagenome-based prediction task | Needs MetaPhlan2 [20] to run, which coult be setup due to high memory requirements |
| Deep Microbes [59] | Applies deep learning for genus identification and abundance estimation | Took longer than 36 hours to set up the reference databases |

1. Training Kraken2 output- We used the relative abundance of Kraken2's taxonomic profile at species rank as input to machine learning models.

2. Normal distribution- For regression, we created a normal distribution using the mean and standard deviation of various features available in the metadata. Values from this distribution were then sampled for each test instance to evaluate if the performance of proposed representation was better than average value. This normal distribution was created using *.random.normal()* function of Python's *numpy* v1.23.4 package.

For classification, we trained logistic classifier with five-fold cross validation using scikit-learn's *LogisticRegressionCV* function. For regression, ridge regression with five-fold cross-validation was applied using scikit-learn's *RidgeCV* function. For all the experiments, k-mer length was set to 31.

## 2.6. SYSTEM DETAILS

All experiments were run on Linux CentOS-7 inside Slurm v21.08.8-2 cluster. A *requirements.txt* file with a list of all Python packages and their versions used in this work is available at `https://github.com/AbeelLab`.

# 3

# RESULTS AND DISCUSSIONS

This chapter discusses the architecture we proposed in this work to generate embeddings or representations for metagenomic data. It beings with an explanation of the proposed approach, followed by its performance evaluation, a review of some limitations, and discussions regarding the research questions posed in this work.

## 3.1. PROPOSED METHOD

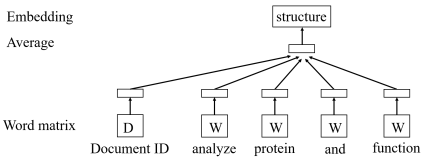
Most of the commonly used approaches for raw read analysis rely on reference databases. This restricts the possibility of finding novel markers, and can introduce errors if the database is incomplete or inaccurate. On the other hand, current reference-free methods often lose a significant amount of information, such as in frequency embeddings. Accordingly, a method which considers context information is required. In this work, we proposed Doc2Vec [45], a popularly used natural language model, to learn meaningful representations of metagenome samples.

A metagonome sample consists from hundreds of thousands to millions of omic reads. To use Doc2vec in such a setting , we considered omic reads, such as "ACTCCGACCTGCT", as a natural language sentence, like "how to analyze protein structure and function for enzymes", as shown in Figure 3.1. The key components of this approach are explained below, followed by the details on input encoding and the final architecture proposed in this work to generate representations.

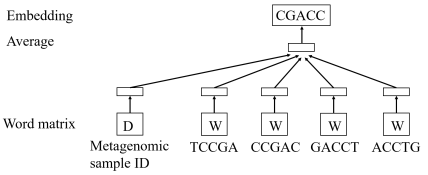
### 3.1.1. KEY COMPONENTS

The key components of Doc2vec we used in this work are context, target, encoder, and decoder, as delineated below.

1. Window size- We analysed reads (sentences) in a sliding window manner, by moving frames across the data in steps of one. The size of this window is user-defined, and can be considered 5 for this example. It represents the maximum distance between a central value and its neighborhood. Thus, for a window distance $w$ on each side, the total length of the window is $N = 2w$.




(a) Doc2vec for natural language embedding




(b) Doc2vec for metagenome embedding

Figure 3.1: (a) Architecture of Doc2vec model for an example phrase "analyze protein structure and function". Here, "analyze", "protein", "and", and "function", are context inputs to predict the embedding of the target word "structure". $W$ and $D$ represent weight matrices for words and documents respectively. (b) Architecture of Doc2vec for metagenomic embedding for an example sequence "TCCGACCTG". Here, "CGACC" is the target k-mer, and "TCCGA", "CCGAC", "GACCT", and "ACCTG" are its context k-mers.



2. Target- Within a window, the objective of doc2vec was to find k-mer (word) embeddings. Each window focuses on generating embedding of the k-mer (word) present at its centre.

   For example, target word in the window "analyze protein structure and function" is "structure" as it is present at the centre. Similarly, in the window "TCCGACCTG", the target k-mer is "CGACC" for a user defined k-mer length of 5.

3. Context- Words surrounding the target word within a maximum window distance form its context. For a window size of $w$, number of context words are $N = 2w$.

   For example, context words for the window "analyze protein structure and function" are "analyze", "protein", "and", and "function". Similarly, context words for the window "TCCGACCTG" are "TCCGA", "CCGAC", "GACCT", and "ACCTG".

4. Sample- We assigned a unique index $i$ to each sample, where $i \in \{0, S-1\}$, and $S$ is the total number of metagenomic samples (documents) in the dataset.

5. Encoder- As the name suggests, the encoder encodes the input into an embedding. Here, the input are the context words and sample representation.

6. Decoder- The decoder decodes the embedding generated by the encoder to output the target word. Thus, Doc2vec in the proposed approach reads context and sample information to generate embedding of the target word at the centre of the window frame.

### 3.1.2. INPUT REPRESENTATION

Alike other reference-free methods, we k-merised our metagenomic samples. However, as k-mers are in text format, they cannot be directly used as inputs. Accordingly, we needed numerical representations of the k-mers and samples.

1. Sample- For a dataset with $S$ samples, each sample was assigned a unique index $i, i \in \{0, S-1\}$. Following this, each sample was represented by a one hot vector of size $1 \times S$, as done in the original algorithm of doc2vec.

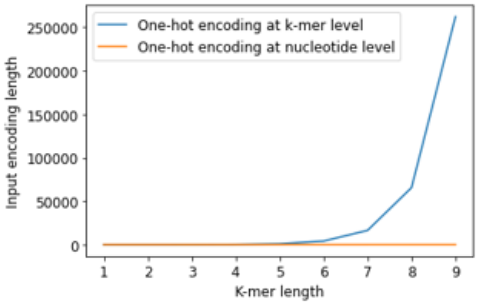



Figure 3.2: Encoding length with Doc2vec's original approach increases exponentially. Modifying the input representation to nucleotide level encoding in the proposed approach significantly reduced the number of parameters.

2. k-mers- The original paper of doc2vec creates a vocabulary of all the words in the dataset, containing a total of $V$ words. Each word $w$ is then assigned a unique index $i, i \in \{0, V-1\}$. Following this, each word is represented by a one hot vector of dimension $1 \times V$. Most of the natural language documents have a vocabulary size of 10,000 to 100,000. However, the number of possible k-mers is much higher than this order. For example, a k-mer length of 11 leads to $4^7 \sim 4.19 \times 10^6$ possible k-mers. This work considered k-mers of length 31 as it is most commonly found in literature [60]. This resulted in a potential vocabulary size of $4^7 \sim 4.61 \times 10^{18}$. A one hot encoding of this order is computationally impractical, and an alternate approach was needed.

   We addressed this bottleneck by changing the input representation of k-mers. Inspired by Chen et al. [61], instead of assigning a unique vector to each k-mer, input encoding was performed at nucleotide level, as shown below.

   A - [1 0 0 0]
   C - [0 1 0 0]
   G - [0 0 1 0]
   T - [0 0 0 1]

   Thus, for a given target k-mer "CGACC", the encoding is be given by

$$\begin{matrix} & C & G & A & C & C \end{matrix}$$
$$Input\ Encoding \begin{bmatrix} 0100 & 0010 & 1000 & 0100 & 0100 \end{bmatrix}$$

   Consequently, for a k-mer of length $k$, the input representation length reduced from $4^k$, to $4 \times k$, as can be observed in Figure 3.2.

### 3.1.3. ARCHITECTURE

The dense neural network architecture has three transformation matrices. These are for sample, context k-mers, and decoding, as explained below.

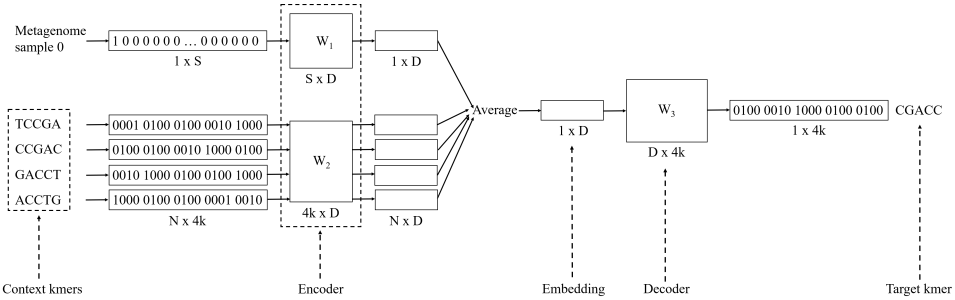



Figure 3.3: Doc2vec architecture for proposed representation. $W_1$, $W_2$, and $W_3$ represent sample encoding, k-mer encoding, and decoding matrices respectively. $D$ is the embedding length hyperparameter.

1. Sample embedding matrix $W_1$- Each row in this matrix of dimension $S \times D$ represents a unique sample $S_i, i \in \{0, S-1\}$. This matrix is shared by all samples.

2. Context embedding matrix $W_2$- The k-mers are mapped to an embedding of size $D$. This matrix is shared by all k-mers irrespective of the sample they are obtained from.

3. Decoding matrix $W_3$- The embedding of size $D$ is decoded to obtain the target k-mer using a transformation matrix of size $D \times 4k$.

4. Training- During training, reads from each sample are parsed in a sliding window manner to obtain target and context k-mers. These make the input and ground truth of the model. Along with context k-mers, sample encoding is used as input. Thus in every instance, the sample embedding matrix is trained along with the context k-mer matrix. Consequently, for multiple training instances of a given sample, context k-mers change, but the sample encoding remains the same, thereby acting as a memory.

## 3.2. COLORECTAL CANCER CLASSIFICATION

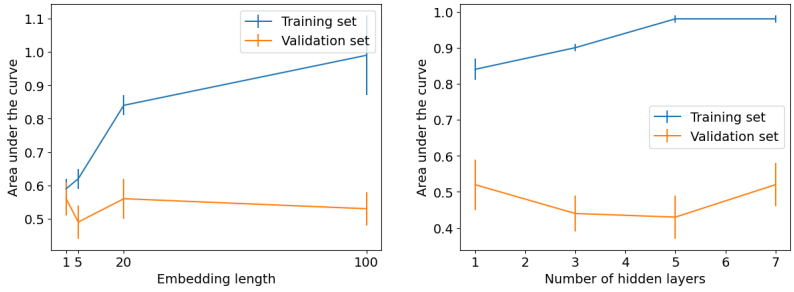
This section evaluates the proposed representation on various hyperparameter settings, and tests its predictive power compared to Kraken2 on unseen data.

### 3.2.1. HYPERPARAMETERS

1. **Performance oscillates as embedding length increases**
   Table 3.1 presents the performance of the proposed representation averaged over 100 randomly sampled validation sets. It can be seen that there is no clear trend in the performance, such as a monotonous increase or decrease in the outcomes, as embedding length increases. With an increase an embedding length, the model has more parameters and flexibility to fit the input data. This would suggest that increasing embedding length should improve performance, until a threshold beyond which the model will exhibit overfitting. Although training error suggests

Table 3.1: Validation performance of proposed representation with different embedding lengths and one hidden layer for colorectal cancer classification. The embeddings were trained using five-fold cross-validation logistic regression model. The reported results are an average and standard deviation of performance on 100 randomly sampled validation sets of size 30.

| Embedding length | Accuracy | Precision | Recall | F1 | Area under the curve | Confusion Matrix |
|---|---|---|---|---|---|---|
| 1 | $0.38 \pm 0.05$ | $0.29 \pm 0.11$ | $0.37 \pm 0.05$ | $0.31 \pm 0.07$ | $0.56 \pm 0.05$ | $\begin{bmatrix} 5.71 \pm 2.14 & 3.21 \pm 2.10 & 1.08 \pm 2.12 \\ 3.60 \pm 1.88 & 4.76 \pm 2.41 & 1.64 \pm 2.64 \\ 4.41 \pm 2.07 & 4.61 \pm 2.50 & 0.98 \pm 2.23 \end{bmatrix}$ |
| 5 | $0.32 \pm 0.07$ | $0.32 \pm 0.10$ | $0.32 \pm 0.07$ | $0.30 \pm 0.08$ | $0.49 \pm 0.05$ | $\begin{bmatrix} 3.49 \pm 1.82 & 3.16 \pm 1.91 & 3.35 \pm 2.21 \\ 3.20 \pm 2.08 & 3.04 \pm 1.85 & 3.76 \pm 2.26 \\ 3.22 \pm 1.75 & 3.62 \pm 2.04 & 3.16 \pm 1.87 \end{bmatrix}$ |
| 20 | $0.38 \pm 0.08$ | $0.39 \pm 0.09$ | $0.38 \pm 0.08$ | $0.38 \pm 0.08$ | $0.56 \pm 0.06$ | $\begin{bmatrix} 3.89 \pm 1.44 & 3.72 \pm 3.72 & 2.39 \pm 2.39 \\ 3.65 \pm 3.65 & 3.07 \pm 3.07 & 3.28 \pm 3.28 \\ 2.23 \pm 2.23 & 3.20 \pm 3.20 & 4.57 \pm 4.57 \end{bmatrix}$ |
| 100 | $0.35 \pm 0.08$ | $0.35 \pm 0.09$ | $0.35 \pm 0.08$ | $0.34 \pm 0.08$ | $0.53 \pm 0.06$ | $\begin{bmatrix} 3.58 \pm 1.42 & 3.18 \pm 1.43 & 3.24 \pm 1.54 \\ 3.53 \pm 1.59 & 3.45 \pm 1.45 & 3.02 \pm 1.54 \\ 3.74 \pm 1.68 & 2.86 \pm 1.33 & 3.40 \pm 1.43 \end{bmatrix}$ |
| 500 | $0.37 \pm 0.06$ | $0.30 \pm 0.11$ | $0.37 \pm 0.06$ | $0.31 \pm 0.06$ | $0.56 \pm 0.05$ | $\begin{bmatrix} 5.99 \pm 1.69 & 2.92 \pm 1.87 & 1.09 \pm 1.62 \\ 3.83 \pm 1.87 & 4.29 \pm 2.33 & 1.88 \pm 2.60 \\ 4.81 \pm 1.75 & 4.28 \pm 2.32 & 0.91 \pm 1.60 \end{bmatrix}$ |
| 1000 | $0.34 \pm 0.07$ | $0.33 \pm 0.08$ | $0.34 \pm 0.07$ | $0.33 \pm 0.08$ | $0.52 \pm 0.05$ | $\begin{bmatrix} 3.76 \pm 1.50 & 3.10 \pm 1.50 & 3.14 \pm 1.60 \\ 3.14 \pm 1.41 & 3.61 \pm 1.46 & 3.25 \pm 1.38 \\ 3.50 \pm 1.51 & 3.65 \pm 1.48 & 2.85 \pm 1.45 \end{bmatrix}$ |
| 1500 | $0.30 \pm 0.08$ | $0.29 \pm 0.09$ | $0.30 \pm 0.08$ | $0.28 \pm 0.08$ | $0.48 \pm 0.06$ | $\begin{bmatrix} 1.93 \pm 1.93 & 3.52 \pm 3.52 & 4.55 \pm 4.55 \\ 3.45 \pm 3.45 & 4.37 \pm 4.37 & 2.18 \pm 2.18 \\ 4.06 \pm 4.06 & 3.32 \pm 3.32 & 2.62 \pm 2.62 \end{bmatrix}$ |




(a) Performance of proposed representation on training and validation set over different embedding lengths

(b) Performance of proposed representation on training and validation set over different number of hidden layers

Figure 3.4: Training and validation performance of the proposed representation in colorectal cancer [50] classification for different hyperparameter settings.

overfitting, as evident in Figure 3.4a and Table A.3, no clear decline in validation performance is observed. One possible reason could be that some embedding lengths offer better data separability, leading to easier classification. Similarly, some hyperparameter settings, such as an embedding length of 500, might have low data separability, thereby resulting in a significant drop in model performance on training set. However, this is difficult to verify as the high dimensional data cannot be visualized, and the few existing methods on quantifying separability of data classes do not have a publicly available source code [62].

An interesting observation is that embedding length of 1 appears to perform the best. This suggests than an entire metagenome sample containing thousands of

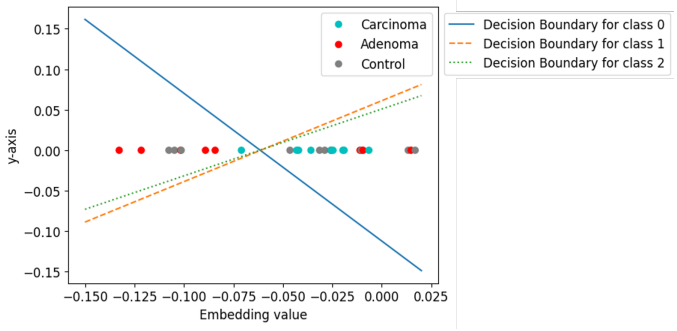



Figure 3.5: Embeddings of length 1 generated with proposed representation approach and one hidden layer for colorectal cancer [50] classification. The decision boundaries were obtained in a one-versus-all approach to classify carcinoma, adenoma, and control, represented as class 0, 1, and 2 respectively.



omic reads can be represented by one value. This is counter-intuitive, especially as no clear distinction is visible between the embeddings, as shown in Figure 3.5.

2. **Increasing model depth can lead to overfitting**
   To evaluate the effect of model depth, we introduced more hidden layers to the architecture, as illustrated in Appendix A.7. The architecture is inspired from U-Net model [63] to incrementally modify data dimension. Table 3.2 presents the observed performance of the proposed representation with embedding length of 1000 for different number of hidden layers. A decline in the performance from one hidden layer to five hidden layers is observed. This is likely due to overfitting in the model as the number of parameters available to fit the input data has increased [64]. This can be confirmed from Figure 3.4b and Table A.4, which present the performance of the proposed approach on training data as well. The results on training data clearly demonstrate that the model is overfitting as the number of layers increases. However, this does not explain the improvement in performance for the hyperparameter setting of seven hidden layers. Similar to embedding length, a possible explanation could be that data separability increases for certain hyperparameter settings.

### 3.2.2. PERFORMANCE ON TEST DATA

Based on the hyperparameter evaluations, we set embedding length and number of hidden layers to 1 to generate test results. The results obtained using proposed representation were compared with Kraken2 to classify colorectal cancer [50] into three classes, namely carcinoma, adenoma, and control, with labels 0, 1, and 2 respectively.

Table 3.3 presents the performance of the proposed representation and Kraken2 representation on unseen data. The results were averaged over 100 randomly sampled test sets of size 18. We see no distinctive difference in performance between the two approaches, as also evident from Figure 3.6.

Although the accuracy, recall, and F1 scores of the proposed approach are higher than Kraken2's representation, the confusion matrix reveals that it does not classify any

Table 3.2: Validation performance of proposed representation with different number of hidden layers and an embedding length of 1000 for colorectal cancer classification. The embedding length is 1, and the representations were trained using five-fold cross-validation logistic regression model. The reported results are an average and standard deviation of performance on 100 randomly sampled validation sets of size 30.

| Number of hidden layers | Accuracy | Precision | Recall | F1 | Area under the curve | Confusion Matrix |
|---|---|---|---|---|---|---|
| 1 | $0.34 \pm 0.10$ | $0.34 \pm 0.10$ | $0.34 \pm 0.10$ | $0.33 \pm 0.10$ | $0.52 \pm 0.07$ | $\begin{bmatrix} 3.50 \pm 1.66 & 3.00 \pm 1.52 & 3.50 \pm 1.53 \\ 3.35 \pm 1.62 & 3.80 \pm 1.47 & 2.85 \pm 1.31 \\ 3.45 \pm 1.60 & 3.75 \pm 1.41 & 2.80 \pm 1.50 \end{bmatrix}$ |
| 3 | $0.28 \pm 0.04$ | $0.30 \pm 0.08$ | $0.28 \pm 0.04$ | $0.27 \pm 0.04$ | $0.44 \pm 0.05$ | $\begin{bmatrix} 2.85 \pm 1.46 & 4.45 \pm 1.80 & 2.70 \pm 1.85 \\ 3.20 \pm 1.60 & 2.15 \pm 1.31 & 4.65 \pm 1.82 \\ 2.25 \pm 1.18 & 4.35 \pm 1.46 & 3.40 \pm 1.50 \end{bmatrix}$ |
| 5 | $0.27 \pm 0.07$ | $0.27 \pm 0.11$ | $0.27 \pm 0.08$ | $0.25 \pm 0.08$ | $0.43 \pm 0.06$ | $\begin{bmatrix} 2.30 \pm 1.49 & 4.25 \pm 2.23 & 3.45 \pm 2.13 \\ 3.15 \pm 2.15 & 4.10 \pm 2.07 & 2.75 \pm 1.67 \\ 3.05 \pm 1.66 & 5.15 \pm 2.10 & 1.80 \pm 1.21 \end{bmatrix}$ |
| 7 | $0.35 \pm 0.07$ | $0.37 \pm 0.07$ | $0.36 \pm 0.07$ | $0.36 \pm 0.07$ | $0.52 \pm 0.06$ | $\begin{bmatrix} 3.00 \pm 1.61 & 4.55 \pm 1.96 & 2.45 \pm 1.36 \\ 4.00 \pm 1.48 & 3.45 \pm 1.94 & 2.55 \pm 1.20 \\ 1.95 \pm 1.16 & 3.55 \pm 1.88 & 4.50 \pm 1.40 \end{bmatrix}$ |

Table 3.3: Test performance of proposed representation and a reference-based method colorectal cancer classification. Embedding length of the proposed representation is 1, and was trained using logistic regression with five-fold cross validation. The reported results are an average and standard deviation of performance on 100 randomly sampled test sets of size 18.

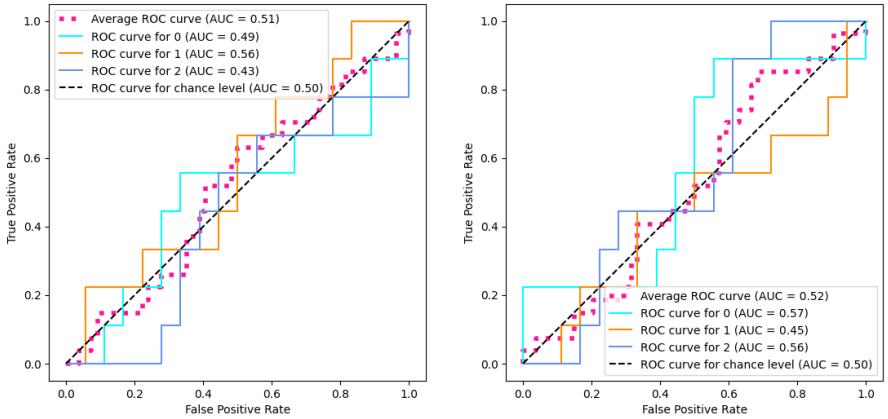
| Representation | Accuracy | Precision | Recall | F1 | Area under the curve | Confusion Matrix |
|---|---|---|---|---|---|---|
| Proposed representation with embedding length = 1 | $0.42 \pm 0.05$ | $0.29 \pm 0.04$ | $0.42 \pm 0.05$ | $0.33 \pm 0.04$ | $0.51 \pm 0.05$ | $\begin{bmatrix} 2.81 \pm 0.64 & 3.19 \pm 0.64 & 0.00 \pm 0.00 \\ 1.20 \pm 0.58 & 4.80 \pm 0.58 & 0.00 \pm 0.00 \\ 1.99 \pm 0.70 & 4.01 \pm 0.70 & 0.00 \pm 0.00 \end{bmatrix}$ |
| Kraken2 representation | $0.26 \pm 0.06$ | $0.26 \pm 0.06$ | $0.26 \pm 0.06$ | $0.25 \pm 0.06$ | $0.52 \pm 0.05$ | $\begin{bmatrix} 2.03 \pm 0.66 & 1.88 \pm 0.68 & 2.09 \pm 0.72 \\ 3.37 \pm 0.76 & 1.37 \pm 0.61 & 1.26 \pm 0.64 \\ 2.08 \pm 0.70 & 2.64 \pm 0.71 & 1.28 \pm 0.57 \end{bmatrix}$ |

Table 3.4: Class-wise test performance of proposed representation on colorectal cancer classification. The embedding length is one, and the representations were trained using logistic regression with five-fold cross validation. The reported results are an average and standard deviation of performance on 100 randomly sampled test sets of size 18.

| Representation | Precision | | | Recall | | | F1 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Class 0 | Class 1 | Class 2 | Class 0 | Class 1 | Class 2 | Class 0 | Class 1 | Class 2 |
| Proposed representation with embedding length = 1 | $0.44 \pm 0.11$ | $0.38 \pm 0.04$ | $0.00 \pm 0.00$ | $0.44 \pm 0.13$ | $0.76 \pm 0.09$ | $0.00 \pm 0.00$ | $0.44 \pm 0.11$ | $0.51 \pm 0.05$ | $0.00 \pm 0.00$ |
| Kraken2 representation | $0.29 \pm 0.08$ | $0.23 \pm 0.09$ | $0.29 \pm 0.14$ | $0.36 \pm 0.10$ | $0.23 \pm 0.09$ | $0.22 \pm 0.10$ | $0.32 \pm 0.08$ | $0.22 \pm 0.08$ | $0.25 \pm 0.12$ |

objects to label 2, which is the control class, as can be observed in Table 3.4 as well. Thus, all samples are classified as cancerous. While we want every data to be classified correctly, often some mis-classifications have a higher cost then others. For example, mis-classifying a healthy patient as cancerous, exposes them to cancer treatment, which can cause nausea and allergic reactions.

On the other hand, not providing treatment to a cancer patient mis-diagnosed as healthy can be potentially fatal. Accordingly, the cost of mis-classification is application dependent.




(a) Proposed representation

(b) Kraken2 representation

Figure 3.6: ROC curves for colorectal cancer classification using (a) proposed representation with embedding length = 1 and (b) Kraken2 relative abundance representation.

## 3.3. TYPE-2 DIABETES CLASSIFICATION

This section evaluates the performance of proposed representation for type-2 diabetes classification, and tests its predictive power compared to Kraken2 on unseen data.

### 3.3.1. HYPERPARAMETERS

1. **Performance oscillates as embedding length increases**
   Table 3.5 presents the performance of the proposed representation averaged over 100 randomly sampled validation sets. The embeddings were used to train a logistic classifier with five-fold cross validation. The pattern observed for type-2 diabetes is similar to the trend noticed for colorectal cancer dataset in Table 3.1. The performance fluctuates as the embedding length increases, which could be because of a difference in data separability for different embedding lengths. On the other hand, the training performance increases with embedding length, leading to overfitting, as evident from Figure 3.7a and Table A.5. This is likely because of an increase in the number of parameters in the model, leading to higher flexibility to overfit to the training data. Despite high overfitting for embedding length of 1000, it also has the highest area under the curve on validation set. This suggests that while the model is overfitting, it is also learning relevant patterns which can be generalized to new samples seen in the validation set.

2. **Increasing model depth can lead to overfitting**
   Table 3.6 presents the performance of the proposed representation over varying model depth. While no clear trend is observed for validation performance, analyzing Figure 3.7b and Table A.6 indicates overfitting as the model depth increases. A performance drop for training data is observed for three hidden layers, which could be due to decrease in data separability for that hyperparameter setting.

Table 3.5: Validation performance of proposed representation over different embedding lengths and one hidden layer for type-2 diabetes classification. The representations were trained using logistic regression with five-fold cross validation for one hidden layer. The reported results are an average and standard deviation of performance on 100 randomly sampled validation sets of size 20.

| Embedding length | Accuracy | Precision | Recall | F1 | Area under the curve | Confusion Matrix |
|---|---|---|---|---|---|---|
| 1 | $0.47 \pm 0.09$ | $0.43 \pm 0.18$ | $0.47 \pm 0.09$ | $0.40 \pm 0.09$ | $0.41 \pm 0.07$ | $\begin{bmatrix} 5.65 \pm 3.62 & 4.35 \pm 3.62 \\ 6.30 \pm 2.99 & 3.70 \pm 2.99 \end{bmatrix}$ |
| 5 | $0.44 \pm 0.10$ | $0.42 \pm 0.14$ | $0.44 \pm 0.10$ | $0.41 \pm 0.10$ | $0.42 \pm 0.12$ | $\begin{bmatrix} 4.79 \pm 2.42 & 5.21 \pm 2.42 \\ 6.07 \pm 2.31 & 3.93 \pm 2.31 \end{bmatrix}$ |
| 20 | $0.49 \pm 0.10$ | $0.49 \pm 0.12$ | $0.49 \pm 0.10$ | $0.47 \pm 0.10$ | $0.47 \pm 0.10$ | $\begin{bmatrix} 4.23 \pm 1.84 & 5.77 \pm 1.84 \\ 4.42 \pm 2.17 & 5.58 \pm 2.17 \end{bmatrix}$ |
| 100 | $0.57 \pm 0.08$ | $0.46 \pm 0.11$ | $0.47 \pm 0.08$ | $0.45 \pm 0.09$ | $0.47 \pm 0.09$ | $\begin{bmatrix} 5.46 \pm 1.92 & 4.54 \pm 1.91 \\ 6.00 \pm 2.13 & 4.00 \pm 2.13 \end{bmatrix}$ |
| 500 | $0.47 \pm 0.08$ | $0.46 \pm 0.09$ | $0.47 \pm 0.08$ | $0.45 \pm 0.08$ | $0.46 \pm 0.09$ | $\begin{bmatrix} 5.00 \pm 1.67 & 5.00 \pm 1.67 \\ 5.66 \pm 1.80 & 4.34 \pm 1.80 \end{bmatrix}$ |
| 1000 | $0.54 \pm 0.09$ | $0.55 \pm 0.10$ | $0.54 \pm 0.10$ | $0.54 \pm 0.10$ | $0.54 \pm 0.11$ | $\begin{bmatrix} 5.5 \pm 1.72 & 4.5 \pm 1.72 \\ 4.59 \pm 1.52 & 5.41 \pm 1.52 \end{bmatrix}$ |
| 1500 | $0.50 \pm 0.10$ | $0.49 \pm 0.12$ | $0.50 \pm 0.10$ | $0.47 \pm 0.10$ | $0.51 \pm 0.11$ | $\begin{bmatrix} 4.18 \pm 1.96 & 5.82 \pm 1.96 \\ 4.27 \pm 2.09 & 5.73 \pm 2.09 \end{bmatrix}$ |

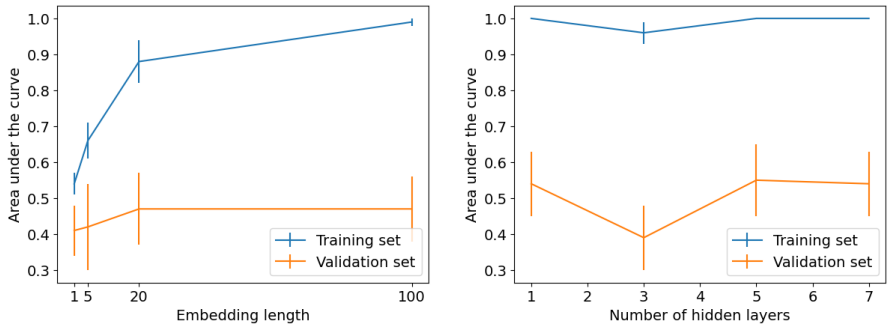
Table 3.6: Validation performance of the proposed representation over different number of hidden layers and an embedding length of 1000 for type-2 diabetes prediction. The representations were trained using ogistic regression with five-fold cross validation. The reported results are an average and standard deviation of performance on 100 randomly sampled validation sets of size 20.

| Number of hidden layers | Accuracy | Precision | Recall | F1 | Area under the curve | Confusion Matrix |
|---|---|---|---|---|---|---|
| 1 | $0.54 \pm 0.09$ | $0.55 \pm 0.10$ | $0.54 \pm 0.10$ | $0.54 \pm 0.10$ | $0.54 \pm 0.11$ | $\begin{bmatrix} 5.5 \pm 1.72 & 4.5 \pm 1.72 \\ 4.59 \pm 1.52 & 5.41 \pm 1.52 \end{bmatrix}$ |
| 3 | $0.39 \pm 0.09$ | $0.36 \pm 0.10$ | $0.40 \pm 0.09$ | $0.36 \pm 0.08$ | $0.33 \pm 0.08$ | $\begin{bmatrix} 5.11 \pm 2.26 & 4.89 \pm 2.26 \\ 7.21 \pm 1.96 & 2.79 \pm 1.96 \end{bmatrix}$ |
| 5 | $0.55 \pm 0.10$ | $0.55 \pm 0.11$ | $0.55 \pm 0.10$ | $0.54 \pm 0.10$ | $0.56 \pm 0.10$ | $\begin{bmatrix} 5.60 \pm 1.77 & 4.40 \pm 1.77 \\ 4.68 \pm 1.52 & 5.32 \pm 1.52 \end{bmatrix}$ |
| 7 | $0.54 \pm 0.09$ | $0.55 \pm 0.11$ | $0.54 \pm 0.09$ | $0.53 \pm 0.10$ | $0.58 \pm 0.10$ | $\begin{bmatrix} 5.98 \pm 1.93 & 4.02 \pm 1.93 \\ 5.08 \pm 2.06 & 4.92 \pm 2.06 \end{bmatrix}$ |

### 3.3.2. PERFORMANCE ON TEST DATA

Based on the hyperparameter evaluations, embedding length of 1000 along with one hidden layer were used to generate the representations. These were then trained on a logistic regression model with five-fold cross validation.

Table 3.7 presents the performance of the proposed representation on unseen data. It can be seen that although embedding length of 1000 gave the highest validation performance, its results are close to random for test dataset. On the other hand, embedding length of one, which has the lowest validation result, performs noticeably well on unseen data, as also shown in Figure 3.8. However, it classifies nearly all samples as control, and collapses for diabetes classification, as also seen in Table 3.8. This can be deduced from Figure 3.8a as well, where only one label in the entire test dataset is classified to diabetes class, while the rest are classified as control. Consequently, the proposed representation does not seem to be a good approach for type-2 diabetes classification.




(a) Performance of proposed representation for different embedding lengths

(b) Performance of proposed representation for different number of hidden layers

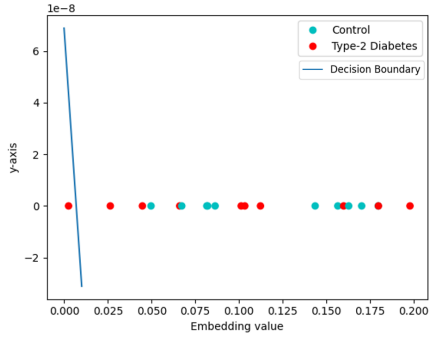
Figure 3.7: Performance of proposed representation on type-2 diabetes dataset [51] for different hyperparameter settings. The reported results are an average and standard deviation of performance on 100 randomly sampled validation sets of size 20.

Table 3.7: Test performance of proposed representation on type-2 diabetes classification. The representations were trained using logistic regression with five-fold cross validation. The reported results are an average and standard deviation of performance on 100 randomly sampled test sets of size 10.

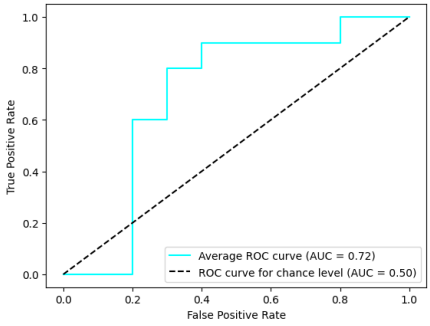
| Representation | Accuracy | Precision | Recall | F1 | Area under the curve | Confusion Matrix |
|---|---|---|---|---|---|---|
| Proposed representation with embedding length = 1 | $0.45 \pm 0.05$ | $0.24 \pm 0.01$ | $0.45 \pm 0.05$ | $0.31 \pm 0.02$ | $0.72 \pm 0.05$ | $\begin{bmatrix} 4.48 \pm 0.50 & 0.52 \pm 0.50 \\ 5.00 \pm 0.00 & 0.00 \pm 0.00 \end{bmatrix}$ |
| Proposed representation with embedding length = 1000 | $0.46 \pm 0.12$ | $0.45 \pm 0.13$ | $0.46 \pm 0.12$ | $0.45 \pm 0.13$ | $0.50 \pm 0.15$ | $\begin{bmatrix} 1.98 \pm 0.86 & 3.02 \pm 0.86 \\ 2.39 \pm 0.81 & 2.61 \pm 0.81 \end{bmatrix}$ |
| Kraken2 representation | $0.74 \pm 0.09$ | $0.76 \pm 0.10$ | $0.74 \pm 0.09$ | $0.74 \pm 0.10$ | $0.90 \pm 0.07$ | $\begin{bmatrix} 3.49 \pm 0.81 & 1.51 \pm 0.81 \\ 1.05 \pm 0.59 & 3.95 \pm 0.59 \end{bmatrix}$ |

Table 3.8: Class-wise test performance of proposed representation with embedding length of one for type-2 diabetes classification. The representations were trained using logistic regression with five-fold cross validation. The reported results are an average and standard deviation of performance on 100 randomly sampled test sets of size 10.

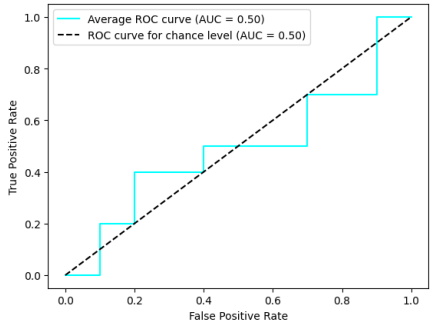
| Representation | Precision | | Recall | | F1 | |
|---|---|---|---|---|---|---|
| | Class 0 | Class 1 | Class 0 | Class 1 | Class 0 | Class 1 |
| Proposed representation with embedding length = 1 | $0.48 \pm 0.03$ | $0.00 \pm 0.00$ | $0.91 \pm 0.10$ | $0.00 \pm 0.00$ | $0.62 \pm 0.05$ | $0.00 \pm 0.00$ |
| Kraken2 representation | $0.78 \pm 0.13$ | $0.74 \pm 0.12$ | $0.70 \pm 0.17$ | $0.79 \pm 0.12$ | $0.73 \pm 0.12$ | $0.76 \pm 0.09$ |



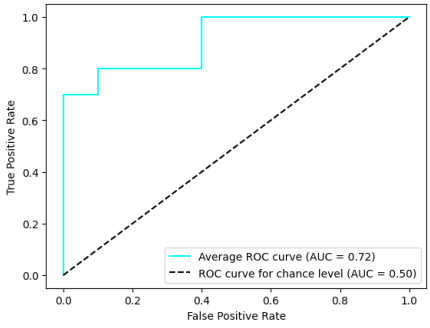

(a) Proposed representation with embedding length = 1




(b) ROC of type-2 diabetes classification using proposed representation of embedding length = 1




(c) ROC of type-2 diabetes classification using proposed representation of embedding length = 1000




(d) ROC of type-2 diabetes classification using Kraken2 representation

Figure 3.8: (a) shows the distribution of proposed embeddings of length one. The line represents the decision boundary for type-2 diabetes versus control classification. (b), (c), and (d) present the ROC curves for type-2 diabetes classification using proposed representation of embedding length = 1 proposed representation of embedding length = 1000, and Kraken2 relative abundance representation respectively

## 3.4. TYPE-2 DIABETES METRIC REGRESSION

This section evaluates the performance of the proposed representation in predicting several measures, namely body mass index, fasting blood glucose, fasting serum insulin, systolic and diastolic blood pressure, triglyceride level, low and high density lipoprotein levels, total cholesterol, and glycosylated hemoglobin level, which are known to be associated with type-2 diabetes. The dataset metadata included height, weight, age, and gender information as well. However, we did not explore them for this work, as they are not known to be closely associated with type-2 diabetes.



Based on the observations made in the hyperparameter evaluations thus far, we posited that increasing model depth often leads to overfitting, especially as the input data is sparse. Accordingly, all the results reported henceforth have been reported for a model architecture with one hidden layer.

### 3.4.1. BODY MASS INDEX

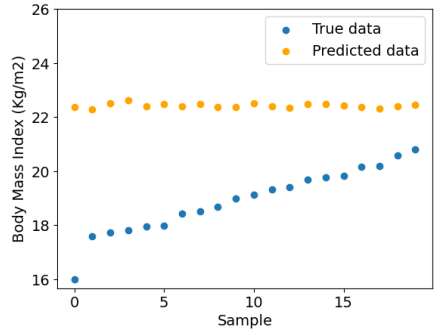
Body mass index (BMI) is a commonly used indicator of body fat based on a patient's weight and height. Elevated BMIs, especially BMI > 30 suggest a high risk of developing several health complications, including type-2 diabetes [65].

Table 3.9: Validation performance of proposed representation over different embedding lengths and one hidden layer for BMI prediction. The representations were trained using ridge regression with five-fold cross validation for 100 randomly sampled validation sets of size 10.

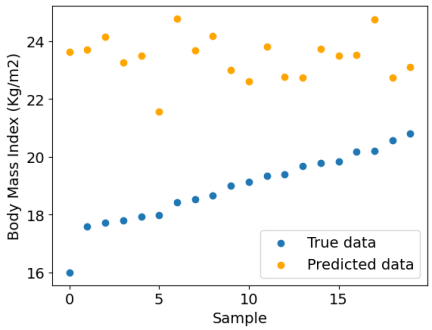
| Embedding length | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| 1 | $0.07 \pm 0.08$ | $14.76 \pm 2.42$ | $3.55 \pm 0.26$ |
| 5 | $0.04 \pm 0.05$ | $14.78 \pm 0.29$ | $3.55 \pm 0.03$ |
| 20 | $0.04 \pm 0.05$ | $14.83 \pm 3.04$ | $3.50 \pm 0.32$ |
| 100 | $0.04 \pm 0.04$ | $5.03 \pm 1.94$ | $1.77 \pm 0.28$ |
| 500 | $0.05 \pm 0.07$ | $4.80 \pm 1.52$ | $1.74 \pm 0.21$ |
| 1000 | $0.02 \pm 0.04$ | $14.27 \pm 2.54$ | $3.49 \pm 0.28$ |
| 1500 | $0.04 \pm 0.07$ | $14.61 \pm 2.80$ | $3.53 \pm 0.31$ |

Table 3.10: Test performance of proposed representation, reference-based method, and randomly sampled values for BMI prediction. The representations were trained using ridge regression with five-fold cross validation for 100 randomly sampled validation sets of size 10.

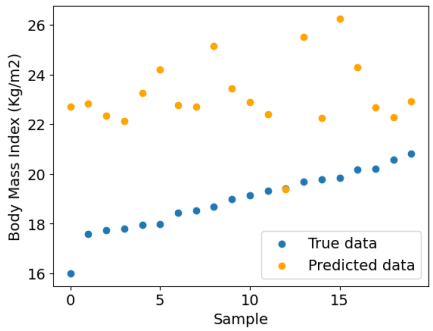
| Embedding length | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| Proposed representation with embedding length = 100 | $0.09 \pm 0.10$ | $13.60 \pm 2.15$ | $3.50 \pm 0.28$ |
| Proposed representation with embedding length = 500 | $0.06 \pm 0.08$ | $22.02 \pm 2.97$ | $4.48 \pm 0.31$ |
| Kraken2 representation | $0.04 \pm 0.05$ | $20.17 \pm 3.15$ | $4.14 \pm 0.40$ |
| Randomly sampled values | $0.12 \pm 0.15$ | $2.69 \pm 1.23$ | $1.31 \pm 0.32$ |



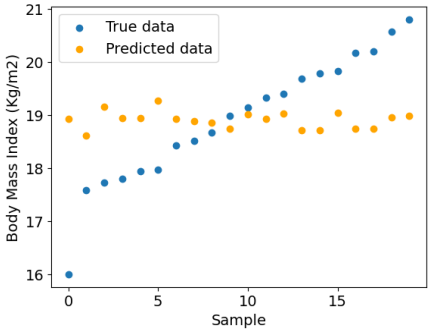

(a) True versus predicted values using proposed representation with embedding length = 100




(b) True versus predicted values using proposed representation with embedding length = 500




(c) True versus predicted values using Kraken2 representation

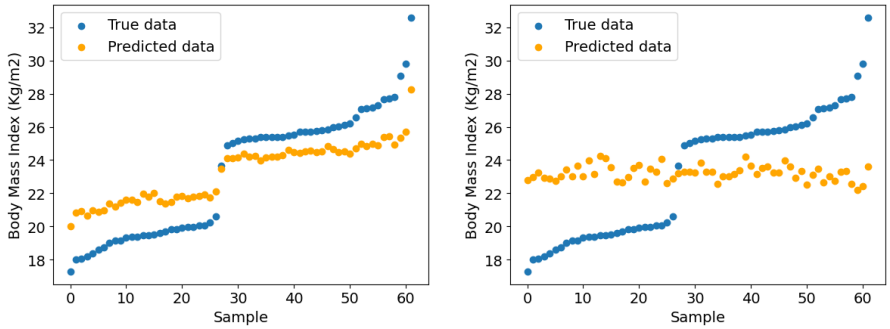



(d) True versus predicted values using randomly samples values

Figure 3.9: Scatter plot of true versus predicted data of (a) proposed representation with embedding length of one, (b) embedding length of 500, (c) Kraken2 representation, and (d) randomly samples values for Body Mass Index prediction.

Table 3.9 presents the performance of the proposed model in predicting BMI based on various embedding lengths. We see that the performance does not improve or decline with embedding length, but oscillates similar to colorectal cancer and type-2 diabetes classification. While embedding length of one has slightly higher r2 score, embedding length of 500 has comparable r2 with lower average mean squared and mean absolute errors, along with lower standard deviations. Accordingly, we evaluated the performance of the proposed representation with embedding length of 500 for BMI prediction. This performance was compared with Kraken2's relative abundance representation and predictions sampled from normal distribution of BMI test data.

Table 3.10 presents the observed predictions on unseen test data. Randomly sampled predictions have the best prediction power, with the closest fit based on r2 score, and lower prediction errors. This can be further verified from Figure 3.9 which shows the values predicted by each representation for various test values. It can be seen that the dynamic range of the true data is around two units. Consequently, values randomly sampled from a distribution centred around the mean of the data has a low error. For a higher dynamic range, random predictions are unlikely to perform well. Predictions of




(a) True and predicted values using proposed representation for training set

(b) True and predicted values using proposed representation for test set

Figure 3.10: Scatter plot of true and predicted values of proposed representation on training and validation data for BMI prediction. The embedding length was set to 500, and the representations were trained using a ridge regression model with five-fold cross validation. The reported results were averaged over 100 randomly sampled test sets of size 10.

Table 3.11: Training and validation performance of proposed representation for BMI prediction with embedding length of 500 and one hidden layer. The representations were trained using ridge regression with five-fold cross validation on 100 randomly sampled test sets of size 10.

| Embedding length | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| Training set | $0.97 \pm 0.03$ | $5.70 \pm 4.55$ | $1.87 \pm 1.17$ |
| Validation set | $0.05 \pm 0.06$ | $14.52 \pm 2.64$ | $3.46 \pm 0.29$ |

the proposed representation and Kraken2's relative abundance representation are far off from the original values, and Kraken2 has only one value that fits closely in the range. This suggests that these representations might be unable to identify the pattern in the data, and might be unsuitable for BMI prediction.

Analysing the training and validation performance of the proposed method in Table 3.11 and Figure 3.10 reveals that the model is overfitting on training data. Similar pattern was observed for colorectal cancer and type-2 diabetes classification, where increasing embedding length led to overfitting. Accordingly, we analyzed the performance of the proposed representation on embedding length 100, as it presented comparable performance to embedding length 500. We found that embedding length of 100 is slightly closer to the true data, but still has a high error, as can be observed in Figure 3.9a.

### 3.4.2. LOW-DENSITY LIPOPROTEIN LEVEL

Low-density lipoprotein levels (LDL) represent the level of unhealthy or bad cholesterol in the body. High density of LDL, approximately > 5.6 mmol/L, has been frequently associated with health problems such as heart diseases and diabetes [66]. Similar to BMI, we evaluated the performance of the proposed representation for different embedding lengths, as shown in Table 3.12.

We found embedding length of 20 to give a slightly higher r2 score, with comparable mean squared and mean absolute errors. However, the standard deviation of r2 score is also higher than other embedding lengths. Nevertheless, we evaluated the performance of the proposed method with embedding length of 20 on unseen test data, and compared the results with Kraken2 representation and random samples.

As shown in Table 3.13, Kraken2 representation has a slightly higher r2 score than the proposed representation. However, it also has a higher mean squared and mean absolute error with comparable standard deviations. Further, Figure 3.11 shows that the performance of Kraken2 representation and randomly sampled values seem similar, whereas the proposed representation does not show much variation, and has a very low dynamic range. Consequently, the proposed representation does not seem to be a good approach for low density lipoprotein level prediction.



Table 3.12: Validation performance of proposed representation over different embedding lengths and one hidden layer for low density lipo-protein level prediction. The representations were trained using ridge regression with five-fold cross validation. The results were averaged over 100 randomly sampled test sets of size 10.

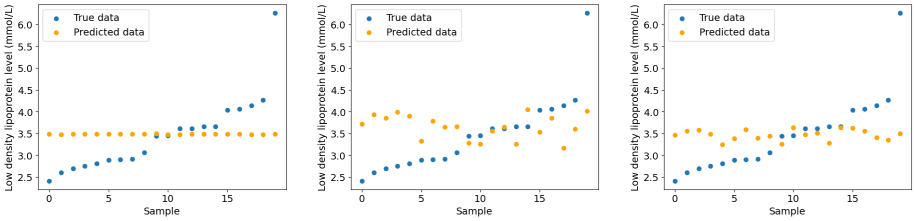
| Embedding length | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| 1 | $0.06 \pm 0.08$ | $0.47 \pm 0.11$ | $0.56 \pm 0.08$ |
| 5 | $0.06 \pm 0.07$ | $0.50 \pm 0.12$ | $0.58 \pm 0.08$ |
| 20 | $0.11 \pm 0.09$ | $0.47 \pm 0.10$ | $0.56 \pm 0.07$ |
| 100 | $0.04 \pm 0.05$ | $0.51 \pm 0.12$ | $0.59 \pm 0.08$ |
| 500 | $0.09 \pm 0.08$ | $0.49 \pm 0.10$ | $0.58 \pm 0.07$ |
| 1000 | $0.04 \pm 0.05$ | $0.48 \pm 0.10$ | $0.57 \pm 0.07$ |
| 1500 | $0.04 \pm 0.05$ | $0.48 \pm 0.10$ | $0.57 \pm 0.07$ |

Table 3.13: Validation performance of proposed representation over different number of hidden layers and an embedding length of one for low density lipo-protein level prediction. The representations were trained using ridge regression with five-fold cross validation. The results were averaged over 100 randomly sampled test sets of size 10.

| Approach | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| Proposed representation | $0.08 \pm 0.10$ | $0.74 \pm 0.35$ | $0.63 \pm 0.12$ |
| Kraken2 representation | $0.11 \pm 0.11$ | $0.84 \pm 0.26$ | $0.74 \pm 0.12$ |
| Randomly sampled values | $0.10 \pm 0.13$ | $1.47 \pm 0.61$ | $0.94 \pm 0.20$ |

### 3.4.3. OTHER DIABETES-RELATED METRICS

Several other metrics, namely fasting blood glucose, fasting serum insulin, systolic and diastolic blood pressure, triglyceride level, high density lipoprotein levels, total cholesterol, and glycosylated hemoglobin level, were also analyzed to evaluate the performance of the proposed representation. We performed hyperparameter evaluations for various embedding lengths on each of these measures, as shown in Tables A.7-A.14. Based on the results obtained, appropriate embeddings lengths were selected for each measure for testing.




(a) Proposed representation with embedding length = 20

(b) Kraken2 representation

(c) Randomly samples values

Figure 3.11: Scatter plot of true and predicted data of proposed representation with embedding length = 20, Kraken2 representation, and randomly samples values for low density lipoprotein level prediction in mmol/L.

For example, for high density lipoprotein prediction, embedding length of 1000 has the highest r2 score and slightly lower standard deviation. Hence, we selected this hyperparameter setting for testing. On the other hand, for triglyceride levels, the embeddings were found to have comparable performances. In such cases, we selected the smaller embedding length. Hence, embedding length of 1 was selected for triglyceride level prediction. Similarly, prediction performance for blood glucose level and glycosylated hemoglobin level was also evaluated with an embedding length of 1. Systolic and diastolic blood pressures and fasting serum insulin levels were found to perform best with embedding lengths of 5. On the other hand, total cholesterol was found to be best represented by embedding length of 500.

As can be seen from Tables A.15-A.22 and Figures A.2-A.9, none of the representations, including the proposed approach, Kraken2 relative abundance, and random sampling, were found to be adequate for this analysis. It can be seen from the Figures that the proposed representation has nearly zero dynamic range, indicating that the model is predicting a constant value for all unseen samples. This suggests the inability of the proposed representation to generalize to unseen data.

## 3.5. EMBEDDING WEIGHTS DO NOT GENERALIZE TO NEW SAMPLES

To obtain embeddings of new samples, the original Doc2vec architecture freezes word transformation matrix $W_2$ and decoder matrix $W_3$, and updates only the sample embedding matrix $W_1$. This is achieved by assigning the new sample an index $S+1$, and adding a row to the transformation matrix $W_1$ pre-trained on $S$ samples. Accordingly, gradient descent is performed only on the sample transformation matrix to generate sample embeddings.

In our analysis of this approach from Doc2vec's original paper, we found the model loss to not decline or converge. Hence, we concluded this approach to be unfit for our application. In this work, every new sample was first added to the existing data, followed by re-training the model on the updated dataset. This approach has been used for all other experiments reported in this document.

## 3.6. COSINE SIMILARITY CANNOT BE USED TO FIND BIOMARKERS

Doc2vec suggests that information with similar context should have high cosine similarities [45]. Hence, we computed the cosine distance of k-mer embeddings and sample embeddings to find k-mers with the highest contribution towards sample representation. Such k-mers would then be posited as potential biomarkers for further analysis.

However, in the experiments, we found that nearly all k-mers had a cosine similarity of 1.0 with the corresponding sample embeddings. This could mean that every k-mer used for training is an important biomarker. However, a more plausible reasoning could be that finding biomarkers based on cosine similarity is not a suitable approach for this application.



## 3.7. REVISITING RESEARCH QUESTIONS

In this section, we answer the research questions put forth in Chapter 1 based on the discussions and observations made in this work.

1. Is the predictive power of proposed representation comparable to reference-based methods?
   For colorectal cancer dataset, we found that neither of the representations performed well in the classification task. On the other hand, while the proposed representation had a high area under the curve performance for type-2 diabetes, performance of reference-based method was better. Further, the proposed representation classified all the samples into one class, rendering it less useful in practical applications.

   For regression tasks, we observed that while the proposed representation showed a quantitatively comparable performance with reference-based method on validation set, its test set predictions were the same for all samples and had nearly zero standard deviation. This suggests that the model is strongly overfitting, as was observed for hyperparameter settings with higher embedding lengths and number of hidden layers for colorectal cancer dataset as well. Accordingly, modifications to address overfitting can be explored.

   One approach is to introduce regularization in the classification model. Alternatively, one of the reasons for overfitting could be the high sparsity in the input data. This can be address by introducing some noise in the input, such as by accounting for the Illumina error rate at nucleotide level [67]. Further, not all possible hyperparameter combinations were tested in this work. We introduced randomness in one aspect of the model at a time, specifically embedding length and model depth. As increasing these factors leads to overfitting, it is expected that increasing both factors is likely to cause overfitting as well. However, in the interest of time, these experiment was not performed, and can be explored in the future.

2. Can the sequences obtained by feature importance be interpreted as biomarkers? For type-2 diabetes classification task, we used cosine similarity between k-mer and sample embeddings to find potential biomarkers. However, we found all k-mers to present a similarity of 1.0. Accordingly, we could not conclude if these k-mers can be interpreted as biomarkers, as it would be impractical to individually analyse all of them. Hence, it is unclear if the seqeunces can be interpreted as biomarkers. However, it can be concluded that cosine similarity is not a suitable approach to find biomarkers in the current setting.

# 4

# Conclusions

The past two decades have witnessed an increased access to millions of reads generated by metagenomic next generation sequencing techniques. This has opened horizons for precision medicine and disease diagnosis. With an aim to efficiently analyze vast volume of data, we proposed a Doc2vec-based method as a reference-free approach to represent these metagenome samples.

We addressed the bottleneck of Doc2vec which limits existing Word2vec-based omic embedding techniques from analyzing long k-mers. This was done by performing nucleotide-level encoding instead of k-mer level. We tested the proposed representation for classification of colorectal cancer and type-2 diabetes, and regression on several type-2 diabetes related measures. For colorectal cancer, neither the proposed representation nor the reference-based approach performed well, suggesting that the dataset might be unsuitable for metagenome study. On the other hand, for type-2 diabetes, reference-based method outperformed the proposed representation. Further, we observed that the proposed approach collapsed for some classes in each dataset, indicating the need for further modifications and improvements. Similar conclusions were drawn for regression tasks, where the proposed representation failed to generalize for unseen samples.

Evaluating cosine similarity between k-mer and sample embeddings revealed a high cosine similarity of 1.0 in all comparisons. This indicates that cosine similarity is not a fit approach to find biomarkers in the current settings.

One of the limitations of the proposed representation is that the trained weights cannot be directly used to generate new embeddings. Instead, the model needs to be retrained on the updated dataset containing the new sample.

Accordingly, several modifications can be made in the proposed representation. These include addressing overfitting with regularization and noise, exploring alternate approaches for biomarker discovery, and modifying the architecture for proposed representation such that the trained weights can be directly used to compute new embeddings. These recommendations can be addressed in future experiments.

# BIBLIOGRAPHY


[1] L. Bragg and G. W. Tyson, "Metagenomics using next-generation sequencing," *Methods in Molecular Biology*, pp. 183–201, 2014. DOI: 10.1007/978-1-62703-712-9_15. [Online]. Available: https://doi.org/10.1007/978-1-62703-712-9_15.

[2] N. D. Olson, T. J. Treangen, C. M. Hill, *et al.*, "Metagenomic assembly through the lens of validation: Recent advances in assessing and improving the quality of genomes assembled from metagenomes," *Briefings in Bioinformatics*, vol. 20, no. 4, pp. 1140–1150, 2019. DOI: 10.1093/bib/bbx098. [Online]. Available: https://doi.org/10.1093/bib/bbx098.

[3] F. D. Vecchio, V. Mastroiaco, A. D. Marco, *et al.*, "Next-generation sequencing: Recent applications to the analysis of colorectal cancer," *Journal of Translational Medicine*, vol. 15, no. 1, 2017. DOI: 10.1186/s12967-017-1353-y. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/29221448/.

[4] D. Qin, "Next-generation sequencing and its clinical application," *Cancer Biology and Medicine*, vol. 16, no. 1, pp. 4–10, 2019. DOI: 10.20892/j.issn.2095-3941.2018.0055. [Online]. Available: https://doi.org/10.20892/j.issn.2095-3941.2018.0055.

[5] H. Duan, X. Li, A. Mei, *et al.*, "The diagnostic value of metagenomic next-generation sequencing in infectious diseases," *BMC Infectious Diseases*, vol. 21, no. 62, 2021. DOI: 10.1186/s12879-020-05746-5. [Online]. Available: https://doi.org/10.1186/s12879-020-05746-5.

[6] B. Schmidt and A. Hildebrandt, "Next-generation sequencing: Big data meets high performance computing," *Drug Discovery Today*, vol. 22, no. 4, pp. 712–717, 2017. DOI: 10.1016/j.drudis.2017.01.014. [Online]. Available: https://doi.org/10.1016/j.drudis.2017.01.014.

[7] R. Daneshjou, Y. Wang, Y. Bromberg, *et al.*, "Working towards precision medicine: Predicting phenotypes from exomes in the critical assessment of genome interpretation (cagi) challenges," *Human Mutation*, vol. 38, no. 9, pp. 1182–1192, 2017. DOI: 10.1002/humu.23280. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5600620/.

[8] M. Morash, H. Mitchell, H. Beltran, *et al.*, "The role of next-generation sequencing in precision medicine: A review of outcomes in oncology," *Journal of Personalized Medicine*, vol. 8, no. 3, 2018. DOI: 10.3390/jpm8030030. [Online]. Available: https://doi.org/10.3390%5C%2Fjpm8030030.

[9] P. Suwinski, C. Ong, M. Ling, *et al.*, "Advancing personalized medicine through the application of whole exome sequencing and big data analytics," *Frontiers in Genetics*, vol. 10, 2019. DOI: 10.3389/fgene.2019.00049. [Online]. Available: https://doi.org/10.3389/fgene.2019.00049.

[10] W. Yang, Y.-C. Lin, W. Johnson, *et al.*, "A genome-phenome association study in native microbiomes identifies a mechanism for cytosine modification in dna and rna," *eLife*, 2021. DOI: 10.7554/eLife.70021. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/34747693/.

[11] E. M. Ross and B. J. Hayes, "Metagenomic predictions: A review 10 years on," *Frontiers in Genetics*, vol. 13, 2022. DOI: 10.3389/fgene.2022.865765. [Online]. Available: https://doi.org/10.3389/fgene.2022.865765.

[12] O. Aasmets, K. L. Krigul, K. Kull, *et al.*, "Gut metagenome associations with extensive digital health data in a volunteer-based estonian microbiome cohort," *Nature Communications*, vol. 13, no. 869, 2022. DOI: 10.1038/s41467-022-28464-9. [Online]. Available: https://doi.org/10.1038/s41467-022-28464-9.

[13] J. Qin, Y. Li, Z. Cai, *et al.*, *Nature*, vol. 490, pp. 55–60, 2012. DOI: 10.1038/nature11450. [Online]. Available: https://doi.org/10.1038/nature11450.

[14] B. Bakir-Gungor, O. Bulut, A. Jabeer, *et al.*, "Discovering potential taxonomic biomarkers of type 2 diabetes from human gut microbiota via different feature selection methods," *Frontiers in Microbiology*, vol. 12, 2021. DOI: 10.3389/fmicb.2021.628426. [Online]. Available: https://doi.org/10.3389/fmicb.2021.628426.

[15] S. Yachida, S. Mizutani, H. Shiroma, *et al.*, "Metagenomic and metabolomic analyses reveal distinct stage-specific phenotypes of the gut microbiota in colorectal cancer," *Nature Medicine*, vol. 25, pp. 968–976, 2019. DOI: 10.1038/s41591-019-0458-7. [Online]. Available: https://doi.org/10.1038/s41591-019-0458-7.

[16] Y. Ma, Y. Zhang, H. Jiang, *et al.*, "Metagenome analysis of intestinal bacteria in healthy people, patients with inflammatory bowel disease and colorectal cancer," *Frontiers in Cellular and Infection Microbiology*, vol. 11, 2021. DOI: 10.3389/fcimb.2021.599734. [Online]. Available: https://doi.org/10.3389/fcimb.2021.599734.

[17] N. LaPierre, M. Alser, E. Eskin, *et al.*, "Metalign: Efficient alignment-based metagenomic profiling via containment min hash," *Genome Biology*, vol. 21, 2020. DOI: 10.1186/s13059-020-02159-0. [Online]. Available: https://doi.org/10.1186/s13059-020-02159-0.

[18] M. Santamaria, B. Fosso, A. Consiglio, *et al.*, "Reference databases for taxonomic assignment in metagenomics," *Briefings in Bioinformatics*, vol. 13, no. 6, pp. 682–695, 2012. DOI: 10.1093/bib/bbs036. [Online]. Available: https://doi.org/10.1093/bib/bbs036.

[19] D. E. Wood, J. Lu, and B. Landmean, "Improved metagenomic analysis with kraken 2," *Genome Biology*, vol. 20, no. 257, 2019. DOI: 10.1186/s13059-019-1891-0. [Online]. Available: https://doi.org/10.1186/s13059-019-1891-0.

[20] D. T. Truong, E. A. Franzosa, T. L. Tickle, *et al.*, "Metaphlan2 for enhanced metagenomic taxonomic profiling," *Nature Methods*, vol. 12, pp. 902–903, 2015. DOI: 10.1038/nmeth.3589. [Online]. Available: https://doi.org/10.1038/nmeth.3589.

[21] D. Kim, L. Song, F. P. Breitwieser, and S. L. Salzberg, "Centrifuge: Rapid and sensitive classification of metagenomic sequences," *Genome research*, vol. 26, no. 12, pp. 1721–1729, 2016. DOI: 10.1101/gr.210641.116. [Online]. Available: https://doi.org/10.1101%5C%2Fgr.210641.116.

[22] A. Drouin, S. Giguere, M. Deraspe, *et al.*, "Predictive computational phenotyping and biomarker discovery using reference-free genome comparisons," *BMC Genomics*, vol. 17, no. 754, 2016. DOI: 10.1186/s12864-016-2889-6. [Online]. Available: https://doi.org/10.1186/s12864-016-2889-6.

[23] S. Albright and S. Louca, "Trait biases in microbial reference genomes," *Scientific Data*, vol. 10, no. 84, 2023. DOI: 10.1038/s41597-023-01994-7. [Online]. Available: https://doi.org/10.1038/s41597-023-01994-7.

[24] C. Yang, D. Chowdhury, Z. Zhang, *et al.*, "A review of computational tools for generating metagenome-assembled genomes from metagenomic sequencing data," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 6301–6314, 2021. DOI: 10.1016/j.csbj.2021.11.028. [Online]. Available: https://doi.org/10.1016/j.csbj.2021.11.028.

[25] A. Sczyrba, P. Hofmann, P. Belmann, *et al.*, "Critical assessment of metagenome interpretation—a benchmark of metagenomics software," *Nature Methods*, vol. 14, pp. 1063–1071, 2017. DOI: 10.1038/nmeth.4458. [Online]. Available: https://doi.org/10.1038/nmeth.4458.

[26] S. Lai, S. Pan, C. Sun, *et al.*, "Metamic: Reference-free misassembly identification and correction of de novo metagenomic assemblies," *Genome Biology*, vol. 23, no. 242, 2022. DOI: 10.1186/s13059-022-02810-y. [Online]. Available: https://doi.org/10.1186/s13059-022-02810-y.

[27] H. W. L. Lischer and K. K. Shimizu, "Reference-guided de novo assembly approach improves genome reconstruction for related species," *BMC Bioinformatics*, vol. 18, no. 474, 2017. DOI: 10.1186/s12859-017-1911-6. [Online]. Available: https://doi.org/10.1186/s12859-017-1911-6.

[28] S. L. Salzberg, A. M. Philippy, A. Zimin, *et al.*, "Gage: A critical evaluation of genome assemblies and assembly algorithms," *Genome Research*, vol. 22, pp. 557–567, 2012. DOI: 10.1101/gr.131383.111. [Online]. Available: https://genome.cshlp.org/content/22/3/557.

[29] K. R. Bradnam, J. N. Fass, A. Alexandrov, *et al.*, "Assemblathon 2: Evaluating de novo methods of genome assembly in three vertebrate species," *GigaScience*, vol. 2, no. 1, 2013. DOI: 10.1186/2047-217X-2-10. [Online]. Available: https://doi.org/10.1186/2047-217X-2-10.

[30] X. Xing, J. S. Liu, and W. Zhong, "Metagen: Reference-free learning with multiple metagenomic samples," *Genome Biology*, vol. 18, no. 187, 2017. DOI: 10.1186/s13059-017-1323-y. [Online]. Available: https://doi.org/10.1186/s13059-017-1323-y.

[31] L. Khachatryan, S. Y. Anvar, R. Vossen, and J. Laros, "Reference-free resolution of long-read metagenomic data," *bioRxiv*, 2019. DOI: 10.1101/811760. [Online]. Available: https://doi.org/10.1101/811760.

[32] H. Dai and Y. Guan, "The nubeam reference-free approach to analyze metagenomic sequencing reads," *Genome Research*, vol. 30, pp. 1364–1375, 2020. DOI: 10.1101/gr.261750.120. [Online]. Available: https://doi.org/10.1101/gr.261750.120.

[33] P. D. Browne, T. K. Nielsen, W. Kot, *et al.*, "Gc bias affects genomic and metagenomic reconstructions, underrepresenting gc-poor organisms," *GigaScience*, vol. 9, no. 2, 2020. DOI: 10.1093/gigascience/giaa008. [Online]. Available: https://doi.org/10.1093/gigascience/giaa008.

[34] Y. Wang, L. Fu, and J. Ren, "Identifying group-specific sequences for microbial communities using long k-mer sequence signatures," *Frontiers in microbiology*, vol. 9, 2018. DOI: 10.3389/fmicb.2018.00872. [Online]. Available: https://doi.org/10.3389/fmicb.2018.00872.

[35] X. Xing, J. S. Liu, and W. Zhong, "Metagen: Reference-free learning with multiple metagenomic samples," *Genome Biology*, vol. 18, no. 187, 2017. DOI: 10.1186/s13059-017-1323-y. [Online]. Available: https://doi.org/10.1186/s13059-017-1323-y.

[36] A. Gkanogiannis, S. Gazut, and M. Salanoubat, "A scalable assembly-free variable selection algorithm for biomarker discovery from metagenomes," *BMC Bioinformatics*, vol. 17, no. 311, 2016. DOI: 10.1186/s12859-016-1186-3. [Online]. Available: https://doi.org/10.1186/s12859-016-1186-3.

[37] S. N. Aakur, V. Indla, S. Narayanan, *et al.*, "Metagenome2vec: Building contextualized representations for scalable metagenome analysis," *arXiv*, 2021. DOI: arXiv:2111.08001v1. [Online]. Available: https://arxiv.org/pdf/2111.08001.pdf.

[38] D. Miller, A. Stern, and D. Burstein, "Deciphering microbial gene function using natural language processing," *Nature Communications*, vol. 13, no. 5731, 2022. DOI: 10.1038/s41467-022-33397-4. [Online]. Available: https://doi.org/10.1038/s41467-022-33397-4.

[39] K. Odrzywolek, Z. Karwowska, J. Majta, *et al.*, "Deep embeddings to comprehend and visualize microbiome protein space," *Scientific Reports*, vol. 12, no. 10332, 2022. DOI: 10.1038/s41598-022-14055-7. [Online]. Available: https://doi.org/10.1038/s41598-022-14055-7.

[40] K. Odrzywolek, Z. Karwowska, J. Majta, *et al.*, "Deep embeddings to comprehend and visualize microbiome protein space," *Scientific Reports*, vol. 12, no. 10332, 2022. DOI: 10.1038/s41598-022-14055-7. [Online]. Available: https://doi.org/10.1038/s41598-022-14055-7.

[41] E. Asgari and M. R. K. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PLoS ONE*, vol. 10, no. 11, 2015. DOI: 10.1371/journal.pone.0141287. [Online]. Available: https://doi.org/10.1371/journal.pone.0141287.

[42] Z. Zhao, S. Woloszynek, F. Agbavor, *et al.*, "Learning, visualizing and exploring 16s rrna structure using an attention-based deep neural network," *Plos computational biology*, 2021. DOI: 10.1371/journal.pcbi.1009345. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1009345.

[43] S. Woloszynek, Z. Zhao, J. Chen, and G. L. Rosen, "16s rrna sequence embeddings: Meaningful numeric feature representations of nucleotide sequences that are convenient for downstream analyses," *PLOS Computational Biology*, vol. 15, no. 2, 2019. DOI: 10.1093/gigascience/giaa008. [Online]. Available: https://doi.org/10.1093/gigascience/giaa008.

[44] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv*, 2013. DOI: arXiv:1301.3781. [Online]. Available: https://doi.org/10.48550/arXiv.1301.3781.

[45] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *arXiv*, 2014. DOI: arXiv.1405.4053. [Online]. Available: https://doi.org/10.48550/arXiv.1405.4053.

[46] Y. Wang, Z.-H. You, S. Yang, *et al.*, "A high efficient biological language model for predicting protein–protein interactions," *Cells*, vol. 8, no. 2, 2019. DOI: 10.3390/cells8020122. [Online]. Available: https://doi.org/10.3390%5C%2Fcells8020122.

[47] P. Ng, "Dna2vec: Consistent vector representations of variable-length k-mers," *arXiv*, 2017. DOI: arXiv.1701.06279. [Online]. Available: https://doi.org/10.48550/arXiv.1701.06279.

[48] R. Laso-Jadart, C. Ambroise, P. Peterlongo, and M.-A. Madoui, "Metavar: Introducing metavariant species models for reference-free metagenomic-based population genomics," *Plos One*, 2020. DOI: 10.1371/journal.pone.0244637. [Online]. Available: https://doi.org/10.1371/journal.pone.0244637.

[49] M. A. Rahman and H. Rangwala, "Idmil: An alignment-free interpretable deep multiple instance learning (mil) for predicting disease from whole-metagenomic data," *Bioinformatics*, vol. 36, 2020. DOI: 10.1093/bioinformatics/btaa477. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32657370/.

[50] Q. Feng, S. Liang, H. Jia, *et al.*, "Gut microbiome development along the colorectal adenoma–carcinoma sequence," *Nature communications*, vol. 6, no. 6528, 2015. DOI: 10.1038/ncomms7528. [Online]. Available: https://doi.org/10.1038/ncomms7528.

[51] J. Qin, Y. Li, Z. Cai, *et al.*, "A metagenome-wide association study of gut microbiota in type 2 diabetes," *Nature*, vol. 490, pp. 55–60, 2012. DOI: 10.1038/nature11450. [Online]. Available: https://doi.org/10.1038/nature11450.

[52] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009. DOI: 10.1016/j.ipm.2009.03.002. [Online]. Available: https://doi.org/10.1016/j.ipm.2009.03.002.

[53] D. E. Wood and S. L. Salzberg, "Kraken: Ultrafast metagenomic sequence classification using exact alignments," *Genome Biology*, vol. 15, no. R46, 2014. DOI: 10.1186/gb-2014-15-3-r46. [Online]. Available: https://doi.org/10.1186/gb-2014-15-3-r46.

[54] G. Manzini, "The burrows-wheeler transform: Theory and practice," *International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science*, vol. 1672, pp. 34–47, 1999. DOI: 10.1007/3-540-48340-3_4. [Online]. Available: https://doi.org/10.1007/3-540-48340-3_4.

[55] P. Ferragina, G. Manzini, V. Makinen, and G. Navarro, "An alphabet-friendly fm-index," *International Symposium on String Processing and Information Retrieval, Lecture Notes in Computer Science*, vol. 3246, pp. 150–160, 2004. DOI: 10.1007/978-3-540-30213-1_23. [Online]. Available: https://doi.org/10.1007/978-3-540-30213-1_23.

[56] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg, "Ultrafast and memory-efficient alignment of short dna sequences to the human genome," *Genome Biology*, vol. 10, 2009. DOI: 10.1186/gb-2009-10-3-r25. [Online]. Available: https://doi.org/10.1186/gb-2009-10-3-r25.

[57] R. Ounit, S. Wanamaker, T. J. Close, and S. Lonardi, "Clark: Fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers," *BMC Genomics*, vol. 16, no. 236, 2015. DOI: 10.1186/s12864-015-1419-2. [Online]. Available: https://doi.org/10.1186/s12864-015-1419-2.

[58] E. Pasolli, D. T. Truong, F. Malik, *et al.*, "Machine learning meta-analysis of large metagenomic datasets: Tools and biological insights," *Plos Computational Biology*, 2016. DOI: 10.1371/journal.pcbi.1004977. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1004977.

[59] Q. Liang, P. W. Bible, Y. Liu, *et al.*, "Deepmicrobes: Taxonomic classification for metagenomics with deep learning," *NAR Genomics and Bioinformatics*, vol. 2, no. 1, 2020. DOI: 10.1093/nargab/lqaa009. [Online]. Available: https://doi.org/10.1093/nargab/lqaa009.

[60] G. Marcais and C. Kingsford, "A fast, lock-free approach for efficient parallel counting of occurrences of k-mers," *Bioinformatics*, vol. 27, no. 6, pp. 747–770, 2011. DOI: 10.1093/bioinformatics/btr011. [Online]. Available: https://doi.org/10.1093/bioinformatics/btr011.

[61] W. Chen, A. McKenna, J. Schreiber, *et al.*, "Massively parallel profiling and predictive modeling of the outcomes of crispr/cas9-mediated double-strand break repair," *Nucleic Acids Research*, vol. 47, no. 15, pp. 7989–8003, 2019. DOI: 10.1093/nar/gkz487. [Online]. Available: https://doi.org/10.1093/nar/gkz487.

[62] A. Schilling, A. Maier, R. Gerum, *et al.*, "Quantifying the separability of data classes in neural networks," *Neural Networks*, vol. 139, pp. 278–293, 2021. DOI: 10.1016/j.neunet.2021.03.035. [Online]. Available: https://doi.org/10.1016/j.neunet.2021.03.035.

[63] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *arXiv*, 2015. DOI: arXiv:1505.04597. [Online]. Available: https://arxiv.org/abs/1505.04597.

[64] Q. Xu, C. Zhang, L. Zhang, and Y. Song, "The learning effect of different hidden layers stacked autoencoder," *International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2016. DOI: 10.1109/IHMSC.2016.280. [Online]. Available: https://doi.org/10.1109/IHMSC.2016.280.

[65] A. Karin, E. Jon, A. Martin, *et al.*, "Body mass index in adolescence, risk of type 2 diabetes and associated complications: A nationwide cohort study of men," *The Lancet Discovery Science*, 2022. DOI: 10.1016/j.eclinm.2022.101356. [Online]. Available: https://doi.org/10.1016/j.eclinm.2022.101356.

[66] R. W. Nesto, "Ldl cholesterol lowering in type 2 diabetes: What is the optimum approach?" *Clinical Diabetes*, vol. 26, no. 1, pp. 8–13, 2008. DOI: 10.2337/diaclin.26.1.8. [Online]. Available: https://doi.org/10.2337/diaclin.26.1.8.

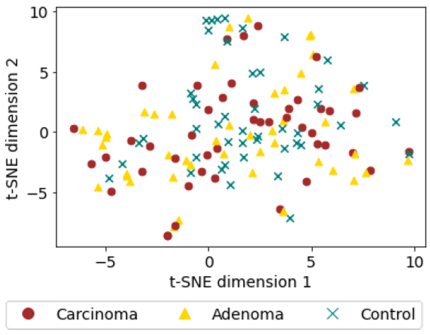
[67] D. I. Lou, J. A. Hussmann, R. M. McBee, *et al.*, "High-throughput dna sequencing errors are reduced by orders of magnitude using circle sequencing," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110, no. 49, pp. 19 872–19 877, 2013. DOI: 10.1073/pnas.1319590110. [Online]. Available: https://doi.org/10.1073/pnas.1319590110.

# A

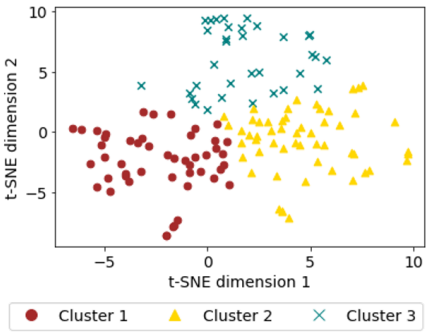
# Appendix

A

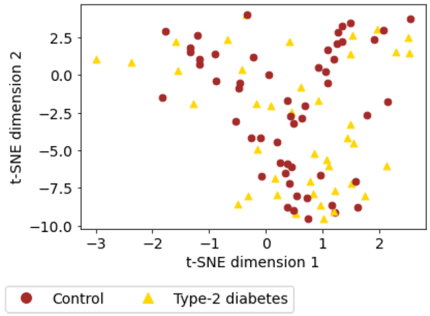
## A.1. BATCH EFFECTS



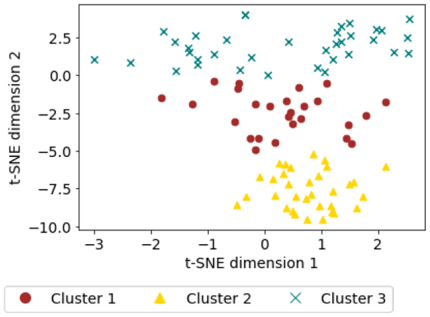

(a) First two components of TSNE on colorectal cancer dataset




(b) K-means clustering on TSNE components of colorectal cancer data




(c) First two components of TSNE on type-2 diabetes dataset




(d) K-means clustering on TSNE components of type-2 diabetes data

Figure A.1: Batch effect analysis on colorectal cancer [50] and type-2 diabetes datasets [51]

Table A.1: Analysis of cluster trueness versus randomness to find batch effects.

| Dataset | Adjusted rand index | Adjusted mutual information index |
|---|---|---|
| Colorectal cancer [50] | -0.0020 | -0.0021 |
| Type-2 diabetes [51] | -0.0095 | -0.0097 |

## A.2. KRAKEN2 RANK

Table A.2: Analysis to find taxonomy rank with most number of features for K-mer length of 31.

| Dataset | Species | Genus | Family | Order | Class | Phylum | Kingdom | Domain |
|---|---|---|---|---|---|---|---|---|
| Colorectal cancer [50] | 5285 | 1464 | 422 | 190 | 86 | 45 | 1 | 4 |
| Type-2 diabetes [51] | 5779 | 1569 | 440 | 197 | 88 | 48 | 3 | 4 |

## A.3. COLORECTAL CANCER CLASSIFICATION HYPERPARAMETERS

Table A.3: Training data performance of proposed representation over different embedding lengths and one hidden layer for colorectal cancer classification. The representations were trained using logistic regression with five-fold cross-validation. The reported results are an average and standard deviation of performance in 100 randomly sampled validation sets of size 81.

| Embedding length | Accuracy | Precision | Recall | F1 | Area under the curve | Confusion Matrix |
|---|---|---|---|---|---|---|
| 1 | $0.40 \pm 0.03$ | $0.33 \pm 0.08$ | $0.40 \pm 0.03$ | $0.33 \pm 0.05$ | $0.58 \pm 0.04$ | $\begin{bmatrix} 17.75 \pm 2.93 & 6.85 \pm 3.98 & 2.40 \pm 3.09 \\ 11.40 \pm 4.93 & 11.95 \pm 6.34 & 3.65 \pm 4.34 \\ 13.90 \pm 4.13 & 10.20 \pm 5.93 & 2.90 \pm 5.04 \end{bmatrix}$ |
| 5 | $0.80 \pm 0.27$ | $0.75 \pm 0.36$ | $0.80 \pm 0.27$ | $0.75 \pm 0.34$ | $0.93 \pm 0.10$ | $\begin{bmatrix} 23.14 \pm 11.13 & 3.65 \pm 9.29 & 3.54 \pm 7.54 \\ 2.32 \pm 7.58 & 25.04 \pm 10.28 & 3.00 \pm 7.57 \\ 2.58 \pm 7.55 & 3.49 \pm 9.34 & 24.24 \pm 11.43 \end{bmatrix}$ |
| 20 | $0.65 \pm 0.05$ | $0.65 \pm 0.05$ | $0.65 \pm 0.05$ | $0.65 \pm 0.05$ | $0.84 \pm 0.03$ | $\begin{bmatrix} 18.80 \pm 2.16 & 5.15 \pm 1.56 & 3.05 \pm 1.36 \\ 6.15 \pm 1.49 & 15.45 \pm 2.18 & 5.40 \pm 1.32 \\ 3.05 \pm 1.16 & 5.15 \pm 1.62 & 18.80 \pm 1.69 \end{bmatrix}$ |
| 100 | $0.96 \pm 0.06$ | $0.97 \pm 0.05$ | $0.96 \pm 0.06$ | $0.96 \pm 0.06$ | $0.99 \pm 0.02$ | $\begin{bmatrix} 25.05 \pm 3.34 & 0.85 \pm 1.49 & 1.10 \pm 2.07 \\ 0.15 \pm 0.36 & 26.50 \pm 0.92 & 0.35 \pm 0.73 \\ 0.10 \pm 0.30 & 0.50 \pm 0.92 & 26.40 \pm 1.02 \end{bmatrix}$ |
| 500 | $0.40 \pm 0.03$ | $0.33 \pm 0.08$ | $0.40 \pm 0.03$ | $0.33 \pm 0.05$ | $0.58 \pm 0.04$ | $\begin{bmatrix} 17.75 \pm 2.93 & 6.85 \pm 3.98 & 2.40 \pm 3.09 \\ 11.40 \pm 4.93 & 11.95 \pm 6.34 & 3.65 \pm 4.34 \\ 13.90 \pm 4.14 & 10.20 \pm 5.93 & 2.90 \pm 5.04 \end{bmatrix}$ |
| 1000 | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $\begin{bmatrix} 27.00 \pm 0.00 & 0.00 \pm 0.00 & 0.00 \pm 0.00 \\ 0.00 \pm 0.00 & 27.00 \pm 0.00 & 0.00 \pm 0.00 \\ 0.00 \pm 0.00 & 0.00 \pm 0.00 & 27.00 \pm 0.00 \end{bmatrix}$ |
| 1500 | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $\begin{bmatrix} 27.00 \pm 0.00 & 0.00 \pm 0.00 & 0.00 \pm 0.00 \\ 0.00 \pm 0.00 & 27.00 \pm 0.00 & 0.00 \pm 0.00 \\ 0.00 \pm 0.00 & 0.00 \pm 0.00 & 27.00 \pm 0.00 \end{bmatrix}$ |

Table A.4: Training data performance of proposed representation over different number of hidden layers and an embedding length of 1000 for colorectal cancer classification. The representations were trained using logistic regression with five-fold cross-validation. The reported results are an average and standard deviation of performance in 100 randomly sampled validation sets of size 81.

| Number of hidden layers | Accuracy | Precision | Recall | F1 | Area under the curve | Confusion Matrix |
|---|---|---|---|---|---|---|
| 1 | $0.65 \pm 0.05$ | $0.65 \pm 0.05$ | $0.65 \pm 0.05$ | $0.65 \pm 0.05$ | $0.84 \pm 0.03$ | $\begin{bmatrix} 18.80 \pm 2.16 & 5.15 \pm 1.56 & 3.05 \pm 1.36 \\ 6.15 \pm 1.49 & 15.45 \pm 2.18 & 5.40 \pm 1.32 \\ 3.05 \pm 1.16 & 5.15 \pm 1.62 & 18.80 \pm 1.69 \end{bmatrix}$ |
| 3 | $0.77 \pm 0.03$ | $0.78 \pm 0.03$ | $0.77 \pm 0.03$ | $0.77 \pm 0.03$ | $0.90 \pm 0.01$ | $\begin{bmatrix} 21.34 \pm 1.64 & 3.08 \pm 1.45 & 2.58 \pm 1.02 \\ 3.28 \pm 1.18 & 21.54 \pm 1.59 & 2.18 \pm 1.12 \\ 3.94 \pm 1.49 & 3.58 \pm 1.74 & 19.48 \pm 1.82 \end{bmatrix}$ |
| 5 | $0.92 \pm 0.02$ | $0.92 \pm 0.02$ | $0.92 \pm 0.02$ | $0.92 \pm 0.02$ | $0.98 \pm 0.01$ | $\begin{bmatrix} 25.36 \pm 1.11 & 0.84 \pm 0.78 & 0.80 \pm 0.80 \\ 1.74 \pm 0.82 & 24.22 \pm 1.15 & 1.04 \pm 0.80 \\ 0.92 \pm 0.93 & 1.08 \pm 0.52 & 25.00 \pm 1.11 \end{bmatrix}$ |
| 7 | $0.94 \pm 0.02$ | $0.94 \pm 0.02$ | $0.94 \pm 0.02$ | $0.94 \pm 0.02$ | $0.98 \pm 0.01$ | $\begin{bmatrix} 25.42 \pm 1.15 & 0.86 \pm 0.92 & 0.72 \pm 0.72 \\ 1.08 \pm 0.87 & 25.12 \pm 1.18 & 0.80 \pm 0.92 \\ 0.84 \pm 0.81 & 0.80 \pm 0.69 & 25.36 \pm 0.95 \end{bmatrix}$ |

A

## A.4. TYPE-2 DIABETES CLASSIFICATION HYPERPARAMETERS

Table A.5: Training data performance of proposed representation over different embedding lengths and one hidden layer for type-2 diabetes classification. The representations were trained using logistic regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled validation sets of size 42.

| Embedding length | Accuracy | Precision | Recall | F1 | Area under the curve | Confusion Matrix |
|---|---|---|---|---|---|---|
| 1 | $0.55 \pm 0.05$ | $0.53 \pm 0.16$ | $0.55 \pm 0.05$ | $0.50 \pm 0.10$ | $0.54 \pm 0.03$ | $\begin{bmatrix} 13.45 \pm 6.95 & 7.55 \pm 6.95 \\ 11.24 \pm 5.94 & 9.76 \pm 5.94 \end{bmatrix}$ |
| 5 | $0.60 \pm 0.06$ | $0.60 \pm 0.10$ | $0.60 \pm 0.06$ | $0.58 \pm 0.09$ | $0.66 \pm 0.05$ | $\begin{bmatrix} 13.47 \pm 4.13 & 7.53 \pm 4.13 \\ 9.22 \pm 4.36 & 11.78 \pm 4.36 \end{bmatrix}$ |
| 20 | $0.79 \pm 0.09$ | $0.81 \pm 0.07$ | $0.79 \pm 0.09$ | $0.79 \pm 0.10$ | $0.88 \pm 0.06$ | $\begin{bmatrix} 15.36 \pm 3.78 & 5.64 \pm 3.78 \\ 3.06 \pm 1.82 & 17.94 \pm 1.82 \end{bmatrix}$ |
| 100 | $0.95 \pm 0.06$ | $0.95 \pm 0.05$ | $0.95 \pm 0.06$ | $0.95 \pm 0.06$ | $0.99 \pm 0.01$ | $\begin{bmatrix} 20.21 \pm 1.43 & 0.79 \pm 1.43 \\ 1.44 \pm 2.23 & 19.56 \pm 2.23 \end{bmatrix}$ |
| 500 | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $\begin{bmatrix} 21.00 \pm 0.00 & 0.00 \pm 0.00 \\ 0.00 \pm 0.00 & 21.00 \pm 0.00 \end{bmatrix}$ |
| 1000 | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $\begin{bmatrix} 21.00 \pm 0.00 & 0.00 \pm 0.00 \\ 0.00 \pm 0.00 & 21.00 \pm 0.00 \end{bmatrix}$ |
| 1500 | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $\begin{bmatrix} 21.00 \pm 0.00 & 0.00 \pm 0.00 \\ 0.00 \pm 0.00 & 21.00 \pm 0.00 \end{bmatrix}$ |

Table A.6: Training data performance of proposed representation over different number of hidden layers and an embedding length of 1000 for type-2 diabetes classification. The representations were trained using logistic regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled validation sets of size 42.

| Number of hidden layers | Accuracy | Precision | Recall | F1 | Area under the curve | Confusion Matrix |
|---|---|---|---|---|---|---|
| 1 | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $1.00 \pm 0.00$ | $\begin{bmatrix} 21.00 \pm 0.00 & 0.00 \pm 0.00 \\ 0.00 \pm 0.00 & 21.00 \pm 0.00 \end{bmatrix}$ |
| 3 | $0.87 \pm 0.13$ | $0.90 \pm 0.09$ | $0.87 \pm 0.13$ | $0.86 \pm 0.15$ | $0.97 \pm 0.03$ | $\begin{bmatrix} 19.89 \pm 1.57 & 1.11 \pm 1.57 \\ 4.30 \pm 5.43 & 16.70 \pm 5.43 \end{bmatrix}$ |
| 5 | $0.97 \pm 0.04$ | $0.97 \pm 0.04$ | $0.97 \pm 0.04$ | $0.97 \pm 0.04$ | $1.00 \pm 0.00$ | $\begin{bmatrix} 20.76 \pm 0.85 & 0.24 \pm 0.85 \\ 1.07 \pm 1.67 & 19.93 \pm 1.67 \end{bmatrix}$ |
| 7 | $0.98 \pm 0.06$ | $0.98 \pm 0.04$ | $0.98 \pm 0.06$ | $0.98 \pm 0.07$ | $1.00 \pm 0.00$ | $\begin{bmatrix} 20.88 \pm 0.35 & 0.12 \pm 0.35 \\ 0.79 \pm 2.44 & 20.21 \pm 2.44 \end{bmatrix}$ |

## A.5. TYPE-2 DIABETES REGRESSION HYPERPARAMETERS



Table A.7: Validation performance of proposed representation over different embedding lengths for high density lipoprotein level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled validation sets of size 20.

| Embedding length | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| 1 | $0.06 \pm 0.07$ | $0.04 \pm 0.01$ | $0.16 \pm 0.02$ |
| 5 | $0.04 \pm 0.06$ | $0.05 \pm 0.01$ | $0.16 \pm 0.02$ |
| 20 | $0.05 \pm 0.07$ | $0.05 \pm 0.02$ | $0.16 \pm 0.03$ |
| 100 | $0.03 \pm 0.04$ | $0.40 \pm 0.12$ | $0.51 \pm 0.07$ |
| 500 | $0.07 \pm 0.08$ | $0.04 \pm 0.01$ | $0.16 \pm 0.02$ |
| 1000 | $0.10 \pm 0.05$ | $0.05 \pm 0.01$ | $0.57 \pm 0.07$ |
| 1500 | $0.07 \pm 0.08$ | $0.04 \pm 0.01$ | $0.16 \pm 0.02$ |

Table A.8: Validation performance of proposed representation over different embedding lengths for triglyceride level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled validation sets of size 20.

| Embedding length | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| 1 | $0.07 \pm 0.08$ | $0.66 \pm 0.28$ | $0.58 \pm 0.09$ |
| 5 | $0.06 \pm 0.07$ | $0.69 \pm 0.33$ | $0.59 \pm 0.10$ |
| 20 | $0.07 \pm 0.08$ | $0.64 \pm 0.30$ | $0.58 \pm 0.10$ |
| 100 | $0.05 \pm 0.06$ | $0.63 \pm 0.28$ | $0.58 \pm 0.09$ |
| 500 | $0.04 \pm 0.05$ | $0.64 \pm 0.31$ | $0.58 \pm 0.10$ |
| 1000 | $0.07 \pm 0.08$ | $0.63 \pm 0.28$ | $0.58 \pm 0.09$ |
| 1500 | $0.05 \pm 0.06$ | $0.70 \pm 0.31$ | $0.59 \pm 0.09$ |

Table A.9: Validation performance of proposed representation over different embedding lengths for fasting blood glucose level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled validation sets of size 20

| Embedding length | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| 1 | $0.07 \pm 0.08$ | $4.77 \pm 1.61$ | $1.73 \pm 0.22$ |
| 5 | $0.03 \pm 0.04$ | $4.94 \pm 1.53$ | $1.76 \pm 0.22$ |
| 20 | $0.03 \pm 0.04$ | $5.07 \pm 1.56$ | $1.78 \pm 0.21$ |
| 100 | $0.06 \pm 0.06$ | $4.44 \pm 1.38$ | $1.69 \pm 0.21$ |
| 500 | $0.02 \pm 0.04$ | $5.25 \pm 1.50$ | $1.81 \pm 0.21$ |
| 1000 | $0.07 \pm 0.08$ | $4.74 \pm 1.59$ | $1.74 \pm 0.22$ |
| 1500 | $0.03 \pm 0.05$ | $5.06 \pm 1.50$ | $1.76 \pm 0.22$ |

Table A.10: Validation performance of proposed representation over different embedding lengths for systolic blood pressure level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled validation sets of size 20

| Embedding length | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| 1 | $0.05 \pm 0.07$ | $195.608 \pm 45.56$ | $11.46 \pm 1.45$ |
| 5 | $0.11 \pm 0.09$ | $190.00 \pm 45.98$ | $11.25 \pm 1.41$ |
| 20 | $0.03 \pm 0.04$ | $197.89 \pm 49.12$ | $11.40 \pm 1.46$ |
| 100 | $0.06 \pm 0.06$ | $200.19 \pm 52.30$ | $11.52 \pm 1.60$ |
| 500 | $0.03 \pm 0.04$ | $206.15 \pm 53.87$ | $11.72 \pm 1.72$ |
| 1000 | $0.40 \pm 0.06$ | $204.41 \pm 46.80$ | $11.59 \pm 1.49$ |
| 1500 | $0.08 \pm 0.08$ | $185.97 \pm 43.82$ | $11.08 \pm 1.46$ |

Table A.11: Validation performance of proposed representation over different embedding lengths for diastolic blood pressure level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled validation sets of size 20

| Embedding length | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| 1 | $0.07 \pm 0.08$ | $84.16 \pm 23.79$ | $7.47 \pm 1.00$ |
| 5 | $0.09 \pm 0.10$ | $83.12 \pm 21.46$ | $7.43 \pm 0.85$ |
| 20 | $0.02 \pm 0.03$ | $82.38 \pm 23.18$ | $7.45 \pm 0.96$ |
| 100 | $0.08 \pm 0.08$ | $85.52 \pm 24.32$ | $7.43 \pm 1.09$ |
| 500 | $0.04 \pm 0.06$ | $80.16 \pm 22.17$ | $7.36 \pm 0.92$ |
| 1000 | $0.04 \pm 0.05$ | $86.65 \pm 23.35$ | $7.60 \pm 1.02$ |
| 1500 | $0.05 \pm 0.07$ | $86.02 \pm 25.74$ | $7.58 \pm 1.08$ |

Table A.12: Validation performance of proposed representation over different embedding lengths for fasting serum insulin level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled validation sets of size 20

| Embedding length | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| 1 | $0.04 \pm 0.06$ | $33.18 \pm 11.31$ | $4.46 \pm 0.76$ |
| 5 | $0.14 \pm 0.10$ | $30.92 \pm 8.82$ | $4.48 \pm 0.63$ |
| 20 | $0.03 \pm 0.04$ | $35.98 \pm 10.14$ | $4.66 \pm 0.71$ |
| 100 | $0.09 \pm 0.09$ | $34.34 \pm 11.01$ | $4.53 \pm 0.78$ |
| 500 | $0.10 \pm 0.10$ | $33.05 \pm 10.25$ | $4.50 \pm 0.75$ |
| 1000 | $0.04 \pm 0.05$ | $86.65 \pm 23.35$ | $7.60 \pm 1.02$ |
| 1500 | $0.02 \pm 0.03$ | $34.57 \pm 8.61$ | $4.54 \pm 0.59$ |

Table A.13: Validation performance of proposed representation over different embedding lengths for glycosylated hemoglobin level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled validation sets of size 20

| Embedding length | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| 1 | $0.04 \pm 0.05$ | $4.37 \pm 1.42$ | $1.69 \pm 0.19$ |
| 5 | $0.03 \pm 0.05$ | $4.12 \pm 1.24$ | $1.68 \pm 0.18$ |
| 20 | $0.03 \pm 0.05$ | $4.12 \pm 1.27$ | $1.66 \pm 0.19$ |
| 100 | $0.04 \pm 0.04$ | $4.39 \pm 1.44$ | $1.69 \pm 0.20$ |
| 500 | $0.03 \pm 0.04$ | $4.15 \pm 1.19$ | $1.69 \pm 0.19$ |
| 1000 | $0.03 \pm 0.04$ | $4.27 \pm 1.32$ | $1.69 \pm 0.19$ |
| 1500 | $0.04 \pm 0.05$ | $4.05 \pm 1.22$ | $1.64 \pm 0.17$ |

Table A.14: Validation performance of proposed representation over different embedding lengths for total cholesterol level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled validation sets of size 20

| Embedding length | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| 1 | $0.04 \pm 0.04$ | $0.39 \pm 0.11$ | $0.50 \pm 0.07$ |
| 5 | $0.06 \pm 0.06$ | $0.04 \pm 0.12$ | $0.50 \pm 0.07$ |
| 20 | $0.05 \pm 0.07$ | $0.40 \pm 0.12$ | $0.50 \pm 0.08$ |
| 100 | $0.03 \pm 0.04$ | $0.40 \pm 0.12$ | $0.51 \pm 0.07$ |
| 500 | $0.07 \pm 0.07$ | $0.39 \pm 0.11$ | $0.51 \pm 0.07$ |
| 1000 | $0.04 \pm 0.05$ | $0.40 \pm 0.13$ | $0.50 \pm 0.08$ |
| 1500 | $0.04 \pm 0.05$ | $0.42 \pm 0.14$ | $0.51 \pm 0.08$ |

A

## A.6. TYPE-2 DIABETES REGRESSION PERFORMANCE ON TEST DATA

Table A.15: Test performance for high density lipoprotein level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled test sets of size 10.

| Approach | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| Proposed representation with embedding length = 1000 | $0.12 \pm 0.15$ | $0.03 \pm 0.01$ | $0.14 \pm 0.03$ |
| Kraken2 representation | $0.16 \pm 0.15$ | $0.03 \pm 0.01$ | $0.15 \pm 0.03$ |
| Randomly sampled values | $0.10 \pm 0.12$ | $0.07 \pm 0.03$ | $0.21 \pm 0.05$ |

Table A.16: Test performance for triglyceride level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled test sets of size 10.

| Approach | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| Proposed representation with embedding length = 1 | $0.13 \pm 0.12$ | $0.85 \pm 0.28$ | $0.71 \pm 0.13$ |
| Kraken2 representation | $0.06 \pm 0.07$ | $0.90 \pm 0.24$ | $0.77 \pm 0.12$ |
| Randomly sampled values | $0.12 \pm 0.14$ | $1.51 \pm 0.67$ | $0.96 \pm 0.24$ |

Table A.17: Test performance for fasting blood glucose level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled test sets of size 10.

| Approach | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| Proposed representation with embedding length = 1 | $0.04 \pm 0.07$ | $10.64 \pm 3.85$ | $2.60 \pm 0.41$ |
| Kraken2 representation | $0.15 \pm 0.14$ | $10.49 \pm 3.81$ | $2.40 \pm 0.46$ |
| Randomly sampled values | $0.16 \pm 0.16$ | $20.21 \pm 9.57$ | $3.53 \pm 0.87$ |

Table A.18: Test performance for systolic blood pressure level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled test sets of size 10.

| Approach | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| Proposed representation with embedding length = 5 | $0.13 \pm 0.10$ | $125.57 \pm 34.71$ | $9.42 \pm 1.32$ |
| Kraken2 representation | $0.10 \pm 0.08$ | $210.96 \pm 58.17$ | $11.60 \pm 1.92$ |
| Randomly sampled values | $0.10 \pm 0.11$ | $265.77 \pm 101.15$ | $13.10 \pm 2.96$ |

Table A.19: Test performance for diastolic blood pressure level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled test sets of size 10.

| Approach | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| Proposed representation with embedding length = 5 | $0.10 \pm 0.12$ | $59.04 \pm 14.93$ | $6.62 \pm 0.96$ |
| Kraken2 representation | $0.05 \pm 0.07$ | $86.62 \pm 26.03$ | $7.27 \pm 1.29$ |
| Randomly sampled values | $0.10 \pm 0.12$ | $114.62 \pm 44.55$ | $8.74 \pm 1.87$ |

Table A.20: Test performance for fasting serum insulin level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled test sets of size 10.

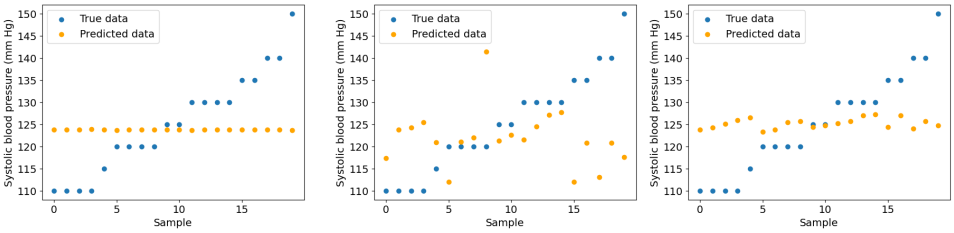
| Approach | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| Proposed representation with embedding length = 5 | $0.16 \pm 0.15$ | $84.39 \pm 26.32$ | $7.55 \pm 1.02$ |
| Kraken2 representation | $0.09 \pm 0.10$ | $78.70 \pm 13.92$ | $7.85 \pm 0.76$ |
| Randomly sampled values | $0.11 \pm 0.15$ | $165.42 \pm 76.93$ | $10.28 \pm 2.42$ |

Table A.21: Test performance for glycosylated hemoglobin level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled test sets of size 10.

| Approach | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| Proposed representation with embedding length = 1 | $0.07 \pm 0.08$ | $5.45 \pm 1.14$ | $2.08 \pm 0.25$ |
| Kraken2 representation | $0.17 \pm 0.16$ | $5.38 \pm 1.36$ | $1.87 \pm 0.30$ |
| Randomly sampled values | $0.12 \pm 0.15$ | $10.28 \pm 4.13$ | $2.59 \pm 0.57$ |

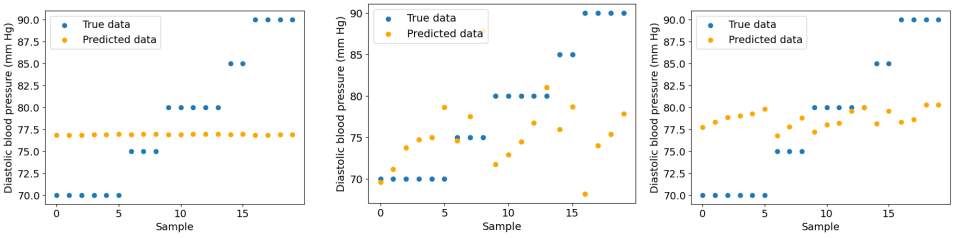
Table A.22: Test performance for total cholesterol level prediction. The representations were trained using ridge regression with five-fold cross-validation. The reported results were averaged over 100 randomly sampled test sets of size 10.

| Approach | r2 | Mean squared error | Mean absolute error |
|---|---|---|---|
| Proposed representation with embedding length = 500 | $0.08 \pm 0.08$ | $0.95 \pm 0.55$ | $0.63 \pm 0.17$ |
| Kraken2 representation | $0.10 \pm 0.10$ | $0.94 \pm 0.48$ | $0.65 \pm 0.17$ |
| Randomly sampled values | $0.11 \pm 0.13$ | $1.47 \pm 0.68$ | $0.94 \pm 0.22$ |



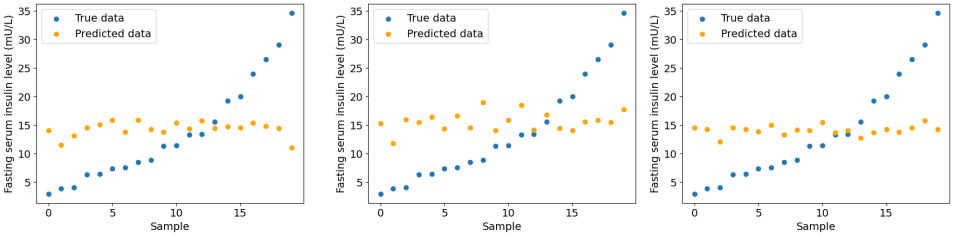

(a) Proposed representation for embedding length = 5 (b) Kraken2 representation (c) Randomly sampled values

Figure A.2: Scatter plot of true and predicted values for systolic blood pressure prediction



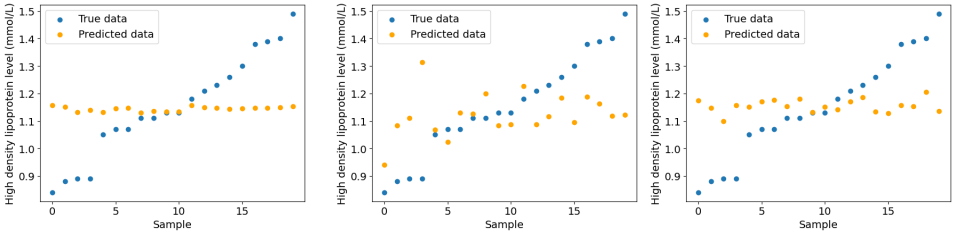

(a) Proposed representation for embedding length = 5 (b) Kraken2 representation (c) Randomly sampled values

Figure A.3: Scatter plot of true and predicted values for diastolic blood pressure prediction




(a) Proposed representation for embedding length = 5 (b) Kraken2 representation (c) Randomly sampled values

Figure A.4: Scatter plot of true and predicted values for fasting serum insulin level prediction

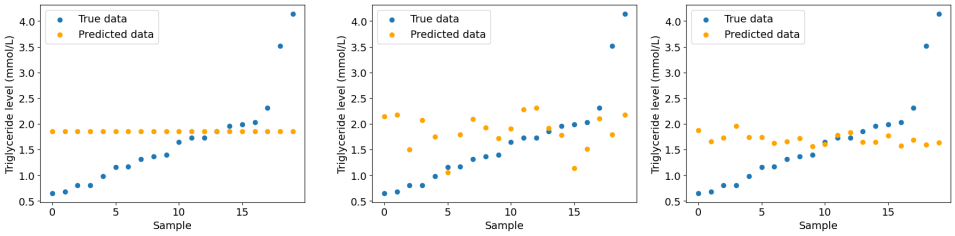



(a) Proposed representation for embedding length = 1000

(b) Kraken2 representation

(c) Randomly sampled values

Figure A.5: Scatter plot of true and predicted values for high density lipoprotein level prediction

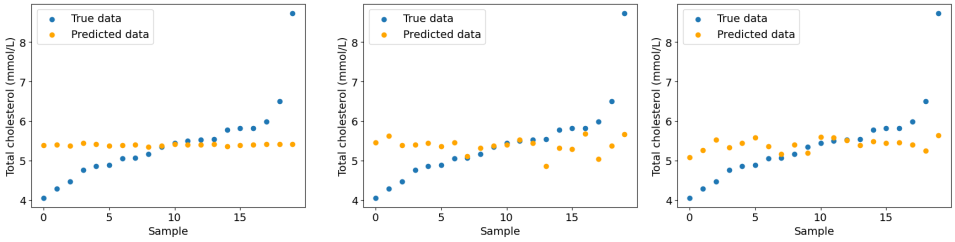



(a) Proposed representation for embedding length = 1

(b) Kraken2 representation

(c) Randomly sampled values

Figure A.6: Scatter plot of true and predicted values for triglyceride level prediction

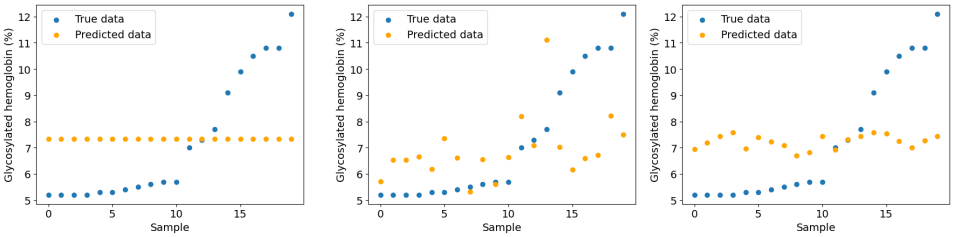



(a) Proposed representation for embedding length = 500

(b) Kraken2 representation

(c) Randomly sampled values

Figure A.7: Scatter plot of true and predicted values for total cholesterol level prediction

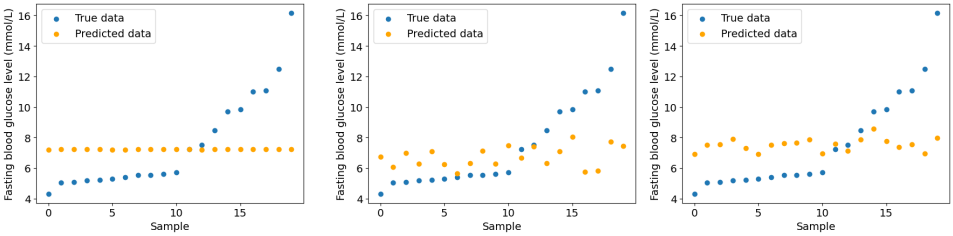



(a) Proposed representation for embedding length = 1

(b) Kraken2 representation

(c) Randomly sampled values

Figure A.8: Scatter plot of true and predicted values for total glycosylated hemoglobin level prediction




(a) Proposed representation for embedding length = 1

(b) Kraken2 representation

(c) Randomly sampled values

Figure A.9: Scatter plot of true and predicted values for fasting blood glucose level prediction

## A.7. HIDDEN LAYER ARCHITECTURE



This section includes the architecture used to increase the model depth. The architecture below is shown for colorectal cancer dataset with 138 samples, and for an embedding length of 1000. However, the same model was used for type-2 diabetes dataset with 95 samples.

1. Three hidden layers

```
Autoencoder(
    (encoder): Encoder(
        (linear_sample_1): Linear(in_features=138,
        out_features=64, bias=True)
        (linear_sample_2): Linear(in_features=64, out_features=1000,
        bias=True)

        (linear_context_1): Linear(in_features=124, out_features=512,
        bias=True)
        (linear_context_2): Linear(in_features=512, out_features=1000,
        bias=True)

        (relu): ReLU()
  )
    (decoder): Decoder(
        (linear_1): Linear(in_features=embedding length, out_features=64,
        bias=True)
        (linear_2): Linear(in_features=64, out_features=124, bias=True)

        (tanh): Tanh()
        (relu): ReLU()
        (sigmoid): Sigmoid()
  )
)
```

A

2. Five hidden layers

```
Autoencoder(
    (encoder): Encoder(
        (linear_sample_1): Linear(in_features=138, out_features=128,
        bias=True)
        (linear_sample_2): Linear(in_features=128, out_features=64,
        bias=True)
        (linear_sample_3): Linear(in_features=64, out_features=1000,
        bias=True)

        (linear_context_1): Linear(in_features=124, out_features=512,
        bias=True)
        (linear_context_2): Linear(in_features=512, out_features=128,
        bias=True)
        (linear_context_3): Linear(in_features=128, out_features=1000,
        bias=True)

        (relu): ReLU()
    )
    (decoder): Decoder(
        (linear_1): Linear(in_features=1000, out_features=64, bias=True)
        (linear_2): Linear(in_features=64, out_features=100, bias=True)
        (linear_3): Linear(in_features=100, out_features=124, bias=True)

        (tanh): Tanh()
        (relu): ReLU()
        (sigmoid): Sigmoid()
    )
)
```

3. Seven hidden layers

```
Autoencoder(
    (encoder): Encoder(
        (linear_sample_1): Linear(in_features=138, out_features=128,
        bias=True)
        (linear_sample_2): Linear(in_features=128, out_features=64,
        bias=True)
        (linear_sample_3): Linear(in_features=64, out_features=48,
        bias=True)
        (linear_sample_4): Linear(in_features=48, out_features=1000,
        bias=True)

        (linear_context_1): Linear(in_features=124, out_features=512,
        bias=True)
        (linear_context_2): Linear(in_features=512, out_features=256,
        bias=True)
        (linear_context_3): Linear(in_features=256, out_features=64,
        bias=True)
        (linear_context_4): Linear(in_features=64, out_features=1000,
        bias=True)

        (relu): ReLU()
    )
    (decoder): Decoder(
        (linear_1): Linear(in_features=1000, out_features=64,
        bias=True)
        (linear_2): Linear(in_features=64, out_features=80,
        bias=True)
        (linear_3): Linear(in_features=80, out_features=100,
        bias=True)
        (linear_4): Linear(in_features=100, out_features=124,
        bias=True)

        (tanh): Tanh()
        (relu): ReLU()
        (sigmoid): Sigmoid()
    )
)
```