Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Genome analysis

Supervised learning on phylogenetically distributed data

Elliot Layne [1,*], Erika Dort [2], Richard Hamelin [2], Yue Li [1], and Mathieu Blanchette [1,*]

[1] School of Computer Science, McGill, Montreal, H3A 0E9, Canada and
[2] Department of Forestry and Conservation Sciences, University of British Columbia, Vancouver, V6T 1Z4, Canada.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: The ability to develop robust machine-learning models is considered imperative to the adoption of ML techniques in biology and medicine fields. This challenge is particularly acute when data available for training is not independent and identically distributed, in which case trained models are vulnerable to out-of-distribution generalization problems. Of particular interest are problems where data corresponds to observations made on phylogenetically related samples (e.g. antibiotic resistance data).

Results: We introduce DendroNet, a new approach to train neural networks in the context of evolutionary data. DendroNet explicitly accounts for the relatedness of the training/testing data, while allowing the model to evolve along the branches of the phylogenetic tree, hence accommodating potential changes in the rules that relate genotypes to phenotypes. Using simulated data, we demonstrate that DendroNet produces models that can be significantly better than non-phylogenetically aware approaches. DendroNet also outperforms other approaches at two biological tasks of significant practical importance: antiobiotic resistance prediction in bacteria, and trophic level prediction in fungi.

Availability: https://github.com/BlanchetteLab/DendroNet

Contact: elliot.layne@mail.mcgill.ca, blanchem@cs.mcgill.ca

1 Introduction

In supervised machine learning, most work operates under the assumption that the available data points are independent and identically distributed. Yet in many bioinformatics applications, this is not the case. This assumption is particularly strongly violated when examples are phylogenetically or genealogically related. Consider for example the problem of predicting a certain phenotype P from genomic sequence G. In some cases, the data can safely be assumed to be identically distributed; that may be the case when molecular phenotypes (e.g. the binding of a given transcription factor) are being predicted based on data (e.g. ChIP-seq) obtained from a given cell type and species(18). In other cases, data points are interrelated. In population genetics, one seeks to establish this link based on data obtained from individuals from a given species. This population may have a structure that makes certain predictors work better on some subgroups than others (19). Finally, in the case that interests us here, data points may each originate from a different species. Important instances of this problem include prediction tasks related to resistance to specific drugs in viruses, bacteria, or eukaryotes, trophic levels in fungi, the ability of an invasive species to survive in a given type of territory or conditions, the susceptibility of a given species to a pathogen, and many other prediction tasks where labeled examples are evolutionarily inter-related.

An important property of these types of problems is that the relation f between G and P may *change* along the branches of phylogenetic tree T. This may be due to changes in environmental conditions or in genetic/epigenetic factors not captured in G. In these cases, one should not aim to learn a single function f, but instead a family of interrelated functions $\mathcal{F} = \{f_u\}_{u \in T}$, potentially one for each node u of a phylogenetic tree T.

To make clear the challenge that phylogenetically distributed training data presents, we can consider the following hypothetical example. Let us imagine that there is some bacterial gene g that confers resistance to

a particular antibiotic A of interest. A researcher may train a machine-learning model to predict resistance to A from a collection of bacterial strains given a set of genetic features including the presence of g, and be able to make accurate predictions about the resistance phenotype. If the model is sufficiently interpretable, it may be possible to identify the presence of g as a significant factor contributing to resistance to A. However, consider the possibility that the dataset of bacterial strains actually consists of two sub-groups: B_1 and B_2. In B_1, g confers resistance to A. In B_2, some factor prevents g from having this effect. If the training information does not capture this factor, there will be no way for the model to learn this difference in relationship between g and resistance to A. If the training set consists mostly of samples from B_2, the model will likely incorrectly predict that samples in B_1 with g will not be resistant to A, and vice-versa.

Given datasets where the relationships between input features and target predictions are subject to variation between phylogenetic sub-groups, we seek to develop a new method of training machine learning that have the following key properties:

- Account for and leverage the existence of phylogenetic relationships among training and test examples
- Consider the possibility that the function to be learned may change along the branches of the tree
- Leverage the phylogenetic structure and evolutionary variation in order to encourage the learning of biologically interpretable models

To achieve these goals, we introduce the a new type of semi-supervised learning approach which we refer to as **DendroNet**. We show that DendroNet is capable of making more accurate predictions than non-phylogenetic approaches, using both simulated data as well as actual antibiotic resistance and fungi trophic level data.

1.1 Related work

A simple approach when facing non-identically distributed data is to ignore the issue, and hope that a single predictor f will manage to achieve a decent prediction accuracy across the entire sample space. Most existing approaches for phenotype prediction fit under this category, and they are relatively successful provided f does not change substantially. If we consider the example of bacterial antibiotic resistance (ABR), the work of Drouin *et al.* (9) is one example of a method that has achieved significant success without explicitly addressing the non-iid nature of the training data. Besides the impact of non-iid data on overall prediction accuracy, an equally problematic issue is that predictors that operate under iid assumptions tend to fail at a much higher rate on under-represented groups (be they ethnic groups in a population genetics study, or a particular clade of bacteria for which a predictor systematically mis-estimates risks).

An alternative approach is to subdivide the data sets in different subgroups (e.g. based on phylogenetic placement or inferred ethnicity) and train separate, independent predictors for each clade. Two drawbacks of this approach are that (i) this may leave insufficiently many examples to train from in each clade, and (ii) it still relies on the assumption that f does not change within the predetermined subgroups.

More sophisticated approaches attempting to directly address the problem of non-iid data sources have also been proposed. For example, Felsenstein's work on Phylogenetic Independent Contrasts uses the independent contrasts between dependent features to allow for statistically sound regressions on the relationship between phylogenetically distributed variables (21). With regards to the use of non-iid data for supervised learning, Earle *et al.* demonstrate that performance on the classification of ABR can be increased by adjusting for lineage effects through the use of a Linear Mixed Model (LMM) based approach (12).

In machine learning, there have been numerous approaches proposed to model non-iid data. Many focus on the problem of non-stationary learning, aiming to handle dataset shifts in a time-series context. The work done by Raza *et al.* is an example of a typical approach to this problem, which they call adaptive learning (11). In their approach, incoming data is examined for signs of a shift event, representing a significant modification to the distribution of the data. Upon detection of a shift event, re-training of the model begins to adapt to the new distribution. Other approaches do not repeatedly re-train the model, but instead iteratively train an ensemble of classifiers optimal with respect to the data distribution at local time points, and combine them to produce predictions with stable global performance. For example, Alippi *et al.* (13) presented an approach dubbed the "Just In Time" ensemble approach, where the detection of a dataset shift event results in the addition of a new learner to the ensemble. They later expanded on this work, seeking to identify and capture recurrent concepts contributing to dataset shift events, and repeatedly adjust the content of the ensemble in order to use learners trained on concepts appropriate for the current state of the data. Finally, the work done by Wang *et al.* (10) examined leveraging the technique of transfer learning in order to manage the problem of non-stationary environments (10). They treat adjusting to modified training data distributions as a transfer learning problem, using a model pre-trained on the prior distribution and fine-tuning it to the new distribution. As part of the process, they assume that the old and new distributions should most often be relatively similar, and regularize the fine-tuning process of the model to encourage smooth changes.

The work that will be presented in this paper could in some sense be considered a generalization of these techniques for handling dataset shifts in a phylogenetic context. We will seek to address a set of conditions that none of the previous works described fully addresses: (i) DendroNet smoothly handles changes in data distribution according to the phylogenetic-tree structure, (ii) it leverages information from the entire dataset when making predictions, rather then using only the samples most closely related to the current point of analysis, and (iii) it also flexibly allows for the training of both linear and non-linear learners, of a wide variety of architectures.

1.2 Problem Formulation

Consider a data set $X = x_1, ..., x_n$ made of n examples, where each example $x_i = x_{i,1}, ..., x_{i,d}$ is a d-dimensional vector. Each example x_i is assigned target value $y_i \in D$, where D could either be categorical (for classification tasks) or continuous (for regression tasks). In a classical supervised learning setting with iid examples, one would express the task as learning $f(x, y) = \Pr[y|x]$. To avoid the iid assumption, we generalize the problem as that of learning $\mathcal{F} = \{f_u\}_{u \in T}$, where $f_u(x_u, y_u) = \Pr[y_u|x_u]$ captures the relation between x and y at node u in the tree.

In the cases that interest us here, the number of training examples at node u may be extremely small: typically either zero (for ancestral nodes in the tree) or one (at the leaves). This would be the case, for example, in antibiotics resistance data sets, where a single measurement is available at each node. Hence one needs to introduce strong regularization on $\mathcal{F}$. A biologically meaningful regularization would be one where f is allowed to evolve along the branches of T, but slowly (or rarely). This would provide the required flexibility to allow for changes in f, while allowing multiple examples from a given portion of the tree to collectively inform the choice of f_u at each node.

2 Methods

2.1 DendroNet approach

The core contribution of this paper, referred to as DendroNet, will now be introduced. Consider a family of functions $\mathcal{H}$ parameterized by weight vector θ. $\mathcal{H}$ could for example correspond to the set of functions implemented by a neural network with fixed architecture, with connection weight vector θ. DendroNet aims to identify $\Theta = \{\theta_u\}_{u \in T}$ so as to minimize

$$L(\Theta) = \sum_{u \in \text{leaves}(T)} L_e(y_u, f(x_u; \theta_u)) + \lambda \sum_{(u,v) \in \text{edges}(T)} L_d(\theta_u, \theta_v; l(u,v)) \quad (1)$$

Here, L_e, called the error loss, may be the sum of squared error (for regression tasks) or cross-entropy functions (for classification tasks). L_d is the DendroNet loss term, which aims to penalize differences between θ vectors at tree nodes connected by an edge of length $l(u,v)$ in the tree, while λ is the DendroNet regularization weight. Depending on the context, L_d may take different forms. If an explicit model of evolution of θ vectors is available, L_d would be chosen as the log-likelihood of the set of changes to the θ vectors that took place along a given branch of T. For example, if θ evolves according to a Brownian motion, $L_d = ||(\theta_u - \theta_v)/l(u,v)||_2^2$ would be appropriate. If the changes to θ follow an exponential distribution, one would choose $L_d = ||(\theta_u - \theta_v/)l(u,v)||_1$. Finally, we may consider a model where weights remain completely unchanged with probability $1-p(l(u,v))$, but change according to a normal or exponential distribution with probability $p(l(u,v))$. This may be relevant for cases where changes in θ are caused by rare discrete events such as genetic mutations or gene gains/losses due to horizontal gene transfers or gene duplication/deletion. The associated L_d function would be discontinuous:

$$L_d(\theta_u, \theta_v) = \begin{cases} \log(1 - p(l(u,v))) & \text{if } \theta_u = \theta_v \\ \log(p(l(u,v))) + ||\theta_u - \theta_v|| & \text{otherwise} \end{cases}$$

To facilitate the optimization process, this function could be approximated using a continuous approximation. In this paper, we focus on the case where changes to θ follow an exponential distribution, and $p(l(u,v)) = 1$ (i.e. small updates happen to the weights along every branch). However, the approach and its implementation can easily accommodate the other functions introduced here.

While the problem of minimizing $L(\Theta)$ for a given choice of L_e and L_d loss functions and a given training set is well defined and could be fed to an automatic differentiation tool such as the GradientTape API provided in the Tensorflow library (20), it offers significant challenges to gradient-based optimizers. This is due to the fact that updates made to θ_u in order to reduce L_e will tend to make θ_u more dissimilar to $\theta_{parent(u)}$ and $\theta_{child(u)}$, hence increasing L_d. In the situation where all θ vectors are identical (but non optimal), and if the regularization weight λ is large, this tends to create an infinite number of saddle points, effectively stalling gradient descent.

To alleviate the problem, and obtain a more interpretable model, we propose a re-parameterization of the above models (Figure 1). Let δ_u be a d-dimensional vector associated to node u, and let θ_r be a weight vector associated to the root r of tree T. Then, we define $\theta_u = \theta_r + \sum_{v \in path(u)} \delta_v$, where path($u$) is the set of nodes on the path between r and u, including u itself but excluding r. Now, θ_u is simply the ancestral weight vector θ_r plus all the changes that happened during the evolution from r to u. The optimization problem is now formulated as aiming to minimize $L(\Theta)$ by selecting θ_r and the set of all update vectors $\Delta = \{\delta_u\}_{u \in T \setminus \{r\}}$.

This formulation offers a number of benefits. First, for any internal node u, updates to δ_u impact all nodes in the subtree rooted at u, effectively eliminating the above mentioned saddle points and making it very easy for the predictor to learn a set of models related to one another in an evolutionarily realistic manner. Second, it improves the interpretability of the model, because once a model is trained, δ vectors can be inspected to identify branches of the tree where the largest changes to the model took place, which may suggest evolutionarily valuable hypotheses.

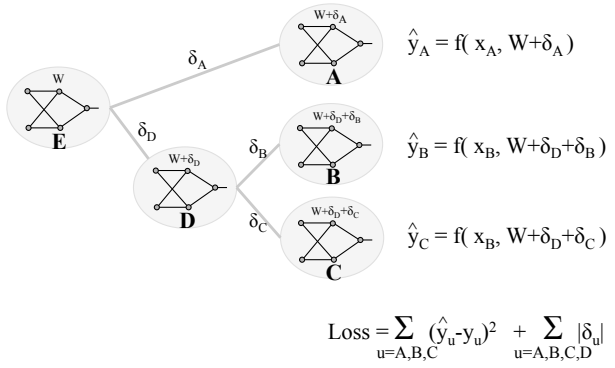

Fig. 1: An example of a DendroNet instance, trained on a dataset of 3 samples (leaves A, B and C), connected by a phylogenetic tree that has root node E and an internal node D. A parameter vector W for a fully-connected neural network is instantiated at root E. Along every edge u, there is a trainable vector δ_u with a corresponding entry for every parameter in W. Predictions at leaves A, B and C are made using model parameters parameters equal to the sum of W and the δ vector along all paths from the root to the leaf.

Once a DendroNet model is trained, prediction on a test example is done as follows. First, the test example is located in the phylogenetic tree. Here, we assume that some external information is available regarding species phylogenetic placement. Since no weight vector is available at that leaf, the prediction is made based on the weight vector of the closest ancestral node.

2.2 Implementation

Machine-learning operations for the fungi and antibiotic resistance experiments were implemented using the Tensorflow Keras library with Tensorflow 2.0 backend (20). The experiments with the simulated data were conducted using an equivalent implementation in PyTorch (22). Implementation of the trainable deltas typically used the following pattern: at the root of the tree, a place-holder vector of zeros was instantiated. A trainable addition-layer of the same shape added an initial value to the zeros, representing the weights W of the root-model. For each edge in the tree, another trainable addition-layer was created, modifying the values of W as they passed between neighbouring nodes down the tree. The magnitude of the parameters of these addition-layers were passed to the DendroNet loss, and were subject to whatever form of regularization was being used. At each leaf, the resulting mutated values of W were re-shaped into the appropriate layers for whatever architecture was being used for predictions.

2.3 Additional Comparison Methods

The performance of the DendroNet method in the classification tasks was compared against several competing baseline methods. For both the fungi trophic level and antibiotic resistance prediction tasks, the DendroNet method was used to train a logistic regression classifier with appropriately mutating weights at various positions in the respective phylogenetic tree. The first comparison method was a traditional logistic regression model trained using the same features, but without the ability to allow weights to vary. The second baseline was again a logistic regression model, but with additional training features added to represent the phylogenetic placement of each sample. The third comparison approach was to train a DendroNet model, but without making use of any training features other than a single bias feature. The purpose of this baseline was to investigate how strongly a parsimonious reconstruction of the phylogenetic signal alone could be used to make predictions. The fourth baseline method included was the LMM, as used by Earle *et. al.* (12). The LMM is a well-documented statistical model that extends linear models to allow superior regressions of non-independent or hierarchical data. The fifth and final baseline method used for comparison was a Gaussian Processes Regression model, in which the data points are modelled as being members of a multivariate Gaussian distribution.

2.4 Fungi Trophic Level Prediction

For the prediction of trophic levels from fungal data, a phylogenetic tree and secondary metabolite cluster counts were obtained from the JGI database (3) for 261 fungal species for which genome data was available and for which we were able to annotate trophic levels by manual curation of the literature (16). DendroNet was used to train a group of logistic regression models, using $\lambda = 1$. Models were trained for 1000 epochs using the Adam optimizer (6) with an initial learning rate of 0.001. AUC scores were evaluated on 10 separate train-test splits of the dataset, each splitting the data into 70% training and 30% test.

2.5 Antibiotic Resistance Prediction

Antibiotic resistance data, and gene counts for gene families associated with either the Antibiotic Resistance category or the Virulence Factor category were obtained from the PATRIC database (14), for Firmicutes. As a fully-resolved phylogenetic tree was not available for PATRIC, a multifurcating tree was constructed based on the species' taxonomy, with internal nodes corresponding to phylum, class, order, family, and genus, and leaves corresponding to the bacterial strains/species.

For each drug, a DendroNet model was trained, again using a logistic regression classifier as the base architecture. The other baseline models previously described were also used for comparison. For the phylogenetic placement features provided to the regression baseline model, we used a one-hot encoding of the taxonomic class, resulting in a total of 77 different classes present in the Firmicutes data.

A test set consisting of 25% of the full dataset was set aside. The appropriate regularization factor λ was found by tuning the Dendronet model on a 4-fold split of the 75% of the dataset allocated for training and validation.

After tuning, λ for DendroNet was set to 0.01 when classifying resistance for the drugs: erythromycin, clindamycin, tetracycline, ceftriaxone, ciprofloxacin, gentamicin, vancomycin, methicillin, penicillin, cloramphenicol. It was set to 0.1 for the drug trimethoprim, and 0.0001 for betalactam.

The value of λ was set for the baseline bias-only parsimony approach by the same method. The methicillin model used a λ of 0.1. The value 0.01 was used for: betalactam, cloramphenicol, erythromycin, and vancomycin. The models for the remaining drugs used a λ of 0.001.

Models were trained for 10,000 epochs, using the ADAM optimizer (6) implemented in Tensorflow 2.0 (20). The initial learning rate was set to 0.0001.

3 Results

Experiments were performed with both simulated and real-world data sets in order to assess the DendroNet.

3.1 Prediction of fungi trophic levels

The trophic level of an organism describes its position in the food chain. In the case of a fungus, the trophic level corresponds to the types of substrates it feeds on. Accurate prediction of trophic levels is important in particular in the context of biosurveillance, e.g. to assess the risk level associated to certain plant pathogens if they were to invade a particular ecosystem (15). We investigated four classes related to plant pathogenicity: Obligate biotrophs (OB) require a living host; hemibiotrophs (HB) are parasitic in a host while it remains living, but can continue to survive in the dead tissue if the host has died. Necrotrophs (N) kill the organism they live on and then feed on it, and saprotrophs (S) feed on decaying matter. Note that these labels are not mutually exclusive.

We assembled a manually curated dataset of 261 fungal species with complete genome sequences and trophic levels documented in the literature (16). Trophic levels have been shown to be linked to presence of genes associated to specific secondary metabolites (17). Here, we used as features the gene counts (integer between 0 and 96) of eight families associated to different secondary-metabolite clusters: DMAT, HYBRID, NRPS, NRPS-Like, PKS, PKS-Like, and TC, with the 8th feature being a sum of the previous seven (see Methods). For each trophic level, the target values are binary values describing whether or not a species belongs to that trophic level. Four independent binary classification tasks were thus formulated, one for each trophic level. In addition to DendroNet itself (operating on a logistic regression model with 8 features and one bias term), we evaluated several alternate approaches: (i) a simple logistic regression model on the 8 features mentioned above, but without DendroNet (i.e. the same model weights apply to all species); (ii) a logistic regression model using the same 8 features, complemented with a bit-wise encoding of the first three levels of the tree along the path from the root to the relevant species, hence enabling the model to factor in phylogenetic placement; (iii) a simplified version of DendroNet called Parsimony, where the eight input features are ignored, leaving only the bias terms to be estimated, yielding a predictor that exclusively relies on phylogenetic placement and makes predictions based on the most parsimonious labeling of internal nodes in the tree; (iv) a standard LMM model using the same bit-wise encoding features previously described; (v) a Gaussian Processes Regression model. Each model was trained and evaluated on 10 random 70% train - 30% test splits.

The results from the fungi classification experiment are shown in 2. DendroNet significantly improved AUC values on two of the classification tasks: Obligate Biotrophs (0.96 vs 0.92) and Necrotrophs (0.75 vs 0.68). On the other two classification tasks, DendroNet fell slightly short of the marks reached by the logistic regression with phylogenetic placement features and the LMM approach. Nonetheless, the average AUC value obtained with DendroNet is superior to that of all other models (75.5% for DendroNet vs 74.3% for LMM). While this difference is not statistically significant owing to the small number of data sets available, it suggests that DendroNet may, in some cases (e.g. necrotrophs), be able to capture properties that other models fail to capitalize on.

Empirically, the trained DendroNet models were observed to have a very small number of branches were significant changes to the weights were inferred. To understand how DendroNet achieved a large AUC gain on the Necrotrophs dataset, we identified the few branches of the tree

Antibiotic Target	Dendronet	Log. Reg.	Log. Reg. + Phylogenetic Placement	Parsimony Approach
Trimethoprim	0.83	0.83	0.83	0.60
Tetracycline	**0.96**	0.94	0.94	0.57
Ceftriaxione	**0.96**	0.95	0.95	0.57
Betalactam	0.83	0.84	0.84	0.50
Gentamicin	**0.96**	0.95	0.95	0.67
Vancomycin	1.0	1.0	0.99	0.96
Ciprofloxacin	0.96	0.96	0.96	0.71
Erythromycin	**0.99**	0.96	0.95	0.70
Clindamycin	**0.97**	0.96	0.96	0.70
Methicillin	0.98	0.98	0.98	0.50
Penicillin	0.97	0.98	0.98	0.48
Cloramphenicol	0.84	0.86	**0.87**	0.58

Table 1. AUC scores on the held-out test set for the prediction of resistance to 12 antibiotics in Firmicutes.

where substantial changes to the weight vectors (large $|\delta_u|$ values) had occurred (Figure 5). We hypothesize that those branches may correspond to changes to the molecular mechanisms that enable organisms of that phylum to succeed in that lifestyle. Substantial changes took place along two branches, indicated in Figure 5. Interestingly, one of the two branches corresponds to the ascomycetes-basidiomycetes split, a very ancient event that may very well have been followed by mechanistic divergence.

3.2 Prediction of Antibiotic Resistance

The prediction of a given bacterial species/strain's susceptibility or resistance to treatment from a given antibiotic is a highly relevant example of a classification task where the training data share phylogenetically distributed relationships. The ability of DendroNet to improve performance on this task was analyzed using data retrieved from the PATRIC database (14) (see Methods). Focusing on the Firmicutes phylum, we retrieved data for 13 drugs with at least 500 species annotated as resistant/susceptible based on a lab experiment (computational annotations were excluded). Rifampin was excluded due to an insufficient number of positive examples, resulting in an analysis of 12 drugs.

The features used to train the models came from PATRIC's specialty gene annotations (see Methods). The set of gene families annotated as being relevant to antibiotic resistance or virulence was collected across all bacterial strains with a resistance phenotype labelled for each drug of interest. For each bacteria, a feature vector was constructed with the number of genes belonging to each functional annotation. For each drug classification task, the feature vector was restricted to gene families occurring at least once in the relevant labelled set of bacteria, resulting in a feature space ranging in size from 50 to 200, depending on the drug.

DendroNet and baseline methods were trained and evaluated on these 12 binary classification tasks, and the results are presented in Table 1. DendroNet obtained the best test AUC for five of the 12 antibiotics tested, and was tied for best (within 1%) in four more cases. Logistic regression with and without phylogenetic placement features had similar, slightly inferior performance. Our parsimony approach, which relies exclusively on phylogenetic placement, performed poorly in all data sets, suggesting that gene count information is essential for accurate predictions. Results for the LMM and GP have not been included for the antibiotic resistance experiments. LMM produced very poor results on this task. Training the GP model was not completed due to excessive run-time.

3.3 Dissecting DendroNet using simulated data

To better study the performance of DendroNet, and its dependency on various evolutionary parameters, we designed several simulation studies, based on the following framework. Assume that the relation between y and x is parameterized by a $d = 3$ dimensional weight vector w: $y = g_w(x) = \sum_{i=1}^{d} tanh(w_i \cdot x_i)$. The form of this function was chosen because it is a simple non-linear function that can be (in principle) be represented exactly by a simple fully connected neural network (see below).

The simulation procedure is as follows. We start by selecting the topology of the tree. Two types of trees are considered: (i) A balanced binary phylogenetic tree with $n = 512$ leaves is first created, and (ii) a multifurcating tree with four main lineages obtained from a complete binary tree of depth 2, each multifurcating into 16 species. At the root r, each component of the weigh vector w_r is sampled from a Normal(0,1) distribution. Weight vectors are then allowed to evolve along the branches of the tree.

For each branch, each weight is assigned a random update drawn from an exponential distribution symmetrized around zero (also known as a Laplace distribution), with expected magnitude of M, which we call the mutation rate. The evolution simulation proceeds until weight vectors are obtained for all leaves. For each leaf u, a feature vector x_u is sampled from a d-dimensional Normal(0,1) distribution, and we obtain

$$y_u = g_{w_u}(x_u) = \sum_{i=1}^{d} tanh(w_{u,i} \cdot x_{u,i}) + \epsilon,$$

where $\epsilon \sim Norm(\mu = 0, \sigma^2 = 0.01)$ for the balanced binary tree, and $\epsilon \sim Norm(\mu = 0, \sigma^2 = 0.8)$ in the case of the multifurcating tree. Simulated datasets were generated with different values of update distribution parameter M.

The data hence generated is used to train a DendroNet model, using L1 mutation regularization (Methods), to match the exponentially distributed weight updates used in the simulation. The architecture of the predictor is a simple fully connected neural network with 3 inputs, one 3-neuron hidden layer with tanh activation, and a single output neuron with linear activation. Note that this 12-weight network can emulate function g by setting 6 of the 9 weights of the hidden layer to zero, and all three weights of the output layer to 1. Thus, the network is actually over-parameterized for the function it is tasked to learn. This over-parameterization is a common feature of neural networks. In a situation where sufficiently many identically distributed training examples are available (e.g. if weight vector θ never changes), this over-parameterization is inconsequential in terms of prediction accuracy. However, it results in entanglement of the nodes in the hidden layer, which is undesirable from the point of view of interpretability. When less training data is available, this over-paramerization leads to over-fitting and reduced prediction accuracy.

For comparison, we also used the same data to train a more classical model that assumes that function f does not change over time. This model has the same architecture as described above, but operates on the training data under the iid assumption. Both models (DendroNet-style and traditional method) were trained using the Adam optimizer (6), for 8,000 epochs using an initial learning rate of 0.001. They were evaluated across 100 repeated 70% train and 30% test splits.

Results from the trials on the simulated data lend support to the conclusion that a properly-tuned λ allows the DendroNet approach to outperform traditional methods in the context of phylogenetically distributed training data. First, we used our 64-leaf multifurcating tree to assess the accuracy of the predictor at different values of the mutation rate (Figure 3, top panel). At a mutation rate of zero, the fully connected neural network (FCNN) achieves optimal performance (MSE= noise level = 0.8), as do the DendroNet models, provided λ is sufficiently large. As the mutation rate increases, the MSE of the FCNN rapidly degrades, because it is unable to

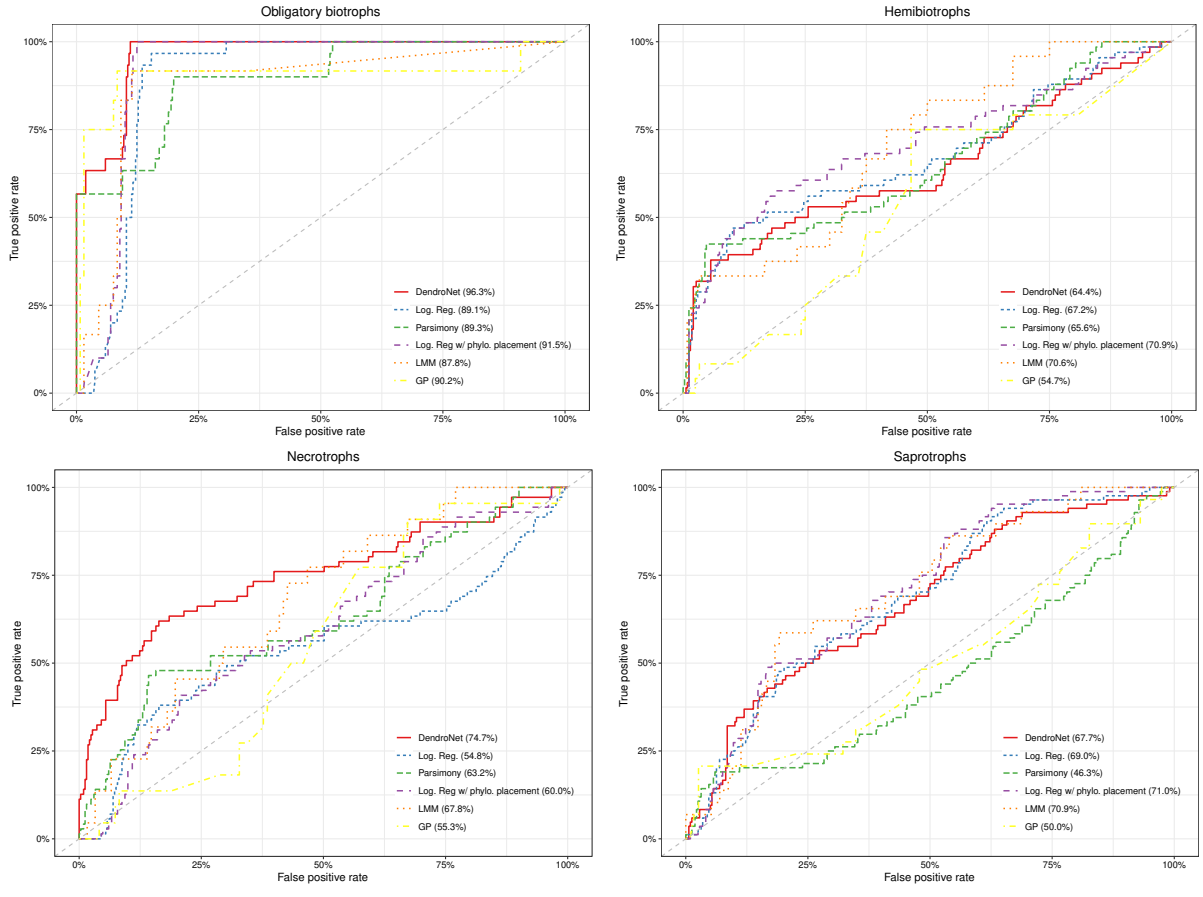

Fig. 2: Comparison of ROC curves for Dendronet and the five comparison models across four trophic level classification tasks

accommodate for changes in the model. DendroNet with $\lambda = 1$ behave similarly, because such a large regularization weight effectively prevents any change in the weights of the network. However, at more moderate regularization weights ($\lambda = 0.1$), we observe that DendroNet manages to retain a low MSE value even at high mutation rate. However, with too little regularization ($\lambda \leq 0.01$), overall accuracy is not as good. Those results indicate that DendroNet is able to effectively learn in the context of an evolving model, but that the choice of λ value is important. Obviously, in a real application, λ would be chosen using hyper-parameter search evaluated on a validation set.

Figure 3 (bottom panel) show analogous results for our complete binary tree with 512 leaves. Again, we observe a benefit to using DendroNet over the FCNN, although in this case the gains in MSE are smaller due to the topology of the tree. Overall, these results show that DendroNet with the right choice of λ provides the flexibility to learn an evolving model while preventing significant overfitting.

Counter-intuitively, optimal results are obtained with $\lambda = 0$. In theory, such an unregularized model should have no chance of generalizing to unseen examples, because it is free to learn a completely different sets of weights at each leaf, i.e. to learn a different network for each individual training example. This intriguing result is actually an interesting side-effect of the optimization procedure used to learn Dendronet's θ_r and $\{\delta_u\}_{u \in T}$ weights. Because the weights θ_r impact the loss of all the training examples, the gradient of the loss function with respect to them are much larger than those on the δ_u weights (especially those located near the leaves). This results in the gradient-based approach to rapidly converge on a set of weights θ_r that are the best compromise to fit all the leaves, and which generalize to unseen examples relatively well. This is somewhat explicitly encouraged by the parameter initialization process used by DendroNet, where the parameters at the root are initialized to larger starting values than all other nodes, encouraging larger gradients at the start of training. Following this first phase, the weights that then have the largest gradients are the δ_u weights associated to the branches near the root, which get estimated based on the data in their (relatively large) subtrees. Eventually, as learning proceeds, branches connected to leaves become those for which the gradient is strongest, and the δ_u weights associated to these branches overfit the single training example at that leaf, without incurring a cost of increased loss elsewhere in the tree. However, this is inconsequential on the generalization of the full model for this architecture, because only internal nodes are involved in the prediction on test examples. Hence, even when $\lambda = 0$, the model tends to converge to weights that generalize well.

This can be further studied by looking at the learning curve of a DendroNet model being trained 4. With λ set to zero, train MSE drops rapidly without significant overfitting, but eventually overfits as δ_u weights associated to nodes low down the tree become optimized to fit the noisy training examples at each leaf. With $\lambda = 0.1$, the optimizer first improves the MSE loss while the parameter-mutation DendroNet-loss increases from near-zero. As the DendroNet-loss grows, eventually, the MSE loss plateaus and the DendroNet-loss starts to become optimized, helping to prevent excessive over-fitting as the model converges.

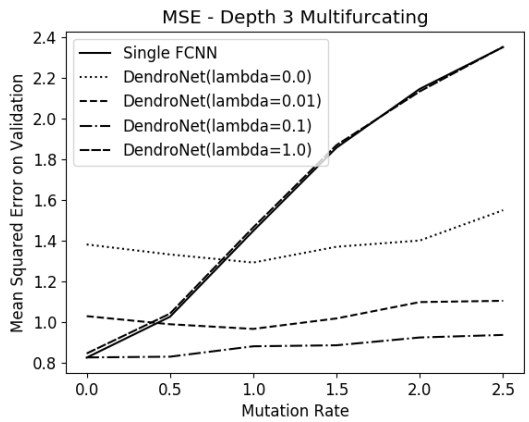

Fig. 3: Validation MSE in simulated data, across various mutation rates, for a fully connected neural network (FCNN) and for DendroNet with different values of regularization weight λ. Top: simulated phylogenetic tree of depth 3, with 16 example species per leaf, and noise $\epsilon = 0.8$. Bottom: simulated phylogenetic tree of depth 9 with one example species per leaf, and noise $epsilon = 0.01$.

These results reveal a complex interplay between regularization and optimization that will be investigated further elsewhere, and which suggests that training seemingly complex DendroNet models involving a very large number of weights may actually be easier than it looks.

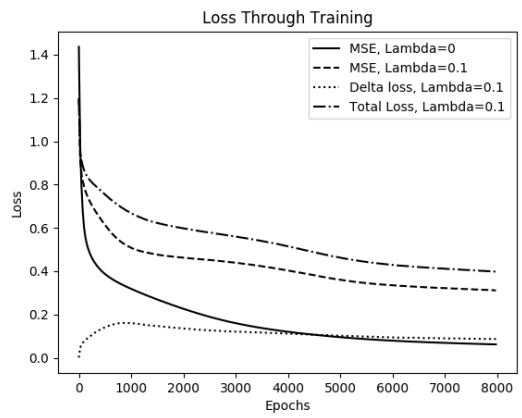

Fig. 4: Progression of the different components of DendroNet's training loss (MSE and DendroNet-loss) as learning progress on simulated data, for $\lambda = 0$ and $\lambda = 0.1$.

4 Discussion and Conclusion

In this paper, we introduced DendroNet, an approach to train predictive models in a context where training examples are phylogenetically related and where the relation between the features and target values changes along the branches of the phylogenetic tree. Notably, the DendroNet approach is applicable to any continuous hypothesis space, including linear and logistic regression as well as most feed-forward and recurrent neural networks architectures. This means that it has the potential of building upon the plethora of models that have been recently been proposed for a variety of tasks in bioinformatics and possibly beyond. Indeed, we demonstrated DendroNet's ability to improve performance on two real-world classification tasks: predicting fungi trophic level from genetic features, and resistance to antibiotics in bacteria.

Although the number of weights that need to be learned is large because it is proportional to the number of training examples, overfitting is kept at bay due to the regularization we introduce, and thanks to an unexpected property of the optimization problem at hand. The positive results from the fungi trophic level and antibiotic resistance prediction tasks, along with a detailed simulation study, provide strong evidence that properly tuned mutation regularization is able to prevent DendroNet-trained models from overfitting.

Although we do not have the space to discuss this in this paper, we also have data suggesting that in some circumstances, the models learned using DendroNet may be better disentangled and more interpretable than standard approaches.

There are some limitations to the DendroNet method that are important to draw attention to. The method is dependent on having a phylogenetic tree relating the training samples. Phylogenetic information is not always available, and can sometimes be inaccurate or even not representable as a tree (e.g. due to horizontal gene transfers). Also, the number of parameters required to train a model with the DendroNet method also increases linearly with the number of samples in the training set. This can cause problems when the datasets of interest are very large and the computational resources available limited. Sparse representations of the δ_u weight vectors, which account for the vast majority of the trainable parameters, may help alleviate this problem. One could also reduce the number of δ vectors to be trained by only allowing changes along pre-selected branches of the tree (e.g. those delineating the major clades). Despite these limitations, it has been shown that the DendroNet method can provide value in the tasks of antibiotic resistance classification, and

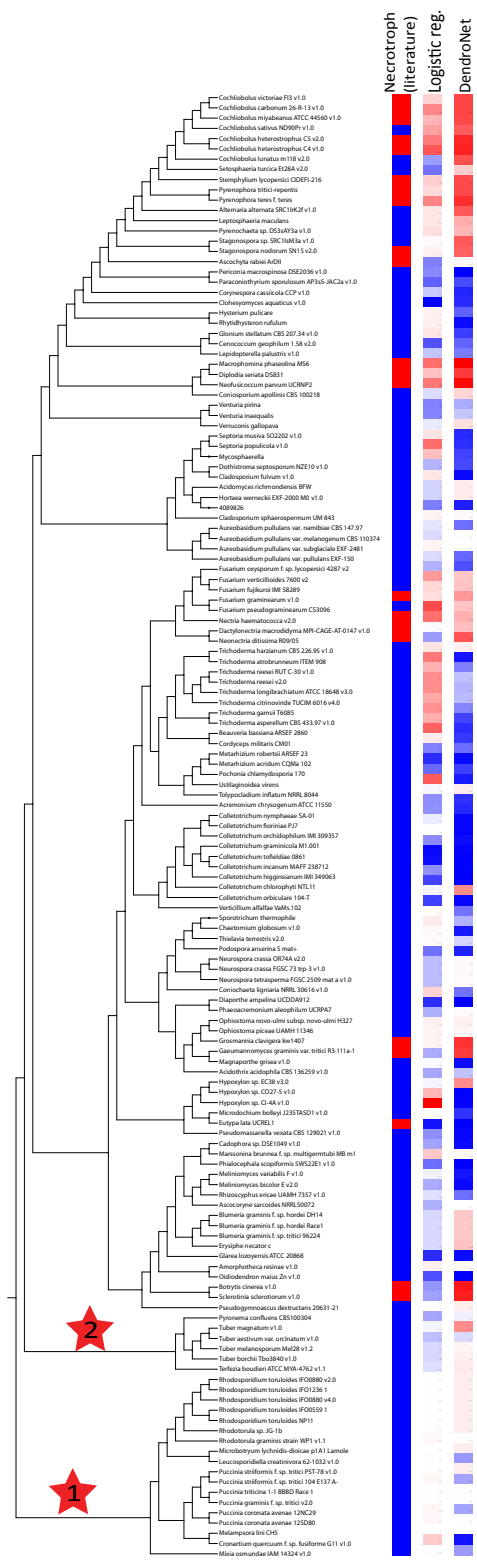

Fig. 5: The phylogenetic tree connecting the Necrotrophs dataset, with true classifications marked along with predictions made by DendroNet and the control Logistic Regression method. The two marked edges represent sites where the trained Dendronet model found it necessary to introduce notable large mutations to its model parameters.

trophic level identification in fungi. Some other areas where it is possible that the DendroNet method would provide value include the identification of transcription-factor binding sites using multi-species datasets, and the calculation of polygenic risk scores from a sufficiently genetically diverse population, provided they could be represented in a tree structure.

Furthermore with regards to future work, a more rigorous analysis of the amount of change in the weights of the models that are necessary for DendroNet to outperform traditional methods is warranted. The relative rarity of the changes observed in the Fungi trophic level model suggests that not too much evolutionary diversity may be required for the DendroNet model to be advantageous, but it would be ideal to quantify exactly where the limit is. We also intend to perform additional experimentation with larger, more complex model architectures, and study potential issues of convergence and memory footprint.

Acknowledgements

The authors wish to thank the following individuals for helpful discussions: Alex Butyaev, Chris Cameron, Samy Coloumbe, Dongjoon Lim, Faizy Ahsan, Nicolas Feau, Amaury Leroy, Ayrin Ahia-Tabibi, Zichao Yan, and Mohammad Nikou Saffat.

Funding

Funding for this work was provided through the bioSAFE project.

References

[1]Bengio, Y., Courville, A., & Vincent, P. (2012). Representation Learning: A Review and New Perspectives. Retrieved from http://arxiv.org/abs/1206.5538

[2]Wang, Z., Yeo, G. H. T., Sherwood, R., & Gifford, D. (2019). Disentangled Representations of Cellular Identity. In L. J. Cowen (Ed.), Research in Computational Molecular Biology (pp. 256–271). Cham: Springer International Publishing.Wang, Z., Yeo, G. H. T., Sherwood, R., & Gifford, D. (2019). Disentangled Representations of Cellular Identity. In L. J. Cowen (Ed.), Research in Computational Molecular Biology (pp. 256–271). Cham: Springer International Publishing.

[3]The genome portal of the Department of Energy Joint Genome Institute: 2014 updates. Nordberg H, Cantor M, Dusheyko S, Hua S, Poliakov A, Shabalov I, Smirnova T, Grigoriev IV, Dubchak I. Nucleic Acids Res. 2014,42(1):D26-31.

[4]Chen JH, Asch SM. Machine Learning and Prediction in Medicine - Beyond the Peak of Inflated Expectations. N Engl J Med. 2017 376(26):2507–2509. doi:10.1056/NEJMp1702071

[5]Gilpin L., Bau D., Yuan B., Bajwa A., Specter M., Michael, Kagal L. (2018). Explaining Explanations: An Overview of Interpretability of Machine Learning. 80-89. 10.1109/DSAA.2018.00018.

[6]Kingma D., Ba J. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[7]Elshawi R, Al-Mallah MH, Sakr S. On the interpretability of machine learning-based model for predicting hypertension. BMC Med Inform Decis Mak. 2019 19(1):146. Published 2019 Jul 29. doi:10.1186/s12911-019-0874-0

[8]Hsu PD, Lander ES, Zhang F. Development and applications of CRISPR-Cas9 for genome engineering. Cell. 2014 157(6):1262-1278. doi:10.1016/j.cell.2014.05.010

[9]Drouin, A., Giguère, S., Déraspe, M. *et al.* Predictive computational phenotyping and biomarker discovery using reference-free genome comparisons. BMC Genomics 17, 754 (2016). https://doi.org/10.1186/s12864-016-2889-6

[10]Wang, X., Huang, T., Schneider, J. (2014). Active Transfer Learning under Model Shift. Proceedings of the 31st International Conference on Machine Learning, in PMLR 32(2):1305-1313

[11]Raza H. , Prasad G., Li Y. (2014). Adaptive learning with covariate shift-detection for non-stationary environments. 2014 14th UK Workshop on Computational Intelligence, UKCI 2014 - Proceedings. 1-8. 10.1109/UKCI.2014.6930161.

[12]Earle, S., Wu, C., Charlesworth, J. *et al.* Identifying lineage effects when controlling for population structure improves power in bacterial association studies. Nat Microbiol 1, 16041 (2016). https://doi.org/10.1038/nmicrobiol.2016.41

[13]Alippi, Cesare & Boracchi, Giacomo & Roveri, Manuel. (2012). Just-in-time ensemble of classifiers. Proceedings of the International Joint Conference on Neural Networks. 1-8. 10.1109/IJCNN.2012.6252540.

[14]Wattam AR, Davis JJ, Assaf R, Boisvert S, Brettin T, Bun C, Conrad N, Dietrich EM, Disz T, Gabbard JL, Gerdes S, Henry CS, Kenyon RW, Machi D, Mao C, Nordberg EK, Olsen GJ, Murphy-Olson DE, Olson R, Overbeek R, Parrello B, Pusch GD, Shukla M, Vonstein V, Warren A, Xia F, Yoo H, Stevens RL. Improvements to PATRIC, the all-bacterial Bioinformatics Database and Analysis Resource Center. Nucleic Acids Res. 2017 Jan 4;45(D1):D535-D542. doi: 10.1093/nar/gkw1017. PMID: 27899627. PMCID: PMC5210524.

[15]Hamelin R.C. and Roe A.D. Genomic biosurveillance of forest invasive alien enemies: A story written in code. Evolutionary Applications (2019). https://doi.org/10.1111/eva.12853

[16]Dort E., Layne E. Feau N., Blanchette M., Hamelin R. FungiTrophDB: A curated database of fungi trophic levels. Submitted.

[17]Osbourn A. Gene Clusters for Secondary Metabolic Pathways: An Emerging Theme in Plant Biology. Plant Physiol. (2010) 154(2): 531–535.

[18]Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat Biotechnol. (2015) 33(8):831-8.

[19]Sul J.H., Martin L.S., Eskin E. Population structure in genetic studies: Confounding factors and mixed models. PLoS Genetics. (2018), https://doi.org/10.1371/journal.pgen.1007309

[20]Abadi M. , Agarwal A., Barham P. , ... Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viégas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., and Zheng X.. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[21]Felsenstein, J. (1985). Phylogenies and the Comparative Method. The American Naturalist, 125(1), 1-15.

[22]Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. & Lerer, A. (2017). Automatic Differentiation in PyTorch. NIPS 2017 Workshop on Autodiff, .