Fast Sparse-Group Lasso Method for Multi-response Cox Model with Applications to UK Biobank

Ruilin Li*[1], Yosuke Tanigawa[2], Johanne M. Justesen[2], Jonathan Taylor[3], Trevor Hastie[2,3], Robert Tibshirani[2,3] and Manuel A. Rivas†[2]

[1]Institute for Computational and Mathematical Engineering, Stanford University
[2]Department of Biomedical Data Science, Stanford University
[3]Department of Statistics, Stanford University

Abstract

We propose a Sparse-Group regularized Cox regression method to analyze large-scale, ultrahigh-dimensional, and multi-response survival data efficiently. Our method has three key components: 1. A Sparse-Group penalty that encourages the coefficients to have small and overlapping support; 2. A variable screening procedure that minimizes the frequency of disk memory access when the data does not fit in the RAM; 3. An accelerated proximal gradient method that optimizes the regularized partial-likelihood function. To demonstrate the efficacy of our method, we implemented the proposed algorithm for human genetics data and applied it to 16 time-to-event phenotypes from the UK Biobank.

1 Introduction

Large scale, ultrahigh-dimensional datasets with numerous time-to-event responses have become increasingly prevalent in recent years. The UK Biobank dataset (Sudlow et al. 2015) contains millions of genetic variants and thousands of survival phenotypes from each of its over $500,000$ participants. Such datasets pose statistical and computational challenges to classical survival models. The statistical challenge lies in the high-dimensionality. When the number of predictors is larger than the number of observations, the association between the predictors and the response becomes unidentifiable without addition assumptions. The computational challenge lies in the overall size of the dataset. Reading the feature matrix of the UK Biobank dataset into R requires more than 4 Terabytes of memory, which is much larger than the RAM size of most computers. While memory mapping (Kane et al. 2013) allows users to access out-of-memory data with ease, it requires lots of disk Input/Output operations, which is much slower than in-memory operation. This becomes even more problematic for iterative optimization algorithms that use the entire feature matrix every iteration. Moreover, to date, no approaches are able to handle the scale of the data for multiple responses, i.e. more than one time-to-event outcome, which is of interest when considering time-to-event disease responses having shared genetic effects. Here, we present a multiresponse Cox model formulation with a fast sparse-group lasso solution implemented in a publicly available package `Multi-snpnet-Cox`, `mrcox` for short, available at `https://github.com/rivas-lab/multisnpnet-Cox_v1` that operates on top of `PLINK2` binary file formats and integrates the C-index algorithm presented in Li et al. (2020).

*Corresponding author: ruilinli@stanford.edu
†Corresponding author: mrivas@stanford.edu

1.1 Cox Proportional Hazard Model

Cox model (Cox 1972) provides a flexible mathematical framework that describes the relationship between the predictors and a time-to-event response. For each individual we observe a triple $\{O, X, T\}$. $X \in \mathbb{R}^d$ are the features. $O \in \{0, 1\}$ is the status indicator. If $O = 1$, then T is the actual time-to-event. If $O = 0$, then we only know that the time-to-event is at least T. The hazard function according to the Cox model can be written as

$$h(t|X) = h_0(t) \exp(X^T \beta),$$

where $\beta \in \mathbb{R}^d$ is the coefficients vector that measures the strength of association between X and the response. This hazard function is equivalent to the cumulative distribution function:

$$P(T \leq t|X) = 1 - \exp\left(-\int_0^t h_0(s) e^{\beta^T X} ds\right).$$

Here $h_0 : \mathbb{R}^+ \mapsto \mathbb{R}^+$ is the baseline hazard function. In our applications we are interested in the association between genetic variants and the response, so the baseline hazard function is a nuisance variable. Fortunately we can estimate the parameters β without knowing the baseline hazard and achieve almost the same estimation accuracy as when the baseline hazard is known. This is done by maximizing the partial likelihood function (Cox 1972):

$$PL(\beta|Data) = \prod_{i:O_i=1} \frac{\exp(X_i^T \beta_k)}{\sum_{j:T_j \geq T_i} \exp(X_j^T \beta)}.$$

1.2 Sparse-Group Lasso

The Lasso method (Tibshirani 1996) makes the assumption that only a small subset of predictors are associated with the response. In other words, it assumes the model has a sparsity structure. This assumption makes Lasso very effective for high-dimensional data. A sparse solution can be obtained by adding the penalty $\lambda\|\beta\|_1$ to the original objective function, for an appropriately chosen regularization parameter $\lambda > 0$ (usually through cross validation).

Sparse-Group Lasso (Simon et al. 2013) is a variation of Lasso. It assumes not only that many individual elements of β are 0, but also that within some predefined groups of variables, the corresponding parameters are 0 simultaneously. For a group $g \subseteq \{1, 2, \cdots, d\}$ of variables that we believe the coefficients are 0 together, we add an additional penalty $\lambda_g \|\beta_g\|_2$.

2 Method

2.1 Preliminaries

In this section we define the notations and the key model assumptions that we will use in the subsequent sections. For an integer n, define $[n] = \{1, 2, \cdots, n\}$ and define $x_+ = \max\{x, 0\}$ for all $x \in \mathbb{R}$.

We analyze $K \geq 1$ time-to-event responses on n individuals. For example, the responses could be time from birth to K different diseases. The data we observed are in the format:

$$\mathcal{D} = \{X_i, T_i^1, \cdots, T_i^K, O_i^1, \cdots, O_i^K\}_{i=1}^n.$$

Here $X_i \in \mathbb{R}^d$ are ith individual's features. Denote the full features matrix $\boldsymbol{X} = [X_1, X_2, \cdots, X_n]^T \in \mathbb{R}^{n \times d}$. For $k = 1, \cdots, K$, $O_i^k = 1$ implies that T_i^k is the true time of the event k for the ith individual,

and $O_i^k = 0$ implies that the true time of the event k is right-censored at T_i^k. We assume each response follows a Cox proportional hazard model:

$$h_k(t|X) = h_{0,k}(t)\exp(X^T\beta_k). \tag{1}$$

where $h_{0,k} : \mathbb{R}^+ \mapsto \mathbb{R}^+$ is the baseline hazard function of the kth response. Let

$$n_k = \sum_{i=1}^{n} O_i^k$$

be the number of observed event k.

We make the assumption that not only β_k is sparse for all $k \in [K]$, but they also have a small common support. That is $\{j \in [p] : \beta_k^j \neq 0, k \in [K]\}$ is a small set relative to p. In human genetics applications, the first assumption means that each response is associated with a small set of genetic variants, and the second assumption implies that there are significant overlap among the genetic variants that are associated with each response. This belief translates to the following regularized partial likelihood objective function:

$$\min_{\beta_1,\cdots,\beta_K} \sum_{k=1}^{K} \frac{1}{n_k}\left[\sum_{i:O_i^k=1} -\beta_k^T X_i + \log\left(\sum_{j:T_j^k \geq T_i^k} \exp(\beta_k^T X_j)\right)\right] + \lambda\left(\sum_{j=1}^{d} \|\beta^j\|_1 + \alpha\|\beta^j\|_2\right). \tag{2}$$

Here the first term is the sum of the K negative log-partial-likelihood, normalized by the number of observed events. The second term is the regularization. $\|\beta^j\|_1$, $\|\beta^j\|_2$ are the 1-norm and 2-norm of the coefficients for the jth variable. That is, if we put all the parameters into a matrix $B = [\beta_1, \beta_2, \cdots, \beta_K] \in \mathbb{R}^{d\times K}$, then β^j is the jth row of B and β_k is the kth column. Note that when $\alpha = 0$, the objective function decouples for each β_k and they can be optimized separately. In our implementation we solve a slightly more general problem:

$$\min_{\beta_1,\cdots,\beta_K} \left\{\sum_{k=1}^{K} \frac{1}{n_k}\left[\sum_{i:O_i^k=1} -\beta_k^T X_i + \log\left(\sum_{j:T_j^k \geq T_i^k} \exp(\beta_k^T X_j)\right)\right] + \lambda\left[\sum_{j=1}^{d} w_j(\|\beta^j\|_1 + \alpha\|\beta^j\|_2)\right]\right\},$$

where $\{w_j\}_{j=1}^d$ are user provided penalty factor for each variables, which may be useful in the setting where protein-truncating or protein-altering variants should be prioritized (Rivas et al. 2015, DeBoever et al. 2018). In our implementation, the hyperparameter α is fixed at $\sqrt{K}$, and the solution is computed on a pre-defined sequence of λ: $\lambda_1 > \lambda_2 > \cdots > \lambda_L$. In our implementation, λ_1 is chosen so that the parameters just become non-zero.

2.2 Variable Screening for Lasso Path

Before explaining how we optimize the objective function (2), we first describe a variable screening procedure that utilizes the sparsity structure of the solution to significantly reduce the number of variables needed for fitting. The main advantage of variable screening is to decrease the frequency of operations done on the full features matrix $\boldsymbol{X}$. Our procedure is similar to the strong rule (Tibshirani et al. 2012) and the Batch Screening Iterative Lasso (Qian et al. 2019).

To simplify the notation, for $j \in [d]$ denote

$$g_j = g_j(\beta_1, \cdots, \beta_k) := \frac{\partial}{\partial\beta^j}\sum_{k=1}^{K} \frac{1}{n_k}\left[\sum_{i:O_i^k=1} -\beta_k^T X_i + \log\left(\sum_{j:T_j^k \geq T_i^k} \exp(\beta_k^T X_j)\right)\right]. \tag{3}$$

Here $g_j \in \mathbb{R}^K$ is the partial derivative of the smooth part of (2) with respect to the coefficients of the variable j. For each $\alpha > 0, v \in \mathbb{R}^K$, define

$$\|v\|_{\alpha*} := \sup\{u^T v : u \in \mathbb{R}^K, \|u\|_1 + \alpha\|u\|_2 \leq 1\}. \tag{4}$$

Here we use the following result to get an optimality condition for the solutions $\beta_1, \cdots, \beta_K$. The proofs are given in the appendix.

Proposition 1. *For any $\lambda > 0$, the gradient at the optimal solution to (2) satisfies:*

$$\|g_j\|_{\alpha*} \begin{cases} \leq \lambda & \text{if the optimal } \beta^j = 0 \\ = \lambda & \text{if the optimal } \beta^j \neq 0 \end{cases} \text{ for all } j \in [d]. \tag{5}$$

This result motivates us to first fit a model (solving (2)) using a small number of variables whose gradient has the largest $\alpha*$-norm, assuming the coefficients for the rest of the variables are all zero. Then to verify the validity of the assumption we check that $\|g_j\|_{\alpha*} \leq \lambda$ for variables assumed to have zero coefficients. We refer to this step as KKT checking. Note that based on its definition (4), it's not clear how we can compute $\|v\|_{\alpha*}$. Here we give a more explicit characterization.

Proposition 2. *Let $S_1(\cdot;\lambda) : \mathbb{R}^K \mapsto \mathbb{R}^K$ be the element-wise soft-thresholding function:*

$$S_1(v;\lambda)_k = sign(v_k)(|v_k| - \lambda)_+ \text{ for all } k \in [K]$$

Then $\|v\|_{\alpha*} \leq \lambda$ if and only if

$$\|S_1(v;\lambda)\|_2 \leq \alpha\lambda.$$

Using the above we can check if $\|v\|_{\alpha*} \leq \lambda$ quite easily and compute $\|v\|_{\alpha*}$ using binary search in $\mathcal{O}(K \log \|v\|_\infty)$ to any fixed precision (since $\|v\|_{\alpha*} \leq \|v\|_\infty$).

Now we are ready to state the overall structure of our algorithm. Suppose valid solutions for $\lambda_1, \cdots, \lambda_l$ have been obtained. Next we do the following steps:

1. (**Screening**) In the last iteration, we cache the full gradient $\{g^j\}_{j=1}^d$ evaluated at the solution at λ_l. In the fitting step we include two types of variabels:
   - We include the ever-active variables $\mathcal{A} := \{j \in [d] : \hat{\beta}^j \neq 0 \text{ for any previously obtained } \hat{\beta}\}$.
   - Top M variables with the largest $\|g_j\|_{\alpha*}$ that are also not ever-active.

   We denote the set of variables used to fit (2) as the strong set $\mathcal{S} \subseteq [d]$.

2. (**Fitting**) In this step, we solve the problem (2) for the next few λ using only variables in $\mathcal{S}$, assuming $\beta^j = 0$ for all $j \notin \mathcal{S}$. The optimization algorithm used for fitting step is described in the next section. To speed-up the computation we initilize the variables at the previous valid solution (warm start).

3. (**KKT Checking**) After obtaining the solution from the fitting step. We compute the full gradient. This is the only step we will need the full data matrix $\boldsymbol{X}$. We check that the KKT condition (5) are satisfied for all variables. We go back to the screening step at the first λ value where the KKT condition fails. We also cache the full gradient at the last valid solutions for the screening step.

4. (**Early Stopping**) We keep a separate validation set to evaluate the current estimated parameters. We choose the optimal λ as the one that gives the highest validation concordance index (C-index) (Harrell et al. 1982, Li et al. 2020). The optimal λ might be different for different responses. Once the validation C-index for response k starts to decrease we stop fitting that response and freeze the value β_k at its last iteration. Once the validation C-index starts to decrease for all K responses. We stop the entire procedure.

These steps are described in algorithm 1.

2.3 Optimization Method

We use a Nesterov-accelerated (Nesterov 1983) proximal gradient method (Daubechies et al. 2004, Beck & Teboulle 2009) to optimize the objective function (2). Proximal gradient algorithm is particularly suitable when the objective function is the sum of a smooth function and a simple function. In our case the smooth function is the sum of the negative log-partial-likelihood functions, and the simple function is the regularization term. The algorithm alternates between two steps until convergence criteria is met:

1. A gradient step that tries to minimize the smooth part of the objective:

$$\beta^j \leftarrow \beta^j - t g_j,$$

where t is the step size that we determine using backtracking line search.

2. A proximal step that tries to keep the regularization term small:

$$\beta^j \leftarrow prox_t(\beta^j).$$

Here the proximal operator $prox_t : \mathbb{R}^K \mapsto \mathbb{R}^K$ is defined as

$$prox_t(x) := \arg\min_z \frac{1}{2t}\|z - x\|_2^2 + \lambda\|z\|_1 + \lambda\alpha\|z\|_2.$$

To simplify the notation we omit the dependency of $prox_t(x)$ on λ, α. Simple calculation shows that

$$prox_t(x) = S_2(S_1(x; t\lambda); t\alpha\lambda).$$

where $S_2(\cdot; t\alpha\lambda) : \mathbb{R}^K \mapsto \mathbb{R}^K$ is a group soft-thresholding operator:

$$S_2(v; t\alpha\lambda) = \begin{cases} 0 & \text{if } \|v\|_2 \le t\alpha\lambda \\ \frac{v}{\|v\|_2}(\|v\|_2 - t\alpha\lambda) & \text{otherwise} \end{cases}.$$

We describe the optimization algorithm in psuedocode, including details about Nesterov acceleration and backtracking line search in algorithm 2.

3 Application to 16 time-to-event responses in UK Biobank

We applied the proposed method to 16 time-to-event responses described in table 1. The 16 phenotypes are first occurrences from UK Biobank (Category 2404) corresponding to Endocrine, nutritional and metabolic diseases. The responses were coded with time of event equal to the age of the individual at first occurrence and the status of the individual as censored if the individual has either died without any coding of the disease or the individual does not have a code for the disease at the latest data refresh.

Algorithm 1: Iterative Screening for Lasso Path

Initialize ever active set $\mathcal{A}^{(0)} = \emptyset$;
Construct the regularization parameters $\lambda_1, \cdots, \lambda_L$;
Initialize a short list of initial regularization parameters $\Lambda^{(0)} = \{\lambda_1, \cdots, \lambda_{L^{(0)}}\}$;
Initialize the parameters $\hat{\beta}_1^{(0)}, \cdots, \hat{\beta}_K^{(0)} = 0$;
Set the iteration counter $i = 0$;
while $\hat{\beta}(\lambda_L)$ *not computed* **do**
  Set v to be the largest number such that the solutions for $\lambda = \lambda_1, \cdots, \lambda_v$ have been obtained;
  Screening:
  Compute (or use the cached) full gradient (3) $g_1, \cdots, g_d$. These gradients are evaluated at $\hat{\beta}_1^{(v)}, \cdots, \hat{\beta}_K^{(v)}$;
  Set $\mathcal{E}_M^{(i)}$ to be the M variables in $[d] \setminus \mathcal{A}^{(i)}$ with the largest $\|g_j\|_{\alpha*}$;
  Set $\mathcal{S}^{(i)} = \mathcal{A}^{(i)} \cup \mathcal{E}_M^{(i)}$. This will be the variables used in the fitting step;
  Fitting:
  for $\lambda \in \Lambda^{(i)}$ **do**
    Obtain parameter estimates by solving (2) using only variables in $\mathcal{S}^{(i)}$;
    The optimization algorithm is described in algorithm 2;
    Coefficients for variables not in $\mathcal{S}^{(i)}$ are set to be zero.;
  end
  Checking:
  Compute the full gradients $g_1(\lambda), \cdots, g_d(\lambda)$ at solutions obtained with regularization parameters $\lambda \in \Lambda^{(i)}$;
  Find the smallest $\lambda \in \Lambda^{(k)}$ such that the KKT condition (5) is satisfied:

$$\overline{\lambda}^{(i)} = \min \left\{ \lambda \in \Lambda^{(i)} : \max_{j \in [d] \setminus \mathcal{S}^{(i)}} \|g_j(\lambda)\|_{\alpha*} < \lambda \right\}$$

  Update ever active set $\mathcal{A}^{(i+1)} = \mathcal{A}^{(i)} \cup \{j : \hat{\beta}^j(\overline{\lambda}^{(i)}) \neq 0\}$;
  Update $\Lambda^{(i+1)} = \{\lambda \in \Lambda^{(i)} : \lambda < \overline{\lambda}^{(i)}\}$. Extend $\Lambda^{(i+1)}$ if it is too short.;
  For $\lambda \in \{\lambda \in \Lambda^{(i)} : \lambda \geq \overline{\lambda}^{(i)}\}$, we obtain valid solutions;
  Set $i = i + 1$;
end
return $\hat{\beta}(\lambda_1), \cdots, \hat{\beta}(\lambda_L)$

Algorithm 2: Proximal Gradient Method for (2)

Let $\mathcal{S} \subseteq [d]$ be the set of variables we use to optimize (2). β^j is assumed to be 0 for $j \notin \mathcal{S}$;
Let $p = |\mathcal{S}|$ be the number of variables that are assumed to be non-zero;
Set line search parameter $\eta > 1$;
Write the parameter matrix $B \in \mathbb{R}^{p \times K}$. The rows of B are β^j for $j \in \mathcal{S}$;
Write the smooth part of the objective as:

$$f(B) = \sum_{k=1}^{K} \frac{1}{n_k} \left[\sum_{i:O_i^k=1} -\beta_k^T X_i + \log \left(\sum_{j:T_j^k \geq T_i^k} \exp(\beta_k^T X_j) \right) \right].$$

Initialize $B^{(0)}$ at a user-specified value, or set $B^{(0)} = 0$;
Set iteration count $i = 0$;
Set initial step-size $t = 1$;
Set Nestrov weights $w_0, w_1 = 1$;
while B *has not converged* **do**
  Nesterov acceleration: $w_1 \leftarrow (1 + \sqrt{1 + 4w_0^2})/2$;
  $B^{(i+0.5)} \leftarrow B^{(i)} + (w_0 - 1)(B^{(i)} - B^{(i-1)})/w_1$;
  $w_0 \leftarrow w_1$;
  Compute the gradient g_j:

$$g_j = \frac{\partial}{\partial \beta^j} f(B^{(i+0.5)}), \text{ for } j \in \mathcal{S}.$$

  Start backtracking line search:
  repeat
    $B \leftarrow B^{(i+0.5)} - t \nabla f(B^{(i+0.5)})$;
    Denote β^j the jth row of B for $j \in \mathcal{S}$;
    Apply proximal step: $\beta^j \leftarrow prox_t(\beta^j)$ for $j \in \mathcal{S}$;
    if $f(B) \leq f(B^{(i+0.5)}) + \langle B - B^{(i+0.5)}, \nabla f(B^{(i+0.5)}) \rangle + \|B - B^{(i+0.5)}\|_2^2/(2t)$ **then**
      break;
    end
    $t \leftarrow t/\eta$;
  until *Appropriate step-size* t *obtained*;
  $B^{(i+1)} \leftarrow B$;
  $i \leftarrow i + 1$;
  Check convergence based on objective value change or parameter change.
end
return $B^{(i)}$

Overall, we find that multiSnpnet-Cox is able to improve prediction across a large number of diseases (in the application in this study 16, Table 1). The sparse solution for the 16 responses range from hundreds to tens of thousands of active variables (Figure 1). In addition, analogous to our proposal in Qian et al. (2020), there is an alternative way to find a low-rank coefficient profile for regression. Instead of pursing to solve a non-convex problem directly, we can follow a two-stage procedure: 1) solve a full-rank regression, and 2) conduct SVD of the resulting coefficient matrix and use its rank approximation as our final estimator. We referred to this approach as the lazy evaluation model and similarly present the decomposition of multiSnpnet-Cox model coefficients in Figure 2 for other hypothyroidism. From the biplot we find that several of the genetic effects are shared with `130706`, corresponding to insulin-dependent diabetes mellitus, and orthogonal to genetic effects for `130736`, corresponding to ovarian dysfunction.

For an individual with genotype x, we define the Polygenic Hazard Score (PHS) to be $\hat{\beta}^T x$, where $\hat{\beta}$ is the fitted regression coefficients obtained from multiSnpnet-Cox. We assess the predictive power of PHS on survival time using the individuals in the held out test set. We applied a couple of procedures to give a high level overview of the results. First, we assessed whether the PHS was significantly associated to the time to event data in the held out test set (so that we obtained a P-value for each UK Biobank first occurrence code indicated in Table 1). Second, we computed the Hazards Ratio (HR) for the different thresholded percentiles (top 1%, 5%, 10%, and bottom 10% compared to the 40-60%) of $\boldsymbol{X}\hat{\beta}$. The Kaplan-Meier curves, representing the proportion of disease events at different age of an individual are shown in Figure 3. Here, we applied multiSnpnet-Cox to tens of responses, but we anticipate that given the computational complexity of the algorithm it should easily handle hundreds if not thousands of responses simultaneously, with the main limiting factor being the number of active variables chosen.

4 Discussion

The main computational bottleneck of our algorithm are matrix-matrix multiplications, which include in-memory multiplications in the proximal gradient step, and out-memory multiplications in the KKT checking step. We plan to utilize GPUs to accelerate these operations since they are particularly suitable for dense matrix multiplications. The challenge here is the limited GPU memory. When the matrices needed for the proximal step do not fit in GPU memory, frequent communication between the system memory and the GPU memory could significantly slow down the algorithm. While advances in hardware (multiple GPUs, fast communication networks) can partially solve this problem, we hope to find algorithms that minimize slow communications. In our current implementation we use single precision floating point number for KKT checking. We plan to use single precision floating point number for all GPU operations.

In our Sparse-Group penalty, the groups are defined as the effect of the same predictor on all the responses. In practice it is often useful to define more general groups (e.g. genetic variants that are known to affect a biological mechanism that are associated with some responses). While general grouping does not change our algorithm by much, it could potentially introduce irregular memory access pattern, which could hurt performance. In future works, we hope to develop methods for this more general setting.

Finally, we freeze the parameters for a response when its coefficients start to overfitting. This proves to be quite effective in preventing irrelevant variables from entering the model, which helps both speed and validation metrics of the other responses. However, this step is not mathematically well-justified. In particular, the proximal operator changes whenever we constrain some parameters to stay at its previous value. We hope to resolve this issue in future works.

ID	descriptions	ncase	nselected	C0	C1	C2
130696	other hypothyroidism	34211	10287	0.660	0.750	0.752
130708	non-insulin-dependent diabetes mellitus	33259	9892	0.593	0.660	0.657
130714	unspecified diabetes mellitus	24129	6691	0.586	0.663	0.655
130700	thyrotoxicosis [hyperthyroidism]	7138	1839	0.649	0.713	0.710
130698	other non-toxic goitre	4658	1459	0.662	0.716	0.718
130706	insulin-dependent diabetes mellitus	4425	1407	0.551	0.669	0.658
130718	other disorders of pancreatic internal secretion	3015	861	0.545	0.598	0.583
130736	ovarian dysfunction	2264	48	0.727	0.728	0.723
130704	other disorders of thyroid	2141	41	0.671	0.676	0.670
130722	hyperparathyroidism and other disorders of parathyroid gland	1762	360	0.602	0.621	0.623
130702	thyroiditis	1210	707	0.653	0.688	0.678
130726	hypofunction and other disorders of pituitary gland	1065	0	0.468	0.468	0.477
130734	other disorders of adrenal gland	961	178	0.550	0.541	0.549
130724	hyperfunction of pituitary gland	782	0	0.597	0.597	0.596
130688	other disorders involving the immune mechanism	584	306	0.467	0.524	0.570
130712	other specified diabetes mellitus	340	0	0.574	0.574	0.531

Table 1: ncase is the number of observed disease events in the whole dataset. nselected is the number of variants selected by our algorithm. C0 is the test C-Index when we fit a Cox model with only the 11 covariates. C1 is the test C-Index of the `Multi-snpnet-Cox` solution. C2 is the test C-Index when the responses are analyzed independently using `snpnet` (Li et al. 2020)

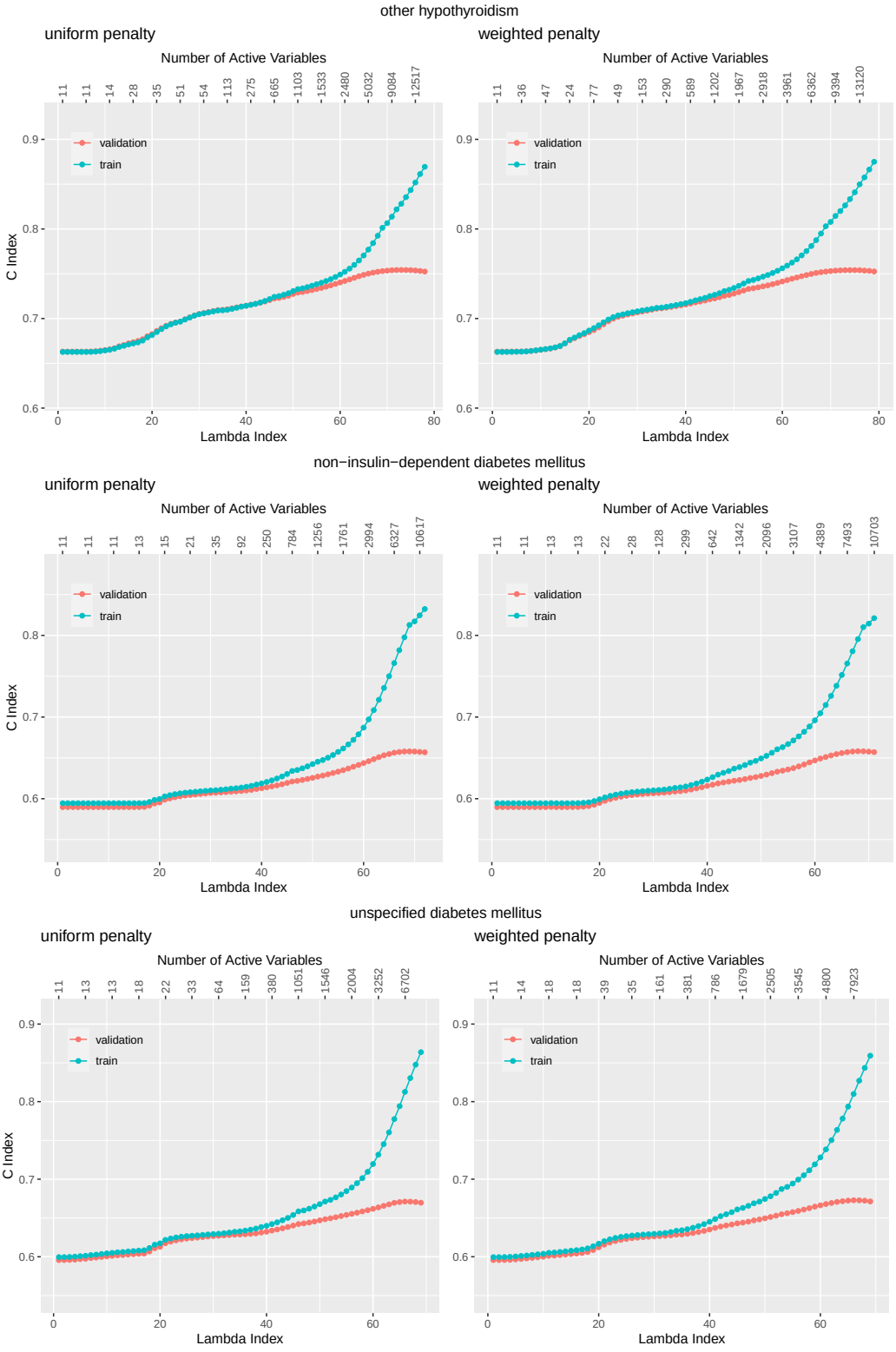

Figure 1: Lasso Path for a few responses. The top axis shows the number of active variables in the model corresponding to the lambda index (bottom). At each lambda the C index for the validation (red) and training set (aqua).

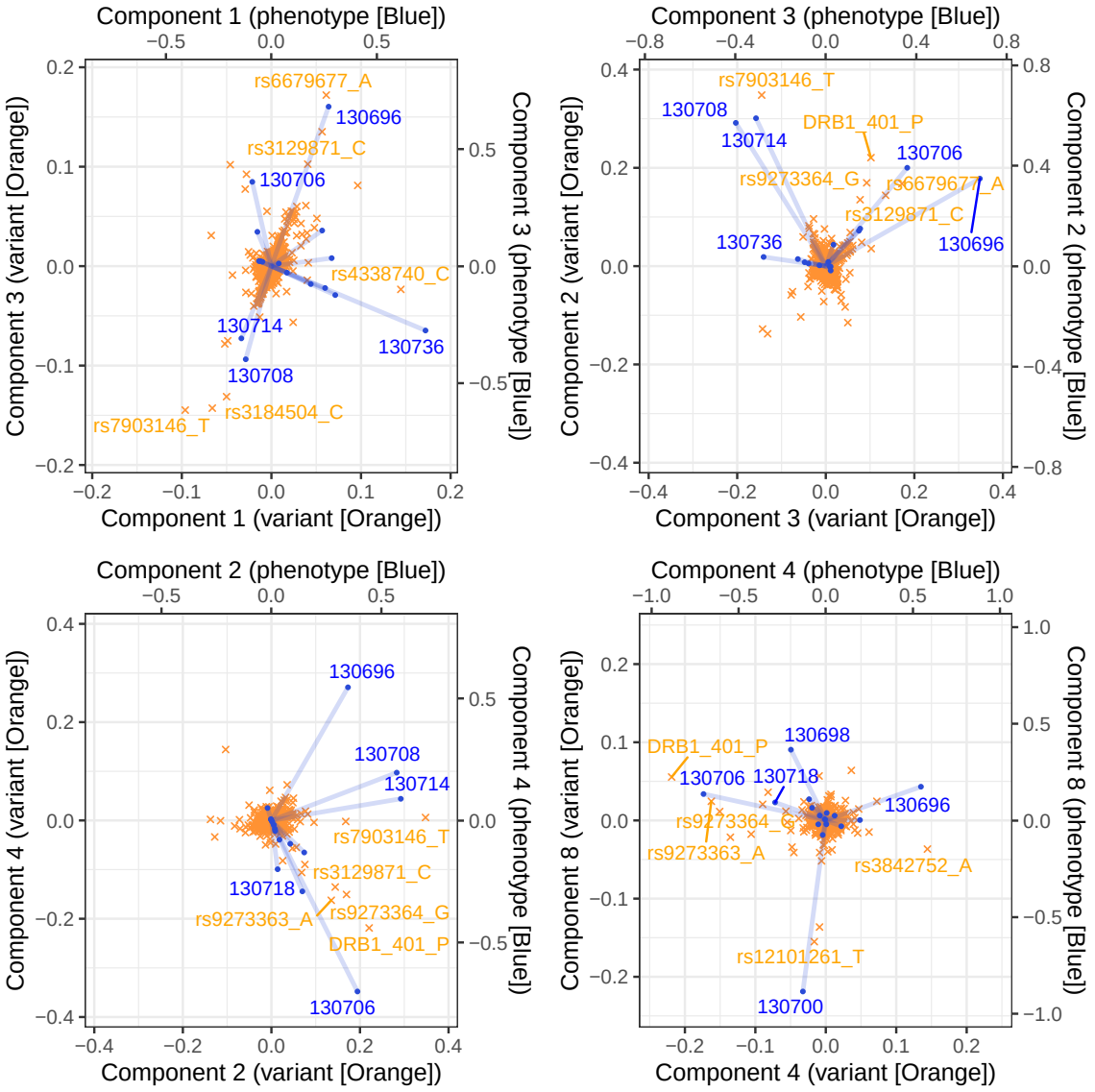

Figure 2: Five principal components of the estimated parameter matrix $\hat{B}$. The components are selected using trait squared cosine score described in Tanigawa et al. (2019), for the response 130696 (other hypothyroidism). These principal components (components 1, 3, 2, 4, and 8) are identified from an SVD of coefficient matrix $\hat{B} = UDV^T$ estimated using `multisnpnet-Cox` and shown as a series of biplots. In each panel, principal components of genetic variants (rows of UD) are shown in blue as scatter plot using the main axis and singular vectors of traits (rows of V) are shown in red dots with lines using the secondary axis, for the identified key components. The five traits and responses with the largest distance from the center of origin are annotated with their name. Table 1 provides the mapping of phenotype IDs to description.

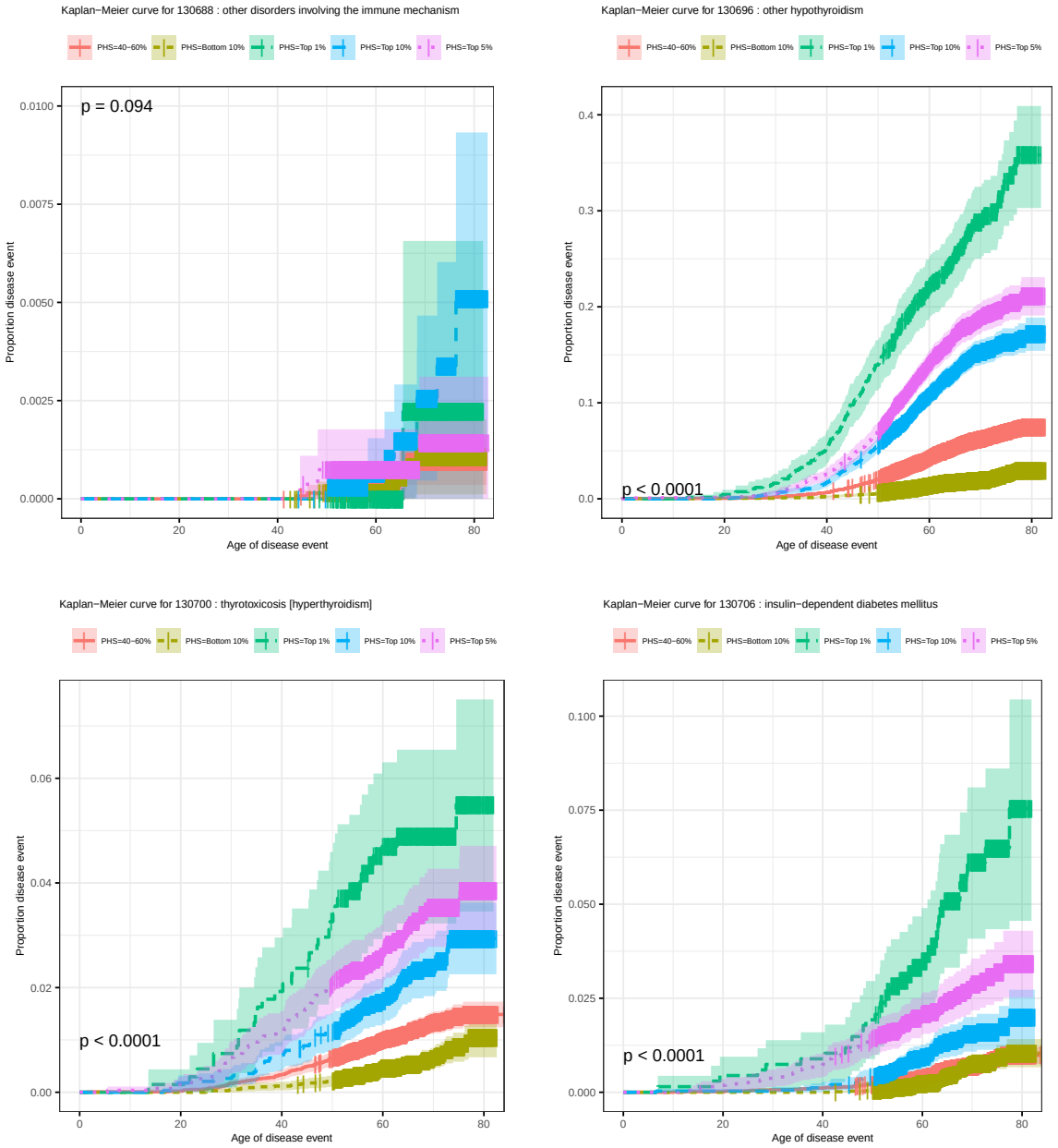

Figure 3: Kaplan-Meier curves for percentiles of Polygenic Hazard Scores for variants selected by `multiSnpnet-Cox`, in the held out test set (green - top 1%, purple - top 5%, blue - top 10%, red - 40-60%, and brown - bottom 10%; ticks represent censored observations. Curves for (top left) other disorders involving the immune mechanism, (top right) other hypothyroidism, (bottom left) thyrotoxicosis [hyperthyroidism], and (bottom right) insulin-dependent diabetes mellitus are shown.

References

Beck, A. & Teboulle, M. (2009), 'A fast iterative shrinkage-thresholding algorithm for linear inverse problems', *SIAM J. Img. Sci.* **2**(1), 183–202.
URL: *https://doi.org/10.1137/080716542*

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

Daubechies, I., Defrise, M. & De Mol, C. (2004), 'An iterative thresholding algorithm for linear inverse problems with a sparsity constraint', *Communications on Pure and Applied Mathematics* **57**(11), 1413–1457.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.20042*

DeBoever, C., Tanigawa, Y., Lindholm, M. E., McInnes, G., Lavertu, A., Ingelsson, E., Chang, C., Ashley, E. A., Bustamante, C. D., Daly, M. J. et al. (2018), 'Medical relevance of protein-truncating variants across 337,205 individuals in the uk biobank study', *Nature communications* **9**(1), 1–10.

Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L. & Rosati, R. A. (1982), 'Evaluating the yield of medical tests', *Jama* **247**(18), 2543–2546.

Kane, M., Emerson, J. & Weston, S. (2013), 'Scalable strategies for computing with massive data', *Journal of Statistical Software, Articles* **55**(14), 1–19.
URL: *https://www.jstatsoft.org/v055/i14*

Li, R., Chang, C., Justesen, J. M., Tanigawa, Y., Qian, J., Hastie, T., Rivas, M. A. & Tibshirani, R. (2020), 'Fast lasso method for large-scale and ultrahigh-dimensional cox model with applications to uk biobank', *bioRxiv* .
URL: *https://www.biorxiv.org/content/early/2020/01/21/2020.01.20.913194*

Nesterov, Y. (1983), A method for solving the convex programming problem with convergence rate $O(1/k^2)$.

Qian, J., Du, W., Tanigawa, Y., Aguirre, M., Tibshirani, R., Rivas, M. A. & Hastie, T. (2019), 'A fast and flexible algorithm for solving the lasso in large-scale and ultrahigh-dimensional problems', *bioRxiv* .
URL: *https://www.biorxiv.org/content/early/2019/05/07/630079*

Qian, J., Tanigawa, Y., Li, R., Tibshirani, R., Rivas, M. A. & Hastie, T. (2020), 'Large-scale sparse regression for multiple responses with applications to uk biobank', *BioRxiv* .

Rivas, M. A., Pirinen, M., Conrad, D. F., Lek, M., Tsang, E. K., Karczewski, K. J., Maller, J. B., Kukurba, K. R., DeLuca, D. S., Fromer, M. et al. (2015), 'Effect of predicted protein-truncating genetic variants on the human transcriptome', *Science* **348**(6235), 666–669.

Simon, N., Friedman, J., Hastie, T. & Tibshirani, R. (2013), 'A sparse-group lasso', *Journal of Computational and Graphical Statistics* **22**(2), 231–245.
URL: *https://doi.org/10.1080/10618600.2012.681250*

Sudlow, C., Gallacher, J., Allen, N., Beral, V., Burton, P., Danesh, J., Downey, P., Elliott, P., Green, J., Landray, M., Liu, B., Matthews, P., Ong, G., Pell, J., Silman, A., Young, A., Sprosen, T., Peakman, T. & Collins, R. (2015), 'Uk biobank: An open access resource for identifying the causes of a wide range of complex diseases of middle and old age', *PLOS Medicine* **12**(3), 1–10.
URL: *https://doi.org/10.1371/journal.pmed.1001779*

Tanigawa, Y., Li, J., Justesen, J. M., Horn, H., Aguirre, M., DeBoever, C., Chang, C., Narasimhan, B., Lage, K., Hastie, T. et al. (2019), 'Components of genetic associations across 2,138 phenotypes in the uk biobank highlight adipocyte biology', *Nature communications* **10**(1), 1–14.

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

Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J. & Tibshirani, R. J. (2012), 'Strong rules for discarding predictors in lasso-type problems', *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **74**(2), 245–266.
URL: *http://www.jstor.org/stable/41430939*

5 Appendix

5.1 Proof of Proposition 1

We show a slightly more general result from convex analysis. Let $f : \mathbb{R}^K \mapsto \mathbb{R}$ be continuously differentiable, convex, and bounded from below. Let $\|\cdot\|$ be a norm on $\mathbb{R}^K$, and $\|\cdot\|_*$ be its corresponding dual norm. Let $\lambda > 0$, and set

$$x^* := \arg\min_x f(x) + \lambda\|x\| \tag{6}$$

We will show that

$$\|\nabla f(x^*)\|_* \begin{cases} \leq \lambda & \text{if } x^* = 0 \\ = \lambda & \text{if } x^* \neq 0 \end{cases}. \tag{7}$$

It is clear that (6) is equivalent to the constrained optimization problem

$$\min_{x,y} f(x) + \lambda\|y\| \text{ such that } x = y. \tag{8}$$

The Lagrangian is

$$\mathcal{L}(x, y, z) = f(x) + \lambda\|y\| + z^T(y - x). \tag{9}$$

The Lagrangian dual is

$$g(z) := \inf_{x,y} \mathcal{L}(x, y, z) = \inf_y \lambda\|y\| + z^T y + \inf_x f(x) - z^T x. \tag{10}$$

Using the definition of dual norm, when $\|z\|_* > \lambda$, the infimum of the first term above is $-\infty$, and when $\|z\|_* \leq \lambda$, the infimum is 0. Therefore

$$g(z) = \begin{cases} -\infty & \text{if } \|z\|_* > \lambda \\ \inf_x f(x) - z^T x & \text{if } \|z\|_* \leq \lambda \end{cases} \tag{11}$$

Therefore the dual solution $z^* := \arg\max_z g(z)$ must satisfy $\|z\|_* \leq \lambda$. Now we go back to the Lagrangian. Since the primal objective is convex and Slaters condition holds, the solution to the primal problem can be obtained through minimizing

$$\mathcal{L}(x, y, z^*) = f(x) + \lambda\|y\| + {z^*}^T(y - x). \tag{12}$$

which implies that, at the optimal x^* we must have $\nabla f(x^*) = z^*$, so

$$\|\nabla f(x^*)\|_* = \|z^*\|_* \leq \lambda, \text{ and } \lambda\|x^*\| + \nabla f(x^*)^T x^* = 0 \tag{13}$$

If $x^* = 0$, then the second equality is already satisfied, so we only need $\|\nabla f(x^*)\|_* \leq \lambda$. If $x^* \neq 0$, then by Holder's inequality

$$\lambda \|x^*\| = |\nabla f(x^*)^T x^*| \leq \|\nabla f(x^*)\|_* \|x^*\| \leq \lambda \|x^*\|, \tag{14}$$

so we must have $\|\nabla f(x^*)\|_* = \lambda$. Proposition 1 is a direct consequence of this result.

5.2 Proof of Proposition 2

We prove the claim for $\lambda = 1$. The regularization term can be written as

$$\|u\|_1 + \alpha \|u\|_2 = \sup_{\|v_1\|_\infty \leq 1} v_1^T u + \sup_{\|v_2\|_2 \leq \alpha} v_2^T u = \sup_{v \in \mathcal{B}_\alpha^*} v^T u \tag{15}$$

where $\mathcal{B}_\alpha^*$ is the unit dual ball

$$\mathcal{B}_\alpha^* := \{v_1 \in \mathbb{R}^K : \|v_1\|_\infty \leq 1\} \oplus \{v_2 \in \mathbb{R}^K : \|v_2\|_2 \leq \alpha\}. \tag{16}$$

That is $\|v\|_{\alpha^*} \leq 1$ if and only if $v \in \mathcal{B}_\alpha^*$, which by definition means that $v = v_1 + v_2$ for some $\|v_1\|_\infty \leq 1, \|v_2\|_2 \leq \alpha$. We must have (and it is sufficient to have)

$$\inf_{\|v_1\|_\infty < 1} \|v - v_1\|_2 \leq \alpha. \tag{17}$$

The infimum on the left-hand side is achieved when $v_1 = v - S_1(v, 1)$, which proves the claim.