Statistics in Biosciences
https://doi.org/10.1007/s12561-024-09434-9

ORIGINAL PAPER

DeepBiome: A Phylogenetic Tree Informed Deep Neural Network for Microbiome Data Analysis

Jing Zhai[1] · Youngwon Choi[2,3] · Xingyi Yang[1] · Yin Chen[4] · Kenneth Knox[5] · Homer L. Twigg III[6] · Joong-Ho Won[2] · Hua Zhou[7] · Jin J. Zhou[7,8,9]

Received: 20 November 2022 / Revised: 28 November 2023 / Accepted: 20 February 2024

Abstract

Evidence linking the microbiome to human health is rapidly growing. The microbiome profile has the potential as a novel predictive biomarker for many diseases. However, tables of bacterial counts are typically sparse, and bacteria are classified within a hierarchy of taxonomic levels, ranging from species to phylum. Existing tools focus on identifying microbiome associations at either the community level or a specific, pre-defined taxonomic level. Incorporating the evolutionary relationship between bacteria can enhance data interpretation. This approach allows for aggregating microbiome contributions, leading to more accurate and interpretable results. We present `DeepBiome`, a phylogeny-informed neural network architecture, to predict phenotypes from microbiome counts and uncover the microbiome–phenotype association network. It utilizes microbiome abundance as input and employs phylogenetic taxonomy to guide the neural network's architecture. Leveraging phylogenetic information, `DeepBiome` reduces the need for extensive tuning of the deep learning architecture, minimizes overfitting, and, crucially, enables the visualization of the path from microbiome counts to disease. It is applicable to both regression and classification problems. Simulation studies and real-life data analysis have shown that `DeepBiome` is both highly accurate and efficient. It offers deep insights into complex microbiome–phenotype associations, even with small to moderate training sample sizes. In practice, the specific taxonomic level at which microbiome clusters tag the association remains unknown. Therefore, the main advantage of the presented method over other analytical methods is that it offers an ecological and evolutionary understanding of host–microbe interactions, which is important for microbiome-based medicine. `DeepBiome` is implemented using Python packages `Keras` and `TensorFlow`. It is an open-source tool available at https://github.com/Young-won/DeepBiome.

Jing Zhai and Youngwon Choi contribute equally to this study.

Extended author information available on the last page of the article

Published online: 14 June 2024

Keywords Metagenomics · Phylogenetic tree · Neural networks · Prediction · Mixed taxonomic levels

Abbreviations

DeepBiome:	A phylogenetic tree informed deep neural network for microbiome data analysis
RNA:	Ribosomal ribonucleic acid
OTUs:	Operational Taxonomic Units
HIV:	Human immunodeficiency virus
DNN:	Deep neural network
BMI:	Body mass index
CNN:	Convolutional neural network
AGP:	American gut project
SVM:	Support vector machine
Adam:	An adaptive gradient algorithm
MSE:	Mean square error
AUC:	Area under the receiver operating characteristics
TPR:	True positive rate
ACC:	Accuracy
PPV:	Precision
TP:	True positive (recall)
TN:	True negative
FP:	False positive
FN:	False negative
T2D:	Type 2 diabetes
COPD:	Chronic obstructive pulmonary disease
FEV1:	Forced expiratory volume in 1 s
SPT:	Supraglottic predominant taxa
BPT:	Background predominant taxa

1 Introduction

Emerging high-throughput sequencing technologies have vastly improved our understanding of the human microbiome's role in many diseases [10, 12, 21, 26]. Despite these achievements, our knowledge of the mechanisms behind microbes' involvement in disease aggravation remains limited. A notable challenge is inferring and visualizing the path from bacteria to disease across various taxonomic levels.

In 16S ribosomal RNA (RNA) sequencing, sequences are clustered into operational taxonomic units (OTUs) at a threshold of 97% sequence similarity [4]. Sequence representatives, those with the fewest mismatches compared to others in a cluster, are used for taxonomic assignment. These assignments are based on databases of known 16S rRNA gene sequences, such as GreenGenes (des), the Ribosomal Database Project [8], and Silva [24]. Consequently, the processed microbiome

data comprise an OTU abundance matrix with rows as samples and columns as OTUs, accompanied by a phylogenetic tree. This tree is crucial for understanding the relationships among OTUs and has been utilized in statistical models to better identify microbiome elements associated with host phenotypes [33, 35, 36]. Closely related microbial taxa often form clusters with similar biological functions, but determining the specific taxonomic level at which association signals are clustered remains a challenge. Are these signals concentrated at a shallow phylogenetic depth (e.g., genus), at a deeper level (e.g., phylum), or across various taxonomic ranks?

Several methods have been proposed to integrate the phylogenetic structure into analyses. Chen et al. [5, 6] introduced a Laplacian penalty, constructed from the sum of branch lengths linking any two OTUs on the evolutionary tree, but their methods are not designed to detect signals at different evolutionary depths. Garcia Tanya et al. [11] proposed a sparse regression model using ℓ_1 and ℓ_2 regularizations to achieve sparsity at multiple taxonomic levels, yet this method requires three tuning parameters and an exhaustive grid search. Furthermore, it can only select taxa at up to three levels, failing to cover the entire range of phylogenetic depths. Wang and Zhao [31] utilized a tree-guided variable fusion method to build predictive models using bacteria at different taxonomic levels based on the assumption that closely related bacteria have similar biological functions. However, Zhou et al. [38] provided a counterexample; in their study, *Corynebacterium* and *Rothia*, belonging to the same order *Actinomycetales*, had opposite effects on lung function changes in an HIV positive population. Xiao et al. [34] developed a generalized linear mixed model using the evolutionary rate as a tuning parameter, capable of identifying clustered signals without prior knowledge of the phylogenetic depth. However, this method is less effective when microbiome effects are clustered at mixed taxonomic levels.

Deep Neural Networks (DNN), an area of growing interest in biomedical research, show promise for analyzing complex microbiome data. Lu et al. [18]'s study applied a DNN model to the gut microbiome, identifying important bacteria associated with body mass index (BMI). Another study by Reiman et al. [25] incorporated phylogenetic data into a Convolutional Neural Network (CNN) architecture to predict various outcomes. They translated taxa abundances at each phylogenetic level into an abundance matrix, capturing the spatial information of taxa in the phylogenetic tree. However, their model, limited to a fixed number of layers (three), resulted in the loss of taxa lineage information and produced neurons lacking biological meaning, thus making the model difficult to interpret. Other deep learning methods, like MDeep [32], comprising multiple convolutional layers followed by fully connected layers, and DeepMicro [22], utilizing various autoencoders, also struggle with interpretability.

We present `DeepBiome`, a DNN-based predictive model designed to capture microbiome signals at different phylogenetic depths, applicable to both regression and classification problems. It processes microbiome taxonomic abundance data as input and regularizes the neural network architecture according to the phylogenetic

structure. This significantly reduces the number of parameters and the tuning burden compared to conventional neural networks, enabling the identification of important taxa associated with outcomes at all taxonomic levels. Phylogeny regularization in `DeepBiome` is achieved through weight decay, a widely used technique to prevent overfitting and enhance neural network performances [13, 16, 20, 37]. Unlike existing weight decay schemes that assume a global rate of decay, `DeepBiome` incorporates the bacteria's evolutionary relationship into a differential weight decay regularization matrix, thereby generating an interpretable effect transfer network for modeling and analyzing microbiome data. Simulation studies and analyses of datasets from a shotgun metagenomic study and a lung microbiome study demonstrate `DeepBiome`'s superior performance over commonly used tools such as support vector machines (SVM), regression with ℓ_1 (Lasso) or $\ell_1 + \ell_2$ (Elastic-Net) penalties, DNN without tree regularization, DNN with ℓ_1 penalty, and Random Forests.

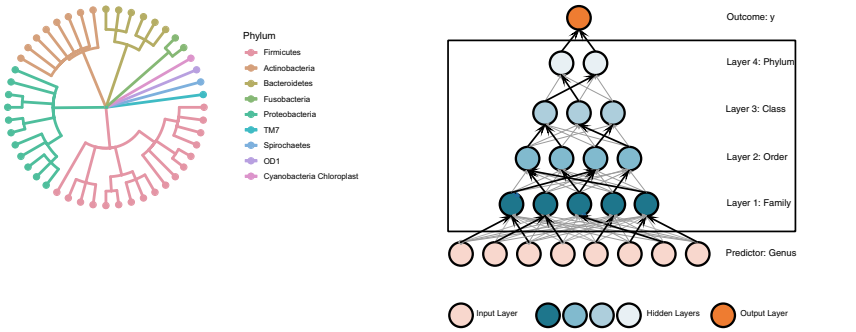
2 Methods and Materials

2.1 Microbiome Data Structure

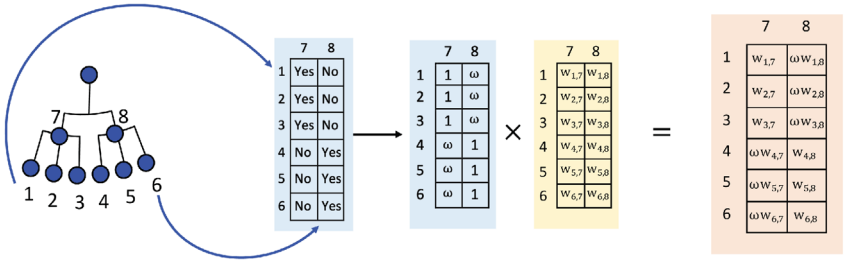
Suppose we have p OTUs from a total of n microbiome samples, along with a phylogenetic tree that outlines the evolutionary relationships among these microbes. In this tree, each OTU is represented as a tip node, while every internal node corresponds to a taxonomic unit, signifying a common ancestor of its descendant taxa. In our study, we focus on consolidating these p OTUs into m genus-level taxa, treating them as our primary units for analysis. Our analysis could also begin at finer taxonomic levels. An illustrative phylogenetic tree, which we refer to throughout our methods and data simulation discussions, is depicted in Fig. 1. Here, let $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ denote the input data, with $\boldsymbol{y} = (y_1, \ldots, y_n)$ representing the targeted outcomes. For each subject i, $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{im})^T$ indicates the abundance of m genera. The outcome variable $\boldsymbol{y}$ can be continuous, binary, or categorical.

2.2 Architecture of `DeepBiome`

`DeepBiome` is a neural network architecture that associates input vectors $\boldsymbol{x}$ (representing microbiome abundance) with a clinical outcome $\boldsymbol{y}$. A major challenge in neural network construction is deciding on the optimal number of layers and neurons. The conventional wisdom advocates for going deep (many layers) and wide (many neurons per layer), a strategy that has achieved great success in various artificial intelligence tasks. These include image pattern recognition and natural language processing. However, this approach requires a significant amount of training data [3, 28], which is often impractical in biomedical studies due to resource constraints. `DeepBiome` pre-specifies the network architecture according to the phylogenetic tree. The number of hidden layers is determined by the number of taxonomic levels, while the number of neurons



(a) An example phylogenetic tree with 48 genera (b) Network layout of `DeepBiome` architecture



(c) Phylogenetic tree regularized weight decay procedure

Fig. 1 **`DeepBiome`** architecture. **a** A phylogenetic tree with 48 genera as tip nodes. Color represents phylum types. **b** Network layout of `DeepBiome` architecture. The input layer is genus level microbiome abundance. Each hidden layer represents one phylogenetic level, e.g., family, order, class, and phylum. The dark lines represent relationship defined by a phylogenetics tree. The gray lines represent association between layers. **c** Phylogenetic tree regularized weight decay. Suppose we have a simple tree as shown in the left panel, which has 6 genera (taxa 1–6) and 2 classes (7–8). Genera 1–3 belong to class 7 and genera 4–6 belong to class 8. The ancestor–descendent information is embedded into a 6×2 matrix. Without loss of generality, we use ω to indicate a regularization factor with small value (e.g., 0.01). For tree regularized weight decay, the weight estimation matrix $\boldsymbol{w}_{6\times2}$ is multiplied with this phylogenetic embedded matrix $\boldsymbol{\Omega}_{6\times2}$ elementwisely, denoted by $\boldsymbol{\Omega}_{6\times2} \circ \boldsymbol{w}_{6\times2}$ (Color figure online)

in each layer corresponds to the number of taxa at that particular level. Figure 1b illustrates an example `DeepBiome` architecture. The input layer accepts microbiome abundances. The information is then propagated through multiple layers of the `DeepBiome` network to the outcome of interest $\boldsymbol{y}$. For example, the input vector $\boldsymbol{x}$, representing the abundances of $m^{(0)}$ genera, is propagated to the first hidden layer vector $\boldsymbol{z}^{(1)}$. This layer contains a total of $m^{(1)}$ neurons, corresponding to the number of family-level taxa. Using an $m^{(1)} \times m^{(0)}$ weight matrix $\boldsymbol{w}^{(1)}$ and an $m^{(1)}$ bias vector $\boldsymbol{b}^{(1)}$, we have

$$\boldsymbol{z}^{(1)} = v(\boldsymbol{w}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}). \tag{1}$$

Each weight parameter $w_{jk}^{(1)}$ represents the effect of the kth input unit on the jth hidden neuron, and $v(\cdot)$ is an activation function. Throughout this paper, we use the rectified linear unit (ReLU) activations [14, 19]

$$\text{ReLu: } v(a) = a^+ = \max(0, a), \tag{2}$$

but it can be easily changed to other activation functions in our software. Similarly, $\boldsymbol{z}^{(\ell)} = v(\boldsymbol{w}^{(\ell)}\boldsymbol{z}^{(\ell)} + \boldsymbol{b}^{(\ell)}), \quad \ell = 2, \ldots, L$, where L is the total number of hidden layers in the neural network. The last hidden layer $\boldsymbol{z}^{(L)}$ is linked to the outcome using either an identity link or a softmax link. Specifically, we use identity link to predict a continuous outcome, $\boldsymbol{y} = \boldsymbol{w}^{(L+1)}\boldsymbol{z}^{(L)} + \boldsymbol{b}$. For categorical outcomes with K categories, softmax function is adopted to predict the probability of ith subject belonging to c-th category,

$$\Pr(y_i = c) = \frac{e^{(\boldsymbol{w}\boldsymbol{z}^{(L)}+\boldsymbol{b})_c}}{\sum_{q=1}^{K} e^{(\boldsymbol{w}\boldsymbol{z}^{(L)}+\boldsymbol{b})_q}}. \tag{3}$$

Finally, we use $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ with parameters $\boldsymbol{\theta} = \{\boldsymbol{w}, \boldsymbol{b}\}$ to represent the whole neural network that maps an input $\boldsymbol{x}$ to an output $\boldsymbol{y}$, where $\boldsymbol{w} = (\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(L)}, \boldsymbol{w}^{(L+1)})$ and $\boldsymbol{b} = (\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(L)}, \boldsymbol{b}^{(L+1)})$.

2.3 Phylogeny Regularization via Weight Decay

We introduce phylogeny regularization through weight decay. We assume that if taxa j and k have an ancestor–descendent relationship, the associations between the corresponding neurons are stronger, i.e., larger weight value w_{jk}. When taxa j and k do not have this ancestral relationship, we assume $w_{j,k}$ to be a small value, i.e., weight decay. Thus, we construct a weight decay matrix $\boldsymbol{\omega}$ to regularize weights in the neural network using the evolutionary relationship carried by the phylogenetic tree. If nodes j and k are ancestor–descendent related, $\omega_{jk} = 1$; if not, ω_{jk} is a small value, e.g., 0.01. See Fig. 1c as an illustration.

2.4 Model Training

Given a training set consisting of training pairs $\{\boldsymbol{x}, \boldsymbol{y}\}$ and a neural network $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ with parameters $\boldsymbol{\theta}$, a supervised training procedure is implemented to learn neural network function by minimizing the empirical loss. We define loss to be Mean Squared Error (MSE) for continuous outcomes and standard Cross-Entropy (CE) for categorical outcomes:

$$\begin{aligned} \text{MSE} &= \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i)^2, \\ \text{CE} &= -\frac{1}{n}\sum_{i=1}^{n}\sum_{q=1}^{K} y_{i,q}\log\hat{y}_{i,q}, \end{aligned} \tag{4}$$

where $y_{i,k}$ is a binary indicator (0 or 1) indicating whether observation i belongs to class k (i.e., one hot encoding). $y_{i,c} = \Pr(y_i = c)$ ($c \in \{1, \ldots, K\}$) is the probability that observation i belongs to class c defined by Eq. (3). We use the holdout validation method to determine the stopping criterion during training. The training process stops when the holdout validation error achieves the minimum.

Algorithm 1 Phylogeny regularized weight decay in `Adam`. β_1, β_2 refer to the exponential decay rates for the moment estimates in Adam. $\epsilon = 10^{-8}$ is used to prevent division from zero error [14].

Data: $\boldsymbol{y}$, $\boldsymbol{x}_i = (x_1, \cdots, x_m)$, phylogenetic tree designed matrix ω, learning rate $\alpha \in \mathcal{R}$

Result: $\hat{\boldsymbol{\theta}}$ such that the lost function Loss($\boldsymbol{\theta}$) is minimized.

Initialize: parameter $(\boldsymbol{w},\ \boldsymbol{b})_{t=0} \in \mathcal{R}$, 1st and 2nd moment vector $m^1_{t=0} \leftarrow \mathbf{0}$ and $m^2_{t=0} \leftarrow \mathbf{0}$, and step index $t \leftarrow 0$,

repeat

- $t \leftarrow t + 1$
- $l'_t \leftarrow \nabla_\theta \text{Loss}(\omega \circ \boldsymbol{w}_{t-1}, \boldsymbol{b}_{t-1})$ (Get gradients w.r.t. stochastic objective at t)
- $m^1_t \leftarrow \beta_1 m^1_{t-1} + (1 - \beta_1) l'_t$ (Update biased 1st moment estimate)
- $m^2_t \leftarrow \beta_2 m^2_{t-1} + (1 - \beta_2) l''_t$ (Update biased 2nd moment estimate)
- $\hat{m}^1_t \leftarrow m^1_t / (1 - \beta^t_1)$ (Update bias-corrected 1st moment estimate)
- $\hat{m}^2_t \leftarrow m^2_t / (1 - \beta^t_2)$ (Update bias-corrected 2nd moment estimate)
- $(\boldsymbol{w},\ \boldsymbol{b})_t \leftarrow (\boldsymbol{w},\ \boldsymbol{b})_{t-1} - \alpha \cdot \hat{m}^1_t / (\sqrt{\hat{m}^2_t} + \epsilon)$ (Update parameters)

until *stopping criterion is met and return* $\hat{\boldsymbol{\theta}} = (\omega \circ \hat{\boldsymbol{w}}, \hat{\boldsymbol{b}})$;

`Adam` optimizer, an adaptive gradient algorithm [15], is used to train `DeepBiome`. Algorithm 1 describes the parameter estimation in `Adam` with the proposed phylogeny regularization. Specifically, at each update t, the estimated weight w_t is elementwisely multiplied by the corresponding regularization factor ω and $\omega \circ w_t$ is used to predict $\hat{y}$ in the loss function [see Eq. (4)]. Once the stopping criteria is met, Algorithm 1 outputs the estimated parameters of each layer $(\hat{\boldsymbol{w}}, \hat{\boldsymbol{b}}) = (\{\omega^{(1)} \circ \hat{\boldsymbol{w}}^{(1)}, \cdots, \omega^{(L)} \circ \hat{\boldsymbol{w}}^{(L)}\}, \{\hat{\boldsymbol{b}}^{(1)}, \cdots, \hat{\boldsymbol{b}}^{(L)}\})$. This phylogeny regularization effectively uses biologically meaningful prior knowledge to limit the number of free parameters in the model. Therefore, it avoids overfitting.

2.5 Performance Metrics

We employ several statistical metrics to evaluate the performance of `DeepBiome` for its prediction, classification and taxa selection performances. For a quantitative outcome, the primary metric is the mean square error (MSE). Additionally, we report the Pearson correlation coefficient ρ between predicted $\hat{y}_i$ and true y_i. For categorical outcomes, i.e., classification problems, we measure their performance using sensitivity [true positive rate (TPR)], specificity, g-measure, accuracy (ACC), precision (PPV), and the F1 score:

$$\begin{aligned}
\text{Sensitivity} &= \frac{\text{TP}}{\text{TP+FN}},\\
\text{Specificity} &= \frac{\text{TN}}{\text{TN+FP}},\\
g\text{-Measure} &= (\text{Sensitivity} \times \text{Specificity})^{\frac{1}{2}},\\
\text{ACC} &= \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}},\\
\text{PPV} &= \frac{\text{TP}}{\text{TP+FP}},\\
\text{F1 score} &= 2 \times \frac{\text{PPV} \times \text{TPR}}{\text{PPV+TPR}} = \frac{2\text{TP}}{2\text{TP+FP+FN}},
\end{aligned}$$

where TP is "true positive" (or recall), TN is "true negative", FP is "false positive", and FN is "false negative". The F1 score is the harmonic mean of precision and sensitivity. An F1 score reaches its best value at one when the prediction has perfect precision and recall and the worst at zero. Note that the F1 score does not take the true negative into account. We use the g-measure, the geometric mean of sensitivity and specificity, to assess the performance of a binary classifier. Same as an F1 score, a g-Measure reaches its best value at one when the sensitivity and specificity are both perfect (one) while the worst at zero if any of the sensitivity and specificity is zero. We also report AUC (area under the receiver operating characteristics), which reports the capability of a model to distinguish between classes. Sensitivity, specificity, g-Measure, and ACC across all hidden layers (see Table 1) are used to report the selection accuracies.

3 Results

3.1 Simulation Studies

We conduct extensive simulation studies to evaluate the performance of `DeepBiome` and compare it with conventional methods in three different schemes, i.e., linear regression, binary, and multiclass ($K \geq 3$) classification design. Throughout the simulation experiments, we use sample size $n = 1000$ and split them into a training set (75%, $n_{\text{training}} = 750$) and a test set (25%, $n_{\text{test}} = 250$). Different proportions of the split give qualitatively similar results (not shown). All results are obtained based on 1000 replicates. Simulation scenario 1 covers continuous outcome models; simulation scenario 2 is for binary outcome cases, and simulation scenario 3 considers the situation when the outcome variable is categorical. Model robustness is evaluated in simulation scenario 4 when tree structure is misspecified and microbiome abundances contain measurement errors.

Detailed procedures to generate microbiome abundance data have been described before by our group [35, 36]. We use a Dirichlet Multinomial (DM) distribution with the mean proportion vector and the dispersion parameter estimated from a real pulmonary microbiome dataset to generate OTU counts. We

then aggregate 2964 OTUs to 48 genus according to the phylogenetic tree [30]. A forward propagation approach is described below to generate $\boldsymbol{y}$.

(1) Read the phylogenetic tree to obtain the number of phylogenetic levels and nodes at each level. The microbiome data are then summarized at genus, family, order, class, and phylum level as shown in Fig. 1. Based on a real lung microbiome dataset, the number of nodes are $m^{(0)} = 48$, $m^{(1)} = 40$, $m^{(2)} = 23$, $m^{(3)} = 17$, and $m^{(4)} = 9$, respectively.

(2) Construct the weight matrix $\boldsymbol{w}^{(1)} \in \mathbb{R}^{m^{(1)} \times m^{(0)}}$ to propagate the input layer $\boldsymbol{x}_{\text{genus}}$ to the 1st hidden layer by

$$\boldsymbol{h}^{(1)} = \boldsymbol{w}^{(1)} \boldsymbol{x}_{\text{genus}} + \boldsymbol{b}^{(1)}.$$

The bias vector $\boldsymbol{b}^{(1)} \in \mathbb{R}^{m^{(1)} \times 1}$ follows a standard normal distribution $\mathcal{N}(0, \sigma_e^2)$ with $\sigma_e^2 = 4$. Suppose we have node j at the genus level and k at the family level, then

$$w_{j,k}^{(1)} \sim \begin{cases} \text{Uniform}(-0.5, 1) & \text{associated with output,} \\ \mathcal{N}(0, 0.01) & \text{not associated with output.} \end{cases} \tag{5}$$

(3) Multiply the $w_{j,k}^{(1)}$ by a small value $\omega_{j,k} = 0.01$, if family taxa k is not a direct ancestor of genus taxa j; otherwise, $w_{j,k}^{(1)}$ stays the same.

(4) Activate the neurons using ReLU in Eq. 2,

$$\boldsymbol{x}_{\text{family}} = \boldsymbol{z}^{(1)} = v(\boldsymbol{w}^{(1)} \boldsymbol{x}_{\text{genus}} + b^{(1)}).$$

(5) Repeating steps 2–4 to compute the $\boldsymbol{x}_{\text{order}}$, $\boldsymbol{x}_{\text{class}}$, and $\boldsymbol{x}_{\text{phylum}}$.

(6) Simulate the continuous or categorical output layer $\boldsymbol{y}$ as follow

$$\begin{aligned} \hat{\boldsymbol{y}} &= \boldsymbol{w}^{(4)} \boldsymbol{x}_{\text{phylum}} + \boldsymbol{b}^{(4)} \\ \hat{\Pr}(y_i = c) &= \frac{(e^{\boldsymbol{w}^{(4)} \boldsymbol{x}_{\text{phylum}} + \boldsymbol{b}^{(4)}})_c}{\sum_{q=1}^{K} (e^{\boldsymbol{w}^{(4)} \boldsymbol{x}_{\text{phylum}} + \boldsymbol{b}^{(4)}})_q}, \end{aligned} \tag{6}$$

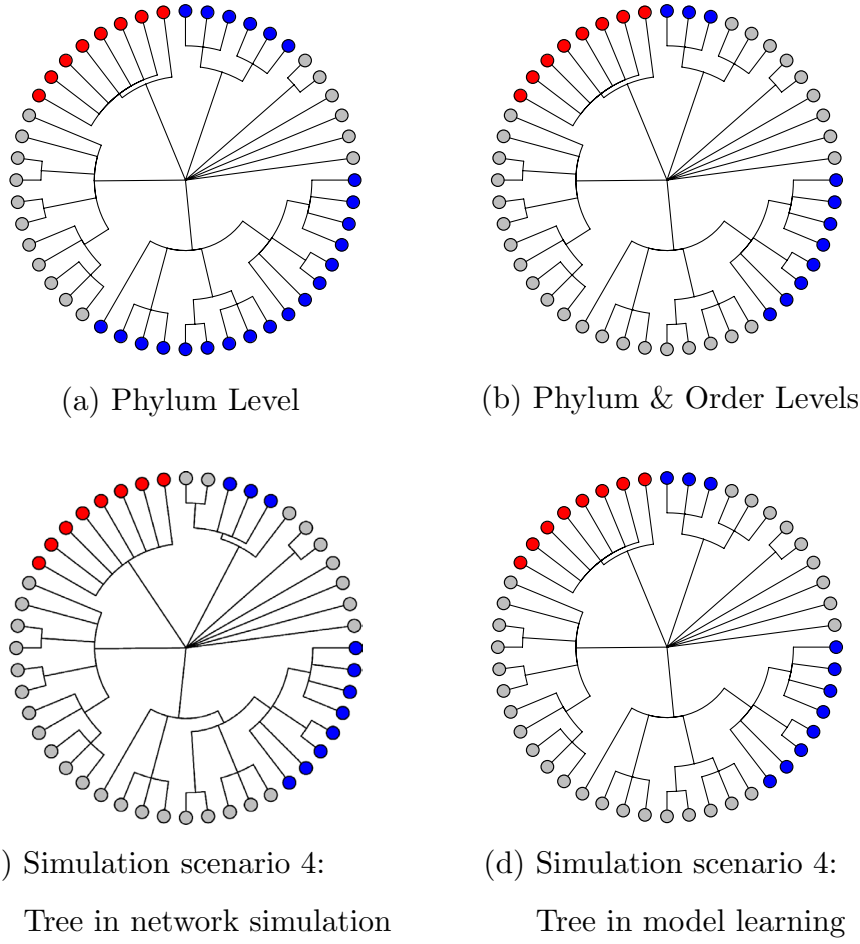
where $K = 2$ for binary classification and $K \geq 3$ for multi-categorical classification.

The following simulation schemes are considered to examine the robustness of `DeepBiome` method.

(1) Abundance contains measurement errors at genus levels. We assume that 10% of the associated genus reads are misclassified to one randomly selected genus from the same phylum. The microbiome abundance data with measurement errors are then used for training models.

(2) The phylogenetic tree for training models is misspecified (see Fig. 2).

- At class level, the genera that belong to *Clostridia* and *Flavobacteria* are mis-classified to *Bacilli* and *Bacteroidia*.



(a) Phylum Level

(b) Phylum & Order Levels

(c) Simulation scenario 4: Tree in network simulation

(d) Simulation scenario 4: Tree in model learning

Fig. 2 Simulation specifications. Outcome associated taxa (blue and red) are specified at **a** the phylum level and **b** a mixture of phylum and order levels. The blue nodes represent "bad" taxa which result in disease status or are negatively associated with continuous phenotype, e.g., FEV1. The red nodes represent "good" taxa which result in a healthy status. In simulation scenario 4, we evaluate the impact of the misspecified phylogenetic tree: **c** indicates the true phylogenetic tree used in simulation scenario 4 and **d** indicates the phylogenetic tree used in model learning (same as the tree shown in **b**) (Color figure online)

- At order level, the genera that belong to *Coriobacteriales* and *Flavobacteriales* are mis-classified to *Actinomycetales* and *Bacteroidales*.

We use fivefold cross-validation to choose the tuning parameters for regularized linear regression models. R package `randomForest` is adopted for Random Forest analysis [17]. Fivefold cross-validation is used to obtain hyper-parameters in `randomForest` [i.e., the number of features selected at each split (mtry) and the number of trees generated in a forest (ntree)] so that out of bag (OOB) error is minimized. The values of mtry and ntree vary across datasets, with a mean (standard deviation) to be 800 (250) and 36 (30), respectively. The

predictors that result in a mean decrease in accuracy or MSE are selected. On average, 55% of the predictors are selected. The variance importance metric is defined as the decrease in MSE or accuracy when the variable is randomly permuted. For the deep learning models (i.e., DNN, DNN + ℓ_1, and `DeepBiome`), we use a holdout validation set including 20% of the training data and train the models until either validation error increases or 5000 epochs are reached, whichever comes the first. Using `Adam` [15] optimizer, we set $\beta_1 = 0.9, \beta_2 = 0.999$, learning rate $lr = 0.01$ and the mini-batch gradient descent with a mini-batch size of 50. The learning rate decays for each epoch with $lr_{\text{epoch}+1} = lr_{\text{epoch}} \frac{1}{1+0.0001}$.

3.1.1 Scenario 1: Regression Design

Table 2 displays the prediction performance for continuous outcomes when the associated taxa are only clustered at one phylogenetic level (i.e., phylum). `DeepBiome` has a higher Pearson correlation and lower MSE on the test set than the regression methods. It also performs the best among all deep learning models. Table 3 shows the prediction performances under a more complex case, where the associated taxa are clustered at different phylogenetic levels (i.e., phylum and order). All regression schemes perform poorly in this case with low correlation, e.g., 0.6. `DeepBiome` has over 80% reduction in MSE compared to regression-based methods. The deep learning models, DNN and DNN + ℓ_1, improve correlation to 0.91 and 0.9, respectively. However, both show a hint of overfitting with lower testing performance. Compared with Random Forest, `DeepBiome` has slightly higher MSE (0.07 vs. 0.04 shown in Table 2, 0.42 vs. 0.30 shown in Table 3) and slightly lower correlation (0.88 vs. 0.94 shown in Table 2 and 0.92 vs. 0.95 shown in Table 2).

Identifying associated taxa at precise levels is critical. We evaluate the selection performance shown in Fig. 3. Regular regression methods do not discriminate associated taxa; therefore, only the results of penalized regressions are shown. Lasso and Elastic-Net can only select the taxa at one phylogenetic level in the penalized regression schemes. We compute the performance metrics for higher-level taxa selected using their phylogeny relationship. For example, if genus *Prevotella* is selected, we assume that its ancestor, phylum *Bacteroidetes*, is also selected. In contrast, the selection performance of regularized neural network models is based on the weights estimated at each hidden layer. g-Measure for `DeepBiome` ranges from 0.8 to 0.9 across different taxa levels (Fig. 3, the first and second row). `DeepBiome` also has excellent sensitivity, specificity, and ACC. Although the Random Forest method performs better in predicting continuous outcomes, its g-measures range from 0.56 to 0.6, indicating poor selection ability. DNN + ℓ_1 fails to identify the true microbiome taxa across all phylogenetic levels with much lower g-measure, e.g., 0.18 for genus level taxa selection (Fig. 3, the first and second row). Overall, for continuous outcome predictions, Random Forest performs slightly better than `DeepBiome`. Yet `DeepBiome` offers better selection ability.

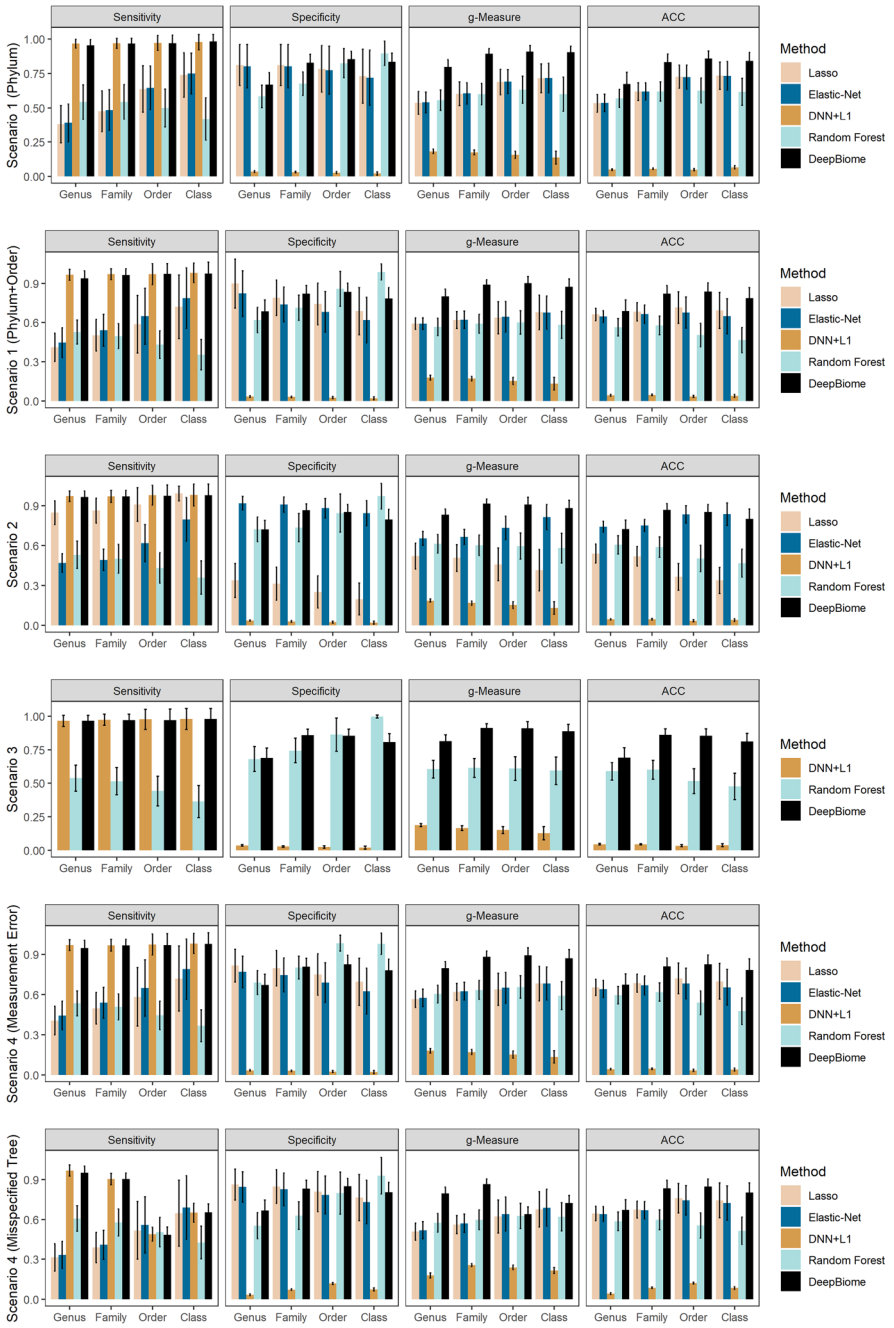


Fig. 3 Taxa selection performance under four simulation schemes at each phylogenetic level. Sensitivity, specificity, g-Measure, and accuracy (ACC) were used to evaluate taxa selection performance. The vertical bar represents the standard deviation over 1000 simulation replicates

3.1.2 Scenario 2: Binary Classification

For simplicity and for demonstrating DeepBiome's ability to discriminate different levels' taxa, we only consider the case that outcome-associated taxa are clustered at mixed phylogenetic levels for the following simulation scenarios (as shown in Fig. 2b). We assume

[(1)] the higher the abundance of blue node taxa, the higher the probability of y belonging to the disease group;

[(2)] the higher the abundance of red node taxa, the higher the probability of y belonging to the healthy control group.

We compare DeepBiome to logistic regression, three penalized logistic regression models, two conventional deep learning networks, and the Random Forest method. In Table 4, we present the metrics for evaluating the classification performance of binary outcomes, including sensitivity, specificity, g-Measure, ACC, and AUC. Logistic regression has satisfying sensitivity, but other metrics are not competitive compared to DeepBiome. Logistic regressions tend to have more false positives. In contrast, DeepBiome achieves the best classification performance with the highest specificity, g-Measure, ACC, and AUC as, 0.84, 0.87, 0.89, and 0.94, respectively. Interestingly, for binary outcome prediction, DeepBiome outperforms the Random Forest method with higher g-measure (0.87 vs. 0.84) and AUC (0.94 vs. 0.85) (see Table 4). Figure 3 (the second row) displays the performance of identifying associated taxa. However, Lasso and DNN + ℓ_1 show good sensitivity at some phylogenetic levels, and g-Measure and ACC are much worse than Elastic-Net and DeepBiome. This suggests that Lasso and DNN + ℓ_1 selected many null taxa (false positive). Using the order level as an example, the g-Measure value of DeepBiome is 0.91, while the DNN + ℓ_1 is 0.15. The selection performance comparison between DeepBiome and Random Forest indicates that DeepBiome also has superior performance, i.e., g-measure of Random Forest vs. DeepBiome was 0.61 vs. 0.83 for genus level taxa selection. It is worth noting that logistic regression with Elastic-Net penalization also offers acceptable selection accuracy, e.g., 0.65 (genus level) vs. 0.81 (class level) (Fig. 3, the third row) (Table 5).

3.1.3 Scenario 3: Multiclass Classification

We evaluate the performance of DeepBiome for multi-category outcomes. We assume, as shown in Fig. 2b, that the blue node taxa lead to "severe" disease, red ones lead to "mild" disease, and gray node taxa are neutral.

We compare DeepBiome to DNN, DNN + ℓ_1, the support vector machine (SVM) with different kernels, and the Random Forest method. For SVM, linear and non-linear kernels, such as the radial or polynomial kernels, are evaluated. The default parameter setting is adopted for SVM. Among the SVMs, the linear SVM has the highest accuracy and AUC, while the SVM with radial kernel yields better recall and F1 score. However, all SVMs are inferior to deep learning models. DeepBiome exhibits the highest AUC, i.e., 0.9; AUCs of DNN and DNN + ℓ_1 are around 0.86. The F1 score of DeepBiome is 0.711, which is 14% higher than the second best, DNN + ℓ_1. We find that DeepBiome offers the best and

most balanced performance with precision and recall, which are 0.72 and 0.71, respectively. Consistent with binary outcome comparisons, DeepBiome also outperforms the Random Forest method for multi-categorical outcome prediction. Since SVM models cannot select the microbiome taxa, we only compare DeepBiome to DNN + ℓ_1 and Random Forest shown in Fig. 3 (the 4th row). DeepBiome surpasses DNN + ℓ_1 in all of the evaluation metrics at all phylogenetic levels. For instance, the g-measure of DNN + ℓ_1 in selecting genus level taxa is only 0.19 while that of DeepBiome is 0.82. Note that across three different outcome types, the selection ability of the Random Forest method remains similar with g-measure ranging from 0.5 to 0.7.

3.1.4 Scenario 4: Robustness Under Tree Misspecification and Measurement Errors of Microbiome Abundance

Table 6 and Fig. 3 (the 5th row) present the results when microbiome abundances contain sequencing errors. Table 7 and Fig. 3 (the 6th row) show the results when using a misspecified phylogenetic tree to train and test the model. Like scenario 1, we simulate continuous outcomes and compare DeepBiome with linear regression, penalized regressions, conventional DNN, ℓ_1-regularized DNN, and Random Forest. When the model is trained using data with measurement errors (case 1), the performance of all methods decreases compared with scenario 1 using data without errors (Table 6; see also Table 3). For example, for the Random Forest method, the MSE is 0.04 (Table 2) without sequencing error vs. 0.42 with sequencing error (Table 6). For DeepBiome, MSE rises from 0.07 (without sequencing error) to 0.24 (with sequencing error). These results indicate that although Random Forest offers better predictive ability when the outcome is continuous, it is more sensitive to sample contamination than DeepBiome. The average Pearson's ρ of DeepBiome is 0.95, while those of DNN, DNN + ℓ_1, and Random Forest are 0.87, 0.91, and 0.92 respectively.

Table 7 displays the prediction performance under case 2 (i.e., using a misspecified phylogenetic tree to train models). DeepBiome outperforms other methods in both MSE and Pearson's ρ except Random Forest. MSE of Random Forest is 0.04 (without misspecified tree) vs. 0.19 (with misspecified tree) while DeepBiome is 0.07 (without misspecified tree) vs. 0.32 (with misspecified tree). Since DeepBiome relies on a polygenetic tree for regularization, the impact of using a misspecified tree to DeepBiome is larger than a Random Forest.

Figure 3 (the 5th and the 6th row) show the ability to identify associated microbiome taxa. When the abundance data contain measurement errors, the sensitivity decreases in penalized regression and deep learning methods. For example, the specificity of DeepBiome at the genus level is 0.67 (compared to 0.95 using correct tree information), leading to a decreased g-measure, i.e., from 0.84 to 0.80. DeepBiome tends to select less associated taxa when input abundance data contain measurement errors. The Lasso, Elastic-Net, and DNN + ℓ_1 perform similarly to scenario 1. Even if DeepBiome uses a wrong tree structure to guide the model, it still has decent performance with a g-Measure of 0.80 at the finest level (i.e., genus).

3.2 Disease Prediction Using Shotgun Metagenomics

The resolution of the shotgun metagenomics data can reach species and strain levels, providing in-depth information to quantify the association between microbiota and human health. As the cost of shotgun metagenomics sequencing keeps decreasing, the number of available human metagenomics datasets keeps increasing. Using eight large-scale publicly available metagenomic datasets, Reference [23] benchmarked statistical learning tools for disease classification. Publicly available software and uniformly processed microbiome profiles (http://segatalab.cibio.unitn.it/tools/metaml) were also provided. For all eight datasets, species-level taxonomic profiling and relative abundances data were processed using `MetaPhlAn2` [29], and the detailed sequence processing procedures were reported [23].

We apply `DeepBiome` to a type 2 diabetes (T2D) cohort among one of the eight studies with the largest sample size. It includes the species-level relative abundances from 170 Chinese T2D patients and 174 controls. We use relative abundances of 572 species-level taxa (i.e., 210 genus-level taxa) and the corresponding phylogenetic tree to predict T2D status and select associated microbiome clusters. Table 8 shows the performance of T2D prediction based on five-fold cross-validation. Although Ridge and Lasso regressions have the highest specificity, their low sensitivity suggests that these methods tend to predict all subjects as healthy. `DeepBiome` performs the best among all methods with the highest g-Measure, accuracy, and AUC, which are 0.620, 0.643, and 0.694, respectively. Consistent with our simulation results, `DeepBiome` shows better predictive power than the Random Forest method. For example, AUC is 0.61 for the Random Forest method and 0.69 for `DeepBiome`. Figure 4 demonstrates the taxa selected by `DeepBiome`. We have selected 86 species, 32 genera, 14 families, 7 orders, and 4 classes. Among these taxa, 32 species, 20 genera, 8 families, 3 orders, and 3 classes are positively associated with T2D, indicating that the higher the abundance of those taxa, the higher the probability of subjects having T2D.

3.3 Computational Efficiency

`DeepBiome` is implemented in Python 3.6 based the TensorFlow [1, 2] and Keras [7] framework. It can be built on Python 3.4, 3.5, and 3.6. All simulations are performed using a workstation equipped with Intel(R) Xeon(R) CPU E5-2650 v4 processor with 24 cores @ 2.20 GHz and one NVIDIA GeForce GTX TITAN X GPU with 3072 CUDA cores @ 1 GHz and 12 GB memory. `DeepBiome` requires 290 ± 69 s to fully train the network for one replicate with 1000 samples, 50 mini-batches, and 5000 epochs. For the same data, DNN takes 282 ± 67 s, and DNN + ℓ_1 takes 282 ± 67 s. `DeepBiome` and all other deep learning approaches take less than 0.004 s for prediction. All real data analysis is performed on a MacBook Pro with 2.8 GHz Intel Core i7 processor and 16 GB 2133 MHz LPDDR3 memory.

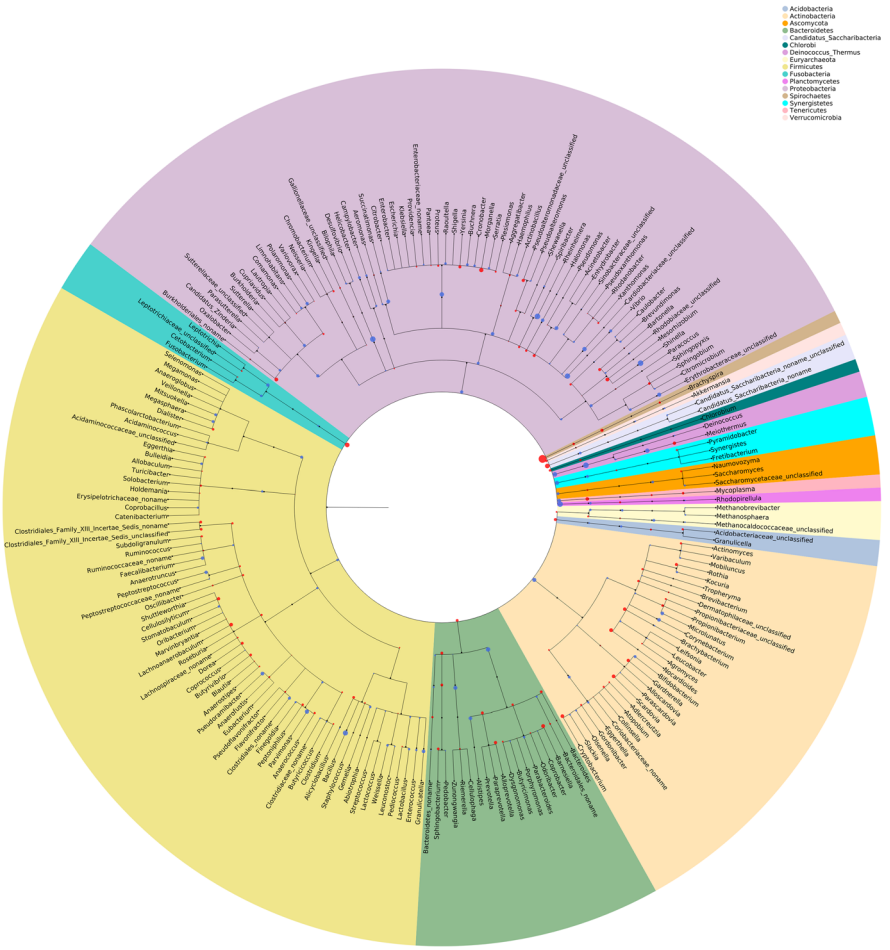


Fig. 4 `DeepBiome` selected T2D-associated taxa using data from a real metagnomic sequencing study. Estimated weights were overlaid on the phylogenetic tree. The red and blue nodes indicate taxa have positive and negative weights, respectively. The size of colored nodes represents the magnitudes of the weights. Black nodes represent non-selected taxa (Color figure online)

Table 1 Metrics used to assess the performance of outcome prediction and microbiome taxa selection

	Prediction			Selection
	Regression	Binary	Multiclass	
Metrics	MSE	Sensitivity	Sensitivity	Sensitivity
	Pearson's ρ	Specificity	PPV	Specificity
		g-Measure	F1 score	g-Measure
		ACC	ACC	ACC
		AUC	AUC	

MSE mean squared error, *PPV* positive predictive value, *ACC* accuracy, *AUC* area under the receiver operating characteristic (ROC) curve

Table 2 Scenario 1: mean squared error (MSE) and Pearson correlation coefficient between predicted and true outcomes for continuous outcome

Method	Testing				Training			
	MSE		Correlation		MSE		Correlation	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Linear Regression	0.104	0.024	0.824	0.049	0.087	0.011	0.851	0.023
Ridge	0.104	0.022	0.824	0.049	0.09	0.012	0.851	0.023
Lasso	0.100	0.023	0.833	0.048	0.092	0.013	0.843	0.025
Elastic-Net	0.100	0.023	0.833	0.048	0.092	0.012	0.844	0.025
DNN	0.076	0.040	0.874	0.077	0.032	0.034	0.947	0.067
DNN + ℓ_1	0.075	0.040	0.875	0.073	0.034	0.039	0.945	0.068
Random Forest	0.044	0.017	0.936	0.032	0.044	0.008	0.943	0.016
DeepBiome	0.071	0.036	0.882	0.069	0.043	0.034	0.929	0.061

The associated taxa are clustered at the phylum level

DNN deep neural network, *DNN* + ℓ_1 Lasso (least absolute shrinkage and selection operator) penalized deep neural network

Table 3 Scenario 1: mean squared error (MSE) and Pearson correlation coefficient between predicted and true outcomes for continuous outcome

Method	Testing				Training			
	MSE		Correlation		MSE		Correlation	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Linear Regression	1.561	0.146	0.639	0.035	1.337	0.068	0.694	0.018
Lasso	1.479	0.115	0.662	0.034	1.411	0.075	0.678	0.020
Ridge	1.546	0.121	0.639	0.034	1.361	0.075	0.694	0.018
Elastic-Net	1.481	0.117	0.662	0.034	1.405	0.076	0.680	0.020
DNN	0.457	0.522	0.905	0.118	0.164	0.337	0.964	0.091
DNN + ℓ_1	0.456	0.516	0.904	0.122	0.176	0.362	0.963	0.085
Random Forest	0.296	0.050	0.949	0.011	0.300	0.028	0.948	0.006
DeepBiome	0.423	1.474	0.916	0.139	0.256	0.463	0.944	0.110

The associated taxa are clustered at the phylum and order levels

DNN deep neural network, *DNN* + ℓ_1 Lasso (least absolute shrinkage and selection operator) penalized deep neural network

4 Discussion and conclusions

The proposed `DeepBiome`, a phylogenetic tree-regularized deep learning model, can be used for prediction and classification tasks. We provide comprehensive simulation experiments and real data applications to demonstrate the superiority of `DeepBiome`. For regression tasks, our results suggest that, compared to sparse regression and other deep learning models, `DeepBiome` has a competitive performance, particularly when microbiome taxa associated with the outcome are

Table 4 Scenario 2: classification performance for binary outcomes

Method	Testing										Training									
	Sensitivity		Specificity		g-Measure		ACC		AUC		Sensitivity		Specificity		g-Measure		ACC		AUC	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Logistic	0.920	0.030	0.725	0.066	0.815	0.038	0.860	0.026	0.822	0.033	0.950	0.021	0.785	0.049	0.863	0.030	0.900	0.020	0.867	0.026
Lasso	0.965	0.016	0.583	0.086	0.747	0.056	0.848	0.027	0.774	0.041	0.973	0.006	0.619	0.081	0.774	0.053	0.866	0.023	0.796	0.040
Ridge	0.957	0.018	0.461	0.077	0.661	0.055	0.805	0.027	0.709	0.037	0.970	0.007	0.522	0.062	0.711	0.043	0.835	0.016	0.746	0.030
Elastic-Net	0.998	0.004	0.022	0.020	0.121	0.082	0.699	0.030	0.510	0.010	0.998	0.002	0.022	0.013	0.140	0.046	0.702	0.015	0.510	0.006
DNN	0.887	0.049	0.725	0.148	0.788	0.133	0.837	0.038	0.897	0.039	0.932	0.032	0.869	0.158	0.887	0.146	0.913	0.043	0.958	0.033
DNN + ℓ_1	0.887	0.048	0.727	0.146	0.790	0.129	0.838	0.038	0.898	0.036	0.931	0.031	0.868	0.154	0.887	0.141	0.912	0.042	0.958	0.030
Random Forest	0.731	0.063	0.977	0.013	0.844	0.036	0.902	0.021	0.854	0.031	0.730	0.044	0.976	0.008	0.844	0.024	0.902	0.011	0.853	0.021
DeepBiome	0.918	0.042	0.835	0.111	0.870	0.093	0.892	0.044	0.941	0.051	0.952	0.033	0.922	0.106	0.932	0.094	0.943	0.041	0.974	0.048

ACC accuracy, *AUC* area under the receiver operating characteristic (ROC) curve, *DNN* deep neural network, *DNN* + ℓ_1 Lasso (least absolute shrinkage and selection operator) penalized deep neural network

Table 5 Scenario 3: classification performance for multi-categorical outcomes

Method	Testing										Training									
	ACC		Precision		Recall		F1		AUC		ACC		Precision		Recall		F1		AUC	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
SVM-Linear	0.667	0.032	0.525	0.060	0.493	0.027	0.508	0.039	0.667	0.027	0.694	0.017	0.571	0.034	0.520	0.019	0.544	0.024	0.679	0.018
SVM-Radial	0.659	0.033	0.527	0.092	0.506	0.024	0.514	0.050	0.665	0.027	0.821	0.012	0.807	0.035	0.663	0.017	0.728	0.020	0.809	0.015
SVM-Polynomial	0.576	0.034	0.443	0.090	0.394	0.026	0.414	0.048	0.566	0.030	0.767	0.014	0.837	0.012	0.622	0.025	0.714	0.019	0.742	0.026
DNN	0.752	0.045	0.620	0.136	0.599	0.105	0.599	0.115	0.856	0.041	0.854	0.063	0.741	0.188	0.718	0.149	0.718	0.166	0.941	0.044
DNN + ℓ_1	0.760	0.042	0.668	0.139	0.612	0.090	0.618	0.104	0.863	0.043	0.864	0.058	0.794	0.160	0.731	0.130	0.739	0.146	0.948	0.037
Random Forest	0.871	0.022	0.861	0.031	0.802	0.033	0.820	0.033	0.807	0.037	0.870	0.012	0.859	0.017	0.800	0.022	0.821	0.021	0.806	0.024
DeepBiome	0.815	0.066	0.720	0.151	0.714	0.115	0.711	0.135	0.900	0.080	0.880	0.072	0.804	0.170	0.793	0.135	0.791	0.157	0.942	0.084

ACC accuracy, *AUC* area under the receiver operating characteristic (ROC) curve, *DNN* deep neural network, *DNN* + ℓ_1 Lasso (least absolute shrinkage and selection operator) penalized deep neural network, *SVM* support vector machine

Table 6 Scenario 4: mean squared error (MSE) and Pearson correlation coefficient between predicted and true outcome (continuous), when the input microbiome abundance data contain measurement errors

Method	Testing				Training			
	MSE		Correlation		MSE		Correlation	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Linear Regression	1.569	0.154	0.639	0.036	1.336	0.066	0.694	0.018
Ridge	1.551	0.128	0.639	0.036	1.358	0.073	0.694	0.018
Lasso	1.488	0.119	0.661	0.034	1.408	0.073	0.679	0.020
Elastic-Net	1.490	0.121	0.660	0.034	1.402	0.075	0.681	0.019
DNN	0.619	0.682	0.873	0.137	0.188	0.317	0.961	0.068
DNN + ℓ_1	0.445	0.351	0.909	0.081	0.129	0.234	0.974	0.050
Random Forest	0.419	0.072	0.922	0.016	0.425	0.037	0.920	0.008
DeepBiome	0.243	0.400	0.950	0.087	0.117	0.244	0.976	0.052

The associated taxa are clustered at the phylum and order levels

DNN deep neural network, *DNN* + ℓ_1 Lasso (least absolute shrinkage and selection operator) penalized deep neural network

Table 7 Scenario 4: mean squared error (MSE) and Pearson correlation coefficient between predicted and true outcome (continuous), when using an mis-specified phylogenetic tree

Method	Testing				Training			
	MSE		Correlation		MSE		Correlation	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Linear Regression	0.872	0.163	0.683	0.046	0.737	0.074	0.726	0.027
Lasso	0.826	0.144	0.706	0.047	0.780	0.080	0.710	0.030
Ridge	0.866	0.138	0.683	0.046	0.752	0.081	0.726	0.027
Elastic-Net	0.826	0.144	0.706	0.047	0.779	0.079	0.711	0.028
DNN	0.437	0.208	0.849	0.077	0.167	0.166	0.944	0.058
DNN + ℓ_1	0.434	0.214	0.850	0.080	0.166	0.171	0.944	0.065
Random Forest	0.194	0.050	0.939	0.019	0.198	0.025	0.937	0.009
DeepBiome	0.316	0.261	0.892	0.094	0.195	0.207	0.933	0.075

The associated taxa are clustered at the phylum and order levels

DNN deep neural network, *DNN* + ℓ_1 Lasso (least absolute shrinkage and selection operator) penalized deep neural network

clustered at different phylogenetic levels. `DeepBiome` also excels in complex classification tasks with higher accuracy and AUC. More importantly, `DeepBiome` enables an intuitive visualization of the microbiome–phenotype association network.

Deep learning models gain lots of popularity due to their supremacy in imaging and natural language analysis. However, typical biomedical studies can rarely afford the huge amount of training data required for hyper-parameter tuning [9, 27]. `DeepBiome` regularizes the neural network structure towards the phylogenetic structure inherent in the microbiome data through weight decay. This way, it greatly

Table 8 Predict Type 2 Diabetes (T2D) status using microbiome counts in the metagenomics sequencing study

Method	Testing										Training									
	Sensitivity		Specificity		g-Measure		ACC		AUC		Sensitivity		Specificity		g-Measure		ACC		AUC	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Logistic	0.533	0.067	0.509	0.086	0.517	0.034	0.526	0.021	0.521	0.029	1.000	0.000	1.000	0.000	1.00	0.000	1.000	0.000	1.00	0.00
Lasso	0.331	0.332	0.819	0.183	0.364	0.335	0.558	0.107	0.575	0.076	0.330	0.320	0.884	0.118	0.395	0.364	0.615	0.093	0.607	0.103
Ridge	0.167	0.373	0.892	0.241	0.124	0.277	0.517	0.082	0.529	0.066	0.187	0.418	0.935	0.146	0.159	0.355	0.569	0.134	0.561	0.136
Elastic-Net	0.409	0.284	0.785	0.156	0.476	0.272	0.578	0.108	0.597	0.067	0.409	0.292	0.785	0.090	0.528	0.310	0.66	0.089	0.653	0.102
DNN	0.585	0.174	0.647	0.184	0.596	0.075	0.616	0.070	0.681	0.054	0.767	0.118	0.792	0.106	0.773	0.023	0.78	0.020	0.853	0.026
DNN + ℓ_1	0.582	0.199	0.680	0.195	0.606	0.100	0.625	0.086	0.678	0.079	0.742	0.106	0.808	0.114	0.768	0.029	0.776	0.026	0.856	0.034
Random Forest	0.520	0.020	0.710	0.020	0.610	0.010	0.590	0.010	0.610	0.010	0.640	0.050	0.700	0.050	0.670	0.041	0.670	0.040	0.670	0.039
DeepBiome	0.640	0.180	0.645	0.200	0.620	0.066	0.643	0.054	0.694	0.037	0.836	0.142	0.888	0.086	0.856	0.052	0.863	0.046	0.948	0.015

reduces the number of parameters, including the architecture itself, to be tuned and trained, avoids overfitting, and allows visualization of the pathway from microbiome counts to phenotypes. The limitations of `DeepBiome` include the possibility of violation of the assumptions: (1) microbiome abundances classified in the same cluster have similar effects to outcomes of interests, and (2) phylogenetic tree structure translates to effects aggregation structure.

In real-world applications, the number of features (e.g., microbial species or genes) may differ between training and testing datasets, posing a significant challenge for most machine learning models, including `DeepBiome`. We have the following considerations:

(1) Prior to training, align the features of both datasets by selecting a common set of features or using techniques like canonical correlation analysis to find a harmonized feature space.
(2) `DeepBiome`'s architecture can handle inputs of varying dimensions.
(3) Train `DeepBiome` on the larger feature set and then apply transfer learning techniques to adapt the model to the smaller feature set in the testing phase.

We defer the details of the investigation to future research.

5 Availability and Requirements

Project name: `DeepBiome`
Project home page: https://github.com/Young-won/DeepBiome
Operating system(s): Platform independent
Programming language: Python
Other requirements: None
License: Open source
Any restrictions to use by non-academics: None

Author Contributions Jing Zhai, Youngwon Choi, Hua Zhou, and Jin J. Zhou designed the model and the computational framework. Jing Zhai and Youngwon Choi implemented the method and carried out the analysis. Kenneth Knox and Homer L. Twigg III provided lung microbiome datasets. Yin Chen, Kenneth Knox, Homer L. Twigg III, and Joong-Ho Won helped to interpret the results. Joong-Ho Won supported computing recourse. All authors provided critical feedback and helped shape the research, analysis, and manuscript. Jin J. Zhou supervised the project.

Funding This research was partially funded by grants from the National Institute of General Medical Sciences (R35GM141798 HZ), the National Human Genome Research Institute (R01HG006139 HZ and JJZ), the National Science Foundation (DMS-2054253 and IIS-2205441 HZ and JJZ), and the National Heart, Lung, and Blood Institute (R21HL150374 JJZ, UO1 HL121831, and UO1 HL098960 HT and KK).

Data Availability AGP metadata and the operational taxonomic unit (OTU) table were downloaded from https://github.com/biocore/American-Gut/tree/master/data. The phylogenetic tree was extracted from the *.biom* file that contains OTU tables and the taxonomic information. Lung microbiome data is available upon request.Consent

for Publication All utilized microbiome datasets are de-identified secondary data analysis. No consent for publication was required for this study.

Declarations

Conflict of interest The authors declare that they have no competing interests.

Ethical Approval All utilized microbiome datasets are de-identified secondary data analysis. No ethics approval was required for this study.

Informed Consent No consent to participate was required for this study.

References

1. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, Kudlur M, Levenberg J, Monga R, Moore S, Murray DG, Steiner B, Tucker P, Vasudevan V, Warden P, Wicke M, Yu Y, Zheng X (2016) TensorFlow: a system for large-scale machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI 16), 2016, pp 265–283. https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf
2. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas Fernanda, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous systems. http://tensorflow.org/
3. Bergstra JS, Bardenet R, Bengio Y, Kégl B (2011) Algorithms for hyper-parameter optimization. In: Shawe-Taylor J, Zemel RS, Bartlett PL, Pereira F, Weinberger KQ (eds) Advances in neural information processing systems 24, 2011. Curran Associates, Inc., pp 2546–2554. http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf
4. Caporaso JG, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, Costello EK, Fierer N, Peña AG, Goodrich JK, Gordon JI et al (2010) QIIME allows analysis of high-throughput community sequencing data. Nat Methods 7(5):335–336
5. Chen J, Bushman FD, Lewis JD, Wu GD, Li H (2012) Structure-constrained sparse canonical correlation analysis with an application to microbiome data analysis. Biostatistics 14(2):244–258
6. Chen L, Liu H, Kocher J-PA, Li H, Chen J (2015) glmgraph: an R package for variable selection and predictive modeling of structured genomic data. Bioinformatics 31(24):3991–3993
7. Chollet F et al (2015) Keras. https://keras.io
8. Cole JR, Wang Q, Fish JA, Chai B, McGarrell DM, Sun Y, Brown CT, Porras-Alfaro A, Kuske CR, Tiedje JM (2014) Ribosomal database project: data and tools for high throughput rRNA analysis. Nucleic Acids Res 42(D1):D633–D642
9. De Fauw J, Ledsam JR, Romera-Paredes B, Nikolov S, Tomasev N, Blackwell S, Askham H, Glorot X, O'Donoghue B, Visentin D et al (2018) Clinically applicable deep learning for diagnosis and referral in retinal disease. Nat Med 24(9):1342
10. Franzosa EA, Sirota-Madi A, Avila-Pacheco J, Fornelos N, Haiser HJ, Reinker S, Vatanen T, Hall AB, Mallick H, McIver LJ et al (2019) Gut microbiome structure and metabolic activity in inflammatory bowel disease. Nat Microbiol 4(2):293
11. Garcia TP, Müller S, Carroll RJ, Walzem RL (2013) Identification of important regressor groups, subgroups and individuals via regularization methods: application to gut microbiome data. Bioinformatics 30(6):831–837
12. Gilbert JA, Quinn RA, Debelius J, Xu ZZ, Morton J, Garg N, Jansson JK, Dorrestein PC, Knight R (2016) Microbiome-wide association studies link dynamic microbial consortia to disease. Nature 535(7610):94

13. Gupta A, Lam SM (1998) Weight decay backpropagation for noisy data. Neural Netw 11(6):1127–1138
14. Haykin S (1994) Neural networks, vol 2. Prentice Hall, New York
15. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. _eprint: 1412.6980
16. Krogh A, Hertz JA (1992) A simple weight decay can improve generalization. In: Advances in neural information processing systems, 1992, pp 950–957
17. Liaw A, Wiener M (2002) Classification and regression by randomForest. R News 2(3):18–22. https://CRAN.R-project.org/doc/Rnews/
18. Lu YY, Fan Y, Lv J, Noble WS (2018) DeepPINK: reproducible feature selection in deep neural networks. CoRR abs/1809.01185. _eprint: 1809.01185. http://arxiv.org/abs/1809.01185
19. Maas AL, Hannun AY, Ng AY (2013) Rectifier nonlinearities improve neural network acoustic models. Proc ICML 30(1):3
20. Mundie DB, Massengill LW (1991) Weight decay and resolution effects in feedforward artificial neural networks. IEEE Trans Neural Netw 2(1):168–170
21. Ni J, David Shen T-C, Chen EZ, Bittinger K, Bailey A, Roggiani M, Sirota-Madi A, Friedman ES, Chau L, Lin A et al (2017) A role for bacterial urease in gut dysbiosis and Crohn's disease. Sci Transl Med 9(416):eaah6888
22. Oh M, Zhang L (2020) DeepMicro: deep representation learning for disease prediction based on microbiome data. Sci Rep 10(1):6026. ISSN 2045-2322. https://doi.org/10.1038/s41598-020-63159-5
23. Pasolli E, Truong DT, Malik F, Waldron L, Segata N (2016) Machine learning meta-analysis of large metagenomic datasets: tools and biological insights. PLoS Comput Biol 12(7):e1004977
24. Quast C, Pruesse E, Yilmaz P, Gerken J, Schweer T, Yarza P, Peplies J, Glöckner FO (2012) The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. Nucleic Acids Res 41(D1):D590–D596
25. Reiman D, Metwally A, Dai Y (2017) Using convolutional neural networks to explore the microbiome. In: 2017 39th Annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2017. IEEE, pp 4269–4272
26. Sartor RB (2008) Microbial influences in inflammatory bowel diseases. Gastroenterology 134(2):577–594
27. Shen D, Wu G, Suk H-I (2017) Deep learning in medical image analysis. Annu Rev Biomed Eng 19:221–248
28. Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (eds) Advances in neural information processing systems 25, 2012. Curran Associates, Inc., pp. 2951–2959. http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf
29. Truong DT, Franzosa EA, Tickle TL, Scholz M, Weingart G, Pasolli E, Tett A, Huttenhower C, Segata N (2015) MetaPhlAn2 for enhanced metagenomic taxonomic profiling. Nat Methods 12(10):902–903
30. Twigg HL III, Knox KS, Zhou J, Crothers KA, Nelson DE, Toh E, Day RB, Lin H, Gao X, Dong Q et al (2016) Effect of advanced HIV infection on the respiratory microbiome. Am J Respir Crit Care Med 194(2):226–235. https://doi.org/10.1164/rccm.201509-1875OC
31. Wang T, Zhao H (2017) Constructing predictive microbial signatures at multiple taxonomic levels. J Am Stat Assoc 112(519):1022–1031
32. Wang Y, Bhattacharya T, Jiang Y, Qin X, Wang Y, Liu Y, Saykin AJ, Chen L (2021) A novel deep learning method for predictive modeling of microbiome data. Brief Bioinform 22(3):bbaa073. ISSN 1467-5463, 1477-4054. https://doi.org/10.1093/bib/bbaa073
33. Xiao J, Cao H, Chen J (2017) False discovery rate control incorporating phylogenetic tree increases detection power in microbiome-wide multiple testing. Bioinformatics 33(18):2873–2881
34. Xiao J, Chen L, Johnson S, Zhang X, Chen JC (2018) Predictive modeling of microbiome data using a phylogeny-regularized generalized linear mixed model. Front Microbiol 9:1391
35. Zhai J, Kim J, Knox KS, Twigg HL III, Zhou H, Zhou JJ (2018) Variance component selection with applications to microbiome taxonomic data. Front Microbiol 9:509
36. Zhai J, Knox K, Twigg HL III, Zhou H, Zhou JJ (2019) Exact variance component tests for longitudinal microbiome studies. Genet Epidemiol 43(3):250–262
37. Zhang G, Wang C, Xu B, Grosse R (2018) Three mechanisms of weight decay regularization. _eprint: 1810.12281

38. Zhou JJ, Zhai J, Zhou H, Chen Y, Guerra S, Robey I, Weinstock GM, Weinstock E, Dong Q, Knox KS, Twigg III HL (2020) Supraglottic lung microbiome taxa are associated with pulmonary abnormalities in an HIV longitudinal cohort. Am J Respir Crit Care Med. 202(12):1727–1731. https://doi.org/10.1164/rccm.202004-1086LE

Authors and Affiliations

Jing Zhai[1] · Youngwon Choi[2,3] · Xingyi Yang[1] · Yin Chen[4] · Kenneth Knox[5] · Homer L. Twigg III[6] · Joong-Ho Won[2] · Hua Zhou[7] · Jin J. Zhou[7,8,9]

✉ Jin J. Zhou
jinjinzhou@ucla.edu

1 Department of Epidemiology and Biostatistics, College of Public Health, University of Arizona, Tucson, AZ 85724, USA

2 Department of Statistics, Seoul National University, Seoul 08826, Korea

3 UCLA Center for Vision & Imaging Biomarkers, Los Angeles, USA

4 Department of Pharmacology and Toxicology, College of Pharmacology, University of Arizona, Tucson, AZ 85724, USA

5 Division of Pulmonary, Allergy, Critical Care, Sleep Medicine, Department of Medicine, University of Arizona, Tucson, AZ 85724, USA

6 Division of Pulmonary, Critical Care, Sleep, and Occupational Medicine, Indiana University Medical Center, Indianapolis, IN 46202, USA

7 Department of Biostatistics, University of California, Los Angeles, CA 90095, USA

8 Department of Medicine, University of California, Los Angeles, CA 90095, USA

9 Department of Medicine and Biostatistics, University of California, Los Angeles, 1100 Glendon Ave. Suite 1820, Los Angeles, CA 90024, USA