Neural Network for Biomedical Data with Structured Features

Author(s)
Affiliation(s)

Abstract

Neural networks have demonstrated strong capabilities of discovering potential complex patterns for predicting outcomes. They typically need to be trained on a large amount of data. However, biomedical studies often have limited sample sizes but large numbers of feature variables that may correlate, interact, and jointly affect outcomes. In this paper, we proposed *Peel Learning*, a novel neural network that incorporates the prior relationship among features. In each layer of learning, overall structure is peeled into multiple local substructures. Within the substructure, dependency among features is reduced by projecting children features onto parents' space and removing the redundant components. The overall structure is gradually reduced in size over layers and the parameters are optimized through backpropagation. We theoretically proved the improved upper bound of our model's complexity over ordinary neural networks. We also evaluated the performance of the method through simulations and applications to two real studies (1) to predict lung transplantation outcome using gene expression profiles in donors' lungs; (2) to predict median response time using cerebral blood flow from functional magnetic resonance imaging data. Our method showed improved prediction accuracy, especially in small data, compared with several existing methods.

Introduction

Recently deep learning methods (LeCun, Bengio, and Hinton 2015) have been increasingly developed, evolved, and applied to mine the patterns from complex data (LeCun, Bengio, and Hinton 2015; Schmidhuber 2015; Goodfellow et al. 2016). These methods are typically purely data-driven and assume little about feature structures. Therefore, they usually train a fairly large amount of data to reach a satisfactory model.

In biomedical studies, it is often challenging to collect such a large amount of training data. For example, the sample sizes of gene expression studies range from tens to hundreds but hundreds of thousands of interrelated genes are potential predictors. Conventional deep learning algorithms can barely make informative predictions with such small samples. Fortunately, feature variables might be correlated, forming a structured space. For example, genes regulate each others, and their directional relationship has been summarized in genetic regulatory networks. Within brain networks, large-scale neurons are connected through synapses and communicate through chemical signals. Either known or approximate structures can serve as prior knowledge to guide the learning process so that model searching path is shortened and optimal models can be obtainable even in small samples.

Several classes of machine learning methods, including l_1-norm regularization Lasso (Alelyani, Tang, and Liu 2013; Tibshirani 1996), decision tree (DT) (Friedman, Hastie, and Tibshirani 2001), and XGboost (Chen and Guestrin 2016) handle high-dimensional feature variables. These methods have been extended for grouped or correlated features, including group Lasso (GL) (Zhao et al. 2009; Toloşi and Lengauer 2011), tree structure group Lasso (TSGL) (Liu and Ye 2010; Kamkar et al. 2015), and non-negative Max-heap (MH) (Liu, Sun, and Ye 2011). GL and TSGL use penalization functions of group or correlation matrix to encourage the selection of correlated feature. MH uses a heredity principle defined by an ordered tree structure to select children features only at the presence of parent features.

To handle high dimensional features, a class of sparsity-inducing or group-sparsity regularization was introduced to neural networks to expedite gradient algorithms (Tartaglione et al. 2018; Scardapane et al. 2017). Another computationally intensive class, dropout neural network (DNN), uses a large number of different network architectures and randomly drop out (Srivastava et al. 2014) nodes during training, to reduce overfitting and improve generalization. The latter has shown more effective than the former (Goodfellow, Bengio, and Courville 2016). Along the line of group-sparsity regularizer, PASNet incorporated prior knowledge of gene pathway into layer construction, allows sparsity between pathway and hidden layers, and assumes all within-pathway genes connected to a hidden pathway node (Hao et al. 2018). Computationally, it applies sparse coding in randomly selected sub-networks. PASNet serves as a suitable baseline for our method because it has shown better predictions than DNN.

In this paper, we developed a neural network called *Peel Learning* (PL) for leveraging specific feature structures to predict outcomes. Our method considers feature values and inherent feature relationship as a combined input and per-

forms prediction over layers as in neural network. In each layer, feature values within a local structure are transformed and summarized as output features. The structure also evolves over layers while conserving the feature relationship.

It is worth mentioning that a group of work, such as GCN (Kipf and Welling 2016), GraphSAGE (Hamilton, Ying, and Leskovec 2017), and GAT (Veličković et al. 2017), deals with connection among subjects, and is fundamentally different from PL that deals with structured feature variables.

Our proposed PL method has several contributions, as highlighted here:

- Our model is built upon feature structure and the model structure is fixed. Layers and local substructures are determined by the initial feature structure, and each substructure is inherited and trimmed hierarchically from the previous layer. PL uses a peeling principle and decomposes features into less dependent components, which sufficiently exploits the *between-feature association* and minimizes *feature redundancy* in linear relation. The decomposition operation is built upon multiple subjects, which increases model robustness against individual variation.
- Compared with PASNet that connects within-pathway genes to the same node, PL allows a more hierarchical and deeper structure for the within-pathway genes, which helps improve computational efficiency and prediction accuracy.
- In theory, we have proved that our PL algorithm has lower model complexity upper bound compared with 2-layer ordinary neural network (NN) when feature structure is relatively sparse, which supports PL's efficiency.

Method

Notations and Model

Suppose there are n independent subjects in a study. For subject i ($i = 1, \ldots, n$), m features/predictors $x_1^i, \ldots, x_m^i$ and an outcome y^i are available. The goal of the study is to learn an optimal model to predict Y using X. We use a column vector $X_j = (x_j^1, \ldots, x_j^n)^\mathsf{T}$ to denote the values of the j^{th} ($j = 1, \ldots, m$) feature, and another column vector $Y = (y^1, \ldots, y^n)^\mathsf{T}$ to denote the outcome of all subjects.

If a feature X_j affects another feature X_k causally, i.e., the change of X_j leads to the change of X_k, we use a pair of indices (j, k) to denote such a directed link. An edge collection $\mathbf{E} = \{(j, k) : j \neq k\}$ contains all links or "*edges*" among X. For convenience, $\mathbf{E}$ is stored in an $m \times m$ binary adjacency matrix E with elements e_{ij} being 1 if $(j, k) \in \mathbf{E}$ and 0 otherwise.

We use $pa(k) = \{j : (j, k) \in \mathbf{E}\}$ to denote the parent index set of X_k, $ch(k) := \{l : (k, l) \in \mathbf{E}\}$ to denote the child index set of X_k, and J_k and S_k to denote the number of parents and the number of children of X_k, respectively. Figure 1 shows a simple tree structure of 3 layers, $\mathbf{E} = \{(1, 2), \ldots, (1, 5), (1, 9), (2, 6), (2, 7), (3, 8), (4, 8), (5, 9)\}$ contains 10 edges. Initially, X_2 has a single parent X_1 and two children X_6 and X_7, so $pa(2) = \{1\}$, $J_2 = 1, ch(2) = \{6, 7\}, S_2 = 2$; X_8 has two parents X_3 and X_4 but no child, so $pa(8) = \{3, 4\}$, $J_8 = 1, ch(8) = \emptyset$, and $S_8 = 0$.

For convenience, we summarize both feature values and feature-linked edges in a graph structure denoted as $D = (X, \mathbf{E})$. Currently, we assume D is directed acyclic, but the procedure can be modified to fit indirect or cyclic graph structure.

Given its parents' values, variable X_k is assumed to follow the distribution, $X_k = f_k(X_{k_1}, \ldots, X_{k_j}) + \epsilon_k$, where $k_1, \ldots, k_j = pa(k)$, ϵ_k is an independent random variable.

The distribution of outcome Y is assumed to follow:

$$Y = F(X_1, \ldots, X_m) + \epsilon_y = G(f_1, \epsilon_1, \ldots, f_m, \epsilon_m) + \epsilon_y, \tag{1}$$

where $F(.)$ and $G(.)$ can be any functions. Our goal is to estimate them through a deep learning procedure.

Peel Learning (PL) Method

PL considers both feature values of X and their relationship $\mathbf{E}$, as summarized in a graph $D = \{X, \mathbf{E}\}$. At layer 1, $D^{(0)} = D$ is the input and $D^{(1)} = \{X^{(1)}, \mathbf{E}^{(1)}\}$, also a DAG, is the output. At layer l, $D^{(l-1)}$ is the input and $m^{(l)}$ is the number of input feature variables. An output $D^{(l)} = \{X^{(l)}, \mathbf{E}^{(l)}\}$ is generated as follows.

Step 1. Around each feature, we define a two-layer substructure formed by this feature and its children features and call it a *local receptive tree* (LRT). The original large structure will be "peeled" into many such small structures, so that summary and tracing can be more computationally efficient. The links between features within each LRT are kept intact and will help recover the original structure after numerical operations. The "peeling" algorithm has demonstrated efficiency in multi-generation pedigree studies where genetic data were summarized per small nuclear families and passed to earlier generations (Elston and Steward 1971).

Within each LRT, each feature $X_k^{(l-1)}$ is projected onto its parents' space and the remaining component $\epsilon_k^{(l-1)}$ (so-called "residual" in regression models) is extracted. Without loss of generality, we start with a linear projection, i.e.,

$$\begin{aligned} \epsilon_k^{(l-1)} &= X_k^{(l-1)} - \hat{X}_k^{(l-1)} \\ &= X_k^{(l-1)} - H_k^{(l)} X_k^{(l-1)} \\ &= [I - H_k^{(l)}] X_k^{(l-1)}, \end{aligned} \tag{2}$$

where $H_k^{(l)} = P_k^{(l-1)} (P_k^{(l-1)\mathsf{T}} P_k^{(l-1)})^{-1} P_k^{(l-1)\mathsf{T}}$ is a projection matrix, $P_k^{(l-1)} = \{(X_1^{(l-1)}, \ldots, X_j^{(l-1)}, \ldots, X_{J_k}^{(l-1)}) : j \in pa(k)\}$ is a matrix with ordered columns and each column being values of a parent of feature k.

Compared with $X_k^{(l-1)}$, $\epsilon_k^{(l-1)}$ is less correlated with its parent(s) and thus all of its ancestral features. If the projection function matches the underlying relationship between parents and children (e.g., linear projection for linear relationship), $\epsilon_k^{(l-1)}$ and parents are expected to be independent/orthogonal. Furthermore, within a substructure, a child feature is only related to other child features through their parents, thus the

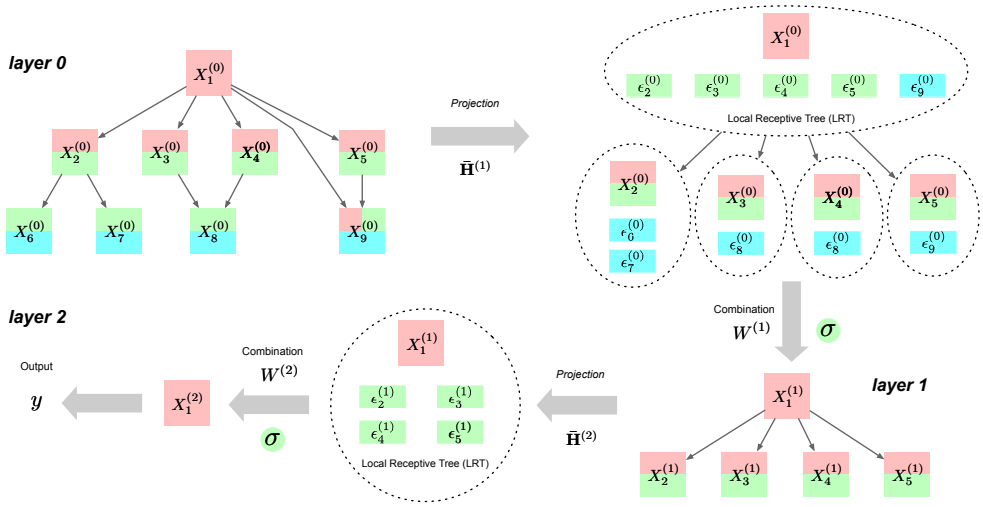

Figure 1: An example of feature structure evolution in PL. Each dash eclipse contains a local receptive tree.

residuals from all children are more independent of each other than before. The reduced dependency among these residuals would allow us to back-propagate more efficiently. The hat matrix H borrows other subjects' information (over n) to infer the strength of parent-child association, which may also reduce bias due to individual variation.

For all children features from the same set of parents, the original estimated matrix $H_k^{(l-1)}$ only needs to be calculated once within one layer. Because the projection procedure can be done simultaneously for all feature variaes, the computation addition over NN is minimal.

Step 2. For each child-bearing feature X_k, a linear combination of its value and its children's residuals is generated and then transformed by the activation function to become a new "feature", also the output value for this feature and the input for next layer. Specifically, $X_k^{(l)}$ is generated by

$$X_k^{(l)} = \sigma(z_k^{(l)}) = \sigma(w_{k,0}^{(l)} X_k^{(l-1)} + \sum_{j \in ch(k)} w_{k,j}^{(l)} \epsilon_j^{(l-1)} + b_k^{(l)}), \tag{3}$$

where transient variable $z_k^{(l)}$ is linear combination of $X_k^{(l-1)}$ and all ϵ, $w_{k,j}^{(l)}$ s are weight parameters, and $b_k^{(l)}$ is a bias parameter. The update of $X^{(l)} = (X_1^{(l)}, \ldots, X_{m_l}^{(l)})$ can be written in a matrix notation as:

$$X^{(l)} = \sigma(Z^{(l)}) = \sigma(W^{(l)} \bar{\mathbf{H}}^{(l)} X^{(l-1)} + \mathbf{b}^{(l)}), \tag{4}$$

where transient vector is $Z^{(l)} = (z_1^{(l)}, z_2^{(l)}, \ldots, z_{m^{(l)}}^{(l)})$, weight vector is $W^{(l)} = (w_{1,1}^{(l)}, \ldots, w_{1,S_1+1}^{(l)}, \ldots, w_{m^{(l)},1}^{(l)}, \ldots, w_{m^{(l)},S_{m^{(l)}}+1}^{(l)})$, intercept vector is $\mathbf{b}^{(l)} = (b_1^{(l)}, \ldots, b_{m^{(l)}}^{(l)})$, and $\bar{\mathbf{H}}^{(l)}$ is an extended estimated matrix, defined in the following equation:

$$\bar{\mathbf{H}}^{(l)} = \begin{pmatrix} \mathbf{I}^{(l)} \\ \mathbf{I}^{(l)} - \mathbf{H}^{(l)} \end{pmatrix}_{2m^{(l-1)} \times m^{(l-1)}} \tag{5}$$

where $\mathbf{I}^{(l)}$ is an $m^{(l-1)} \times m^{(l-1)}$ identity matrix, $\mathbf{H}^{(l)}$ is a diagonal matrix formed by blocks $H_1^{(l)}, \ldots, H_{m^{(l-1)}}^{(l)}$.

We use Figure 1 to illustrate how steps 1 and 2 are performed. At layer 0, as the children of a single parent X_2, X_6 and X_7 are projected/regressed on X_2 to obtain ϵ_6 and ϵ_7. Then a feature X_2 in layer 1 is formed by $X_2^{(1)} = \sigma(w_{2,0}^{(1)} X_2^{(0)} + w_{2,1}^{(1)} \epsilon_6^{(0)} + w_{2,2}^{(1)} \epsilon_7^{(0)} + b_2^{(1)})$. As a feature with multiple parents, X_8 is projected on both X_3 and X_4 to obtain ϵ_8 and then $X_3^{(1)} = \sigma(w_{3,0}^{(1)} X_3^0 + w_{3,1}^{(1)} \epsilon_8^{(0)} + b_3^{(1)})$ becomes the new feature for X_3 in layer 1. The features with multi-generation parents are complicated, the value of X_1 is formed by $X_1^{(1)} = \sigma(w_{1,0}^{(1)} X_1^{(0)} + w_{1,1}^{(1)} \epsilon_2^{(0)} + w_{1,2}^{(1)} \epsilon_3^{(0)} + w_{1,3}^{(1)} \epsilon_4^{(0)} + w_{1,4}^{(1)} \epsilon_5^{(0)} + w_{1,5}^{(1)} \epsilon_9^{(0)} + b_1^{(1)})$.

A new edge collection $\mathbf{E}^{(l)}$ is formed by keeping the indices of all child-bearing features and their links with parents in $\mathbf{E}^{(l-1)}$, i.e. $\mathbf{E}^{(l)} = \{(j,k) : (j,k) \in \mathbf{E}^{(l-1)}, ch(k) \neq \emptyset\}$. So $\mathbf{E}^{(l)}$ is a subset of $\mathbf{E}^{(l-1)}$. A childless feature does not form a summary by itself and thus does not appear in the next layer. This procedure is similar to trimming the bottom branches of a tree. A special exception applied to singletons (features without parent and child) that they will be reserved in each layer, i.e., the summary around a singleton is simply a function of itself.

Step 3. At the end of layer L, a small structure $D^{(L)}$ with a small number of features and edges remains. They will be used to generate $\hat{Y}$, i.e., $\hat{Y} = W^{(L+1)} \bar{\mathbf{H}}^{(L+1)} X^{(L)} + \mathbf{b}^{(L+1)}$. If the final features do not have any child, $\hat{Y}$ is simplified to $W^{(L+1)} \mathbf{X}^{(L)} + \mathbf{b}^{(L+1)}$.

Step 4. The final objective is to minimize a cost/loss function, e.g. $\mathcal{C}(Y, \hat{Y}) = \frac{1}{2n} \sum_i (Y_i - \hat{Y}_i)^2$ for a continuous outcome. In each iteration, the values of W and b are updated by the stochastic gradient descent approach.

Note that ordinary neural network (NN) only uses X as the input of each layer, ignoring $\mathbf{E}$.

Peel Learning Algorithm Summary. For large sample

sizes, mini-batching (Hinton 2010) can be used as the estimate $\bar{\mathbf{H}}$ improves with additional samples. The PL procedure is summarized in Algorithm 1.

Algorithm 1 PEEL LEARNING

1: **Initialize:** hierarchical graph D, training epochs T, learning step η, number of layers L, activation function σ, and convergence threshold γ.
2: **Initialize:** weights $W \leftarrow N(0,1)$, biases $\mathbf{b} \leftarrow N(0,1)$.
3: **for** epoch $t = 1$ **to** T **do**
4: **for** layer $l = 1$ **to** L **do**
5: Based on structure $D^{(l)}$, project each child feature onto its parents' space and generate residual values for each child. Then for each feature, combine its values and the remaining residual components of its children, i.e. $\bar{\mathbf{H}}^{(l+1)} X^{(l)}$, where $\bar{\mathbf{H}}^{(l+1)}$ is defined in Equation 5.
6: Calculate new feature matrix $X^{(l+1)}$

$$X^{(l+1)} = \sigma(W^{(l+1)} \bar{\mathbf{H}}^{(l+1)} X^{(l)} + \mathbf{b}^{(l+1)})$$

7: The new structure $D^{(l+1)}$ is automatically inherited from $D^{(l)}$ by removing childless features. The edge collection is in reduced dimensions: $\mathbf{E}^{(l)} = \{(j,k) : (j,k) \in \mathbf{E}^{(l-1)}\ ,\ ch(k) \neq \emptyset\ \}$
8: Calculate the total cost $\mathcal{C}^{(l)}$ and check whether the loss difference $|\mathcal{C}^{(l)} - \mathcal{C}^{(l-1)}| \leq \gamma$. If yes, algorithm stops; Else, continue to step 4.
9: Update weight and bias parameters $W^{(l)}$ and $\mathbf{b}^{(l)}$. Continue with the next epoch.

The Model Complexity of Peel Learning

In this section, we will provide the the upper bound of the PL's model complexity.

Deep neural network aims to minimize the empirical risk $\mathcal{R}_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{C}(f(X_i), y_i)$, where $(X_i, y_i)_{i=1}^{n}$ is the training set, $\mathcal{C}$ is the loss function. This $\mathcal{C}$ can be any loss functions such as l_2 loss or cross entropy loss. $f(X;\boldsymbol{\theta})$ is a hypothesis with $\boldsymbol{\theta}$ being a vector of all parameters. With the same training error, simple solutions generalize better than complex solutions, which relies on whether the loss optimizer can converge towards low-complexity solution areas (Wu, Zhu et al. 2017). Some theoretical work has focused on the geometry of the loss function around a global minimum (Wu, Zhu et al. 2017; Zhang et al. 2016). The key is that the optimal solution needs to lie in relatively flat valleys (Hirsch, Devaney, and Smale 1974) of the loss function to be able to generalize well.

Here we show that for a 2-layer PL, the expected value of the derivative with respect to the input X, which reflects the spatial fluctuation of f (Wu, Zhu et al. 2017), are bounded and that the upper bound is smaller than that in the corresponding 2-layer NN. The following theorem provides the model complexity of 2-layers' PL's upper bound.

Theorem 1 *The prediction $f(x;\boldsymbol{\theta})$ derived from a 2-layer PL model can be written as $f(X;\boldsymbol{\theta}) := W^{(2)}\sigma(W^{(1)}\bar{\mathbf{H}}X + \mathbf{b}^{(1)})$, where $W^{(1)}$ and $W^{(2)}$ are the weight matrices in layers 1 and 2, $\bar{\mathbf{H}}$ is the extended hat matrix as defined in Equation (5), and $\mathbf{b}^{(1)}$ is the bias vector in layer 1. If $\kappa \leq \sqrt{m/16} - 1$, then*

$$\begin{aligned} &2\mathbb{E}_X \left\|\nabla_X f(\mathbf{x})\right\|_2^2 \\ &\quad \leq (2m + 2 \times \text{rank}(X)) \left\|W^{(1)}\right\|_F^4 + \|I_{\mathbf{b}^{(1)}}\|_F^2 \end{aligned} \tag{6}$$

where $\kappa = |E|/m^{(1)}$ denotes the measure of the model *sparsity*, $|E|$ is the number of edges in this structure, $m^{(1)}, m$ is the number of features in layer 1, and layer 0, respectively, $I_{\mathbf{b}^{(1)}}$ is the Fisher information matrix with respect to $\mathbf{b}^{(1)}$, and $\|\cdot\|_F$ is the Frobenius norm.

Proof: See Supplement .

Theorem 1 establishes the relationship between the spatial fluctuation of f, measured by the norm of expected input gradient, and the Frobenius norm of the weight matrix and the Fisher information matrix of $\mathbf{b}^{(1)}$. This upper bound is smaller than that for the 2-layer NN (Wu, Zhu et al. 2017) as long as the model is not too dense than the threshold κ, which is a mild condition to be satisfied. If the sparsity κ is less than the order $\mathcal{O}(\sqrt{m})$, the generalization performance of PL's solution is better than NN, when utilize the prior structure information $\bar{\mathbf{H}}$.

Corollary 1 *For a prediction f derived from a 2-layer PL, we have*

$$\begin{aligned} &2\mathbb{E}_X \left\|\nabla_X f(\mathbf{x})\right\|_2^2 \\ &\quad \leq \left\|\nabla^2_{\mathbf{b}^{(1)}}\mathcal{C}\right\|_F^2 + (2m + 2 \times \text{rank}(X)) \left\|W^{(1)}\right\|_F^4 \\ &\quad + 2R_\sigma D\sqrt{\mathcal{C}} + \mathcal{O}(\sqrt{\frac{\mathbf{Var}\left\|\nabla_X f\right\|_2^2}{n}}) \end{aligned} \tag{7}$$

where $R_\sigma = \|\sigma''\|_\infty$, $D = \max_k |w_k^{(2)}| \left\|\mathbf{w}_k^{(1)}\right\|_2^2$ and $\nabla^2_{\mathbf{b}^{(1)}}\mathcal{C}$ denotes the Hessian matrix with respect to $\mathbf{b}^{(1)}$, and the last term is the Monte Carlo approximation error, which is closely related to the complexity of the solution $\hat{f}$.

Proof: See Supplement B.

The upper bound of corollary 1 links to the landscape of $\mathcal{C}$. Note that the last term is ignorable if the sample size in the training set is large enough. For many activation functions including the hyperbolic tangent function, R_σ is small finite, so the boundary is largely determined by the Frobenius norm of Hessian $\nabla^2_{\mathbf{b}^{(1)}}\mathcal{C}$. As long as $(2m+2rank(X))\left\|W^{(1)}\right\|_F^4 + 2\max_k |w_k^{(2)}| \left\|\mathbf{w}_k^{(1)}\right\|_2^2$ is small, low-complexity solutions will lie in the areas with small Hessian.

The time complexity of NN, PL and PASNet is shown in Supplement C.

Real Data Experiments

Lung Transplantation Study (Given Features' Structures)

Data Description. The data were collected through the Prospective Registry of Outcomes in Patients Electing Lung Transplantation study at xxx (for review) between October 2011 and December 2017. Before surgery, a $1cm^2$ lung biopsy was obtained from the periphery of the lung using a mechanical stapler (Anraku et al. 2008) and RNA was extracted using a standard protocol. Then whole-genome gene expression was obtained using the Affymetrix Human Gene 2.1 ST Array (Gellert et al. 2011). Raw data were corrected for background, normalized using the robust multi-array average (RMA) method, and summarized at 23,819 gene-levels. The primary outcome is binary primary graft dysfunction (PGD) (grade 3) in the 1st 48-72 hours following lung transplantation (Christie et al. 2005, 2010). Among 113 enrolled subjects, 28 (24.8%) developed PGD at post-surgery 48-72 hours (Cantu et al. 2020).

We intended to predict recipients' PGD status using the pre-operative donors' expression in immunology pathways : (i) Chemokine signaling pathway (185 genes); (ii) Toll-like receptor signaling pathway (102 genes); (iii) JAK-STAT signaling pathway (153 genes); (iv) Nod-like receptor signaling pathway (64 genes); (v) Graft versus host disease pathway (28 genes); (vi) Primary immunodeficiency pathway (34 genes). The pathway structures are shown in Supplement.

Analysis Method. Within each pathway, we used PL to derive the prediction and compared it with Lasso, GL, TSGL, MH, DT, XGboost, NN, and PASNet. Each regulatory pathway was extracted from GenomeNet Database Resources and the functional/directional relationship among genes has been functionally validated (Kanehisa and Goto 2000; Kanehisa et al. 2009). Connected genes generally have strongly correlated expressions and unconnected has weaker expressions. The same graph structure was used for GL, TSGL, MH, PASNet, and PL. Across six pathways, the number of layers ranged from 4 to 11.

We used cross-entropy as the loss function, hyperbolic tangent function (tanh) as the activation functions for PL and NN, area under the receiver-operating-characteristic curve (AUC) and optimal prediction accuracy (ACC) as the evaluation criteria. ACC is defined as the percentage of both positive and negative correct predictions out of all subjects. For each fitted model, the optimal ACC was determined based on all cut-off probabilities. Model parameters for all methods were optimized to minimize MSE through grid search and *5-fold cross-validation*. (The final values were in Supplement).

Results. Figure 2 shows the AUC and optimal ACC of PGD prediction from all methods using all gene expressions within each pathway. The standard errors are shown in Supplement. Compared with Lasso and DT, PL consistently had 3.52% to 18.55% larger AUCs. GL, TSGL, and MH had similarly and slightly larger AUCs than Lasso and DT, but 2.37% to 8.15% smaller than PL. PL outperformed PASNet by 3.09% to 15.40%, NN by 0.14% to 5.08%, except in the Jak Stat pathway, where NN had slightly better AUC than PL partially because JAK-STAT pathway had genes less connected (Supplement Figure 5). PASNet and XGboost had larger variation and smaller AUCs than PL. PASNet performed worse as the number of features become smaller, but there was no clear trend for XGboost.

Lasso in average had worst ACC (5.61-15.08% lower than PL). GL, TSGL, MH, DT, and XGboost were in the similar cluster with 3.62% to 6.62% smaller ACC than PL. PL outperformed NN and PASNet from 0.14% to 4.35% and from 3.39% to 17.84% separately, except in the JAK-STAT and graft vs. host disease pathwnays. In JAK-STAT pathway with more loosely connected genes, NN had slightly larger AUC (0.94%) than PL; in graft vs. host disease pathway with smaller number of genes, NN had larger ACC (1.10%) than PL.

Overall, PL had either optimal or sub-optimal AUCs or ACCs. Besides,the *s.d.* of AUCs and ACCs in Supplement table 5 and table 5 also show that PL had sub-optimal or sub-optimal standard deviation. As we expected, Lasso-like methods, including Lasso, GL, TSGL and MH, performed much worse than others because they extracted information on global or group levels. DT and XGboost performed slightly better because they incorporated multiple levels of the features variables. Neural network based methods, NN and PASNet, were the best among the baselines because in their layers, multiple nonlinear functions were used to estimate the underlying function. Besides, PASNet used the gene-pathway information through the connections between genes and pathway were uniform and generic. Instead, PL used more specific structures to incorporate the gene-gene correlation strength and gradually remove the redundancy along the relational directions, so it is more efficient for neural network to extract valuable information.

Among the 9 pathways, Toll-like receptor signaling pathway has the most prevailing evidence related to lung transplant outcomes (Cantu et al. 2020), for which PL had noticeably better predictions than others.

Brain Image Study (Approximate Features' Structures)

Data Description. A total of 70 healthy adults participated in a 5-day and 4-night in-laboratory experiment (Fang et al. 2015; Yang et al. 2018). Institutional review board approval and informed consent were obtained at xxx (for review). Every subject slept for 9 hours during night 1; 54 subjects were deprived for sleep during night 2 and fully recovered during nights 3 and 4; the remaining 16 controls slept for 8 hours at nights 2-4. A 10-minute psychomotor vigilance test (PVT) and functional magnetic resonance imaging (fMRI) scan were conducted on day 2 and day 5.

The outcome of interest was the median reaction time (MRT, in milliseconds) from PVT. The goal is to predict MRT using regional cerebral blood flow (CBF) obtained through fMRI. Nine candidate regions were known to be closely associated with MRT (Drummond et al. 2005), including bilateral frontal cortex (Frontal Mid L and R), bilateral supplementary motor areas (Supp Motor Area L and R), bilateral inferior Parietal cortex (Parietal Inf L and R), bilateral Putamen (Putamen L and R), and left Caudate (Caudate L).

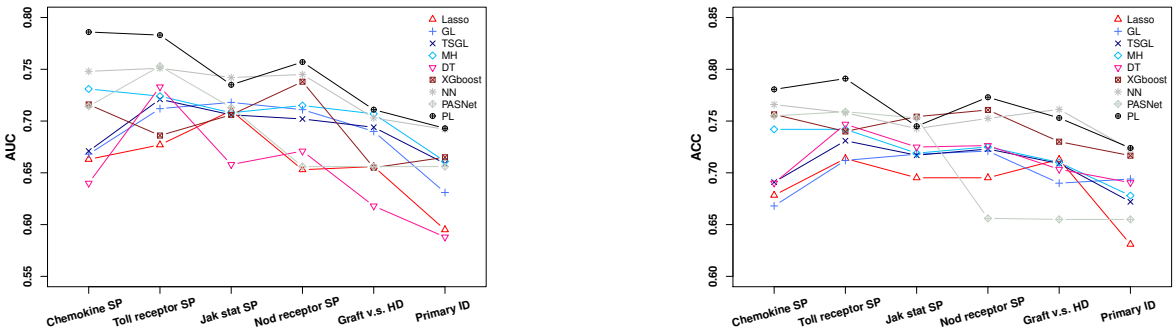

Figure 2: Left: AUC of each method on the prediction of lung transplantation outcome task using genes within 6 different pathways. Right: ACC of each method on the prediction of lung transplantation outcome task using genes within 6 different pathways. The standard deviation of AUC and ACC are in Supplement (table 5) and (table 6)

Analysis Method. Imaging data were preprocessed and quantitative CBF were mapped for each subject using Statistical Parametric Mapping, ASLtbx, and fMRI-Grocer toolboxes. CBF maps were normalized to standard brains using the Montreal Neurological Institute template and then segmented into 116 brain regions per the Automated Anatomical Labeling (AAL) template.

For each of the 9 regions, we derived a prediction model with day-2 MRT as response and image voxel value of CBF during the day-2 PVT as feature variables, using the 8 aforementioned baseline methods and PL. To increase reproducibility, we applied the same model to predict the day-5 MRT using the day-5 CBF.

Intuitively, the features were assumed to have a structured relationship. Because there was no ready-to-use structure, we estimated it using the Gaussian Graphic Lasso (GGL) method (Zhao et al. 2012) (the parameter setup in Supplement). The estimated structures were then used for GL, TSGL, MH, PASNet, and PL. Across the 9 structures, the number of features ranged from 100 to 5000 and the number of layers from 3 to 8. We used MSE as the loss function, tanh as the activation function, final MSE as the evaluation criteria.

Result. Among the 9 regions, PL results in the smallest MSE for the prediction of MRT in 8 regions compared with other methods (Table 1). The best MSEs from GL, TSGL, MH were 6.21, 5.76, and 6.91 respectively. The best MSEs from Lasso, DT, and XGboost were 6.86, 4.36, and 3.72 respectively. PL decreased MSE around 8.55% to 53.62% compared with the best method in each region. In each region, PL performed better compared with the NN method due to added structure information (MSE decreased by 20.63% - 72.80%). In the Parietal Inf L region, the inferred correlations among all voxels were very sparse (1% of all possible pairwise connections), so PL gains little useful information over others. Besides, the PL method performed better compared with the PASNet due to the robustness of the structure information (MSE decreased by 19.84% - 65.67%). These results suggest that PL had improved predictions over other methods. PL also had optimal or sub-optimal variation compared with other methods (Supplement table 7).

Simulation

Design

To understand how much model prediction can improve with additional feature structure, we conducted a series of simulations to compare three neural networks NN, PASNet, and PL. NN does not use any feature correlation at all and PASNet considers fully connected feature within a group/pathway. In contrast, PL takes advantage of specific feature correlations. We used three representative tree structures with one founder feature and a fixed number of children $s = 2, 5, 10$ for each parent (Supplement Figure 3). The binary tree ($s = 2$) represents a simple structure with less dependency while the quinary ($s = 5$) and decimal trees ($s = 10$) have more dependency among features. The total number of feature variables m was fixed and the number of layers was determined by the structure (here, number of generations).

The values of the founder feature were simulated from the normal distribution $N(0, 1)$. The values of other features were generated per a top-down structure. Specifically, the value of each child was generated from a normal distribution with the mean equals to a linear or nonlinear (sin) function of its parents' values. The strength of the parent-child association was determined by γ, i.e., the proportion of variance of X_k explained by their parents was set to be $20\%, 50\%, 80\%$ and the total variance of X_k was set to be 1.

After all X were generated, the outcome Y was generated as a $linear$ or sin function of the summation of randomly selected 1% of X.

We also varied the number of input variables m from 200, 1000, to 5000, and the sample size n from 100, 200, to 500. We generated 50 datasets for each set of parameters. We chose tanh as the activation function for NN and PL. NN used 3 layers: [1000, 1000, 10]. PASNet used 3 layers: [1000, 1, 1], as recommended. PL automatically determined the number of layers per graph structure. The batch size was the training sample size for PL and 10 for NN and PASNet. The optimal learning rate was 0.05, 0.0001, and 0.1 for NN,

Table 1: *MSE* (10^3ms^2) of the optimal model prediction of mean response time in fMRI study.

MSE	S Motor Area L	S Motor Area R	Parietal Inf L	Parietal Inf R	Caudate L	Putamen L	Putamen R	Frontal Mid L	Frontal Mid R
Lasso	10.57	14.71	7.94	8.05	13.48	12.45	6.86	9.06	12.70
GL	9.37	9.82	10.18	16.05	6.48	6.30	6.21	10.55	10.47
TSGL	9.31	8.91	10.44	13.41	6.76	5.76	5.90	10.73	11.17
MH	20.30	23.26	37.48	26.67	6.91	12.57	15.10	37.67	36.10
DT	14.20	5.54	8.03	4.89	18.41	7.97	4.36	11.47	7.76
XGboost	3.86	3.94	**4.08**	3.86	3.72	4.01	3.81	4.20	4.07
NN	3.87	3.68	7.45	7.83	4.24	2.30	2.75	13.39	12.72
PASNet	4.41	3.75	4.20	5.21	3.46	2.13	2.42	8.01	5.26
PL	**1.79**	**1.79**	5.24	**3.53**	**2.48**	**1.45**	**1.51**	**2.75**	**3.46**

PASNet, and PL, respectively, to favor each algorithm. SGD was the optimizer for all methods.

To avoid overfitting, we divided each dataset into training, validation, and testing sets with an 8:1:1 ratio. We used MSE as the loss function. Smaller MSE values in the testing set generally indicate better prediction performance. We chose the initial learning rate from 0.01 to 1, 0.0001 to 0.1, 0.1 to 0.1, and then narrowed down to more precise values with an interval size of 0.01, 0.0001, 0.1 for NN, PASNet, and PL respectively.

Last, we tested the robustness of PL when only partial edge information is available and some ambiguous directions were misspecified. In this setting, we randomly selected $10\%, 20\%, 30\%, 40\%, 50\%$ of the edges and assumed the opposite/wrong directions.

Result

Supplement Table 2 shows the mean and s.d. of MSE of NN, PASNet, and PL out of 50 simulations, for $m = 1000$ and $n = 200$. PL had the smaller mean MSE consistently among the three methods regardless of the complexity of the input dependency structures. When X-X and X-Y relationships were linear. The MSEs in nonlinear scenarios were larger than linear scenarios because of increased relation complexity.

For all nonlinear scenarios, each method performed better for binary trees than for other trees; while for linear scenarios, NN, PASNet, and PL favoring decimal trees and fewer layers.

Supplement Table 3 shows the mean and s.d. of MSE from NN and PL in simulated data. In general, as m decreased or n increased, mean MSE decreased, implying a more precise prediction. At $n = 200$, PL had a mean MSE reduction of 10%-21% compared with NN. However, as n increased, PL and NN resulted in similar MSEs. When both m and n were large, NN performed slightly better than PL.

Supplement Table 4 shows the mean and s.d. of MSE in PL when the data structure was misspecified. With more misspecified directions (e increased), the prediction of PL became less accurate; however, as long as the link between feature variables was specified, the performance of PL was stable regardless of correct/wrong directions, except for a simple binary tree with a high error rate (40% or 50%).

Conclusion and Discussion

As a proof of concept, our study provided a straightforward peeling approach to improve predictions when features are structured. The PL proceeds over fixed layers with reserved relations among features. In real studies and simulations, PL outperformed other methods in almost all scenarios, regardless of sample size, variable number, linear or nonlinear feature relationships. Even when relational directions were mis-specified, PL's results remained robust. If the relations among the features are inferred in mistake, the benefit of PL over normal NN diminishes. At extreme cases where the features were either loosely or densely linked, PL performed similarly as NN or PASNet.

We also proved that the upper bound of the 2-layer PL's model complexity is less than that of the 2-layer NN if the relational links are relatively sparse. It also means that for the same model complexity, PL requires smaller sample sizes to achieve the same performance as NN, which was empirically demonstrated in both real data and simulations.

Our PL is similar to Bayesian methods, where prior feature relationship plays a bigger role in smaller samples. When sample size (n) is much larger than the number of predictors (m), simpler methods such as NN may work equally well.

In PL, features within an LRT are decomposed through a projection on parents' surface. Our current choice of linear projection balances easy propagation and over-fitting, though it is possible to use a variety of functions for nonlinear relations. Within each LRT, the original values of parents rather their post-projection residuals are used for summary because using all residuals may amplify the effect of over projection and lead to sub-optimal performance.

Another group of methods for high dimensional features selected relevant features before running NN, such as, DNP (Liu et al. 2017) and Diet Network (Romero et al. 2016). We will conduct future studies to investigate this topic.

Non-Euclidean embeddings like Poincaré learning (Nickel and Kiela 2017) can learn and discover latent hierarchical relationship in large-scale taxonomies and graph data. Such method provides a state-of-the-art option for complimenting and improving PL and methods alike.

PL can be generalized to other structured data, such as, natural language data with similar structure, normalized natural images, and sequential data collected at multiple time points.

References

Alelyani, S.; Tang, J.; and Liu, H. 2013. Feature Selection for Clustering: A Review. *Data Clustering: Algorithms and Applications* 29: 110–121.

Anraku, M.; Cameron, M.; Waddell, T.; Liu, M.; Arenovich, T.; Sato, M.; Cypel, M.; Pierre, A.; De Perrot, M.; Kelvin, D.; et al. 2008. Impact of human donor lung gene expression profiles on survival after lung transplantation: a case-control study. *American Journal of Transplantation* 8(10): 2140–2148.

Cantu, E.; Yan, M.; Suzuki, Y.; Buckley, T.; Galati1, V.; Majeti1, N.; Bermudez, C.; Diamond, J.; Christie, J.; and Feng, R. 2020. Pre-Procurement In Situ Donor Lung Tissue Gene Expression Classifies Primary Graft Dysfunction Risk. *American Journal of Respiratory and Critical Care Medicine* In print.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794. ACM.

Christie, J. D.; Bellamy, S.; Ware, L. B.; Lederer, D.; Hadjiliadis, D.; Lee, J.; Robinson, N.; Localio, A. R.; Wille, K.; Lama, V.; et al. 2010. Construct validity of the definition of primary graft dysfunction after lung transplantation. *The Journal of Heart and Lung Transplantation* 29(11): 1231–1239.

Christie, J. D.; Van Raemdonck, D.; De Perrot, M.; Barr, M.; Keshavjee, S.; Arcasoy, S.; and Orens, J. 2005. Report of the ISHLT Working Group on Primary Lung Graft Dysfunction part I: introduction and methods. *The Journal of heart and lung transplantation* 24(10): 1451–1453.

Drummond, S. P.; Bischoff-Grethe, A.; Dinges, D. F.; Ayalon, L.; Mednick, S. C.; and Meloy, M. 2005. The neural basis of the psychomotor vigilance task. *Sleep* 28(9): 1059–1068.

Elston, R.; and Steward, J. 1971. A general model for the genetic analysis of pedigree data. *Human Heredity* 21(6): 523–42.

Fang, Z.; Spaeth, A. M.; Ma, N.; Zhu, S.; Hu, S.; Goel, N.; Detre, J. A.; Dinges, D. F.; and Rao, H. 2015. Altered salience network connectivity predicts macronutrient intake after sleep deprivation. *Scientific reports* 5: 8215.

Friedman, J.; Hastie, T.; and Tibshirani, R. 2001. *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Friedman, J.; Hastie, T.; and Tibshirani, R. 2008. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9(3): 432–441.

Gellert, P.; Teranishi, M.; Jenniches, K.; De Gaspari, P.; John, D.; grosse Kreymborg, K.; Braun, T.; and Uchida, S. 2011. Gene Array Analyzer: alternative usage of gene arrays to study alternative splicing events. *Nucleic acids research* 40(6): 2414–2425.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Goodfellow, I.; Bengio, Y.; Courville, A.; and Bengio, Y. 2016. *Deep learning*, volume 1. MIT press Cambridge.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, 1024–1034.

Hao, J.; Kim, Y.; Kim, T.-K.; and Kang, M. 2018. PASNet: pathway-associated sparse deep neural network for prognosis prediction from high-throughput data. *BMC bioinformatics* 19(1): 510.

Hinton, G. 2010. A practical guide to training restricted Boltzmann machines. *Momentum* 9(1): 926.

Hirsch, M. W.; Devaney, R. L.; and Smale, S. 1974. *Differential equations, dynamical systems, and linear algebra*, volume 60. Academic press.

Kamkar, I.; Gupta, S. K.; Phung, D.; and Venkatesh, S. 2015. Stable feature selection for clinical prediction: Exploiting ICD tree structure using Tree-Lasso. *Journal of biomedical informatics* 53: 277–290.

Kanehisa, M.; and Goto, S. 2000. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic acids research* 28(1): 27–30.

Kanehisa, M.; Goto, S.; Furumichi, M.; Tanabe, M.; and Hirakawa, M. 2009. KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic acids research* 38(suppl_1): D355–D360.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* .

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature* 521(7553): 436.

Liu, B.; Wei, Y.; Zhang, Y.; and Yang, Q. 2017. Deep Neural Networks for High Dimension, Low Sample Size Data. In *IJCAI*, 2287–2293.

Liu, J.; Sun, L.; and Ye, J. 2011. Projection onto a nonnegative max-heap. In *Advances in Neural Information Processing Systems*, 487–495.

Liu, J.; and Ye, J. 2010. Moreau-Yosida regularization for grouped tree structure learning. In *Advances in neural information processing systems*, 1459–1467.

Nickel, M.; and Kiela, D. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, 6338–6347.

Romero, A.; Carrier, P. L.; Erraqabi, A.; Sylvain, T.; Auvolat, A.; Dejoie, E.; Legault, M.-A.; Dubé, M.-P.; Hussin, J. G.; and Bengio, Y. 2016. Diet networks: Thin parameters for fat genomics. *arXiv preprint arXiv:1611.09340* .

Scardapane, S.; Comminiello, D.; Hussain, A.; and Uncini, A. 2017. Group sparse regularization for deep neural networks. *Neurocomputing* 241: 81–89. ISSN 0925-2312. doi: 10.1016/j.neucom.2017.02.029. URL http://dx.doi.org/10.1016/j.neucom.2017.02.029.

Schmidhuber, J. 2015. Deep learning in neural networks: An overview. *Neural networks* 61: 85–117.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1): 1929–1958.

Tartaglione, E.; Lepsøy, S.; Fiandrotti, A.; and Francini, G. 2018. Learning Sparse Neural Networks via Sensitivity-Driven Regularization. *CoRR* abs/1810.11764. URL http://arxiv.org/abs/1810.11764.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

Toloşi, L.; and Lengauer, T. 2011. Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics* 27(14): 1986–1994.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* .

Wu, L.; Zhu, Z.; et al. 2017. Towards Understanding Generalization of Deep Learning: Perspective of Loss Landscapes. *arXiv preprint arXiv:1706.10239* .

Yang, F. N.; Xu, S.; Chai, Y.; Basner, M.; Dinges, D. F.; and Rao, H. 2018. Sleep deprivation enhances inter-stimulus interval effect on vigilant attention performance. *Sleep* 41(12): zsy189.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* .

Zhang, Y.; and Shen, X. 2010. Model selection procedure for high-dimensional data. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 3(5): 350–358.

Zhao, P.; Rocha, G.; Yu, B.; et al. 2009. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics* 37(6A): 3468–3497.

Zhao, T.; Liu, H.; Roeder, K.; Lafferty, J.; and Wasserman, L. 2012. The huge package for high-dimensional undirected graph estimation in R. *Journal of Machine Learning Research* 13(Apr): 1059–1062.

Supplementary Material for "Neural Network for Biomedical Data with Structured Features"

Proof of Theorem 1

Proof. The model defined by the 2-layer PL can be written as,

$$\begin{aligned} f(X;\boldsymbol{\theta}) &:= W^{(2)}\sigma(W^{(1)}\bar{\mathbf{H}}X + \mathbf{b}^{(1)}) \\ &= \sum_{k=1}^{m^{(1)}} w_k^{(2)}\sigma({\mathbf{w}_k^{(1)}}^{\mathsf{T}}\bar{\mathbf{H}}X + b_k^{(1)}) \\ &= \sum_{k=1}^{m^{(1)}} w_k^{(2)}\sigma({\mathbf{w}_k^{(1)}}^{\mathsf{T}}\bar{X} + b_k^{(1)}) \quad (\text{where} \bar{X} = \bar{\mathbf{H}}X) \end{aligned} \tag{8}$$

where $m^{(1)}$ is the number of nodes in layer 1, $\mathbf{w}_k^{(1)}$ is the k^{th} row of matrix $W^{(1)}$, $w_k^{(2)}$ is a scalar, the k^{th} element of vector $W^{(2)}$. σ denotes the activation function, and $\boldsymbol{\theta} = \{W^{(2)}, \mathbf{b}^{(1)}, \mathbf{w}_1^{(1)}, \ldots, \mathbf{w}_{m^{(1)}}^{(1)}\}$ denotes a vector of all parameters. Then

$$\begin{aligned} &\frac{\partial f}{\partial W_k^{(2)}} = \sigma({\mathbf{w}_k^{(1)}}^{\mathsf{T}}\bar{X} + b_k^{(1)}), \quad \frac{\partial f}{\partial W_k^{(1)}} = W_k^{(2)} s_k(\bar{X})\bar{X}, \\ &\frac{\partial f}{\partial b_k^{(1)}} = W_k^{(2)} s_k(\bar{X}), \quad \frac{\partial f}{\partial X_l} = \sum_{k=1}^{m^{(1)}} W_k^{(2)} s_k(\bar{X}){\mathbf{w}_k^{(1)}}^{\mathsf{T}}\bar{\mathbf{H}}_l \end{aligned} \tag{9}$$

where $s_k(\bar{X}) = \sigma'({\mathbf{w}_k^{(1)}}^{\mathsf{T}}\bar{X} + b^{(1)})$, $\bar{\mathbf{H}}_l$ is the l^{th} column vector of $\bar{\mathbf{H}}$. To measure the complexity of f, we choose $\mathbb{E}_X \|\nabla_X f\|_2^2$ due to its merit of considering derivatives w.r.t input X, which reflects the fluctuation of f, both the training data and testing data.

$$\begin{aligned} \mathbb{E}_X \|\nabla_X f(\mathbf{x})\|_2^2 &= \int \rho(dx) \sum_l \sum_{k_1,k_2} W_{k_1}^{(2)} W_{k_2}^{(2)} (\sum_{j=1}^{m^{(1)}} {\mathbf{w}_{k_1,j}^{(1)}}^{\mathsf{T}} \bar{\mathbf{H}}_{j,l}) \\ &\quad (\sum_{j=1}^{m^{(1)}} {\mathbf{w}_{k_2,j}^{(1)}}^{\mathsf{T}} \bar{\mathbf{H}}_{j,l}) s_{k_1}(\bar{X}) s_{k_2}(\bar{X}) \\ &= \int \rho(dx) \sum_{k_1,k_2} W_{k_1}^{(2)} W_{k_2}^{(2)} {\mathbf{w}_{k_1}^{(1)}}^{\mathsf{T}} \\ &\quad \sum_l \bar{\mathbf{H}}_l \bar{\mathbf{H}}_l^{\mathsf{T}} \mathbf{w}_{k_2}^{(1)} s_{k_1}(\bar{X}) s_{k_2}(\bar{X}) \\ &= \sum_{k_1,k_2} {\mathbf{w}_{k_1}^{(1)}}^{\mathsf{T}} \bar{\mathbf{H}}\bar{\mathbf{H}}^{\mathsf{T}} \mathbf{w}_{k_2}^{(1)} \mathbb{E}[\frac{\partial f}{\partial b_{k_1}^{(1)}} \frac{\partial f}{\partial b_{k_2}^{(1)}}] \\ &= \sum_{k_1,k_2} {\mathbf{w}_{k_1}^{(1)}}^{\mathsf{T}} \bar{\mathbf{H}}\bar{\mathbf{H}}^{\mathsf{T}} \mathbf{w}_{k_2}^{(1)} I_{\mathbf{b}^{(1)}}(k_1,k_2) \\ &\le \frac{1}{2}(\sum_{k_1,k_2} {\mathbf{w}_{k_1}^{(1)}}^{\mathsf{T}} \bar{\mathbf{H}}\bar{\mathbf{H}}^{\mathsf{T}} \mathbf{w}_{k_2}^{(1)})^2 + \frac{1}{2} I_{\mathbf{b}^{(1)}}^2(k_1,k_2) \\ &\le \frac{1}{2} \sum_{k_1,k_2} \left\|\mathbf{w}_{k_1}^{(1)}\right\|_2^2 \left\|\bar{\mathbf{H}}\bar{\mathbf{H}}^{\mathsf{T}} \mathbf{w}_{k_2}^{(1)}\right\|_2^2 \\ &\quad + \frac{1}{2} \sum_{k_1,k_2} I_{\mathbf{b}^{(1)}}^2(k_1,k_2) \\ &\le \frac{1}{2} \sum_{k_1,k_2} \left\|\mathbf{w}_{k_1}^{(1)}\right\|_2^2 \left\|\bar{\mathbf{H}}\bar{\mathbf{H}}^{\mathsf{T}}\right\|_F^2 \left\|\mathbf{w}_{k_2}^{(1)}\right\|_2^2 \\ &\quad + \frac{1}{2} \sum_{k_1,k_2} I_{\mathbf{b}^{(1)}}^2(k_1,k_2) \\ &= \frac{1}{2} \left\|\bar{\mathbf{H}}\bar{\mathbf{H}}^{\mathsf{T}}\right\|_F^2 (\sum_k \left\|\mathbf{w}_k^{(1)}\right\|_2^2)^2 + \frac{1}{2} \|I_{\mathbf{b}^{(1)}}\|_F^2 \\ &= \frac{1}{2} \left\|\bar{\mathbf{H}}^{\mathsf{T}}\bar{\mathbf{H}}\right\|_F^2 (\sum_k \left\|\mathbf{w}_k^{(1)}\right\|_2^2)^2 + \frac{1}{2} \|I_{\mathbf{b}^{(1)}}\|_F^2 \\ &= \frac{1}{2} \text{trace}(\bar{\mathbf{H}}^{\mathsf{T}}\bar{\mathbf{H}}) \left\|W^{(1)}\right\|_F^4 + \frac{1}{2} \|I_{\mathbf{b}^{(1)}}\|_F^2 \\ &\le \frac{1}{2}(2m + \text{rank}(X)) \left\|W^{(1)}\right\|_F^4 + \frac{1}{2} \|I_{\mathbf{b}^{(1)}}\|_F^2 \end{aligned} \tag{10}$$

where $I_{b^{(1)}}$ is the Fisher information matrix with respect to parameters $\mathbf{b}^{(1)}$, and $\text{rank}(X) \le \min\{n, m\}$. By the Cauchy-Schwarz inequality, we proved the following theorem to relate the complexity of hypothesis with the Fisher information matrix w.r.t. model parameters.

$$B^{(1)} = \begin{pmatrix} w_{1,1}^{(1)} & w_{1,2}^{(1)} & w_{1,3}^{(1)} & \cdots & w_{1,m}^{(1)} \\ w_{2,1}^{(1)} & w_{2,2}^{(1)} & w_{2,3}^{(1)} & \cdots & w_{2,m}^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{m^{(1)},1}^{(1)} & w_{m^{(1)},2}^{(1)} & w_{m^{(1)},3}^{(1)} & \cdots & w_{m^{(1)},m}^{(1)} \end{pmatrix}_{m^{(1)} \times m} \tag{11}$$

and equation (10) will be bounded by

$$
\begin{aligned}
2\mathbb{E}_X \left\|\nabla_X f(\mathbf{x})\right\|_2^2 &\leq (2m + 2 \times \text{rank}(X)) \left\|W^{(1)}\right\|_F^4 + \|I_{\mathbf{b}^{(1)}}\|_F^2 \\
&= 4(2m + 2 \times \text{rank}(X))(\frac{m^{(1)} + |E|}{m^{(1)} m})^2 \|B\|_F^4 + \|I_{\mathbf{b}^{(1)}}\|_F^2 \\
&\leq \frac{4(2m + 2 \times \text{rank}(X))(m^{(1)} + |E|)^2}{(m^{(1)} m)^2} \|B\|_F^4 + \|I_{\mathbf{b}^{(1)}}\|_F^2 \\
&\leq \frac{16m(m^{(1)} + |E|)^2}{(m^{(1)} m)^2} \|B\|_F^4 + \|I_{\mathbf{b}^{(1)}}\|_F^2, \text{where rank}(X) \leq m \\
&= \frac{16(\kappa + 1)^2}{m} \|B\|_F^4 + \|I_{\mathbf{b}^{(1)}}\|_F^2, \text{when } \frac{|E|}{m^{(1)}} = \kappa \\
&\leq \|B\|_F^4 + \|I_{\mathbf{b}^{(1)}}\|_F^2, \text{when } \kappa \leq \sqrt{\frac{m}{16}} - 1
\end{aligned} \tag{12}
$$

□

Proof of Corollary 1

Proof. From equation (10), we know

$$
\begin{aligned}
\mathbb{E}_X \left\|\nabla_X f(\mathbf{x})\right\|_2^2 &= \sum_{k_1,k_2} \mathbf{w}_{k_1}^{(1)^\mathsf{T}} \bar{\mathbf{H}} \bar{\mathbf{H}}^\mathsf{T} \mathbf{w}_{k_2}^{(1)} I_{\mathbf{b}^{(1)}}(k_1, k_2) \\
&\approx \sum_{k_1,k_2} \mathbf{w}_{k_1}^{(1)^\mathsf{T}} \bar{\mathbf{H}} \bar{\mathbf{H}}^\mathsf{T} \mathbf{w}_{k_2}^{(1)} I^n_{\mathbf{b}^{(1)}}(k_1, k_2) \\
&\quad + \mathcal{O}(\sqrt{\frac{\mathbf{Var}\left\|\nabla_X f\right\|_2^2}{n}}) \\
&= \sum_{k_1,k_2} \mathbf{w}_{k_1}^{(1)^\mathsf{T}} \bar{\mathbf{H}} \bar{\mathbf{H}}^\mathsf{T} \mathbf{w}_{k_2}^{(1)} (\frac{\partial^2 C}{\partial b_{k_1}^{(1)} \partial b_{k_2}^{(1)}} \\
&\quad - \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - y_i)^2 \frac{\partial^2 f}{\partial b_{k_1}^{(1)} \partial b_{k_2}^{(1)}}) \\
&\quad + \mathcal{O}(\sqrt{\frac{\mathbf{Var}\left\|\nabla_X f\right\|_2^2}{n}})
\end{aligned} \tag{13}
$$

since $\frac{\partial^2 f}{\partial b_{k_1}^{(1)} \partial b_{k_2}^{(1)}} = \delta_{k_1,k_2} \sigma^2(\mathbf{w}_{k_1}^{(1)^\mathsf{T}} \bar{\mathbf{H}} X + b_{k_1}^{(1)})$, combined with Theorem 1, we have the following inequality,

$$
\begin{aligned}
\mathbb{E}_X \left\|\nabla_X f(\mathbf{x})\right\|_2^2 &\leq \frac{1}{2}(2m + 2 \times \text{rank}(X)) \left\|W^{(1)}\right\|_F^4 \\
&\quad + \frac{1}{2} \left\|\nabla^2_{\mathbf{b}^{(1)}} \mathcal{C}\right\|_F^2 \\
&\quad \underbrace{-\frac{1}{n} \sum_{k_1,k_2,i} (f(X_i) - y_i) \mathbf{w}_{k_1}^{(1)^\mathsf{T}} \bar{\mathbf{H}} \bar{\mathbf{H}}^\mathsf{T} \mathbf{w}_{k_2}^{(1)} \delta_{k_1,k_2} w_{k_1}^{(2)} \sigma^2(\mathbf{w}_{k_1}^{(1)^\mathsf{T}}}_{G} \\
&\quad \underbrace{\bar{\mathbf{H}} X_i + b_{k_1}^{(1)})}_{G} \\
&\quad + \mathcal{O}(\sqrt{\frac{\mathbf{Var}\left\|\nabla_X f\right\|_2^2}{n}})
\end{aligned} \tag{14}
$$

Next we assume the second-order derivative of activation function is bounded, $R_\sigma = \|\sigma''\|_\infty$ is finite. This assumption is satisfied by commonly-used activation function, like sigmoid, tanh and relu. So we have,

$$
\begin{aligned}
G &= \frac{1}{n} \sum_k (f(X_i) - y_i) \left\|\mathbf{w}_k^{(1)}\right\|^2 w_k^{(2)} \sigma^2(\mathbf{w}_k^{(1)^\mathsf{T}} \bar{\mathbf{H}} X_i + b_k^{(1)}) \\
&\leq \frac{1}{n} \sum_k |f(X_i) - y_i| \left\|\mathbf{w}_k^{(1)}\right\|^2 R_\sigma \\
&\leq R_\sigma D \sqrt{\mathcal{C}}
\end{aligned} \tag{15}
$$

where $D = \max_k |w_k^{(2)}| \left\|\mathbf{w}_k^{(1)}\right\|_2^2$. Combining equation (14) and equation (15), we obtain the following characterization of $\mathcal{C}$.

$$
\begin{aligned}
2\mathbb{E}_X \left\|\nabla_X f(\mathbf{x})\right\|_2^2 &\leq \left\|\nabla^2_{\mathbf{b}^{(1)}} \mathcal{C}\right\|_F^2 + (2m + 2 \times \text{rank}(X)) \left\|W^{(1)}\right\|_F^4 \\
&\quad + 2R_\sigma D \sqrt{\mathcal{C}} + \mathcal{O}(\sqrt{\frac{\mathbf{Var}\left\|\nabla_X f\right\|_2^2}{n}})
\end{aligned} \tag{16}
$$

□

Time Complexity

For a 2-layer NN, the time complexity is $\mathcal{O}(2n(m \times m^{(1)} + m^{(1)} \times m^{(2)})) = \mathcal{O}(nmm^{(1)})$. For a 3-layer PASNet (their newtork setting), the time complexity is $\mathcal{O}(2n(smm^{(1)} + m^{(1)}m^{(2)} + m^{(2)}m^{(3)}))$. Since they have gene-pathway layer and pathway-hidden layer. The time complexity is dominated by hidden neuron numbers $m^{(2)}$ and s, the average link number for each pathway. For a 2-layer PL, the time complexity is $\mathcal{O}(2n(s \times m \times m^{(1)} + m(n^2 s + ns^2 + s^3) + m^{(1)} \times m^{(2)})) = \mathcal{O}(smm^{(1)} + smn^2)$, which is dominated by s, the average number of children for each node. Because $n > s$ and $m \gg n$, the time complexity of PL is much less than that of NN. In addition, PL gains further efficiency due to (1) more

independent components, less singular matrix, faster convergence; (2) less number of free parameters; and (3) smaller number of batches.

Simulation Results and Figures

Tables and figures are in the next 3 pages.

Analytical Details for Lung Transplantation Study

Hyperparameters. All models' hyperparameters were selected by grid search. The numbers of groups for tree-based and group-based methods were set to be 5, 10, 15, 20, and 25 (except max heap); The maximum iteration for GL, TSGL, and MH was 100 (max heap is 5) and the maximum number of iterations was 10K (max heap is 50); The regularization ratio for the above methods was set to 1; The tree's max depth for DT and XGboost ranged from 3 to 10; XGboost's features' sample rate was 0.8 and the sub-sample rate was 0.8; NN had 3 hidden layers and hidden layer's neuron size was [500, 500, 10] (if $m \leq 1000$) or [1000, 1000, 50] (if $m > 1000$), which is the best setting we tried. PASNet had two hidden layers and the first hidden layers' size was the number of separate structures, the second hidden layer's neuron size was 10. The batch size was 20 for NN and PASNet. PL's hidden size and the number of hidden layer were determined by the feature structure. The default learning rate was 0.3 for PL.

Figure 4 and Figure 5 illustrate the prior structure Chemokine signaling pathway and JAK-STAT signaling pathway.

Table 5 and Table 6 showed the standard deviation of the AUC and ACC across multiple sets of parameters for each method.

Analytical Details for Brain Image Study

Hyperparameters for Gaussian Graphical Lasso. We set the regularization parameter λ ranging from 0.1 to 10 with 10 evenly-spaced values on the log scale. The optimal graph was selected according to the best risk inflation criterion (Friedman, Hastie, and Tibshirani 2008; Zhang and Shen 2010). Then the edges with weak associations (less than 0.1) were removed to avoid spurious connections. We further assumed the features with the most connections were the first ancestries and the offspring nodes were linked from the ancestries sequentially.

Table 7 shows the standard deviation of MSE.

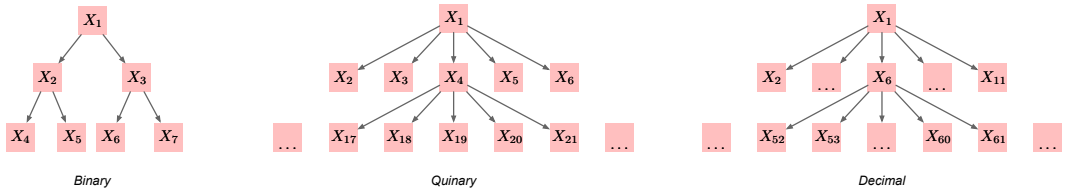

Figure 3: Binary, quinary, and decimal tree structures in simulations. Side and bottom branches are skipped due to space.

Table 2: MSE (s.d.) of the prediction from NN, PASNet, and PL for $n = 200$ and $m = 1000$.

layers s	X-X, X-Y Relationship	**NN**	**PASNet**	**PL**
2	Linear	1.096 (0.262)	0.935 (0.293)	**0.633 (0.276)**
	Nonlinear	1.948(0.845)	1.079 (0.020)	**0.896 (0.358)**
5	Linear	0.699 (0.203)	1.012 (0.278)	**0.399 (0.120)**
	Nonlinear	1.439 (0.538)	1.228 (0.585)	**1.156 (0.550)**
10	Linear	0.288 (0.121)	1.189 (0.441)	**0.209 (0.139)**
	Nonlinear	2.421 (1.280)	1.698 (0.532)	**1.379 (0.569)**

Table 3: MSE (s.d.) of the prediction from NN and PL for different numbers of input variables (m) or sample sizes (n)

m $(n = 200)$	s	**NN**	**PL**
200	2	1.462 (0.374)	**1.321** (0.291)
	5	**0.796** (0.206)	1.446 (0.221)
	10	**0.934** (0.280)	0.956 (0.132)
1000	2	1.948 (0.845)	**0.896** (0.358)
	5	1.439 (0.538)	**1.156** (0.545)
	10	2.421 (1.280)	**1.379** (0.569)
5000	2	2.844 (1.291)	**0.942** (0.457)
	5	1.968 (0.995)	**1.029** (0.550)
	10	2.520 (1.566)	**1.320** (0.873)

n $(m = 1000)$	s	**NN**	**PL**
100	2	2.138 (0.864)	**1.012** (0.388)
	5	2.801 (1.226)	**1.330** (0.563)
	10	1.887 (0.931)	**1.248** (0.536)
500	2	1.425 (0.571)	**1.149** (0.455)
	5	1.237 (0.524)	**0.967** (0.314)
	10	1.245 (0.567)	**0.891** (0.524)
2000	2	**0.538** (0.284)	0.842 (0.449)
	5	**0.487** (0.247)	0.689 (0.210)
	10	**0.483** (0.232)	0.629 (0.237)

Table 4: MSE (s.d.) of the prediction from PL with mis-specified directions, at mis-specification rates (e) of $0\% - 50\%$ with $m = 5000$ and $n = 200$.

s	X-X, X-Y Relationship	Correct $e = 0$	$e = 10\%$	$e = 20\%$	$e = 30\%$	$e = 40\%$	$e = 50\%$
2	Linear	**0.787** (0.277)	0.973 (0.355)	1.040 (0.307)	1.882 (1.603)	4.438 (2.379)	5.551 (2.54)
	Nonlinear	**0.942** (0.457)	1.312 (0.478)	1.337 (0.679)	1.329 (0.830)	1.470 (0.978)	1.967 (1.594)
5	Linear	**0.614** (0.120)	0.793 (0.437)	0.946 (0.625)	1.035 (0.579)	1.087 (0.717)	1.173 (0.924)
	Nonlinear	**1.029** (0.550)	1.568 (0.675)	1.426 (0.761)	1.676 (0.711)	1.613 (0.844)	2.035 (2.253)
10	Linear	**0.892** (0.495)	1.084 (0.513)	1.426 (0.752)	1.540 (0.699)	2.150 (2.039)	2.735 (1.678)
	Nonlinear	**1.320** (0.873)	1.653 (0.888)	1.498 (0.597)	1.886 (1.074)	1.802 (0.814)	2.283 (1.472)

Table 5: *s.d* of AUC of the prediction of PGD in the lung transplant study from all methods using genes within 6 different pathways

Method	Chemokine SP	Toll receptor SP	Jak stat SP	Nod receptor SP	Graft v.s. HD	Primary ID
Lasso	0.112	0.131	0.09	0.133	0.107	0.197
GL	0.072	0.048	0.14	0.154	0.132	0.169
TSGL	0.087	0.11	0.093	0.072	0.134	0.167
MH	0.124	0.103	0.097	0.081	0.105	0.149
DT	0.23	0.138	0.168	0.221	0.167	0.178
XGboost	0.085	0.07	0.079	0.087	0.095	0.076
NN	0.061	0.106	0.128	0.124	0.163	0.175
PASNet	0.097	0.105	0.081	0.077	0.082	0.102
PL	0.125	0.097	0.074	0.089	0.1	0.141

Table 6: *s.d* of ACC of the prediction of PGD in the lung transplant study from all methods using genes within 6 different pathways

Method	Chemokine SP	Toll receptor SP	Jak stat SP	Nod receptor SP	Graft v.s. HD	Primary ID
Lasso	0.085	0.084	0.040	0.067	0.039	0.087
GL	0.072	0.048	0.14	0.154	0.132	0.169
TSGL	0.087	0.11	0.093	0.072	0.134	0.167
MH	0.124	0.103	0.097	0.081	0.105	0.149
DT	0.147	0.069	0.082	0.211	0.060	0.149
XGboost	0.103	0.057	0.037	0.098	0.064	0.108
NN	0.047	0.164	0.119	0.018	0.021	0.018
PASNet	0.161	0.159	0.161	0.154	0.161	0.160
SDL	0.021	0.101	0.038	0.018	0.059	0.066

Table 7: *s.d.* (10^3ms^2) of the optimal model prediction of median response time in fMRI study.

Method	S Motor Area L	S Motor Area R	Parietal Inf L	Parietal Inf R	Caudate L	Putamen L	Putamen R	Frontal Mid L	Frontal Mid R
Lasso	2.28	3.61	2.26	2.10	1.20	1.46	2.00	1.95	1.58
GL	2.40	2.64	2.57	2.96	3.58	3.07	2.89	2.27	3.11
TSGL	2.55	3.04	2.53	3.47	3.89	3.35	3.31	2.18	2.54
MH	3.17	3.12	1.33	2.13	2.52	3.75	3.66	1.12	1.37
DT	7.92	3.06	4.90	2.94	7.26	4.86	2.57	11.64	4.01
XGboost	2.40	3.07	2.77	2.92	2.61	2.42	2.66	2.82	2.80
NN	1.91	1.89	1.52	1.58	1.87	1.76	1.71	1.49	1.62
PASNet	1.56	1.07	2.45	1.77	1.62	1.56	1.39	2.23	1.64
PL	1.33	1.54	2.13	1.23	1.23	0.70	0.89	2.13	1.75

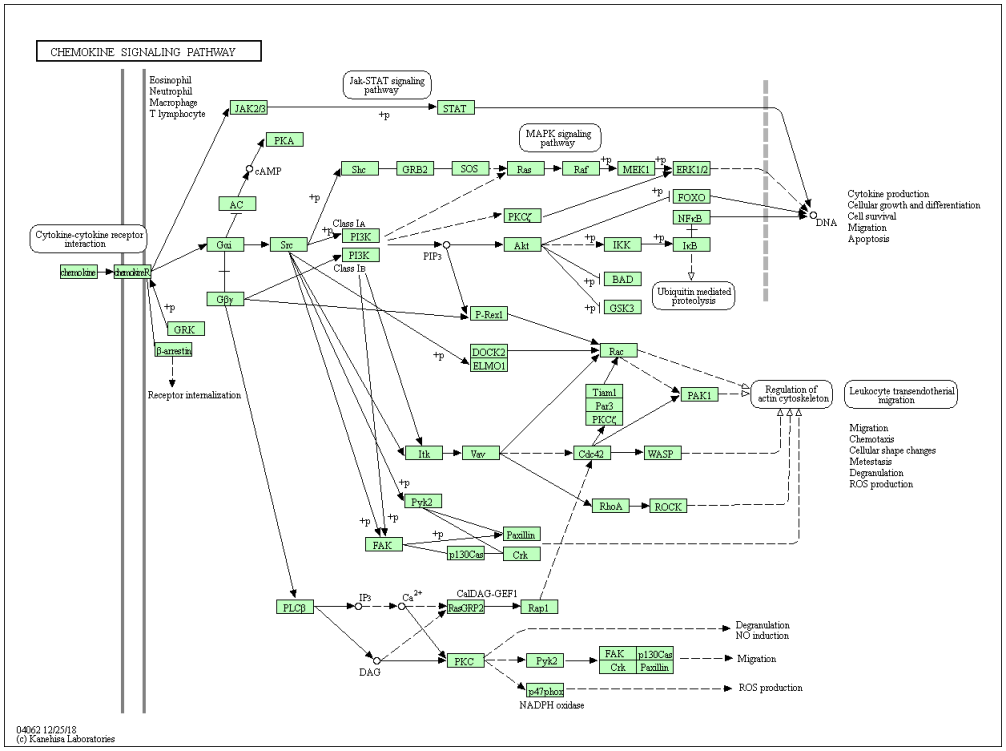

Figure 4: Chemokine Signaling Pathway

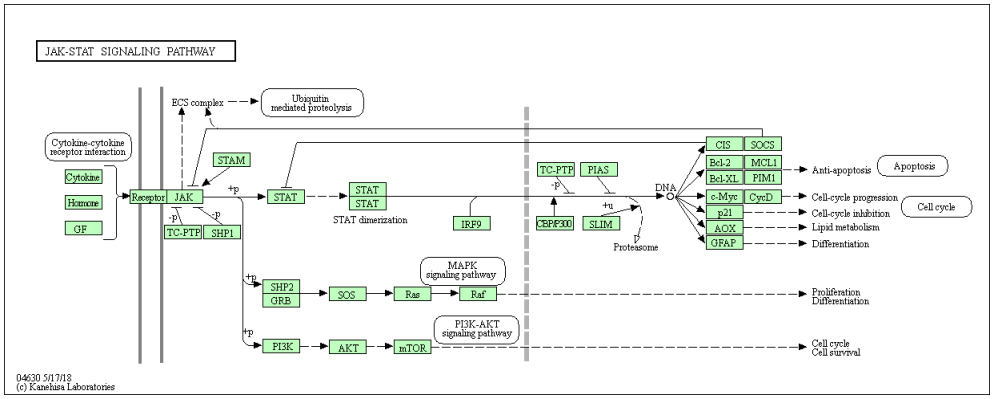

Figure 5: Jak-STAT Signaling Pathway