PopPhy-CNN: A Phylogenetic Tree Embedded Architecture for Convolutional Neural Networks to Predict Host Phenotype From Metagenomic Data

Derek Reiman, *Member, IEEE*, Ahmed A. Metwally, *Member, IEEE*, Jun Sun, and Yang Dai, *Member, IEEE*

***Abstract*—Accurate prediction of the host phenotype from a metagenomic sample and identification of the associated microbial markers are important in understanding potential host-microbiome interactions related to disease initiation and progression. We introduce PopPhy-CNN, a novel convolutional neural network (CNN) learning framework that effectively exploits phylogenetic structure in microbial taxa for host phenotype prediction. Our approach takes an input format of a 2D matrix representing the phylogenetic tree populated with the relative abundance of microbial taxa in a metagenomic sample. This conversion empowers CNNs to explore the spatial relationship of the taxonomic annotations on the tree and their quantitative characteristics in metagenomic data. We show the competitiveness of our model compared to other available methods using nine metagenomic datasets of moderate size for binary classification. With synthetic and biological datasets, we show the superior and robust performance of our model for multi-class classification. Furthermore, we design a novel scheme for feature extraction from the learned CNN models and demonstrate improved performance when the extracted features. PopPhy-CNN is a practical deep learning framework for the prediction of host phenotype with the ability of facilitating the retrieval of predictive microbial taxa.**

***Index Terms*—Microbiome, disease prediction, deep learning, feature evaluation.**

Manuscript received June 4, 2019; revised January 28, 2020; accepted February 7, 2020. Date of publication May 11, 2020; date of current version October 5, 2020. *(Corresponding author: Yang Dai.)*

Derek Reiman and Yang Dai are with the Department of Bioengineering, University of Illinois at Chicago, Chicago, IL 60612 USA (e-mail: dreima2@uic.edu; yangdai@uic.edu).

Ahmed A. Metwally is with the Department of Bioengineering and the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60612 USA and also with the Department of Systems and Biomedical Engineering, Faculty of Engineering, Cairo University, Giza 12613, Egypt (e-mail: ametwall@stanford.edu).

Jun Sun is with the Department of Medicine, University of Illinois at Chicago, Chicago, IL 60612 USA (e-mail: junsun7@uic.edu).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/JBHI.2020.2993761

I. Introduction

NUMEROUS metagenomic studies of the gut microbiome have linked dysbiosis to many host diseases [1]. A metagenomic sample is usually described by its microbial taxanomic composition, i.e., the relative abundance of microbial taxa at one of the taxonomic levels (Super-kingdom, Phylum, Class, Order, Family, Genus, and Species), represented as nodes on a phylogenetic taxonomic tree. The identification of microbial taxa that are associated with the host disease can benefit the early diagnosis, the development of microbial reconstitution (e.g., Probiotic) therapies [2], and the understanding of the disease mechanism [3].

One primary effort on the analysis of microbiome has been the disease association study and the identification of microbial biomarker signatures for disease prediction. The detection of the associations relies on statistical analyses (parametric or non-parametric) to identify differentially abundant taxa between disease and control groups [4]–[8]. However, the association of the individual microbes to a particular type of disease has shown contradictory results [9], [10]. This can be due to various reasons such as the dynamic nature of microbes, small sample size, and disease complexity.

Alternative approaches using machine learning (ML) models, e.g., Random Forest (RF), least absolute shrinkage and selection operator (LASSO), and Support Vector Machines (SVMs) based on input representations of relative abundance of microbial taxa or gene annotations have demonstrated the potential of developing a microbial biomarker signature for the prediction of the host phenotype [11]–[13]. These types of approaches are motivated by the findings that a microbial signature for the host phenotype may be complex, involving simultaneous over- and under-representations of multiple microbial taxa at distinct taxonomic levels and potentially interacting with each other [9], [14]. The initial applications of ML models did not extensively explore the relationship among the taxa, achieving moderate level of predictive performance.

Recent studies have shown that constructing abundance of features using the hierarchical structure of the taxonomic tree can lead to better classification performance over the use of only raw features [15], [16]. The use of phylogenetic tree to imprint

relevant biological knowledge in metagenomic data has been seen in different machine learning models. For example, a class of phylogenetic-based feature weighting algorithms was proposed to group the relevant taxa into clades, and the highly ranked clade groups in conjunction with RF had an improved classification performance [17]. In another study [18], a phylogeny-based smoothness penalty is introduced to smooth the coefficients of the microbial taxa with respect to the phylogenetic tree in both linear and logistic regression models. It was shown that the models improved over other regression-based models in both biological and synthetic datasets. However, this model is limited in exploring the natural correlation structure among microbial taxa that exists according to phylogenetic relationships.

Several methods using deep neural networks (DNNs) were proposed in the hope that DNNs could identify more complex relationship among the microbial taxa that benefit the phenotype prediction. The first relatively large scale evaluation is the application of multi-layer perceptron neural network (MLPNN) and recursive neural network (RNN) using the input form of the relative Operational Taxonomic Unit (OTU) vectors [19] for a metagenomic sample. However, it has been shown that a simple layer neural network and RF performed better than DNN models, although RNNs could reveal a hierarchical structure among the samples. More recently, Fioravanti *et al*. proposed a convolutional neural network (CNN) architecture that explores the distance between nodes on a phylogenetic tree by the patristic distance (the sum of the lengths of all branches connecting two OTUs on the tree) [20]. Their approach is to embed the phylogenetic tree in an Euclidean space and apply convolution over k nearest neighbors. The evaluation for their method (called Ph-CNN) reported promising results on synthetic data using gut metagenomic data from 222 inflammatory bowel disease patients and 38 healthy subjects compared to linear SVMs, RF, and a fully connected multi-layer perceptron neural network.

The performance of these DNN models is encouraging, owing the ability of deep architectures in identifying potential interactions of microbial taxa for host phenotype prediction. However, the results also raise the skepticism that DNNs may not be suitable learning models due to their requirement of large amounts of training data, which is impractical in present metagenomic studies [19]. A recent work summarized all available standard ML and DNN models for host phenotype prediction and shows the evaluation of DNNs is incomplete and DNNs were superior than other standard ML models [21]. Furthermore, DNNs are often used as black-boxes, making it difficult to extract informative features from the learned models.

In this work, we introduce PopPhy-CNN, a novel CNN framework expanded from our previous work [22] to address the criticism to DNNs mentioned above. Our model takes advantage of CNNs' ability in generating convolutional layers with multiple feature maps that capture spatial information in training data, such as in images [23]. Since a metagenomic profile is usually represented by a vector of relative abundances of microbial taxa in arbitrary orders, a scheme to convert this information into a biological structure is needed. To empower CNNs for metagenomic phenotype prediction, we construct a phylogenetic tree to preserve the relationship among the microbial taxa in the profiles. The tree is then populated with the relative abundance of microbial taxa in each individual profile and represented in a 2D matrix. The constructed matrices provide spatial and quantitative information in the metagenomic data, which are more suitable to CNNs compared to the input vectors of relative microbial taxa abundances in an arbitrary order. Our method takes a completely distinct approach in utilizing the information of a phylogenetic tree compared to the one used in Ph-CNN [20].

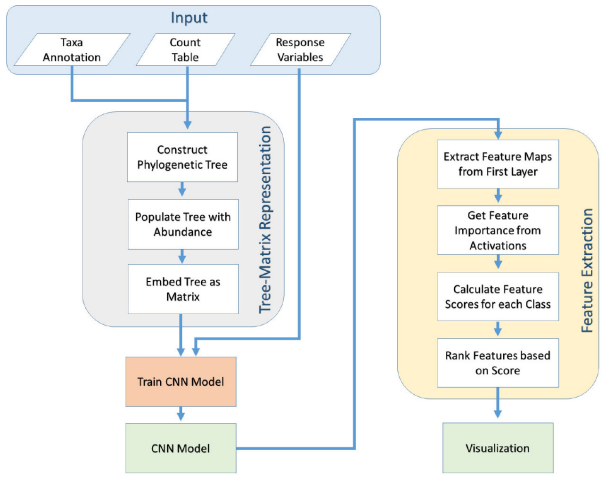

Fig. 1. Flowchart of PopPhy-CNN. The taxa and count table are used to create and populate a phylogenetic tree, which is represented as a matrix and used to train a CNN. Features are extracted from the trained model.

We demonstrate PopPhy-CNN's competitive and robust performance for binary classification using nine publicly available datasets by benchmarking against several well-established ML methods and other DNNs including Ph-CNN. Using both biological and synthetic datasets, we also compare their abilities in handling multi-class classification problems. In order to gauge what has been learned by PopPhy-CNN, we further design a novel procedure to retrieve informative microbial taxa from the trained CNN models and evaluate their usefulness. Finally, we include a visualization to facilitate the interpretation of the retrieved taxa on the phylogenetic tree.

II. Materials and Methods

The major components of PopPhy-CNN is shown in Fig. 1. We first describe how a microbial taxonomic abundance profile obtained from a sample can be represented in a 2D matrix based on the use of a populated phylogenetic tree. Then, we describe our CNN architecture and the training procedure. Last, the scheme of feature extraction will be presented.

A. Representing Metagenomic Profiles in 2D Matrices

We developed a prototype of our algorithm to transform the microbial taxonomic abundance profiles into a structured data by using a phylogenetic tree [22]. The detail of the algorithms for tree construction and population can be found in the Appendix. Our method is demonstrated using taxonomic profile data

represented by the relative abundances of the OTUs. However, it is applicable to profiles of any level of taxonomic annotation obtained from metagenomic study.

Briefly, a phylogenetic tree that captures similarity information among OTUs can be constructed by comparing the microbial genomes based on multiple sequence alignment and organizing similar taxa into clades. The similarity between taxa is represented by their closeness in the tree. In our work, PhyloT [24] was used to create the phylogenetic tree, and a constant distance of one between nodes in the tree is assumed. The phylogenetic tree is structured using ancestral nodes from both taxonomic groups and subgroups with no defined distances between nodes. Therefore, we define the distance between any two nodes by the number of nodes between them and the tree is essentially a taxonomic tree.

The tree is used as a template to construct a populated tree for each sample in the dataset. The value of each OTU from a sample is assigned to its respective node in the tree. The tree is then populated such that an abundance value for each internal node is equal to the sum of its children's abundance values. Once the tree has been annotated with abundance values, it is transformed into a matrix by placing the root's abundance in the top left corner of a matrix. Then for a given row, the children of the nodes from that row are selected and their abundances are placed in the subsequent row in the order that their parents appear, starting with the left most column. The rest of the matrix is filled with zeros. We represent the tree this way in order to allow the CNN model to have a dense pocket of data. Given a graph $G = \{V, E\}$, this representation has a memory complexity of $O(V^2)$. However, in our evaluation, we found the matrix size to scale at 4.93 V on average, showing that it uses drastically less memory than an adjacency matrix of size V^2. Compared to adjacency matrix, our representation maintains a smaller number of features so PopPhy-CNN can be trained without excessive amount of data.

B. Architecture of Convolutional Neural Network

Standard CNNs are composed of multiple convolutional layers followed usually by at least one fully connected layer. Each convolutional layer is composed of multiple kernels, each of which transforms an input matrix M into a set of feature maps of velocities through a convolutional operation. The feature maps composed of these velocities are then passed through a non-linear activation function and subsampled through max or mean pooling to give a matrix of activations.

Our CNN architectures consist of two convolutional layers followed by a single fully connected layer and a single output layer. The first convolutional layer contains a rectangular filter to scan areas of local features. The second convolutional layer consists of a single 1×1 kernel. This collapses the set of feature maps from the first convolutional layer into a single feature map in order to reduce the number of network parameters. Each layer uses the exponential linear unit (ELU) activation function. In our studies, we observed that max-pooling was sometimes detrimental to prediction, so our model does not perform any pooling. The softmax activation function was applied to the output layer for class prediction. The model was trained using a weighted cross entropy loss function to help address class imbalance. To prevent overfitting, we regularize the networks using both L1 and L2 normalization penalties on the weights as well as dropout in the fully connected layers [25].

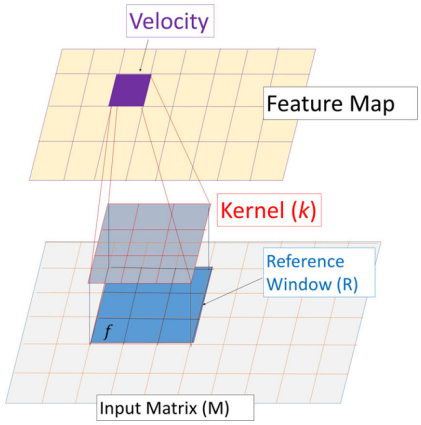

Fig. 2. A kernel k slides over the input matrix M. Each position in the feature map contains a velocity which is the element wise sum of the Hadamard product between k and a submatrix of M. We call this submatrix a reference window and denote it as R.

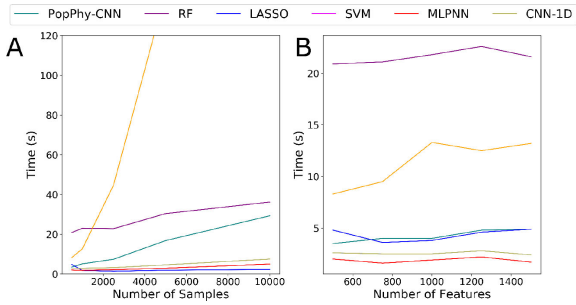

Fig. 3. Time complexity for machine learning models based on (A) number of samples and (B) number of features.

C. Extraction of the Informative Features

To address the criticism that deep learning models lack interpretability, we attempt to push past the black-box of the CNNs by extracting the important features for the learned models. A previous study has shown that using feature maps captured by CNN models as features for other machine learning models (i.e., RF and SVMs) yielded better results than using the raw features [26]. Even though deeper layers yielded better features in the previous study, the loss of resolution through subsampling and extra layers of nonlinear transformations could jeopardize interpretability. Therefore, we focus on the post analysis of the feature maps generated by the first convolutional layer. A visualization for the generation of a feature map is shown in Fig. 2.

To do this, we take the feature maps generated by a kernel k across all the samples for a specific class c in the training set. For each of these feature maps, we take the positions of a proportion of maximum values specified by a given hyper-parameter, θ_1. We then select the maximums which were found in at least a

proportion, θ_2, of the samples for that class. For each velocity selected, we trace its location in the feature map back to the submatrix of the input M from which it was calculated. We call this matrix R our reference window.

Every position (i, j) of a reference window represents some node v from the phylogenetic tree with an OTU label, f. We calculate the importance of each feature f given the reference window R for sample S as its proportion of the velocity.

$$I_s^{(k)}(f \mid R) = \frac{W^{(k)}(i,j) * R_S(i,j)}{\sum(|W^{(k)}| \odot R_S)} \quad s.t. \quad R(i,j) \leftrightarrow f \quad (1)$$

Here k is our current kernel with weights $W^{(k)}$ and the summation is over all positions in R_S. The absolute values of the weights in the denominator are used to scale all importance values to be between 1 and -1, and the absolute value of all importance values within a reference window will sum to 1.

Within a single reference window, some taxa may score highly in a small subset of samples but may not be important considering all of the samples. In order to better capture the taxa which were consistently found important, we calculate the mean importance value of a feature f across all samples in class c given a single reference window R and kernel k.

$$I_c^{(k)}(f \mid R) = \frac{\sum_{s \in c} I_s^{(k)}(f \mid R)}{n_c} \quad (2)$$

Here n_c represents the number of samples in class c. Since a feature is present in multiple reference windows and kernels, a single feature may have multiple importance values. To handle this, we selected the importance of f to be the maximum over all reference windows containing f and over all kernels, k.

$$I_c(f) = \max_{R,k} \{\, I_c^{(k)}(f \mid R) \,\} \quad (3)$$

Lastly, we assigned a score for a feature from the perspective of class c as the difference of the feature importance using all the samples within the class and the feature importance using all the samples not in the class.

$$S_c(f) = I_c(f) - I_{\bar{c}}(f) \quad (4)$$

From these scores we create a list of feature scores for each class, allowing the analysis of feature importance from the perspective of different classes that can then be ranked. The algorithm for this feature extraction is shown in Algorithm 1.

Algorithm 1: CNN Feature Evaluation.

Data: A set of inputs where each input M is a matrix representation of tree $G = \{V,E\}$ with class c, a trained CNN model with kernels k of weights $W^{(k)}$, and two filtering parameters θ_1, θ_2

Result: A list of taxa scores $Sc(f)$ for each class

```
Z ← zero matrix with dimensions |c| × |K| × |V|;
for each sample s with class c and each kernel k do
    Generate and vectorize the feature map;
    for each index ℓ1 of the top (θ1 * |V|) values do
        Increment Z(c, k, ℓ1) by 1;
    end
end
for each class c, kernel k, and sample s ∈ c with input
 M do
    for each index ℓ2 in the top θ2 values of Z(c, k,:) do
        Find the submatrix R ∈ Ms used to calculate ℓ;
        for each position (i, j) in R where R(i,j) ≡
         node v ∈ V do
            f ← the taxa label of v;
            I_s^(k)(f | R) ← W(i,j)*R(i,j) / Σ|W^(k)| ⊙ R;
        end
        I_c^(k)(f | R) ← mean of I_s^(k)(f | R) for s ∈ c ;
    end
end
S_c(f) ← max_{R,k} I_c^(k)(f | R) − max_{R,k} I_c̄^(k)(f | R);
Return the set of scores S_c(f) for each class c ;
```

D. Datasets Used in Evaluation

We used nine publicly available datasets to evaluate PopPhy-CNN. Three datasets are contained within the MetAML package [13]: cirrhosis, type 2 diabetes (T2D), and obesity. They were selected due to the varying difficulty for prediction. The cirrhosis dataset was taken from a study of 114 cirrhosis patients and 118 healthy subjects [27]. The T2D dataset was a combination of two studies [4], [28] yielding a total of 223 patients with T2D and 223 healthy subjects. The obesity dataset comes from a study of 292 individuals of which 89 individuals with a BMI lower than 25 kg/m^2 were studied against 164 individuals with a BMI greater than 30 kg/m^2 [29]. Each of these datasets was generated using Metagenomic Shotgun (MGS) sequencing. In the MetAML study [13], the OTUs for each dataset were assigned by MetaPhlAn2, which selects OTUs based on the read coverage of clade-specific markers and then estimates their relative abundance [30].

The six other datasets were taken from a study on inflammatory bowel disease (IBD) investigating the differences between the microbial community during disease remission and flares [31]. The dataset contains 38 healthy samples and 222 samples from patients with IBD. In our experiment, we separated the data into three disease categories: Crohne's disease (CD), ileal Crohne's disease (iCD), and ulcerative cholitis (UC). The datasets were further broken into two sets where one set constitutes patients with lessening conditions who were in remission and one set with patients whose condition was worsening. This gave six total datasets: 59 patients with iCD disease who were in remission (iCDr), 44 patients with iCD whose symptoms were worsening (iCDf), 76 patients with CD who were in remission (CDr) or whose symptoms were worsening (CDf), 44 patients with UC whose symptoms were in remission (UCr), and 41 patients with UC whose symptoms were worsening (UCf). These datasets were selected for benchmarking against Ph-CNN since the coordinates required for the method are provided [20].

The OTUs in each dataset were aggregated at genus level. In addition, the OTUs from the cirrhosis, T2D, and obesity dataset were also aggregated at the species level. Full taxonomic trees were obtained using PhyloT [24] and were pruned based on the observed OTUs. For any OTU which was specified as "unclassified", the label for the that OTU at the next highest

TABLE I
TABLE SHOWING NUMBER OF CASE AND CONTROL SAMPLES, ORIGINAL OTUS, AND NODES IN THE TREE FOR BINARY DATASETS

	Case	Control	Genus		Species	
			OTUs	Nodes	OTUs	Nodes
Cirrhosis	114	118	184	428	542	933
T2D	223	227	214	478	606	1054
Obesity	164	89	181	426	5465	872
CDf	60	38	224	388	-	-
CDr	76	38	219	386	-	-
iCDf	44	38	213	373	-	-
iCDr	59	38	219	386	-	-
UCf	41	38	213	384	-	-
UCr	44	38	203	361	-	-

TABLE II
TABLE SHOWING NUMBER OF SAMPLES IN EACH CLASS, ORIGINAL OTUS, AND NODES IN THE TREE FOR MULTICLASS DATASETS

	Class Samples	OTUs	Nodes
Obesity	164, 114, 89	465	872
IBD	38, 60, 76, 44, 59, 41, 44, 89	301	412
Multi-Disease	488, 118, 232, 164, 89, 21, 4, 48, 26, 13	772	1318
Syn3	224, 255, 246	500	992
Syn5	224, 255, 246, 272, 240	500	971
Syn7	224, 255, 246, 272, 240, 246, 250	500	978
Syn9	224, 255, 246, 272, 240, 246, 250, 260, 269	500	978

taxonomic level was used. A summary of these datasets is shown in Table I.

We also evaluated PopPhy-CNN using three multiclass datasets. In the first dataset, we included samples in the obesity between 25kg/m^2 and 30kg/m^2 as a third class. The second multiclass dataset was constructed by grouping the six IBD binary datasets into a single dataset containing seven classes. The third dataset was the combination of the cirrhosis, T2D, and obesity datasets as well as a colorectal cancer dataset [32] and another IBD dataset [33], both contained within the MetaML package, resulting in a dataset with 10 different classes. Additionally, using the R package SparseDOSSA [34], we constructed synthetic datasets containing 3, 5, 7 and 9 classes. Each synthetic dataset contained 500 features and about 250 samples per class. A summary of the these datasets is shown in Table II. The visualization of the IBD, Multi-Disease, and Syn9 datasets using Principal Coordinate Analysis (PCoA) based on the Bray-Curtis dissimilarity as the distance metric is shown in the Appendix (Fig. S1).

Lastly, we construct two binary synthetic datasets in order to evaluate the robustness of PopPhy-CNN. The smaller dataset (SynA) contains 750 samples and 500 features. The larger dataset (SynB) contains 1500 samples and 1000 features.

III. RESULTS

A. PopPhy-CNN is Competitive in Host Phenotype Prediction

PopPhy-CNN was benchmarked against RF, SVM, LASSO, an MLPNN with two fully connected layers, a 1D-CNN model using one convolutional layer with two fully connected layers, and Ph-CNN, which was designed using information of the phylogenetic tree. The 1D-CNN model serves as the baseline to evaluate if the addition of phylogenetic information improves the prediction in CNN. In addition, we compared PopPhy-CNN to Ph-CNN [20] for the datasets in which the coordinates for the method were available. Each model was trained using 10-fold cross validation, using the same partitions across all methods. The area under the receiver operating characteristic curve (AUC-ROC), area under the precision-recall curve (AUC-PR), Matthews correlation coefficient (MCC), and F1-Score are reported.

In order to train CNNs under cross validation efficiently, each network was trained using early stopping. To do so, 20% of the training set was set aside for a validation holdout set and the loss for this set was calculated each epoch. Each model was trained until the loss on the validation set had not decreased for 100 consecutive epochs, and the previous best weights were restored. The final model was then evaluated on 10% of the data that was set aside for a blind test. The learning rate, number of kernels, hidden layer size, and the regularization parameters on network weights were tuned by doubling the values until a drop in performance was observed. The number of kernels and hidden layer size started at 4 and the learning rate and regularization parameters started at 0.00001. The kernel sizes tested for PopPhy-CNN were $5 \times 3, 4 \times 3$, and 3×3. The kernel sizes for the 1D-CNN were the same width as the kernels used in PopPhy-CNN. In order to obtain stable results, an ensemble of 10 networks were trained for each partition and the mean predictions across the models was used for the final prediction.

Before evaluating all datasets, we first used the cirrhosis dataset to confirm that there is no significant difference on the performance when using two different representations of the populated tree. We also evaluated different tree matrix representation schemes by using 0 padding and -1 padding to align children starting directly under the parent node and found no significant difference either.

RF, SVM, and LASSO were trained using Python's scikit package. In RF training a maximum of 200 trees was set and all other parameters were left as the default. The SVMs were trained using a grid search 5-fold cross validation over the linear and Gaussian kernels with an exhaustive search using the set 1, 10, 100, 1000 for error terms and the set 0.001, 0.0001 for γ values in Gaussian kernels. The LASSO model was trained using iterative fitting of the error term α using the set of 50 numbers from 10^{-4} to $10^{-0.5}$ that were spaced evenly on a log-scale. The best model parameters were again evaluated using 5-fold cross validation. For each model, the data were min-max normalized between 0 and 1.

The summary of the benchmarking results for datasets aggregated at the genus level and species level are shown in Table III and Table IV, respectively. The standard ML methods were trained using the original OTU features while PopPhy-CNN and Ph-CNN were trained using their respective input formats. We observe that PopPhy-CNN outperforms Ph-CNN and is comparable to the other methods with RF generating slightly better prediction. We also benchmarked PopPhy-CNN by using a vector containing the values from all the nodes from a populated tree as input to other methods. The performance of RF is reduced, suggesting it may have difficulty in taking advantage

TABLE III
THE AUC-ROC, AUC-PR, MCC, AND F1-SCORE VALUES FROM LASSO, RF, SVM, MLPNN, AND 1D-CNN MODELS ARE REPORTED FOR ALL BINARY CLASS DATASETS AT THE GENUS LEVEL. THE VALUES FOR PH-CNN ARE REPORTED IN THE DATASETS IN WHICH COORDINATES WERE AVAILABLE TO PERFORM THE METHOD

		PopPhy-CNN	RF	SVM	LASSO	MLPNN	1D-CNN	Ph-CNN
Cirrhosis	AUC-ROC	0.901	**0.928**	0.888	0.872	0.861	0.898	-
	AUC-PR	0.914	**0.927**	0.899	0.886	0.875	0.913	-
	MCC	0.610	**0.731**	0.568	0.548	0.568	0.695	-
	F1-Score	0.798	**0.858**	0.772	0.757	0.776	0.841	-
T2D	AUC-ROC	0.681	**0.718**	0.510	0.659	0.645	0.666	-
	AUC-PR	0.692	**0.737**	0.555	0.671	0.658	0.673	-
	MCC	0.231	**0.297**	-0.024	0.246	0.185	0.204	-
	F1-Score	0.611	**0.643**	0.459	0.614	0.586	0.595	-
Obesity	AUC-ROC	0.589	**0.627**	0.568	0.493	0.563	0.571	-
	AUC-PR	0.414	0.476	0.457	**0.637**	0.431	0.478	-
	MCC	**0.181**	0.079	0.008	-0.014	0.081	0.078	-
	F1-Score	**0.587**	0.558	0.524	0.508	0.529	0.529	-
CDr	AUC-ROC	0.799	**0.897**	0.878	0.828	0.805	0.837	0.714
	AUC-PR	0.895	**0.953**	0.947	0.924	0.912	0.923	-
	MCC	0.433	0.562	0.571	0.312	0.427	0.494	0.241
	F1-Score	0.726	0.796	**0.804**	0.677	0.735	0.768	0.756
CDf	AUC-ROC	0.926	**0.982**	0.931	0.931	0.932	0.940	0.808
	AUC-PR	0.957	**0.990**	0.967	0.965	0.965	0.971	-
	MCC	0.706	0.758	**0.790**	0.700	0.744	0.783	0.630
	F1-Score	0.847	0.875	**0.888**	0.844	0.867	**0.888**	0.836
iCDr	AUC-ROC	0.866	**0.898**	0.879	0.844	0.858	0.852	0.768
	AUC-PR	0.92	**0.948**	0.934	0.884	0.923	0.920	-
	MCC	0.613	0.640	**0.647**	0.614	0.611	0.609	0.556
	F1-Score	0.812	**0.822**	0.820	0.808	0.800	0.787	0.731
iCDf	AUC-ROC	0.950	**0.968**	0.848	0.951	0.951	0.959	0.842
	AUC-PR	0.958	**0.978**	0.900	0.962	0.965	0.970	-
	MCC	**0.851**	0.842	0.609	0.830	0.805	0.746	0.704
	F1-Score	0.917	**0.918**	0.764	0.904	0.887	0.853	0.845
UCr	AUC-ROC	0.855	**0.903**	0.740	0.828	0.837	0.841	0.722
	AUC-PR	0.843	**0.931**	0.759	0.863	0.824	0.820	-
	MCC	**0.696**	0.656	0.601	0.551	0.579	0.654	0.445
	F1-Score	**0.837**	0.821	0.770	0.752	0.785	0.817	0.745
UCf	AUC-ROC	0.946	0.960	0.480	0.920	**0.969**	0.935	0.822
	AUC-PR	0.957	**0.975**	0.632	0.945	0.972	0.943	-
	MCC	0.688	**0.812**	0.303	0.657	0.749	0.770	0.668
	F1-Score	0.829	**0.896**	0.532	0.806	0.856	0.871	0.825

TABLE IV
THE AUC-ROC, AUC-PR, MCC, AND F1-SCORE VALUES FROM LASSO, RF, SVM, MLPNN, AND 1D-CNN MODELS ARE REPORTED FOR ALL BINARY CLASS DATASETS AT THE SPECIES LEVEL

		PopPhy-CNN	RF	SVM	LASSO	MLPNN	1D-CNN
Cirrhosis	AUC-ROC	**0.946**	0.943	0.924	0.902	0.908	0.936
	AUC-PR	**0.947**	0.939	0.932	0.908	0.910	0.933
	MCC	0.727	**0.763**	0.686	0.619	0.640	0.725
	F1-Score	0.857	**0.876**	0.835	0.797	0.817	0.856
T2D	AUC-ROC	0.690	**0.737**	0.425	0.654	0.685	0.659
	AUC-PR	0.690	**0.742**	0.480	0.637	0.692	0.665
	MCC	0.256	**0.310**	0.026	0.264	0.251	0.201
	F1-Score	0.620	**0.647**	0.398	0.624	0.620	0.591
Obesity	AUC-ROC	0.666	**0.683**	0.601	0.512	0.635	0.645
	AUC-PR	0.518	0.533	0.421	**0.624**	0.489	0.481
	MCC	**0.227**	0.084	0.079	-0.007	0.206	0.214
	F1-Score	0.621	0.553	0.564	0.514	**0.623**	**0.623**

TABLE V
THE MCC VALUES FROM POPPHY-CNN, RF, MLPNN, AND 1D-CNN MODELS ARE REPORTED FOR MULTI-CLASS DATASETS AT THE SPECIES LEVEL

	PopPhy-CNN	RF	MLPNN	1D-CNN
Obesity (3)	**0.159**	0.089	0.048	0.086
IBD (7)	**0.158**	0.073	0.114	0.149
Multi-Disease (10)	**0.343**	0.316	0.314	0.297

TABLE VI
THE MCC VALUES FROM POPPHY-CNN AND RF FOR SYNTHETIC MULTI-CLASS DATASETS OF VARYING NUMBER OF CLASSES AT THE SPECIES LEVEL

	# of Classes	PopPhy-CNN	RF
Syn3	3	**0.884**	0.814
Syn5	5	**0.871**	0.712
Syn7	7	**0.863**	0.650
Syn9	9	**0.835**	0.583

of information provided by the additional internal nodes in the phylogenetic trees (data not shown).

Next, we evaluated our model in a multiclass setting. We benchmarked our model against a multiclass instance of RF, MLPNN, and a 1D-CNN using both the original OTU features as well as the set of nodes in the populated tree as features. We observed that PopPhy-CNN performed the best on biological

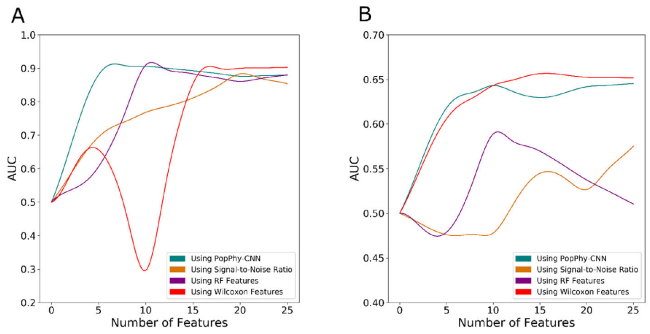

Fig. 4. Benchmarking of top 25 features extracted from PopPhy-CNN for cirrhosis (A) and obesity (B). Features extracted from PopPhy-CNN (teal) are benchmarked against features found by RF (purple), signal-to-noise ratio (brown), and a Wilcoxon rank-sum test (red).

data. When using synthetic data, the neural network models all performed similarly and were robust to the number of classes. Since features in the synthetic datasets were randomly spiked, we did not expect PopPhy-CNN to benefit from considering local areas in the input matrices. On the other hand, a noticeable decrease in RF models as the number of classes increased was observed.

In summary, PopPhy-CNN is competitive to the other standard ML models without requiring a large amount of training data; using hierarchical features in other methods did not show an overall improvement. In addition, when expanding to multi-class data, PopPhy-CNN did not suffer from larger numbers of classes, suggesting its strength in establishing better predictive models on datasets containing multiple disease states.

B. Computational Complexity and Robustness

To evaluate the complexity of our model, we recorded the amount of time it took to train a single model using different numbers of sample and feature sizes. To do so, we created synthetic datasets with 500, 1000, 2500, 5000, and 10,000 samples, each with 500 features. Additionally, we created datasets with 500, 750, 1000, 1250, and 1500 features, each with 500 samples. The average training for training a single model during a 10-fold cross-validation is reported in Fig. 4. For consistency, all neural network models were trained to 50 epochs using an NVIDIA Titan XP GPU, which was sufficient for accurate prediction in each case. We observed that RF models had the largest overhead and that SVM models scaled the worst based on both sample and feature size. PopPhy-CNN increased more than the other neural network models based on sample size, but that was expected due to the increased input space of the matrix representations of the populated trees. Despite this, it was still observed to train faster than the RF models.

Next, we evaluated how the parameter size of the neural network models scaled based on the original input size. We observed that MLPNN and CNN-1D models scaled almost identically and that PopPhy-CNN scaled at a rate 5.08 times faster than the other two models (Fig. S2). However, this was expected since the matrix representation used in PopPhy-CNN was shown to scale in size on average 4.93 times the number of nodes in the tree.

Lastly, we tested the robustness of PopPhy-CNN using 5-fold and 3-fold cross-validation for larger heldout sets using the cirrhosis dataset as well as two synthetic datasets, SynA and SynB. When holding out 20% for testing, the AUC-ROC for cirrhosis was 0.916 (2.66% decrease), for SynA was 0.928 (0.24% decrease), and for SynB was 0.927 (0.24% decrease). When using 33% as held out data, the AUC-ROC for cirrhosis was 0.917 (2.66% decrease), for SynA was 0.900 (2.92% decrease), and for SynB was 0.904 (2.78% decrease). Together, this shows that PopPhy-CNN is robust to using larger sets of held-out data even for datasets with moderate size.

C. PopPhy-CNN Identifies Important Features

We used the top three datasets in Table I at the genus level for this evaluation. Feature scores for each dataset were generated following the procedure outlined in the Materials and Methods section. In the results shown, we used $\theta_1 = 0.01$ and $\theta_2 = 0$, indicating that we consider only the top 1% of values in each feature map of each sample. This allows a fair baseline comparison across the datasets from which the tuning of the parameters may lead to stronger feature evaluations. We constructed a single ranked list using the feature scores. The method for constructing the joint ranked list is described in the Appendix.

To evaluate the informativeness of the extracted features, we examined whether they could be used in building better prediction models in SVM. This is because SMV is the only model that does not have any feature selection capacity in our evaluation. To do this, we trained SVM models using the top ranked features from the original OTUs ranging from the top 5, 10, 15, 20, and 25.

For comparisons, we used ranking lists based on signal-to-noise ratio, the significance from the Wilcoxon test, as well as the average feature rankings from the RF models that achieved an AUC score greater than the average AUC over the 10 times 10-fold cross validated training. We chose not to use differential abundance analysis methods for feature ranking due to the fact that the abundance values were normalized as relative abundance and no longer followed a negative binomial distribution. The SVM models were trained in the same way as described in the model evaluation.

For the cirrhosis dataset (Fig. 4A), we observed that the higher ranked features of PopPhy-CNN performed best, followed by the features identified by RF. The features identified by the Wilcoxon rank-sum test were not stable and showed a decrease in prediction performance before increasing afterwards. In the obesity dataset, we observe that PopPhy-CNN and the Wilcoxon rank-sum features perform similarly, however the RF features perform poorly (Fig. 4B). For the T2D dataset, all models performed about the same (Fig. S3). PopPhy-CNN was the only method to perform competitively in all three datasets. Additionally, PopPhy-CNN captured unique OTUs and we observed little overlap between the OTUs captured by the three methods (Fig. S4).

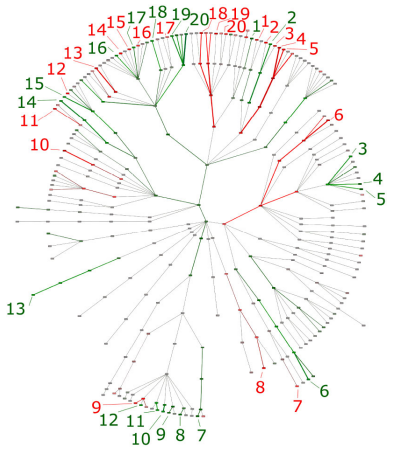

Fig. 5. Visualization of the cirrhosis features found by PopPhy-CNN. An annotated phylogenetic tree from the cirrhosis dataset shows subtrees found important in the cirrhosis patients (red) as well as the healthy subjects (green). The table highlights nodes from the tree who are leaves of their annotated subtree. The top 5 features in each class are shown in bold.

D. Biological Relevance of the Extracted Features: A Case Study of the Cirrhosis Dataset

In the cirrhosis patients, *Veillonella*, *Streptococcus*, *Haemophilus*, *Prevotella*, and *Actinomyces* were found important. In the healthy subjects, *Alistipes*, *Rumminococcus*, *Roseburia*, *Clostridium*, and *Bilophila* were found to be the most important features. Many of the top ranked features were also identified in the original study [27]. A separate study on a different cohort of subjects with Cirrhosis found similar results, showing that *Streptococcus*, *Veillonella*, and *Prevotella* were associated with Interleukin-23 (IL-23) and Interleukin-2 (IL-2), two cytokines which have been shown to be associated with inflammatory gut diseases [35], [36]. In addition, among the identified features, *Veillonella* and *Lactobacillus* have been shown in previous studies to be correlated with the mortality rate of subjects with cirrhosis, while *Clostridium* may be protective against cirrhosis mortality [37], [38].

When analyzing the scores of internal nodes of the tree, we observed cases in which ancestral nodes have much larger importance scores than their children. This could imply that no single child feature was discriminative between disease states, however the collection of them was. For example, the family Bifidobacteriaceae, had a score of 0.292 while its children had much lower scores. This family of microbes was not identified as important in the original analysis of this dataset, however, a different study has shown that microbes in the Bifidobacteriaceae family produce glutamate dehydrogenase, a protein found to have higher expression levels in patients with Cirrhosis [39]. Therefore, the aggregation of all the genera under Bifidobacteriaceae should be more discriminative than any single genus observed in our feature analysis.

E. Visualization of Extracted Features

We used Cytoscape [40] to visualize the phylogenetic tree and annotate the nodes and edges based on the calculated importance scores, the score differences used in creating the joint rank list as the annotated scores for each node. Nodes and edges were then colored based on which phenotype they were associated with where green represents healthy and red represents disease. The edges are colored in a similar way based on the average score of the connected nodes. This visualization can facilitate the interpretation of extracted features in the context of the phylogenetic tree. The annotated tree for the cirrhosis dataset shows that the feature extraction not only capture OTUs presented in the original datasets, but the ancestral nodes as well (Fig. 5).

IV. Conclusions

We have developed a novel CNN framework, PopPhy-CNN, for the prediction of the host disease status from a metagenomic sample of the host. PopPhy-CNN leverages biological knowledge in microbial taxa relative abundance profiles through a phylogenetic tree by our novel propagation and matrix-representation procedure. Using nine binary class metagenomic datasets, we have shown that PopPhy-CNN models are competitive compared to RF, SVMs, LASSO, 1D-CNN, MLPNN, and Ph-CNN models in benchmarking for binary classification datasets. Our evaluation establishes the evidence that PopPhy-CNN can deliver robust performance without requiring excessively large training sets. Additionally, PopPhy-CNN showed the best performance for multi-class biological and synthetic datasets and is robust with respect to the number of classes.

We have also demonstrated the feasibility of our novel procedure for retrieving informative features from the learned CNN models. Our procedure provides an unique way to interpreter the network models. We showed that SVMs with the selected feature sets performed better than SVMs trained on features ranked based on the criteria of the signal-to-noise ratio, the Wilcoxon test, and by RF models. This result is especially intriguing as it provides the evidence that the feature maps of the first convolutional layer maintain spatial relationship between the microbial taxa on the phylogenetic tree. This implies that PopPhy-CNN benefits from learning informative features on the populated phylogenetic tree represented in the matrix format, which may explain the effectiveness of PopPhy-CNN. The limitation of our current procedure of feature extraction is that we only look at the kernel map activation at the first convolutional layer for easy interpretation. This procedure may miss important features involving in complex nonlinear relationship to the phenotype.

There are several directions for further study. The phylogenetic tree is one of the core components in the PopPhy-CNN learning framework. Different trees constructed from different methods may affect the predictive performance and may also identify different microbial features. Furthermore, the current representation scheme is designed to prevent sparsity in the matrix while preserving spatial phylogenetic relationships. This can create areas where the descendant nodes are not directly under their ancestors, allowing for unique patterns to be picked up by the CNN. However, if descendant nodes are shifted far enough away from the ancestral nodes the CNN kernels may not capture them together. Therefore, different ways of representing the populated trees as matrix image may also affect the model performance. It may even be possible to expand beyond a rooted tree to a graph structure, a domain in which the

newer graph convolutional networks can be explored [41]. Also, if the number of microbial taxa substantially outnumbers that of the learning samples, more effective regularization schemes or algorithms that promote the learning of important features in CNNs are likely necessary. Another direction is to expand the feature extraction method to include deeper layers of the network, further exploiting the non-linear patterns learned by PopPhy-CNN.

Appendix

S1. Appendix of Supporting Information Availability of Data and Code

The datasets and the code used in this study can be found at https://github.com/YDaiLab/PopPhy-CNN.

Acknowledgment

The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. This work is partially supported by a UIC Chancellor's Graduate Research Fellowship, and UIC CCTS Pre-doctoral Education for Clinical and Translational Scientists fellowship (UL1TR002003) both awarded to AAM.

References

[1] J. Marchesi *et al.*, "The gut microbiota and host health: A new clinical frontier," *Gut*, vol. 65, no. 2, pp. 330–339, 2016.
[2] M. Stephen *et al.*, "The intestinal microbiome, barrier function, and immune system in inflammatory bowel disease: A tripartite pathophysiological circuit with implications for new therapeutic directions," *Therapeutic Adv. Gastroenterol.*, vol. 9, no. 4, pp. 606–625, 2016.
[3] J. Sun and E. Chang, "Exploring gut microbes in human health and disease: Pushing the envelope," *Genes Diseases*, vol. 1, no. 2, pp. 132–139, 2014.
[4] J. Qin *et al.*, "A metagenome-wide association study of gut microbiota in type 2 diabetes," *Nature*, vol. 490, no. 7418, pp. 55–60, 2012.
[5] V. Hale *et al.*, "Shifts in the fecal microbiota associated with adenomatous polyps cancer epidemiology biomarkers," *Prevention*, vol. 26, no. 1, pp. 1–10, 2017.
[6] V. Pascal *et al.*, "A microbial signature for crohn's disease," *Gut*, vol. 66, no. 5, pp. 813–822, 2017.
[7] H. Koh *et al.*, "A powerful microbiome-based association test and a microbial taxa discovery framework for comprehensive association mapping," *Microbiome*, vol. 5, no. 45, 2017.
[8] A. Metwally *et al.*, "MetaLonDA: A flexible R package for identifying time intervals of differentially abundant features in metagenomic longitudinal studies," *Microbiome*, vol. 6, no. 1, 2018, Art. no. 32.
[9] D. Knights *et al.*, "Human-associated microbial signatures: Examining their predictive value," *Cell Host Microbe*, vol. 10, no. 4, pp. 292–296, 2011.
[10] M. Finucane *et al.*, "A taxonomic signature of obesity in the microbiome? getting to the guts of the matter," *PLOS ONE*, vol. 9, no. 1, 2014, Art. no. e84689.
[11] Q. Zhang *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," *Bioinf.*, vol. 31, no. 10, pp. 1607–1613, 2015.
[12] B. Wingfield *et al.*, "A metagenomic hybrid classifier for paediatric inflammatory bowel disease," *Proc. Int. Joint Conf. Neural Netw.*, 2016, pp. 1083–1089.
[13] E. Pasolli *et al.*, "Machine learning meta-analysis of large metagenomic datasets: Tools and biological insights," *PLOS ONE*, vol. 12, no. 1, 2016, Art. no. e1004977.
[14] T. Wang and H. Zhao, "Constructing predictive microbial signatures at multiple taxonomic levels," *J. Amer. Statistical Assoc.*, vol. 112, no. 519, pp. 1022–1031, 2017.
[15] Y. Qiu *et al.*, "Infer metagenomic abundance and reveal homologous genomes based on the structure of taxonomy tree," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 12, no. 5, pp. 1112–1122, Sep./Oct. 2015.
[16] M. Oudah *et al.*, "Taxonomy-aware feature engineering for microbiome classification," *BMC Bioinf.*, vol. 19, 2018, Art. no. 227.
[17] D. Albanese *et al.*, "Explaining diversity in metagenomic datasets by phylogenetic-based feature weighting," *PLOS Comput. Biol.*, vol. 11, no. 3, 2015, Art. no. e1004186.
[18] Xiao *et al.*, "A phylogeny-regularized sparse regression model for predictive modeling of microbial community data," *Frontiers Microbiol.*, vol. 9, 2018, Art. no. 3112.
[19] G. Ditzler *et al.*, "Multi-layer and recursive neural networks for metagenomic classification," *IEEE Trans. NanoBiosci.*, vol. 14, no. 6, pp. 608–616, Sep. 2015.
[20] D. Fioravanti *et al.*, "Phylogenetic convolutional neural networks in metagenomics," *BMC Bioinf.*, vol. 19, no. (Supp2), 2018, Art. no. 49.
[21] N. LaPierre *et al.*, "Metapheno: A critical evaluation of deep learning and machine learning in metagenome-based disease prediction," *Methods*, vol. 166, pp. 74–82, 2019.
[22] D. Reiman *et al.*, "Using convolutional neural networks to explore the microbiome," *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2017, pp. 4269–4272.
[23] Y. LeCun *et al.*, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[24] PhyloT, "Phylot: A tree generator," [Online]. Available: http://phylot.biobyte.de/
[25] N. Srivastava *et al.*, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.
[26] B. Athiwaratkun and K. Kang, "Feature representation in convolutional neural networks," 2015, *arXiv:1507.02313v1*.
[27] N. Qin *et al.*, "Alterations of the human gut microbiome in liver cirrhosis," *Nature*, vol. 513, no. 7516, pp. 59–64, 2014.
[28] F. Karlsson *et al.*, "Gut metagenome in european women with normal, impaired and diabetic glucose control," *Nature*, vol. 498, no. 7452, pp. 99–103, 2013.
[29] E. Le Chatelier *et al.*, "Richness of human gut microbiome correlates with metabolic markers," *Nature*, vol. 500, no. 7464, pp. 541–546, 2013.
[30] D. Truong *et al.*, "Metaphlan2 for enhanced metagenomic taxonomic profiling," *Nature Methods*, vol. 12, no. 10, pp. 902–903, 2015.
[31] H. Sokol *et al.*, "Fungal microbiota dysbiosis in IBD," *Gut*, vol. 66, pp. 1039–1048, 2016.
[32] G. Zeller *et al.*, "Potential of fecal microbiota for early-stage detection of colorectal cancer," *Mol. Syst. Biol.*, vol. 10, no. 11, 2014, Art. no. 766.
[33] J. Qin *et al.*, "A human gut microbial gene catalogue established by metagenomic sequencing," *Nature*, vol. 464, no. 7285, pp. 59–65, 2010.
[34] B. Ren, E. Schwager, T. Tickle, and C. Huttenhower, "sparseDOSSA Sparse data observations for simulating synthetic abundance." *R package version 1.12.0*.
[35] J. Bajaj *et al.*, "Linkage of gut microbiome with cognition in hepatic encephalopathy," *Amer. J. Physiol. Gastrointest Liver Physiol.*, vol. 302, no. 1, pp. G168–G175, 2011.
[36] E. Duvallet *et al.*, "Interleukin-23: A key cytokine in inflammatory diseases." *Ann. Med.*, vol. 43, no. 7, pp. 503–11, 2011.
[37] C. Sung *et al.*, "Predicting clinical outcomes of cirrhosis patients with hepatic encephalopathy from the fecal microbiome," *Cellular Mol. Gastroenterol. Hepatol.*, vol. 8, no. 2, pp. 301–308.e2, 2019.
[38] A. Horvath *et al.*, "Intestinal colonisation by veillonella spp. is predictive for mortality in stable cirrhosis and could be partially reduced by a multispecies probiotic in a randomized placebo controlled trial," *J. Hepatol.*, vol. 66, no. 1, 2017, Paper S128.
[39] X. Wei *et al.*, "Cirrhosis related functionality characteristic of the fecal microbiota as revealed by a metaproteomic approach," *BMC Gastroenterol.*, vol. 16, no. 1, 2016, Art. no. 121.
[40] P. Shannon *et al.*, "Cytoscape: A software environment for integrated models of biomolecular interaction networks," *Genome Res.*, vol. 13, no. 11, pp. 2498–2504, 2003.
[41] M. Defferrard *et al.*, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.