Multi-Layer and Recursive Neural Networks for Metagenomic Classification

Gregory Ditzler, *Member, IEEE,* Robi Polikar, *Senior Member, IEEE,* and Gail Rosen, *Senior Member, IEEE*

***Abstract*—Recent advances in machine learning, specifically in deep learning with neural networks, has made a profound impact on fields such as natural language processing, image classification, and language modeling; however, feasibility and potential benefits of the approaches to metagenomic data analysis has been largely under-explored. Deep learning exploits many layers of learning nonlinear feature representations, typically in an unsupervised fashion, and recent results have shown outstanding generalization performance on previously unseen data. Furthermore, some deep learning methods can also represent the structure in a data set. Consequently, deep learning and neural networks may prove to be an appropriate approach for metagenomic data. To determine whether such approaches are indeed appropriate for metagenomics, we experiment with two deep learning methods: (i) a deep belief network, and (ii) a recursive neural network, the latter of which provides a tree representing the structure of the data. We compare these approaches to the standard multi-layer perceptron, which has been well-established in the machine learning community as a powerful prediction algorithm, though its presence is largely missing in metagenomics literature. We find that traditional neural networks can be quite powerful classifiers on metagenomic data compared to baseline methods, such as random forests. On the other hand, while the deep learning approaches did not result in improvements to the classification accuracy, they do provide the ability to learn hierarchical representations of a data set that standard classification methods do not allow. Our goal in this effort is not to determine the best algorithm in terms accuracy – as that depends on the specific application – but rather to highlight the benefits and drawbacks of each of the approach we discuss and provide insight on how they can be improved for predictive metagenomic analysis.**

I. Introduction

METAGENOMICS is the study of DNA from micro-organisms obtained directly from an environmental sample [1]. In a metagenomic study, a sample is collected from the environment, which can be a gram of soil [2], [3], a milliliter of ocean water [4], or a swab from an object [5] including any living organism such as humans [5], [6]. DNA from this sample is then isolated and sequenced resulting in thousands of DNA reads generated from a next-generation sequencer. This process leads to several questions that need to be answered by biologists to draw a meaningful conclusion from the sample: "who (which organisms) are here in this sample?", "how much of each is here?", and – particularly as more whole genome shotgun and transcriptomic data are being sequenced – "what are they doing?" [7]–[10]. Answering these questions allows us to determine the identity, abundance and the functionality of the organisms found in the sample. Algorithms from machine learning and data science allow us to begin to make sense of the data, and answer questions about the data collected. For example, researchers are interested in the human gut, and how its microbes develop [11], and change over time [5], [6].

Machine learning focuses on the mathematical and algorithmic development of methods that can learn the underlying statistical characteristics of the data, and make predictions on the values (regression), or categories (classification) of future data, typically in an unsupervised or supervised setting. In a supervised setting, the objective is to find a function $f : \mathcal{X} \mapsto \mathcal{Y}$ that provides good generalization in predicting future data, where $\mathcal{X}$ is the feature space of prediction variables, and $\mathcal{Y}$ is the discrete label space. Hence, the function f learns the mapping between the feature space $\mathcal{X}$ and the outcome space $\mathcal{Y}$. Conversely, in an unsupervised setting, where labels of the training data are not available, the objective is to determine a natural clustering – or grouping – of the data based on the information only in the feature space $\mathcal{X}$. Both settings are commonly observed in metagenomics. For example, recent works have examined supervised learning of "microbiome phenotypes" [12], where labeled data are available; and other recent works have examined using unsupervised learning for developing a binning algorithms for clustering sequences with a high level of similarity [13]. The primary advantage of machine learning is that once f is learned from training data, it can be used to predict the classification of future data – whose labels are unknown. While there are also semi-supervised approaches that attempt to learn from both labeled and unlabeled data, there is also a new cadre of algorithms that have been developed that first identify the natural groupings in the data by exploiting several different layers of feature representations in an unsupervised manner, followed by supervised combinations of such representations to learn the label associations. Learning in such a mechanism is known as *deep learning*. Deep learning is a relatively new field in machine learning and has seen great success in areas of image classification, natural language processing, and signal processing [14], [15]. Specifically, deep learning uses algorithms to learn multiple levels of data representations to model complex relationships among data.

The typical pipeline for classification involves pre-processing stages as well as feature extraction or feature selection prior to the learning phase, as shown in Figure 1 [16]. *Feature extraction* typically involves a mathematical transformations applied to the raw data, providing a projec-

G. Ditzler and G. Rosen are with the Department of Electrical & Computer Engineering, Drexel University, Philadelphia, PA 19104.

R. Polikar is with the Department of Electrical & Computer Engineering, Rowan University, Glassboro, NJ 08028.

Author emails: gregory.ditzler@gmail.com, polikar@rowan.edu, gailr@ece.drexel.edu.

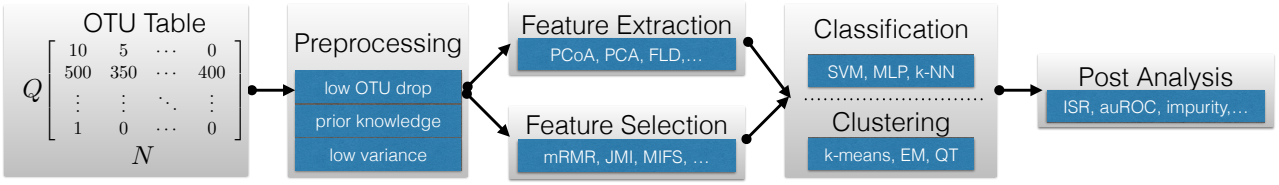


Fig. 1. Typical pipeline for pattern classification problems for the classification of metagenomic samples. A community data matrix, also known as an operational taxonomic unit (OTU) table, is constructed after representative sequences have been classified and counted for each sample site. After the OTU table is determined, pre-processing is performed followed by, possibly, a feature extractor before classification is performed.

tion to a new feature space where the predictors are more informative than the ones in the original raw data. When prior knowledge is available, heuristics and/or hand picking a subset of the features known to be more informative can also be used, leading to *feature selection*. We explore the ability of deep learning and neural networks to: (i) learn new features derived from the raw data, and (ii) make predictions on new unlabeled data. A distinct advantage of deep learning, compared to more traditional approaches is that deep learning allows us to perform (i) and (ii) simultaneously. To the best of our knowledge, this is the first effort in determining the efficacy of deep learning for metagenomic applications.

In this contribution, we first describe neural network-based approaches, for (i) representing samples in a manner similar to the hierarchical representations of Unweighted Pair Group Method with Arithmetic Mean (UPGMA) trees, and (ii) classification of environmental metagenomic samples. We then benchmark the deep learning and traditional neural networks against the widely used random forest algorithm. We also demonstrate the general advantages to using traditional neural networks for environmental sample classification of metagenomic data. Our primary goal is not to demonstrate the superiority of any single approach, but rather to explore feasibility, advantages and disadvantages of deep learning methods when applied to metagenomics problem, and also to introduce metagenomic researchers to a new machine learning tool that may be beneficial in solving a wide spectrum of metagenomic data analysis problems.

II. Metagenomic Data & Deep Learning Methods

A. Obtaining a Vector Representation of the 16S rRNA Sequences

Metagenomic sequences, or short reads, are a collection of DNA sequence fragments collected directly from an environmental sample, which may contain thousands of microorganisms, many of which have not had their genomes yet cultured or sequenced. We consider a sample to be a set of reads obtained from a round of sequencing. For example, all bacterial DNA sequences collected from an individual's fingertips is considered a sample. Thousands of reads from the 16S rRNA gene (commonly selected as it is highly conserved across bacteria) are typically collected from each observation (i.e., sample site). Before machine learning methods can be applied, the sequences need to be converted into a vector representation of a set of features $\mathbf{x} \in \mathbb{R}^d$, believed to carry information for classification.

There are many such pipelines that can be implemented to perform the conversion from sequences to vectors in $\mathbb{R}^d$. One such approach would be to begin by removing short and low quality score reads from the entire collection of sequences. The remaining sequences are then clustered, using an algorithm such as CD-HIT, which forms a representative sequence for each cluster [13], [17]. Representative sequences are aligned using NAST [18] and grouped into taxonomic classifications using tools such as Ribosomal Database Project's (RDP) naïve Bayes classifier (NBC) [19]. Each sample consists of thousands of reads that are classified into taxa or operational taxonomic units (OTUs) then each sample can be represented as a vector $\mathbf{x} \in \mathbb{N}_+^d$, where d is the number of different taxa in the database and $\mathbb{N}_+$ is the set of positive integers. The ith entry of $\mathbf{x}$ is the abundance of taxa i, or simply the number of times the ith OTU was detected in the sample. Other features, such as k-mers can also be considered; however, using k-mers leads to an extremely high dimensional space ($d = 4^k$), whereas the OTU features lie in a much smaller feature space.

B. Finding Meaning in Supervised Classification

Supervised classification schemes are not needed for every metagenomic study; however, many studies can benefit from a supervised model. The obvious benefit of a supervised classifier is that it can classify unlabeled sequences to taxonomic units, or function if the application needs it. For example, the naïve Bayes classifier has been used by biologists for classifying sequences into an OTU using k-mer frequencies [19], [20]. In addition to predicting a class label for unlabeled observations, supervised classification approaches can also indicate class separability, which provides insights into the quality of the data used for classification. For example, the error – or loss – of a classifier can indicate the degree of separability between the OTU abundance representations as described in Section II-A for multiple groups (i.e., phenotypes) in a metagenomic study. A low error (loss) may indicate a high degree of separability, whereas a high error (loss) typically indicates a low degree of separability between multiple groups.

C. Developing a Graphical Representation of Metagenomic Data

Determining the structure of a microbial community provides important information for understanding the relationships between different structural groups in the data set, because such a structure allows for a graphical representation

of how the samples compare to each other. Representing the structure in the data can be obtained, for example, by generating a tree using UPGMA, which employs a distance metric to measure the distances between the samples in the data set. Popular distance measures in metagenomics include UniFrac [21], Hellinger, and Bray-Curtis, though, UniFrac is the most commonly used metric in metagenomic studies [5], [6], [22]. The resulting pairwise distances are then used by UPGMA to build a tree, which represents the hierarchical structure in the data. However, there are several concerns worth mentioning about the representation of data using UPGMA. First, the nodes in the tree developed by UPGMA represent a merger of the distances of all the nodes that fall below the node under examination, which collapses the information in the features to a single statistic. Second, the tree structure is a simple representation and is not *learned* from representations of the data. In this work, we present methods to develop a tree like structure for metagenomic data sets that is learned using neural networks.

D. Neural Networks and Deep Learning

Deep learning implements several layers of feature extraction such that the model learns a better feature representation for prediction [23], [24]. Deep learning offers distinct advantages over other supervised learning methods with respect to optimization and overfitting. From an optimization perspective, deep neural networks can converge to a better local optima by using pre-training rather than simply using backpropagation. From an overfitting perspective, the unsupervised pre-training learns solely from data (i.e., labels are initially ignored), and then uses the labeled information at the end to fine–tune the weights of the entire neural network to learn the label associations [25]. The key components that facilitate deep learning include a structured network that contains non-linear layers for learning, and a learning algorithm that works at multiple layers to extract and learn features that can lead to lower generalization loss. In this section, we discuss the multi-layer perceptron and two deep learning neural network architectures.

1) Multi-layer Perceptron Neural Networks: The multi-layer perceptron neural network (MLPNN) is one of the most popular forms of artificial neural networks, and have been extremely well established and studied in the literature [26]. A MLPNN is a highly interconnected feed-forward network with weights attached to each edge in the network, which allows the network to form a mapping between the input/feature space $\mathcal{X}$ and the outcome space $\mathcal{Y}$. The shallow MLPNN used in this work consists of a projection layer connecting the input features to a single hidden layer, which is associated with a non-linear activation function. The output layer contains another projection layer with a softmax activation function applied to the output of the network. A "deep" MLPNN, on the other hand, can include many hidden layers, as shown in Figure 2. The weights of the network can be optimized by using stochastic backpropagation (i.e., gradient descent). However, training a deep MLPNN with backpropagation is computationally inefficient and expensive, rendering the need

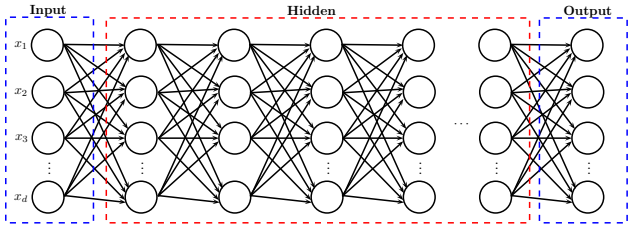


Fig. 2. Deep neural network architecture consisting of many stacked hidden layers. The number of input node sare controlled by the dimensionality of the data and the number of nodes on the output layer is controlled by the number of classes in the learning problem.

to use a deep belief network for learning a large neural network. In addition to computational complexity, another concern with backpropagation for training deep networks is that the error gradient progressively diminishes as it is back-propogated through the network, causing the backpropagation algorithm quickly reach a poor local minimum (see [27]).

2) Deep Belief Networks: Deep belief networks (DBN) are artificial neural networks that contain several hidden layers (typically three or more) with many nodes in each layer. Hinton et al. introduced a method for training such deep networks though a greedy optimization procedure using restricted Boltzmann machines (RBM) [28], where each layer of the network is trained one layer at a time. Other types of deep neural networks can be obtained by swapping the RBM layers with an auto-encoder; in which case the classifier is referred to simply as a deep neural network as opposed to deep belief network, a term reserved for using the RBM for training.

An RBM is a stochastic neural network for learning a probability distribution over a set of training vectors. The RBM has an input $\mathbf{x}$ and a hidden layer $\mathbf{h}$ (that also serves as the output), which are connected with a set of weights W. An energy function for the RBM is defined as:

$$E(\mathbf{x}, \mathbf{h}) = -\mathbf{a}^{\mathsf{T}}\mathbf{x} - \mathbf{b}^{\mathsf{T}}\mathbf{h} - \mathbf{h}^{\mathsf{T}}W\mathbf{x} \tag{1}$$

where $\mathbf{a}$ and $\mathbf{b}$ are a set of bias weights for the input and hidden nodes, respectively. Using the energy function, the joint probability distribution of the variables $\mathbf{x}$ and $\mathbf{h}$ is given by:

$$p(\mathbf{x}, \mathbf{h}) = \frac{1}{Z}\mathrm{e}^{-E(\mathbf{x},\mathbf{h})} \tag{2}$$

where Z is a normalization constant. The RBM learns a probability distribution over a set of training vectors, therefore, we can find the marginal distribution

$$p(\mathbf{x}) = \frac{1}{Z}\sum_{\mathbf{h}}\mathrm{e}^{-E(\mathbf{x},\mathbf{h})} \tag{3}$$

from which, for a data set $\mathcal{D}$, we obtain:

$$W^* = \arg\max_{W}\prod_{\mathbf{x}\in\mathcal{D}} p(\mathbf{x}) \tag{4}$$

where W^* is the optimal solution to the problem. Each RBM layer is trained sequentially, in a greedy fashion, using contrastive divergence [29].

Finally, the output layer is added to the network and standard backpropagation is used – only once – to fine-tune all of the weights of the network.

3) Recursive Neural Networks (RNN): The DBNs described in Section II-D2 are conceptually easy to understand; however, there are other deep learning methods that can provide additional insight. Socher et al. present a deep learning approach for parsing natural scenes and language using recursive neural networks, which allow for structure prediction [14]. The structure predicted by the approach can be represented as a tree for visualization purposes. In this section, we simply present an overview of their approach, as it is used in our comparative experiments. An in-depth discussion of the recursive neural network can be found in [14].

The model for deep learning presented by Socher et al. requires a data set containing features $\mathbf{x}_n$, labels l_n (e.g., low/medium/high pH level), and an adjacency matrix, A (let $n \in [N]$ be the indices of samples in the data set). The N observations in the OTU table contain C classes such that $\mathcal{Y} = \{\omega_1, \ldots, \omega_C\}$ is the set of class labels available for learning. The symmetric adjacency matrix A is required for learning the parameters θ of the neural network, where $\{A\}_{ij} = 1$ if sample i and j are adjacent to each other in some respect. For example, samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are adjacent if they have neighboring pH levels, or they are samples collected from one individual on different days in a time series study.

The N samples in the data set form the leaves of the tree, where a tree is denoted by $\mathcal{T}$. The candidate child nodes of a tree are concatenated and sent through the neural network. The neural network outputs a *merging score* and a parent feature, which is a merger of the two features learned through a non-linear mapping. Mathematically, a parent is the result of a non-linear mapping applied to a concatenation of the child features, that is

$$\mathbf{p}_{i,j} = \Phi\left(W[\mathbf{c}_i; \mathbf{c}_j] + b\right) \tag{5}$$

where $[\mathbf{c}_i; \mathbf{c}_j]$ is a concatenation of child features i & j, W is a weight matrix of the neural network, b is a bias parameter of the neural network, $\Phi(\cdot)$ is continuously differentiable function, and $\mathbf{p}_{i,j}$ is the resulting parent feature. For this work, Φ was selected to be the logistic sigmoid, whose derivative can be written in terms of the original function. Samples are considered eligible for merging on the tree by examining a *merging score* learned by the neural network and – when applicable – if the samples are adjacent. Only the samples with the highest score can be merged, thus creating a binary tree. The score of the parent feature is computed using (6).

$$s_{ij} = W_{\text{score}}^{\mathsf{T}} \mathbf{p}_{ij} \tag{6}$$

The parameters W, b, and W_{score} of the neural network can be obtained by maximizing the score of a tree $t \in \mathcal{T}$,

$$\arg\max_{t \in \mathcal{T}} s\left(\text{RNN}\left(\theta, \mathbf{x}, t\right)\right)$$

where θ are all the parameters needed to compute a score s with a neural network, and $\mathcal{T}$ any possible tree generated through the merging process described above. The parameters of the neural network are optimized using limited memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) approximation [30]. The details of the optimization procedure for the RNN can be found in [14].

III. Related Work in Classification in Metagenomics

The development of next-generation sequencing technologies makes data collection easier, less expensive, and allows for large scale data analysis of a variety of studies such as those of the human microbiome, infant gut and differences between gut microbiome in twins [5], [11], [31]. In this section, we highlight several recent works that have focused on classifying samples into "microbiome phenotypes".

Knights et al. look at supervised learning algorithms as an alternative to the traditional α– and β–diversity analysis, where α is a measure of species richness in a sample and β measures species change between samples [12]. OTU tables are used directly with discrete labels, and several classifiers are evaluated on widely publicized data sets. The classifiers selected for Knights study includes random forest [32], nearest shrunken centroids [33], elastic nets [34], and support vector machines [35]. One of their primary findings is that the random forest is the top performer for nearly every benchmark tested. This is not surprising for several reasons. First, ensemble classifiers have been shown to perform well on a variety of problems including those that have too much data, too little data, high dimensional data, or noisy data [36]. Second, random forests have a built-in feature selection mechanism that can determine the most relevant features. Lui et al. developed a tool called *MetaDist* for supervised classification of metagenomic data [37]. The authors demonstrated that their approach, which is based on distance learning with k-NN and SVMs, works well for problems with small sample sizes and unbalanced classes. Lui et al.'s method provides a sparse solution, which can be viewed as a form of feature selection. Lan et al. and Ditzler et al. have recently studied the use of feature selection in a study of the gut microbiome [38], [39].

Other works with supervised classification in metagenomics include the classification of organisms given a sequence from the 16S rRNA gene [19], [20]. Furthermore, Ditzler et al. used a microbial forensics data set with several ordination methods, including kernel methods, for classification of objects touched by an individual [16].

Knights et al. provide an in depth discussion of the need for feature selection in conjunction with a classifier in [12], while a large part of their analysis is on the different classifiers (some of which have their own form of feature selection, e.g., random forests). However, feature extraction is not addressed, which leads us to address – in this work – the problems of feature extraction and classification through deep learning.

IV. Experiments

We selected several metagenomic data sets for benchmarking the MLPNNs, DBNs and RNNs presented in Section II-D. For the purpose of this work, we focus on the accuracy of different classifiers as well as the hierarchical feature representations that can be obtained by the RNN method. Here, we first describe the data sets used in our experiments followed by a description of the preprocessing of the data and design of the experiments, and finally the results obtained from these experiments.

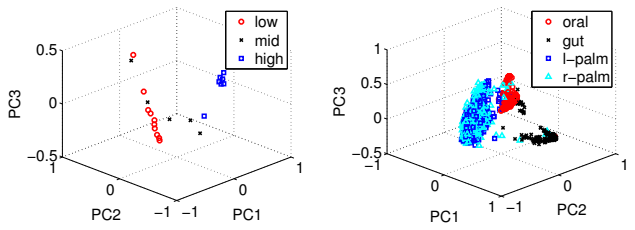


Fig. 3. Principal coordinate analysis (PCoA) plots of the (**left**) Rousk pH data [2], and (**right**) human microbiome data [5]. The Hellinger distance is used in the calculation of the coordinates. The left and right palm classes for the human microbiome data set were combined into a single class prior to the optimization of the neural networks.

A. Benchmark Data Sets

We used two real-world datasets in this study, the pH dataset from an environmental study, and the microbiome dataset for determining in which body parts do certain microbial communities live. The pH data set is comprised of arable soil measurements from 22 different sites along a pH gradient [2]. Rousk et al. found that the bacterial communities collected from various soil sites were closely defined by the pH level, and recent work has demonstrated that this is the case for pH ordering purposes [40]. The data set includes approximately 6,400 OTUs with a pH gradient given by $pH \in [4, 9]$. The samples are sorted in ascending order and assigned one of three class labels: low, mid, and high pH. We define the levels as $low \in [4.0, 4.3]$ (ten samples), $mid \in [4.73, 6.68]$ (five samples), and $high \in [7.1, 9.0]$ (seven samples). Recall that an adjacency matrix is needed by the RNN deep learning approach. The adjacency matrix, $A \in \{0, 1\}^{22\times 22}$, is formed by marking neighbors as the pH level above and below the value of pH of each sample in the data. Thus, the adjacency matrix for the pH data set is similar to that of the natural language processing problem because the adjacency for a word in a sentence is formed by marking the words before and after the word under test. The adjacency matrix is formed for both the training and testing data sets. In order to pretest the learnability of the problem, we examined a representation of the data in a 3-dimensional space. A principal coordinate analysis (PCoA), also known as multidimensional scaling [41], plot of the pH data using the Hellinger distance is illustrated Figure 3. From this figure, we observe a clear trend in the varying pH by using only the first three principal coordinates. We observe from Figure 3 that the selection of high, mid and low pH levels provides a reasonable separability between the different classes.

The second data set was collected from a study of the human microbiome. The study was carried out by Caporaso et al. over 15 months by sampling from four body sites (gut, tongue, left palm, and right palm), from two individuals [5]. The final labels we use for learning are skin (combine left/right palm), gut and tongue. We decided to combine the left and right palm samples into a single class because of the significant amount of overlap between the left and right palm samples, which is clearly shown in the PCoA plot of the human microbiome samples (see Figure 3). We also use a second version of the data set that uses the same features; however, the gender of a subject is used as the label. The microbiome data was obtained from the MG-RAST server [42], which hosts a large database of metagenomes and tools for analyzing and comparing them. The data set contains approximately 4,300 OTUs detected against the Greengenes database in 1,967 samples. The data are divided into two groups for training, one for each subject and the data are sorted based on the date of sample collection. An adjacency matrix for the RNN was determined for each person by observing the days on which the samples were collected (we use the days before and after the current sample to form adjacency).

We implemented a simple pre-processing step for each of the data set tested: the unassigned OTUs are discarded from the OTU table and only the 500 most frequently abundant OTUs are retained. Infrequent OTUs are dropped to reduce the dimensionality of the feature space. Each sample in the data is then normalized by each sample.

B. Experimental Procedure

We implemented our experiments in Matlab® and Torch7[1] using freely available deep learning implementations. The MATRBM toolbox[2] was used for the implementation of the deep belief network and Richard Socher's implementation[3] was used for the recursive neural networks. We also tested the deep learning approaches using the standard multilayer perceptron neural network, which was implemented in Torch7. We ran 10-fold cross-validation and examined different configurations of each of the network to gauge the sensitivity of the network to parameter selection. The classification error defined by the 1–0 loss function is used to assess the performance of the various classification methods. All variants of neural networks (deep or not) have a limit of 1000 training epochs to reach convergence, after which the training is stopped. The deep networks have three hidden layers with the number of hidden nodes being a variable. The number of hidden layer nodes for each layer of a neural network was varied between 250 and 500. Finally, the weights of the neural networks are initialized as normal Gaussian random variables.

For completeness of comparison, we also benchmark against random forests, as they have previously been found to perform well on several metagenomic data set benchmarks [12], [32], and in general have proven to be a very powerful classifier across a broad spectrum of problems [43]. The random forest implementation generates 25 decision trees using 25 random features selected for possible splits at nodes of the tree. Specific implementation details can be found in [32].

C. Results

1) Structure Learning of a pH Gradient: We begin by evaluating the trees that are produced by the recursive neural network with 50 hidden layer nodes using Rousk et al.'s pH gradient data. Figure 4 shows four trees generated by the RNN from the pH data hold-out sets during cross-validation. The

[1]http://www.torch.ch/
[2]http://code.google.com/p/matrbm/
[3]http://www.socher.org

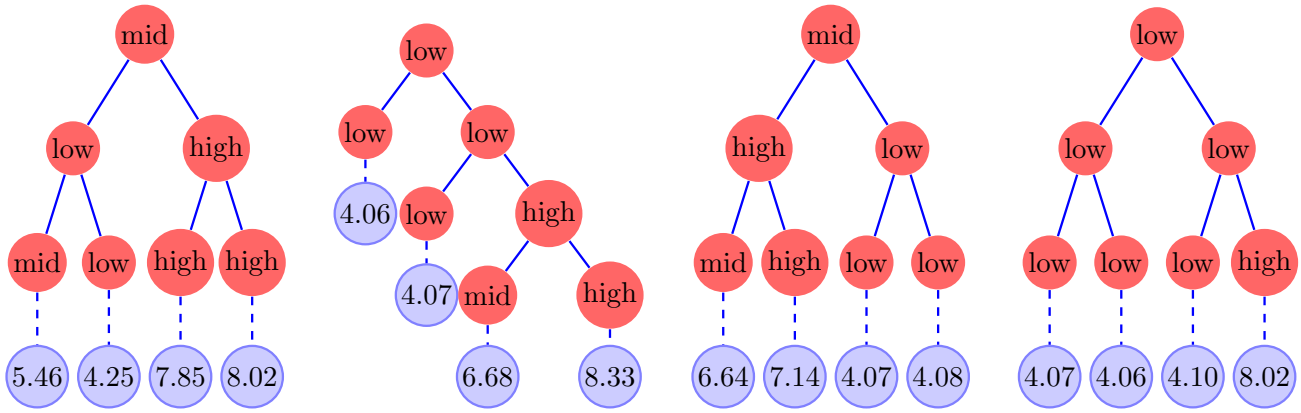


Fig. 4. Representations of trees generated by using the recursive network for learning the structure in Rousk et al.'s pH gradient data set. Only a few of the trees produced from randomly sampled test data sets are shown here due to redundancy and space limitations. The bottom nodes (i.e., leaves) of the tree represent the test samples and the higher level nodes represent merged features of child nodes. The tag inside the red nodes indicate the class associated with the node as determined by the recursive neural network. The blue node indicate the true pH level (and previously unknown to the network) of the leaf nodes in the tree. Recall that the levels are defined as $low \in [4.0, 4.3]$, $mid \in [4.73, 6.68]$, and $high \in [7.1, 9.0]$.

average loss of the RNN classifier was 0.15, even though the trees presented in Figure 4 are classifying the test samples correctly (classifier accuracy is discussed in more detail below). The small size of the pH data set allows easy interpretation of the trees. The bottom nodes in the tree, i.e., leaves of the tree, represent the test samples in the hold-out data set. The higher level nodes represent merged features of child nodes. The tag inside the red nodes indicate the class associated with that node as determined by the RNN. The blue nodes indicate the true pH level (and previously unknown to the network). Recall that the levels are defined as $low \in [4.0, 4.3]$, $mid \in [4.73, 6.68]$, and $high \in [7.1, 9.0]$. The classification of the leaf nodes agree well with the definitions of pH levels. Furthermore, the classification of the mergers in the tree intuitively makes sense. For example, consider the left most tree in Figure 4. The mergers of the two high pH samples result in a parent feature that is classified as a high pH sample.

Also, recall that the nodes in the tree generated by the RNN are represented by features that are learned through a merger of the child nodes. The node features are represented in the same dimensional space as the number of hidden layer nodes (which is typically large) in the network; therefore, they cannot be easily visualized. To visualize the internal node in more detail features, we apply PCoA with the Hellinger distance to the inner nodes of the trees generated with the recursive neural network (see Figure 5). We use the data (training and testing) to generate the tree so that we may observe where data are represented by a node (including leaves and higher level nodes). We observe for both the pH data and the microbiome data that the features of all of the parent nodes in the tree lie in a near central location compared to the data at the leaves of the tree (inner parent nodes are hi-lighted in Figure 5). Conversely, the leaf nodes are the points outside the central region in Figure 5. Intuitively, this result makes makes sense because the inner nodes are mergers of the leaves.

While the plots in Figure 5 (PCoA after RNN processing) appear to be similar to those in Figure 3 (PCoA on raw data), there are some important differences due to feature learning. For example, consider the pH data set. We observe improved class separability for the three classes using deep learning. This result can be attributed to using labeled information during the optimization (i.e., pH levels in the optimization of the neural network).

2) Environmental Classification Performance: The classification errors (i.e., $1 -$ accuracy) of the DBN, RNN, MLPNN, and RFC are presented in Table I. We make several observations about the errors of the classifiers. First, the DBNs – for both hidden unit sizes evaluated – and the MLPNN are the top performers on the pH data set. The RFC and RNN tie in terms of their error rates. We note that the RNN is not only learning a function to predict the labels, but also learning a hierarchical structure in the data as well. The data from the Caporaso et al. study (i.e., microbiome) was more difficult for the neural networks; however, the random forests and MLPNN offered improvements over the deep learning approaches. The RNN offers tangible alternative benefits not available with MLP or RFC, namely, a tree that represents the structure of the data among multiple samples. Hence, not only can the RNN perform the task of classification, but it also provides a hierarchal representation among all samples being tested. The MLPNNs are competitive in terms of accuracy to the other classifiers, such as RFC; in fact, they perform better than the RFC on 2 of the 3 data sets evaluated. Furthermore, these results are obtained without tuning parameters of the MLPNN. Figure 6 shows the training and testing loss of the MLPNN with respect to the number of epochs. One positive quality to note about the MLPNN is that its test error converges quite rapidly (e.g., 20 epochs).

V. Discussion

The experiments discussed in the previous section demonstrated that: (i) the deep learning approaches are not superior, at least on the data sets we evaluated, and (ii) traditional MLPNNs are quite competitive with the RFCs, and in general perform better. However, none of the classifiers – deep or shallow – uniformly performs better than the RFCs across different experiments. The performance of the deep learning

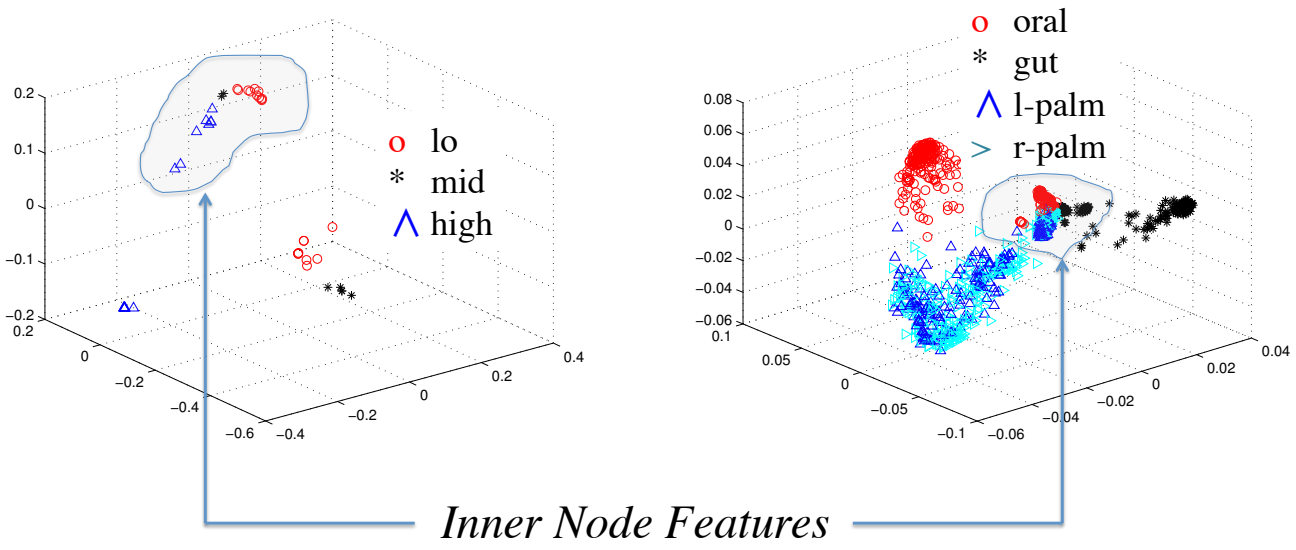
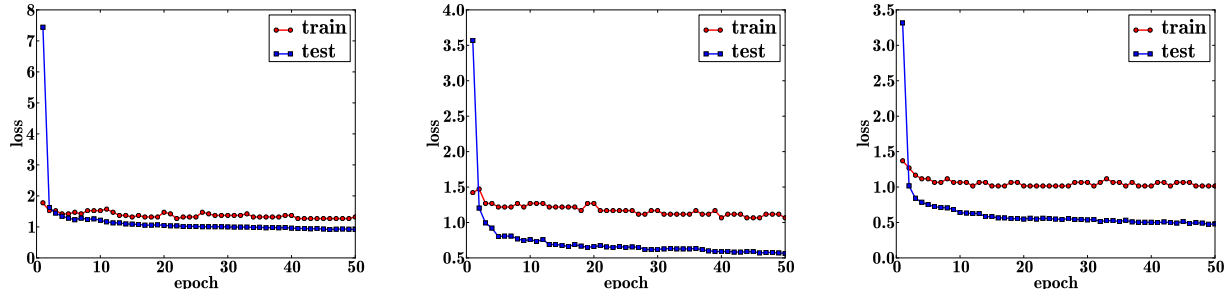


Fig. 5. PCoA plots for nodes of the RNN generated for the pH data (**left**) and the human microbiome data (**right**). The higher level features (inner nodes) of the tree are indicated. All other points represent leafs of the tree.



(a) 25 nodes (b) 500 nodes (c) 1000 nodes

Fig. 6. Training and testing loss after 50 epochs of a single layer MLP neural network on the 15-month data set.

TABLE I
LOSS INCURRED BY THE DEEP LEARNING ALGORITHMS (DBN, RNN), MLPNN AND THE RANDOM FOREST CLASSIFIER (RFC) ON SEVERAL METAGENOMIC BENCHMARK DATA SETS. THE NUMBER IN THE PARENTHESIS FOR THE NEURAL NETWORKS INDICATES THE NUMBER OF HIDDEN LAYER NODES.

	pH		**microbiome**	
			host	*body site*
RNN (50)	0.150	RNN (250)	0.1474	0.1687
DBN (500)	0.075	RNN (500)	0.1915	0.1581
DBN (750)	0.075	DBN (250)	0.2442	0.0315
RFC	0.150	DBN (500)	0.2421	0.0279
MLPNN (500)	0.000	RFC	0.0284	0.0086
		MLPNN (500)	0.0806	0.0056

approaches may be improved upon with data sets that are much larger. It appears that – at least based on accuracy alone – the deep learning approaches may not be suitable for metagenomic applications. Accuracy, however, is not the only figure of merit.

A particular advantage of the RNN, for example, that is not available through other classifiers, is its ability to provide a representation of the hierarchical structure of the collection of metagenomic samples, as seen in Figure 4, when a sample adjacency matrix is available. Such a representation provides additional information and insight into the dataset that would not be otherwise available. The RNN can also be used to annotate future data with class associations, while also providing the high-dimensional feature representation. Furthermore, given an RNN trained on data using labels and adjacency information, equation (6) may still be used to score the merger of samples that do not "share" adjacency. Thus, the RNN can be used to determine which samples are more similar to each other based on the scoring function. Of course, generating such a tree requires the adjacency information, however, even without the adjacency matrix, it is still possible to score other data samples against a tree, or even another collection of observations. If an RNN has already been trained on data using labels and adjacency information collected from a training data set, we use equation (5) to determine a new feature merger and equation (6) to score the merger of samples that may not be strictly adjacent. Thus, the RNN can be used to determine which samples are more similar to each other based on the RNN scoring function.

As mentioned above, both the MLPNN and the RFC out-

performed the DBN – at least in terms of raw accuracy – in our experiments. One of the potential concerns with deploying neural networks for a metagenomic application, however, is the selection of the network parameters (e.g., learning rate, number of epoch, etc). There are generally several options available for optimization of the weights in the network whereas classifiers such as the random forest have far fewer free parameters. However, we have observed that the current selection of network parameters was sufficient over the data sets tested. If proper care is taken in the selection of these parameters, the neural network can be an effective predictor that provide a small loss on a large variety of classification scenarios.

VI. Conclusion

In this work, we evaluated both deep learning and traditional multi-layer neural networks for: (i) learning hierarchical structure in a metagenomic sample, and (ii) classification of phenotypes. The deep learning methods used in this study are becoming increasingly popular in areas such as language modeling, natural language processing, and speech & image classification; however, their utility on metagenomic data sets had not been previously explored, which motivated our efforts. In general, we found that the standard single-hidden layer MLPNN as well as the commonly used RFC performed better than their deep learning counterparts. However, the deep learning approaches provide some additional advantages. For example, the recursive neural network classifier is the only method tested that is not only able to classify metagenomic sample phenotypes, but to also produce a hierarchical relationship of the samples that can be visualized as a tree. The deep belief network, on the other hand, may provide better accuracy than the RNN, but does not supply a hierarchical representation that the RNN provides, and it under performed the MLPNN. As mentioned previously, our goal in this work was not to pick a winner as uniformly the most accurate prediction model for metagenomic data, as we already know from the no free lunch theorem to be an unnecessary exercise [44]. Rather, we highlight the benefits and drawbacks of each of the approaches we presented, and provided insight on how they can be improved upon in the future for predictive metagenomic applications. Our experiments suggest that – at least for smaller datasets and on the basis of accuracy only – the RFC appears to be a better fit for such applications, however, the deep learning approaches may prove to be more effective for much larger datasets, a claim whose assessment remains the focus of our current and near future work.

Acknowledgements

G. Ditzler and G. Rosen are supported by the NSF #1120622, and the DoE #SC004335. R. Polikar is supported by NSF #1310496.

References

[1] J. C. Wooley, A. Godzik, and I. Friedberg, "A primer on metagenomics," *PLoS Computational Biology*, vol. 6, no. 2, pp. 1–13, 2010.

[2] J. Rousk, E. Bååth, P. C. Brookes, C. L. Lauber, C. Lozupone, J. G. Caporaso, R. Knight, and N. Fierer, "Soil bacterial and fungal communities across a pH gradient in an arable soil," *ISME Journal*, vol. 4, pp. 1340–1351, 2010.

[3] R. M. Bowers, S. McLetchie, R. Knight, and N. Fierer, "Spatial variability in airborne bacterial communities across land-use types and their relationship to the bacterial communities of potential source environments," *ISME Journal*, vol. 5, pp. 601–612, 2011.

[4] S. Williamson, D. Rusch, S. Yooseph, A. Halpern, K. Heidelberg, J. Glass, C. Andrews-Pfannkoch, D. Fadrosh, C. Miller, G. Sutton, M. Frazier, and J. C. Venter, "The Sorcerer II global ocean sampling expedition: Metagenomic characterization of viruses within aquatic microbial samples," *PLoS Biology*, no. 1, 2008.

[5] J. G. Caporaso, C. L. Lauber, E. K. Costello, D. Berg-Lyons, A. Gonzalez, J. Stombaugh, D. Knights, P. Gajer, J. Ravel, N. Fierer, J. I. Gordon, and R. Knight, "Moving pictures of the human microbiome," *Genome Biology*, vol. 12, no. 5, 2011.

[6] E. K. Costello, C. L. Lauber, M. Hamady, N. Fierer, J. I. Gordon, and R. Knight, "Bacterial community variation in human body habitats across space and time," *Science*, vol. 326, pp. 1694–1697, 2009.

[7] J. Handelsman, *Committee on Metagenomics: Challenges and Functional Applications*. The National Academies Press, 2007.

[8] J. Raes, K. U. Foerstner, and P. Bork, "Get the most out of your metagenome: computational analysis of environmental sequence data," *Current Opinion in Microbiology*, vol. 10, pp. 1–9, 2007.

[9] J. A. Eisen, "Environmental shotgun sequencing: Its potential and challenges for studying the hidden world of microbes," *PLoS Biology*, vol. 5, no. 3, 2007.

[10] W. Valdivia-Granda, "The next meta-challenge for bioinformatics," *Bioinformation*, vol. 2, no. 8, pp. 358–362, 2008.

[11] J. E. Koenig, A. Spor, N. Scalfone, A. D. Fricker, J. Stombaugh, R. Knight, L. T. Angenent, and R. E. Ley, "Succession of microbial consortia in the developing infant gut microbiome," *Proceedings of the National Academy of Sciences*, pp. 4578–4585, 2010.

[12] D. Knights, E. K. Costello, and R. Knight, "Supervised classification of human microbiota," *FEMS Microbiology Reviews*, vol. 35, no. 2, pp. 343–359, 2011.

[13] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, pp. 1658–1659, 2006.

[14] R. Socher, C. C.-Y. Lin, A. Ng, and C. Manning, "Parsing natural scenes and natural language with recursive neural networks," in *ICML*, 2011.

[15] L. Deng and D. Yu, "Deep convex network: A scalable architecture for speech pattern classification," in *Interspeech*, pp. 2285–2288, 2011.

[16] G. Ditzler, R. Polikar, and G. Rosen, "Forensic identification using environmental samples," in *International Conference on Acoustics, Speech and Signal Processing*, pp. 1861–1864, 2012.

[17] W. Li, L. Jaroszewski, and A. Godzik, "Clustering of highly homologous sequences to reduce the size of large protein databases," *Bioinfomatics*, vol. 17, no. 3, pp. 282–283, 2001.

[18] T. DeSantis, P. Hugenholtz, K. Keller, E. Brodie, N. Larsen, Y. Piceno, R. Phan, and G. Andersen, "NAST: a multiple sequence alignment server for comparative analysis of 16s rrna genes," *Nucleic Acids Research*, vol. 34, pp. W394–W399, 2006.

[19] J. R. Cole, Q. Wang, E. Cardenas, J. Fish, B. Chai, R. J. Farris, A. S. Kulam-Syed-Mohideen, D. M. McGarrell, T. Marsh, G. M. Garrity, and J. M. Tiedje, "The ribosomal database project: improved alignments and new tools for rRNA analysis," *Nucleic Acids Research*, vol. 37, pp. 141–145, 2009.

[20] G. L. Rosen, E. R. Reichenberger, and A. M. Rosenfeld, "NBC: the naïve bayes classification tool webserver for taxonomic classification of metagenomic reads," *Bioinfomatics*, vol. 27, no. 1, pp. 127–129, 2011.

[21] C. Lozupone and R. Knight, "UniFrac: a new phylogenetic method for comparing microbial communities," *Applied Environmental Microbiology*, vol. 71, no. 12, 2005.

[22] N. Fierer, C. L. Lauber, N. Zhou, D. McDonald, E. K. Costello, and R. Knight, "Forensic identification using skin bacterial communities," *Proceedings of the National Academy of Sciences*, vol. 107, no. 14, pp. 6477–6481, 2010.

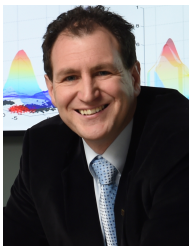
[23] I. Arel, D. Rose, and T. Karnowski, "A deep learning architecture comprising homogeneous cortical circuits for scalable spatiotemporal pattern inference," in *Advances in Neural Information Processing Systems*, 2009.

[24] Y. Bengio, *Learning Deep Architectures for AI*, vol. 2. Foundations and Trends in Machine Learning, 2009.

[25] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems*, 2006.

[26] S. Haykin, *Neural Networks and Learning Machines*. Pearson, 2009.
[27] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Neural Networks* (I. Press, ed.), 2001.
[28] G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neur. Comp.*, vol. 18, no. 7, pp. 1527–1554, 2006.
[29] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, pp. 1771–1800, 2002.
[30] R. H. Byrd, P. Lu, and J. Nocedal, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific and Statistical Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.
[31] P. Turnbaugh, M. Hamady, T. Yatsunenko, B. Cantarel, A. Duncan, R. Ley, M. Sogin, W. Jones, B. Roe, J. Affourtit, M. Egholm, B. Henrissat, A. Heath, R. Knight, and J. Gordon, "A core gut microbiome in obese and lean twins," *Nature*, vol. 475, pp. 480–485, 2009.
[32] L. Breiman, "Random forest," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[33] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proceedings of the National Academy of Sciences*, vol. 99, no. 10, pp. 6567–6572, 2002.
[34] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society*, vol. 67, no. 2, pp. 301–320, 2005.
[35] V. Vapnik, *Statistical Learning Theory*. Wily-Interscience, 1989.
[36] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.
[37] Z. Liu, W. Hsiao, B. Cantarel, E. F. Drábek, and C. Fraser-Liggett, "Sparse distance based learning for simultaneous multiclass classification and feature selection of metagenomic data," *Oxford Bioinformatics*, vol. 27, no. 23, 2011.
[38] Y. Lan, A. Kriete, and G. Rosen, "Selecting age-related functional characteristics in the human gut microbiome," *Microbiome*, vol. 1, no. 2, 2013.
[39] G. Ditzler, R. Polikar, and G. Rosen, "Information theoretic feature selection for high dimensional metagenomic data," in *International Workshop on Genomic Signal Processing and Statistics*, 2012.
[40] S. Essinger, R. Polikar, and G. Rosen, "Ordering samples along environmental gradients using particle swarm optimization," in *International Engineering in Medicine and Biology Conference*, pp. 4382–4385, 2011.
[41] J. Gower, "Multivariate analysis and multidimensional geometry," *Journal of Royal Statistics Society*, vol. 17, no. 1, pp. 13–28, 1967.
[42] R. K. Aziz et al., "The RAST server: Rapid annotations using subsystems technology," *BMC Genomics*, vol. 9, no. 75, 2008.
[43] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?," *Journal of Machine Learning Research*, vol. 15, pp. 3133–3181, 2014.
[44] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

Gregory Ditzler (S'04–M'15) received a BSc from the Pennsylvania College of Technology (2008) and a MSc degree from Rowan University (2011). He is currently a PhD student at Drexel University. He received the Best Student Paper at the IEEE/INNS International Joint Conference on Neural Networks (2014), a Nihat Bilgutay Research Fellowship (2013), Koerner Family Engineering Research Fellowship (2014), and Rowan Research Achievement Award (2009). His current research interests include large-scale feature subset selection, incremental learning, multiple classifier concept drift, online and incremental learning, multiple classifier systems, and applications of machine learning in bioinformatics.



Robi Polikar (S'93–M'00–SM'08) received the B.Sc. degree in electronics and communications engineering from Istanbul Technical University, Istanbul, Turkey, in 1993, and the M.Sc. and Ph.D. degrees in electrical engineering and biomedical engineering from Iowa State University, Ames, IA, USA, in 1995 and 2000, respectively. He is a Professor of electrical and computer engineering with Rowan University, Glassboro, NJ, USA. His current research interests include computational intelligence including ensemble systems, incremental and nonstationary learning, and various applications of pattern recognition in bioinformatics and biomedical engineering. Dr. Polikar is a member of ASEE, Tau Beta Pi, and Eta Kappa Nu. His recent and current works are funded primarily through NSF's CAREER and Energy, Power and Adaptive Systems Programs. He is also an Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.

Gail Rosen (S'98-M'06-SM'13) received her B.Sc., M.Sc., and Ph.D. in Electrical and Computer Engineering from the Georgia Institute of Technology in 2000, 2002, and 2006 respectively. She is currently an associate professor of electrical and computer engineering at Drexel University in Philadelphia, PA, USA. Dr. Rosen received an NSF CAREER award in 2009 and a Louis and Bessie Stein Travel Fellowship in 2012. She is on the editorial board of BMC Microbiome. Her research interests are in machine learning and signal processing methods to improve comparative genomics analyses, many of which are used to understand data from 'omics studies.