Deep Feature Selection: Theory and Application to Identify Enhancers and Promoters

Yifeng Li, Chih-Yu Chen, and Wyeth W. Wasserman

Centre for Molecular Medicine and Therapeutics
University of British Columbia
950 West 28th Avenue, Vancouver, BC V5Z 4H4 Canada
{yifeng,juliec,wyeth}@cmmt.ubc.ca

Abstract. Sparse linear models approximate target variable(s) by a sparse linear combination of input variables. The sparseness is realized through a regularization term. Since they are simple, fast, and able to select features, they are widely used in classification and regression. Essentially linear models are shallow feed-forward neural networks which have three limitations: (1) incompatibility to model non-linearity of features, (2) inability to learn high-level features, and (3) unnatural extensions to select features in multi-class case. Deep neural networks are models structured by multiple hidden layers with non-linear activation functions. Compared with linear models, they have two distinctive strengths: the capability to (1) model complex systems with non-linear structures, (2) learn high-level representation of features. Deep learning has been applied in many large and complex systems where deep models significantly outperform shallow ones. However, feature selection at the input level, which is very helpful to understand the nature of a complex system, is still not well-studied. In genome research, the *cis*-regulatory elements in non-coding DNA sequences play a key role in the expression of genes. Since the activity of regulatory elements involves highly interactive factors, a deep tool is strongly needed to discover informative features. In order to address the above limitations of shallow and deep models for selecting features of a complex system, we propose a deep feature selection model that (1) takes advantages of deep structures to model non-linearity and (2) conveniently selects a subset of features right at the input level for multi-class data. We applied this model to the identification of active enhancers and promoters by integrating multiple sources of genomic information. Results show that our model outperforms elastic net in terms of size of discriminative feature subset and classification accuracy.

Key words: deep learning, feature selection, enhancer, promoter

1 Introduction

Sparse regularized linear models are widely used in machine learning and bioinformatics for classification and regression. These models are shallow feed-forward neural networks which approximate the response variable by a sparse superposition of input variables (or features), that is $y \approx f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{w} + b$. From a

maximum a posteriori (MAP) estimation (or regularization) perspective, its optimization can be generally formulated to $\min_{\boldsymbol{w},b} f(\boldsymbol{w},b) = l(\boldsymbol{w},b) + \lambda r(\boldsymbol{w})$, where $l(\boldsymbol{w},b)$ is a loss function corresponding to the negative log-likelihood of data, and $r(\boldsymbol{w})$ is a sparse regularization term corresponding to the prior of model parameter. Typical loss functions include 0-1 loss, hinge loss, logistic loss (for classification), squared loss (for both classification and regression), ε-sensitive loss (for regression), etc. A regularization term aims to reduce model complexity, thus avoids overfitting. Also, a *sparse* regularization can help to select features by taking features with nonzero weights in $\boldsymbol{w}$. Commonly used sparse regularization terms include l_1-norm (LASSO) [24], non-negativity [16], and SCAD [5]. Perhaps LASSO and its variant, elastic net [28] (as formulated in Equation (1)), are the most popularly used techniques.

$$\begin{cases} \text{LASSO:} & r(\boldsymbol{w}) = \|\boldsymbol{w}\|_1 \\ \text{elastic net:} & r(\boldsymbol{w}) = \frac{1-\alpha}{2}\|\boldsymbol{w}\|_2^2 + \alpha\|\boldsymbol{w}\|_1 \end{cases} \tag{1}$$

LASSO is a special case of elastic net by setting $\alpha = 1$.

The popularity of sparse linear models is due to the reasons as follows. First, their concept is easy to understand. Second, variables can be selected. Third, the learning of model parameter $\boldsymbol{\theta}$ is often convex in the parameter landscape, thus many fast implementations are available. However, linear models have three main limitations. (1) Non-linear correlation among variables can not be considered (except by handcrafted features or a kernel extension). (2) High-level representation of features can not be learned due to the shallow structure. (3) There does not exist a "natural" way to extend a two-class linear model to multi-class case in classification and feature selection. Two common multi-class extensions are one-versus-one and one-versus-rest. The corresponding feature selection is accomplished by taking the union of results generated by two-class linear models. For instance, given C classes, softmax regression (a one-versus-rest multi-class extension of logistic regression, see Fig. 1a) with LASSO will produce C subsets of class-specific features. These subsets are then pooled as the final result. Since the final subset of features depends on one specific strategy of multi-class extension, different strategies may yield different results.

Through piling up hidden layers, deep neural networks are able to model nonlinearity of features. Fig. 1c is an example of such a deep model called *multilayer perceptrons* (MLP) which is a deep feed-forward neural network. The techniques of learning deep models and their inferences fall into an active research frontier – deep learning [10], which has four attractive strengths for applying it to complex intelligent systems: First, deep models often dramatically increase prediction accuracy. Second, they can model processes of complex systems. Third, they can generate structured high-level representation of features which can help the interpretation of data. Fourth, (convolutional) deep learning models are robust to temporal or spatial variation. But the learning of such models are usually non-convex in optimization, and the back-propagation algorithm (a first-order method) does not perform well on deeper structures. The optimization strategy using greedy layer-wise unsupervised pretraining and finetuning, proposed

in [10], is considered as a breakthrough. While high-level feature extraction and representation have been intensively studied in the surge of deep learning research [3], feature selection at the input level is still not well-studied. However, in bioinformatics and other studies on complex systems, selecting key input features are crucial to understand the mechanisms of the systems. Thus, existing models mentioned above do not meet this need. In our current bioinformatics research, we are committed to devising a deep learning model for the identification and understanding of *cis*-regulatory elements in the human genome.

Genome researchers have discovered that non-coding DNA sequences (previously viewed as junk DNA) are composed of many regulatory elements [22]. These elements (including enhancers and promoters) precisely control the expression level of genes. Promoters are *cis*-acting DNA sequences that switch on or off the expression of genes, while enhancers are generally *cis*-acting DNA sequences that tune the expression level of genes [21]. A promoter resides closely to its target gene, while an enhancer is distal to its target gene(s) making it difficult to identify. The identification of active enhancers and promoters in a genome is of key importance, as it can help to elucidate the regulatory mechanism in the genome, and interpret disease-causing variants within *cis*-regulatory elements. However, since the regulatory landscapes of DNA are quite different among cell types, and the regulatory events are precisely and dynamically controlled by multiple factors, including epigenetic marks, transcription factors, microRNAs, and their interactions, it is a difficult task to identify active enhancers and promoters in a given cell type. The emergence of both deep sequencing and deep computing techniques casts light on this problem.

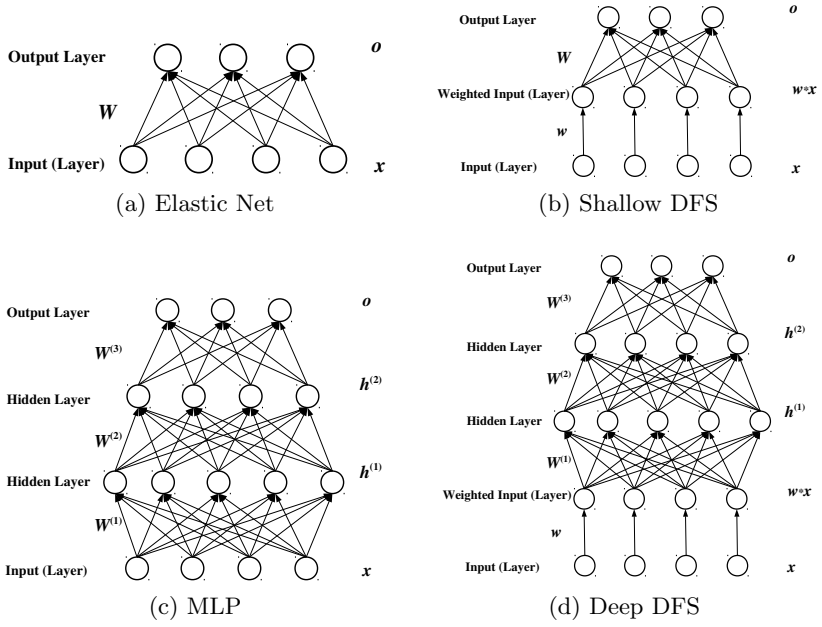
In order to select key input features for the identification and understanding the regulatory events, we propose a deep feature selection model that enables variable selection for deep neural networks. In this model, a sparse one-to-one layer, where each input feature is weighted, is added between the input and the first hidden layer, giving two advantages: (1) a single subset of features for multiple classes (multiple output nodes) can be conveniently selected, which addresses the challenge of multi-class extension of linear models; (2) through selecting features at the input level of the deep structure, we are able to identify informative features that have non-linear behaviours.

2 Method

2.1 Deep Feature Selection

We focus our research on feature selection for multi-class data using deep neural networks. We propose a *deep feature selection* (DFS) model that can select features at the input level of a deep network. An example of such a model is illustrated in Fig. 1d. Our main idea is to add a sparse one-to-one linear layer between the input layer and the first hidden layer of a MLP. In this one-to-one layer, the input feature x_i only connects to the i-th node with linear activation function. Thus, the output of the one-to-one layer becomes $\boldsymbol{w} * \boldsymbol{x}$, where $*$ is element-wise multiplication. In order to select input features, $\boldsymbol{w}$ has to be sparse,

and only the features corresponding to nonzero weights are selected. Although we can resort to *any* sparse regularization term on $\boldsymbol{w}$. In our current study, we use elastic-net $\lambda_1\left(\frac{1-\lambda_2}{2}\|\boldsymbol{w}\|_2^2 + \lambda_2\|\boldsymbol{w}\|_1\right)$ [28]. Such a DFS model can be called *deep elastic net*. As in regular MLP, the activation function in the hidden layers of DFS is also nonlinear (e.g. sigmoid or tangent). The output layer is a softmax layer, where the output of unit i is defined as $p(y=i|\boldsymbol{x}) = \frac{e^{-\boldsymbol{w}_i^{(K+1)^{\mathrm{T}}}\boldsymbol{h}^{(K)}}}{\sum_{c=1}^{C} e^{-\boldsymbol{w}_c^{(K+1)^{\mathrm{T}}}\boldsymbol{h}^{(K)}}}$, where $\boldsymbol{w}_i^{(K+1)}$ is the i-th column of weight matrix $\boldsymbol{W}^{(K+1)}$ from the last hidden layer (that is the K-th hidden layer) to the softmax layer. Our DFS model has at least two distinctive advantages. First, given a parameter setting, it always selects a single subset of features for multi-class problems. It overcomes the limitation of linear models for multi-class data making feature selection more convenient. Second, by using a deep nonlinear structure, it can automatically identify non-linear features, which is superior over shallow linear models.



(a) Elastic Net (b) Shallow DFS

(c) MLP (d) Deep DFS

Fig. 1: A structural comparison of our DFS models and previous ones.

2.2 Learning Model Parameter

Suppose there are K hidden layers in a DFS model. Its model parameter can be denoted by $\boldsymbol{\theta} = \{\boldsymbol{w}, \boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}, \cdots, \boldsymbol{W}^{(K+1)}, \boldsymbol{b}^{(K+1)}\}$, where $\boldsymbol{W}^{(k)}$ is the

weight matrix connecting the $k-1$-th layer to the k-th layer, and $\boldsymbol{b}^{(k)}$ is the corresponding biases in the k-th layer. The size of $\boldsymbol{W}^{(k)}$ is $n_{k-1} \times n_k$, where n_k is the number of units in the k-th layer. In order to learn the model parameter, we minimize the objective function below,

$$\begin{aligned}\min_{\boldsymbol{\theta}} \; & f(\boldsymbol{\theta}) = l(\boldsymbol{\theta}) + \lambda_1 \Big(\frac{1-\lambda_2}{2} \|\boldsymbol{w}\|_2^2 + \lambda_2 \|\boldsymbol{w}\|_1 \Big) \\ & + \alpha_1 \Big(\frac{1-\alpha_2}{2} \sum_{k=1}^{K+1} \|\boldsymbol{W}^{(k)}\|_F^2 + \alpha_2 \sum_{k=1}^{K+1} \|\boldsymbol{W}^{(k)}\|_1 \Big), \end{aligned} \tag{2}$$

which is explained as follows.

1. $l(\boldsymbol{\theta})$ is the log-likelihood of data. Recall that the top layer of our model is a softmax regression model with a multinoulli distribution for the probability of targets:

$$h(\boldsymbol{h}^{(K)}, \boldsymbol{\theta}) = \begin{bmatrix} p(y=1|\boldsymbol{h}^{(K)}, \boldsymbol{\theta}) \\ \vdots \\ p(y=C|\boldsymbol{h}^{(K)}, \boldsymbol{\theta}) \end{bmatrix}.$$

   Therefore, $l(\boldsymbol{\theta})$ in Equation (2) is

$$l(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \log p(y_i|\boldsymbol{h}_i^{(K)}) = -\sum_{i=1}^{N} \log \frac{e^{-\boldsymbol{w}_{y_i}^{(K+1)\mathrm{T}} \boldsymbol{h}_i^{(K)} - b_{y_i}^{(K+1)}}}{\sum_{c=1}^{C} e^{-\boldsymbol{w}_c^{(K+1)\mathrm{T}} \boldsymbol{h}_i^{(K)} - b_c^{(K+1)}}}, \tag{3}$$

   where $\boldsymbol{h}_i^{(K)}$ is the output of the K-th hidden layer given input sample $\boldsymbol{x}_i$, thus, it is a function of $\boldsymbol{\theta}/\{\boldsymbol{W}^{K+1}, \boldsymbol{b}^{(K+1)}\}$.
2. Regularization term $\lambda_1 \Big(\frac{1-\lambda_2}{2} \|\boldsymbol{w}\|_2^2 + \lambda_2 \|\boldsymbol{w}\|_1 \Big)$ is an elastic-net-like term, where user-specified parameter $\lambda_2 \in [0, 1]$ controls the trade-off between smoothness and sparsity of $\boldsymbol{w}$.
3. Regularization term $\alpha_1 \Big(\frac{1-\alpha_2}{2} \sum_{k=1}^{K+1} \|\boldsymbol{W}^{(k)}\|_F^2 + \alpha_2 \sum_{k=1}^{K+1} \|\boldsymbol{W}^{(k)}\|_1 \Big)$ is another elastic-net-like term that helps to reduce the model complexity and speed up the optimization. Another effect of this term is to avoid the shrinking of $\boldsymbol{w}$ in the one-to-one layer causing the swelling of $\boldsymbol{W}^{(k)}$ in the upper layers (that is, w_i is very small, but its downstream weights are very large).

In the neural network community, it is well-known that Equation (2) is non-convex, and gradient descent method (back-propagation) converges only to a local minimum of the weight space. Practically, it performs fairly well with a small number of hidden layers. However, as the number of hidden layers increases, this algorithm would deteriorate, because gradient information disperses in lower layers. So for a small number of hidden layers, we directly use a back-propagation algorithm to train our DFS model. For a large value of K, if back-propagation does not perform well, we resort to *stacked contractive autoencoder* (ScA) or *deep belief network* (DBN). The ScA and DBN based DFS models are pretrained in a greedy layerwise way, and then finetuned by back-propagation.

Although the objective $f(\boldsymbol{\theta})$ in Equation (2) is non-differentiable everywhere, it is semi-differentiable. This is the reason that back-propagation can still be used for our DFS model. However, it is indeed a practical challenge to explicitly derive the first-order derivative with respect to the parameter of a complex model. Thanks to the `Theano` package [4] which is a symbolic expression compiler, we are able to escape the explicit derivation of gradients. The `Deep Learning Tutorials` [17] is a well documented Python package including the example implementations of softmax regression, MLP, stacked autoencoders [9], *restricted Boltzmann machine* (RBM) [1], DBN [10], and *convolutional neural network* (CNN) [13]. It aims to teach researchers how to build deep learning models using `Theano`. We implemented our DFS model on top of `Theano` and `Deep Learning Tutorials`. We also substantially modified the `Deep Learning Tutorials` in the following points in order to allow users to apply it in their fields conveniently. We add training and test functions for each method. Learning rate can decay as the number of epochs increases. Momentum is added for faster and stable convergence. These methods result in a `deep learning package`, which is publicly available at [15].

2.3 Shallow DFS is not Equivalent to LASSO

Is the result of a shallow DFS model (Fig. 1b) equivalent to that of LASSO (Fig. 1a)? If so, there is no need to build the DFS model except for a practical reason; features could be simply selected by making $\boldsymbol{W}^{(1)}$ sparse in the model as illustrated in Fig. 1c. Fortunately, the answer is "no". It is because the sparse weight matrices $\boldsymbol{W}$ produced by both models are different. To prove this, we simplify both models but without hurting the nature of this question, and formulate the corresponding optimizations below:

$$\min_{\boldsymbol{\theta}} \ f(\boldsymbol{\theta}) = l(\boldsymbol{\theta}) + \lambda \|\boldsymbol{W}\|_1 \quad \text{(LASSO)}, \tag{4}$$

$$\min_{\boldsymbol{\theta}} \ f(\boldsymbol{\theta}) = l(\boldsymbol{\theta}) + \lambda_1 \|\boldsymbol{w}\|_1 + \lambda_2 \|\boldsymbol{W}\|_1 \quad \text{(Shallow DFS)}. \tag{5}$$

The optimal solution to Equation (4) is not equivalent to that to Equation (5).

Proof. The parameter of LASSO in Equation (4) is $\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{b}\}$, and the parameter of the shallow DFS in Equation (5) is $\boldsymbol{\theta} = \{\boldsymbol{w}, \boldsymbol{W}, \boldsymbol{b}\}$. We can combine parameter $\{\boldsymbol{w}, \boldsymbol{W}\}$ of Equation (5) to $\bar{\boldsymbol{W}}$, where its i-th row is $w_i * \boldsymbol{W}_{i,:}$. Obviously, $\bar{\boldsymbol{W}}$ is a matrix with a row-wise sparseness, while, from the property of l_1-norm, all elements of $\boldsymbol{W}$ in LASSO follow the same Laplace distribution. If we could rewrite Equation (5) to the following form

$$\min_{\bar{\boldsymbol{W}}, \boldsymbol{b}} \ f(\bar{\boldsymbol{W}}, \boldsymbol{b}) = l(\bar{\boldsymbol{W}}, \boldsymbol{b}) + \beta \|\bar{\boldsymbol{W}}\|_1, \tag{6}$$

then Equation (5) would be equivalent to Equation (4). However, we cannot. This is because $\beta\|\bar{\boldsymbol{W}}\|_1 = \beta \sum_i \sum_j |w_i w_{ij}|$ in Equation (6) and $\lambda_1\|\boldsymbol{w}\|_1 + \lambda_2\|\boldsymbol{W}\|_1 = \lambda_1 \sum_i |w_i| + \lambda_2 \sum_i \sum_j |w_{ij}|$ in Equation (5). Therefore, we cannot find a value of β to guarantee $\beta\|\bar{\boldsymbol{W}}\|_1 = \lambda_1\|\boldsymbol{w}\|_1 + \lambda_2\|\boldsymbol{W}\|_1 + \text{constant}$. The only exception being if $\boldsymbol{w}$ is a nonzero constant.

3 Applying DFS to Enhancer-Promoter Classification

We applied the DFS model in the challenging problem of enhancer-promoter classification. In order to assess the performance of this model, we compared four models, including our deep DFS model having two hidden layers (Fig. 1d), our shallow DFS model having no hidden layer (Fig. 1b), elastic-net based softmax regression (Fig. 1a), and random forest [7]. We shall first describe the genomic data we used. Then, we compare the prediction accuracy and computing time. Finally, we provide new insights into the features selected.

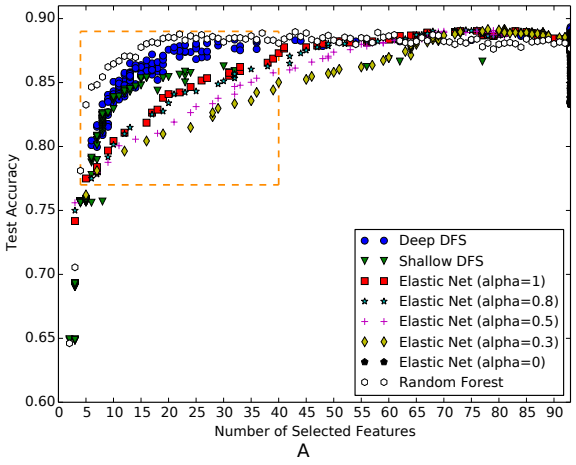
3.1 Data

We compared the models on our processed data sampled from annotated DNA regions of GM12878 cell line (a lymphoblastoid cell line). This data set has 93 features and three classes, each of which contains 2,156 samples. Based on the FANTOM5 promoter and enhancer atlases [23,2], each sample comes from one of the three classes of annotated DNA regions including active enhancer regions, active promoter regions, and background. The background class is a pool of inactive enhancers, inactive promoters, active exons, and unknown regions. The features include cell-ubiquitous characteristics such as CpG-islands and evolutionary reservation Phastcons score, and cell-specific events including DNA-accessibility, histone modifications, and transcription factor binding sites captured by the ENCODE consortium using ChIP-seq techniques [22]. For a fair comparison, we split our data set equally into training set, validation set, and test set. All models were trained on the same training set. The validation accuracy is used to monitor the training of the DFS models to avoid overfitting. The same test set was blinded in the training of all three models, so the test accuracy was used to examine the quality of feature subsets.

3.2 Comparing Test Accuracy and Computing Time

In our deep DFS model, we take the structure of $\{93 \to 93 \to 128 \to 64 \to 3\}$ by a *manual* rough model selection, due to a concern about the efficiency of automatic model selection for deep models. We set the minbatch size to 100, the maximum number of epochs to 1000, the initial learning rate $s = 0.1$, the coefficient of momentum $\alpha = 0.1$, $\lambda_2 = 1$, $\alpha_1 = 0.0001$, and $\alpha_2 = 0$. We conducted feature selection for values of λ_1 from the range of $[0, 0.03]$ by a step of 0.0002. Our shallow DFS model has a structure of $\{93 \to 93 \to 3\}$. For this model, we tried values of λ_1 from $[0, 0.07]$ by a step of 0.0005. The rest of the user-specified parameters were kept the same as the deep DFS above. Elastic-net based softmax regression simply has a structure of $\{93 \to 3\}$. We tried different values of α. We used the `glmnet` package for it, thus the full regularization path with a fixed value of α was produced by a cyclic coordinate descent algorithm [8]. For random forest, we applied the `randomForest` package in R.

The test accuracies versus the sizes of feature subsets are illustrated in Fig. 2(A). From a feature selection context, we focus the comparison on the critical region as highlighted by a rectangle in this plot. In this region, the paired Wilcoxon signed-rank test was conducted to check whether a classifier significantly outperforms another one (see Fig. 2(B)). In addition to accuracy, the confusion matrices of different models, when selecting 16 features, are given in Fig. 2(C). First of all, with the comparison between our shallow DFS model and elastic net, it can be seen that our shallow model has a significantly higher test accuracy than elastic net for the same number of selected features. From a computational viewpoint, it hence corroborates that adding a sparse one-to-one layer is a better technique than the tradition of simply combining the feature subsets selected for each class. Second, from the comparison of our deep and shallow DFS models, it shows that a significantly better test accuracy can be obtained by our deep model. It hence supports that considering the non-linearity among the variables can contribute to the improvement of prediction capability. Third, it is interesting to see that random forest with certain top-ranked features performs better than the deep learning model. This may be because the structure and parameter of the deep model was not optimized. Finally, from the confusion matrices as shown in Fig. 2(C), we highlight that some active promoters tend to be classified as active enhancers.



p-value	Shallow DFS	LASSO	Random Forest
Deep DFS	2.33e-4	9.82e-4	8.30e-6
Shallow DFS		1.21e-2	5.96e-5
LASSO			1.32e-4

B

Deep DFS		Predicted		
		A-E	A-P	BG
Actual	A-E	587	75	56
	A-P	63	641	15
	BG	50	50	619

Shallow DFS		Predicted		
		A-E	A-P	BG
Actual	A-E	582	64	72
	A-P	50	648	21
	BG	56	53	610

LASSO		Predicted		
		A-E	A-P	BG
Actual	A-E	549	124	45
	A-P	58	652	9
	BG	77	78	564

Random Forest		Predicted		
		A-E	A-P	BG
Actual	A-E	622	50	46
	A-P	59	640	20
	BG	45	34	640

C

Fig. 2: (A) The number of selected features by different methods and corresponding test accuracy. The critical region is highlighted by the orange rectangle. (B) p-value of the paired Wilcoxon signed-rank test in the critical region. (C) Confusion matrices when selecting 16 features by different models, respectively. A-E: Active Enhancer, A-P: Active Promoter, BG: Background.

We recorded the training times of the four models on the same computer. The shallow DFS, elastic net, and random forest took only 3.32, 6.56, and 2.68

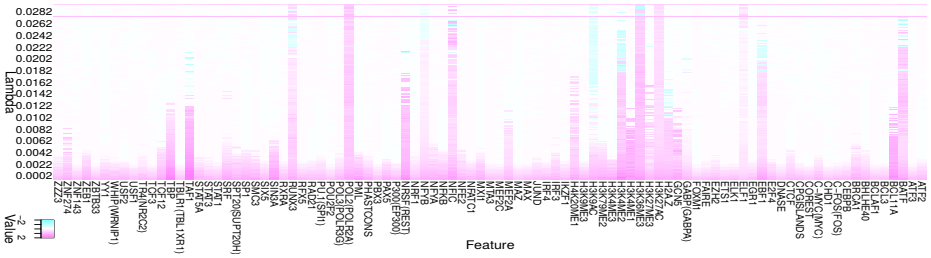
seconds, respectively, to learn from data. However, the deep DFS model "unsurprisingly" consumed around 69.10 seconds.

3.3 Feature Analysis

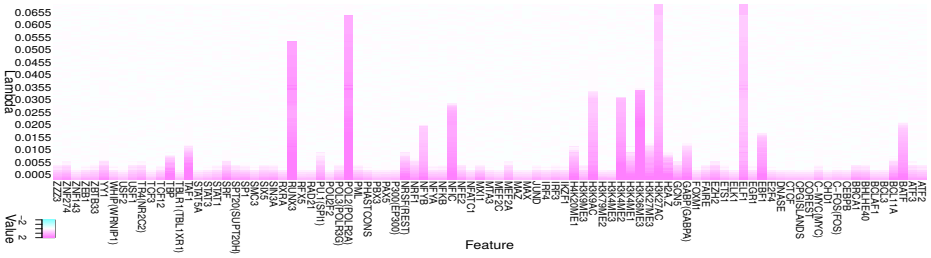
We analyzed the features selected by the DFS models, LASSO, and random forest. We used a heatmap, as shown in Fig. 3, to visualize the regularization path in the sparse models. Since LASSO is supposed to have three (each for a class) heatmaps, we combined them by taking the corresponding maximal absolute values of its coefficients. That is, for a value of λ, we convert matrix $\boldsymbol{W}$ to a vector $\boldsymbol{w}^{\text{new}}$ by $w_i^{\text{new}} = \max\{|w_{i1}|, |w_{i2}|, |w_{i3}|\}$.

First, we can see that the heatmaps of our shallow and deep DFS models are much sparser than that of LASSO. This implies that our scheme using a sparse one-to-one weighting layer is able to select a small subset of features along the regularization path, while LASSO tends to select more features, because it fuses all class-specific feature subsets. Second, comparing the result of the shallow DFS and LASSO, we can see many differences. For example, LASSO emphasizes CpG islands, TBLP1, and TBP, while they are not selected by the shallow DFS until later in the process. Instead, the heatmap of the shallow DFS indicates that ELF1, H2K27ac, Pol2, RUNX3, etc, are important features.

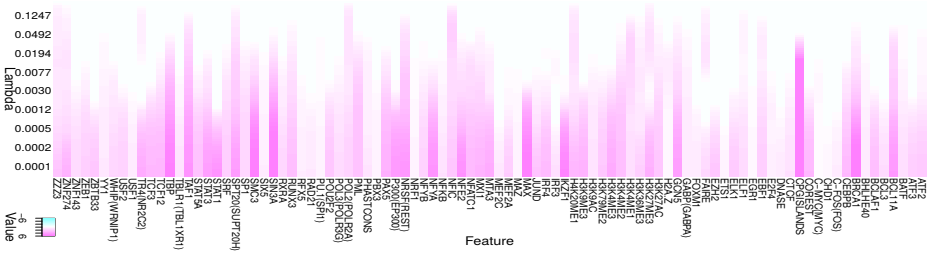
From GeneCards [20] and literature, we surveyed the known functionality of features selected by the deep and shallow DFS in an early phase. The functionality and specificity of these features are given in Table 1, where the last column is our conclusion about the binding specificity of the features based on the box-plots (not given due to page limit) of our data. First, the table shows that deep DFS identifies more key features earlier than shallow DFS, such as BCL11A, HeK27me3, H3K4me1, H4K20me1, NRSF, TAF1, and TBP. Interestingly, the deep DFS found a non-linear relation: TAF1 and TBP are actually components of TFIID functioning as RNA polymerase II locator. Second, we can see that the known functionality of the majority of selected features, as highlighted in bold in Table 1 (i.e. ELF1, H3K27ac, Pol2, BATF, EBF1, H3K36me3, H3K4me2, NFYB, RUNX3, BCL11A, H3K27me3, H3K4me1, and NRSF), are consistent with the binding specificity drawn from our data. From the box-plots (not given) of our data, we are also able to identify novel enrichment of some features (emphasized by italic type in Table 1) in enhancers and inactive elements. For example, H3K9ac is thought to be enriched in actively transcribed promoters [12], our result show that it is also enriched in active enhancers. H4K20me1 is reported to be enriched in exons [25], our result also show that both inactive enhancers and inactive promoters are enriched with H4K20me1. TAF1 and TBP is known as a promoter binder, our result shows that they are also associated with active enhancers. Finally, it has to be mentioned that some cell-specific features can be identified by the DFS models. From Table 1, we can see that ELF1 [6], BATF [11], EBF1 [18], and BCL11A [14] are specific to lymphoid cells (recall that GM12878 is a lymphoblastoid cell line from blood). This thus further confirms that the selected features are highly informative.



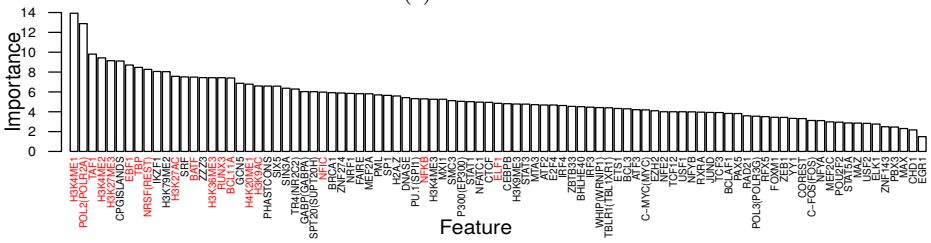
(a) Deep DFS



(b) Shallow DFS



(c) LASSO



(d) Random Forest

Fig. 3: Coefficients heatmaps of the DFS and LASSO models, and the importance of features ranked by random forest. In the heatmaps, as the value of λ decreases vertically down, more and more coefficient becomes nonzero. The strength of the colors indicates the involvement of features in classification. The higher a bar is, the earlier the corresponding feature affects the classification. Eventually, all features turn to nonzero, affecting the classification. A pink horizontal line in (a) is due to a failure of the stochastic gradient descent algorithm, which can be overcome by a different initial solution. In (d), the key features listed in Table 1 is coloured in red.

Table 1: Key features selected by the deep and shallow DFS models. The last column is the binding specificity of these features based on box-plots (not given) of our data. Features consistent between known functions and binding specificity are highlighted in boldface. Features having novel enrichment are emphasized in italic type. A: Active, I: Inactive, P: Promoter, E: Enhancer, Ex: Exon.

Feature	Known Functions	Specificity
ELF1	Primarily expressed in lymphoid cells. Bind to promoters and enhancers [6]. Act as both activator and repressor.	A-P, A-E
H3K27ac	Enriched in the flanking regions of active enhancers and active promoters [21].	A-P, A-E
Pol2	Encode RNA polymerase II to initialize transcription.	A-P
BATF	From AP-1/ATF superfamily. A negative regulator of AP-1/ATF transcriptional events. Interact with Jun family to recognize immune-specific regulatory element. Bind to enhancers [11].	A-E
EBF1	Bind to enhancers of PAX5 for B lineage commitment [18].	A-E
H3K36me3	Enriched in transcribed gene body.	A-E
H3K4me2	Define TF binding regions [26].	P, A-E
H3K9ac	Enriched in transcribed promoters [12].	A-P, A-E
NFIC	Promoter binding transcription activator [19].	A-P, A-E
NFYB	Bind specifically to CCAAT motifs in the promoter regions.	A-P
RUNX3	Serve as both activator and repressor. Bind to a core DNA sequence of a number of enhancers and promoters.	A-P, A-E
BCL11A	Involved in lymphoma pathogenesis, leukemogenesis, and hematopoiesis. Bind to promoters and enhancers [14].	A-E, A-P
H3K27me3	Enriched in closed or poised enhancers [21] and poised promoters [27].	I-P, I-E
H3K4me1	Enriched in enhancer regions [21].	A-E
H4K20me1	Enriched in exons [25].	A-Ex, I-P, I-E
NRSF/REST	Represses neuronal genes in non-neuronal tissues. With corepressors, recruit histone deacetylase to the promoters of REST-regulated genes.	A-P
TAF1	TAFs serve as coactivators. TAFs and TBP assemble TFIID to position RNA polymerase II to initialize transcription.	A-P, A-E
TBP	TATA-binding Protein. Interact with TAFs. Bind to core promoters.	A-P, A-E

Random forest is suitable for multi-class data. It can return the importance of each feature by measuring the decrease of out-of-bag error by permuting the values of this feature [7]. We compared the features selected by our models with the ones ranked by random forest, as shown in Fig. 3d. The majority of the key features selected by the DFS models are top-ranked in random forest, except that NFKB and ELF1 are scored as less important. It may be because

our DFS model considers the dependency of the features, while random forest independently measures the impact of removing each feature from the model.

4 Conclusion

Linear methods do not model the non-linearity of variables and can not be extended to multi-class in a natural way for feature selection, while deep models learn non-linearity and high-level representation of features. In this paper, we propose the deep feature selection model for selecting input features in a deep structure, especially for multi-class data. We applied this model to distinguish active promoters and enhancers from the rest of the genome. Our result shows that our shallow and deep DFS models are able to select a smaller subset of features than LASSO with comparable accuracy. Furthermore, our deep DFS can select discriminative features that may be overlooked by the shallow DFS. Through looking into the genomic features selected, we find that the features selected by DFS are biological plausible. Furthermore, some selected features have novel enrichment in regulatory elements. In future work, we will evaluate the new model on simulated data in order to further understand its behaviour. New sparse regulators and efficient model selection methods will be investigated to improve the performance of our model.

Acknowledgments. Dr. Anthony Mathelier (UBC) and Wenqiang Shi (UBC) provided valued suggestions. Dr. Anshul Kundaje (Stanford) provided valuable instruction during the processing of ChIP-seq data from ENCODE.

References

1. Ackley, D., Hinton, G., Sejnowski, T.: A learning algorithm for Boltzmann machines. Cognitive Science pp. 147–169 (1985)
2. Andersson, R., Gebhard, C., Miguel-Escalada, I., et al.: An atlas of active enhancers across human cell types and tissues. Nature 507, 455–461 (2014)
3. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(8), 1798–1828 (2013)
4. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: a CPU and GPU math expression compiler. In: The Python for Scientific Computing Conference (SciPy) (Jun 2010)
5. Bradley, P., Mangasarian, O.: Feature selection via concave minimization and support vector machines. In: International Conference on Machine Learning. pp. 82–90. Morgan Kaufmann Publishers Inc. (1998)
6. Bredemeier-Ernst, I., Nordheim, A., Janknecht, R.: Transcriptional activity and constitutive nuclear localization of the ETS protein Elf-1. FEBS Letters 408(1), 47–51 (1997)
7. Breiman, L.: Random Forests. Machine learning 45, 5–32 (2001)
8. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 33, 1–22 (2010)

9. Hinton, G., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science 313, 504–507 (2006)
10. Hinton, G., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. Neural Computation 18, 1527–1554 (2006)
11. Ise, W., Kohyama, M., Schraml, B., Zhang, T., Schwer, B., Basu, U., Alt, F., Tang, J., Oltz, E., Murphy, T., Murphy, K.: The transcription factor BATF controls the global regulators of class-switch recombination in both B cells and T cells. Nature Immunology 12(6), 536–543 (2011)
12. Kratz, A., Arner, E., Saito, R., Kubosaki, A., Kawai, J., Suzuki, H., Carninci, P., Arakawa, T., Tomita, M., Hayashizaki, Y., Daub, C.: Core promoter structure and genomic context reflect histone 3 lysine 9 acetylation patterns. BMC Genomics 11, 257 (2010)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
14. Lee, B., Dekker, J., Lee, B., Iyer, V., Sleckman, B., Shaffer, A.I., Ippolito, G., Tucker, P.: The BCL11A transcription factor directly activates rag gene expression and V(D)J recombination. Molecular Cell Biology 33(9), 1768–1781 (2013)
15. Li, Y.: Deep learning package, `https://github.com/yifeng-li/deep_learning`
16. Li, Y., Ngom, A.: Classification approach based on non-negative least squares. Neurocomputing 118, 41–57 (2013)
17. LISA Lab: Deep learning tutorials, `http://deeplearning.net/tutorial`
18. Nechanitzky, R., Akbas, D., Scherer, S., Gyory, I., Hoyler, T., Ramamoorthy, S., Diefenbach, A., Grosschedl, R.: Transcription factor EBF1 is essential for the maintenance of B cell identity and prevention of alternative fates in committed cells. Nature Immunology 14(8), 867–875 (2013)
19. Pjanic, M., Pjanic, P., Schmid, C., Ambrosini, G., Gaussin, A., Plasari, G., Mazza, C., Bucher, P., Mermod, N.: Nuclear factor I revealed as family of promoter binding transcription activators. BMC Genomics 12, 181 (2011)
20. Rebhan, M., Chalifa-Caspi, V., Prilusky, J., Lancet, D.: Genecards: Integrating information about genes, proteins and diseases. Trends in Genetics 13(4), 163 (1997)
21. Shlyueva, D., Stampfel, G., Stark, A.: Transcriptional enhancers: From properties to genome-wide predictions. Nature Review Genetics 15, 272–286 (2014)
22. The ENCODE Project Consortium: An integrated encyclopedia of DNA elements in the human genome. Nature 489, 57–74 (2012)
23. The FANTOM Consortium, The RIKEN PMI, CLST (DGT): A promoter-level mammalian expression atlas. Nature 507, 462–470 (2014)
24. Tibshirani, R.: Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological) 58(1), 267–288 (1996)
25. Vakoc, C., Sachdeva, M., Wang, H., , Blobel, G.: Profile of histone lysine methylation across transcribed mammalian chromatin. Molecular and Cellular Biology 26(24), 9185–9195 (2006)
26. Wang, Y., Li, X., Hua, H.: H3K4me2 reliably defines transcription factor binding regions in different cells. Genomics 103(2-3), 222–228 (2014)
27. Zhou, V., Goren, A., Bernstein, B.: Charting histone modifications and the functional organization of mammalian genomes. Nature Review Genetics 12, 7–18 (2011)
28. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. R. Stat. Soc. Series B Stat. Methodol. 67(2), 301–320 (2005)