JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 28, Number 11, 2021

Pp. 1063–1074
DOI: 10.1089/cmb.2021.0273

MetaMLP: A Fast Word Embedding Based Classifier to Profile Target Gene Databases in Metagenomic Samples

GUSTAVO A. ARANGO-ARGOTY,[1] LENWOOD S. HEATH,[1,i] AMY PRUDEN,[2] PETER J. VIKESLAND,[2] and LIQING ZHANG[1,ii]

ABSTRACT

The functional profile of metagenomic samples enables improved understanding of microbial populations in the environment. Such analysis consists of assigning short sequencing reads to a particular functional category. Normally, manually curated databases are used for functional assignment, and genes are arranged into different classes. Sequence alignment has been widely used to profile metagenomic samples against curated databases. However, this method is time consuming and requires high computational resources. While several alignment-free methods based on *k*-mer composition have been developed in recent years, they still require large amounts of computer main memory. In this article, MetaMLP (Metagenomics Machine Learning Profiler), a machine learning method that represents sequences as numerical vectors (embeddings) and uses a simple one hidden layer neural network to profile functional categories, is developed. Unlike other methods, MetaMLP enables partial matching by using a reduced alphabet to build sequence embeddings from full and partial *k*-mers. MetaMLP is able to identify a slightly larger number of reads compared with DIAMOND (one of the fastest sequence alignment methods), as well as to perform accurate predictions with 0.99 precision and 0.99 recall. MetaMLP can process 100M reads in ∼10 minutes on a laptop computer, which is 50 times faster than DIAMOND.

Keywords: antibiotic resistance, metagenomic, short reads, word embedding.

1. INTRODUCTION

THE WIDE AND RAPID ADOPTION of next-generation sequencing (NGS) techniques such as metagenomics for the analysis of microbial diversity, antibiotic resistance, and other functional profiling analyses creates a gap between scalability and processing efficiency. In other words, large amounts of data require the design of computational tools that are both accurate and fast. Sequence comparison algorithms such as BLAST (Altschul et al., 1990), FASTA (Pearson, 1990), HMMER (Finn et al., 2011), and PSI-BLAST (Altschul et al., 1997) were created with the aim to find correspondence of sequence distributions in two or more sequences. To date, BLAST is the most popular and trusted tool for sequence alignment.

Departments of [1]Computer Science and [2]Civil and Environmental Engineering, Virginia Tech, Blacksburg, Virginia, USA.
[i]ORCID ID (https://orcid.org/0000-0003-1608-431X).
[ii]ORCID ID (https://orcid.org/0000-0003-4660-9199).

However, BLAST does not scale well when comparing millions of sequences. The reason behind this issue is that BLAST uses a computationally demanding strategy consisting of a seed and extend algorithm (Li and Homer, 2010). Although sequence alignment is considered the gold standard approach for sequence analysis, there have been several cases where this technique has produced dubious results (Zielezinski et al., 2017). For instance, alignment-based methods assume that homologous sequences share a certain degree of conservation. Although this assumption is considered to be true when analyzing conserved domains, organisms such as viruses that exhibit extensive mutation challenge this conservation principle. When analyzing short sequences (e.g., Illumina sequencing reads), the percentage of identity does not guarantee an accurate implication. For example, highly identical sequences do not imply homology (Bengtsson-Palme et al., 2017). In the opposite case, sequences with <30% identity can potentially be considered as homologous (Pearson, 2013).

DIAMOND (Buchfink et al., 2015), BLAT (Kent, 2002), USEARCH (Edgar, 2015), and RAPSearch (Ye et al., 2011) are alternatives to BLASTX that can run much faster, but with lost sensitivity. Particularly, the dramatic speedup of DIAMOND (20,000 times) is achieved by using a double indexing strategy, spaced seeds (longer seeds where not all positions are used), and a reduced alphabet. In detail, DIAMOND implements a seed and extend algorithm that first indexes both query and reference sequences. Then, the list of seeds in both the query and reference is linearly traversed to determine all the matched seeds and their locations. Finally, seeds are extended using the Smith–Waterman algorithm (Pearson, 1991).

Alignment-free methods have been proposed as an alternative to quantify sequence similarity without performing sequence alignment (Vinga and Almeida, 2003; Zielezinski et al., 2017; Pierce et al., 2019). These methods do not use the seed and extend paradigm. Therefore, their computational complexity is often linear and only depends on the query sequence length. In NGS, several alignment-free strategies have been developed for a number of different applications, including transcript quantification [kallisto (Weijers et al., 2012), sailfish (Patro et al., 2014), Salmon (Patro et al., 2015), and RNA-Skim (Zhang and Wang, 2014)], variant calling [ChimeRScope (Li et al., 2017), FastGT (Pajuste et al., 2017)], de novo genome assembly [minimap (Li, 2016), MHAP (Berlin et al., 2015)], and the profiling of metagenomics taxonomy by using a *k*-mer matching approach [Kraken (Wood and Salzberg, 2014), Mash (Ondov et al., 2016), CLARK (Ounit et al., 2015), and stringMLST (Gupta et al., 2016)].

The word embedding technique, where words are represented as a numerical vector, is one of the most successful learning methods applied in natural language processing. For instance, the Word2vec technique (Goldberg and Levy, 2014) uses a shallow two-layer neural network to train and aggregate word embeddings by using the continuous bag of words (CBOW) approach. In this manner, semantic associations for a target word are established given its context. The concept of using word vectors to represent protein or DNA sequences is not new and has been explored previously. For instance, DNA2Vec (Ng, 2017) explores the associations between variable length *k*-mers to generate an embedding space that correlates with sequence alignment. Yang et al. (2018) explored the performance of word embeddings for classification of protein functions compared with classical representation techniques and demonstrated that *k*-mer embeddings outperformed other techniques. Embeddings in the study were learned in an unsupervised way, that is, the embeddings were learned first, and then, the classifier was built using those embeddings.

In this article, MetaMLP, an alignment-free method that uses word embeddings to represent target protein databases, is described for the functional profiling of metagenomic samples. The strategy behind MetaMLP relies on the CBOW model. However, the target word is replaced by the label or functional class of the sequence, and the context words correspond to *k*-mers and fragmented *k*-mers. Therefore, MetaMLP is a novel strategy that uses a combination of hash indexing, six open reading frame translation, a reduced amino acid alphabet, and an embedding representation to process metagenomic data. In addition, MetaMLP incorporates the C++ FastText (Joulin et al., 2016) library and is composed of two main stages: MetaMLP-index that processes protein sequences to build a machine learning model and MetaMLP-classify to annotate reads from metagenomic DNA sequence libraries.

2. METHODS

The overall structure of MetaMLP is shown in Figure 1 and consists of two main components: (1) an indexing stage that processes protein reference sequences into a word vector representation to train a classifier, and (2) a prediction stage that processes short sequencing reads and classifies them into one of the predefined classes from the reference database.

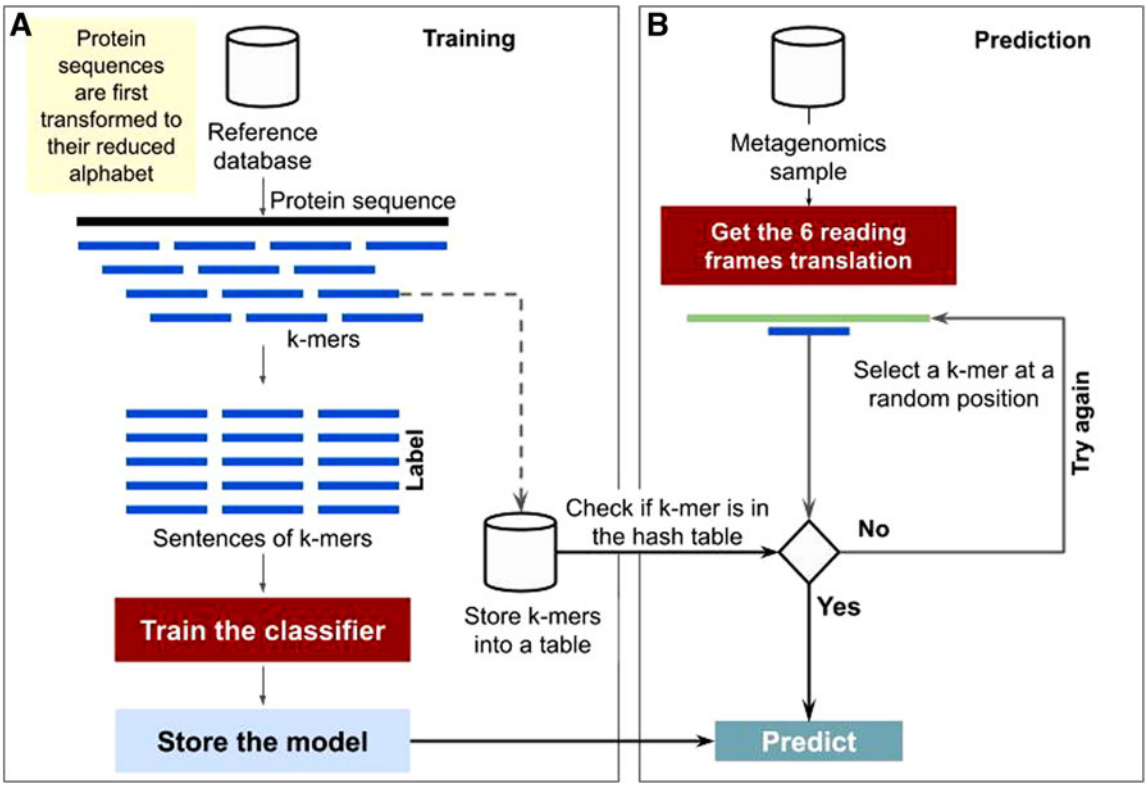

FIG. 1. Overview of MetaMLP. **(A)** *k*-mer sentences extracted from reference proteins are used to train the machine learning model. **(B)** Once a model is trained, it will be used later to profile short sequencing reads to produce a relative abundance profile and the individual predictions of the reads.

2.1. Indexing protein reference databases

2.1.1. Reference database preprocessing. To increase the chances of detecting sequences with mismatches, the reference proteins are first transformed into their equivalent 10 amino acid alphabet version using the murphy.10 alphabet representation used in rapsearch (A [KR] [EDNQ] C G H [ILVM] [FYW] P [ST]) (Zhao et al., 2011). Then, *k*-mers of a fixed length are extracted from each protein sequence. However, to consider all *k*-mers within a sequence, a sliding window of one amino acid is used. Thus, each protein comprises *k* versions, each one corresponding to a different starting location [1, …, *k*]. Thereafter, a ''sentence'' of *k*-mers is extracted by taking three to five consecutive *k*-mers (the equivalent of reads of 100 to 150 bp) (Fig. 1A). At the same time, a table with unique *k*-mers is built and stored to later be used for filtering sequences that diverge greatly from the reference database during the prediction stage.

Formally, proteins are strings of amino acids arranged in a particular order. Thus, the protein P is represented by $P=\{p_1, \ldots, p_i, \ldots, p_N\}$, $p_i \in \{A, K, R, E, D, N, Q, C, G, H, I, L, V, M, F, Y, W, P, S, T\}$, where each position corresponds to 1 of the 20 amino acids found in nature, and N corresponds to the length of the protein sequence. The protein P can be reduced to its simplified 10 amino acids version $R=\{r_1, \ldots, r_N\}$, where r_i corresponds to one of the reduced alphabet amino acids. A *k*-mer $k_{i,l}$ of length l is defined as a substring of size l within the protein sequence R at the position i, such that $k_{i,l}=R[i:i+l]=\{r_i, r_{i+1}, \ldots, r_{i+l-1}\}$, where $1 \leq i \leq N-l+1$. Finally, the sentence of *k*-mers s_i, starting at the position i is defined as $s_i=\{k_{i,l}, k_{i+l,l}, \ldots, k_{i+(m-1)*l,l}\}$, where m corresponds to the number of *k*-mers used to build the sentence. Therefore, the protein sequence R can be represented as a set of sentences $S=\{s_1, s_2, \ldots, s_p, \ldots, s_{n-m*l-1}\}$, $1 \leq p \leq N-m*l+1$.

2.1.2. Training. MetaMLP uses the FastText (Joulin et al., 2016) implementation of the CBOW technique to learn the semantic relations between protein sequences and their labels by using the protein *k*-mers (Fig. 1A). In detail, a protein sequence is represented as a series of *k*-mer sentences S (analog to sentence of words in text documents), where each sentence $s=\{k_1, k_2, \ldots, k_j, \ldots, k_m\}$ is composed of a

set of *k*-mers. The supervised CBOW model learns the embedding space by looking at the *k*-mers distribution and the class label. Thus, the first weight matrix *A* in the classifier comprises the *k*-mer embeddings $X = \{x_1, \ldots, x_j, \ldots, x_m\}$, where x_j represents a *r*-dimensional vector. Then, these *k*-mer vectors are averaged into the vector $y = \frac{1}{m}\sum_{j=1}^{m} x_j$, that is supplied to a single hidden layer neural network. Then, it is multiplied by a sentence embeddings matrix *B* to output the probability distribution over the established classes by using a softmax layer *f*. For a total number of *Q k*-mer sentences in the reference data set, the classifier minimizes the negative log-likelihood over the reference classes as follows:

$$-\frac{1}{Q}\sum_{q=1}^{Q} c_q * log\left(f\left(B * A * X_q\right)\right),$$

where X_q corresponds to the *k*-mer vectors for the *q*-th sentence, $c_q$is the class label, *A* and *B* are the matrices, and *f* is the softmax function.

MetaMLP enables the bag of *n*-grams from FastText to capture partial information from the *k*-mers. These *n*-grams are subsequences from the *k*-mers passed along with the full size *k*-mer allowing to identify *k*-mers with partial matching.

2.2. Prediction of short sequencing reads

MetaMLP is designed to efficiently profile metagenomic samples consisting of millions of reads from short read sequencing libraries against a target reference database. As reads consist of nucleotides, MetaMLP first translates each sequence into six reading frames. Then, for each reading frame, a random *k*-mer is selected from its sequence and checked against the hash table that was built during the indexing stage. If a *k*-mer is found in the hash table, all *k*-mers are extracted from the read and classified using the trained CBOW model. If not, a new random *k*-mer is selected from the read at a different position. This process is repeated a maximum number of tries as defined by the user. If more than one reading frame is classified, MetaMLP picks up the reading frame with the highest classification probability (Fig. 1B).

Once a full metagenomic data set is processed, MetaMLP counts the number of reads per class using a minimum probability cutoff defined by the user and reports the absolute abundance table. Additionally, MetaMLP also reports a FASTA file containing the read name along with its classifications, probabilities, and sequence. This file is useful for cases where MetaMLP is used as a filter to target a particular functional class.

2.3. Databases

2.3.1. Pathway reference database. Bacterial protein sequences from the Universal Protein Resource (UniProt) were downloaded and filtered by only proteins that have been manually curated, have been reviewed, and contain evidence at the protein level. In total, 20,161 proteins were obtained, and 4105 of those were annotated to at least one pathway. Finally, pathways with fewer than 50 proteins were discarded to obtain a total of 3216 proteins and 21 different pathways (Supplementary Table S1).

2.3.2. Antibiotic resistance database. MetaMLP was trained to identify short reads associated with antibiotic resistance genes (ARGs) from metagenomic short sequencing data. Accordingly, the DeepARG-DB-v2 database (Arango-Argoty et al., 2018) containing a total of 12,260 sequences distributed through 30 antibiotic categories was downloaded. However, only antibiotic resistance categories with at least 50 protein sequences were considered for downstream analysis. Thus, a total of 12,147 proteins and 14 categories were used to train the MetaMLP model (Supplementary Table S2).

2.3.3. Gene Ontology reference database. Protein sequences associated with the bacterial stress response (GO:0006950) were downloaded from the UniProt web site. However, only bacterial curated sequences and biological processes with at least 100 sequences were considered for downstream analysis (Supplementary Table S3). In addition, the Gene Ontology (GO) database comprises proteins with multiple associated labels. For instance, the protein sequence Q55002 is associated with response to antibiotics (GO:0046677) and with translation (GO:0006412). Therefore, reads from this protein will be classified to

both categories. However, as MetaMLP uses a softmax layer for prediction, it will distribute the probability between both categories. In an ideal scenario, both classes would have a probability of 0.5. This database was used to test the ability of MetaMLP to represent sequences associated with multiple labels.

2.4. True positive data set

The pathway reference database was used to build a true positive database. Because MetaMLP uses amino acid sequences for training and nucleotide sequences for querying, it was necessary to identify the corresponding nucleotide sequences for each one of the proteins in the pathway reference database. Therefore, UniProt identifiers were cross-referenced against the RefSeq database, and a list of gene candidates was identified. Then, those candidates were aligned to the protein sequences using DIAMOND BlastX, with a 90% identity and a 90% overlap. If multiple alignments were obtained by this criterion, the best hit was selected as the representative gene sequence for the target protein sequence. Thus, each entry in the database contained a respective gene sequence. Finally, the pathway database was randomly split into training (80%) and validation (20%). The training set was used to train the model, whereas the validation set was exclusively used to test the method after the training was done. Importantly, the validation set was never used during the training process. Note that the training set corresponds to amino acid sequences, whereas the validation set consists of nucleotide sequences. To simulate a library of short sequence reads, sequences of 100 bp long were randomly extracted from each nucleotide sequence from the validation data set. A total of 35,751 short reads were generated.

DIAMOND is currently one of the widely used tools for metagenomic analysis. Therefore, to test the performance of MetaMLP, DIAMOND BlastX with the best hit approach was compared with MetaMLP. DIAMOND was run using a sequence alignment identity of 80%, whereas MetaMLP was set with a minimum probability of 0.8. Precision, recall, and F1 score were computed to measure the performance of both approaches.

2.5. False positive data set

To test the ability of MetaMLP to filter out sequences that are not associated with any of the selected pathways (false positives), a synthetic data set was constructed by using the same number of reads from the true positive data set. However, each nucleotide position on this data set was randomly selected. This negative data set was then employed as input to MetaMLP and the best hit approach using DIAMOND with default parameters. Precision, recall, and F1 score were computed to measure the performance of both methods.

2.6. Time and memory profiling

To evaluate the time performance and memory footprint of MetaMLP, a data set of 100k, 1M, 10M, and 100M reads were built by randomly extracting reads from a real metagenomic soil sample of 407,645,066 reads. This sample is under the Sequence Read Archive (SRA) accession number SRR2901746 and corresponds to a 250 bp long read sample from the Illumina HiSeq 2000 sequencer. Along with MetaMLP, DIAMOND was executed using the same data sets as input. Both methods executed with only one enabled CPU in the same Linux 16.4 environment.

2.7. Functional annotation of metagenomic data sets

MetaMLP was used to profile 4 different environments comprising a total of 68 metagenomic samples through the functional composition analysis, including pathway detection, response to stress, and antibiotic resistance composition. Sixty-eight publicly available metagenomes were downloaded from the SRA from the National Center for Biotechnology Information, spanning 4 different environments as follows: 10 soil, 15 human gut, 15 freshwater, and 28 wastewater samples. Results from MetaMLP were compared against the best hit approach using DIAMOND BlastX, with an identity of 80%.

For the GO reference database, MetaMLP was executed with a permissive 0.5 minimum probability to retrieve multiple classifications. Relative abundance results were compared against those obtained using sequence alignment with DIAMOND BlastX at an 80% identity cutoff.

2.8. *Availability of MetaMLP*

Source code for MetaMLP is available at https://bitbucket.org/gaarangoa/metamlp/src/master

3. RESULTS AND DISCUSSION

3.1. *Effect of* k*-mer size*

The *k*-mer size is one of the key parameters for MetaMLP to perform accurate predictions. Therefore, MetaMLP was evaluated using the true positive data set, with a *k*-mer size ranging from 3 to 20 amino acids (Fig. 2). It was observed that for a large *k*-mer size, MetaMLP generates accurate predictions but penalizes the number of predicted reads. For instance, Figure 2 shows that for a *k*-mer size of $k=3$ MetaMLP is able to predict 99% of the reads, but, with a very low performance with a 0.7F1 score. However, when using a *k*-mer size of $k=20$, MetaMLP achieves a 0.99F1 score. However, it was only able to detect 17% of the total number of reads. Interestingly, for a *k*-mer size of $k=11$, it is enough to achieve a 0.99F1 score with a 29% of predicted reads. As shown in Figure 2, performance of MetaMLP does not improve when using a *k*-mer sizes larger than 11. Therefore, a *k*-mer size of $k=11$ was set as default for training the MetaMLP models.

3.2. *Sequence embedding in MetaMLP*

The sequence embedding strategy allows MetaMLP to represent amino acid sequences as numerical vectors (embedding dimension) taking account of the distribution of the *k*-mers in the protein sequence as well as their labels. Thus, MetaMLP uses the supervised embedding implementation from FastText to learn these numerical vectors and minimize the inner distance within members of a class and maximize the outer distance to other classes. For instance, proteins that belong to the beta-lactamase class are expected to cluster together and to remain distant from members of other classes. Figure 3 shows the distribution of the MetaMLP embeddings in a two-dimensional space generated using the t-SNE algorithm (Maaten and Hinton, 2008). For targeted databases such as the ARG categories or pathways database, MetaMLP clustered categories according to their labels with a representative cohesion and separation (silhouette score: 0.56 and 0.62 for pathways and ARGs, respectively) (Fig. 3A, B).

Interestingly, in a complex classification problem represented by the GO database, where proteins contain multiple labels, MetaMLP demonstrated a consistent distribution over the clusters and its corresponding categories. Clusters shown in Figure 3C describe the relationship between different biological processes involved in response to stress. For example, proteins responding to antibiotics are also associated with other biological processes, such as response to toxic substances, pathogenesis, virus defenses, chemotaxis, and response to DNA damage. Such associations can be clearly seen from the visualizations of the embeddings. Therefore, the embedding strategy adopted in MetaMLP is suitable for representing reference databases where proteins contain multiple labels.

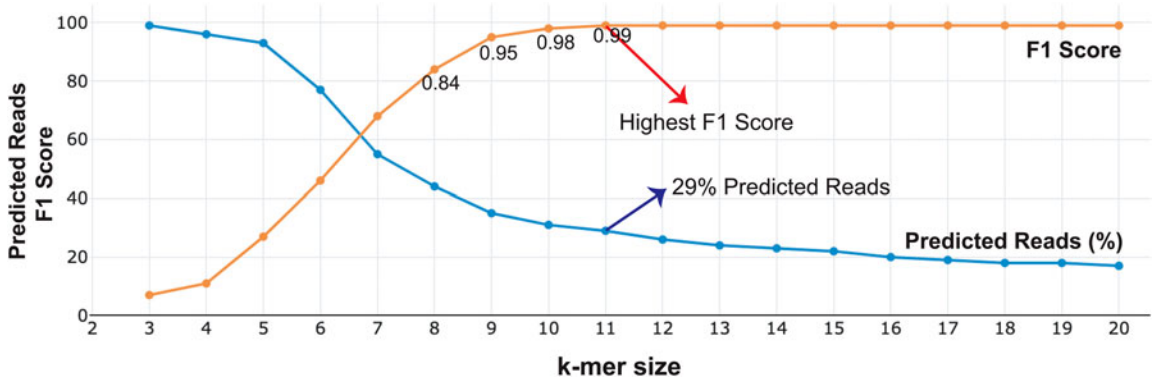

FIG. 2. Performance of MetaMLP with different *k*-mer sizes.

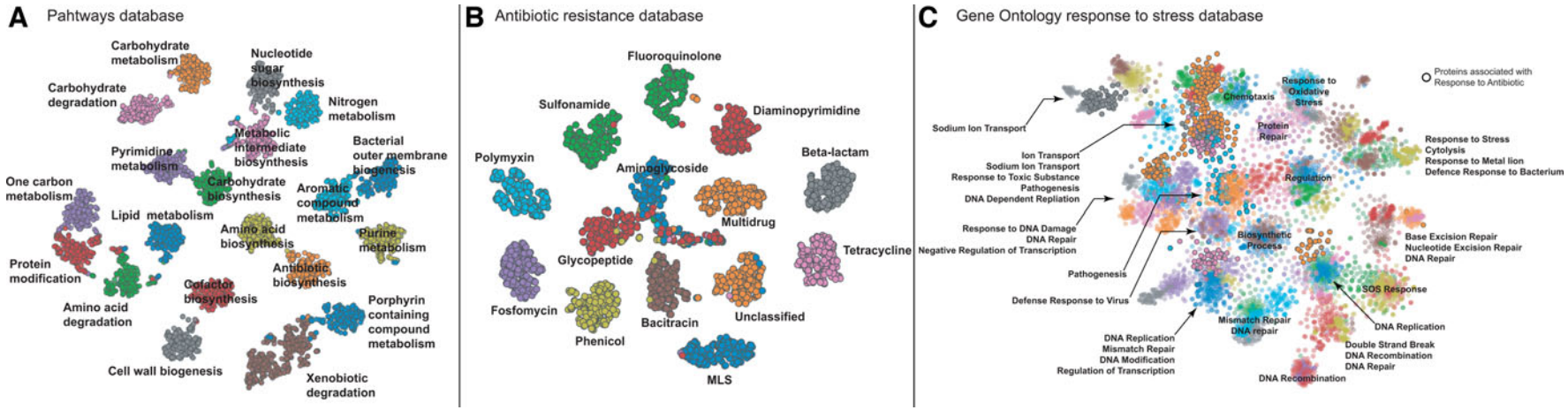

FIG. 3. MetaMLP embeddings representation in two-dimensional space for the pathways (**A**), ARGs (**B**), and GO (**C**) response to stress databases. ARGs, antibiotic resistance genes; GO, Gene Ontology.

3.3. Detection of true positive hits

The pathway reference database was used to assess the ability of MetaMLP to discriminate between pathway-like reads and to evaluate the performance of MetaMLP on classifying short sequences from a particular pathway. To compare the performance of MetaMLP, the best hit approach using DIAMOND BlastX was used. In total, MetaMLP was able to identify 10,433 (29%) pathway-like reads out of the total 35,751, with a probability greater than 0.8, whereas the baseline approach was able to identify 8695 (24%) reads out of the 35,751. This means that MetaMLP was able to identify 5% more of the reads than the best hit approach at 80% identity. Furthermore, both methods were compared on their positive predictions to evaluate their performance for discriminating reads from a particular pathway. As expected, the sequence alignment approach at 80% identity performed with a high average precision (0.99) and recall (1.00) (Supplementary Table S4), whereas MetaMLP was also near to a perfect prediction with a 0.99 average precision and 0.99 average recall (Supplementary Table S5) indicating the potential of the *k*-mer vectors to represent protein sequences to profile metagenomes.

It is also worth mentioning that MetaMLP and the best hit approach did not perform well for three categories (aromatic compound metabolism, bacterial outer membrane biogenesis, and xenobiotic degradation). Interestingly, the best hit approach was not able to identify any reads from bacterial outer membrane biogenesis, whereas MetaMLP obtained a 1.00 precision but a low 0.13 recall, indicating a high sensitivity of MetaMLP in discriminating true positives from this category, but failing for false negatives. In terms of relative abundance, the comparison of the read counts between the best hit approach and MetaMLP was very close with a correlation of 0.988, indicating that MetaMLP correctly characterized the composition of the pathways in the simulated data set (Supplementary Fig. S1).

3.4. Detection of false positive hits

A false positive is a negative sample predicted as positive. For instance, a read that does not belong to any pathway class is predicted to occur in a particular pathway. In this false positive scenario, MetaMLP was tested against the number of predicted random reads by counting how many out of the 35,751 negative reads were classified in any pathways. As a result, MetaMLP classified only 2 reads (0.005%) of the 35,751 negative reads indicating a very low false positive rate. As expected, the best hit approach did not produce any relevant alignment.

3.5. Time and memory usage of MetaMLP

The main advantage for constructing a classifier instead of performing a sequence alignment is the speed improvement when making annotations. Results have shown that MetaMLP maintains an almost identical level of sensitivity compared with Diamond BlastX. However, the advantage of MetaMLP is its speed. Table 1 shows the speed benchmarking over data sets with different numbers of reads. Note that MetaMLP is >50 times faster than DIAMOND for all the sample sizes. MetaMLP produces very similar results in terms of relative abundance using the ARG database and the pathway reference database with correlations of 0.951 and 0.953, respectively (Supplementary Fig. S2). Note that, in this test, MetaMLP identified 35% more ARG-like reads (253,370) compared with the number of reads (186,736) detected with DIAMOND BlastX. In addition, MetaMLP is memory efficient, with memory utilization depending mostly on the size of the reference database. For instance, it requires a minimum RAM memory of 1.0 Gb to run the pathway reference database, 1.2 Gb when using the ARG database, and 2.8 Gb for the GO database. When processing 100M reads, MetaMLP required 1.7 Gb in total with the pathway reference database, whereas DIAMOND BlastX required 6.68 Gb. The low memory utilization in MetaMLP is a consequence of its

Table 1. Time Profiling of MetaMLP Compared with Diamond BlastX Over Different Sample Sizes

No. of reads	*MetaMLP*	*Diamond*
100,000	9 seconds	38 seconds
1,000,000	27 seconds	6 minutes
10,000,000	1 minutes	67 minutes
100,000,000	14 minutes	714 minutes

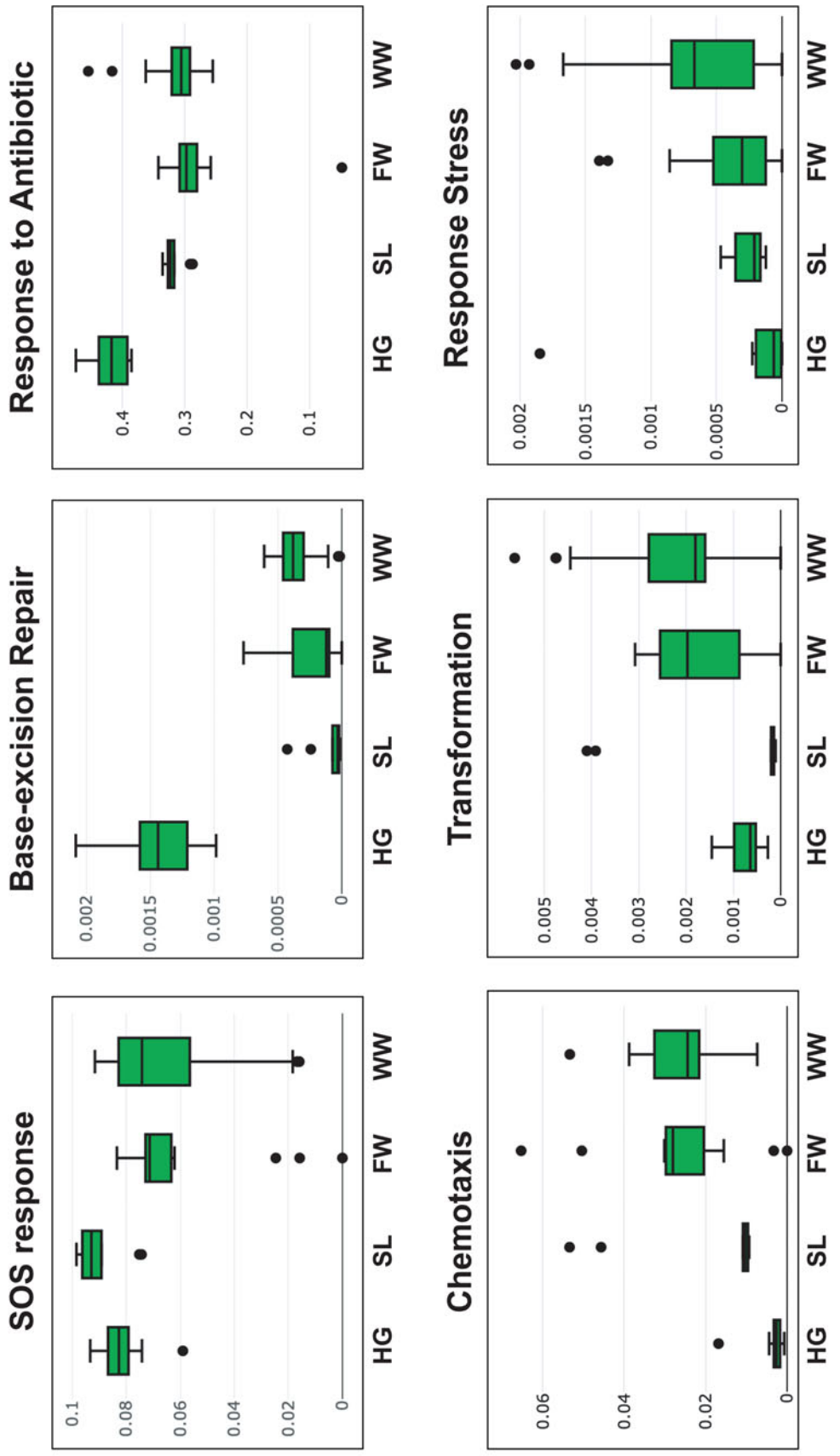

FIG. 4. Relative abundance of biological process from the GO response to stress database. FW, freshwater; HG, human gut; SL, soil; WW, wastewater.

classification strategy, where reads are loaded in small units of 10,000 reads for input efficiency. Therefore, MetaMLP can be executed in a personal computer without the need of using a cluster with a high amount of RAM.

3.6. Functional annotation of different environments

MetaMLP was executed over the 67 real metagenomic samples, processing a total of 2,186,933,071 reads. Of those reads, MetaMLP was able to predict 2,343,026 as ARG-like reads in 710 minutes using only one CPU, whereas DIAMOND BlastX identified 2,003,050 reads in 5256 minutes using 20 CPUs. The average correlation of abundances between DIAMOND and MetaMLP was 0.94 (0.88 log transformed abundance). Interestingly, human gut microbiota and wastewater were the two environments where both methods had the highest correlation with respect to their log transformed abundance (0.96 and 0.93, respectively), whereas soil and freshwater environments each had correlations of 0.83.

3.7. Observation of MetaMLP annotations against an extensive metagenomics study

An extensive study carried out by Pal et al. (2016) uses over 800 metagenomic samples spanning several environments with a sequence alignment strategy at 90% identity cutoff for annotation. This study (named Pal800 for simplicity) demonstrated that the human gut microbiota is one of the environments with the highest relative abundance compared with other microbiomes (soil, wastewater, and freshwater). Also, when MetaMLP was executed on the 68 real metagenomic samples using the GO database, it also profiled the human gut microbiome as the highest relative abundance for the response to antibiotic process (Fig. 4). Note that Pal800 used a curated ARG database, and therefore, it did not consider the induction of false positives. However, the GO database only provides a general overview of the functional composition of those environments. Therefore, a more detailed analysis was obtained by looking at the results from MetaMLP using the specialized ARG database. As a result, the same trend was observed when comparing both analyses (MetaMLP, Pal800).

For example, the tetracycline category had the highest relative abundance in the human microbiome, sulfonamide had the highest relative abundance in the wastewater environment, and the relative abundance of the beta-lactamase class was higher in the freshwater environment compared with the wastewater environment, and both are higher than in the human gut or soil environments. Pal800 also performed a composition profile of the mobile genetic elements present in the microbiomes. It demonstrated that wastewater, freshwater, and soil environments had a higher relative abundance compared with the human gut. Interestingly, for MetaMLP, the GO response to stress database conveyed a similar trend in relative abundance for the biological process ''establishment of competence for transformation'' (see Transformation in Fig. 4). This term is associated with genetic transfer between organisms and is described by the GO consortium as the process whereby exogenous DNA is acquired by a bacterium. In all, despite only using 67 real metagenomes, the functional annotation carried out by MetaMLP described a very similar trend for relative abundances when compared with the Pal800 study, indicating a real usage scenario of MetaMLP.

4. CONCLUSIONS

MetaMLP is an alignment-free method for profiling metagenomic samples to specific target groups of proteins (e.g., ARGs, pathways, and GO terms) using a machine learning classifier. It uses sequence embeddings to represent protein and DNA sequences as numerical vectors and a linear classifier to discriminate between protein functions. Results show that MetaMLP identifies more reads than the widely used best hit approach (sequence alignment with identity >80%) and has as good a performance as the sequence alignment method. Remarkably, MetaMLP is $\sim$50 times faster than the DIAMOND aligner, the most widely used sequence alignment tool for metagenomic data sets. MetaMLP can be trained using any collection of protein sequences (reference database) and maintains a low main memory footprint for the specialized databases used in this article. Finally, MetaMLP is open source and freely available at https://bitbucket.org/gaarangoa/metamlp/src/master

AUTHOR DISCLOSURE STATEMENT

The authors declare they have no conflicting financial interests.

FUNDING INFORMATION

This work was funded in part by USDA NIFA AFRI awards 2014-05280 and 2017-68003-26498, National Science Foundation Partnership in International Research and Education award 1545756, and National Science Foundation award 2004751.

SUPPLEMENTARY MATERIAL

Supplementary Figure S1
Supplementary Figure S2
Supplementary Table S1
Supplementary Table S2
Supplementary Table S3
Supplementary Table S4
Supplementary Table S5

REFERENCES

Altschul, S.F., Gish, W., Miller, W., et al. 1990. Basic local alignment search tool. *J. Mol. Biol.* 215, 403–410.

Altschul, S.F., Madden, T.L., Schäffer, A.A., et al. 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389–3402.

Arango-Argoty, G., Garner, E., Pruden, A., et al. 2018. DeepARG: A deep learning approach for predicting antibiotic resistance genes from metagenomic data. *Microbiome* 6, 23.

Bengtsson-Palme, J., Larsson, D.J., and Kristiansson, E. 2017. Using metagenomics to investigate human and environmental resistomes. *J. Antimicrob. Chemother.* 72, 2690–2703.

Berlin, K., Koren, S., Chin, C.-S., et al. 2015. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nat. Biotechnol.* 33, 623–630.

Buchfink, B., Xie, C., and Huson, D.H. 2015. Fast and sensitive protein alignment using DIAMOND. *Nat. Methods* 12, 59–60.

Edgar, R.C. 2010. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* 26, 2460–2461.

Finn, R.D., Clements, J., and Eddy, S.R. 2011. HMMER web server: Interactive sequence similarity searching. *Nucleic Acids Res.* 39, W29–W37.

Goldberg, Y., and Levy, O. 2014. word2vec Explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722.

Gupta, A., Jordan, I.K., and Rishishwar, L. 2016. stringMLST: A fast k-mer based tool for multilocus sequence typing. *Bioinformatics* 33, 119–121.

Joulin, A., Grave, E., Bojanowski, P., et al. 2016. Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.

Kent, W.J. 2002. BLAT—The BLAST-like alignment tool. *Genome Res.* 12, 656–664.

Li, H. 2016. Minimap and miniasm: Fast mapping and de novo assembly for noisy long sequences. *Bioinformatics* 32, 2103–2110.

Li, H., and Homer, N. 2010. A survey of sequence alignment algorithms for next-generation sequencing. *Brief. Bioinformatics* 11, 473–483.

Li, Y., Heavican, T.B., Vellichirammal, N.N., et al. 2017. ChimeRScope: A novel alignment-free algorithm for fusion transcript prediction using paired-end RNA-Seq data. *Nucleic Acids Res.* 45, gkx315.

Maaten, L.V.D., and Hinton, G. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, 2579–2605.

Ng, P. 2017. dna2vec: Consistent vector representations of variable-length k-mers. arXiv preprint arXiv:1701.06279.

Ondov, B.D., Treangen, T.J., Melsted, P., et al. 2016. Mash: Fast genome and metagenome distance estimation using MinHash. *Genome Biol.* 17, 132.

Ounit, R., Wanamaker, S., Close, T.J., et al. 2015. CLARK: Fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genom.* 16, 236.

Pajuste, F.-D., Kaplinski, L., Möls, M., et al. 2017. FastGT: An alignment-free method for calling common SNVs directly from raw sequencing reads. *Sci. Rep.* 7, 2537.

Pal, C., Bengtsson-Palme, J., Kristiansson, E., et al. 2016. The structure and diversity of human, animal and environmental resistomes. *Microbiome* 4, 54.

Patro, R., Duggal, G., Love, M.I., et al. 2017. Salmon provides fast and bias-aware quantification of transcript expression. *Nat. Methods* 14, 417–419.

Patro, R., Mount, S.M., and Kingsford, C. 2014. Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nat. Biotechnol.* 32, 462.

Pearson, W.R. 1990. Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods in Enzymol*. 183, 63–98.

Pearson, W.R. 1991. Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. *Genomics* 11, 635–650.

Pearson, W.R. 2013. An introduction to sequence similarity (''homology'') searching. *Curr. Protoc.* Bioinformatics Chapter 3, Unit3.1.

Pierce, N.T., Irber, L., Reiter, T., et al. 2019. Large-scale sequence comparisons with sourmash. F1000Res 8, 1006.

Vinga, S, and Almeida, J. 2003. Alignment-free sequence comparison—A review. *Bioinformatics* 19, 513–523.

Weijers, S., De Jonge, J., Van Zanten, O., et al. 2012. KALLISTO: Cost effective and integrated optimization of the urban wastewater system Eindhoven. *Water Pract. Tech.* 7, wpt2012036.

Wood, D.E., and Salzberg, S.L. 2014. Kraken: Ultrafast metagenomic sequence classification using exact alignments. *Genome Biol.* 15, R46.

Yang, K.K., Wu, Z., Bedbrook, C.N., et al. 2018. Learned protein embeddings for machine learning. *Bioinformatics* 34, 2642–2648.

Ye, Y., Choi, J.-H., and Tang, H. 2011. RAPSearch: A fast protein similarity search tool for short reads. *BMC Bioinformatics* 12, 159.

Zhang, Z., and Wang, W. 2014. RNA-Skim: A rapid method for RNA-Seq quantification at transcript level. *Bioinformatics* 30, i283–i292.

Zhao, Y., Tang, H., and Ye, Y. 2011. RAPSearch2: A fast and memory-efficient protein similarity search tool for next-generation sequencing data. *Bioinformatics* 28, 125–126.

Zielezinski, A., Vinga, S., Almeida, J., et al. 2017. Alignment-free sequence comparison: Benefits, applications, and tools. *Genome Biol.* 18, 186.

Address correspondence to:
Prof. Liqing Zhang
Department of Computer Science
Virginia Tech
Blacksburg, VA 24060
USA

E-mail: lqzhang@vt.edu