The Linkage Disequilibrium LASSO: An R Package for Case-Control Genetic Association Studies

Samuel G. Younkin, Joseph Nadeau, J. Sunil Rao

February 14, 2011

1 Introduction

Genetic epidemiologists have long sought the identification of genetic markers that are associated with disease. Models predict that there should be many such risk variants, and predicted effect size distributions show that the number of risk variants increases quickly as the effect size decreases. Past genetic association studies have identified many of the large effect size variants, yet the remaining variants with low effect size have not been identified. The cost of increasing sample sizes and large-scale collaborations has caused many to abandon the search, and so there is need for methods that can identify low-effect variants without these costs. The *ldlasso* package implements a method that uses the well-known LASSO methodology of Tibshirani et al. to identify the presence of weak genetic associations [1].

2 Genetic Association

A common measure of genetic association to disease is the increase in relative risk between subjects with the genetic variant and subjects without the genetic variant, approximated in a case-control study by the odds ratio. In the case of genetic association we commonly have data for a dense set of di-allelic markers, known as SNPs. If f_0 is the frequency of the minor allele in the controls, and f_1 the frequency in the cases, then the odds ratio is $\text{OR} = \frac{f_1/(1-f_1)}{f_0/(1-f_0)}$. The odds ratio can also be computed in a logistic regression setting with genotype values coded in an additive fashion and a test for a nonzero log odds ratio is often used to identify SNPs that are associated with disease. Under an additive model

the p-value computed is equivalent to the p-value that results in a χ^2 test for independence of alleles and disease status. Methods for identification of disease associated SNPs have been implemented in the R package *GenABEL* [2]. The *GenABEL* package was designed for use in large scale genetic associations studies in which hundreds of thousands of SNPs are interrogated across the entire genome. These studies are known as **g**enome-**w**ide **a**ssociation **s**tudies, or GWAS. *GenABEL* is designed to accommodate large data sets and uses binary compression to reduce the size of the resulting R objects. The package *ldlasso* accepts objects of class 'gwaa.data' from the *GenABEL* package. Methods such as the χ^2 allelic test of association consider SNPs in isolation. In the search for weak genetic association signals, a one-at-a-time approach such as this is doomed for failure. We improve on this approach by considering an entire set of SNPs simultaneously and by imposing restrictions on the solutions that incorporate two basic characteristics of genetic association, namely linkage disequilibrium and haplotype block structure.

2.1 Linkage Disequilibrium & Haplotype Blocks

Before the transmission of genetic material from parent to offspring, homologous chromosomes in each parent combine to form a distinct chromosome that is then passed on to the offspring. The mechanics of the recombination are complicated, but it is well established that sites on a chromosome that are physically close together are likely to both be included on the chromosome that is transmitted to the offspring. This concept is known as **l**inkage **d**isequilibrium, or LD. Due to LD, when SNPs are densely spaced we observe

are large amount of correlation among the SNPs. The degree to which LD occurs is not constant throughout the genome. There are known to be regions in which there is a great deal of correlation due to LD, punctuated by regions in which there is very little LD. These boundaries create what are known as haplotype blocks, which are large segments of the genome that are inherited together. The *ldlasso* package was written to use the Markov chain Monte Carlo methods incorporated in the R package *MATILDE* to identify the boundaries of haplotype blocks, and these boundaries are used to help identify weak genetic associations [3]. It is not necessary to use *MATILDE* to generate the haplotype block boundaries, and *ldlasso* will run without any boundaries, or with a user-defined block boundary vector.

3 The LD LASSO Algorithm

Because we have densely spaced SNPs, which are highly correlated, we expect there to be signals present at sites other than the causal variant. It is also likely that regions of the genome that contain genetic variants that increase susceptibility to disease by a small amount will contain multiple variants that do so independently. We also expect that the large number of SNPs in even a moderately sized candidate region will result in signal which is nonetheless sparse. The genetic association signal we expect to observe will have breadth, due to both correlation and multiple disease variants. However, due to haplotype block boundaries we expect there to be an abrupt absence of signal as well as low frequency SNPs that do not exhibit the signal. To begin we

consider a non-parametric regression model with y_j, the normalized log odds ratio for SNP j and β is the mean vector, which represents the association signal, which has blocks of nonzero signal, with point-wise gaps.

$$y_j = \beta_j + \epsilon_j \tag{1}$$

The density of our SNP data is such that the error terms, ϵ_i and ϵ_j are dependent. The LASSO is a penalized least squares estimator in which an l_1 penalty creates estimates that are identical to zero, and is ideal as a subset selector for sparse solutions [1]. LASSO regression may be expressed as the convex optimization problem with LASSO parameter s_1.

$$\hat{\beta}_{\text{LASSO}} = \underset{\beta \in R^p}{\operatorname{argmin}} \left\{ \sum_{j=1}^{p} (y_j - \beta_j)^2 \right\}, \quad \sum_{j=1}^{p} |\beta_j| \leq s_1 \tag{2}$$

A variant of the LASSO, the fused LASSO adds an additional constraint to the optimization problem, $\sum |\beta_j - \beta_{j+1}| < s_2$, which encourages a flat and block-like solution, ideal when the true signal is expected to be constant across a series of consecutive ordered predictor variables [4]. In the case of genetic association we do not expect our signal to be flat, but we do expect it to exhibit a block structure. In place of the constraint from the fused LASSO, we choose a set of constraints defined for a collection of groups of SNPs, or blocks. For the LD LASSO we define the blocks in terms of a haplotype block map, $\mathcal{B}$, although any method of block definition is acceptable here.

$$\mathcal{B} = \{(i, j);\ \text{SNP } i \text{ and } j \text{ are in the same block } \} \tag{3}$$

Since we are concerned with SNP selection and not effect size estimation, we regress only as a means of subset selection and not parameter estimation. Furthermore, the direction of the effect is defined arbitrarily in terms of the minor allele, and so we allow for a change in the sign of β. We do so by defining the LD LASSO constraint in terms of $|\beta|$.

$$\hat{\beta}_{\text{LD}} = \underset{\beta \in R^p}{\operatorname{argmin}} \left\{ \sum (y_j - \beta_j)^2 \right\}, \quad \sum |\beta_j| \leq s_1, \quad ||\beta_i| - |\beta_j|| \leq \varphi(r_{ij}^2) \quad (4)$$

The constraint function, φ, is defined below in terms of the LD LASSO parameter, s_2 and a minimum value of correlation, r_{cut}^2.

$$\varphi(r_{ij}^2) = \begin{cases} -s_2 \log r_{ij}^2 + \delta & \text{, if } r_{ij}^2 > r_{\text{cut}}^2 \text{ and } (i,j) \in \mathcal{B} \\ \infty & \text{, otherwise} \end{cases} \quad (5)$$

with $\delta = 10^{-10}$ to allow for computer precision when $r_{ij}^2 = 1$. To find a solution using the parameter selection algorithm described below use the function *ld_lasso_method*.

```
> ld_lasso_method(block.obj, block.cood)
```

3.1 Parameter Selection

To find the LD LASSO estimate we solve the corresponding convex optimization problem using the R package *quadprog* which applies the method of Goldfarb and Idnani [5]. The optimization problem is defined with linear constraints and the size of the constraint matrix must be considered when choosing the

parameters. If the block map has size b, then the constraint matrix has dimension $3p \times (3p + 2b + 1)$, where p is the number of SNPs. If p or b is large the algorithm may be prohibitively slow and so we place a limit of the size of the constraint matrix by finding the value of r^2_{cut} necessary to reduce the size of $\mathcal{B}$. We choose r^2_{cut} to be the minimum value that corresponds to a constraint matrix with no more than 5,000 columns.

3.1.1 Haplotype Block Map

To estimate the positions of block boundaries we use the Markov Chain Monte Carlo method implemented in the R package *MATILDE* [3]. This package is not available at the CRAN repository. To find this package see the link given in the help page for *ldlasso*. Here we may choose a cutoff value for the probability necessary to deem the interval between SNPs a haplotype block boundary. We prefer to have large blocks, for small blocks will result in SNPs that are unconstrained by the LD LASSO, as constraints are finite only when SNP-pairs occur within the same block. Thus we choose a high value for this cutoff of 95% and implement the function *find.bounds* given in the help page for *ldlasso*. Note that this function is not inherent to the *ldlasso* package and is dependent on the package *MATILDE*.

```
> block.cood <- find.bounds(block.obj, prob.threshold = 0.95)
```

It may take considerable time for *find.bounds* to complete. In it all pairwise LD estimates must be computed, and two MCMC runs completed.

3.1.2 LASSO Parameter

The LASSO parameter, s_1, controls the sparsity of the solution. For small values of s_1 many of the SNPs will have estimates equal to zero, and thus will not be in the model. The number of SNPs allowed in the model is an increasing function of s_1. Under the null hypothesis we expect that no SNPs are in the model; however, we expect some number of false positives. Using a permuted phenotype vector we find the value of s_1 that corresponds to a solution with 10% of the total number of SNPs included. Thus, we have a value for s_1 that corresponds roughly to a false positive rate of 10%. To find s_1 we perform a simple bisection algorithm that approximates the root of the function $g(s_1) = f(s_1) - \frac{1}{10}p$, where p is the total number of SNPs, and f maps s_1 to the number of SNPs included with the permuted phenotype vector. Note that $f(s_1)$ is an integer valued function that may skip a value and increase by two if two SNPs are in perfect correlation. There is not always a root, and so we use an approximate solution.

3.1.3 LD LASSO Parameter

The LD LASSO parameter, s_2, controls the amount of fusion present in the solution. The package *ldlasso* requires the user to provide a vector of s_2 values to be considered. The *ldlasso* package then returns three solutions, the fused, unfused, and intermediate solutions. The fused solution corresponds to the smallest value of s_2, the unfused solution corresponds to the largest value of s_2, and the intermediate solution that corresponds to the value of s_2 that provides the optimal solution, in which an estimate of prediction error, c_p is minimized.

The formulation of c_p is similar to that of Efron et al., and estimates for the covariance term are estimated using bootstrap sampling [6].

$$\hat{c}_p = ||\mathbf{y} - \hat{\beta}||^2 - p^\star + 2\sum_{j=1}^{p} \hat{\text{cov}}\left(\hat{\beta}_j, y_j\right) \tag{6}$$

In the following section we demonstrate the use of *ldlasso* by examining the sample data provided with the package.

4 Example

First we load *ldlasso* along with the example data.

```
> library("ldlasso")
> data("ldlasso_example")
```

The sample data consist of 'block.obj', a GenABEL object with class gwaa.data, 'block.cood,' a vector of block boundaries, and 'ldlasso.obj,' an ldlasso object. The ldlasso object was computed using the block map defined by 'block.cood' along with an algorithm used for parameter selection implemented in the function *ld_lasso_method*. This method identifies three solutions based on the choice of s_2; the c_p optimal solution, the fused solution, and the unfused solution as described above. Here we use $B = 100$ bootstrap samples for each of the 75 s_2 values considered. With parameters $B = 100$ and s2.vec.length = 75 the following code may take up to an hour to complete. For the sake of example use the ldlasso object provided with the example data, 'ldlasso.obj,' or decrease the values of the two parameters.

```
> ldlasso.obj <- ld_lasso_method(block.obj, block.cood, B = 100,
+     s2.vec.length = 75)
```

With the ldlasso object in hand we may use the plot_ldlasso function to create a summary figure with four diagnostic plots presented in Figure 1. The *plot_ldlasso* function provides four plots. In the upper-left panel we have the $\log_{10} p-$ values for the allelic test of association by physical position in kilobasepairs (kb). Plotted in the upper-right panel are the three solutions found by the *ld_lasso_method* function. In black we have the c_p- optimal solution, and in red and green we have the fused and unfused solutions respectively. In this case the c_p-optimal solution overlaps almost completely with the unfused solution. In the bottom-left panel we have a trace plot for the ldlasso solutions with s_1 fixed. In it each line represents an estimate for a SNP, and we see as the LD LASSO parameter, s_2, decreases, SNP estimates are fused together and converge to the fused solution. Finally, in the lower-right panel we have the estimates for c_p plotted against s_2. Here we see that the value of s_2 that minimizes c_p is represented in the trace plot by a broken vertical line, and that it is essentially the same as the unfused solution. However, we also note that a value of s_2 much closer to the fused solution also has a low value for s_2.

```
> plot_ldlasso(ldlasso.obj)
```

The choice for the optimal intermediate solution is a difficult one, and we have provided in the *ldlasso* package, one method for selection, the c_p- optimal solution. We encourage investigation into other solutions and demonstrate below how the ldlasso object may be manipulated to do so. The first constraint

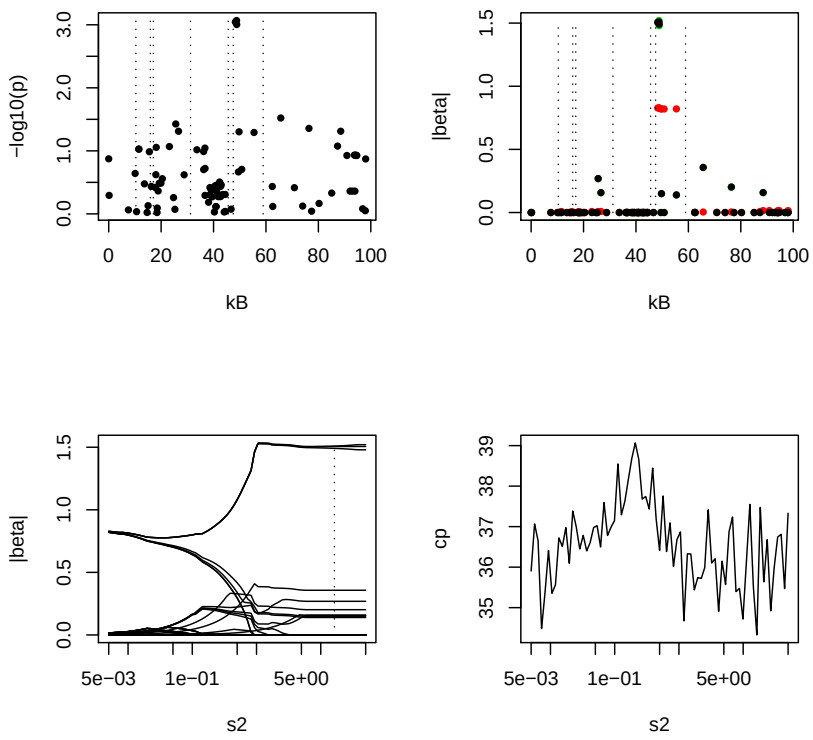

Figure 1: Plot created by *plot_ldlasso*: Top-left: Allelic test of association; Top-right: c_p − optimal, fused, and unfused solutions (black, red, green); Bottom-left: Trace plot with fixed s_1; Bottom-right: c_p estimates vs. s_2

of the ldlasso, s_1, controls the sparsity of the solution, and we have provided a method to select s_1 based on control of the false positive rate. It is interesting, however, to observe the relationship between the vector of solutions and s_1. To that end we would like to investigate the l_1 norm of the solution vector β and how it relates to s_2. The function *ld_lasso_method* computes the ldlasso solution over a vector of s_2 values in order to identify the value that minimizes c_p, and these are stored in the matrix *beta0.mat*. Each row of this matrix

corresponds to the ldlasso solution for the corresponding s_2 value from the s_2 vector. Thus, to find $\sum |\beta_j|$ we simply have to find row sums as follows.

```
> size.vec <- rowSums(abs(ldlasso.obj$cp.obj$beta0.mat))
> ldlasso.obj$s1

[1] 5.9375
```

We then plot size.vec in the top-left panel of Figure 2. It is important to note here that the scale of the vertical axis is on the order of 10^{-13}. Thus, what we see is that for large values of s_2 the l_1 is constant and equal to s_1, but at a critical value of s_2 numerical instability arises, and the l_1 norm becomes unstable. By examination of the trace plot presented in Figure 1 it appears that as s_2 decreases all but one block of identical estimates remains in the fused solution. To investigate this we would like to display the number of nonzero SNP estimates as a function of s_2. Intuitively, we expect that only s_1 will affect the number of nonzero SNP estimates, and to investigate it we must compute $\sum_j I_{|\beta_j|>10^{-6}}$.

```
> p.vec <- rowSums(abs(ldlasso.obj$cp.obj$beta0.mat) > 1e-06)
```

In the top-right panel of Figure 2 we display p.vec plotted against s_2, and see that as s_2 decreases the number of SNPs with nonzero estimates increases monotonically, and thus the fused solution in fact contains SNPs with very small nonzero estimates. As we decrease the value of s_2 we encourage SNP estimates to fuse together. We are interested in an intermediate solution for which the value of s_2 that is used is such that the fusion identifies a weak

signal with breadth without forcing SNPs to be fused together, unnecessarily resulting in a weakening of the apparent signal strength. Thus we would like to create a third diagnostic plot in which we investigate the amount of fusion taking place. To do so we would like to compute $\sum_{(i,j)\in\mathcal{B}} ||\beta_i| - |\beta_j||$. This represents the total difference between SNP estimates that are affected by the fusion constraint s_2. To do so we define the function d.fn in terms of the block map defined by block.cood.

```
> d.fn <- function(beta, block.cood) {
+     d.mat <- abs(outer(abs(beta), abs(beta), FUN = "-"))
+     block.mat <- block.map.matrix(block.cood)
+     tri.mat <- lower.tri(matrix(1, ncol(d.mat), ncol(d.mat)))
+     sum(d.mat[block.mat & tri.mat])
+ }
```

We then apply this function iteratively to the matrix bet0.mat and plot the results in the bottom-left panel of Figure 2. We again observe a critical value of s_2 for which $\sum_{(i,j)\in\mathcal{B}} ||\beta_i| - |\beta_j||$ departs from a constant value and abruptly begins to decrease. The lower-right panel of Figure 2 is the solution using this value of s_2 denoted by s_2'. Using the trace plot, the three basic solutions provided by *ld_lasso_method*, as well as the solution presented in Figure 2, we have an idea of how the ldlasso solution behaves in this region. It is difficult, however, to identify how the solution changes over varying values of s_2. We have presented methods for choosing solutions but to understand the solution space more fully we construct a map of the solution space using the function in the *ldlasso* package 'heatmap.' In Figure 3 a grid is presented with the s_2

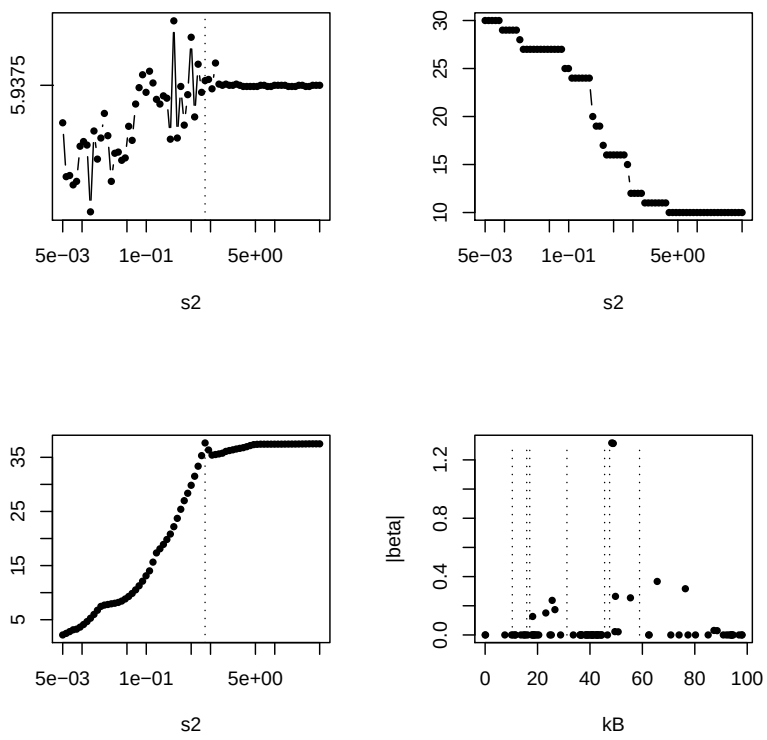

Figure 2: Top Left: $||\beta||^1$ vs. s_2 (note that the vertical axis has length on the order of 10^{-13}) Top Right: $\sum_j I_{|\beta_j|>10^{-6}}$ Lower Left: $\sum_{(i,j)\in\mathcal{B}} ||\beta_i| - |\beta_j||$ with a vertical line drawn at s_2' Lower Right: LD LASSO solution for s_2'

considered along the vertical axis and SNP indices along the horizontal axis. Each element in the grid is colored a shade of gray ranging from white to black in increasing size of the magnitude of the SNP estimate. By default the 'heatmap' function plots a horizontal broken line at the value of s_2 that corresponds to the c_p − optimal solution. To include a line at s_2' found above we simply include it as the s2.indx.

```
> heatmap(ldlasso.obj, s2.indx = s2.indx)
```

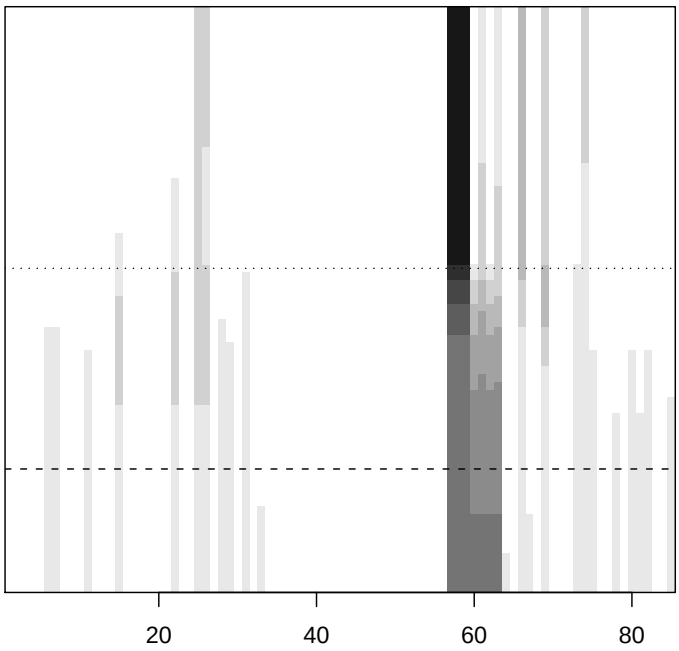

Figure 3: Heat Map for LD LASSO solutions. Vertical axis represents $s_2 \in (5 \times 10^{-3}, 50)$, horizontal axis is the SNP index. Horizontal broken lines at s_2 corresponding to the $c_p -$ optimal solution and s_2'.

In Figure 3 we see that for the largest value of s_2, the solution has a band of three SNPs with estimates that are much darker than the rest, and this corresponds to the unfused solution of the ldlasso. At the bottom of this figure we see the fused solution in which the initial block of three SNPs has been extended to seven SNPs with equal estimates, all much lighter than the original, and this represents the fused solution displayed in red in Figure 1. Intermediate to those solutions are the c_p-optimal solution and the s_2' solution. The dotted line corresponds to the c_p- optimal solution and the dashed line corresponds to the s_2' solution. We see that initially two SNPs to the right of the band are included, but with only a weak signal, and that these SNPs are separated from the original signal by gaps where the signal is absent. If we follow this down to the c_p-optimal solution we see that this solution occurs at the height at which the gaps are filled. We conclude that the region spanned by the seven SNPs is likely to contain a SNP or SNPs that are weakly associated with disease, but that to continue to fuse SNPs by decreasing s_2 is detrimental to the predictive ability of the SNP subset.

5 Discussion

Here we have presented the R package *ldlasso* which allows the user to apply the modified LASSO methodology presented in Younkin et al. An object of class 'gwaa.data' created in the package *GenABEL* is accepted for input and diagnostic plots are easily generated. The package *ldlasso* computes three solutions and displays these, along with diagnostic plots for examination. The

complete space of solutions is displayed as a heat map, and with it choices about what regions are candidates for further study are easily made. In the example data discussed here we see two striking features. First, that the optimal solution is found when the gaps in the signal are filled in, and that to fuse further is detrimental. Also, we see that an entire interval of SNPs from SNP index 35 to 55 shows a complete absence of signal. We expect that some small presence of signal should exist here, and its absence is cause for further study. It is likely, that we will find here SNPs associated with diease which were not included in this data set. For these reasons the region spanned by SNPs with index from 55 to 65 should be considered as a strong candidate region for future sequencing.

References

[1] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society B*, vol. 58, pp. 267–288, 1996.

[2] Y. S. Aulchenko, S. Ripke, A. Isaacs, and C. M. van Duijn, "Genabel: an r library for genome-wide association analysis," *Bioinformatics*, vol. 23(10), pp. 1294–1296, 2007.

[3] C. Pattaro, I. Ruczinski, D. M. Fallin, and G. Parmigiani, "Haplotype block partitioning as a tool for dimensionality reduction in snp association studies," *BMC Genomics*, vol. 9, p. 405, 2008.

[4] R. Tibshirani and M. Saunders, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society B*, vol. 67, pp. 91–108, 2005.

[5] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, pp. 1–33, 1983.

[6] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, pp. 407–451, 2004.