


Gene expression

Overcoming the inadaptability of sparse group lasso for data with various group structures by stacking

Huan He, Xinyun Guo, Jialin Yu, Chen Ai and Shaoping Shi  *

Department of Mathematics and Numerical Simulation and High-Performance Computing Laboratory, School of Sciences, Nanchang University, Nanchang 330031, China

*To whom correspondence should be addressed.

Associate Editor: Anthony Mathelier

Received on August 5, 2021; revised on December 8, 2021; editorial decision on December 9, 2021; accepted on December 13, 2021

Abstract

Motivation: Efficiently identifying genes based on gene expression level have been studied to help to classify different cancer types and improve the prediction performance. Logistic regression model based on regularization technique is often one of the effective approaches for simultaneously realizing prediction and feature (gene) selection in genomic data of high dimensionality. However, standard methods ignore biological group structure and generally result in poorer predictive models.

Results: In this article, we develop a classifier named Stacked SGL that satisfies the criteria of prediction, stability and selection based on sparse group lasso penalty by stacking. Sparse group lasso has a mixing parameter representing the ratio of lasso to group lasso, thus providing a compromise between selecting a subset of sparse feature groups and introducing sparsity within each group. We propose to use stacked generalization to combine different ratios rather than choosing one ratio, which could help to overcome the inadaptability of sparse group lasso for some data. Considering that stacking weakens feature selection, we perform a *post hoc* feature selection which might slightly reduce predictive performance, but it shows superior in feature selection. Experimental results on simulation demonstrate that our approach enjoys competitive and stable classification performance and lower false discovery rate in feature selection for varying sets of data compared with other regularization methods. In addition, our method presents better accuracy in three public cancer datasets and identifies more powerful discriminatory and potential mutation genes for thyroid carcinoma.

Availability and implementation: The real data underlying this article are available from https://github.com/huanheaha/Stacked_SGL; <https://zenodo.org/record/5761577#.YbAUyciEwk2>.

Contact: shishaoping@ncu.edu.cn

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

With the development of high-throughput molecular techniques, the emergence of large-scale omics data provides an opportunity to study the genetic factors of complex diseases by statistical methods. A central problem in genomic research is to identify genes and pathways involved in diseases and other biological processes and to build a prediction model for future outcomes by linking high-dimensional genomic data, such as microarray gene expression data, to various cancer types (Li and Li, 2008).

Indeed, the number of recorded features p (as gene expression) being far larger than the sample size n , classical regression or classification methods are inappropriate (Hastie *et al.*, 2009), because spurious dependencies between features lead to singularities in the optimization processes, with neither unique nor stable solution (Durif *et al.*, 2018). From biological perspective, most genes do not provide useful information for diseases diagnosis, only a few genes

are strongly relevant with target diseases. Moreover, in genomics, each gene belongs to one or more genetic pathways, and genes in the same pathway may be expected to have correlated effects on the diseases of interest since these genes do not function (or fail to function) independently. These irrelevant or highly correlated genes usually introduce noise that reduces prediction performance. Consequently, this specific context of high dimensionality constitutes a major challenge for the development of new statistical methodologies (Donoho, 2000; Durif *et al.*, 2018), it calls for effective feature selection methods to help improve the accuracy of prediction and identify particularly ‘important’ genes in pathways of interest. Statistical machine learning methods based on regularization technique are well suitable for analyzing gene expression data.

In many genetic conditions, genes can be grouped into known biological pathways via the Kyoto encyclopedia of genes and

genomes (KEGG) pathways (Kanehisa *et al.*, 2016) or ‘genesets’ using cytogenetic position data (Simon *et al.*, 2013) which can be used in improving analysis. It is advisable to utilize such biological group information for modeling, because it can improve the interpretability and prediction accuracy of the model. Recently, some methods have been proposed to incorporate the group information into modeling. One of the popular methods is group lasso (GL) (Yuan and Lin, 2006) which performs feature selection on group level, it either includes a group so that none of the coefficients in the group are zero, or it excludes a group so that all of the coefficients in the group are zero. Meier *et al.* (2008) proposed an especially suitable algorithm to solve logistic regression model with GL penalization for high dimensional problem. Considering that GL cannot achieve sparsity within each group, Friedman *et al.* (2010) proposed the sparse group lasso (SGL) which is a more general penalty to perform sparsity at both the group and individual feature level by combining lasso (Tibshirani, 1996) penalty and GL penalty. Then the SGL penalization (Simon *et al.*, 2013) was extended to linear, logistic and cox regression with application of cancer classification. Besides, clinical, mRNA expression and methylation data can be available for the same patient cohort. On this basis, Klau *et al.* (2018) introduced a simple hierarchical approach named Priority-Lasso which works on data blocks with a pre-specified order/priority. Another similar approach is Integrative L1-Penalized Regression with Penalty Factors (IPF-Lasso, Boulesteix *et al.*, 2017), which is also a variant of Lasso that essentially assigns different penalty weights to multiple omics modalities or blocks that are usually clinical data, gene expression, methylation and so on. This method based on the idea of weighting to integrate various data sources is also applied to WDFSMT (Qiu *et al.*, 2021). In certain cases, the ‘modalities’, ‘blocks’ or ‘data from various sources’ can essentially be treated as ‘groups’.

Unfortunately, the SGL method is not adaptable to some data with various group structures, which leads to unstable prediction results. It has been shown that it is certainly not perfect for every case of grouped data. This is largely because that different sparsity levels within the group require different ratios of lasso and GL, which are usually unknown. One issue with the SGL is that it has two tuning parameters: usually a regularization parameter λ and a mixing parameter α which is used to determine the ratio between lasso and GL. In some previous applications of SGL method (Che *et al.*, 2020; Simon *et al.*, 2013), α was fixed and then the optimal regularization parameter λ was found through cross-validation. Generally, fixing α close to GL could cause the model to be not sparse enough. Or fixing α to lasso might result in a sparser model, but it introduces a higher false discovery rate. In other applications (Chen *et al.*, 2020; Mendez-Civieta *et al.*, 2020; Samal *et al.*, 2017), a good combination of the two parameters can be selected via cross-validation using training data over a grid of α and λ . However, it is easy to miss the optimal α . Moreover, tuning both α and λ is notoriously hard due to the flat cross-validated likelihood landscape (van de Wiel *et al.*, 2019).

In this article, we first propose to use stacked generalization (Wolpert, 1992) to combine multiple values of α by integrating biological group structures in logistic regression framework for cancer classification, rather than fixing or tuning α . We illustrate the effect of different α on the prediction performance of SGL through simulation experiments. Each α denotes one learner. Then, the meta learner combines the results of all base learners using a set of prediction class probabilities of the base learners to replace the single class prediction as the input of the meta learner. We apply multi-response linear regression (MLR) with l_1/l_2 regularization as meta learner which could transform the classification problem into a multi-response regression problem. Especially, we demonstrate that stacking improves the prediction accuracy, and that our method is more stable regarding the data with various group information compared to other methods. Furthermore, decoupling shrinkage and selection (Hahn and Carvalho, 2015) allow us to still achieve feature selection after model fitting. After that, the optimal α based on cross-

validation results of multiple base learners is used as the mixing parameter of the feature selection model. Therefore, given all the criteria (prediction, stability, selection), our approach can perform correctly, while the other methods have a weakness.

Here, prediction and feature selection can be executed by two models. The training accuracy of predictive model will influence the prediction performance of later *post hoc* feature selection model and slightly reduce its prediction performance. However, there is a trade-off between predictive performance and feature selection. The *post hoc* feature selection can be performed correctly despite having lower predictive performance.

The rest of the article is composed of four sections as follows. In Section 2, we first introduce stacked SGL in logistic regression framework and present how to perform feature selection after model fitting. In Section 3, we assess the performance of our proposed approach on extensive simulated data and compare with other methods. In Section 4, the proposed method is applied to classify tumor and normal tissue in three public cancer gene expression data and identify potential mutation genes of thyroid carcinoma. In Section 5, conclusion of the article is presented.

2 Materials and methods

2.1 Base learner

In this article, we focus on a general binary classification problem. For example, for a cancer classification problem, we have the binary responses as cancer or health and genes are grouped into some gene pathways. Suppose that data consist of an n -dimensional binary response vector $y = \{y_1, y_2, \dots, y_n\}$, and an n by p variate matrix $X = \{X^1, X^2, \dots, X^m\}$ whose columns corresponding to the features are divided into m groups. X_i^l represents i th sample with variates in the feature group l , β^l is the regression coefficient vector of that group and p_l is the length of β^l . Define a classifier $f(x) = e^x / (1 + e^x)$ and the logistic regression is defined as

$$P_i = P(y_i = 1|X_i) = f(\eta(X_i)) = \frac{\exp(\eta(X_i))}{1 + \exp(\eta(X_i))},$$

where $\eta(X_i) = \sum_{l=1}^m X_i^l \beta^l$. $\beta \in R^p$ denotes the whole parameter vector, i.e. $\beta = (\beta^1, \beta^2, \dots, \beta^m)^T$.

The logistic SGL estimator $\hat{\beta}$ is given by

$$\arg \min_{\beta} l(\beta) + (1 - \alpha)\lambda \sum_{l=1}^m \sqrt{p_l} \|\beta^l\|_2 + \alpha\lambda \|\beta\|_1,$$

where $l(\cdot)$ is the negative log-likelihood function:

$$l(\beta) = \frac{1}{n} \sum_{i=1}^n (-y_i \eta(X_i) + \ln[1 + \exp(\eta(X_i))]).$$

$\lambda > 0$ is regularization parameter which controls the amount of penalization, and generally a larger λ would punish more coefficients to zeros. $\alpha \in [0, 1]$ is mixing parameter that determines the ratio between lasso and GL penalty. In other words, it strikes a balance between the sparse selection of the entire feature groups and the sparse selection within each feature group.

In this article, we propose stacked SGL to fit the logistic regression for classifying different cancer types. Most previous studies had fixed α and selected an optimal parameter λ via cross-validation. However, here we consider T different values for α that are equally spaced in $(0, 1)$ and indexed by $t \in \{1, 2, \dots, T\}$. For each α_t , cross validation is performed to select the optimal regularization parameter λ_t . We record the predicted probabilities of the corresponding validation set in the $n * T$ matrix P^{cv} .

2.2 Meta learner

Any classification problem with real-valued attributes can be transformed into a multi-response regression problem (Ting and Witten, 1999). If the original classification problem has K classes, it is

converted into K separate regression problems. In dichotomy problems, for the samples of the positive class we set their responses to be equal to one and zero otherwise.

We regress the outcome on the base learner by MLR model which takes the form as $Y = \hat{P}^{cv} W + W^*$, where $\hat{P}^{cv} \in R^{n \times T}$ is the probability matrix of validation set outputted from T base learners, $W = (w_{ik})_{1 \leq i \leq T, 1 \leq k \leq K}$ is a $T \times K$ regression coefficient matrix and an $n \times K$ matrix W^* represents zero-mean noise.

Then, the l_1/l_2 -constrained objective function called multi-response linear regression lasso (MLR lasso) is adopted via

$$\arg \min_{W \in R^{T \times K}} \frac{1}{2n} \|Y - \hat{P}^{cv} W\|_F^2 + \lambda \|W\|_{l_1/l_2},$$

where $Y \in R^{n \times K}$ is the matrix of observations, the $\|\cdot\|_F$ is the Frobenius norm (Obozinski et al., 2011), $\|W\|_{l_1/l_2}$ is the block l_1/l_2 norm

$$\|W\|_{l_1/l_2} = \sum_{i=1}^T \sqrt{\sum_{j=1}^K w_{ij}^2} = \sum_{i=1}^T \|w_i\|_2,$$

when $K = 1$, it is the standard lasso. The input of MLR lasso is the cross-validated outcome matrix \hat{P}^{cv} where the attributes are probabilities. Using the same cross-validation folds as the base learners, we select the optimal penalty parameter λ^* for the meta learner. It is worth noting that, in the two consecutive cross-validation loops, we use the same training sets to estimate the base and meta parameters (β given α_t for all t ; W and W^*), and use the same validation sets to tune the base and meta regularization parameters (λ_t for all t ; λ^*).

For the sample i in class k , it is clear that $y_k^i = \sum_{j=1}^T w_{jk} \hat{P}_{ij}^{cv} + w_{ik}^*$, where w_{jk} and w_{ik}^* are the corresponding coefficient of k column in W and W^* , \hat{P}_{ij}^{cv} represents the output probability of the i^{th} sample on the j^{th} base learner. With this in place, we then can describe the working of MLR. To classify a new sample, compute y_k for all K classes and assign the sample to class k which has the greatest value:

$$\arg \max_{k \in \{1, \dots, K\}} y_k.$$

2.3 Feature selection

By solving above model, we can get the predicted result \hat{y} of the dataset X . Stacked SGL might still perform feature selection. It selects a feature if and only if the meta learner selects the base learner that selects that feature. However, it would worsen the feature selection property of the SGL and make the model not get enough sparse solution, so we suggest performing *post hoc* feature selection (Rauschenberger et al., 2020). Decoupling shrinkage and selection (Hahn and Carvalho, 2015) allow us to perform feature selection after model fitting.

The idea is to approximate $\sum_{l=1}^m X_i^l \beta^l$ by $\sum_{l=1}^m X_i^l \gamma^l$ where β is dense but γ is sparse. The new loss function could be defined as

$$l(\gamma) = \frac{1}{n} \sum_{i=1}^n (-\hat{y}_i \hat{\eta}(X_i) + \ln[1 + \exp(\hat{\eta}(X_i))]), \hat{\eta}(X_i) = \sum_{l=1}^m X_i^l \gamma^l.$$

Considering that features are grouped, we replace l_1 norm of equation (A.5) in Hahn and Carvalho (2015) by SGL penalization:

$$\arg \min_{\gamma} l(\gamma) + (1 - \alpha^*) \lambda \sum_{l=1}^m \sqrt{p_l} \|\gamma^l\|_2 + \alpha^* \lambda \|\gamma\|_1.$$

As λ increases from 0 to ∞ , the number of features selected by the model is gradually reduced to 0. Here, we could select the optimal α^* based on cross-validated average area under the receiver operating characteristic (ROC) curve (AUC) values of T base learners trained by different mixing parameters α . The highest AUC values of the validation set are recorded in the training process of T base learners since one α denotes one base learner. The *post hoc* feature selection depends on the performance of preceding stacked SGL prediction. Determining the optimal λ via the same cross-validation set, or adjusting λ makes the model contain a certain number of non-zero coefficients.

3 Simulations

In this section, simulation studies are conducted to evaluate the predictive performance and feature selection ability of the proposed method. The simulation design is similar to Simon et al. (2013), but accounts for additional complexities of varied group structures. Different features, observations and groups are simulated our feature matrix. The columns of X are *iid* Gaussian, and the binary response y_i is transformed by probability p_i which is constructed as

$$p_i = \frac{\exp(5y_i^*)}{1 + \exp(5y_i^*)}, \quad (1)$$

where $y_i^* = \sum_{l=1}^g X_i^l \beta^l + \sigma \epsilon$, $\epsilon \sim N(0, 1)$, β^l is generative group for $l = 1, \dots, g$, and σ is a signal-to-noise ratio set to 0.1. Setting $y_i = 1$ when $p_i > 0.5$ and 0 otherwise. The number of generative groups, g , varies from 2 to 4 to 6 changing the amount of the sparsity groups.

3.1 Simulation of data

We run three blocks of simulation for different complexities, including number of non-null groups, number of the significant variables within group, group size and spread of the significant variables among groups. In order to show the effect of number of non-zero groups on modeling, we vary the number of generative groups (number of non-null groups) to 2, 4 and 6 for each dataset.

In the first scenario, we generate $n = 300$ observations, each with a binary response y_i and a vector of $p = 1000$ continuous variables X_i . The 1000 variables are divided into $m = 20$ groups with equal size $p_l = 50$. In this scenario, we are interested in studying the effect of the number of the significant variables within group. So, two cases of significant variable distributions with different degree of intra-group sparsity are considered for each generative group. Then, the number of significant variables is set to 10 and 20 respectively for each generative group.

In the second scenario, we generate $n = 300$ observations, each with a binary response y_i and a vector of $p = 3000$ continuous variables X_i . To investigate the effect of group size on modeling, the 3000 variables are organized into $m = 20$ groups with equal size $p_l = 150$ and $m = 50$ groups with equal size $p_l = 60$ for varying group sizes by considering two cases. The number of significant variables is set to 10 for each generative group.

In the third scenario, we simulate $n = 300$ observations and $p = 5000$ variables. And there are $m = 200$ groups of size $p_l = 25$ each. Another important factor is the spread of the significant variable among different groups. To study this aspect, two cases are considered. For the case1, each generative group includes 5 significant variables. For the case2, half of the generative groups are sparse distribution with 5 significant variables within group, while the other halves are dense distribution with 25 significant variables within group. The details of coefficient definitions for the above three simulation scenarios are shown in Supplementary Information.

3.2 Metric

In each simulation scenario, we repeat the experiments 100 times, with 80% of the samples used for 10-fold cross-validation and 20% of the samples for independent testing. To examine the prediction performance of the stacked SGL, we compare Priority-Lasso (PLasso) in the R package prioritylasso, IPF_Lasso in the R package ipflasso, lasso, GL, SGL (with α and λ choosing by cross-validation), stacked SGL and stacked SGL with *post hoc* feature selection (stacked-hoc) by calculating the AUC values and accuracy (Acc) on independent test set. To indicate statistical significance of the performance difference between any two methods, we record the AUC values under 10-fold cross-validation for each method, then calculate the P -value of performance between each pair of compared models through the paired t -test. The details of P -values are listed in Supplementary Tables S1–S3.

According to the pre-setup of the generated model, we can know the effect of the features on the outcome. Not only can we examine the prediction performance of our method, but we can assess its feature selection ability for the same simulation dataset. Some

applications require model to select a limited number of features. To prove that stack would weaken the ability to select features, we allow stacked SGL, stacked SGL with *post hoc* feature selection (stacked-hoc), lasso and SGL to perform feature selection so that the number of non-zero coefficients included in the model matches the true number of non-zero coefficients in the generative model in Equation (1). For example, choosing at most 20 features correspond to case1 of the first scenarios for $g = 2$. Additionally, the metrics used to evaluate feature selection performance are defined as follows:

$$\begin{aligned} \text{Tp} &:= \sum_{i=1}^p \mathbb{I}(\hat{\beta}_i \neq 0 \cap \beta_i \neq 0) & \text{Fp} &:= \sum_{i=1}^p \mathbb{I}(\hat{\beta}_i \neq 0 \cap \beta_i = 0) \\ \text{Fn} &:= \sum_{i=1}^p \mathbb{I}(\hat{\beta}_i = 0 \cap \beta_i \neq 0) & \text{Tp} &:= \sum_{i=1}^p \mathbb{I}(\hat{\beta}_i = 0 \cap \beta_i = 0). \end{aligned}$$

We mainly report the results of false discovery rate, false negatives rate and F1-score in simulation study, among them, minimizing false discovery rate is our primary objective:

False discovery rate $\text{FDR} := \text{Fp}/(\text{Tp} + \text{Fp})$, i.e. the rate of inactive selected features (false positive Fp) among those selected (Fp and true positive Tp).

False negatives rate $\text{FNR} := \text{Fn}/(\text{Tp} + \text{Fn})$, i.e. the rate of features excluded by the model (false negatives Fn) among the active ones (Tp and Fn).

F1-score := $2 * \text{Tp}/(2 * \text{Tp} + \text{Fp} + \text{Fn})$, summarizing the FDR and the FNR.

In real applications, reliable selection methods are those who are characterized by a low FDR, together with as low FNR as possible, namely a high F1-score (Belhechmi *et al.*, 2020).

3.3 Results of simulation

Results in Table 1 report the average Acc on corresponding independent test set of each scenario based on 10-fold cross-validation. From the Table 1, we can see that the stacked SGL improves prediction performance in most cases especially in the case2 of all scenarios. Besides, IPF_Lasso and PLasso perform worse than stacked SGL and even lower than SGL in almost all case ($P \leq 0.05$), since they are more suitable for the smaller number of groups which usually ranges from 2 to 10 (Klau *et al.*, 2018). For a large number of

groups, it is difficult for researchers to acquire the accurate rank of group priorities, so multiple penalty parameters need to be selected through cross verification, which will increase the cost of calculation.

Supplementary Figures S1 and S2 represent the average AUC on the independent test of case 1 and case 2 in the three scenarios under 10-fold cross-validation, respectively. In scenario 2 and scenario 3, comparing our proposed classifier with lasso, GL and SGL, respectively, stacked SGL nearly attains higher Acc and AUC than other methods. For example, in case1 of scenario 2, stacked SGL performs better than the GL and SGL with a 0.934 accuracy rate to 0.900 for the SGL and 0.833 for the GL when $g = 2$. In case1, the lasso or lasso variant of IPF_Lasso sometimes perform well mainly because of the low intra-group sparsity in these sets. Similarly, in case2 of scenario 3, GL, SGL and Stacked SGL perform equally well due to the presence of groups that are completely dense. Furthermore, as shown in Table 1, comparing to SGL, stacked SGL maintains relatively superior and stable performance. This matches our expectation because stacking leads to more stable outcome than single learner.

From the last row of Supplementary Figures S1 and S2, comparing from case1 to case2, if the whole group level effect is introduced, the performance of GL and SGL almost all improve while lasso significantly decreases, while stacked SGL remains the highest. In other words, as the number of candidate features in group level rises, it seems more reasonable to employ GL or SGL than lasso, while stacked SGL may still the best choice.

Figure 1 presents the cross-validated AUC of Lasso, GL, SGL, Stacked SGL and stacked-hoc under 10-fold cross-validation. We observe that Stacked SGL remains superior almost consistently on validation sets. From Table 1 and Figure 1, the prediction performance of stacked-hoc is closer to that of a single SGL and sometimes lower than SGL, but always lower than that of stacked SGL. The training performance of Stacked SGL does not always reach 1, resulting in reduced predictive performance of the stacked-hoc model. However, this imperfect nature does not reduce its feature selection ability. Actually, we do not execute stacked-hoc for forecasting. Researchers can implement stacked SGL directly by focusing only on predictive performance. If they also pursue feature selection, they could perform *post hoc* feature selection. Additionally, we summarize the average values of each measure index for feature selection in Table 2. As is shown in Table 2, the

Table 1. Average accuracy on the independent test set of simulation based on 10-fold cross-validation

Method	Scenario 1			Scenario 2			Scenario 3		
	Number of generative groups			Number of generative groups			Number of generative groups		
G	2	4	6	2	4	6	2	4	6
Case1									
PLasso	0.834	0.728	0.683	0.900	0.707	0.660	0.838	0.816	0.740
IPF_Lasso	0.833	0.750	0.750	0.850	0.817	0.667	0.850	0.883	0.717
Lasso	0.800	0.791	0.802	0.867	0.617	0.617	0.850	0.888	0.674
GL	0.883	0.717	0.750	0.833	0.650	0.550	0.817	0.783	0.754
SGL	0.867	0.783	0.783	0.900	0.717	0.620	0.850	0.833	0.767
Stacked SGL	0.850	0.781	0.788	0.934	0.728	0.629	0.900	0.850	0.767
Stacked-hoc	0.833	0.783	0.783	0.883	0.713	0.595	0.833	0.850	0.683
Case2									
PLasso	0.804	0.718	0.643	0.750	0.751	0.682	0.849	0.789	0.761
IPF_Lasso	0.800	0.700	0.617	0.750	0.783	0.667	0.717	0.733	0.650
Lasso	0.783	0.717	0.669	0.800	0.727	0.683	0.783	0.700	0.683
GL	0.883	0.750	0.700	0.733	0.684	0.667	0.883	0.850	0.800
SGL	0.883	0.733	0.667	0.800	0.783	0.683	0.883	0.850	0.800
Stacked SGL	0.900	0.783	0.736	0.800	0.802	0.700	0.883	0.850	0.817
Stacked-hoc	0.900	0.750	0.658	0.800	0.800	0.600	0.850	0.833	0.783

Note: The numbers in bold are the highest Acc between these methods of the data. Stacked-hoc represents the proposed stacked SGL with *post hoc* feature selection.

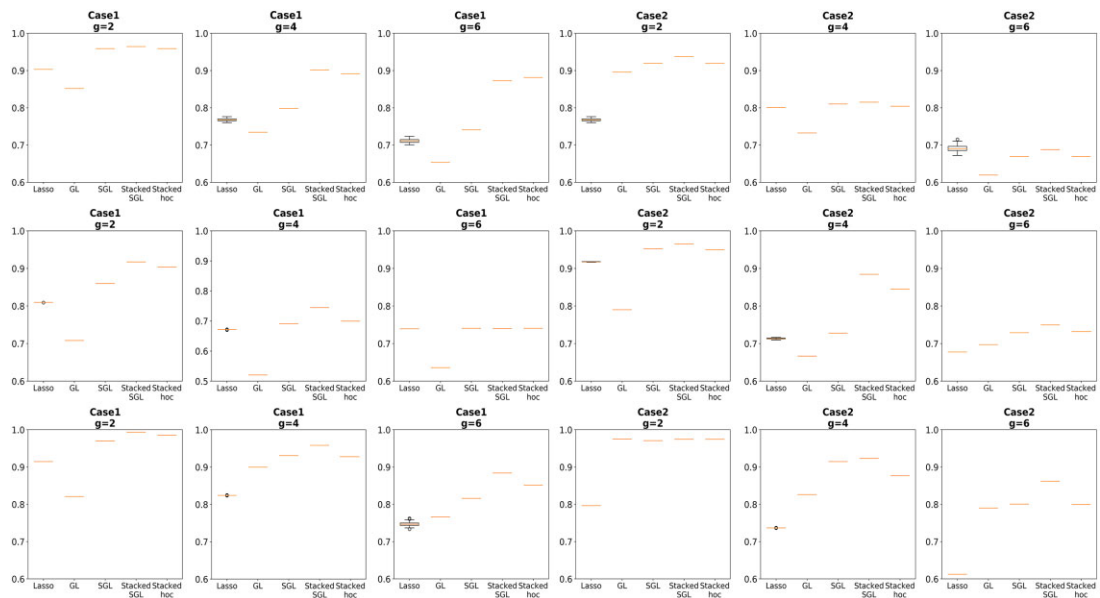


Fig. 1. The first row, second row and last row express the AUC on the corresponding cross-validated set under 10-fold cross-validation in scenario 1, scenario 2 and scenario 3, respectively

Table 2. The results of three metrics for feature selection in simulation

G	Method	Scenario 1			Scenario 2			Scenario 3		
		FDR	FNR	F1	FDR	FNR	F1	FDR	FNR	F1
Case1										
2	Lasso	0.474	0.500	0.513	0.333	0.500	0.572	0.333	0.400	0.632
	SGL	0.350	0.350	0.650	0.235	0.350	0.703	0.200	0.200	0.800
	Stacked SGL	0.913	0.913	0.088	0.803	0.813	0.192	0.857	0.871	0.135
	Stacked-hoc	0.389	0.450	0.579	0.294	0.400	0.649	0.200	0.200	0.800
4	Lasso	0.641	0.650	0.354	0.625	0.625	0.375	0.450	0.450	0.550
	SGL	0.391	0.650	0.444	0.550	0.550	0.450	0.375	0.500	0.555
	Stacked SGL	0.654	0.670	0.338	0.846	0.850	0.152	0.675	0.800	0.247
	Stacked-hoc	0.333	0.400	0.632	0.629	0.675	0.347	0.400	0.400	0.600
6	Lasso	0.552	0.567	0.441	0.741	0.750	0.254	0.667	0.667	0.333
	SGL	0.517	0.533	0.475	0.678	0.700	0.310	0.696	0.767	0.264
	Stacked SGL	0.680	0.722	0.297	0.799	0.808	0.196	0.573	0.627	0.397
	Stacked-hoc	0.517	0.517	0.483	0.678	0.700	0.310	0.586	0.600	0.407
Case2										
2	Lasso	0.531	0.625	0.417	0.632	0.650	0.359	0.700	0.700	0.300
	SGL	0.575	0.575	0.425	0.316	0.350	0.667	0.000	0.200	0.889
	Stacked SGL	0.509	0.585	0.445	0.742	0.750	0.254	0.000	0.167	0.909
	Stacked-hoc	0.484	0.600	0.451	0.143	0.400	0.706	0.000	0.167	0.909
4	Lasso	0.675	0.675	0.325	0.639	0.675	0.342	0.814	0.817	0.185
	SGL	0.641	0.650	0.354	0.486	0.525	0.494	0.317	0.317	0.683
	Stacked SGL	0.568	0.615	0.405	0.645	0.675	0.339	0.385	0.488	0.552
	Stacked-hoc	0.633	0.725	0.314	0.436	0.450	0.557	0.246	0.283	0.735
6	Lasso	0.750	0.750	0.250	0.783	0.783	0.217	0.867	0.867	0.133
	SGL	0.643	0.708	0.321	0.641	0.767	0.283	0.403	0.211	0.679
	Stacked SGL	0.736	0.744	0.260	0.800	0.835	0.180	0.590	0.622	0.393
	Stacked-hoc	0.640	0.700	0.327	0.633	0.700	0.310	0.221	0.178	0.800

Note: The numbers in bold are the optimal values in the column of metrics. Stacked-hoc represents the proposed stacked SGL with *post hoc* feature selection.

experimental results help provide evidence that stacking weakens feature selection. Compared to the case2 in scenario 1 and scenario 2, stacked-hoc has a lower FDR in case2 of scenario 3 which introduces non-null groups with whole non-null coefficients, it is 0.484, 0.633, 0.640 in scenario 1 and 0.143, 0.436, 0.633 in scenario 2 respectively against 0.000, 0.246, 0.221 in scenario 3. In scenario 3, SGL presents greater advantage in feature selection compared to

lasso, while stacked-hoc has lower FDR, a more comparable FNR and a higher F1-score than SGL. Moreover, the performance of feature selection for all three methods decreases as the number of generative groups g increases, which usually introduces stronger shrinkage amount and much more noise, but the FDR of stacked-hoc remains lower than the other methods. It is verified that stacked-hoc is expected to achieve better quality in feature selection

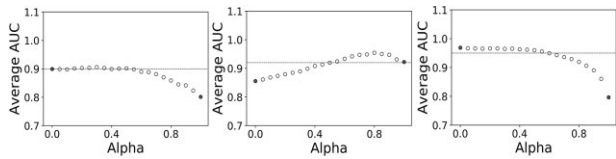


Fig. 2. Average cross-validated AUC against the stacked SGL mixing parameter, for $g = 2$ in case2 of scenario 1 (left), $g = 2$ in case2 of scenario 2 (center), $g = 2$ in case2 of scenario 3 (right)

than lasso and SGL for dealing with the situation of lower sparse level within the groups.

From Table 2, it is not surprising that lasso maintains the worst selection performance with highest FDR in almost all cases, suggesting that modeling sparsity within groups would contribute to identify more candidate features and reduce FDR for some datasets with group information. For example, in case1 of scenario 1, lasso has a higher mean FDR, a lower mean F1-score than stacked-hoc in $g = 2(0.474 > 0.389, 0.513 < 0.579)$, $g = 4(0.641 > 0.333, 0.354 < 0.632)$, $g = 6(0.552 > 0.517, 0.441 < 0.483)$.

Figure 2 shows the average cross-validated AUC from base learners with different mixing parameters for $g = 2$ in case2 of scenario 1 (left), $g = 2$ in case2 of scenario 2 (center), $g = 2$ in case2 of scenario 3 (right). In the ‘center’ situation, the AUC of cross-validation set increases between 0 (GL) and some α , and then decreases between this α and 1 (lasso). However, for the ‘left’ and ‘right’ situation, the AUC nearly keeps decreasing from 0 to 1. The optimal mixing parameter of SGL is $\alpha = 0.3$ for ‘left’, $\alpha = 0.8$ for ‘center’, $\alpha = 0$ for ‘right’ among all cross-validation repetitions. This means that for data with different dimensionality and varying group information such as group sizes and sparsity level within the group, the corresponding optimal mixing parameter α is different. In other words, if the optimal mixing parameter value for new data has been known before analysis, we would reach preferable prediction performance. Seeking for the optimal α across cross-validation iteration, we would either find or miss it. This is why stacked SGL may outperform the SGL with optimal α . For example, for the center situation, the AUC value of stacked SGL is 0.969, compared to 0.954 of SGL with optimal $\alpha = 0.8$.

4 Applications to real data

To further investigate the effectiveness of stacked SGL on real genomic data, we applied our method to analyze three public real datasets: liverhepatocellular carcinoma (LIHC), thyroid carcinoma (THCA) and two subtypes of lung cancer including lung adenocarcinoma (LUAD) and lung squamous-cell carcinoma (LUSC) downloaded from The Cancer Genome Atlas (TCGA, <http://cancergenome.nih.gov/>). We grouped genes into ‘genesets’ via using cytogenetic position data (the C1 set from Subramanian et al., 2005) (Simon et al., 2013). We first downloaded gene expression data via R/TCGAbiolinks and then employed R/tibble to isolate mRNA expression data. For each cancer data, the details are described below.

4.1 Cancer Data

4.1.1 LIHC dataset (mRNA-sequencing data)

Our first data is LIHC mRNA gene expression data which is total 100 samples, including 50 liver tumor tissue samples along with 50 normal liver tissue samples from TCGA. Each sample has expression data of original 19 565 genes. Firstly, we reduced the number of genes to 10 057 of the most differentially expressed genes with $p \leq 0.01$, by using a standard *t*-test with a Benjamini-Hochberg correction (Durif et al., 2018). Then these genes were mapped to 192 genesets including 5183 genes via performing GSEA C1 analysis. We repeatedly trained and validated the models for 5-fold cross-validation with 30 tumor samples and 30 normal samples, and

Table 3. Classification results of the three cancer datasets under 5-fold cross-validation

Dataset	Method	Cross-validation Acc	Testing Acc	Testing AUC
LIHC	PLasso	*	0.8770	0.9239
	IPF_Lasso	*	0.9750	0.9525
	Lasso	1.0000	0.9750	0.9500
	GL	0.9833	0.9750	0.9625
	SGL	0.9833	0.9750	0.9500
	Stacked SGL	1.0000	0.9750	0.9775
	Stacked-hoc	1.0000	0.9750	0.9525
THCA	PLasso	*	0.8513	0.9237
	IPF_Lasso	*	0.9278	0.9500
	Lasso	0.9875	0.9438	0.9733
	GL	0.9750	0.8958	0.9759
	SGL	0.9750	0.9167	0.9778
	Stacked SGL	0.9625	0.9583	0.9815
	Stacked-hoc	0.9625	0.9167	0.9722
Lung	PLasso	*	0.9483	0.9800
	IPF_Lasso	*	0.9617	0.9808
	Lasso	0.9454	0.9570	0.9792
	GL	0.9343	0.9650	0.9842
	SGL	0.9357	0.9554	0.9847
	Stacked SGL	0.9471	0.9682	0.9849
	Stacked-hoc	0.9357	0.9554	0.9859

Note: The numbers in bold are the optimal values in the column of metrics. The * indicates that corresponding metric value has not been calculated.

tested the models with the remaining 20 tumor samples and 20 normal samples.

4.1.2 THCA dataset (mRNA-sequencing data)

The second dataset includes THCA mRNA gene expression data which is total 128 samples, including 70 thyroid tumor tissue samples along with 58 thyroid normal tissue samples from TCGA. Each sample has expression data of original 19 565 genes. Similar with the steps for above dataset, we got 3073 genes mapped to 188 pathways via performing GSEA C1 analysis. 40 tumor samples and 40 normal samples were randomly chosen to repeatedly implement 5-fold cross-validation for determining the λ value yielding the maximum cross-validation AUC and the remaining samples were used to test.

4.1.3 Lung cancer dataset (mRNA-sequencing data)

We downloaded two subtypes of non-small cell lung cancer mRNA gene expression dataset with 513 lung adenocarcinoma (LUAD) samples and 501 lung squamous-cell carcinoma (LUSC) samples from TCGA via removing repeated sequencing samples. Then, we mapped 4874 genes to 230 pathways by same preprocess. 350 LUAD patients and 350 LUSC patients were chosen at random to build the models via 5-fold cross-validation. The remaining 314 samples served as the independent test set. The description of the three cancer datasets is listed in the Supplementary Table S4.

Considering that both LIHC and THCA datasets are small sample sets, we implemented an extra leave-one-out cross-validation (LOOCV) to test the prediction performance. The results are presented in Supplementary Table S5.

4.2 Results of cancer data

Since there are a large number of small pathways, we chose the SGL with optimal α and λ as the prediction performance comparison. Model complexity is discussed in Supplementary Information. Table 3 summarizes the prediction results of the three cancer datasets. For the three datasets, the proposed stacked SGL generally performs better

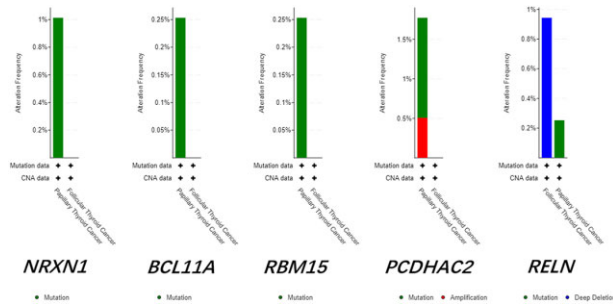


Fig. 3. Frequency of mutations in selected genes in thyroid cancer downloaded from cBioPortal website

than the other methods based on the Acc and AUC in the independent test set. For THCA, the classifier of the stacked SGL approach gave the average Acc of 0.9583 and the average AUC of 0.9815 in the independent test set, however the classifiers of PLasso, IPF_Lasso, lasso, GL and SGL gave the average Acc of 0.8513, 0.9278, 0.9438, 0.8958, 0.9167 and the average AUC of 0.9237, 0.9500, 0.9733, 0.9759, 0.9778 respectively. It can be seen that the stacked SGL likewise achieves the best performance with the highest Acc in the cross-validation set and independent test set of Lung.

Furthermore, we extracted different top 30 genes with the largest absolute value of estimated coefficient corresponding to lasso, SGL, stacked-hoc for THCA gene expression dataset. Based on the top 30 genes selected by the three methods, we applied the typical logistic regression to re-train the training set of THCA. The results of independent test set based on logistic regression are displayed in the [Supplementary Table S6](#).

The classification accuracy of logistic regression based on selected genes by stacked-hoc was 0.9792, higher than 0.9583 based on genes extracted by SGL and 0.9375 based on genes extracted by lasso. Besides, we noticed that the performance of the logistic regression model with the top 30 genes selected by stacked-hoc in the independent test set was improved to 0.9792 from before 0.9583 in [Table 3](#). This implies that it is valid to perform feature selection for THCA gene expression dataset, whose original data would have noise resulting in a negative impact on prediction performance.

In addition to considering the effect of the selected feature to predictive performance, we also explored whether these selected genes were potentially mutated genes in a biological sense. For this purpose, we sought to identify genes that might be involved in the causation of the THCA and selected exactly 100 genes for each method ([Wang et al., 2019](#)). We filtered the selected genes by intersection with Catalogue of Somatic Mutations in Cancer (COSMIC) ([Forbes et al., 2015](#)). Next, we generated a list of potential mutation in thyroid carcinoma via combining the annotations of thyroid carcinoma in COSMIC, IntOGen ([Gonzalez-Perez et al., 2013](#)) and CBioProtal ([Cerami et al., 2012](#)).

The results were that lasso and SGL both chose *NRXN1*, *BCL11A*, *RBM15*, while stacked-hoc selected *NRXN1*, *BCL11A*, *RBM15*, *PCDHAC2*, *RELN*. Not only does the stacked-hoc identify more genes with known potential oncogenic somatic mutations than other two methods do, but also the combination of selected genes by stacked-hoc is more contributed to classify thyroid cancer. For example, the stacked-hoc selected *NRXN1* and *PCDHAC2*, which both have been potentially mutated genes with a mutation frequency of more than 1% recorded in CBioProtal ([Fig. 3](#)). In contrast, another two methods both only selected one probable somatic mutation, *NRXN1*. Besides, *BCL11A*, *RELN* and *RBM15* have been recorded as oncogenic genes in OncoKB ([Zehir et al., 2017](#)).

Finally, we simply considered training a logistic regression model with these potential oncogenic genes. The accuracy of logistic classifier with *NRXN1*, *BCL11A*, *RBM15* was 0.8958, lower than 0.9167 based on *NRXN1*, *BCL11A*, *RBM15*, *PCDHAC2*, *RELN* which were selected by our method. The results indicate stacked-hoc selection can select more powerful discriminatory genes with the addition of group information.

5 Conclusion

The SGL is a popular method for identifying genes associated with the phenotype of interest in biogenetics applications, because it explicitly considers the correlation between features. It has a mixing parameter α that represents the ratio of lasso to GL, however the optimal ratio is not known in most case. Consequently, it may not be suitable for some datasets and result in unstable prediction performance. Instead of choosing one α via tuning, we propose to combine multiple α in logistic regression framework by stacking and consider a set of predicted probability values as the input on MLR which can convert an original classification problem into multiple regression problems. The results of simulation and real experiments show that stacked SGL presents competitive and stable classification performance for varying sets of data. Moreover, the increase in computational complexity is negligible, because only one additional low-dimensional linear regression calculation from the meta learner is added.

In contrast to single SGL, stacked SGL weakens feature selection, because it selects a feature if and only if the meta learner selects the base learner that selects that feature. Since it may not be possible to obtain a sufficiently sparse model, we suggest achieving feature selection after model fitting. It would be associated with the predicted accuracy of stacked SGL. According to the analysis of experimental results, this process enjoys superior performance in terms of FDR compared to other methods. In a word, our method can meet the three criteria of prediction, stability and selection based on the integration of feature group information.

The imperfect property that stacked-hoc usually has lower predictive performance than Stacked SGL would not weaken the feature selection ability of it. The predictive performance and feature selection need to trade off. When the predictive performance reaches the optimum, it could simply mean that the selected features have good discriminability, but it does not represent that these features meet the quantitative limit or have some deeper significance. For three public cancer datasets, our method presents better accuracy and identifies more powerful discriminatory and potential mutation genes. From the perspective of experiment, the two models could be performed separately. Certainly, researchers can decide whether to perform the second model based on their individual needs, after all, the second model would degrade predictive performance.

A simple extension of our method would be to incorporate sparse group penalty in linear model, correspondingly, applying linear regression model as a meta learner. The procedure is similar to the methods proposed by [Rauschenberger et al. \(2021\)](#). Besides, the normal tissue is generally much less than cancer tissue in most of the cancer datasets of TCGA. Another extension would be potentially to take combining bootstrapping with stacking into account for more powerful predictive.

Funding

This work was supported by the National Natural Science Foundation of China [21665016 and 21305062]; the Natural Science Foundation of Jiangxi Province [20192BAB204010]; and the Provincial-Level Project on the Teaching Reform of Colleges and Universities in Jiangxi Province [JX]G-18-1-25].

Data availability

The real data underlying this article are available from https://github.com/huanheaha/Stacked_SGL

Conflict of Interest: none declared.

References

- Belhechmi, S. et al. (2020) Accounting for grouped predictor variables or pathways in high-dimensional penalized cox regression models. *BMC Bioinformatics*, 21, 277.
- Boulesteix, A.L. et al. (2017) IPF-LASSO: integrative L1-penalized regression with penalty factors for prediction based on multi-omics data. *Comput. Math. Method Med.*, 2017, 1.

- Cerami, E. *et al.* (2012) The cBio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. *Cancer Discov.*, **2**, 401–404.
- Che, K. *et al.* (2020) Genetic variants detection based on weighted sparse group lasso. *Front. Genet.*, **11**, 155.
- Chen, H. *et al.* (2020) The sparse group lasso for high-dimensional integrative linear discriminant analysis with application to Alzheimer's disease prediction. *J. Stat. Comput. Simul.*, **90**, 3218–3231.
- Donoho, D.L. (2000) High-dimensional data analysis: the curses and blessings of dimensionality. *AMS Math. Challe. Lect.*, 1–32.
- Durif, G. *et al.* (2018) High dimensional classification with combined adaptive sparse PLS and logistic regression. *Bioinformatics*, **34**, 485–493.
- Forbes, S.A. *et al.* (2015) Cosmic: exploring the world's knowledge of somatic mutations in human cancer. *Nucleic Acids Res.*, **43**, D805–D811.
- Friedman, J. *et al.* (2010) A note on the group lasso and a sparse group lasso, arXiv:1001.0736v1.
- Gonzalez-Perez, A. *et al.* (2013) Intogen-mutations identifies cancer drivers across tumor types. *Nat. Methods*, **10**, 1081–1082.
- Hahn, P.R., and Carvalho, C.M. (2015) Decoupling shrinkage and selection in Bayesian linear models: a posterior summary perspective. *J. Am. Stat. Assoc.*, **110**, 435–448.
- Hastie, T. *et al.* (2009) The elements of statistical learning. In: Simo Puntanen (ed.) *Springer Series in Statistics*, 2nd edn. Springer, New York, NY.
- Kanehisa, M. *et al.* (2016) KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Res.*, **44**, D457–D462.
- Klau, S. *et al.* (2018) Priority-Lasso: a simple hierarchical approach to the prediction of clinical outcome using multi-omics data. *BMC Bioinformatics*, **19**, 322.
- Li, C.Y., and Li, H.Z. (2008) Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, **24**, 1175–1182.
- Meier, L. *et al.* (2008) The group lasso for logistic regression. *J. Royal Stat. Soc. Ser. B*, **70**, 53–71.
- Mendez-Civieta, A. *et al.* (2020) Adaptive sparse group lasso in quantile regression. *Adv. Data Anal. Classif.*, **15**, 547–573.
- Obozinski, G. *et al.* (2011) Union support recovery in high-dimensional multivariate regression. *Ann. Stat.*, **39**, 1–47.
- Qiu, Y. *et al.* (2021) Matrix factorization-based data fusion for the prediction of RNA-binding proteins and alternative splicing event associations during epithelial-mesenchymal transition. *Brief. Bioinform.*, **22**. <https://doi.org/10.1093/bib/bbab332>.
- Rauschenberger, A. *et al.* (2021) Predictive and interpretable models via the stacked elastic net. *Bioinformatics*, **37**, 2012–2016.
- Samal, S.S. *et al.* (2017) Linking metabolic network features to phenotypes using sparse group lasso. *Bioinformatics*, **33**, 3445–3453.
- Simon, N. *et al.* (2013) A Sparse-Group Lasso. *J. Comput. Graph. Stat.*, **22**, 231–245.
- Subramanian, A. *et al.* (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA*, **102**, 15545–15550.
- Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B*, **58**, 267–288.
- Ting, K.M., and Witten, I.H. (1999) Issues in stacked generalization. *J. Artif. Intell. Res.*, **10**, 271–289.
- van de Wiel, M.A. *et al.* (2019) Learning from a lot: empirical Bayes for high dimensional model-based prediction. *Scand. J. Stat.*, **46**, 2–25.
- Wang, H.H. *et al.* (2019) Precision Lasso: accounting for correlations and linear dependencies in high-dimensional genomic data. *Bioinformatics*, **35**, 1181–1187.
- Wolpert, D.H. (1992) Stacked generalization. *Neural Netw.*, **5**, 241–259.
- Yuan, M., and Lin, Y. (2006) Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B*, **68**, 49–67.
- Zehir, A. *et al.* (2017) Mutational landscape of metastatic cancer revealed from prospective clinical sequencing of 10,000 patients. *Nat. Med.*, **23**, 703–713.