A Fast and Flexible Algorithm for Solving the Lasso in Large-scale and Ultrahigh-dimensional Problems

Junyang Qian[1], Wenfei Du[1], Yosuke Tanigawa[2], Matthew Aguirre[2], Robert Tibshirani[1,2], Manuel A. Rivas[2], and Trevor Hastie[*1,2]

[1]Department of Statistics, Stanford University
[2]Department of Biomedical Data Science, Stanford University

Abstract

Since its first proposal in statistics (Tibshirani, 1996), the lasso has been an effective method for simultaneous variable selection and estimation. A number of packages have been developed to solve the lasso efficiently. However as large datasets become more prevalent, many algorithms are constrained by efficiency or memory bounds. In this paper, we propose a meta algorithm batch screening iterative lasso (BASIL) that can take advantage of any existing lasso solver and build a scalable lasso solution for large datasets. We also introduce **snpnet**, an R package that implements the proposed algorithm on top of **glmnet** (Friedman et al., 2010a) for large-scale single nucleotide polymorphism (SNP) datasets that are widely studied in genetics. We demonstrate results on a large genotype-phenotype dataset from the UK Biobank, where we achieve state-of-the-art heritability estimation on quantitative and qualitative traits including height, body mass index, asthma and high cholesterol.

1 Introduction

The past two decades have witnessed rapid growth in the amount of data available to us. Many areas such as genomics, neuroscience, economics and Internet services are producing big datasets that have high dimension, large sample size, or both. A variety of statistical methods and computing tools have been developed to accommodate this change. See, for example, Friedman et al. (2009); Efron and Hastie (2016); Dean and Ghemawat (2008); Zaharia et al. (2010); Abadi et al. (2016) and the references therein for more details.

1.1 Variable selection via the lasso

In high-dimensional regression problems, we have a large number of predictors, and it is likely that only a subset of them have a relationship with the response and will be useful for prediction. Identifying such a subset is desirable for both scientific interests and the ability to predict outcomes in the future.

*Corresponding author: hastie@stanford.edu

The lasso (Tibshirani, 1996) is a widely used and effective method for simultaneous estimation and variable selection. Given a continuous response $y \in \mathbb{R}^n$ and a model matrix $X \in \mathbb{R}^{n \times p}$, it solves the following regularized regression problem [1]

$$\hat{\beta}(\lambda) = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1, \tag{1}$$

where $\|x\|_q = \left(\sum_{i=1}^n |x_i|^q\right)^{1/q}$ is the vector ℓ_q norm of $x \in \mathbb{R}^n$ and $\lambda \geq 0$ is the tuning parameter. The ℓ_1 penalty on β allows for selection as well as estimation. One typically finds an entire lasso solution path by solving (1) over a grid of λ values $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_L$ and chooses the best λ by cross-validation or by predictive performance on an independent validation set. In R (R Core Team, 2017), several packages, including **glmnet** (Friedman et al., 2010a) and **ncvreg** (Breheny and Huang, 2011), provide efficient procedures to obtain the solution path of (1) for the Gaussian model, and for other generalized linear models with the residual sum of squared replaced by the negative log-likelihood of the corresponding model. Among them, **glmnet**, equipped with highly optimized Fortran subroutines, is widely considered the fastest off-the-shelf lasso solver. It can, for example, fit a sequence of 100 logistic regression models on a sparse dataset with 54 million samples and 7 million predictors within only 2 hours (Hastie, 2015).

1.2 Computational challenges in large-scale problems

The packages mentioned above assume that the dataset or at least its sparse representation can be fully loaded in memory and that the intermediate computational results can all be stored in memory as well. In the case of big data, this can be a real bottleneck. For instance, genotype data commonly used for genome-wide association studies (GWAS) provide a class of ultrahigh-dimensional examples where the number of predictors can easily be in the millions. Researchers used to deal with *wide* data in such studies, where the number of variables was large but the sample size was fairly limited. We were still able to conduct somewhat sophisticated statistical analyses in memory and within a reasonable amount of time, though many of the analyses were actually limited to univariate methods identifying significant SNPs associated with a phenotype. However, recent studies have collected genetic and disease information from very large cohorts. For example, the UK Biobank genotypes and phenotypes dataset (Bycroft et al., 2018) contains about 500,000 individuals and more than 800,000 genotyped SNP measurements per person. This provides unprecedented opportunities to explore more comprehensive genotypic relationships with phenotypes of interest. For polygenic traits such as height and body mass index, specific variants discovered by GWAS only explain a small proportion of the estimated heritability (Visscher et al., 2017). While GWAS with larger sample size have been used to detect more SNPs or rare variants, this extended data also allows us to optimize a prediction problem. Using the lasso, in particular, we can obtain estimates of heritability while also selecting associated SNPs. However, building a multivariate prediction model on a large-scale dataset poses a great computational challenge. Fortunately, each bi-allelic SNP value can be represented by only two bits and a tailored compression scheme can be designed to alleviate the storage burden. In fact, the **PLINK** library (Chang et al., 2015) stores such SNP datasets in a binary format, and implements a number of fast data processing operations and classical statistical procedures directly for that format. However, most general-purpose statistical packages including

[1] Normally there is an unpenalized intercept in the model, but for simplicity we leave it out, or we may assume that both X and y have been centered with mean 0.

those for the lasso assume the data are in the normal double-precision format. If every SNP value is converted to a 32-bit double-precision number, the SNP matrix alone will take up almost a terabyte of storage, and the intermediate computational results will require even more. This highlights the need for efficient and memory-friendly lasso algorithms designed for large datasets.

1.3 A screening-based solution

In this paper, we propose an efficient and scalable meta algorithm for the lasso called Batch Screening Iterative Lasso (BASIL) that is applicable to larger-than-RAM datasets. It can be built on top of any existing mature package with minimal effort and solve the entire lasso solution path. As the name suggests, it is done in an iterative fashion on an adaptively screened subset of variables. Although it works repeatedly on subsets of variables, our procedure guarantee that the solution is not an approximation, but is exact within numerical precision as if we were solving the full lasso problem on all variables. At each iteration, we exploit an efficient, parallelizable screening operation to significantly reduce the problem to a manageable size, solve the resulting much smaller lasso problem, and then assemble and validate the full solution through another efficient, parallelizable step. In particular, the Karush-Kuhn-Tucker (KKT) condition (Boyd and Vandenberghe, 2004) is checked for the full solution after combining the solution of the smaller problem and the assumed solution (often 0's) for the left-out variables. For the lasso, the KKT condition states that $\hat{\beta} \in \mathbb{R}^p$ is a solution to (1) if for all $1 \leq j \leq p$,[2]

$$(1/n) \left| x_j^\top (y - X\hat{\beta}) \right| \begin{cases} = \lambda, & \text{if } \hat{\beta}_j \neq 0, \\ < \lambda, & \text{if } \hat{\beta}_j = 0. \end{cases} \tag{2}$$

The KKT condition allows us to adopt a general strategy: fit the lasso only on a subset of variables assuming the rest having coefficients 0, and then combine into the full solution once the second condition in (2) is verified for the left-out variables. Moreover, with repeated application of this strategy, we are able to obtain an iterative procedure to compute the entire lasso solution path across different λ values.

The screening is inspired by the strong rules proposed in Tibshirani et al. (2012): assume $\hat{\beta}(\lambda_{k-1})$ is the lasso solution in (1) at λ_{k-1}, then the jth predictor is discarded at λ_k if

$$|x_j^\top (y - X\hat{\beta}(\lambda_{k-1}))| < \lambda_k - (\lambda_{k-1} - \lambda_k). \tag{3}$$

The key idea is that the inner product above is almost "non-expansive" in terms of λ and as a result the KKT condition suggests that the discarded variables would have coefficient 0 at λ_k. However it is not a guarantee. The strong rules can fail, though failures occur rarely when $p > n$. In any case, the KKT condition is checked to ensure the exact solution is found. These authors propose an iterative algorithm based on this idea for solving the entire path that is already built into **glmnet**. At each λ, the lasso is fit on variables that survive the strong rule and the KKT condition is checked after each fit to safely set the coefficients of the weak variables to zero. Our algorithm proceeds in a similar way but is designed to efficiently handle datasets that are too big to fit into the memory. Considering the fact that screening and KKT check are costly in the sense of disk Input/Output (I/O) operations, we solve *a series of* models per iteration, trying to reduce the total number

[2]Strictly speaking, some variables may have "=" sign even when their coefficients are 0. They are probably in a transition state from zero to nonzero or the other way on the solution path.

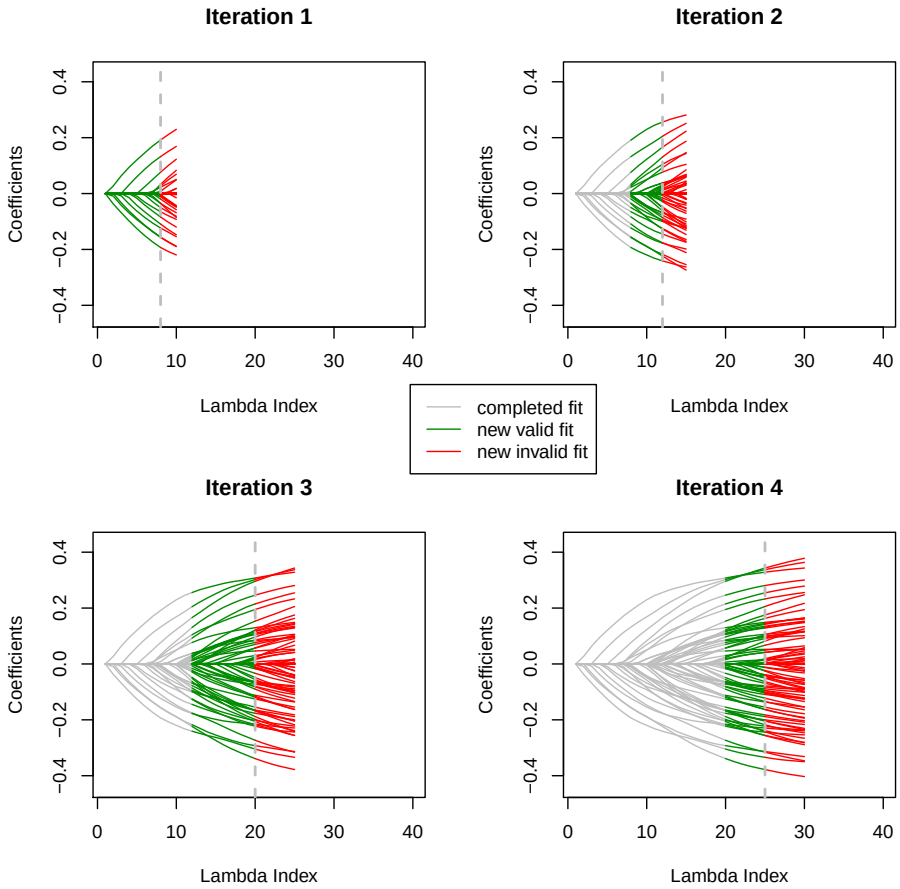

Figure 1: The lasso coefficient profile that shows the progression of the BASIL algorithm. The previously finished part of the path is colored grey, the newly completed and verified is in green, and the part that is newly computed but failed the verification is colored red.

of expensive disk read operations. At each iteration, we roll out the solution path progressively, which is illustrated in Figure 1 and will be detailed in the next section. In addition, we propose optimization specific for the SNP data in the UK Biobank studies to speed up the procedure.

1.4 Outline of the paper

The rest of the paper is organized as follows.

- Section 2 describes the proposed batch screening iterative lasso (BASIL) algorithm for the Gaussian family in detail and its extension to other problems such as logistic regression.
- Section 3 discusses related methods and packages for solving large-scale lasso problems.

- In Section 4, we present an analysis of the UK Biobank data using our implementation of the proposed algorithm. To our best knowledge, this is the first whole-genome multi-SNP-phenotype association analysis at a biobank-scale dataset, which gives improved heritability estimates for the traits concerned.
- In Section 5, we close the paper with a discussion of possible variations of the algorithm and future work.

2 Methods and algorithms

For convenience, we introduce some notation. Let $\Omega = \{1, 2, \ldots, p\}$ be the universe of variable indices. For $1 \leq \ell \leq L$, let $\hat{\beta}(\lambda_\ell)$ be the lasso solution at $\lambda = \lambda_\ell$, and $\mathcal{A}(\lambda_\ell) = \{1 \leq j \leq p : \hat{\beta}_j(\lambda_\ell) \neq 0\}$ be the active set. When X is a matrix, we use $X_\mathcal{S}$ to represent the submatrix including only columns indexed by $\mathcal{S}$. Similarly when β is a vector, $\beta_\mathcal{S}$ represents the subvector including only elements indexed by $\mathcal{S}$. Given any two vectors $a, b \in \mathbb{R}^n$, the dot product or inner product can be written as $a^\top b = \langle a, b \rangle = \sum_{i=1}^n a_i b_i$. We use predictors, features, variables and variants interchangeably.

2.1 Batch Screening Iterative Lasso (BASIL)

We introduce our new iterative algorithm to fit the lasso for ultrahigh-dimensional problems. Recall that our goal is to compute the exact lasso solution (1) over a sequence of regularization parameters $\lambda_1 > \lambda_2 > \cdots > \lambda_L \geq 0$. As in **glmnet**, we often choose $L = 100$ and $\lambda_1 = \max_{1 \leq j \leq p} |x_j^\top r^{(0)}|$, the largest λ at which the estimated coefficients start to deviate from zero. Here $r^{(0)} = y$ if we do not include an intercept term and $r^{(0)} = y - \bar{y}$ if we do. In general, $r^{(0)}$ is the residual of regressing y on the unpenalized variables, if any. The other λ's can be determined, for example, by an equally spaced array on the log scale. Two key algorithmic components that contribute to the efficiency of **glmnet** are warm starts and the strong rules. Warm start provides a good initialization for solving the lasso at a new λ, while the strong rules temporarily leave out a significant portion of the variables so that we only need to consider solutions containing the remaining subset of variables.

The BASIL algorithm can be viewed as a batch version of the strong rules. At each iteration we attempt to find a valid solution for *multiple* λ values in the path. This reduces disk reads of the big dataset. In detail, the algorithm progresses in the following way. We start with an empty strong set $\mathcal{S}^{(0)} = \emptyset$ and active set $\mathcal{A}^{(0)} = \emptyset$. In our context, the strong set refers specifically to the presumably much smaller subset of variables on which the lasso fit is computed at each iteration. The active set is the subset of variables with nonzero lasso coefficients. Each iteration has three major steps: screening, fitting and checking.

In the screening step, an updated strong set is found as the candidate for the subsequent fitting. Suppose that so far (valid) lasso solutions have been found for $\lambda_1, \ldots, \lambda_\ell$ but not for $\lambda_{\ell+1}$. The new set will be based on the lasso solution at λ_ℓ. In particular, we will select the top M variables with largest absolute inner products $|\langle x_j, y - X\hat{\beta}(\lambda_\ell)|$. They are the variables that are most likely to be in the lasso model for the next values of λ. In addition, we include the ever-active variables at $\lambda_1, \ldots, \lambda_\ell$ because they have been "important" variables and might continue to be important at a later stage. Also, for packages such as **glmnet** that are designed to compute the solution path from the beginning, the inclusion of ever-active variables allows the solutions at earlier λ's but computed in this iteration to be consistent with those from the previous iterations.

In the fitting step, the lasso is fit on an updated strong set for the subsequent λ's along our predetermined sequence: $\lambda_{\ell+1}, \ldots, \lambda_{\ell'}$. Here ℓ' is often smaller than L because we do not have to solve for all of the remaining λ values on this strong set. The full lasso solutions at much smaller λ's are very likely to have active variables outside of the current strong set. In other words even if we were to compute solutions for those very small λ values on the current strong set, they would probably fail the KKT test. These λ's are left to later iterations, when the strong set is expanded.

In the checking step, we check if the newly obtained solution on the strong set can be part of the full solution by computing the KKT condition. Given a solution $\hat{\beta}_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$ to the sub-problem, if we can verify for every left-out variable j that $(1/n)|\langle x_j, y - X_{\mathcal{S}}\hat{\beta}_{\mathcal{S}}\rangle| < \lambda$, we can then safely set their coefficients to 0. The full lasso solution $\hat{\beta}(\lambda) \in \mathbb{R}^p$ is then assembled by letting $\hat{\beta}_{\mathcal{S}}(\lambda) = \hat{\beta}_{\mathcal{S}}$ and $\hat{\beta}_{\Omega\setminus\mathcal{S}}(\lambda) = 0$.

The three steps above can be applied repeatedly to roll out the complete lasso solution path for the original problem. However, if our goal is choosing the best model along the path, we can stop fitting once an optimal model is found evidenced by the performance on a validation set. At a high level, we run the iterative procedure on the training data, monitor the error on the validation set, and stop when the model starts to overfit, or in other words, validation error shows a clear upward trend.

We describe below some extensions that can be incorporated into our procedure. The full version is given in Algorithm 1.

Relaxed lasso The lasso is known to shrink coefficients to exclude noise variables, but sometimes such shrinkage can degrade the predictive performance due to its effect on actual signal variables. Meinshausen (2007) introduces the relaxed lasso to correct for the potential over-shrinkage of the original lasso estimator. They propose a refitting step on the active set of the lasso solution with less regularization, while a common way of using it is to fit a standard OLS on the active set. The active set coefficients are then set to

$$\hat{\beta}_{\mathcal{A},\text{Relax}}(\lambda) = \underset{\beta_{\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}}{\operatorname{argmin}} \|y - X_{\mathcal{A}}\beta_{\mathcal{A}}\|_2^2,$$

whereas the coefficients for the inactive set remain at 0. This refitting step can revert some of the shrinkage bias introduced by the vanilla lasso. It doesn't always reduce prediction error due to the accompanied increase in variance when there are many variables in the model or when the signals are weak. That being said, we can still insert a relaxed lasso step with little effort in our iterative procedure: once a valid lasso solution is found for a new λ, we may refit with OLS. As we iterate, we can monitor validation error for the lasso and the relaxed lasso. The relaxed lasso will generally end up choosing a smaller set of variables than the lasso solution in the optimal model.

Adjustment covariates In some applications such as GWAS, there may be confounding variables $Z \in \mathbb{R}^{n\times q}$ that we want to adjust for in the model. Population stratification, defined as the existence of a systematic ancestry difference in the sample data, is one of the common factors in GWAS that can lead to spurious discoveries. This can be controlled for by including some leading principal components of the SNP matrix as variables in the regression (Price et al., 2006). In the presence of such variables, we instead solve

$$(\hat{\alpha}(\lambda), \hat{\beta}(\lambda)) = \underset{\alpha\in\mathbb{R}^q, \beta\in\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n}\|y - Z\alpha - X\beta\|_2^2 + \lambda\|\beta\|_1. \tag{4}$$

This variation can be easily handled with small changes in the algorithm. Instead of initializing the residual with the response y, we set $r^{(0)}$ equal to the residual from the regression of y on the covariates. In the fitting step, in addition to the variables in the strong set, we include the covariates but leave their coefficients unpenalized as in (4). Notice that if we want to find relaxed lasso fit with the presence of adjustment covariates, we need to include those covariates in the OLS as well, i.e.,

$$(\hat{\alpha}_{\text{Relax}}(\lambda), \hat{\beta}_{\mathcal{A},\text{Relax}}(\lambda)) = \underset{\alpha \in \mathbb{R}^q, \beta_{\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}}{\text{argmin}} \|y - Z\alpha - X_{\mathcal{A}}\beta_{\mathcal{A}}\|_2^2. \tag{5}$$

Algorithm 1 BASIL for the Gaussian Model

1: **Initialization**: active set $\mathcal{A}^{(0)} = \emptyset$, initial residual $r^{(0)}$ (with respect to the intercept or other unpenalized variables), a short list of initial parameters $\Lambda^{(0)} = \{\lambda_1, \ldots, \lambda_{L^{(0)}}\}$.

2: **for** $k = 0$ **to** K **do**

3: **Screening**: for each $1 \leq j \leq p$, compute inner product with current residual $c_j^{(k)} = \langle x_j, r^{(k)} \rangle$. Construct the strong set

$$\mathcal{S}^{(k)} = \mathcal{A}^{(k)} \cup \mathcal{E}_M^{(k)},$$

where $\mathcal{E}_M^{(k)}$ is the set of M variables in $\Omega \setminus \mathcal{A}^{(k)}$ with largest $|c^{(k)}|$.

4: **Fitting**: for $\lambda \in \Lambda^{(k)}$, solve the lasso only on the strong set $\mathcal{S}^{(k)}$, and find the coefficients $\hat{\beta}^{(k)}(\lambda)$ and the residuals $r^{(k)}(\lambda)$.

5: **Checking**: search for the smallest λ such that the KKT conditions are satisfied, i.e.,

$$\bar{\lambda}^{(k)} = \min \left\{ \lambda \in \Lambda^{(k)} : \max_{j \in \Omega \setminus \mathcal{S}^{(k)}} (1/n)|x_j^\top r^{(k)}(\lambda)| < \lambda \right\}.$$

Let the current active set $\mathcal{A}^{(k+1)}$ and residuals $r^{(k+1)}$ defined by the solution at $\bar{\lambda}^{(k)}$. Define the next parameter list $\Lambda^{(k+1)} = \{\lambda \in \Lambda^{(k)} : \lambda < \bar{\lambda}^{(k)}\}$. Extend this list if it consists of too few elements. For $\lambda \in \Lambda^{(k)} \setminus \Lambda^{(k+1)}$, we obtain new valid lasso solutions:

$$\hat{\beta}_{\mathcal{S}^{(k)}}(\lambda) = \hat{\beta}^{(k)}(\lambda), \quad \hat{\beta}_{\Omega \setminus \mathcal{S}^{(k)}}(\lambda) = 0.$$

6: (Optional) Relaxed Lasso: for $\lambda \in \Lambda^{(k)} \setminus \Lambda^{(k+1)}$, find the relaxed lasso fit as in (5).

7: (Optional) Early Stopping: exit the iteration when the mean squared prediction error on an independent validation set starts to increase for validated lasso solutions.

8: **end for**

2.2 Computational considerations

Screening and checking are the steps where we need to deal with the full dataset. To deal with the memory bound, we can use memory-mapped I/O. In R, **bigmemory** (Kane et al., 2013) provides a convenient implementation for that purpose. That being said, we do not want to rely on that for intensive computation modules such as cyclic coordinate descent, because frequent visits to the on-disk data would still be slow. Instead, since the subset of strong variables would be small, we

[2] If the parameter list did not change from the previous iteration, include more variables (e.g., $2M$) with largest $|c^{(k)}|$.

can afford to bring them to memory and do fast lasso fitting there. We only use the full memory-mapped dataset in KKT checking and screening. Moreover since checking in the current iteration can be done together with the screening in the next iteration, effectively only one expensive pass over the full dataset is needed every iteration.

2.3 Extension to general problems

It is straightforward to extend the algorithm from the Gaussian case to more general problems. In fact, the only changes we need to make are the screening step and the strong set update step. Wherever the strong rules can be applied, we have a corresponding version of the iterative algorithm. In Tibshirani et al. (2012), the general problem is

$$\hat{\beta}(\lambda) = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \; f(\beta) + \lambda \sum_{j=1}^{r} c_j \|\beta_j\|_{p_j}, \tag{6}$$

where f is a convex differentiable function, and for all $1 \leq j \leq r$, $c_j \geq 0, p_j \geq 1$, and β_j can be a scalar or vector. The general strong rule discards predictor j if

$$\|\nabla_j f(\hat{\beta}(\lambda_{k-1}))\|_{q_j} < c_j(2\lambda_k - \lambda_{k-1}), \tag{7}$$

where $1/p_j + 1/q_j = 1$. Hence, our algorithm can adapt and screen by choosing variables with large values of $\|\nabla_j f(\hat{\beta}(\lambda_{k-1}))\|_{q_j}$ that are not in the current active set.

Logistic regression In the lasso penalized logistic regression (Friedman et al., 2010b) where the observed outcome $y \in \{0,1\}^n$, the convex differential function in (6) is

$$f(\beta) = -\frac{1}{n} \sum_{i=1}^{n} \left(y_i \log p_i + (1 - y_i) \log(1 - p_i) \right).$$

where $p_i = 1/(1 + \exp(-x_i^\top \beta))$ for all $1 \leq i \leq n$. The rule in (7) is reduced to

$$|x_j^\top (y - \hat{p}(\lambda_{k-1}))| < \lambda_k - (\lambda_{k-1} - \lambda_k),$$

where $\hat{p}(\lambda_{k-1})$ is the predicted probabilities at $\lambda = \lambda_{k-1}$. Similar to the Gaussian case, we can still fit relaxed lasso and allow adjustment covariates in the model to adjust for confounding effect.

Cox's proportional hazards model In the usual survival analysis framework, for each sample, in addition to the predictors $x_i \in \mathbb{R}^p$ and the observed time y_i, there is an associated right-censoring indicator $\delta_i \in \{0,1\}$ such that $\delta_i = 0$ if failure and $\delta_i = 1$ if right-censored. Let $t_1 < t_2 < ... < t_m$ be the increasing list of unique failure times, and $j(i)$ denote the index of the observation failing at time t_i. The Cox's proportional hazards model (Cox, 1972) assumes the hazard for the ith individual as $h_i(t) = h_0(t) \exp(x_i^\top \beta)$ where $h_0(t)$ is a shared baseline hazard at time t. We can let $f(\beta)$ be the negative log partial likelihood in (6) and screen based on its gradient at the most recent lasso solution as suggested in (7). In particular,

$$f(\beta) = -\sum_{i=1}^{m} \left(x_{j(i)}^\top \beta - \log \left(\sum_{j \in R_i} \exp(x_j^\top \beta) \right) \right),$$

where R_i is the set of indices j with $y_j \geq t_i$ (those at risk at time t_i). The implementation is not provided in our package yet but will be added in the future.

3 Related methods and packages

There are a number of existing screening rules for solving big lasso problems. Sobel et al. (2009) use a screened set to scale down the logistic lasso problem and check the KKT condition to validate the solution. Their focus, however, is on selecting a lasso model of particular size and only the initial screened set is expanded if the KKT condition is violated. In contrast, we are interested in finding the whole solution path (before overfitting). We adopt a sequential approach and keep updating the screened set at each iteration. This allows us to potentially keep the screened set small as we move along the solution path. Other rules include the SAFE rule (El Ghaoui et al., 2010), Sure Independence Screening (Fan and Lv, 2008), and the DPP and EDPP rules (Wang et al., 2015).

We expand the discussion on these screening rules a bit. Fan and Lv (2008) exploits marginal information of correlation to conduct screening but the focus there is not optimization algorithm. Most of the screening rules mentioned above (except for EDPP) use inner product with the current residual vector to measure the importance of each predictor at the next λ — those under a threshold can be ignored. The key difference across those rules is the threshold defined and whether the resulting discard is safe. If it is safe, one can guarantee that only one iteration is needed for each λ value, compared with others that would need more rounds if an active variable was falsely discarded. Though the strong rules rarely make this mistake, safe screening is still a nice feature to have in single-λ solutions. However, under the batch mode we consider due to the desire of reducing the number of full passes over the dataset, the advantage of safe threshold may not be as much. In fact, one way we might be able to leverage the safe rules in the batch mode is to first find out the set of candidate predictors for the several λ values up to λ_k we wish to solve in the next iteration based on the current inner products and the rules' safe threshold, and then solve the lasso for these parameters. Since these rules can often be conservative, we would then have strong incentive to solve for, say, one further λ value λ_{k+1} because if the current screening turns out to be a valid one as well, we will find one more lasso solution and move one step forward along the λ sequence we want to solve for. This can potentially save one iteration of the procedure and thus one expensive pass over the dataset. The only cost there is computing the lasso solution for one more λ_{k+1} and computing inner products with one more residual vector at λ_{k+1} (to check the KKT condition). The latter can be done in the same pass as we compute inner products at λ_k for preparing the screening in the next iteration, and so no additional pass is needed. Thus under the batch mode, the property of safe screening may not be as important due to the incentive of aggressive model fitting. Nevertheless it would be interesting to see in the future EDPP-type batch screening. It uses inner products with a modification of the residual vector. Our algorithm still focuses of inner products with the vanilla residual vector.

To address the large-scale lasso problems, several packages have been developed such as **biglasso** (Zeng and Breheny, 2017), **bigstatsr**(Privé et al., 2018), **oem** (Huling and Qian, 2018) and the lasso routine from **PLINK** 1.9 (Chang et al., 2015).

Among them, **oem** specializes in tall data (big n) and can be very slow when $p > n$. In many real data applications including ours, the data can be both large-sample and high-dimensional. However, we might still be able to use **oem** for the small lasso subroutine since a large number of variables have already been excluded. The other packages, **biglasso**, **bigstatsr**, **PLINK** 1.9, all provide efficient implementations of the pathwise coordinate descent with warm start. **PLINK** 1.9 is specifically developed for genetic datasets and is widely used in GWAS and research in population genetics. In **bigstatsr**, the `big_spLinReg` function adapts from the `biglasso` function in **biglasso** and incorporates a Cross-Model Selection and Averaging (CMSA) procedure, which is

a variant of cross-validation that saves computation by directly averaging the results from different folds instead of retraining the model at the chosen optimal parameter. They both use memory-mapping to process larger-than-RAM, on-disk datasets as if they were in memory, and based on that implement coordinate descent with strong rules and warm start.

The main difference between BASIL and the algorithm these packages use is that BASIL tries to solve a series of models every full scan of the dataset (at checking and screening) and thus effectively reduce the number of passes over the dataset. This difference may not be significant in small or moderate-sized problems, but can be critical in big data applications especially when the dataset cannot be fully loaded into the memory. A full scan of a larger-than-RAM dataset can incur a lot of swap-in/out between the memory and the disk, and thus a lot of disk I/O operations, which is known to be orders of magnitude slower than in-memory operations. Thus reducing the number of full scans can greatly improve the overall performance of the algorithm.

Aside from potential efficiency consideration, all of those packages aforementioned have to re-implement a variety of features existent in many small-data solutions but for big-data context. Nevertheless, currently they don't provide as much functionality as needed in our real-data application. First, current implementations of **biglasso**, **bigstatsr** and **PLINK** 1.9 all standardize the predictors beforehand, but in the application we show in the next section, it is more reasonable to leave the predictors unstandardized. Also, it can take some effort to convert the data to the desired format by these packages. This would be a headache if the raw data is in some special format and one cannot afford to first convert the full dataset into an intermediate format for which a tool is provided to convert to the desired one by **biglasso** or **bigstatsr**. This can happen, for example, if the raw data is highly compressed in a special format. For the BED binary format we work with in our application, `readRAW_big.matrix` function from **BGData** can convert a raw file to a `big.matrix` object desired by **biglasso**, and `snp_readBed` function from **bigsnpr** allows one to convert it to `FBM` object desired by **bigstatsr**. However, **bigsnpr** doesn't take input data that has any missing values, which are prevalent in an SNP matrix ($\approx 70\%$ in our dataset). Although **PLINK** 1.9 works directly with the BED binary file, its lasso solver currently only supports the Gaussian family, and it doesn't return the full solution path. Instead it returns the solution at the smallest λ value computed and needs a good heritability estimate as input from the user, which may not be immediately available.

We summarize the main advantages of the BASIL algorithm:

- **Input data flexibility**. Our algorithm allows one to deal directly with any data type as long as the screening and checking steps are implemented, which is often very lightweight development work like matrix multiplication. This can be important in large-scale applications especially when the data is stored in a compressed format or a distributed way since then we would not need to unpack the full data and can conduct KKT check and screening on its original format. Instead only a small screened subset of the data needs to be converted to the desired format by the lasso solver in the fitting step.

- **Model flexibility**. We can easily transfer the modeling flexibility provided by existing packages to the big data context, such as the options of standardization, sample weights, lower/upper coefficient limits and other families in generalized linear models provided by existing packages such as **glmnet**. This can be useful, for example, when we may not want to standardize predictors already in the same unit to avoid unintentionally different penalization of the predictors due to difference in their variance.

- **Effortless development**. The BASIL algorithm allows one to maximally reuse the existing lasso solutions for small or moderate-sized problems. The main extra work would be an implementation of batch screening and KKT check with respect to a particular data type. For example, in the **snpnet** package, we are able to quickly extend the in-memory **glmnet** solution to large-scale, ultrahigh-dimentional SNP data. Moreover, the existing convenient data interface provided by the **BEDMatrix** package further facilitates our implementation.

- **Computational efficiency**. Our design reduces the number of visits to the original data that sits on the disk, which is crucial to the overall efficiency as disk read can be orders of magnitude slower than reading from the RAM. The key to achieving this is to bring batches of promising variables into the main memory, hoping to find the lasso solutions for more than one λ value each iteration and check the KKT condition for those λ values in one pass of the entire dataset.

4 Application: UK Biobank

In this section, we describe a real-data application on the UK Biobank that in fact motivates our development of the BASIL algorithm.

The UK Biobank (Bycroft et al., 2018) is a very large, prospective population-based cohort study with individuals collected from multiple sites across the United Kingdom. It contains extensive genotypic and phenotypic detail such as genomewide genotyping, questionnaires and physical measures for a wide range of health-related outcomes for over 500,000 participants, who were aged 40-69 years when recruited in 2006-2010. In this study, we are interested in the relationship between an individual's genotype and his/her phenotypic outcome. While GWAS focus on identifying SNPs that may be marginally associated with the outcome using univariate tests, we would like to find relevant SNPs in a multivariate prediction model using the lasso. A recent study (Lello et al., 2018) fits the lasso to a similar subset of the dataset after one-shot univariate p-value screening and suggests improvement in explaining the variation in the phenotypes. However, the left-out variants with relatively weak marginal association may still provide additional predictive power in a multivariate environment. The BASIL algorithm enables us to fit the lasso model at full scale and gives further improvement in the explained variance over the alternative models considered.

We focused on 337,199 White British unrelated individuals out of the full set of over 500,000 from the UK Biobank dataset (Bycroft et al., 2018) that satisfy the same set of population stratification criteria as in DeBoever et al. (2018): (1) self-reported White British ancestry, (2) used to compute principal components, (3) not marked as outliers for heterozygosity and missing rates, (4) do not show putative sex chromosome aneuploidy, and (5) have at most 10 putative third-degree relatives. These criteria are meant to reduce the effect of confoundedness and unreliable observations. Each individual has up to 805,426 measured variants, and each variant is encoded by one of the four levels where 0 corresponds to homozygous major alleles, 1 to heterozygous alleles, 2 to homozygous minor alleles and NA to a missing genotype. In addition, we have available covariates such as age, sex, and forty pre-computed principal components of the SNP matrix.

There are thousands of measured phenotypes in the dataset. For demonstration purpose, we analyze four phenotypes that are known to be highly or moderately heritable and polygenic. For these complex traits, univariate studies may not find SNPs with smaller effects, but the lasso model may include them and predict the phenotype better. We look at two quantitative traits — standing

height and body mass index (BMI) (Tanigawa et al., 2019), and two qualitative traits — asthma and high cholesterol (HC) (DeBoever et al., 2018).

4.1 Implementation details

In this section, we describe several aspects of the experimental details in our application.

Training/Validation/Test splitting Since the number of observations is large, we can afford to set aside an independent validation set without resorting to the costly cross-validation to find an optimal regularization parameter. We also leave out a subset of observations as test set to evaluate the final model. In particular, we randomly partition the original dataset so that 60% is used for training, 20% for validation and 20% for test. The lasso solution path is fit on the training set, the desired regularization selected on the validation set, and the resulting model is evaluated on the test set.

Adjustment for confounders In genetic studies, spurious associations are often found due to confounding factors. Among the others, one major source is the so-called population stratification (Patterson et al., 2006). To adjust for that effect, it is common is to introduce the top principal components and include them in the regression model. Therefore in the lasso method, we are going to solve (4) where in addition to the SNP matrix X, we let Z include covariates such as age, sex and the top 10 PCs left unpenalized.

Missing values Missing values are present in the dataset. As quality control normally done in genetics, we first discard observations whose phenotypic value of interest is not available. We further exclude variants whose missing rate is greater than 10% or the minor allele frequency (MAF) is less than 0.1%, which results in around 685,000 SNPs for height. [3] For those remaining variants, mean imputation is conducted to fill the missing SNP values; that is, the missing values in every SNP are imputed with the mean observed level of that SNP in the population under study.

Standardization in lasso When it comes to the lasso fitting, there are some subtleties that can affect its variable selection and prediction performance. One of them is variable standardization. It is often a step done without much thought to deal with heterogeneity in variables so that they are treated fairly in the objective. However in our studies, standardization may create some undesired effect. To see this, notice that all the SNPs can only take values in 0, 1, 2 and NA — they are already on the same scale by nature. As we know, standardization would use the current standard deviation of each predictor as the divisor to equalize the variance across all predictors in the lasso fitting that follows. In this case, standardization would unintentionally inflate the magnitude of rare variants and give them an advantage in the selection process since their coefficients effectively receive less penalty after standardization. In Figure 2, we can see the distribution of standard deviation across all variants in our dataset. Hence, to avoid potential spurious findings, we choose not to standardize the variants in the experiments.

[3] In particulr, 685,362 for height, 685,371 for BMI, 685,357 for asthma and 685,357 for HC. The number varies because the criteria are evaluated on the subset of individuals whose phenotypic value is observed (after excluding the missing ones), which can be different across different phenotypes.

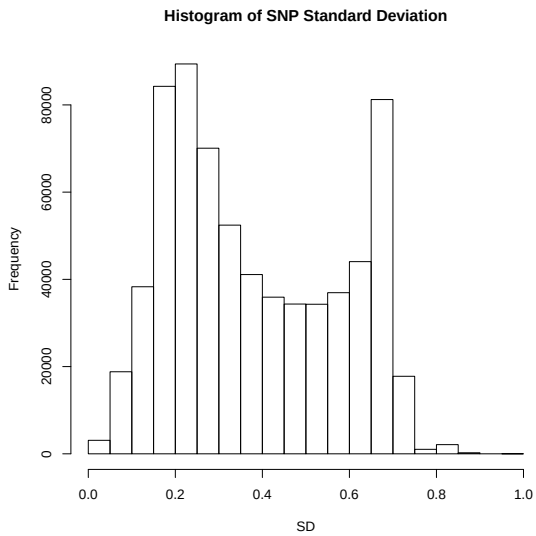

Figure 2: Histogram of the standard deviations of the SNPs. They are computed *after* mean imputation of the missing values because they would be the exact standardization factors to be used if the lasso were applied with variable standardization on the mean-imputed SNP matrix.

SNP-specific optimization On the computational side, we use several techniques to speed up the computation. First, the KKT check can be easily parallelized by splitting on the features when multi-core machines are available. The speedup of this part is immediate and (slightly less than) proportional to the number of cores available. Second, specific to the application, we exploit the fact that there are only 4 levels for each SNP value and design a faster inner product routine to replace normal float number multiplication in the KKT check step. In fact, given any SNP vector $x \in \{0, 1, 2, \mu\}^n$ where μ is the imputed value for the missing ones, we can write the dot product with a vector $r \in \mathbb{R}^n$ as

$$x^\top r = \sum_{i=1}^{n} x_i r_i = 1 \cdot \sum_{i:x_i=1} r_i + 2 \cdot \sum_{i:x_i=2} r_i + \mu \cdot \sum_{i:x_i=\mu} r_i.$$

We see that the terms corresponding to 0 SNP value can be ignored because they don't contribute to the final result. This will significantly reduce the number of arithmetic operations needed to compute the inner product with rare variants. Further, we only need to set up 3 registers, each for one SNP value accumulating the corresponding terms in r. A series of multiplications is then converted to summations. In our UK Biobank studies, although the SNP matrix is not sparse enough to exploit sparse matrix representation, it still has around 70% 0's. We conduct a small experiment to compare the time needed to compute $X^\top R$, where $X \in \{0, 1, 2, 3\}^{n \times p}, R \in \mathbb{R}^{p \times k}$. The proportions for the levels in X are about $70\%, 10\%, 10\%, 10\%$, similar to the distribution of SNP levels in our study, and R resembles the residual matrix when checking the KKT condition. The number of residual vectors is $k = 20$. The mean time over 100 repetitions is shown in Table 1.

We implement the procedure with all the optimizations in an R package called **snpnet**, which is currently available at `https://github.com/junyangq/snpnet`. It assumes BED file format (Chang

Multiplication Method	$n = 200, p = 800$	$n = 2000, p = 8000$
Standard	3.20	306.01
SNP-Optimized	1.32	130.21

Table 1: Timing performance (milliseconds) on multiplication of SNP matrix and residual matrix. The methods are all implemented in C++ and run on a Macbook with 2.9 GHz Intel Core i7 and 8 GB 1600 MHz DDR3.

R Package	Elapsed Time (minutes)
bigstatsr (Privé et al., 2018)	2.93 + 56.80
biglasso(Zeng and Breheny, 2017)	4.55 + 54.27
PLINK (Chang et al., 2015)	53.52
snpnet	**44.79**

Table 2: Timing comparison on a synthetic dataset of size $n = 50,000$ and $p = 100,000$. The time for **bigstatsr** and **biglasso** has two components: one for the conversion to the desired data type and the other for the actual computation. The experiments are all run with 16 cores and 64 GB memory.

et al., 2015) of the SNP matrix, fits the lasso solution path and allows early stopping if a validation dataset is provided. In order to achieve better efficiency, we suggest using **snpnet** together with **glmnetPlus**, a warm-started version of **glmnet**, which is currently available at `https://github.com/junyangq/glmnetPlus`. It allows one to provide a good initialization of the coefficients to fit part of the solution path instead of always starting from the all-zero solution by **glmnet**.

Timing performance Lastly, we are going to provide some timing comparison with existing packages. As mentioned in previous sections, those packages provide different functionalities and have different restrictions on the dataset. For example, most of them (**biglasso, bigstatsr**) assume that there are no missing values, or the missing ones have already been imputed. In **bigsnpr**, for example, we shouldn't have SNPs with 0 MAF either. Some packages always standardize the variants before fitting the lasso. To provide a common playground, we create a synthetic dataset with no missing values, and follow a standardized lasso procedure in the fitting stage, simply to test the computation. The dataset has 50,000 samples and 100,000 variables, and each takes value in the SNP range, i.e., in 0, 1, or 2. We fit the first 50 lasso solutions along a prefix λ sequence that contains 100 initial λ values (like early stopping for most phenotypes). The total time spent is displayed in Table 2. We uses 128GB memory and 16 cores for the computation.

From the table, we see that **snpnet** is at about 20% faster than other packages concerned. The numbers before the "+" sign are the time spent on converting the raw data to the required data format by those packages. The second numbers are time spent on actual computation.

It is important to note though that the performance relies not only on the algorithm, but also heavily on the implementations. The other packages in comparison all have their major computation done with C++ or Fortran. Ours, for the purpose of meta algorithm where users can easily integrate with any lasso solver in R, still has a significant portion (the iterations) in R and multiple rounds of cross-language communication. That can degrade the timing performance to some degree. If there is further pursuit of speed performance, there is still space for improvement by more designated implementation.

4.2 Evaluation

Goodness of fit For quantitative response, a common measure for goodness-of-fit is R^2. For any given linear estimator $\hat{\beta}$ and data (y, X),

$$R^2 = 1 - \frac{\|y - X\hat{\beta}\|^2}{\|y - \bar{y}\|^2}.$$

We evaluate this criteria for all the training, validation and test datasets. For dichotomous response, misclassification error could be used but it would also depend on the calibration. Instead the receiver operating characteristic (ROC) curve provides more information and illustrates the tradeoff between true positive and false positive rates under different thresholds. The AUC computes the area under the ROC curve — a larger value indicates a generally better classifier. We will evaluate AUCs on the training, validation and test sets.

Heritability In genetic studies, one of the central questions is whether the variation in a trait is due to genetic factors, environmental factors, or interaction of both. Heritability provides a measure that quantifies the contribution of the genetic component. Different models for heritability include twin studies (Polderman et al., 2015) and linear mixed models (Patterson and Thompson, 1971; Yang et al., 2010, 2011). There is a distinction between narrow-sense and broad-sense heritability. The former is defined as the proportion of total phenotypic variance in a population that is due to variation in additive genetic factors and the latter is the proportion due to variation in total genetic factors including interactions between genes (Visscher et al., 2008). We assume an additive linear model and use R^2 on the test set to measure narrow-sense heritability for quantitative traits; in fact, such test R^2 provides a lower bound of the true narrow-sense heritability and we would like to achieve as tight a bound as possible. For binary traits, there are methods that use latent factors to define heritability (Lee et al., 2011). However this is not the focus of the paper, and we will only compare heritability estimation for quantitative traits.

4.3 Other methods

We compare the performance of the lasso with related methods to have a sense of the contribution of different components. Starting from the baseline, we fit a linear model that includes only age and sex (Model 1 in the tables below), and then one that includes additionally the top 10 principal components (Model 2). These are the adjustment covariates used in our main lasso fitting and we use these two models to highlight the contribution of the SNP information on top of that contained in age, sex and the top 10 PCs. In addition, the strongest univariate model is also evaluated (Model 3). This includes adjustment covariates together with a single SNP that is most correlated with the outcome after adjusted for the covariates.

We also compare with a univariate method that has some multivariate flavor (Mode 4 and 5). We select a subset of the K most marginally significant variants (after adjusting for the covariates), and use their univariate coefficients to form a linear combination as the new variable. An OLS is then fit on the new variable together with the adjustment variables. It is similar to a one-step partial least squares (Wold, 1975) with p-value based truncation. We take $K = 10,000$ and $100,000$ in the experiments.

In addition, we compare with a hierarchical sequence of linear models where each is fit on a subset of the most significant SNPs. In particular, the ℓ-th model selects $\ell \times 1000$ SNPs with the

smallest univariate p-values, and a multivariate linear or logistic regression is fit on those variants jointly. The sequence of models are evaluated on the validation set, and the one with the smallest validation error is chosen. We call this method Sequential LR for convenience in the following result part (Model 6).

4.4 Results

We present results of the lasso and related methods for quantitative traits including standing height and BMI, and for qualitative traits including asthma and high cholesterol. A comparison of the univariate p-values and the lasso coefficients for all these traits is showed in the form of Manhattan plots in the Appendix A (Supplementary Figure 13, 14).

4.4.1 Quantitative Traits

Standing Height Height is a polygenic and heritable trait that has been studied for a long time. It has been used as a model for other quantitative traits, since it is easy to measure reliably. From twin and sibling studies, the narrow sense heritability is estimated to be 70-80% (Silventoinen et al., 2003; Visscher et al., 2006, 2010). Recent estimates controlling for shared environmental factors present in twin studies calculate heritability at 0.69 (Zaitlen et al., 2013; Hemani et al., 2013). A linear based model with common SNPs explains 45% of the variance (Yang et al., 2010) and a model including imputed variants explains 56% of the variance, almost matching the estimated heritability (Yang et al., 2015). So far, GWAS studies have discovered 697 associated variants that explain one fifth of the heritability (Lango Allen et al., 2010; Wood et al., 2014). Recently, a large sample study was able to identify more variants with low frequencies that are associated with height (Marouli et al., 2017). Using lasso with the larger UK Biobank dataset allows both a better estimate of the proportion of variance that can be explained by genomic predictors and simultaneous selection of SNPs that may be associated. We obtain R^2 values that are close to the estimated heritability. The results are summarized in Table 3. The associated R^2 curves for the lasso and the relaxed lasso are shown in Figure 3. The residuals of the optimal lasso prediction are plotted in Figure 4.

Model	Form	R^2_{train}	R^2_{val}	R^2_{test}	Size
(1)	Age + Sex	0.5300	0.5260	0.5288	2
(2)	Age + Sex + 10 PCs	0.5344	0.5304	0.5336	12
(3)	Strong Single SNP	0.5364	0.5323	0.5355	13
(4)	10K Combined	0.5482	0.5408	0.5444	10,012
(5)	100K Combined	0.5833	0.5515	0.5551	100,012
(6)	Sequential LR	0.7416	0.6596	0.6601	17,012
(7)	Lasso	0.8304	0.6992	**0.6999**	47,673
(8)	Relaxed Lasso	0.7789	0.6718	0.6727	13,395

Table 3: R^2 values for height. For sequential LR, lasso and relaxed lasso, the chosen model is based on maximum R^2 on the validation set. Model (3) to (8) each includes Model (2) plus their own specification as stated in the Form column.

A large number (47,673) of SNPs need to be selected in order to achieve the optimal $R^2_{\text{test}} = 0.6992$ for the lasso. Comparatively, the relaxed lasso sacrifices some predictive performance by including a much smaller subset of variables (13,395). Past the optimal point, the additional

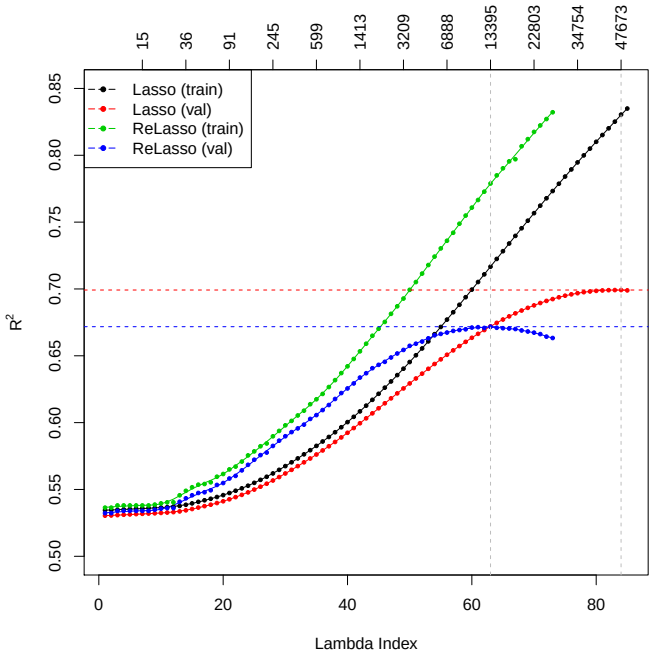

Figure 3: R^2 plot for height. The top axis shows the number of active variables in the model.

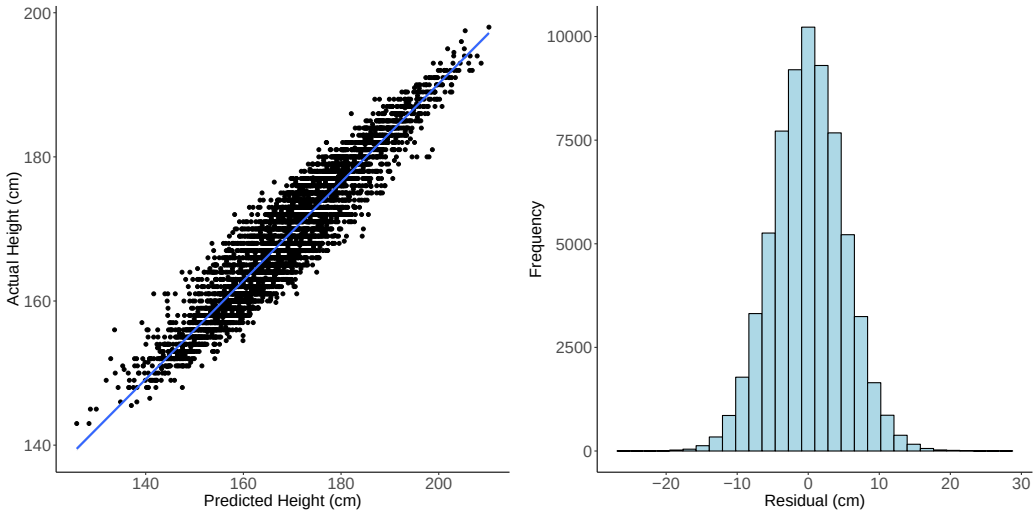

Figure 4: Left: actual height versus predicted height on 5000 random samples from the test set. The correlation between actual height and predicted height is 0.9416. Right: histogram of the lasso residuals for height. Standard deviation of the residual is 5.05 (cm).

variance introduced by refitting such large models may be larger than the reduction in bias. The large models confirm the extreme polygenicity of standing height.

Method	R^2_{val}	R^2_{test}	Cor_{test}	$\text{Cor}_{\text{test}}-\{\text{age, sex}\}$
Lasso	69.92%	69.99%	0.8366	0.4079
Prescreened lasso	69.40%	69.56%	0.8340	0.4025

Table 4: Comparison of prediction results on height with the model trained following the same procedure as ours except for an additional prescreening step as done in Lello et al. (2018). In addition to R^2, correlation between the fitted values and actual values is computed. We also compute an adjusted correlation between the residual after regressing age and sex out from the prediction and the residual after regressing age and sex out from the true response, both on the test set.

In comparison to the other models, the lasso performs significantly better in terms of R^2_{test} than all univariate methods, and outperforms multivariate methods based on univariate p-value ordering. That demonstrates the value of simultaneous variable selection and estimation from a multivariate perspective, and enables us to predict height to within 10 cm about 95% of the time based only on SNP information (together with age and sex). We also notice that the sequential linear regression approach does a good job, whose performance gets close to that of the relaxed lasso. It is straightforward and easy to implement using existing softwares such as **PLINK** (Chang et al., 2015).

Recently Lello et al. (2018) apply a lasso based method to predict height and other phenotypes on the UK Biobank. Instead of fitting on all QC-satisfied SNPs (as stated in Section 4.1), they pre-screen 50K or 100K most significant SNPs in terms of p-value and apply lasso on that set only. In addition, although both datasets come from the same UK Biobank, the subset of individuals they used is larger than ours. While we restrict the analysis to the unrelated individuals who have self-reported white British ancestry, they look at Europeans including British, Irish and Any Other White. For a fair comparison, we follow their procedure (pre-screening 100K SNPs) but run on our subset of the dataset. The results are shown in Table 4. We see that the improvement of the full lasso over the prescreened lasso is around 0.5% in the absolute sense, and 2.7% relatively if we are concerned about the gain over the baseline method consisting only of age, sex and the top 10 PCs. We would like to point out though that any improvement in the estimate close to the heritability bound becomes harder. In fact, based on twin studies on an Australian population, Macgregor et al. (2006) reported the narrow-sense heritability of human height to be approximately 0.8, and on a slightly different subset of the UK Biobank, Ge et al. (2017) reported 0.685. Those studies suggest we might already get close to the upper bound defined by narrow-sense heritability.

Further, we compare the full lasso coefficients and the univariate p-values from GWAS in Figure 5. The vertical grey dotted line indicates the top 100K cutoff in terms of p-value.

We see although a general decreasing trend appears in the magnitude of the lasso coefficients with respect to increasing p-values (decreasing $-\log_{10}(p)$), there are a number of spikes even in the large p-value region which is considered marginally insignificant. This shows that variants beyond the strongest univariate ones contribute to prediction.

Body Mass Index (BMI) BMI is another polygenic trait that is commonly studied. Like height, it is heritable and easily measured. It is also a trait of interest, since obesity is a risk factor for diseases such as type 2 diabetes and cardiovasclar disease. Recent studies estimate heritability at 0.42 (Zaitlen et al., 2013; Hemani et al., 2013) and 27% of the variance can be explained using a genomic model (Yang et al., 2015). We expect the heritability to be lower than that for height, since intuitively speaking, one component of the body mass, weight, should heavily depend on

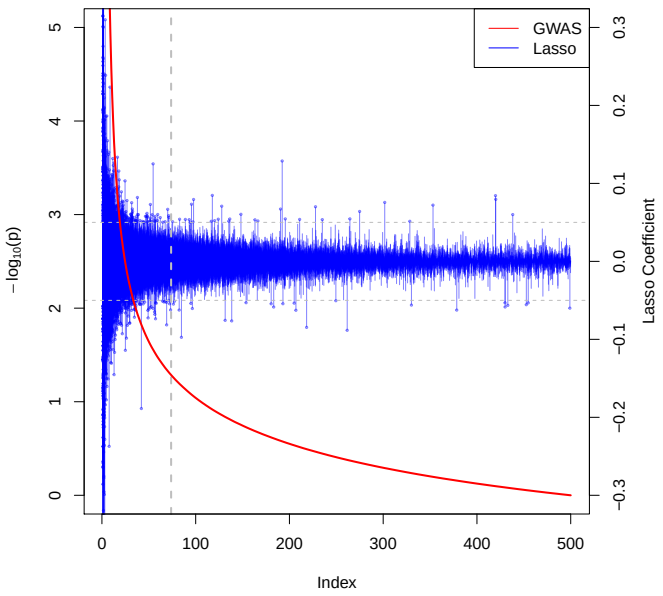

Figure 5: Comparison of the lasso coefficients and univariate p-values for height. The index on the horizontal axis represents the SNPs sorted by their univariate p-values. The red curve associated with the left vertical axis shows the $-\log_{10}$ of the univariate p-values. The blue bars associated with the right vertical axis show the corresponding lasso coefficients for each (sorted) SNP. The horizontal dotted lines in gray identifies lasso coefficients of ± 0.05. The vertical one represents the 100K cutoff used in Lello et al. (2018).

environmental factors, for example, individual's lifestyle. From GWAS studies, 97 associated loci have been identified, but they only account for 2.7% of the variance (Speliotes et al., 2010; Locke et al., 2015). Although the estimates of heritability are not precise, there may be more missing heritability for BMI than for height. We also find lower R^2 values using the lasso. The results are summarized in Table 5. The R^2 curves for the lasso and the relaxed lasso are shown in Figure 6. From the table, we see that more than 26,000 variants are selected by the lasso to attain an R^2 greater than 10%. In constrast, the relaxed lasso and the sequential linear regression use around one-tenths of the variables, and end up with degraded predictive performance both at around 5%. From Figure 7, we see further evidence that the actual BMI is of high variability and hard to predict with the lasso model — the correlation between the predicted value and the actual value is 0.3256. From the residual histogram on the right, we also see the distribution is skewed to the right, suggesting a number of exceedingly high observed values than the ones predicted by the model. Nevertheless, we are able to predict BMI within 9 kg/m^2 about 95% of the time.

4.4.2 Qualitative Traits

Asthma Asthma is a common respiratory disease characterized by inflammation of airways in the lungs and difficulty breathing. It is another complex, polygenic trait that is associated with

Model	Form	R^2_{train}	R^2_{val}	R^2_{test}	Size
(1)	Age + Sex	0.0092	0.0089	0.0083	2
(2)	Age + Sex + 10 PCs	0.0104	0.0103	0.0099	12
(3)	(2) + Single SNP	0.0134	0.0128	0.0124	13
(4)	(2) + 10K Combined	0.0384	0.0195	0.0210	10,012
(5)	(2) + 100K Combined	0.1307	0.0064	0.0093	100,012
(6)	Sequential LR	0.0865	0.0385	0.0395	2,012
(7)	Lasso	0.3196	0.1017	**0.1052**	26,060
(8)	Relaxed Lasso	0.1609	0.0504	0.0537	2,585

Table 5: R^2 values for BMI. For lasso and relaxed lasso, the chosen model is based on maximum R^2 on the validation set. Model (3) to (8) each includes Model (2) plus their own specification as stated in the Form column.

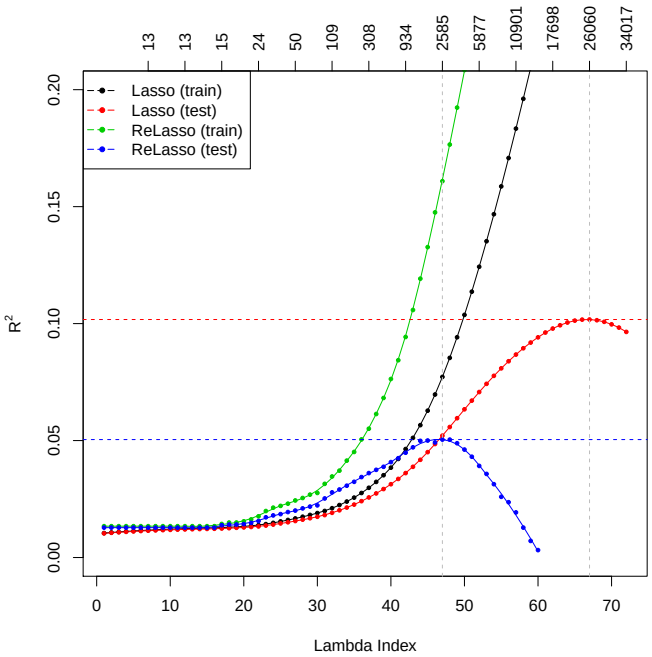

Figure 6: R^2 plot for BMI. The top axis shows the number of active variables in the model.

both genetic and environmental factors. Our results are summarized in Table 6. The AUC curves for the lasso and the relaxed lasso are shown in Figure 8. In addition, for each test sample, we compute the percentile of its predicted score/probability among the entire test cohort, and create box plots of such percentiles separately for the control group and the case group. We see on the left of Figure 9 that there is a significant overlap between the box plots of the two groups, suggesting that asthma is difficult to predict. This can also be seen from the AUC value and the ROC curve in Figure 12. That being said, the multivariate lasso still does much better than the baseline model and the strongest univariate model. On the right of Figure 9, we stratify the prediction percentile

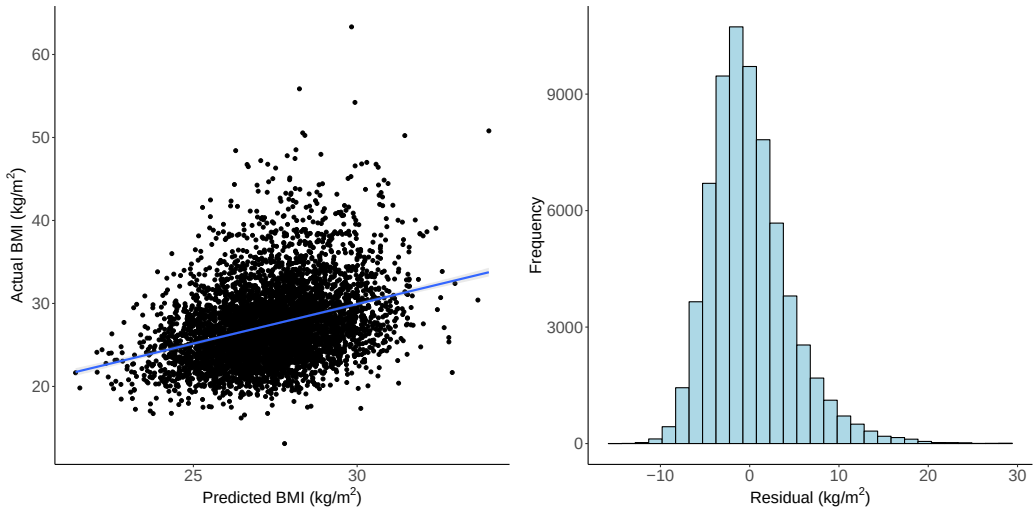

Figure 7: Left: actual BMI versus predicted BMI on 5000 random samples from the test set. The correlation between actual BMI and predicted BMI is 0.3256. Right: residuals of lasso prediction for BMI. Standard deviation of the residual is 4.51 kg/m^2.

Model	Form	$\text{AUC}_{\text{train}}$	AUC_{val}	AUC_{test}	Size
(1)	Age + Sex	0.5293	0.5297	0.5320	2
(2)	Age + Sex + 10 PCs	0.5342	0.5344	0.5367	12
(3)	(2) + Single SNP	0.5463	0.5476	0.5454	13
(4)	(2) + 10K Combined	0.5783	0.5580	0.5531	10,012
(5)	(2) + 100K Combined	0.6884	0.5644	0.5580	100,012
(6)	Sequential LR	0.6601	0.5883	0.5884	2,012
(7)	Lasso	0.7692	0.6159	**0.6126**	5,936
(8)	Relaxed Lasso	0.6747	0.5988	0.5955	621

Table 6: AUC values for asthma. For lasso and relaxed lasso, the chosen model is based on maximum AUC on the validation set. Model (3) to (8) each includes Model (2) plus their own specification as stated in the Form column.

into 10 bins, and compute the overall prevalence within each bin. We observe a clear upward trend that provides further evidence that we manage to capture some genetic signal there.

High Cholesterol High cholesterol is characterized by high amounts of cholesterol present in the blood and is a risk factor for cardiovascular disease. It is highly heritable and may be polygenic. Our results are summarized in Table 7. The AUC curves for the lasso and the relaxed lasso are shown in Figure 10. Similarly the ROC curve for the best lasso model is shown in Figure 12, and box plots for the two groups and a stratified prevalence plot are shown in Figure 11. We see that the distributions of predictions made on non-HC individuals and on HC individuals are clearly different from each other, suggesting good classification results. That is reflected in the AUC measure listed in the table. Nevertheless, it is not much better than the result of the base model including only

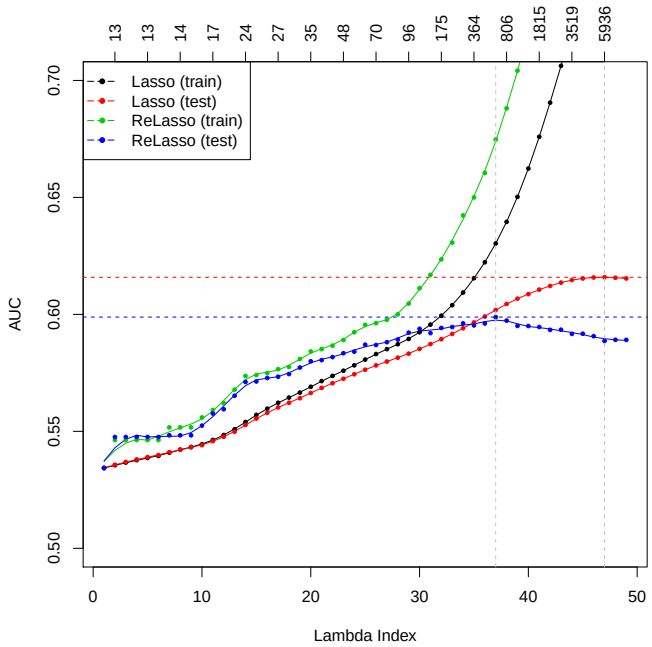

Figure 8: AUC plot for asthma. The top axis shows the number of active variables in the model.

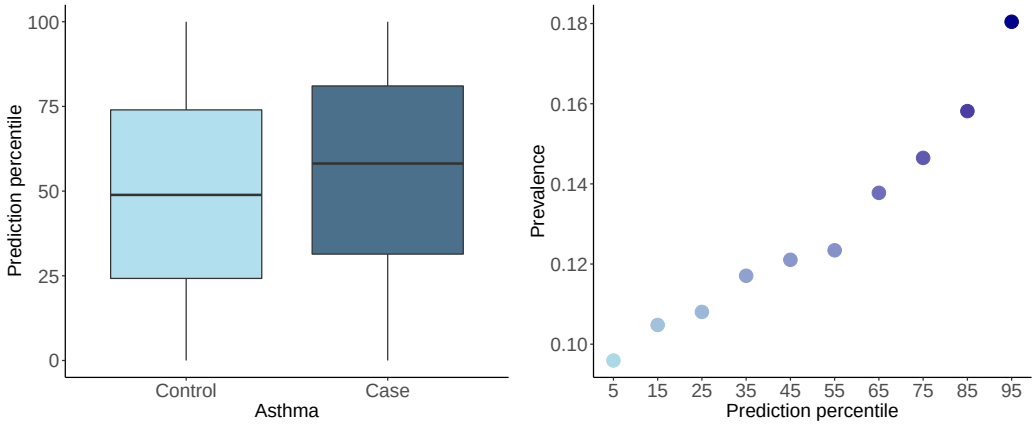

Figure 9: Results for asthma based on the best lasso model. Left: box plot of the percentile of the linear prediction score among cases versus controls. Right: the stratified prevalence across different percentile bins based on the predicted scores by the optimal lasso.

covariates age and sex.

5 Summary and discussion

Model	Form	AUC_{train}	AUC_{val}	AUC_{test}	Size
(1)	Age + Sex	0.6918	0.6952	0.6883	2
(2)	Age + Sex + 10 PCs	0.6927	0.6959	0.6889	12
(3)	(2) + Single SNP	0.6963	6982	0.6921	13
(4)	(2) + 10K Combined	0.7402	0.6956	0.6880	10,012
(5)	(2) + 100K Combined	0.8518	0.6607	0.6547	100,012
(6)	Sequential LR	0.7540	0.7167	0.7137	1,012
(7)	Lasso	0.7832	0.7259	**0.7191**	1,371
(8)	Relaxed Lasso	0.7273	0.7220	0.7166	239

Table 7: AUC values for high cholesterol. For lasso and relaxed lasso, the chosen model is based on maximum AUC on the validation set. Model (3) to (8) each includes Model (2) plus their own specification as stated in the Form column.

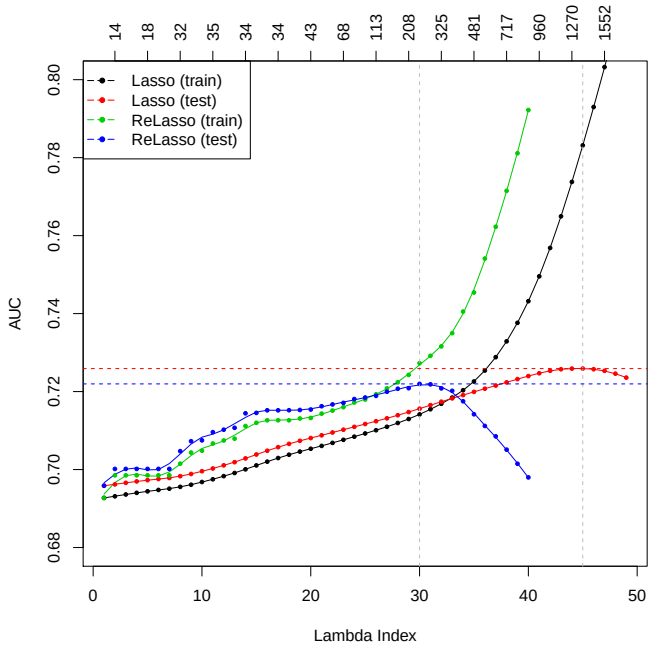

Figure 10: AUC plot for high cholesterol. The top axis shows the number of active variables in the model.

In this paper, we propose a novel batch screening iterative lasso (BASIL) algorithm to fit the full lasso solution path for very large and high-dimensional datasets. It can be used, among the others, for Gaussian linear model, logistic regression and Cox regression. It enjoys the advantages of high efficiency, flexibility and easy implementation. For SNP data as in our applications, we develop an R package **snpnet** that incorporates SNP-specific optimizations and are able to process datasets of wide interest from the UK Biobank.

Our numerical studies demonstrate that the iterative procedure effectively reduces a big-n-big-p lasso problem into one that is manageable by in-memory computation. In each iteration, we

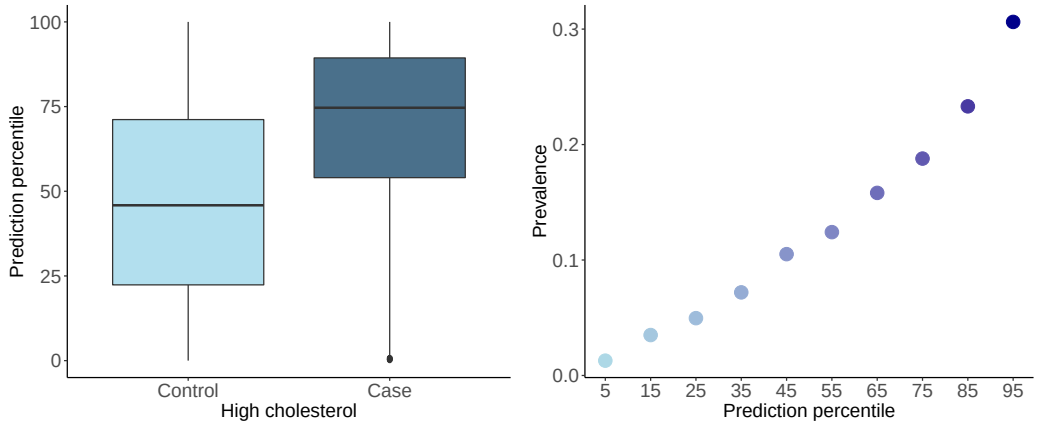

Figure 11: Results for high cholesterol based on the best lasso model. Left: box plot of the percentile of the linear prediction score among cases versus controls. Right: the stratified prevalence across different percentile bins based on the predicted scores by the optimal lasso.

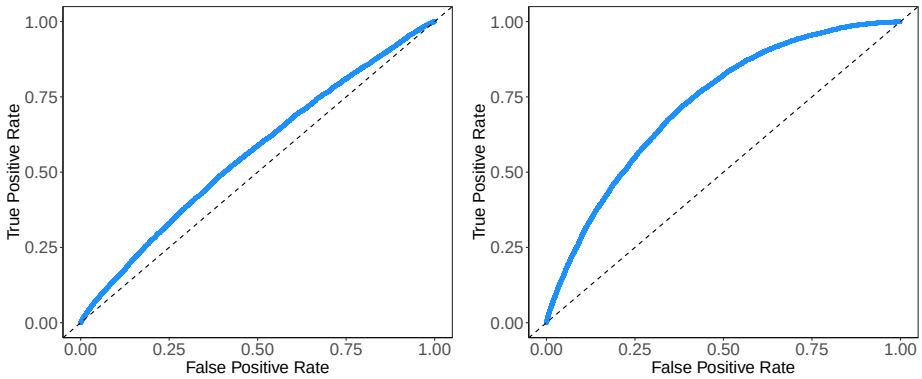

Figure 12: ROC curves. Left: asthma. Right: high cholesterol.

are able to use parallel computing when applying screening rules to filter out a large number of variables. After screening, we are left with only a small subset of data on which we are able to conduct intensive computation like cyclical coordinate descent all in memory. For the subproblem, we can use existing fast procedures for small or moderate-size lasso problems. Thus, our method allows easy reuse of previous software with lightweight development effort.

When a large number of variables is needed in the optimal predictive model, it may still require either large memory or long computation time to solve the smaller subproblem. In that case, we may consider more scalable and parallelizable methods like proximal gradient descent (Parikh and Boyd, 2014) or dual averaging (Xiao, 2010; Duchi et al., 2012). One may think why don't we directly use these methods for the original full problem? First, the ultra high dimension makes the evaluation of gradients, even on mini-batch very expensive. Second, it can take a lot more steps for such first-order methods to converge to a good objective value. Moreover, the speed of convergence depends on the choice of other parameters such as step size and additional constants in dual averaging. For those reasons, we still prefer the tuning-free and fast coordinate descent

methods when the subproblem is manageable.

Acknowledgement

We thank Balasubramanian Narasimhan for helpful discussion on the package development, Kenneth Tay, the members of the Rivas lab for insightful feedback. J.Q. is partially supported by the Two Sigma Graduate Fellowship. Y.T. is supported by Funai Overseas Scholarship from Funai Foundation for Information Technology and the Stanford University School of Medicine.

M.A.R. is supported by Stanford University and a National Institute of Health center for Multi and Trans-ethnic Mapping of Mendelian and Complex Diseases grant (5U01 HG009080). This work was supported by National Human Genome Research Institute (NHGRI) of the National Institutes of Health (NIH) under awards R01HG010140. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

R.T was partially supported by NIH grant 5R01 EB001988-16 and NSF grant 19 DMS1208164.

T.H. was partially supported by grant DMS-1407548 from the National Science Foundation, and grant 5R01 EB 001988-21 from the National Institutes of Health.

This research has been conducted using the UK Biobank Resource under application number 24983. We thank all the participants in the study. The primary and processed data used to generate the analyses presented here are available in the UK Biobank access management system (`https://amsportal.ukbiobank.ac.uk/`) for application 24983, "Generating effective therapeutic hypotheses from genomic and hospital linkage data" (`http://www.ukbiobank.ac.uk/wp-content/uploads/2017/06/24983-Dr-Manuel-Rivas.pdf`), and the results are displayed in the Global Biobank Engine (`https://biobankengine.stanford.edu`).

Some of the computing for this project was performed on the Sherlock cluster. We would like to thank Stanford University and the Stanford Research Computing Center for providing computational resources and support that contributed to these research results.

References

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246. URL `http://www.jstor.org/stable/2346178`.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent, 2010a. ISSN 1548-7660. URL `https://www.jstatsoft.org/v033/i01`.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer series in statistics. Springer-Verlag, 2009. doi: 10.1007/978-0-387-84858-7.

Bradley Efron and Trevor Hastie. *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*, volume 5. Cambridge University Press, 2016.

Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008. ISSN 0001-0782. doi: 10.1145/1327452.1327492. URL `http://doi.acm.org/10.1145/1327452.1327492`.

Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. **Spark**: Cluster computing with working sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association. URL `http://dl.acm.org/citation.cfm?id=1863103.1863113`.

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. **TensorFlow**: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, pages 265–283, Berkeley, CA, USA, 2016. USENIX Association. ISBN 978-1-931971-33-1. URL `http://dl.acm.org/citation.cfm?id=3026877.3026899`.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL `https://www.R-project.org/`.

Patrick Breheny and Jian Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics*, 5(1):232–253, 03 2011. doi: 10.1214/10-AOAS388. URL `https://doi.org/10.1214/10-AOAS388`.

Trevor Hastie. Statistical learning with big data. Presentation at Data Science at Stanford Seminar, 2015.

Clare Bycroft, Colin Freeman, Desislava Petkova, Gavin Band, Lloyd T. Elliott, Kevin Sharp, Allan Motyer, Damjan Vukcevic, Olivier Delaneau, Jared O?Connell, Adrian Cortes, Samantha Welsh, Alan Young, Mark Effingham, Gil McVean, Stephen Leslie, Naomi Allen, Peter Donnelly, and Jonathan Marchini. The uk biobank resource with deep phenotyping and genomic data. *Nature*, 562(7726):203–209, 2018. ISSN 1476-4687. doi: 10.1038/s41586-018-0579-z. URL `https://doi.org/10.1038/s41586-018-0579-z`.

Peter M. Visscher, Naomi R. Wray, Qian Zhang, Pamela Sklar, Mark I. McCarthy, Matthew A. Brown, and Jian Yang. 10 years of gwas discovery: Biology, function, and translation. *The American Journal of Human Genetics*, 101(1):5–22, 2017. ISSN 0002-9297. doi: 10.1016/j.ajhg.2017.06.005. URL `https://doi.org/10.1016/j.ajhg.2017.06.005`.

Christopher C Chang, Carson C Chow, Laurent CAM Tellier, Shashaank Vattikuti, Shaun M Purcell, and James J Lee. Second-generation PLINK: rising to the challenge of larger and richer datasets. *GigaScience*, 4(1), 02 2015. ISSN 2047-217X. doi: 10.1186/s13742-015-0047-8. URL `https://doi.org/10.1186/s13742-015-0047-8`.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

Robert Tibshirani, Jacob Bien, Jerome Friedman, Trevor Hastie, Noah Simon, Jonathan Taylor, and Ryan J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 74(2):245–266, 2012. ISSN 13697412, 14679868. URL `http://www.jstor.org/stable/41430939`.

Nicolai Meinshausen. Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1):374 – 393, 2007. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2006.12.019. URL `http://www.sciencedirect.com/science/article/pii/S0167947306004956`.

Alkes L. Price, Nick J. Patterson, Robert M. Plenge, Michael E. Weinblatt, Nancy A. Shadick, and David Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, 38:904, 2006. doi: 10.1038/ng1847. URL `https://doi.org/10.1038/ng1847`.

Michael J. Kane, John Emerson, and Stephen Weston. Scalable strategies for computing with massive data. *Journal of Statistical Software*, 55(14):1–19, 2013. URL `http://www.jstatsoft.org/v55/i14/`.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software, Articles*, 33(1):1–22, 2010b. ISSN 1548-7660. doi: 10.18637/jss.v033.i01. URL `https://www.jstatsoft.org/v033/i01`.

D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972. ISSN 00359246. URL `http://www.jstor.org/stable/2985181`.

Eric Sobel, Kenneth Lange, Tong Tong Wu, Trevor Hastie, and Yi Fang Chen. Genome-Wide Association Analysis by Lasso Penalized Logistic Regression. *Bioinformatics*, 25(6):714–721, 01 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp041. URL `https://doi.org/10.1093/bioinformatics/btp041`.

Laurent El Ghaoui, Vivian Viallon, and Tarek Rabbani. Safe feature elimination for the lasso and sparse supervised learning problems. *arXiv preprint arXiv:1009.4219*, 2010.

Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911, 2008. doi: 10.1111/j.1467-9868.2008.00674.x. URL `https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2008.00674.x`.

Jie Wang, Peter Wonka, and Jieping Ye. Lasso screening rules via dual polytope projection. *Journal of Machine Learning Research*, 16:1063–1101, 2015. URL `http://jmlr.org/papers/v16/wang15a.html`.

Yaohui Zeng and Patrick Breheny. The **biglasso** package: A memory-and computation-efficient solver for lasso model fitting with big data in R. *arXiv preprint arXiv:1701.05936*, 2017.

Florian Privé, Michael G B Blum, Hugues Aschard, and Andrey Ziyatdinov. Efficient Analysis of Large-Scale Genome-Wide Data with Two R packages: **bigstatsr** and **bigsnpr**. *Bioinformatics*, 34(16):2781–2787, 03 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty185.

Jared D Huling and Peter ZG Qian. Fast penalized regression and cross validation for tall data with the **oem** package. *arXiv preprint arXiv:1801.09661*, 2018.

Louis Lello, Steven G. Avery, Laurent Tellier, Ana I. Vazquez, Gustavo de los Campos, and Stephen D. H. Hsu. Accurate genomic prediction of human height. *Genetics*, 210(2):477–497, 2018. ISSN 0016-6731. doi: 10.1534/genetics.118.301267. URL `http://www.genetics.org/content/210/2/477`.

Christopher DeBoever, Yosuke Tanigawa, Malene E. Lindholm, Greg McInnes, Adam Lavertu, Erik Ingelsson, Chris Chang, Euan A. Ashley, Carlos D. Bustamante, Mark J. Daly, and Manuel A. Rivas. Medical relevance of protein-truncating variants across 337,205 individuals in the uk biobank study. *Nature Communications*, 9(1):1612, 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-03910-9. URL `https://doi.org/10.1038/s41467-018-03910-9`.

Yosuke Tanigawa, Jiehan Li, Johanne Marie Justesen, Heiko Horn, Matthew Aguirre, Christopher DeBoever, Chris Chang, Balasubramanian Narasimhan, Kasper Lage, Trevor Hastie, Chong Yon Park, Gill Bejerano, Erik Ingelsson, and Manuel A. Rivas. Components of genetic associations across 2,138 phenotypes in the uk biobank highlight novel adipocyte biology. *bioRxiv*, 2019. doi: 10.1101/442715. URL `https://www.biorxiv.org/content/early/2019/03/19/442715`.

Nick Patterson, Alkes L Price, and David Reich. Population structure and eigenanalysis. *PLOS Genetics*, 2(12):1–20, 12 2006. doi: 10.1371/journal.pgen.0020190. URL `https://doi.org/10.1371/journal.pgen.0020190`.

Tinca J. C. Polderman, Beben Benyamin, Christiaan A. de Leeuw, Patrick F. Sullivan, Arjen van Bochoven, Peter M. Visscher, and Danielle Posthuma. Meta-analysis of the heritability of human traits based on fifty years of twin studies. *Nature Genetics*, 47:702, 2015. doi: 10.1038/ng.3285. URL `https://doi.org/10.1038/ng.3285`.

H. D. Patterson and R. Thompson. Recovery of inter-block information when block sizes are unequal. *Biometrika*, 58(3):545–554, 1971. ISSN 00063444. URL `http://www.jstor.org/stable/2334389`.

Jian Yang, Beben Benyamin, Brian P. McEvoy, Scott Gordon, Anjali K. Henders, Dale R. Nyholt, Pamela A. Madden, Andrew C. Heath, Nicholas G. Martin, Grant W. Montgomery, Michael E. Goddard, and Peter M. Visscher. Common snps explain a large proportion of the heritability for human height. *Nature Genetics*, 42:565, 2010. doi: 10.1038/ng.608. URL `https://doi.org/10.1038/ng.608`.

Jian Yang, S. Hong Lee, Michael E. Goddard, and Peter M. Visscher. Gcta: A tool for genome-wide complex trait analysis. *The American Journal of Human Genetics*, 88(1):76–82, 2011. ISSN 0002-9297. doi: 10.1016/j.ajhg.2010.11.011. URL `https://doi.org/10.1016/j.ajhg.2010.11.011`.

Peter M. Visscher, William G. Hill, and Naomi R. Wray. Heritability in the genomics era ? concepts and misconceptions. *Nature Reviews Genetics*, 9:255, 2008. doi: 10.1038/nrg2322. URL `https://doi.org/10.1038/nrg2322`.

SangăHong Lee, NaomiăR Wray, MichaelăE Goddard, and PeterăM Visscher. Estimating missing heritability for disease from genome-wide association studies. *The American Journal of Human Genetics*, 88(3):294–305, 2011. ISSN 0002-9297. doi: 10.1016/j.ajhg.2011.02.002. URL `https://doi.org/10.1016/j.ajhg.2011.02.002`.

Herman Wold. Soft modelling by latent variables: The non-linear iterative partial least squares (nipals) approach. *Journal of Applied Probability*, 12(S1):117?142, 1975. doi: 10.1017/S0021900200047604.

Karri Silventoinen, Sampo Sammalisto, Markus Perola, Dorret I. Boomsma, Belinda K. Cornes, Chayna Davis, Leo Dunkel, Marlies de Lange, Jennifer R. Harris, Jacob V.B. Hjelmborg, and

et al. Heritability of adult body height: A comparative study of twin cohorts in eight countries. *Twin Research*, 6(5):399?408, 2003. doi: 10.1375/twin.6.5.399.

Peter M Visscher, Sarah E Medland, Manuel A. R Ferreira, Katherine I Morley, Gu Zhu, Belinda K Cornes, Grant W Montgomery, and Nicholas G Martin. Assumption-free estimation of heritability from genome-wide identity-by-descent sharing between full siblings. *PLOS Genetics*, 2(3):1–10, 03 2006. doi: 10.1371/journal.pgen.0020041. URL `https://doi.org/10.1371/journal.pgen.0020041`.

Peter M. Visscher, Brian McEvoy, and Jian Yang. From galton to gwas: Quantitative genetics of human height. *Genetics Research*, 92(5-6):371?379, 2010. doi: 10.1017/S0016672310000571.

Noah Zaitlen, Peter Kraft, Nick Patterson, Bogdan Pasaniuc, Gaurav Bhatia, Samuela Pollack, and Alkes L. Price. Using extended genealogy to estimate components of heritability for 23 quantitative and dichotomous traits. *PLOS Genetics*, 9(5):1–11, 05 2013. doi: 10.1371/journal.pgen.1003520. URL `https://doi.org/10.1371/journal.pgen.1003520`.

Gibran Hemani, Jian Yang, Anna Vinkhuyzen, JosephăE Powell, Gonneke Willemsen, Jouke-Jan Hottenga, Abdel Abdellaoui, Massimo Mangino, AnaăM Valdes, SarahăE Medland, PamelaăA Madden, AndrewăC Heath, AnjaliăK Henders, DaleăR Nyholt, EcoăJ C. deăGeus, PatrikăK E. Magnusson, Erik Ingelsson, GrantăW Montgomery, TimothyăD Spector, DorretăI Boomsma, NancyăL Pedersen, NicholasăG Martin, and PeterăM Visscher. Inference of the genetic architecture underlying bmi and height with the use of 20,240 sibling pairs. *The American Journal of Human Genetics*, 93(5):865–875, 2013. ISSN 0002-9297. doi: 10.1016/j.ajhg.2013.10.005. URL `https://doi.org/10.1016/j.ajhg.2013.10.005`.

Jian Yang, Andrew Bakshi, Zhihong Zhu, Gibran Hemani, Anna A. E. Vinkhuyzen, Sang Hong Lee, et al. Genetic variance estimation with imputed variants finds negligible missing heritability for human height and body mass index. *Nature Genetics*, 47:1114, 2015. doi: 10.1038/ng.3390. URL `https://doi.org/10.1038/ng.3390`.

Hana Lango Allen, Karol Estrada, Guillaume Lettre, Sonja I. Berndt, Michael N. Weedon, Fernando Rivadeneira, et al. Hundreds of variants clustered in genomic loci and biological pathways affect human height. *Nature*, 467:832, 2010. doi: 10.1038/nature09410. URL `https://doi.org/10.1038/nature09410`.

Andrew R. Wood, Tonu Esko, Jian Yang, Sailaja Vedantam, Tune H. Pers, Stefan Gustafsson, et al. Defining the role of common variation in the genomic and biological architecture of adult human height. *Nature Genetics*, 46:1173, 2014. doi: 10.1038/ng.3097. URL `https://doi.org/10.1038/ng.3097`.

Eirini Marouli, Mariaelisa Graff, Carolina Medina-Gomez, Ken Sin Lo, Andrew R. Wood, Troels R. Kjaer, et al. Rare and low-frequency coding variants alter human adult height. *Nature*, 542:186, 2017. doi: 10.1038/nature21039. URL `https://doi.org/10.1038/nature21039`.

Stuart Macgregor, Belinda K. Cornes, Nicholas G. Martin, and Peter M. Visscher. Bias, precision and heritability of self-reported and clinically measured height in australian twins. *Human Genetics*, 120(4):571–580, Nov 2006. ISSN 1432-1203. doi: 10.1007/s00439-006-0240-z. URL `https://doi.org/10.1007/s00439-006-0240-z`.

Tian Ge, Chia-Yen Chen, Benjamin M. Neale, Mert R. Sabuncu, and Jordan W. Smoller. Phenome-wide heritability analysis of the uk biobank. *PLOS Genetics*, 13(4):1–21, 04 2017. doi: 10.1371/journal.pgen.1006711. URL `https://doi.org/10.1371/journal.pgen.1006711`.

Elizabeth K. Speliotes, Cristen J. Willer, Sonja I. Berndt, Keri L. Monda, Gudmar Thorleifsson, Anne U. Jackson, et al. Association analyses of 249,796 individuals reveal 18 new loci associated with body mass index. *Nature Genetics*, 42:937, 2010. doi: 10.1038/ng.686. URL `https://doi.org/10.1038/ng.686`.

Adam E. Locke, Bratati Kahali, Sonja I. Berndt, Anne E. Justice, Tune H. Pers, Felix R. Day, Corey Powell, et al. Genetic studies of body mass index yield new insights for obesity biology. *Nature*, 518:197, 2015. doi: 10.1038/nature14177. URL `https://doi.org/10.1038/nature14177`.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1 (3):127–239, January 2014. ISSN 2167-3888. doi: 10.1561/2400000003. URL `http://dx.doi.org.stanford.idm.oclc.org/10.1561/2400000003`.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57(3): 592–606, March 2012. ISSN 0018-9286. doi: 10.1109/TAC.2011.2161027.

Stephen D. Turner. **qqman**: An R package for visualizing gwas results using q-q and manhattan plots. *Journal of Open Source Software*, 3(25):731, 2018. doi: 10.21105/joss.00731.

A Manhattan Plots

The Manhattan plots in Figure 13 (generated using the **qqman** package (Turner, 2018)) show the magnitude of the univariate p-values and the size of the lasso coefficients for each gene for the two quantitative traits and two binary traits. The coefficients are plotted for the model with the optimal R^2 value on the validation set. The variants highlighted in red in both plots are those that have coefficient magnitudes above the 99th percentile of all coefficient magnitudes for the trait. The horizontal line in the p-value plot is plotted at the genome-wide Bonferroni corrected p-value threshold 5×10^{-8}. There are two main points we would like to highlight:

- The lasso manages to capture significant univariate predictors in each genetic region. Due to possible correlation it does not pick up the variants with similarly small p-values located nearby.
- Some of the variants with weak univariate signals are also identified and turn out to be crucial to the predictive performance of the lasso.

For the two qualitative traits plotted in Figure 14, there are fewer p-values above the threshold, and many of the significant ones are located close to each other. The size of the lasso fit is correspondingly smaller, and the large coefficients pick up the important locations as before. However, the nonzero coefficients are still spread across the whole genome.

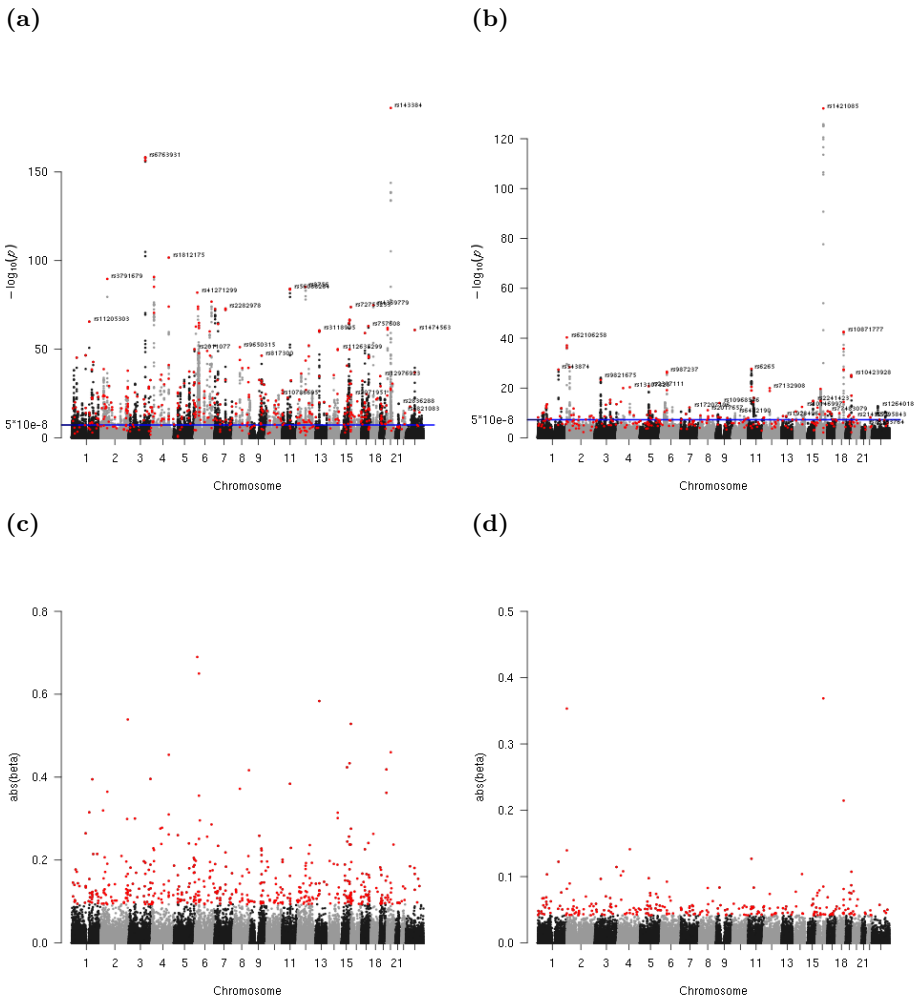

Figure 13: Manhattan plots of the univariate p-values and lasso coefficients for height (a, c) and BMI (b, d). The vertical axis of the p-value plots shows $-\log_{10}(p)$ for each SNP, while the vertical axis of the coefficient plots shows the magnitude of the coefficients from **snpnet**. The SNPs with relatively large lasso coefficients are highlighted in red. The blue horizontal line on the p-value plot represents a reference level of $p = 5 \times 10^{-8}$.

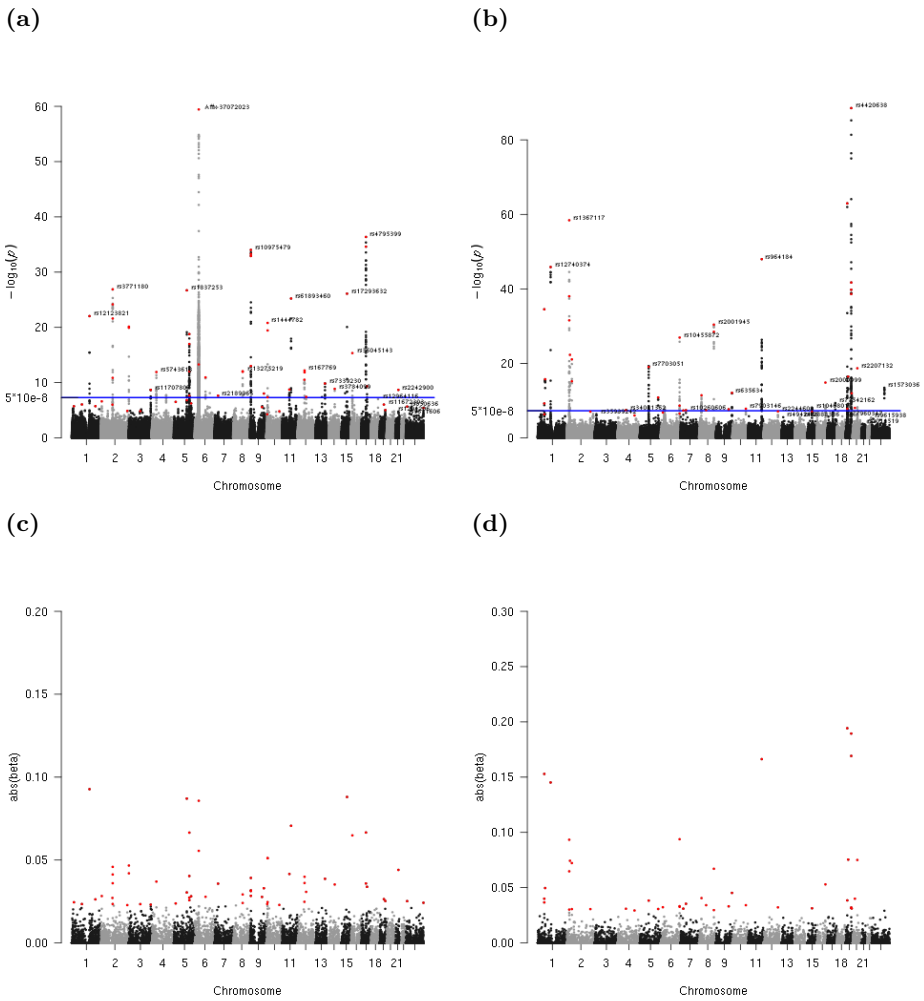

Figure 14: Manhattan plots of the univariate p-values and lasso coefficients for asthma (a, c) and high cholesterol (b, d). The vertical axis of the p-value plots shows $-\log_{10}(p)$ for each SNP, while the vertical axis of the coefficient plots shows the magnitude of the coefficients from **snpnet**. The SNPs with relatively large lasso coefficients are highlighted in red. The blue horizontal line on the p-value plot represents a reference level of $p = 5 \times 10^{-8}$.