所以是用CNN来分类OTU构造的树.

Using Convolutional Neural Networks to Explore the Microbiome

Derek Reiman[1*], Ahmed Metwally[2*], and Yang Dai[3§]

Abstract— **The microbiome has been shown to have an impact on the development of various diseases in the host. Being able to make an accurate prediction of the phenotype of a genomic sample based on its microbial taxonomic abundance profile is an important problem for personalized medicine. In this paper, we examine the potential of using a deep learning framework, a convolutional neural network (CNN), for such a prediction. To facilitate the CNN learning, we explore the structure of abundance profiles by creating the phylogenetic tree and by designing a scheme to embed the tree to a matrix that retains the spatial relationship of nodes in the tree and their quantitative characteristics. The proposed CNN framework is highly accurate, achieving a 99.47% of accuracy based on the evaluation on a dataset 1967 samples of three phenotypes. Our result demonstrated the feasibility and promising aspect of CNN in the classification of sample phenotype.**

I. Introduction

The microbiome of the gut has been linked to many diseases including, but not limited to, inflammatory bowel disease, irritable bowel disease, diabetes, obesity, cancer, autoimmune diseases, and various metabolic disorders [1]. A microbiome study is usually started with the characterization of the microbial community in an environmental/genomic sample using next generation sequencing (NGS) technologies. The output from the analysis is the abundance of taxa at one of the taxonomic levels (Super-kingdom, Phylum, Class, Order, Family, Genus, and Species), depending on whether the amplicon sequencing on 16S rRNA genes or the MetaGenome Shotgun (MGS) Sequencing is used for analyzing DNA from the sample [2, 3]. The next step is to detect statistical associations between microbial taxa and phenotype. Alternative approaches based on predictive models have been proposed using Random Forest Classifier (RFC) and Support Vector Machine (SVM) [4]. The difficulty in establishing these prediction models is the selection of features relative to the phenotypical response from a large number of microbial taxa.

Recently, multiple deep learning frameworks have been applied to the microbiome phenotype prediction [5], including Recursive Neural Network (RNN) and Deep Belief Network (BDN), in additional to the traditional Multi-Layer Perceptron Neural Network (MLPNN). These methods create hierarchical layers of abstract complex features from simple features. For example, the RNN model relies on representing samples by the hierarchical tree constructed based on the similarity between the taxa abundance profiles of the samples. Their results revealed that the RNN classifier is the only one being able to produce a hierarchical relationship of the samples that can be visualized as a tree. However, its classification accuracy is not as satisfactory compared with MLPNN and other models such as support vector machines and random forest.

In this work, we propose a novel framework using Convolutional Neural Network (CNN), which is a deep learning model based on the visual cortex and is more commonly used in image processing and speech recognition. A CNN incorporates spatial information into the model and generates convolution layers with multiple feature maps. It uses a kernel of shared weights to traverses the input for constructing combinations of local features. A convolutional layer is usually followed by a nonlinear activation function and a pooling method. This inherently reduces the feature space and performs feature selection. Since microbial taxonomic abundance profiles imply structure information, we take advantage of the CNN modeling approach to explore this structure by constructing a phylogenetic tree. This tree is further populated with the observed microbial abundance of taxa for individual samples and then is embedded in a 2D matrix which preserves most of the spatial relationship between the nodes in the tree. These matrices are input to our CNN model. Our computational study shows the strong performance of the CNN and its ability to handle large datasets.

II. Models and Methods

In this section, we first describe how to represent the microbial taxonomic abundance data obtained from a sample on a phylogenetic tree as well as how to embed the tree into a matrix to preserve the spatial information of the phylogenetic tree. Our procedure is shown in Fig. 1.

A. The Phylogenetic Tree

In our paper, we used the OTUs data generated from the 16S rRNA sequencing technology to describe our approach. It can be readily applied to microbial taxonomic abundance data produced from the MGS sequencing as well. Determining phylogenetic relationships can be done by comparing the microbial genomes. In most cases, the

[1]Derek Reiman is with the Bioinformatics Program of the Department of Bioengineering, University of Illinois at Chicago, Chicago, IL 60612, USA. e-mail: dreima2@uic.edu .

[2]Ahmed Metwally is with the Bioinformatics Program of the Department of Bioengineering, University of Illinois at Chicago, Chicago, IL 60612, USA. e-mail: ametwa2@uic.edu.

[3]Yang Dai is with the Faculty of the Department of Bioengineering, University of Illinois at Chicago, Chicago, IL 60612, USA. e-mail: yangdai@uic.edu .

*Authors contributed equally.
§Corresponding author.

978-1-5090-2809-2/17/$31.00 ©2017 IEEE

sequences of 16S rRNA genes are compared based on multiple sequence alignments and a similarity matrix can be constructed based on the distance between the sequences. This can be extrapolated even further into the construction of a tree in which similar species are organized into clades and are closer together. The construction of this tree is not the focus of this paper. In this work, we used the online tools, PhyloT [6] and iTol [7], to create and visualize the phylogenetic tree.

A diagram of an example of the basic components of a phylogenetic tree is shown in Fig.1 (the second panel). The leaf nodes in the tree represent individual OTUs. A parent node represents a common ancestor for its children. The set of children for an ancestral node is called a clad. The nodes are used to mark a distance of one.

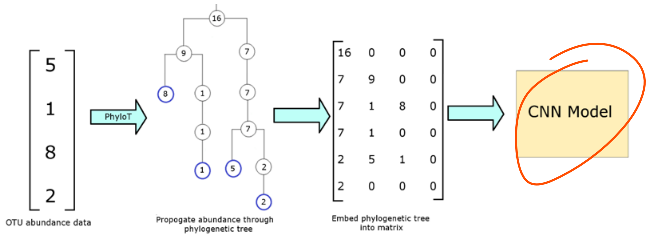

Figure 1. A flowchart of the proposed procedure for a hypothetical OTU count vector of a sample.

B. *Populating the Phylogenetic Tree*

With the phylogenetic tree constructed using the OTUs, the next step is to assign abundance to the nodes in the tree in a way that the tree contains denser meaningful data on ancestral nodes as well maintains similarity distances both horizontally and vertically. More specifically, the tree is populated by assigning the abundance of an ancestral node to be the sum of its children's abundances. Any node without a child node is considered a leaf node and its abundance is taken from the input vector by matching the feature label to the node label in the phylogenetic tree.

C. *Matrix Construction*

With a populated tree that contains abundance data (nodes) as well as phylogenetic similarity (topology), the next challenge is embedding the tree structure into a matrix format so that it can be used as an input for a CNN model. The goal here is to be able to represent the OTU count data in a two-dimensional way such that the spatial information from the tree is represented by the location of the data in the matrix. This embedding allows CNNs to exploit the spatial information presented in the input data. There are multiple ways to represent this tree in a matrix format, which may provide future research developments in this area. We formulate this problem as follows.

Starting from the top, the matrix was constructed by following from the left to the right of the populated phylogenetic tree. The root would occupy the top left corner of the matrix. Each row down in the matrix was filled by taking the set of the child nodes in the order appeared in the tree from left to right. The rest is padded with zeroes. Fig. 2 shows the embedding process on a diagram, while the formal algorithm is described in Algorithm 1, and the symbols used in the algorithm are defined in Table I.

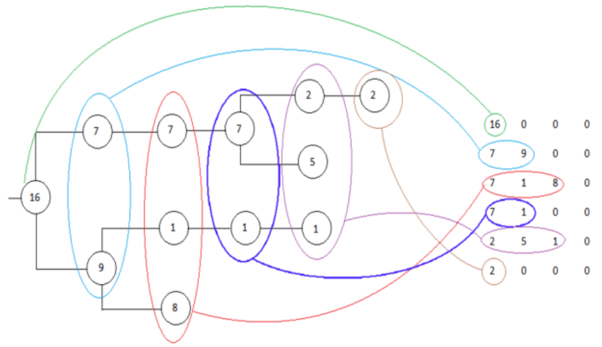

Figure 2. Tree embedding in which we structure the tree in a matrix format. The root(s) are placed in the top left corner.

Algorithm 1 Tree Embedding

Input: A populated phylogenetic tree $\boldsymbol{G} = (\boldsymbol{N}, \boldsymbol{E})$

Output: A matrix $\boldsymbol{M}$ which preserves both the abundance data and the spatial information of the tree.

```
Construct a zero matrix $\boldsymbol{M}$ with $L$ rows and $|\boldsymbol{S}|$ columns
Node List ← Root Node of the tree $\boldsymbol{G}$
For $j$ from 0 to $L$ do
    $k$ ← 0
    Next List ← {}
    For each node $n$ in Node List do
        $\boldsymbol{M}_{k,j}$ ← $a_n$
        Push $\boldsymbol{C}_n$ onto Next List
        $k$ ← $k + 1$
    End For
    Node List ← Next List
End For
Return $\boldsymbol{M}$
```

.

TABLE I. SYMBOLS AND DEFINITIONS USED IN ALGORITHM 1

Symbol	**Definition**
$\boldsymbol{S}$	Input vector of microbial taxa abundance (OTUs)
s	Number of taxa in input vector
$\boldsymbol{G} = (\boldsymbol{N}, \boldsymbol{E})$	A phylogenetic tree
$\boldsymbol{N}$	Taxa/clade nodes in phylogenetic tree
$\boldsymbol{E}$	Edges linking taxa/clades in phylogenetic tree
L	Total layers in phylogenetic tree
$\boldsymbol{C}_n$	The set of children of node n
a_n	The abundance of node n
$\boldsymbol{M}$	An L by s matrix created form the phylogenetic tree
j, k	Row and column index of matrix $\boldsymbol{M}$

D. *The CNN Architecture*

Our CNN model has three convolutional layers. The tangent hyperbolic activation function and max pooling subsampling were used in each layer. The numbers of feature maps were 20, 40, and 60 respective to the layers. Then we used a fully connected layer of 100 neurons with the tangent hyperbolic activation function, and lastly a softmax layer with three output neurons (skin, gut, and oral cavity). The desired outputs from the ground truth were constructed as a 1 for that output neuron and zeroes in the other two.

The activation function of the output layer is the softmax function. To make the softmax function work, one must first calculate the velocities of the previous layer:

$$v_a = \Sigma_b \, w^L_{a,b} \, y^{L-1}_b + b^L_a \tag{2}$$

where v_a represents the velocity of the a^{th} output neuron, w^L represents the weights between layers L and $L-1$, y is the output of the previous layer, and $b_a{}^L$ is the bias of the a^{th} output neuron. The softmax function then gives us:

$$y_a = \frac{e^{v_a}}{\Sigma_b e^{v_b}} \tag{3}$$

The softmax output is used to develop a probability distribution over all three outputs. The output with the highest probability is determined as the winner and that is how the site is classified. The network is trained using the log-likelihood cost function as follows:

$$C \equiv -\ln y_a \tag{4}$$

With the gradients:

$$\frac{\partial C}{\partial b^L_a} = y^L_a - d_a \tag{5}$$

$$\frac{\partial C}{\partial w^L_{a,b}} = y^{L-1}_b \, (y^L_a - d_a) \tag{6}$$

Here d_a represents the vector output for the a^{th} output neuron. L2 regularization with a penalty term was added to give a total cost of:

$$C = -\ln y_a + \lambda \parallel W \parallel^2 \tag{7}$$

The dropout method was implemented in addition to L2 normalization in order to further prevent overfitting. The dropout method works by randomly selecting nodes within the fully connected hidden layers and temporarily removing them from the training. This prevents any feed-forward information from that node as well as any feedback information from the back-propagation algorithm. In each iteration of dropout, the network is trained using a subnet, allowing the full network to be able to make accurate predictions using multiple pathways. Each epoch the network was trained using the training set, and then the validation set is tested. If the accuracy of the validation set is equal to or better than the best reported validation accuracy, the test set is then tested and the accuracy is reported.

E. *Dataset*

The dataset used in this experiment comes from Caporaso *et al.* [8], which were downloaded from the Qiita database (http://qiita.microbio.me). In this dataset, microbial samples were taken from a man and a woman over a period of 15 months and 6 months, respectively. There were roughly an equal number of samples from the fecal matter, oral cavity, left hand, and right hand. There is a total of 1967 samples with 4300 OTUs. Table II shows a breakdown of the sample classes based on both the host as well as the sites of the samples.

In the previous studies using this dataset [5], rare OTUs were dropped, resulting in 1762 OUTs. When constructing the phylogenetic tree, some "vague" OTUs that were not recognized by PhyloT were discarded. The removal of these OTUs resulted in a total of 1706 OTUs in the phylogenetic tree. Following the procedure described in the previous section, the matrix constructed for each example for CNN input was 29 x 1706. Our tree had 29 layers due to the fact that it contained various subgroups between taxonomic ranks. It is important to note that no further feature selection was considered in our experiment since a CNN performs feature selection intrinsically during training.

TABLE II. CLASS BREAKDOWN OF THE DATASET

	Male	**Female**	**Total**
Fecal Matter	330	137	467
Oral Cavity	376	132	508
Left Palm	368	131	499
Right Palm	359	134	493
Total	1433	534	1967

F. *Experimental Procedure*

The CNN model was implemented using the Theano Python library [9]. In preprocessing the data, we only attempted to classify the site location for this experiment. We did not take into account the host gender. However, in looking at the data, it was obvious that the left hand and right hand had very similar profiles. Therefore, we grouped these two classes into a single class. By doing that, our goal is to classify sites from skin, gut, and oral cavity sites. For each class, the data set was shuffled and the first 80% was used in the training set. The next 10% was used in the validation set, and the final 10% was used in the test set. Since CNNs require a larger amount of inputs to train, additional training points were generated as follows. The same amount of data points from each class were sampled at random with replacement from the training partition, each time adding a small amount of noise to the original data. The process was repeated to generate the validation and test sets as well using their respective partitions, resulting in 2400 training points and 300 points in both the validation and test sets. The CNN was trained using minibatch gradient descent with λ of 0.1 and a learning rate of 0.005, and the holdout method was applied for validation. The dropout rate was 0.5 and the numbers of the feature maps at the three convolution layers are 20, 40 and 60 respectively.

III. RESULTS

The CNN used in this work was implemented using the Theano Python Library [9]. In our experiment, we compared the proposed CNN model using the phylogenetic tree to matrix method (CNN-2D) with a simple one-dimensional application using a sliding window kernel (CNN-1D). In addition, we also compared the CNN approach with other deep learning frameworks including RNN, DBN, a traditional MLPNN and RFC. These methods were evaluated on the same datasets [5]. We report the performance of the CNNs together with the results reported in that paper for other methods in Table III.

The CNN-2D model achieved an accuracy of 99.47%, which is better than that of the CNN-1D (94.67%). This indicates the effectiveness of our approach in representing structure relationship of the microbial taxa abundance vectors. Among the other methods, the MLPNN with three hidden layers of 500 neurons performs the best. The MLPNN is closely followed by the RFC, which has been shown to be able to perform well in small metagenomic data.

TABLE III. ACCURACY OF CNN AND OTHER PREVIOUSLY STUDIED METHODS (THE NUMBERS IN PARENTHESIS IS THE NUMBER OF NEURONS)

Method	Accuracy (%)
RNN (250)	83.13
RNN (500)	84.19
DBN (250)	96.85
DBN (500)	97.21
MLPNN (500)	99.44
RFC	99.14
CNN-1D	94.67
CNN-2D	99.47

IV. DISCUSSION

Deep learning methods have been shown to be promising on large and complex datasets by finding a hierarchy of abstract features. CNNs are particularly successful in applications of speech and imaging recognition where the features are spatially structured in input. They try to capture data based on three key ideas: local connectivity, spatial invariance, and transitional invariance. CNNs have also been applied to other bioinformatic applications. For example, Quan *et al.* developed a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequence [10]. The convolution layers have captured various regulatory motifs, demonstrating the capability of CNNs in detecting discriminative features. Kelly *et al.* developed another CNN framework to predict the change of accessibility of genome between variant alleles [11]. The spatial relationship in the DNA sequences could be attributed to the success in both applications.

The use of microbial taxa abundance profiles in the CNN learning framework presents a challenge. Identifying the structure between taxa and preserving the spatial relationship are key to the effectiveness of the CNN. In our work, the phylogenetic tree populated with the microbial abundance data is a natural choice for the structure and is biologically meaningful. Our tree embedding scheme can be thought of using a rectangular filter sliding through the phylogenetic tree to observe multiple clades as well as a combination of clades and children of other clades. The created matrix contains spatial information between nodes based on their locations in the tree. This idea was inspired by the work of Mou *et al.* [12] in which a tree based CNN was designed for applications in program language processing. Their method uses convolution over vectors, selecting all children and the parent in the kernel, and three-way pooling for subsampling.

There are several challenges involved in using CNNs for microbiome phenotype prediction. Our CNN's performance benefits from the spatial information captured with the proposed scheme of embedding, however, there may be a misalignment of parents and children. Exploring other methods of the matrix representation of the phylogenetic tree may allow the CNN to find more relevant patterns for optimized performance. Finally, how to extract feature maps predictive to phenotype from convolution layers is yet to be solved.

V. CONCLUSION

We developed a CNN model for classification of a microbiome sample based on its microbial taxonomic abundance profile. The proposed CNN model exploits the topological structure of the phylogenetic tree constructed from the abundance data. This approach can be readily applicable to the microbial from the whole genome shotgun sequencing study. We believe that the performance of a CNN model on metagenomic data can be further improved. The future direction is the development of an effective way to extract and interpret the predictive feature maps learned from the CNN in order to reveal the biological relevance to the host phenotype.

REFERENCES

[1] J. R. Marchesi, D. H. Adams, F. Fava, G. D. A. Hermes, G. M. Hirschfield, G. Hold*, et al.*, "The gut microbiota and host health: a new clinical frontier," *Gut,* vol. 65, pp. 330-339, 2016.

[2] A. A. Metwally, Y. Dai, P. W. Finn, and D. L. Perkins, "WEVOTE: Weighted Voting Taxonomic Identification Method of Microbial Sequences," *PLOS ONE,* vol. 11, p. e0163527, 2016.

[3] T. Thomas, J. Gilbert, and F. Meyer, "Metagenomics - a guide from sampling to data analysis," *Microbial Informatics and Experimentation,* vol. 2, pp. 3-3, 2012.

[4] Q. Zhang, H. Abel, A. Wells, P. Lenzini, F. Gomez, M. A. Province*, et al.*, "Selection of models for the analysis of risk-factor trees: leveraging biological knowledge to mine large sets of risk factors with application to microbiome data," *Bioinformatics,* vol. 31, pp. 1607-1613, 2015.

[5] G. Ditzler, R. Polikar, and G. Rosen, "Multi-Layer and Recursive Neural Networks for Metagenomic Classification," *IEEE Transactions on NanoBioscience,* vol. 14, pp. 608-616, 2015.

[6] "PhyloT: a Tree Generator. ," http://phylot.biobyte.de/.

[7] I. Letunic and P. Bork, "Interactive tree of life (iTOL) v3: an online tool for the display and annotation of phylogenetic and other trees," *Nucleic Acids Research,* vol. 44, pp. W242-W245, 2016.

[8] J. G. Caporaso, C. L. Lauber, E. K. Costello, D. Berg-Lyons, A. Gonzalez, J. Stombaugh*, et al.*, "Moving pictures of the human microbiome," *Genome Biology,* vol. 12, p. R50, 2011// 2011.

[9] T. T. D. Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv:1605.02688,* 2016.

[10] D. Quang and X. Xie, "DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences," *Nucleic Acids Research,* vol. 44, pp. e107-e107, 2016.

[11] D. R. Kelley, J. Snoek, and J. Rinn, "Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks," *Genome Research,* 2016.

[12] L. Mou, G. Li, L. Zhang, and T. Wang, "TBCNN: A Tree-Based Convolutional Neural Network for Programming Language Processing," *arxiv.org/pdf/1409.5718v1,* 2014.