Statistical Genetics and Bioinformatics Coursework 2

Kyriacos Xanthos
CID: 01389741

April 26, 2022

Question 1

Part 1

The dataset available is a gene expression dataset with rheumatoid arthritis patients. Our target variable is their C-reactive protein (CRP) levels and we have in total 1,500 genes as predictors. We would like to first find groups within the genes. For this, we are going to use a K-Means clustering approach.

We will apply the K-means algorithm to cluster the genes into groups. We will use standard scaling for the genes so that all features have zero mean and variance equal to 1. K-means is an unsupervised learning algorithm with the objective to group similar points together and find patterns in the data [1].

The `KMeans` function from `sklearn` performs the clustering algorithm and it uses a special K-means++ algorithm that initializes the clusters using a specific initialization method where all the centers of the clusters are first clustered together before we proceed with the k-means optimization. We do not specify any hyperparameters, except the numbers of of groups we want to find. Since we have a large amount of genes we do not know exactly how many clusters we want to cluster our genes into, we use a silhouette score measure to decide the best number of clusters.

The silhouette score measures how tightly grouped samples in the clusters are [2]. The equation to calculate the score is given by:

$$s_i = \frac{b_i - a_i}{\max\{b_i, a_i\}} \tag{1}$$

where a_i measures the average distance between sample x_i and all other points in the same cluster. b_i is a measure of how separated a cluster is from the next closest cluster. s_i is taking values in the range -1 to 1 where -1 means that sample x_i is assigned to the wrong cluster and 1 means x_i was assigned to its own cluster *and* it is away from other clusters.

We tried a grid of 29 values starting from k=2 going up to k=29. The silhouette score for each k is shown in figure 1.

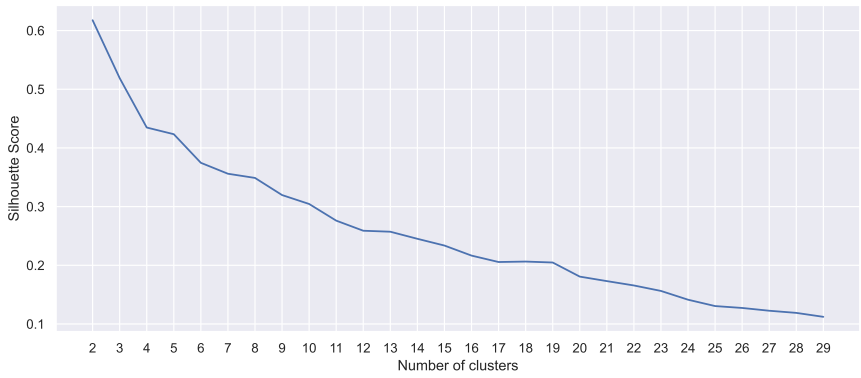


Figure 1: Grid from 2 to 29 clusters and their relative silhouette score according to equation 1.

Clearly 2 clusters produces the best silhouette score and we can visualize the silhouette score and distribution of the two clusters in figure 2.

Figure 2: Silhouette score for 2 clusters as found optimum using 1. The red line represents the average silhouette score.

We can create a 2-dimensional representation for this using Principal Component Analysis (PCA) which is another unsupervised algorithm that performs dimensionality reduction to just 2 linearly independent principal components. This can be seen in figure 3.

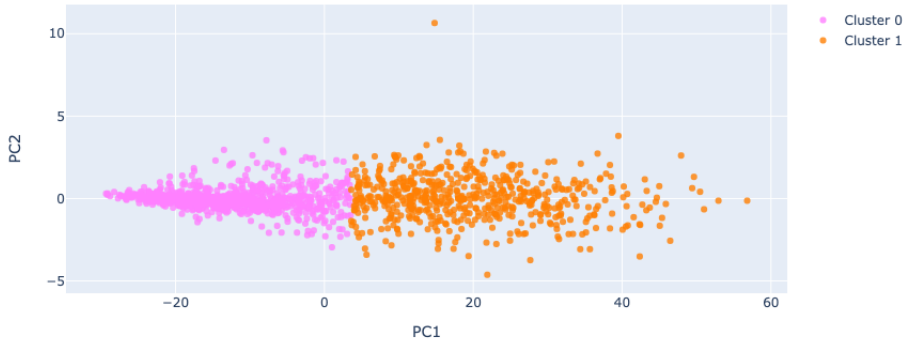


Figure 3: PCA plot for the two clusters found using the K-Means algorithm

We conclude that the number of groups amongst the genes are 2 and there are 916 genes in group 1 and 584 genes in group 2. We repeated this analysis using an agglomerate hierarchical clustering method that starts at the bottom and recursively joins clusters until we are left with one cluster. This method also showed that 2 clusters was the optimum decision.

Part 2

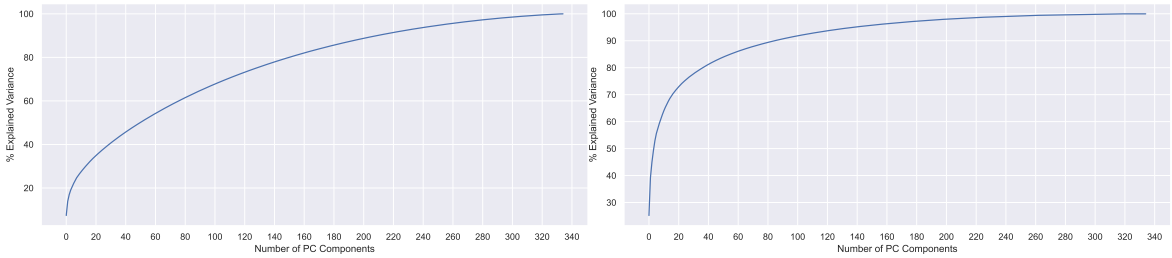
Now we consider the global null hypothesis: *The group is not associated with CRP*. We will use two approaches to to tackle this hypothesis, principal component regression and a self contained method.

PCA Regression

We choose to use PC Regression because multicollinearity leads to bad fitted models since the least squares estimates have very large variances. PCA finds an orthogonal projection of the data onto a lower dimensional linear space by maximizing the variance of the data and minimizing the average projection cost [3]. With PC Regression we can analyze multiple regression data. For the two gene groups we have, we transformed the scaled gene data using PCA. To choose the number of components we want to include in our model we looked at how the explained variance in the dataset changed as we included more principal components [1]. This can be seen in figures 5a and 5b.

We decided a threshold of 85% was good for explaining the variance in our dataset and moved on with a linear regression on the selected principal components. Group 1 used 177 principal components and Group 2 used 55. The ordinary least squares regression gave an

[1]Another method that could be implemented would be to use the MSE for each model with recursively adding PCs and choosing the lowest MSE.



(a) Group 1 (b) Group 2

Figure 4: Percentage of variance explained by the principal components of each group

R^2 score [2] of 0.58 for the first group and 0.21 for the second group. These are considered to be very low scores but the objective of this question is to understand the association between the phenotype and the genes, so looking at the F-test of the regression, group 1 gives a p-value of 10^{-16} and group 2 a p-value of 10^{-7}. These show that there is sufficient evidence to reject the null hypothesis that the phenotype is not associated with the genes for both groups.

Self-Contained Method

The Self-Contained Null Hypothesis is defined as: no genes in the group are differentially expressed. To implement this we run a linear regression of each gene with the response (CRP) and obtained a p-value for the significance of that gene. Since genes in a pathway are clearly correlated we needed to adjust our p-values using the Fisher's method. This method takes the individual p-values of the linear regression from each gene and it combines them using the equation:

$$FM = -2\sum_{i=1}^{M} \log(p_i) \sim \chi^2(2M) \tag{2}$$

where M is the total number of genes. Since both genes and SNPs are correlated between them we can obtain an approximate null distribution for the test statistic by permuting the data. We can permute the CRP and obtain the global test-statistic of interest.

This involves three steps:

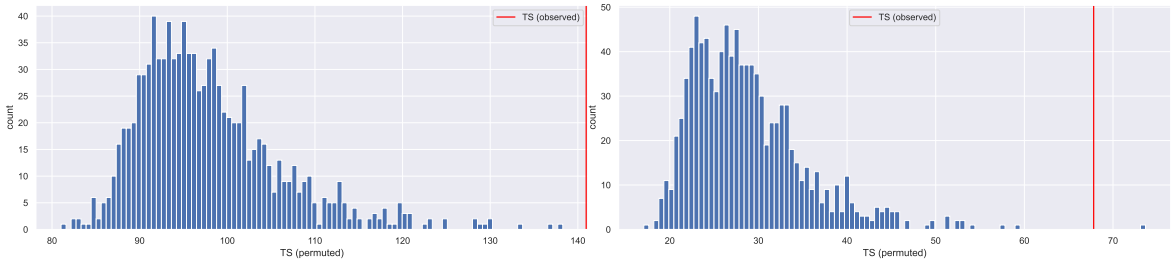
1. Compute Fisher's statistic ($TS_{observed}$) [2] on the original (non-permuted data)
2. Repeat for B = 1,000 steps:
   (a) Permute response labels
   (b) Calculate p-values using Linear regression
   (c) Compute Fisher statistic on permuted data ($TS_{permuted}$)
3. Calculate global p-value using:

$$\frac{\sum_{i=1}^{B} I\left(TS_{\text{permuted},i} \geq TS_{\text{observed}}\right) + 1}{B+1}$$

The global p-value calculated for our dataset was 0.001 for the first group (group with 916 genes) and 0.002 for the second group (group with 584 genes). To visualize how these p-values were computed we include histograms to see the distributions of TS_{permuted} in comparison to TS_{observed} in figure 5.

The histograms show all of the permuted statistics lie before the observed one for the first group, which gives us very high confidence that the results of the linear regression with

[2] R^2 measures the amount of explained variability in the model. It usually takes values between 0 and 1 with 1 corresponding to all of the variability in the dataset is explained by the model.



(a) Group 1 (b) Group 2

Figure 5: Distribution of TS_{permuted} values in comparison to TS_{observed}

the true responses were much better than the results of a linear regression with random permutations. This actually let to the simple global p-value $\frac{1}{1{,}000+1} = 0.001$ which is significant enough to reject the null hypothesis. The second group seems to only have one instance of a permuted linear regression to be better than a randomly permuted one and therefore again result in a very small p-value significant enough to reject the null hypothesis.

This shows that both PC Regression and the self-contained method agree on their conclusions. This was expected since the two approaches try in different ways to reduce the multi-collinearity and they both achieve to create logical models.

Question 2

Part 1

We now create a simulation study for gene expression data that investigates the power of Lasso regression for identifying truly associate variables.

To create the simulated data we follow [4] where the expression data are simulated using a multivariate normal distribution:

$$\mathbf{X_i}_{(1\times m)} \sim MVN(0, \mathbf{\Sigma}) \quad i = 1, \ldots, n \tag{3}$$

where n is the number of samples, m is the number of genes, and $\mathbf{\Sigma}$ is the covariance matrix which will be used to explore the correlation structure of our gene expressions later on.

To simulate the response variable which will be a quantitative phenotype we will use:

$$Y_i \sim N(\mu, \sigma^2) \text{ with } \mu = \boldsymbol{\beta}\mathbf{X}_i \tag{4}$$

where σ^2 is the standard deviation of the Normal distribution followed by the errors of the equation. We will set this equal to one which means the equation we use to simulate $\boldsymbol{Y}$ is:

$$Y_i = 1 + X_i\beta^T + \epsilon_i \quad , \quad \epsilon \sim N(0, 1) \tag{5}$$

β in the equation above indicates the effect size as noted by Fridley [4], and for our simulations we will randomly select one of two values corresponding to the m features: 0 with probability 0.9 indicating no effect size and 2 with probability 0.1. This intuitively quantifies the amount of association between the phenotype and the genes, with larger effect size giving better model fits.

We will then apply Lasso Regression on our simulated data. Lasso Regression is an extension to linear regression. We add an l_1 penalty on the coefficients responsible for introducing regularisation to our model, and use ordinary least square method with the objective is to solve the following problem:

$$\underset{\boldsymbol{\beta}}{\text{argmin}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \tag{6}$$

where $\boldsymbol{y}$ is the $(n \times 1)$ response we generated above, $\boldsymbol{X}$ is the $(n \times m)$ covariates matrix for all our genes, $\boldsymbol{\beta}$ is the $(m \times 1)$ coefficients vector where each coefficient corresponds to

one gene, and λ is the hyperparameter that quantifies the amount of regularisation in our model.

We will apply this model using the `Lasso` function from `sklearn` where the only parameter we specify is the regularisation λ. In order to evaluate the performance of our model we will use the root mean squared error metric (rmse) defined as:

$$\text{rmse} = \left(\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \right)^{1/2} \tag{7}$$

where $\hat{\boldsymbol{y}}$ are the predicted target values from our model. To make sure we have found the optimum λ, we will use K-fold cross validation. This means that we split the covariates matrix to randomly chosen K-folds, we train the model on the first $K-1$ folds and validate the model on the K^{th} fold using rmse. By averaging across multiple folds we decrease the variability in the model fit and the validation error that comes from randomly splitting the dataset. The grid we choose to search over is $\lambda \in [0.001, 0.5]$ with 50 points. The range is relatively large and this is because we have a varying number of features in our analysis, and we noticed that larger amount of features tend to choose a significantly smaller optimum λ than a lower dimensional feature matrix. To implement this, we use the `GridSearchCV` function from `sklearn`.

In order to try and reduce overfitting in our models, each time we run a model we split it into random training and testing sets with compositions 70% and 30%. We run the cross validation procedure on the train set to choose the optimum model, and then we evaluate the performance of the model on the test set. In each run, we return the rmse and R^2 of the model aswell as the False Positive Rate (FPR) defined as:

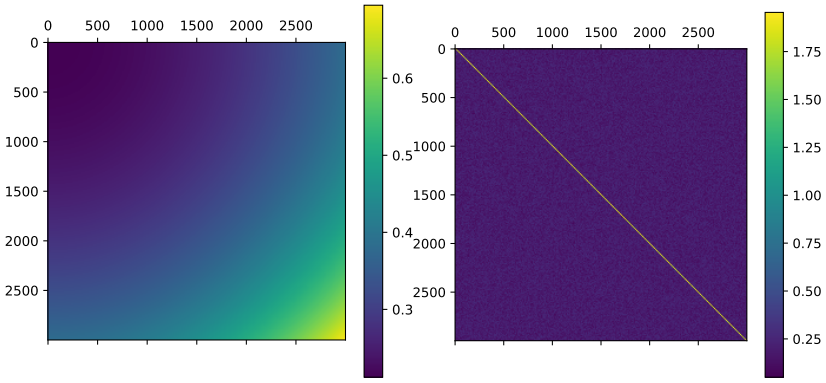
$$\text{FPR} = \frac{FP}{FP + TN} \tag{8}$$

where FP are the instances that our model finds assigns a non-zero coefficient to a gene that was constructed to not have any correlation with the phenotype, and TN are the instances that the model correctly predicts that the coefficient of a gene should be equal to zero (no association).

Part 2

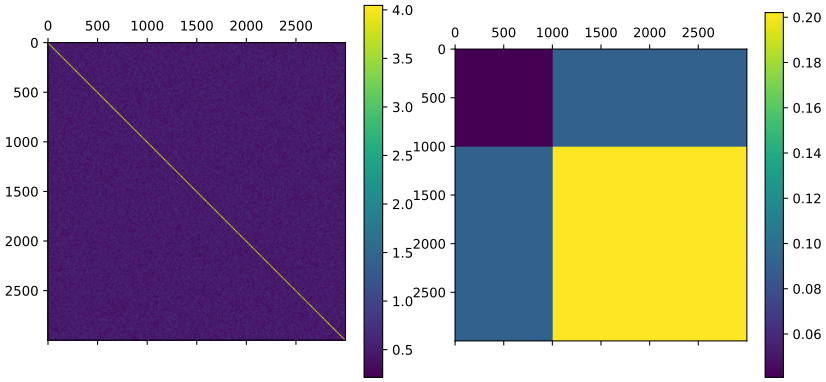
We run the simulation study with $n = 500$ and $m = 3000$. We need to explore a varying correlation structure and we do that in two different methods. The first method is to create the covariance matrix $\boldsymbol{\Sigma}$ with ones on the diagonal and $\rho = [0.1, 0.9]$ with steps of 0.1 which explores different sizes of correlation between our genes. $\rho = 0.1$ suggests a very small correlation between our genes and $\rho = 0.9$ suggests a very large correlation. This is similar to what was done in Fridley's study [4] where they equate ρ to either 0, 0.1 or 0.3. All off-diagonal terms having the same value mean that we have the same correlation across all genes the pathway.

As an alternative method, we try different structures of our covariance matrix. One of them is choosing randomly the amount of correlation between genes between medium ($\rho = 0.3$) and high ($\rho = 0.7$) correlation (**RMH** seen in 6c), and low ($\rho = 0.1$) and medium ($\rho = 0.3$) correlation (**RLM** seen in 6b). This is similar to what was done in [5], where they explored high correlations between variables and also mixtures of medium and high correlations. In addition, we checked two different correlation structures were in the first case the correlation increased (from 0 to 0.5) as the position of thee gene is increases sequentially (**INC** seen in 6a) and another that the first 1/3 of the covariance matrix was set to 0 correlation and the last two thirds to small correlation of 0.2 (**UEQ** seen in 6d). The final two correlation structures could mean that the correlation between genes heavily relies on physical distance between the genes and that there are could be two very different families of genes leading to very different correlation structures between them, similar to what we explored in question 1. We needed to ensure that these covariance matrices were positive-definite to be able to sample from `scipy`'s MVN distribution, so we used a method proposed to find the closest positive definite matrix to the structure we wanted to explore [6].



(a) **INC**: Increasing covariance from ρ =0 to ρ =0.5 sequentially

(b) **RLM**: Randomly choosing between ρ =0.1 and ρ =0.3 with equal probability



(c) **RMH**: Randomly choosing between ρ =0.3 and ρ =0.7 with equal probability

(d) **UEQ**: First third of the matrix set to $\rho = 0$ and the rest to $\rho = 0.2$

Figure 6: Graphical representations of covariance matrix structures used for 3,000 genes. The bold acronyms are the names we use for each structure.

For each covariance matrix we apply the LASSO method, where we perform cross-validation each time to choose the optimum λ using the train split of the dataset, and evaluate the performance of each model using the test split of the dataset. Running the models with $K = 5$ cross validation, the optimum $\lambda \approx 0.01$. The optimum λ was the same for all correlation structures.

The results for increasing levels of correlations within the genes are shown in table 1. It is clear that as the correlation between the genes increases, the LASSO model performs better. This is easily visualised in figure 7, where the rate of increase of test R^2 is larger for small amounts of correlation than towards the end. We achieve an R^2 score of more than 0.9 for $\rho = 0.6$ which shows that the model built is extremely good at predicting the phenotype. This is because the large correlation between genes allows the model to understand the pattern of variation between the train set and it expects the same variation in the test set, therefore giving a very high score.

However, the trend is not the same when we look at the FPR in figure 7. We ideally want the FPR to be as low as possible, and even though all correlation structures seem to give a low FPR, the lowest FPR levels are given by the highest and lowest correlation we are checking. This shows that the LASSO model correctly predicts a non-zero coefficient, when there is a true association of the gene and the phenotype, far more times when the correlation structure is at the two extreme values.

Moving on, we summarise the remaining four correlation structures results in table 2. We can see that the **INC** and **UEQ** structures produce the highest R^2 of almost equal to 1. However, **INC** seems to always assign a non-zero value to the coefficients which brings the FP to extremely high levels giving a poor FPR for it to be considered a reliable model. **UEQ** gives the lowest FPR with just 3 TP values, which is again suspicious since only 3

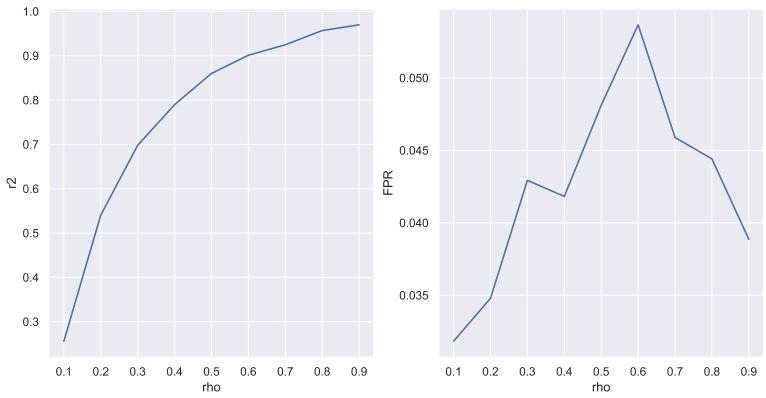


Figure 7: R^2 (left) and FPR (right) scores of LASSO model as correlation between genes increases

rho	rmse	R^2	TP	FN	FP	TN	FPR
0.1	0.79	0.25	36	262	86	2616	0.032
0.2	0.55	0.54	45	253	94	2608	0.035
0.3	0.48	0.70	25	273	116	2586	0.043
0.4	0.30	0.79	28	270	113	2589	0.042
0.5	0.33	0.86	27	271	130	2572	0.048
0.6	0.29	0.90	24	274	145	2557	0.054
0.7	0.25	0.92	18	280	124	2578	0.046
0.8	0.19	0.96	23	275	120	2582	0.044
0.9	0.16	0.97	17	281	105	2597	0.039

Table 1: Summary of results when varying the amount of correlation between the genes. TP is the amount of instances when the correct coefficient was correctly classified as non-zero in Lasso. FN when the coefficient was not picked up by lasso but there was a true association of the gene with the phenotype. FP when it was given a non-zero coefficient by lasso when there was truly no association. TN when there was no association in the data and it was correctly predicted. FPR is defined in 8.

genes were found to be associated, whereas a total of 295 were actually associated. Lastly, **RLM** and **RMH** seem to give similar results, with the more correlated one, **RLM** giving overall better metrics.

We conclude in this section that the LASSO model favours larger correlation between the genes and less variation within the correlation structure for better predictive results.

Part 3

In this part we keep the correlation structure the same: a covariance matrix of ones on the diagonals and $\rho = 0.3$ on the off-diagonals, signaling the same, and small correlation between all genes. We will vary the number of samples in the simulation and check the values $n = [100, 300, 500, 700, 1000]$ and $m = [100, 300, 1000, 3000, 5000]$.

For each value of n, we check all values of m. The relevant table is included in appendix A and we provide graphical results of the runs in figure 8. We can observe that as the sample size increase R^2 increases and the rmse decreases. This is because the model has

corr	rmse	R^2	TP	FN	FP	TN	FPR
INC	0.008	0.99	295	3	2664	38	0.985
RLM	0.691	0.35	51	247	108	2594	0.039
RMH	0.672	0.41	47	251	144	2558	0.053
UEQ	0.012	0.99	3	295	9	2693	0.003

Table 2: Summary of results for the 4 different correlation structures explored. TP, FN, FP, TN, and FPR are as defined in table 1.

more samples to train on and therefore has access to more data to adjust the weights of the parameters. However, we can see that after 600 samples the improvements in the models are not very clear. It seems that irrespective of number of genes in the dataset, the accuracy of the models does not become significantly better. Smaller number of genes gives the most accurate models overall, which shows that lower dimensional models are favoured by Lasso.

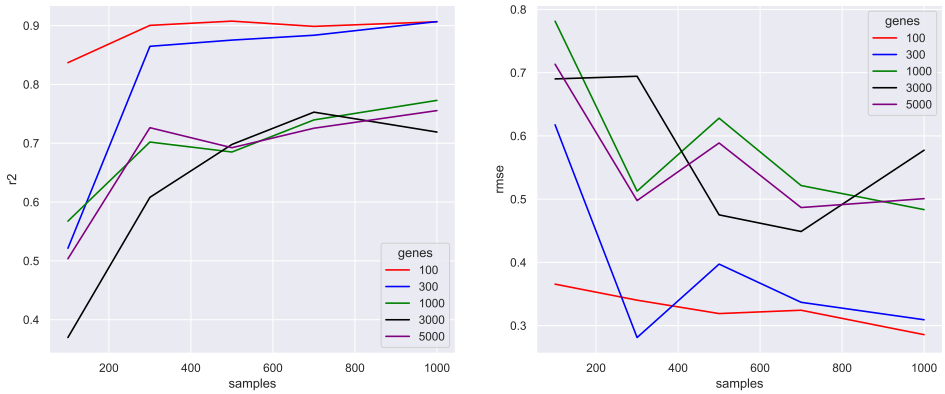


Figure 8: R^2 (left) and rmse (right) scores of LASSO model as number of genes and samples in the simulation study change

Looking at figure 9, we observe that as the number of genes increases, the FPR increases aswell. This relationship stems from the idea that the more genes there are in the dataset, the more are truly correlated with the response (in the way that our simulation was set up), and the lower the probability that the model will assign them a non-zero coefficient. The sample size does not seem to have a large impact to the FPR, except that the more the samples the greater the chance of misclassifying a coefficient and therefore producing a large FPR.

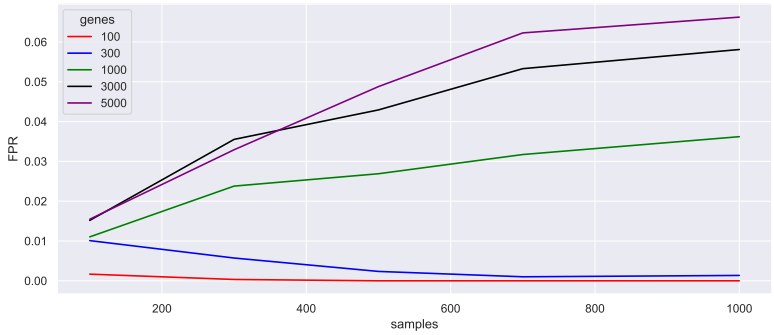


Figure 9: FPR scores of LASSO model as number of genes and samples in the simulation study change

Part 4

Our short analysis has touched on some important issues with using models like LASSO to identify truly associated variables with a phenotype. We explored the correlation structure of the genes and how that affects the accuracy of our models. We concluded that more correlation between genes, usually means more accurate models in terms of rmse and R^2 but not always with higher classification accuracy, finding truly associated genes. This is in agreement with literature, where Waldmann *et al* [5] found the same results when changing the amount of correlation between their genes. They were using mean squared error for validation and the lowest mse was given by the model with most correlation.

We went a step further and explored different patterns of correlation within our genes and found some interesting results, were models performed better when the correlation between them was uneven, although the FPR suffered in exchange.

Exploring performance of models for different number of samples was also very insightful, concluding that more genes in the data does not seem to create more accurate models, both

in rmse scores and FPR. Increasing the number of samples also showed a steady increase in performance but after 600 samples the results did not seem to be very affected. This is also in agreement with literature where Yang *et al* [7] have showed that their method for selecting the lasso tuning hyperparameter λ performs better for the simulation studies with more samples and not a large variation with genes.

Overall, there are a lot more methods we could explore. We could explore how changing the β parameter in our simulation could affect the results. We could run the models multiple times and average the values of the rmse, R^2, and FPR. We did not do this in this analysis because the computation times would be too long, but it is a very easy extension to the existing code.

Question 3

Part 1

Wang *et al.* [8] propose that pathway-based approaches complement the most-significant SNPs approach and they demonstrate that their method finds new insights when interpreting GWA data on diseases. They calculate for each SNP in the gene a test statistic value r_i. They propose a statistic like χ^2 but other statistics could also be applied. This tests the association of the single SNP with the phenotype (response) and the statistic takes larger values when there is a strong association between them, and low values otherwise. Then for each gene, they select the highest SNP test statistic and assign it to be the test statistic of the gene. This is identical to taking the smallest p-value when conducting a statistical test.

Part 2

Alternative gene statistics would be Fisher's product method defined in equation 2 or Stouffer's method [1] defined as

$$FM = -2\sum_{i=1}^{M} \log(p_i) \quad \text{where} \quad p_i \sim U(0,1) \forall i \tag{9}$$

where M is the total number of SNPs in the gene in this case. Both methods combine p-values from individual tests. The individual tests in this setting would be if a certain SNP is associated with the response. By combining all of these individual tests we can then assign a global p-value to each statistic. All these methods are different from the proposed test statistic because they all rely on p-values from the hypothesis tested instead of the raw test statistic values. The tests to find those p-values can be the same as the proposed method, ie. a χ^2 test.

Both alternative methods do not require drastic changes in the approach that the authors followed for adjusting for gene size, since neither Fisher's method nor Stouffer's method account for gene size. This means that the two step permutation procedure (explained in part 3) would not change at all. As an alternative, we could use the Weighted Fisher's method proposed by Yoon *et al* [9], where they propose a gamma distribution to assign non-integer weights (instead of integer weights that are only allowed by the χ^2 distribution) to each p-value that they are actually proportional to the exact sample sizes. The authors explain that this gives both higher power to the test and high robustness. Since this test takes into consideration the size of each gene, Wang *et al.* could amend the second step of their permutation procedure.

Part 3

The permutation procedure the authors present is a two-step correction procedure. In the first step they permute disease labels for all samples present in the dataset. They ensure that the same number of individuals are present in each phenotype group. In each permutation they calculate the enrichment score defined as

$$\mathrm{ES}(S) = \max_{1 \leqslant j \leqslant N} \left\{ \sum_{G_j \in S, j^* \leqslant j} \frac{\left|r_{(*)}\right|^p}{N_R} - \sum_{G_{j^*} \notin S, j^* \leqslant j} \frac{1}{N - N_H} \right\} \tag{10}$$

In the above definition N is the number of genes represented by SNPs in the GWA study and their statistic values are represented by $r_{(1)}, \ldots, r_{(N)}$ which are sorted in descending order. S is the gene set containing N_H genes and $N_R = \sum_{G_{j^* \in S}} \left|r_{(j^*)}\right|^p$ with p giving higher weight to genes with extreme statistic values. The first step calculates the enrichment score for all gene sets aswell as all permutations.

In the second step they calculate a normalized enrchment score (NES) defined as:

$$\frac{\mathrm{ES}(S) - \mathrm{mean}[\mathrm{ES}(S, \pi)]}{\mathrm{SD}[\mathrm{ES}(S, \pi)]} \tag{11}$$

where $E(S, \pi)$ is the enrichment score for all permutations π.

The importance of this procedure is that it ensures that genes with larger number of SNPs that would consequently have a maximum statistic larger than genes with smaller number of genes are adjusted for their gene size. In this way, gene sets of different sizes are "directly comparable with each other" [8].

Part 4

One of the disadvantages of the above approach is that permuting and recalculating statistics for potentially millions of SNPs and thousands of samples is very computationally expensive. For this reason, the authors propose another approach that instead of shuffling the phenotype labels, the alternative method shuffles the test statistic values for all genes, and then calculates the normalized enrichment score as before. The advantage of this method is that we do not need to use raw genotype data and we also do not need to do phenotype shuffling which is very computationally expensive.

One intricacy of this alternative approach is that the preranked module of GSEA [10] needs the p-value of each gene and that means a decision should be made on how we choose the single p-value. The authors propose two approaches for this, one is choosing the most significant p-value from all SNPs around a particular gene. Another approach is the Simes method [11] that is noted by the authors that is it an over conservative approach leading to loss of power.

One disadvantage of the alternative approach is that using a preranked list of genes is biased because the order will have a discriminatory effect for genes towards the end of the list. This was actually shown in the paper by applying the Simes method on three datasets where a huge loss of power was observed because the enrichment signals disappeared. Therefore whenever possible (computationally) the first approach would be better to use because there is no bias towards how the genes are ranked, but if it is computationally infeasible, the second approach should be used, being mindful of the potential loss of power.

A Appendix

samples	genes	rmse	R^2	TP	FN	FP	TN	FPR
100	100	0.36	0.83	9	0	5	2986	0.001
100	300	0.61	0.52	12	17	30	2941	0.010
100	1000	0.78	0.56	9	91	32	2868	0.011
100	3000	0.69	0.36	7	291	41	2661	0.015
100	5000	0.71	0.50	3	476	39	2482	0.015
300	100	0.34	0.90	9	0	1	2990	0.000
300	300	0.28	0.86	29	0	17	2954	0.005
300	1000	0.51	0.70	25	75	69	2831	0.023
300	3000	0.69	0.60	17	281	96	2606	0.035
300	5000	0.49	0.72	15	464	83	2438	0.032
500	100	0.31	0.90	9	0	0	2991	0.000
500	300	0.39	0.87	29	0	7	2964	0.002
500	1000	0.62	0.68	35	65	78	2822	0.026
500	3000	0.47	0.69	25	273	116	2586	0.042
500	5000	0.58	0.69	18	461	123	2398	0.048
700	100	0.32	0.89	9	0	0	2991	0.000
700	300	0.33	0.88	29	0	3	2968	0.001
700	1000	0.52	0.73	56	44	92	2808	0.031
700	3000	0.44	0.75	36	262	144	2558	0.053
700	5000	0.48	0.72	28	451	157	2364	0.062
1000	100	0.28	0.90	9	0	0	2991	0.000
1000	300	0.30	0.90	29	0	4	2967	0.001
1000	1000	0.48	0.77	59	41	105	2795	0.036
1000	3000	0.57	0.71	43	255	157	2545	0.058
1000	5000	0.50	0.75	37	442	167	2354	0.066

Table 3: Summary of results for different sample and gene sizes. TP, FN, FP, TN, and FPR are as defined in table 1.

References

[1] Marina Evangelou. Statistical Genetics and Bioinformatics Lecture Notes. *Imperial College London*, March 2022.

[2] Sebastian Raschka. *Python Machine Learning*. Packt Publishing - ebooks Account, 2015. ISBN 1783555130. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1783555130.

[3] Dr Sarah Filippi. Machine learning lecture notes. *Imperial College London*, pages 30–40, October 2021.

[4] Brooke L Fridley, Gregory D Jenkins, and Joanna M Biernacka. Self-contained gene-set analysis of expression data: an evaluation of existing and novel methods. *PloS one*, 5 (9):e12693, 09 2010. doi: 10.1371/journal.pone.0012693. URL https://pubmed.ncbi.nlm.nih.gov/20862301.

[5] Patrik Waldmann, Gábor Mészáros, Birgit Gredler, Christian Fürst, and Johann Sölkner. Evaluation of the lasso and the elastic net in genome-wide association studies. *Frontiers in Genetics*, 4, 2013. ISSN 1664-8021. doi: 10.3389/fgene.2013.00270. URL https://www.frontiersin.org/article/10.3389/fgene.2013.00270.

[6] Nicholas J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103:103–118, 1988. ISSN 0024-3795. doi:

https://doi.org/10.1016/0024-3795(88)90223-6. URL https://www.sciencedirect.com/science/article/pii/0024379588902236.

[7] Songshan Yang, Jiawei Wen, Scott T Eckert, Yaqun Wang, Dajiang J Liu, Rongling Wu, Runze Li, and Xiang Zhan. Prioritizing genetic variants in GWAS with lasso using permutation-assisted tuning. *Bioinformatics*, 36(12):3811–3817, 04 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa229. URL https://doi.org/10.1093/bioinformatics/btaa229.

[8] Kai Wang, Mingyao Li, and Maja Bucan. Pathway-based approaches for analysis of genomewide association studies. *The American Journal of Human Genetics*, 81(6): 1278–1283, 2007. ISSN 0002-9297. doi: https://doi.org/10.1086/522374. URL https://www.sciencedirect.com/science/article/pii/S0002929707637756.

[9] Sora Yoon, Bukyung Baik, Taesung Park, and Dougu Nam. Powerful p-value combination methods to detect incomplete association. *Scientific Reports*, 11(1):6980, 2021. doi: 10.1038/s41598-021-86465-y. URL https://doi.org/10.1038/s41598-021-86465-y.

[10] UC San Diego. Gene set enrichment analysis. https://www.gsea-msigdb.org/gsea/doc/GSEAUserGuideFrame.html?_GSEAPreranked_Page, 2022. [Online; accessed 19-April-2022].

[11] R. J. Simes. An improved bonferroni procedure for multiple tests of significance. *Biometrika*, 73(3):751–754, 2022/04/19/ 1986. ISSN 00063444. doi: 10.2307/2336545. URL https://doi.org/10.2307/2336545. Full publication date: Dec., 1986.